

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 9**

**Título: Sistema de Gestión de  
Procesos para la Dirección de  
Televisión Universitaria.**



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

**Autor:**

**Yusbel Ramos Núñez.**

**Tutor:**

**Ing. Yuset Martínez Carballo.**

**Ciudad de la Habana, Junio de 2009.**

**“Año del 50 Aniversario del Triunfo de la Revolución”**

**DEDICO MI TRABAJO DE DIPLOMA, DONDE HE PUESTO TODO MI ESFUERZO Y  
DEDICACIÓN, A: MI FAMILIA Y A TODAS AQUELLAS PERSONAS QUE SE  
PREOCUPAN POR MÍ.**

## AGRADECIMIENTOS.

*A todas aquellas personas que de alguna forma han hecho posible la realización de este trabajo.*

*A toda mi familia por depositar en mí plena confianza y apoyarme en todos estos años. Sepan todos que fueron mi mayor inspiración para lograr hacer realidad este trabajo y obtener un título.*

*A mi Mamá (Nereyda) y Mami (Juanita), por depositar en mí toda su confianza y apoyarme en todo momento.*

*A mi tía Rafaela que es mi tercera madre, por su incondicional apoyo.*

*A mi Papá (Eugenio), por sentirse orgulloso de mí.*

*A mis abuelos Papi (Rafael), Mongo (Ramón) y Fela (Rafaela), por ver en mí un sueño hecho realidad.*

*A mis tíos Vismar, Anita, Tony, Bolo, Bertha, Ñequi, Rey, Alejandro y Victoria, por creer en mí.*

*A mis primos Yuniel, Alexis, Yoba, Gallo, Aliuska, Lianni, Lianet, Lismeuris, Nilda, Neyda, Adiel, Reiniel, por ser parte de mi vida.*

*A los vecinos y amigos del barrio, por su preocupación.*

*A mis colegas de la zona Vanega, Odannis, Adriana, Daynier, Ana, Yuliet, Dolores, Reynaldo por ser como son. En especial a Dennis por su apoyo en los momentos buenos y malos y llegar a ser un amigo de los que ya no vienen asesorado por Suammy.*

*Le agradezco a mis conocidos y compañeros que han compartido conmigo estos 5 años desde el (9108) y me han extendido su mano cuando me ha hecho falta, los conocidos son aquellos que están tan cerca de la amistad que “a veces” se les confunde con amigos.*

*A los conocidos Ivanni, Yoangel, Geo, Frank, Angel, Raudo, Chirino, Raúl, Hugo, Yudiel, Liuba, Ekaterina, Yadira, Lleney, Sisi, Carelis, Mailén, Yisel y Suset por haberme soportado como soy.*

*A otros amigos, de la UCI a Gilberto, Mabel, Fito, Katia, Mercedes, Yaitza, Juanita, Estrella, Yucely, Yordán, Yamy, Dunier, Ariadna y fuera de la UCI a Zenén y Dulce, a todos gracias por darme un voto de confianza.*

*A mi tutor Yuset por asesorarme en todo.*

*A mis compañeros del proyecto.*

*A mis Profesores por haber aportado su granito de arena en el desarrollo de esta carrera.*

*Si queda alguien, sabe que mi agradecimiento será de corazón aunque no esté en esta hoja...*

*A todos **MUCHAS GRACIAS** por existir.*

Yusbel Ramos Núñez, declaro que soy el único autor de este trabajo y autorizo a la Dirección de Televisión Universitaria de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

---

Autor: Yusbel Ramos Núñez

---

Tutor: Ing. Yuset Martínez Carballo

**Tutor:**

**Nombre y apellidos:** Yuset Martínez Carballo.

**Sexo:** M.      **Institución:** Universidad de las Ciencias Informáticas (UCI).

**Dirección de la institución:** Carretera a San Antonio de los Baños, Km. 2 ½, Reparto Torrens,  
Municipio Boyeros, Provincia Ciudad de La Habana.

**Correo electrónico:** [ycarballo@uci.cu](mailto:ycarballo@uci.cu)

**Teléfono del trabajo:** 07-835-8284.

**Categoría docente:** Adiestrado.

**Cargo del trabajador:** Profesor.

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas.

**Año de graduación:** 2008.

**Institución donde se graduó:** Universidad de las Ciencias Informáticas (UCI).

Desde los inicios de la Universidad de las Ciencias Informáticas (UCI), debido a la gran cantidad de personal que en ella participa, su Dirección se dio a la tarea de buscar un medio interno para mantener a todos informados, valorando que nada mejor que los medios tradicionales de comunicación para lograr este objetivo; es así como surge la Dirección de Televisión Universitaria (DTU). Luego, con las experiencias adquiridas en el país con el uso de la televisión para la masificación de la cultura y la implementación por parte del Ministerio de Educación del Sistema de Tele clases en la televisión nacional, se apreció la idea de crear un medio que llevara a cabo este programa de manera interna, no sólo quedándose en la rama de la docencia sino promoviéndose al esparcimiento, del cual se tiene como ejemplo el Canal Cultural y más tarde la transmisión por el sistema de televisión interno de la Radio Universitaria, representada por el proyecto Radio Ciudad Digital.

Luego del desarrollo de la Universidad y la creación de nuevas facilidades en lo que a medios de comunicación se refiere, en este caso muy particular la televisión, el cúmulo de informaciones derivadas de estas actividades se hace cada vez mayor y difícil de controlar en la forma tradicional, por lo cual, en un momento dado se torna difícil buscar algún tipo de información necesaria guardada en archivo y con ello el incumplimiento del programa del sistema de comunicación televisiva.

Lo anteriormente expuesto necesita de medidas inmediatas para su solución con vistas a lograr mayor fortaleza en el sistema de comunicación televisiva, por lo cual se plantea la implementación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, dejando las bases creadas para la implementación de éste, en otro marco dentro o fuera del país. Este trabajo persigue como objetivo principal dotar a la DTU de un sistema que le permita controlar de forma centralizada todos los procesos de información que se llevan a cabo en la misma.

## **PALABRAS CLAVES**

Sistema de Gestión de Procesos.

Dirección de Televisión Universitaria.

## ÍNDICE DE FIGURAS

FIGURA 1: MODELO CLIENTE/SERVIDOR. ....	7
FIGURA 2: LENGUAJES DE PROGRAMACIÓN WEB. ....	13
FIGURA: 3 ARQUITECTURA MODELO VISTA CONTROLADOR. ....	18
FIGURA 4: FASES Y FLUJOS DE TRABAJO EN RUP. ....	25
FIGURA: 5 DIAGRAMA DE COMPONENTES POR PAQUETES. ....	56
FIGURA 6: PAQUETE ACTIONS. ....	57
FIGURA 7: PAQUETE SUCCESS. ....	58
FIGURA 8: PAQUETE DE ACCESO A DATOS. ....	59
FIGURA 9: ARCHIVOS GENERADOS POR SYMFONY A PARTIR DE LA BASE DE DATOS. ....	60
FIGURA 10: PAQUETE DE SYMFONY. ....	60
FIGURA 11: FORMULARIOS GENERADOS POR SYMFONY. ....	61
FIGURA 12: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR USUARIOS. ....	62
FIGURA 13: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR ROL. ....	63
FIGURA 14: DIAGRAMA DE COMPONENTES DEL CASO DE USO AUTENTICAR. ....	64
FIGURA 15: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR PLAN DE PRODUCCIÓN. ....	65
FIGURA 16: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR MEDIA. ....	66
FIGURA 17: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR EQUIPAMIENTO. ....	67
FIGURA 18: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR PLAN DE PROGRAMACIÓN. ....	68
FIGURA 19: DIAGRAMA DE COMPONENTES DEL CASO DE USO GESTIONAR SOLICITUD DE MEDIAS. ....	69
FIGURA: 20 DIAGRAMA DE DESPLIEGUE. ....	105

## ÍNDICE DE TABLAS

TABLA 1: DESCRIPCIÓN DEL CU GESTIONAR USUARIO. ....	28
TABLA 2: DESCRIPCIÓN DEL CU GESTIONAR ROL. ....	31
TABLA 3: DESCRIPCIÓN DEL CU AUTENTICAR USUARIO. ....	34
TABLA 4: DESCRIPCIÓN DEL CU GESTIONAR PLAN DE PRODUCCIÓN. ....	36
TABLA 5: DESCRIPCIÓN DEL CU GESTIONAR MEDIA. ....	39
TABLA 6: DESCRIPCIÓN DEL CU GESTIONAR EQUIPAMIENTO. ....	43
TABLA 7: DESCRIPCIÓN DEL CU GESTIONAR PLAN DE PROGRAMACIÓN. ....	47
TABLA 8: CASO DE PRUEBA ADICIONAR USUARIO. ....	109
TABLA 9: CASO DE PRUEBA MODIFICAR USUARIO. ....	109
TABLA 10: CASO DE PRUEBA DETALLES DE USUARIO. ....	110
TABLA 11: CASO DE PRUEBA ELIMINAR USUARIO. ....	110
TABLA 12: CASO DE PRUEBA ADICIONAR ROL. ....	111
TABLA 13: CASO DE PRUEBA MODIFICAR ROL. ....	111
TABLA 14: CASO DE PRUEBA DETALLES DEL ROL. ....	112
TABLA 15: CASO DE PRUEBA ELIMINAR ROL. ....	112
TABLA 16: CASO DE PRUEBA ADICIONAR FOTO. ....	113
TABLA 17: CASO DE PRUEBA MODIFICAR FOTO. ....	114

TABLA 18: CASO DE PRUEBA DETALLES DE LA FOTO. ....	114
TABLA 19: CASO DE PRUEBA ELIMINAR FOTO. ....	115
TABLA 20: CASO DE PRUEBA ADICIONAR SOLICITUD DE MEDIA. ....	115
TABLA 21: CASO DE PRUEBA MODIFICAR SOLICITUD DE MEDIA. ....	116
TABLA 22: CASO DE PRUEBA DETALLES DE LA SOLICITUD DE MEDIA. ....	116
TABLA 23: CASO DE PRUEBA ADICIONAR EQUIPAMIENTO. ....	117
TABLA 24: CASO DE PRUEBA MODIFICAR EQUIPAMIENTO. ....	117
TABLA 25: CASO DE PRUEBA DETALLES DEL EQUIPAMIENTO. ....	118
TABLA 26: CASO DE PRUEBA ASIGNAR EQUIPAMIENTO. ....	118
TABLA 27: CASO DE PRUEBA CAMBIAR ASIGNACIÓN DEL EQUIPAMIENTO. ....	119
TABLA 28: CASO DE PRUEBA ELIMINAR EQUIPAMIENTO. ....	119
TABLA 29: CASO DE PRUEBA ADICIONAR PLAN DE PROGRAMACIÓN. ....	120
TABLA 30: CASO DE PRUEBA MODIFICAR PLAN DE PROGRAMACIÓN. ....	120
TABLA 31: CASO DE PRUEBA DETALLES DEL PLAN DE PROGRAMACIÓN. ....	121
TABLA 32: CASO DE PRUEBA ADICIONAR PROGRAMACIÓN A UN PLAN DE PROGRAMACIÓN. ....	121
TABLA 33: CASO DE PRUEBA MODIFICAR PROGRAMACIÓN DE UN PLAN DE PROGRAMACIÓN. ....	122
TABLA 34: CASO DE PRUEBA DETALLES DE PROGRAMACIÓN DE UN PLAN DE PROGRAMACIÓN. ....	123
TABLA 35: CASO DE PRUEBA ELIMINAR PROGRAMACIÓN DE UN PLAN DE PROGRAMACIÓN. ....	123
TABLA 36: CASO DE PRUEBA ELIMINAR PLAN DE PROGRAMACIÓN. ....	123
TABLA 37: CASO DE PRUEBA ADICIONAR PLAN DE PRODUCCIÓN. ....	124
TABLA 38: CASO DE PRUEBA MODIFICAR PLAN DE PRODUCCIÓN. ....	124
TABLA 39: CASO DE PRUEBA DETALLES DEL PLAN DE PRODUCCIÓN. ....	125
TABLA 40: CASO DE PRUEBA ADICIONAR ACTIVIDAD DE UN PLAN DE PRODUCCIÓN. ....	125
TABLA 41: CASO DE PRUEBA MODIFICAR ACTIVIDAD DE UN PLAN DE PRODUCCIÓN. ....	126
TABLA 42: CASO DE PRUEBA DETALLES DE UNA ACTIVIDAD DE UN PLAN DE PRODUCCIÓN. ....	127
TABLA 43: CASO DE PRUEBA ELIMINAR ACTIVIDAD DE UN PLAN DE PRODUCCIÓN. ....	127
TABLA 44: CASO DE PRUEBA ELIMINAR PLAN DE PRODUCCIÓN. ....	127

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
Fundamentación Metodológica. ....	4
<b>CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 Introducción. ....	5
1.2 Sistema de gestión. ....	5
¿Qué es un sistema de gestión?.....	5
¿Por qué son necesarios los sistemas de gestión? .....	5
1.3 Tendencias y tecnologías actuales utilizadas en los sistemas de gestión. ....	6
Modelo Cliente/Servidor.....	7
Aplicaciones Web. ....	9
1.4 Sistemas de gestión similares. ....	11
Internacional. ....	11
1.5 Arquitectura y patrones de diseño definidos para la implementación del Sistema de Gestión de Procesos de la Dirección de Televisión Universitaria. ....	12
1.5.1 Lenguajes de Programación Web. ....	12
1.5.1.1 Lenguaje del lado del servidor.....	13
PHP5. ....	14
1.5.1.2 Lenguaje del lado del cliente. ....	15
HTML. ....	15
JavaScript. ....	16
1.5.2 Servidores Web.....	16
Servidor HTTP Apache 2.0. ....	16
1.5.3 Lenguaje Unificado de Modelado. ....	17
1.5.4 Patrón Arquitectónico.....	18
Modelo Vista Controlador. ....	18
1.5.5 Framework. ....	20
Symfony.....	20
1.5.6 Sistema Operativo.....	22
Ubuntu. ....	22
1.5.7 Entorno Integrado de Desarrollo. ....	23
Eclipse PDT. ....	23
1.5.8 Metodología de desarrollo de software. ....	24
Proceso Unificado de Desarrollo.....	24
1.6 Conclusiones.....	26
<b>CAPÍTULO II. IMPLEMENTACIÓN DEL SISTEMA.</b> .....	<b>27</b>
2.1 Introducción. ....	27
2.2 Descripción de los requisitos funcionales.....	27
2.3 Estándares de codificación. ....	53
2.4 Diagramas de componentes. ....	55
2.5 Código fuente. ....	70
2.5.1 Módulo Seguridad (Caso de uso Autenticar). ....	70
2.5.2 Módulo Administración (Caso de uso Gestionar Usuario). ....	71
2.5.3 Módulo de Administración (Caso de uso Gestionar Rol).....	75
2.5.4 Módulo de Administración (Caso de uso Gestionar Media (Foto)).....	78
2.5.5 Módulo de Administración (Caso de uso Gestionar Solicitud de Media). ....	85
2.5.6 Módulo de Administración (Caso de uso Gestionar Equipamiento). ....	90

2.5.7	Módulo de Administración (Caso de uso Gestionar Plan de Programación)	96
2.5.8	Módulo de Administración (Caso de uso Gestionar Plan de Producción)	100
2.6	Diagrama de despliegue	104
2.7	Conclusiones	105
<b>CAPÍTULO III. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA</b>		<b>107</b>
3.1	Introducción	107
3.2	Casos de pruebas	108
3.2.1	Caso de prueba Autenticar	108
3.2.2	Caso de prueba Gestionar Usuario	109
3.2.2.1	Adicionar usuario	109
3.2.2.2	Modificar usuario	109
3.2.2.3	Detalles del usuario	110
3.2.2.4	Eliminar Usuario	110
3.2.3	Caso de prueba Gestionar Rol	111
3.2.3.1	Adicionar rol	111
3.2.3.2	Modificar rol	111
3.2.3.3	Detalles del rol	112
3.2.3.4	Eliminar rol	112
3.2.4	Caso de prueba Gestionar Media	113
3.2.4.1	Adicionar foto	113
3.2.4.2	Modificar foto	114
3.2.4.3	Detalles de la foto	114
3.2.4.4	Eliminar foto	115
3.2.5	Caso de prueba Gestionar Solicitud de Media	115
3.2.5.1	Adicionar solicitud de media	115
3.2.5.2	Modificar solicitud de media	116
3.2.5.3	Detalles de una solicitud de media	116
3.2.6	Caso de prueba Gestionar Equipamiento	117
3.2.6.1	Adicionar equipamiento	117
3.2.6.2	Modificar equipamiento	117
3.2.6.3	Detalles del equipamiento	118
3.2.6.4	Asignar equipamiento	118
3.2.6.5	Cambiar asignación del equipamiento	119
3.2.6.6	Eliminar equipamiento	119
3.2.7	Caso de prueba Gestionar Plan de Programación	119
3.2.7.1	Adicionar plan de programación	119
3.2.7.2	Modificar plan de programación	120
3.2.7.3	Detalles del plan de programación	121
3.2.7.4	Adicionar programación a un plan de programación	121
3.2.7.5	Modificar programación de un plan de programación	122
3.2.7.6	Detalles de programación de un plan de programación	123
3.2.7.7	Eliminar programación de un plan de programación	123
3.2.7.8	Eliminar plan de programación	123
3.2.8	Caso de prueba Gestionar Plan de Producción	124
3.2.8.1	Adicionar plan de producción	124
3.2.8.2	Modificar plan de producción	124
3.2.8.3	Detalles del plan de producción	125
3.2.8.4	Adicionar actividad a un plan de producción	125

3.2.8.5	Modificar actividad de un plan de producción.....	126
3.2.8.6	Detalles de actividad de un plan de producción.....	127
3.2.8.7	Eliminar actividad de un plan de producción.....	127
3.2.8.8	Eliminar plan de producción.....	127
3.3	Conclusiones.....	128
<b>CONCLUSIONES.....</b>		<b>129</b>
<b>RECOMENDACIONES .....</b>		<b>130</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>131</b>
<b>BIBLIOGRAFÍA.....</b>		<b>133</b>
<b>GLOSARIO DE TÉRMINOS.....</b>		<b>136</b>

## INTRODUCCIÓN

---

En las circunstancias actuales como parte del desarrollo informático que se está llevando a cabo en el país, la Ciudad Digital, como avanzada, realiza un gran esfuerzo para que todos o la mayoría de sus procesos, estén automatizados, dentro de lo cual se pueden encontrar los servicios prestados por la Dirección de Televisión Universitaria con la implantación del sistema de televisión interna en la Universidad de las Ciencias Informáticas.

Esta Dirección fue creada en los mismos inicios de la Universidad debido a la necesidad de un sistema de comunicación que abarcara el campus universitario y llevara a ellos en principio un conjunto de informaciones importantes para el desarrollo de la vida y la convivencia dentro de la Institución; inmediatamente esta idea fue desarrollándose y abarcando otras ramas de suma importancia como la docencia.

Tomando el ejemplo del sistema empleado por el país para la masificación de la cultura, basado fundamentalmente en la explotación de forma provechosa de los diferentes medios de comunicación masiva, poniéndose en práctica la transmisión de tele clases que contribuyen a la formación académica de los estudiantes. El proyecto fue un poco más ambicioso y logró la puesta en marcha del Canal Cultural, que es una manera sana de esparcimiento y relajación brindada a la comunidad universitaria a través de la televisión por cable implementada en esta; además se implementó un espacio para la radio con el proyecto Radio Ciudad Digital (RCD), tanto en la web como en el sistema de televisión interna, ha tenido gran acogida por parte de la comunidad universitaria.

Después de lo anteriormente expuesto es de vital importancia que el flujo de información que opera la DTU, sea tratada de forma adecuada para su consulta en cualquier momento o lugar; el gran flujo de información que es manipulada dentro de sus departamentos de una forma descentralizada, hace difícil un control de todas las informaciones, por lo que se hace necesaria la implementación de un sistema que permita el control de las mismas de forma centralizada, logrando que todo el cúmulo de información que aquí se maneje, logrando el buen funcionamiento de la institución y con ello cumplir las expectativas que deposita en ella la universidad en general. En la actualidad uno de los recursos más utilizados en el mundo es la red de redes o internet, la cual brinda una gran cantidad de herramientas desarrolladas por

terceros, factibles a utilizar para el desarrollo de sistemas de gestión de información; aprovechando estas herramientas, con la realización de este sistema se lograría la confluencia de todos los datos manejados en los departamentos de la DTU y con ello un mejor servicio por parte de ésta a la UCI.

Concretamente la **situación problemática** existente hoy en la DTU se debe a la inexistencia de un sistema que gestione centralizadamente todos los flujos de información que en ella se manejan, conllevando a la mala gestión de la información. Entonces así es que surge el **problema científico**. *¿Cómo lograr un sistema que permita la gestión de los procesos que se realizan en los departamentos: Dirección, Producción, Informativo y Programación de la Dirección de Televisión Universitaria?*

Siendo así, el **objeto de estudio** asociado a este problema es: *los procesos que se desarrollan en los distintos departamentos de la DTU para garantizar un buen servicio.*

Luego se define que el **campo de acción** es *el sistema automatizado que gestiona los procesos existentes en la DTU.*

Entonces llegamos a la conclusión de que la investigación tiene como **objetivo general** *la implementación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria.*

Para el logro del cumplimiento del objetivo general, se hace necesario el planteamiento de dos **objetivos específicos**, el primero *implementar los módulos de cada uno de los departamentos Dirección, Producción, Informativo y Programación;* y en segundo lugar *implementar el Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria como un único sistema de gestión,* con lo cual se le daría solución al objetivo general.

Luego de plantear los objetivos tanto general como específicos se logra definir que la **hipótesis** del problema planteado es: *con la creación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, se lograría una mejor fluidez de la información que se manipula en todos los departamentos y se agilizarían así los procesos de la DTU.*

Para el cumplimiento de los objetivos propuestos se desarrollarán las siguientes **tareas**:

- *Identificación de las distintas técnicas de programación que existen a nivel internacional, nacional y de la Universidad.*
- *Valoración de las tendencias, técnicas, tecnologías y metodologías relacionadas con las técnicas de programación actuales, así como de las plataformas de desarrollo que la soportan.*
- *Construcción del Modelo de Implementación.*
- *Definición de un estándar de codificación.*
- *Implementación del sistema diseñado.*
- *Desarrollo del prototipo de interfaz de usuario.*
- *Desarrollo y mejora de algoritmos para el procesamiento de información.*
- *Integración con otros componentes o partes del sistema.*
- *Desarrollo de Casos de Prueba que certifiquen la veracidad de los algoritmos empleados.*
- *Desarrollo de los manuales de usuario y de instalación del sistema.*

Una vez que se ponga en marcha la realización de las tareas, estas permitirán la creación de **posibles resultados** dentro de los que se pueden nombrar:

- *Diagrama de componentes de los subsistemas.*
- *Modelo de Implementación.*
- *Documento de estándares de codificación.*
- *Documento de defectos encontrados.*
- *Prototipo funcional de los subsistemas.*

- *Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria.*

## Fundamentación Metodológica.

### Métodos teóricos.

Los métodos teóricos son aquellos que nos permiten estudiar las características del objeto de investigación, estos ayudan a la construcción de modelos e hipótesis de investigación, contribuyendo al desarrollo de las teorías científicas. Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa dada y su relación con otros fenómenos.

- **Analítico-sintético:** Será utilizado para el estudio preliminar del fenómeno, logrando distinguir los rasgos esenciales, creando un resumen de las principales características del mismo y así conocer lo que falta para lograr el objetivo planteado.

### Métodos empíricos.

Como parte de la investigación es necesario establecer el método de recolección de datos y el tipo de instrumento que se utilizará para ello. En esta fase, la investigación debe ser metódica y coherente a la vez, ya que la confección de un buen instrumento determina en gran medida la calidad de la información que permitirá, a su vez, obtener los datos necesarios para dar respuesta al problema planteado.

- **Observación:** Mediante la observación de las actividades realizadas por la DTU en la UCI, se logrará llegar a conclusiones de que será necesario implementar para obtener el sistema de gestión de procesos que centralice el flujo de información que en la DTU se maneja.
- **La entrevista:** Con este método se investigará cuál es la situación de la cual se va a partir para desarrollar el sistema de gestión de procesos de la DTU en la UCI.

## CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

---

### 1.1 Introducción.

En el presente capítulo se realizará una breve descripción de que es un sistema de gestión y que representa este para una institución se hace además un acercamiento a sistemas similares existentes en el mundo, así como las herramientas que fueron dispuestas para la implementación del mismo.

### 1.2 Sistema de gestión.

Cualquier institución hoy día debe contar con un sistema de gestión ordenado para lograr que la entidad funcione bien y esto es logrado sólo con una buena implementación y mantenimiento de éste; el sistema de gestión debe plantearse mejorar continuamente las prestaciones de la entidad, por medio de la identificación las necesidades de las partes interesadas.

#### ¿Qué es un sistema de gestión?

Un sistema de gestión es aquel que ayuda a mantener el buen funcionamiento de una entidad: “*Sistema para establecer la política y los objetivos para lograr dichos objetivos*” (6), logrando así la mejora continua de las estrategias, los procedimientos y procesos de una organización.

#### ¿Por qué son necesarios los sistemas de gestión?

Las empresas de hoy cuentan con una gran cantidad de procesos enfrentándose así a muchos desafíos, dentro de los cuales están:

- Rentabilidad.
- Velocidad de cambio.
- Capacidad de adaptación.
- Crecimiento.

- Tecnología.

Para lograr que todos estos desafíos se mantengan en total armonía, el proceso puede tornarse arduo y desalentador; es en ese momento en que entra a jugar su papel el sistema de gestión, permitiendo explotar al máximo el potencial existente en la institución.

Con la implementación de un sistema de gestión efectivo se puede lograr:

1. Reducir la carga de trabajo.
2. Reducir los costos.
3. Aumentar la satisfacción de los clientes y partes interesadas.
4. Lograr mejoras continuas en la producción.
5. Potenciar la innovación tecnológica.
6. Aportar claridad a los procesos desarrollados en la institución.

La utilización de un sistema de gestión le permite a la institución replantear continuamente sus objetivos, estrategias, operaciones y niveles de servicio.

### **1.3 Tendencias y tecnologías actuales utilizadas en los sistemas de gestión.**

Las tendencias actuales de las tecnologías apuntan al desarrollo acelerado de todos sus productos tengan en menor o mayor medida una vinculación estrecha con el mundo digital. Debido a la gran competencia que se ha llevado a cabo desde la última década del siglo veinte, las tecnologías se han desarrollado a tal punto, que las que van gestándose están prácticamente obligadas a utilizar medios informáticos para su puesta en explotación.

Actualmente las comunicaciones y la automatización entre las empresas han logrado avances significativos, permitiendo el aumento de la miniaturización de los sistemas, la portabilidad optimizada y la facilidad de uso de algunas herramientas de software; lo que ha permitido que el personal no

especializado lo utilice y obtenga de ellos los mejores resultados. Para el logro de estos avances se requirió de un mayor uso de la tecnología, reduciendo con ello el personal necesitado por las empresas, mientras que permite que se desarrollen procesos modernos y complejos, a la vez de la gerencia común y de la contabilidad.

### Modelo Cliente/Servidor.

Con el desarrollo de internet hoy día, se hace prácticamente imprescindible la utilización de esta autopista de información para el desarrollo de los distintos procesos que rigen el mundo, desde conversaciones telefónicas y transferencias bancarias hasta simples búsquedas de información o herramientas. Para ello la red de redes actualmente cuenta con una infraestructura basada fundamentalmente en el modelo Cliente/Servidor.

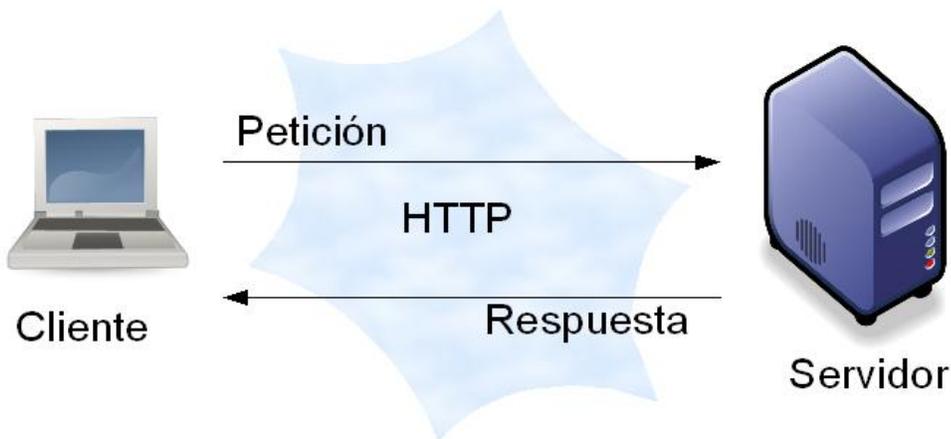


Figura 1: Modelo Cliente/Servidor.

La expresión Cliente se utiliza para referirse al ordenador que un usuario maneja para entrar a una red como por ejemplo Internet; desde dicho ordenador el usuario instaaura conexiones con otros ordenadores, nombrados ordenadores remotos, a los que se les solicita algún servicio. Estos ordenadores remotos que

ofrecen servicios, adoptan también el calificativo de servidores o hosts. Por lo tanto el modelo Cliente/Servidor concurre en la relación que existe entre estos dos ordenadores.

El empleo de las distintas aplicaciones o servicios brindados por Internet se lleva a cabo respondiendo al citado Modelo Cliente/Servidor. Cuando se emplea un servicio en Internet, como transferir un fichero, participar en un foro de discusión o examinar una base de datos, se establece un proceso en el que entran en juego varias partes. Por una parte, el usuario o máquina cliente, quien ejecuta una aplicación en el ordenador local, este programa cliente es el encargado de ponerse en contacto con el ordenador remoto para solicitar el servicio deseado; en el otro extremo se encuentra el ordenador remoto que responderá a lo solicitado por el programa cliente con otro denominado programa servidor.

El programa o programas cliente que el usuario utiliza para acceder a los servicios de Internet realizan dos funciones distintas. Primeramente, se encargan de gestionar la comunicación con el ordenador servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste y luego, es la herramienta que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las prestaciones que ofrece el servidor. Por ejemplo, cuando se utiliza el servicio de correo electrónico se pone de manifiesto este modelo, ya que el ordenador en el que vamos a revisar el correo (cliente) establece una conexión con el servidor de correos para poder descargar la información sobre los correos existentes y mostrárselos al usuario.

Cuba, no exenta del desarrollo mundial de la informática y las comunicaciones, se está sumando de manera paulatina al mismo, utilizando las técnicas que marcan puntera en estas, siendo la arquitectura Cliente/Servidor una de ellas, por lo cual es factible la utilización de este modelo en nuestro sistema de gestión.

*“La arquitectura Cliente/Servidor es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realiza se efectúa con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema.” (4)*

*“Características de la arquitectura Cliente/Servidor:*

- *Orientado a servicios. El servidor los ofrece y el cliente los consume.*

- *Compartición de recursos. Servicios ofrecidos a muchos clientes. Un servidor puede atender muchos clientes que solicitan esos servicios.*
- *Transparencia de ubicación. El servidor es un proceso que puede residir en el mismo ordenador que el cliente o en uno distinto a lo largo de una red. Un programa puede ser un servidor en un momento y convertirse en un cliente posteriormente.*
- *Mezcla e igualdad. Esta es una de las más importantes ventajas de este paradigma. Una aplicación cliente/servidor, idealmente es independiente del hardware y de sistemas operativos; mezclando e igualando estas plataformas.*
- *Interacción a través de mensajes, para envío y respuesta de servicios.*
- *Servicios encapsulados, exponiendo los servicios a través de interfaces, lo que facilita la sustitución de servidores sin afectar los clientes; permitiendo a la vez una fácil escalabilidad.” (2)*

### **Ventajas de la arquitectura Cliente/Servidor:**

Después de conocidas las características de la arquitectura Cliente/Servidor se puede distinguir claramente, que el servidor no debe estar dotado obligatoriamente de gran potencia en el procesamiento; parte del procesamiento la realizan los clientes, por lo cual el intercambio de información en la red disminuye considerablemente; el cliente sólo se conecta al servidor cuando es necesario, obteniendo los datos precisos y cierra su conexión disminuyendo así la sobrecarga en la red y en el servidor.

### **Aplicaciones Web.**

En un futuro muchas de las herramientas informáticas que usted va a precisar para escribir, editar, coleccionar y publicar archivos, estarán situadas en un servidor remoto y no en el disco duro de su ordenador local.

Hace varios años atrás que personas con mentalidad en el futuro se dieron cuenta de que el desarrollo de las tecnologías informáticas se basaría en el acceso a redes. La gran mayoría de los usuarios de Internet simplemente han visto pantallazos de esta nueva realidad a través del uso cada vez más crecientes de los

motores de búsqueda, instrumentos de social networking soportadas por la web, archivos de imágenes y herramientas de colaboración en línea. Pero para la mayoría de los usuarios todavía está fuertemente anclado a su filosofía el escritorio, generalmente viendo a Outlook como un programa equivalente al e-mail, Word como editor de textos y otras con fines diversos.

Las aplicaciones web poseen ventajas que logran que sus prestaciones sean muy utilizadas y difundidas en el mundo ya que requiere sólo de la instalación de esta en pocos ordenadores y a través de ellos los usuarios puedan hacer uso de ellas a través de las redes computacionales. Además, éstas son independientes del sistema operativo que utilice el ordenador del cliente y de sus capacidades (video, memoria, entre otros.) y sólo exigen un navegador de internet como por ejemplo Mozilla Firefox, Internet Explorer, entre otros, logrando que la aplicación este trabajando en el ordenador que la aloja (servidor), dado que puedes ejecutarlas desde cualquier ordenador con conexión a una red que de una forma u otra logre conectarse con el ordenador de la aplicación.

El flujo de información que se maneja se accede a través de las redes de internet, por lo cual son más apropiadas para desarrollar aplicaciones multiusuario partiendo de la base de la compartición de información. Además, el servidor debe estar preparado para ejecutar la aplicación de manera fluida para que los usuarios la utilicen de forma concurrente.

Existen muchas posibilidades para acceder a la información desde muchos sitios del mundo gracias a la red de redes o Internet, lo cual es un logro y a su vez un desafío para todas las personas que desarrollan la Web pues los constantes adelantos tecnológicos solicitan día tras día más rapidez, ligereza y robustez en las aplicaciones Web. La Web de hoy día es flexible y económica en lo referido a las comunicaciones, el acceso diario a información, servicios, entretenimiento, entre otros., estampan un gran valor económico y cultural a los usuarios que hacen uso de estas aplicaciones.

A partir de las tecnologías primarias: Lenguaje de Marcas de Hipertexto (*HyperText Markup Language*, HTML) y el Protocolo de Transferencia de Hipertexto (*HyperText Transfer Protocol*, HTTP) han emergido varias tecnologías innovadoras como: Interfaz de Entrada Común (*Common Gateway Interface*, CGI), Java, JavaScript, Páginas Activas del Servidor (*Active Server Pages*, ASP), Páginas de Hipertexto Pre procesadas (*Preprocessed Hypertext Pages*, PHP), Flash, Lenguaje de Marcado Extensible (*EXtensible*

*Markup Language*, XML), sólo por citar algunas de las más difundidas, que permiten la creación de una mejor Web, más extensa, más poderosa y más flexible o fácil de mantener.

Con la unión de manera coherente de las tecnologías antes mencionadas se logra una Aplicación Web que no es más que un sistema informático con el cual los usuarios intercambian con un servidor Web a través de las redes dependiendo del alcance que tenga acceso el usuario, estas son notorias debido a lo práctico de utilizar que se ha tornado en el transcurso de los años el navegador Web. La maestría que los desarrolladores utilizan para actualizar y mantener aplicaciones Web sin distribuir ni instalar software en miles de clientes potenciales es otra razón muy fuerte que determina en gran medida su popularidad. De forma práctica hay dos tipos de aplicaciones Web, según su entorno de funcionamiento, Intranet o Extranet. En el caso de la Intranet, la aplicación está alojada y disponible sólo dentro de la red interna de la institución que la desarrolla y en la Extranet está disponible desde cualquier punto que posea acceso a Internet; entonces si no se cuenta con una buena conexión y/o se prefiere tener las aplicaciones dentro de la organización se puede hacer sin desertar de las ventajas que brinda Internet.

### 1.4 Sistemas de gestión similares.

#### Internacional.

Sistema de Gestión de la Documentación Audiovisual en la Televisión Valenciana.

*“En la Unidad de Documentación de RTVV la documentación audiovisual esta soportada por una serie de bases de datos, todas gestionadas por el Sistema de Gestión Documental BRS/Search, que se han ido implementando desde su fundación. Algunas acogen los materiales de los cuales se hace un análisis documental simple (RPRO, parte de VIDE e IMAT), y las otras guardan el análisis completo de los contenidos de aquellos documentos que se han considerado dignos de una descripción más elaborada.”*

(7)

Luego de una exhaustiva búsqueda en internet sobre sistemas de gestión similares al que nos ocupa en el desarrollo de esta tesis en el entorno internacional el único sistema que se asemeja al nuestro es este, debido al uso dado a las nuevas tecnologías para el desarrollo y la información que en ella se maneja.

## **1.5 Arquitectura y patrones de diseño definidos para la implementación del Sistema de Gestión de Procesos de la Dirección de Televisión Universitaria.**

### **1.5.1 Lenguajes de Programación Web.**

Lenguaje de programación web es aquel que logra un estrecho vínculo entre los componentes esenciales de la Telaraña de Alcance Mundial (*World Wide Web*, WWW) que es la interrelación de varios Localizadores Uniforme de Recursos (*Uniform Resource Locators*, URL), HTTP y HTML, con el desarrollo numerosos lenguajes de programación Web se ha logrado llevar a internet a la puntera de la ciencia y la técnica en estos últimos años. Logrando así satisfacer las crecientes y exigentes necesidades de la gran autopista de la información.

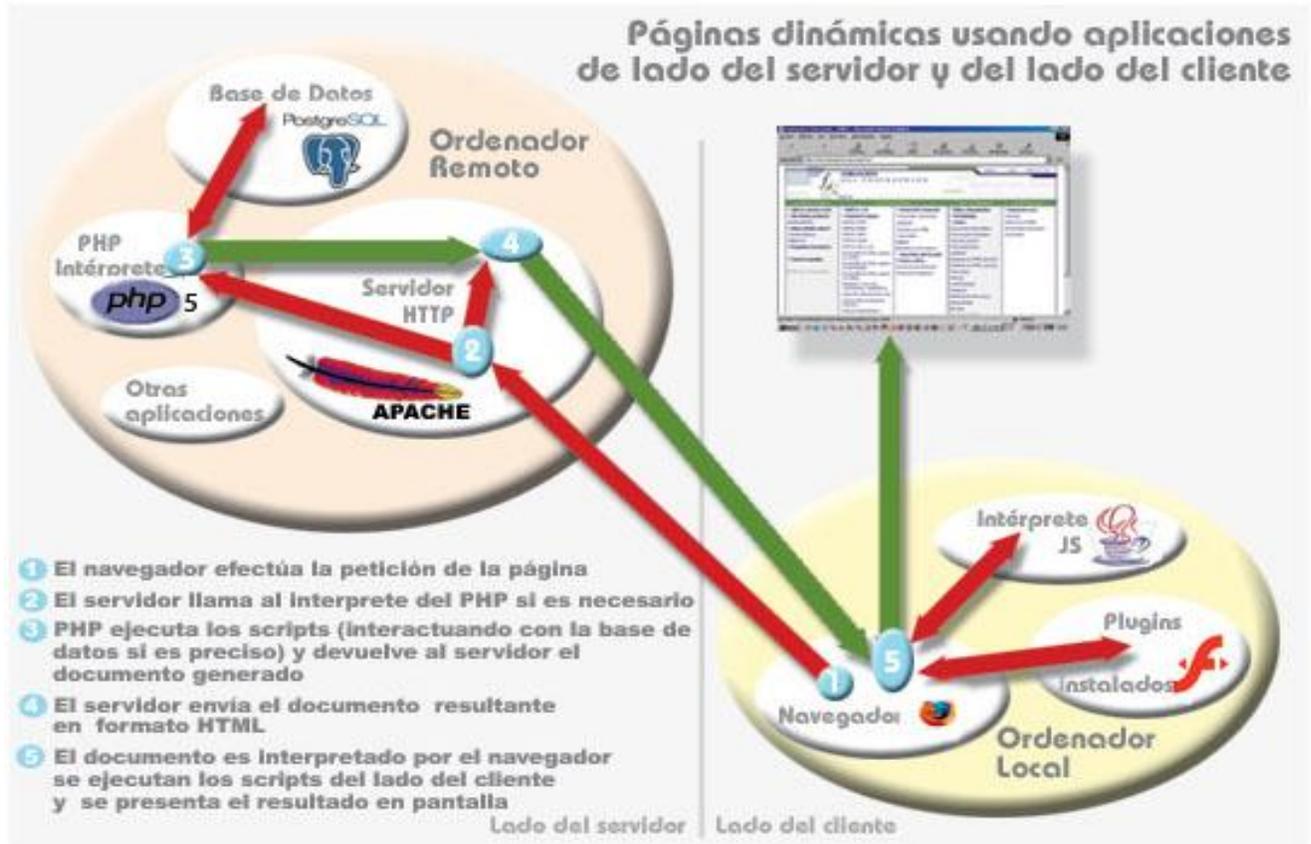


Figura 2: Lenguajes de programación web.

### 1.5.1.1 Lenguaje del lado del servidor.

Lenguaje del lado servidor es aquel que es totalmente independiente del navegador utilizado, debido a que no solicitará ningún componente o PLUG-IN (en inglés) especial para representar cabalmente cualquier página, el código de procesamiento por lotes o script (en inglés) puede ser ocultado al terminal cliente, que sólo notará el código HTML terminado. No obstante, la prueba y depuración de un script tendrá que hacerse en el servidor.

## PHP5.

PHP, hoy es el lenguaje más utilizado para la creación de aplicaciones web. Las versiones anteriores (PHP3 y PHP4) aportaron a este lenguaje una sólida base para el desarrollo de aplicaciones web, con la implementación de la **P**rogramación **O**rientada a **O**bjetos (POO). La creación de PHP5 tiene como objetivo principal mejorar los mecanismos de POO implementados en versiones anteriores.

*“PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. Soporta la mayoría de servidores web de hoy en día, incluyendo Apache y Microsoft Internet Information Server.... Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos.” (11)*

Dentro de las principales que nos brinda el PHP5 ventajas se pueden mencionar:

- *“Muy fácil de aprender.*
- *Se caracteriza por ser un lenguaje muy rápido.*
- *Soporta en cierta medida la orientación a objeto. Clases y herencia.*
- *Es un lenguaje multiplataforma: Linux, Windows, entre otros.*
- *Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.*
- *Capacidad de expandir su potencial utilizando módulos.*
- *Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.*
- *Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.*
- *Incluye gran cantidad de funciones.*

- *No requiere definición de tipos de variables ni manejo detallado del bajo nivel.” (12)*

Siendo así PHP un poderoso lenguaje e intérprete, ya sea como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, tiene la capacidad de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web, sea insegura por naturaleza; lo cual no le resta reconocimiento en su merecido lugar dentro de los lenguajes de programación. Además como una ventaja esencial presenta la característica de ser multiplataforma, logrando con ello una aceptación masiva por parte de todo tipo de usuarios sin importar el sistema operativo o sistema gestor de bases de datos utilizado o preferido.

### **1.5.1.2 Lenguaje del lado del cliente.**

#### **HTML.**

A partir del nacimiento de internet y a través de sus años de existencia se generó un medio totalmente nuevo de intercambio de información; con ello surgieron los navegadores web, para los cuales era necesario establecer un lenguaje que fuese entendible para todos y en este preciso momento fue que surgió el lenguaje de interpretado HTML. La función esencial de este lenguaje es indicar al navegador donde colocar cada texto, imagen o video y la forma que tendrán estos al ser colocados en la página.

*“HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos -ya sea en su computadora o en otras- así como otros recursos de Internet, como FTP y Gopher.”(1)*

## JavaScript.

Java Script es un lenguaje de programación que se emplea para implementar pequeños programas que se encargan de realizar funciones dentro del ámbito de una página web. Se conoce que es un lenguaje de programación de la parte del cliente, porque es el navegador el que soporta la carga de procesamiento.

Su utilización está basada principalmente en la creación de efectos visuales en las páginas y la definición de interactividades con el usuario para así dividir la carga de procesamiento entre el cliente y el servidor, logrando que el cliente sea capaz de realizar varias funcionalidades del sistema que no precisen datos del servidor y con ello reducir el tráfico de red.

*“El código JavaScript puede ser integrado dentro de nuestras páginas Web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).” (12)*

### 1.5.2 Servidores Web.

Los servidores web son programas que implementan el protocolo HTTP, perteneciendo así a la capa de aplicación del Modelo de Referencia de Interconexión de Sistemas Abiertos (*Open System Interconnection*, OSI) y está planteado para la transferencia de lo que se conoce como hipertextos, páginas web o páginas HTML. Estos programas se ejecutan continuamente en el ordenador manteniéndose así escuchando las peticiones de los clientes a través de los puertos habilitados en todo momento.

### Servidor HTTP Apache 2.0.

Con la utilización que se le da hoy día a internet y el avance alcanzado por los sistemas que esta soporta, ha logrado desarrollar una cultura internacional de utilización de esta, para búsquedas de información por lo que se hace necesario que los proveedores que brindan esta información se desarrollen aún más. Ejemplo de ello son los servidores que almacenan el gran cumulo de información que en la red de redes existe, con este fin es la versión 2.0 de Apache.

*“Hoy en día es considerada la plataforma Web más utilizada del mundo, pues aumentan día a día el número de usuarios que aceptan este código fuente abierto en su infraestructura.” (9).*

Las principales características son:

- ✓ Multi-plataforma.
- ✓ Código abierto.
- ✓ Gratuito.
- ✓ Popular (fácil de conseguir ayuda y soporte).
- ✓ Organización por módulos.
- ✓ Extensible.

Estas características hacen de este servidor una poderosa herramienta debido a su fácil manejo y configuración, logrando grandes prestaciones y aceptación por parte de todos los usuarios.

### 1.5.3 Lenguaje Unificado de Modelado.

El Lenguaje Unificado del Modelado (**Unified Modeling Language**, UML) es un lenguaje estándar en la modelación, de fácil entendimiento para los usuarios que lo utilizan con una cuantiosa notación gráfica, permite además la creación de un conjunto de diagramas y elementos de fácil comprensión, que permiten la visualización, especificación, construcción y documentación de los artefactos que mostraran como funcionará el sistema de desarrollo de software. Es compatible con varias de las metodologías de desarrollo de software existentes en el mundo como el Proceso Unificado de Rational (**Rational Unified Process**, RUP), pero por sí mismo no es capaz de representar ese proceso o metodología.

*“UML son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelación), notación con que se construyen sistemas por medio de conceptos orientados a objetos.” (10)*

Este lenguaje ayuda a simplificar el proceso de diseño de software, logrando hacer ver al desarrollador diferentes vistas de un modelo. Dentro de los grandes méritos que se le pueden adjudicar a este lenguaje es la manera en que ayuda en el comienzo del proceso de desarrollo con los nombrados casos de uso. Este lenguaje posee una alta exactitud, lo cual le permite crecerse como un estándar industrial muy utilizado; lo que ha conllevado a que la competencia existente se base en las herramientas que mejor soporten UML y no en la creación de herramientas para los distintos lenguajes de modelado; lo cual es mucho más conveniente para los usuarios.

#### 1.5.4 Patrón Arquitectónico.

##### Modelo Vista Controlador.

El **Modelo Vista Controlador (MVC)**, no es más que la consumación de la arquitectura en tres capas, extendida. Permite la separación del código de cada una de las capas, ayudando así a los desarrolladores y diseñadores a vincular y mantener el código fuente más fácilmente, permite además que el software sea de fácil modificación. Hoy por hoy la arquitectura MVC es la opción más utilizada en la creación de sitios web.

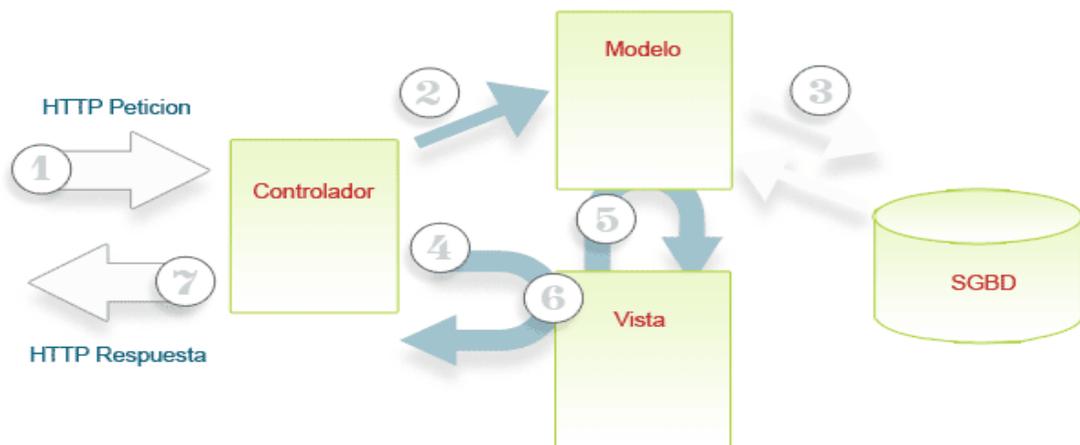


Figura: 3 Arquitectura Modelo Vista Controlador.

- 1) El Controlador recibe una petición. Esto puede ocurrir de forma directa poniendo el usuario en el navegador la url del Controlador `http: //sitioweb.algo/controlador.php`, o mediante un evento a través de una Vista ya renderizada (envío de un formulario por ejemplo), o por una petición ajax.
- 2) El Controlador analiza la petición y usa un Modelo para obtener datos de algún tipo.
- 3) El Modelo puede leer datos y/o los escribirlos en una base de datos. Normalmente sucede esto, aunque no siempre.
- 4) El Controlador usa una Vista para poder presentar los resultados al usuario.
- 5) La Vista usa el Modelo, de tal forma, que algún tipo de respuesta obtenido por la acción previa será presentado en la Vista.
- 6) La Vista está renderizada con los datos del Modelo y el Controlador envía esta respuesta al usuario.
- 7) El usuario obtiene la respuesta

Ventajas:

- Se obtienen múltiples vistas del modelo.
- Todas las vistas están sincronizadas.
- No acoplamiento y facilidad de evolución, para cambiar las vistas y los controladores.
- La aplicación puede soportar un tipo de interfaz para cada usuario (rol).

En un proyecto implementado con la ayuda de una arquitectura distinta al MVC si se desea cambiar algo, esto podría tornarse algo engorroso, en cambio la utilización del MVC le proveerá de una manera fácil de cambiar las capas de presentación sin que se afecte la de acceso de datos y viceversa.

### 1.5.5 Framework.

Un Framework es un sistema informático diseñado para facilitar el desarrollo de software, procurándole más tiempo al programador para centrarse en los requerimientos del software que en detalles menos importantes en el desarrollo de este.

#### Symfony.

Symfony es un framework diseñado para la optimización y reutilización de funciones en el desarrollo de aplicaciones web. En principio este se basa en la separación de la aplicación en 3 capas fundamentales; lógica de negocio, lógica del servidor y la aplicación web, para lograr reducir el tiempo de la implementación de una aplicación web compleja, automatizando las tareas más frecuentes, permitiendo al programador centrarse en aspectos concretos de la aplicación. En fin, el logro benéfico de este framework es la reutilización de código.

*“Symfony está desarrollado completamente con PHP 5. 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, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.*

*Características de Symfony.*

- *Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix 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.*

*Symfony automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo:*

- *La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.*
- *La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.*
- *Los formularios incluyen validación automatizada y relleno automático de datos ("repopulation"), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.*
- *Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.*
- *La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.*
- *La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.*
- *El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.*

- *El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores.*
- *Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.*
- *Los plugins, las factorías (patrón de diseño "Factory") y los "mixin" permiten realizar extensiones a medida de Symfony.*
- *Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código. “ (3)*

Debido a las características de este framework se convierte en una herramienta imprescindible a utilizar en el desarrollo de nuestro sistema.

### 1.5.6 Sistema Operativo

El sistema operativo es aquel programa (o software) más importante de un ordenador. Para que desempeñen sus tareas los otros programas, cada ordenador debe contar con un sistema operativo. Los sistemas operativos ejecutan las tareas básicas del ordenador, tales como reconocimiento de la conexión del teclado, mouse, enviar información a pantalla, localizar y mostrar archivos y directorios en el sistema de almacenamiento, así como controlar otros dispositivos periféricos tales como impresoras, escáner, etc.

En grandes sistemas, el sistema operativo tiene incluso mucha más carga y poder, se cerciora de que los programas y usuarios que están funcionando concurrentemente no interfieran entre ellos. El sistema operativo también se hace responsable de la seguridad, asegurándose de que los usuarios o programas no autorizados no posean acceso al sistema.

#### Ubuntu.

El sistema operativo Ubuntu es una distribución GNU/Linux que ofrece un enfoque a ordenadores personales, aunque, proporciona además soporte para servidores, llegando a ser así una de las más

importantes distribuciones de GNU/Linux a nivel mundial. Está basada en el sistema operativo Debian GNU/Linux y se centra en la facilidad e independencia de uso, la fluida instalación y el lanzamiento de versiones regularmente cada 6 meses. Su principal benefactor es Canonical Ltd., una empresa privada instituida y sufragada por el empresario Mark Shuttleworth.

El nombre de esta distribución está dado por el concepto zulú y xhosa de Ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano guiado por el obispo Desmond Tutu, quien obtuvo el Premio Nobel de la Paz en 1984 por su constante lucha en contra del Apartheid en Sudáfrica. El sudafricano Mark Shuttleworth, se encontraba muy familiarizado con las ideas del obispo, luego de ver similitudes entre las bases de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de la distribución –“Linux para seres humanos” (*Linux for Human Beings*) – sintetiza una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

### 1.5.7 Entorno Integrado de Desarrollo.

#### Eclipse PDT.

El Entorno de Desarrollo Integrado (*Integrated Development Environment*, IDE) Eclipse con el uso de Herramientas de Desarrollo de PHP (*PHP Development Tools*, PDT) brinda un framework y un agregado de herramientas para el desarrollo en PHP sobre la plataforma de Eclipse, tratando de crear el mejor IDE operable para este lenguaje de programación web. Además encapsula los componentes necesarios para el desarrollo de software en PHP y facilita su extensibilidad. Algunas de sus principales características son:

- Adherencia a los estándares de Eclipse.
- Extensibilidad.
- Soporte continuo al desarrollo de PHP.

- Soporte para Subversion.
- Refactorización.

### **1.5.8 Metodología de desarrollo de software.**

Metodología de desarrollo de software es un marco de trabajo utilizado para ordenar, planificar y controlar el proceso de desarrollo en sistemas de información. Para el desarrollo de cualquier herramienta informática se hace imprescindible el uso de alguna de las metodologías de desarrollo de software.

#### **Proceso Unificado de Desarrollo.**

El Proceso Unificado de Desarrollo es una metodología de desarrollo de software orientada a objetos. Esta proporciona a los desarrolladores guías, plantillas y ejemplos en todos los aspectos que esta abarca en todas las etapas del desarrollo de software. Con su utilización se pretenden poner en práctica las mejores herramientas en lo que a Ingeniería de Software se refiere, ordenando el proceso del desarrollo (fases, técnicas y prácticas) con otros elementos del desarrollo (documentos, modelos, manuales y código) en una sola plataforma.

*“El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.” (5)*

RUP instituye cuatro fases de desarrollo como se muestra en la figura 4, cada una de las cuáles está estructurada por un cierto número de iteraciones, que deben lograr al final de cada una de éstas, satisfacer los objetivos trazados antes de pasar a la fase siguiente. En la fase de Inicio se precisa el alcance del proyecto, en la fase de Elaboración se define la arquitectura, en la de Construcción se implementan las clases del diseño y por último, en la fase de Transición, el sistema es entregado a los usuarios finales.

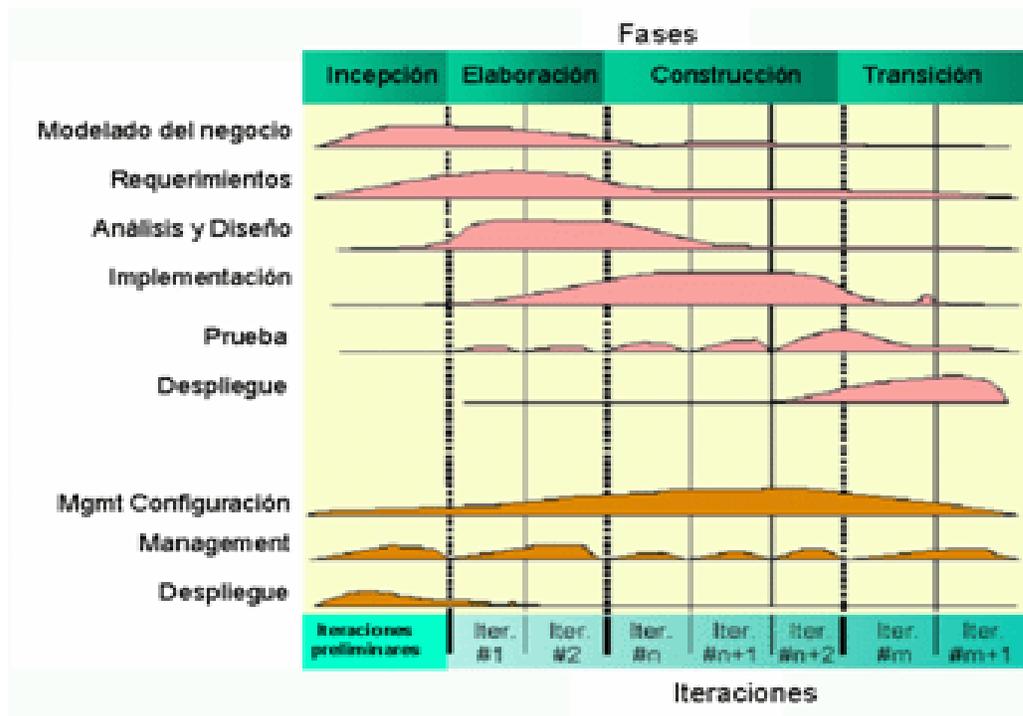


Figura 4: Fases y flujos de trabajo en RUP.

Principales características:

- Forma disciplinada de asignar tareas y responsabilidades.
- Desarrollo iterativo e incremental.
- Es centrado en la arquitectura, guiado por los casos de uso e incluye artefactos y roles.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

## **1.6 Conclusiones.**

Con el uso en la práctica de todas las herramientas antes mencionadas se logrará como conclusión, un buen producto de software debido a la buena estructuración que brindan las metodologías propuestas, logrando con ello que nuestro producto este desarrollado siguiendo las mejores prácticas de diseño y programación.

## CAPÍTULO II. IMPLEMENTACIÓN DEL SISTEMA.

---

### 2.1 Introducción.

En el presente capítulo, se enumeran los requisitos funcionales con sus descripciones, que servirán de guía para la realización de la aplicación. De esta manera se exponen los artefactos generados por el Rol de Programador según RUP, siendo estos los diagramas de componentes, código fuente aunque sólo se mostrará parte de este, por lo extenso que este sea, haciendo énfasis en las funciones más importantes en el desarrollo de la aplicación. El código se desarrollo haciendo uso del framework Symfony, lo que trajo consigo que la herramienta generara 32 clases peer, 32 no peer y 32 clases base peer, e igual cantidad de clases base no peer.

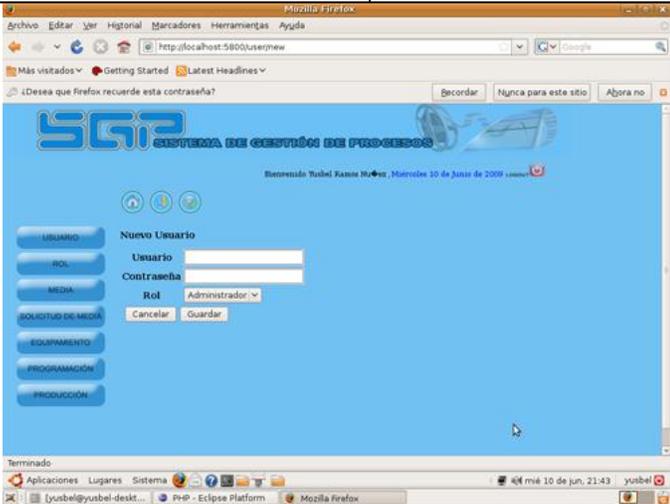
### 2.2 Descripción de los requisitos funcionales.

Para hacer más fácil el entendimiento de los requisitos funcionales que debe cumplir del sistema se utilizara la notación R\* para definir cuál es el requisito del que se habla así como los derivados de este con R\*.\*.

- ❖ R1 Gestionar usuario.
  - R 1.1 Adicionar usuario.
  - R 1.2 Modificar usuario.
  - R 1.3 Eliminar usuario.
  - R 1.4 Buscar usuario.

El requerimiento permitirá que el administrador (persona encargada de administrar y gestionar los principales permisos de la aplicación) del sistema adicione, modifique o elimine cualquier usuario del sistema, así como también podrá buscar un determinado usuario dentro del sistema.

Tabla 1: Descripción del CU Gestionar Usuario.

<b>Caso de Uso:</b>	Gestionar Usuario.	
<b>Actores:</b>	Administrador.	
<b>Resumen:</b>	El administrador del sistema adiciona, modifica, elimina o busca un usuario.	
<b>Precondiciones:</b>	Se debe haber autenticado como administrador.	
<b>Referencias</b>	R1, R1.1, R1.2, R1.3, R1.4.	
<b>Prioridad</b>	Secundaria.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Adicionar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción adicionar.		
2- El administrador entra los datos del nuevo usuario y oprime el botón “Guardar”.	3- El sistema verifica los datos.	
	4- El sistema guarda los datos.	
	5- El sistema muestra un mensaje de confirmación.	
		
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4- El sistema muestra un mensaje de error que	

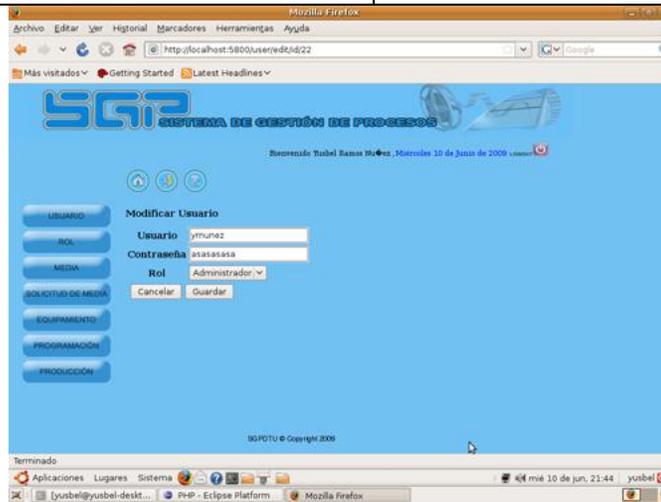
	alerta que ese usuario ya existe.
--	-----------------------------------

**Flujos Alternos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Modificar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción modificar.	2- El sistema muestra los datos del usuario.
3- El administrador modifica los datos y oprime el botón “Guardar”.	4- El sistema verifica los datos.
	5- El sistema actualiza los datos.
	6- El sistema muestra un mensaje de confirmación.



**Flujos Alternos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	5- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Eliminar”**

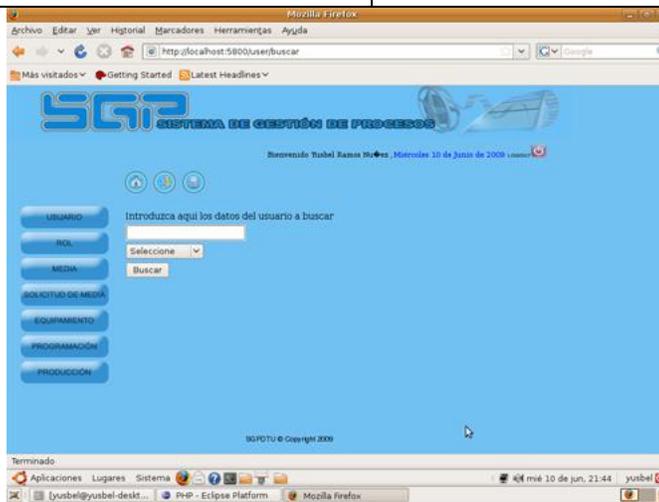
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción eliminar.	2- El sistema muestra un mensaje de

	confirmación.
3- El administrador oprime el botón "Aceptar".	4- El sistema elimina los datos.



**Sección "Buscar"**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción buscar.	
2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos del usuario buscado.



**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	3- El sistema muestra un mensaje de error que indica que no se encontró el usuario.

**Poscondiciones**

Se actualiza la tabla de usuarios o se encuentra o no el usuario buscado.

- ❖ R2 Gestionar rol.
  - R 2.1 Asignar permiso a un rol.
  - R 2.2 Adicionar rol.
  - R 2.3 Modificar rol.
  - R 2.4 Eliminar rol.
  - R 2.5 Buscar rol.

El requerimiento permitirá que el administrador (persona encargada de administrar y gestionar los principales permisos de la aplicación) del sistema asigne, modifique o elimine cualquier rol referente a un usuario del sistema, así como también podrá buscar un determinado rol dentro del sistema.

Tabla 2: Descripción del CU Gestionar Rol.

<b>Caso de Uso:</b>	Gestionar Rol.	
<b>Actores:</b>	Administrador.	
<b>Resumen:</b>	El administrador del sistema adiciona, modifica, elimina o busca un rol.	
<b>Precondiciones:</b>	Se debe haber autenticado como administrador.	
<b>Referencias</b>	R2, R2.1, R2.2, R2.3, R2.4.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Adicionar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción adicionar.		
2- El administrador entra los datos del nuevo rol y oprime el botón “Guardar”.	3-El sistema verifica los datos.	
	4-El sistema guarda los datos.	
	5-El sistema muestra un mensaje de confirmación.	



**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que ese rol ya existe.

**Sección “Modificar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona el rol que se va a modificar.	
2- El administrador modifica los datos y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.
	5- El sistema muestra un mensaje de confirmación.

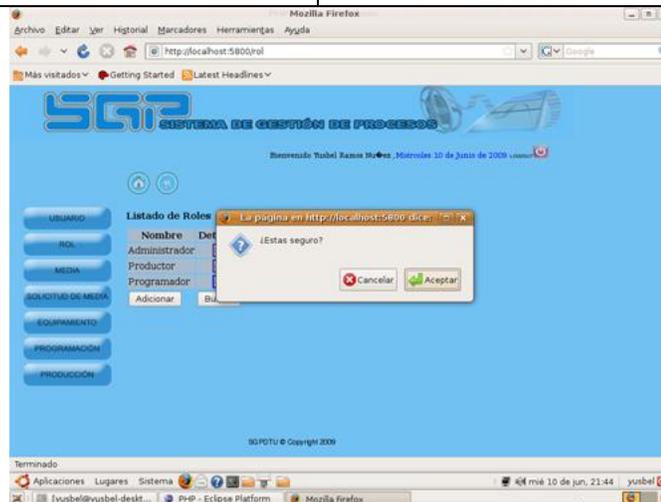


**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Eliminar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción eliminar.	2- El sistema muestra un mensaje de confirmación.
	3- El sistema Elimina los datos.



**Sección “Buscar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción buscar.	

2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos del rol buscado.
	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3- El sistema muestra un mensaje de error que indica que no se encontró el rol.
<b>Poscondiciones</b>	Se actualiza la tabla roles o se encuentra o no el rol buscada.

❖ R3 Autenticar usuario.

El requerimiento permitirá a todos los usuarios que deseen interactuar con el sistema su previa autenticación. Este requisito es indispensable, ya que un usuario no podrá acceder al sistema sin una previa autenticación.

Tabla 3: Descripción del CU Autenticar Usuario.

<b>Caso de Uso:</b>	Autenticar Usuario.
<b>Actores:</b>	Usuario.
<b>Resumen:</b>	El usuario del sistema se autentica para obtener permisos sobre determinadas opciones del sistema.
<b>Precondiciones:</b>	El usuario debe estar registrado.
<b>Referencias</b>	R3.
<b>Prioridad</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Autenticar”</b>	

Acción del Actor		Respuesta del Negocio
1- El usuario entra a la aplicación.		
2- El usuario entra su usuario y su contraseña		3- El sistema valida los datos.
		4- El sistema da acceso al usuario.
		
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		4- El sistema muestra un mensaje de error alertando que el usuario o la contraseña no son correctos.
Poscondiciones	El usuario queda o no autenticado y obtiene permisos o no para entrar al sistema.	

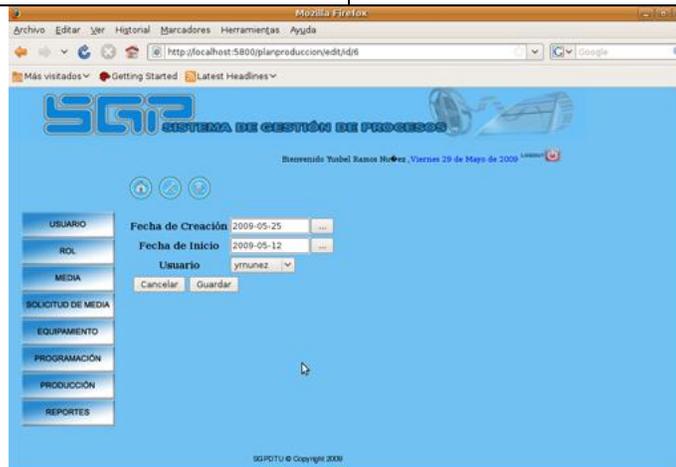
- ❖ R4 Gestionar plan de producción.
  - R 4.1 Adicionar plan de producción.
  - R 4.2 Modificar plan de producción.
  - R 4.3 Eliminar plan de producción.
  - R 4.4 Buscar plan de producción.

El requerimiento permitirá que el especialista de la DTU (persona encargada de gestionar todo lo referente al plan de producción) que debe interactuar con el sistema, adicione, modifique o elimine cualquier valor referente al plan de producción insertado en el sistema, así como también permitirá buscar un plan de producción.

Tabla 4: Descripción del CU Gestionar Plan de Producción.

<b>Caso de Uso:</b>	Gestionar Plan de Producción.	
<b>Actores:</b>	Administrador o Productor.	
<b>Resumen:</b>	El administrador o productor del sistema adiciona, modifica, elimina o busca un plan de producción.	
<b>Precondiciones:</b>	Se debe haber autenticado como administrador o productor.	
<b>Referencias</b>	R4, R4.1, R4.2, R4.3, R4.4.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Adicionar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción adicionar.		
2- El administrador entra los datos del nuevo plan de producción y oprime el botón “Guardar”.	3- El sistema verifica los datos.	
	4- El sistema guarda los datos.	
	5- El sistema muestra un mensaje de confirmación.	
		
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.	
<b>Sección “Modificar”</b>		

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona el plan de producción que se va a modificar.	
2- El administrador modifica los datos y oprime el botón "Guardar".	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.
	5- El sistema muestra un mensaje de confirmación.

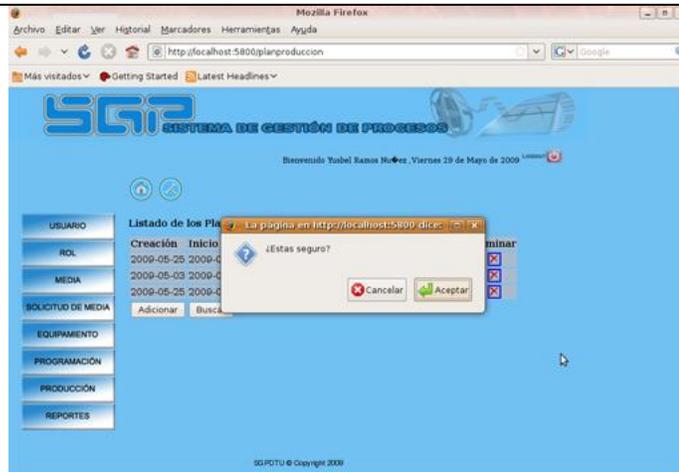


**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

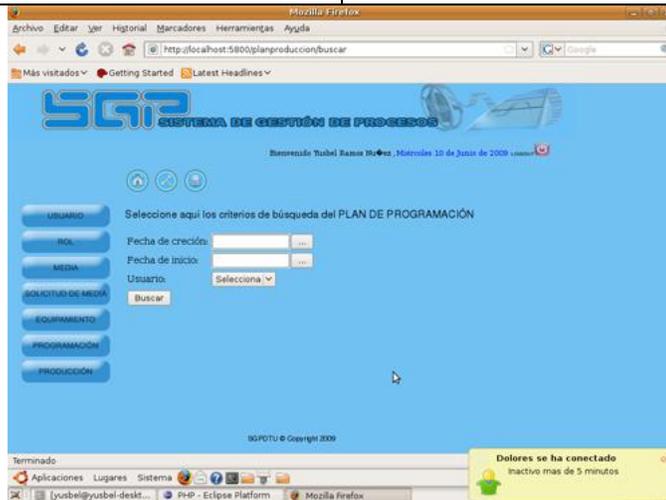
**Sección "Eliminar"**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona el plan de producción a eliminar.	2- El sistema muestra un mensaje de confirmación.
	3- El sistema elimina los datos.



**Sección “Buscar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción buscar.	
2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos del plan de producción buscado.



**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	3- El sistema muestra un mensaje de error que indica que no se encontró el plan de producción.
<b>Poscondiciones</b>	Se actualiza la tabla plan de producción o se encuentra o no el plan de producción buscado.

- ❖ R5 Gestionar media.
  - R 5.1 Adicionar media.
  - R 5.2 Modificar media.
  - R 5.3 Eliminar media.
  - R 5.4 Buscar media.

El requerimiento permitirá que el especialista de la DTU (persona encargada de gestionar todo lo referente a las medias) que debe interactuar con el sistema, adicione, modifique o elimine cualquier media insertada previamente en el sistema, así como también permitirá realizar una búsqueda de una determinada media.

Tabla 5: Descripción del CU Gestionar Media.

<b>Caso de Uso:</b>	Gestionar Medias.	
<b>Actores:</b>	Administrador.	
<b>Resumen:</b>	El administrador del sistema adiciona, modifica, elimina o busca una media.	
<b>Precondiciones:</b>	Se debe haber autenticado como administrador.	
<b>Referencias</b>	R5, R5.1, R5.2, R5.3, R5.4.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Adicionar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción adicionar.		
2- El administrador entra los datos de la nueva media y oprime el botón “Guardar”.	3-El sistema verifica los datos.	
	4-El sistema guarda los datos.	
	5-El sistema muestra un mensaje de confirmación.	

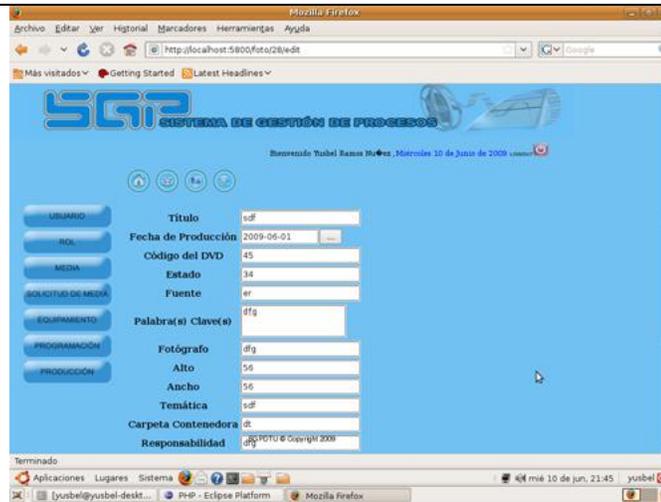


**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Modificar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la media que se va a modificar.	
2- El administrador modifica los datos y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.
	5- El sistema muestra un mensaje de confirmación.

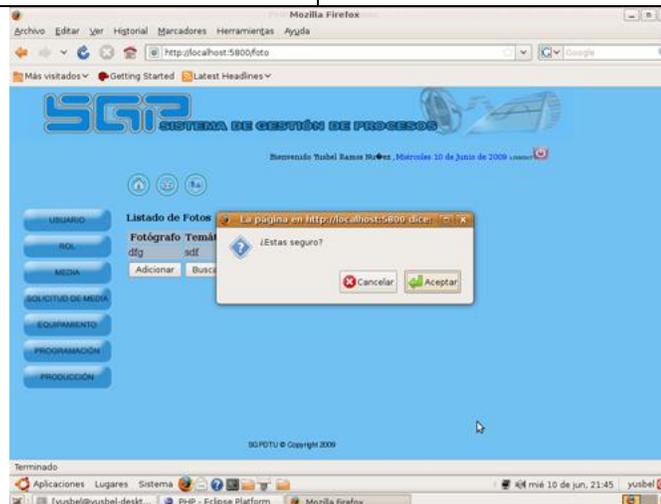


**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Eliminar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la media a eliminar.	2- El sistema muestra un mensaje de confirmación.



**Sección “Buscar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción buscar.	

2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos de la media buscada.
	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3- El sistema muestra un mensaje de error que indica que no se encontró la media.
<b>Poscondiciones</b>	Se actualiza la tabla media o se encuentra o no la media buscada.

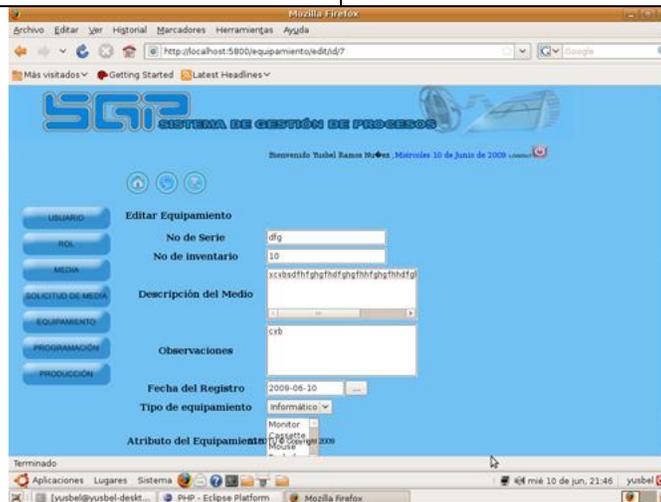
- ❖ R6 Gestionar equipamiento.
  - R 6.1 Adicionar equipamiento.
  - R 6.2 Modificar equipamiento.
  - R 6.3 Eliminar equipamiento.
  - R 6.4 Asignar equipamiento.
  - R 6.5 Modificar asignación de equipamiento.
  - R 6.6 Eliminar asignación de equipamiento.

El requerimiento permitirá que el especialista de la DTU (persona encargada de asignar equipos informáticos) que debe interactuar con el sistema, modifique o elimine cualquier asignación de equipamiento informático a un determinado usuario.

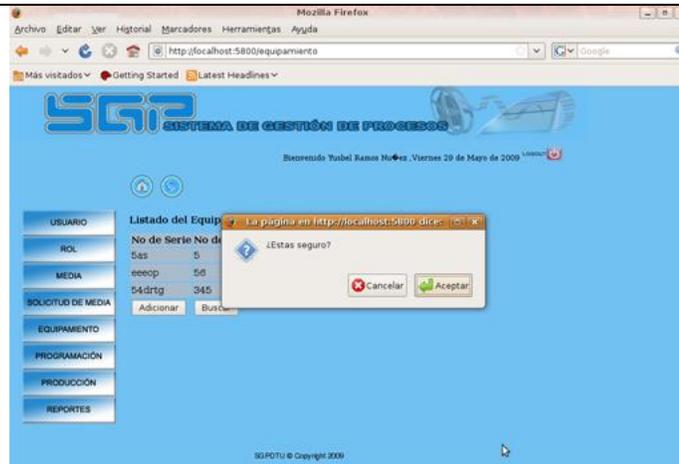
Tabla 6: Descripción del CU Gestionar Equipamiento.

<b>Caso de Uso:</b>	Gestionar Equipamiento.	
<b>Actores:</b>	Administrador.	
<b>Resumen:</b>	El administrador del sistema adiciona, modifica, elimina un equipamiento o adiciona, modifica o elimina una asignación de un equipamiento a un usuario.	
<b>Precondiciones:</b>	Se debe haber autenticado como administrador.	
<b>Referencias</b>	R6, R6.1, R6.2, R6.3, R6.4, R6.5, R6.6.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Adicionar”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción adicionar.		
2. El administrador entra los datos del nuevo equipamiento y oprime el botón “Guardar”.	3. El sistema verifica los datos.	
	4. El sistema guarda los datos.	
	5. El sistema muestra un mensaje de confirmación.	
		
<b>Flujos alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	4. El sistema muestra un mensaje de error que	

	alerta que existen campos vacíos o con datos incorrectos.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona el equipamiento que se va a modificar.	
2- El administrador modifica los datos y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.
	5- El sistema muestra un mensaje de confirmación.



<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.
<b>Sección “Eliminar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona el equipamiento a eliminar.	2- El administrador selecciona la media a eliminar.



**Sección “Buscar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción buscar.	
2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos de la media buscada.



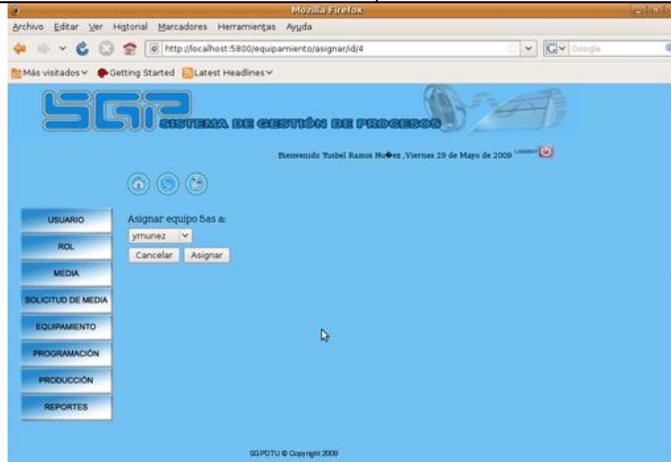
**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	3- El sistema muestra un mensaje de error que indica que no se encontró el equipamiento.

**Flujo Normal de Eventos**

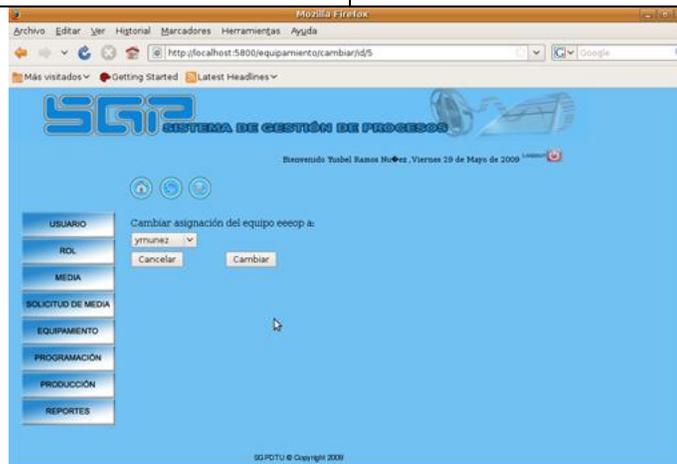
**Sección “Asignar”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la opción asignar equipamiento.	2- El sistema muestra los usuarios.
3- El administrador selecciona el usuario al que se le asignará el equipamiento.	4- El sistema muestra un mensaje de confirmación.

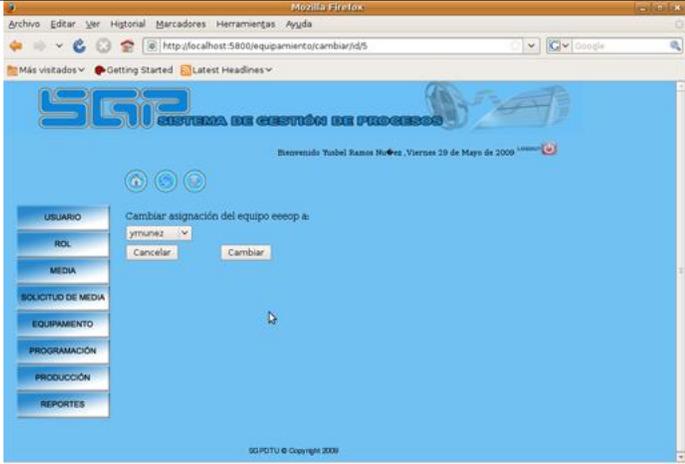


**Sección “Modificar Asignación”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona la asignación que va a modificar.	
2- El administrador modifica los datos.	3- El sistema actualiza los datos.
	4- El sistema muestra un mensaje de confirmación.



**Sección “Eliminar Asignación”**

Acción del Actor	Respuesta del Sistema
1- El administrador selecciona el equipamiento a eliminar.	2- El sistema muestra un mensaje de confirmación.
	3- El sistema elimina los datos.
	
Poscondiciones	Se adiciona o actualiza un equipamiento así como sus asignaciones.

- ❖ R7 Gestionar plan de programación.
  - R 7.1 Adicionar plan de programación.
  - R 7.2 Eliminar plan de programación.
  - R 7.3 Visualizar plan de programación.

El requerimiento permitirá que el especialista de la DTU (persona encargada de gestionar todo lo referente al plan de programación) que debe interactuar con el sistema, adicione o elimine cualquier valor referente al plan de producción insertado en el sistema, así como también permitirá visualizar un determinado plan de programación.

Tabla 7: Descripción del CU Gestionar Plan de Programación.

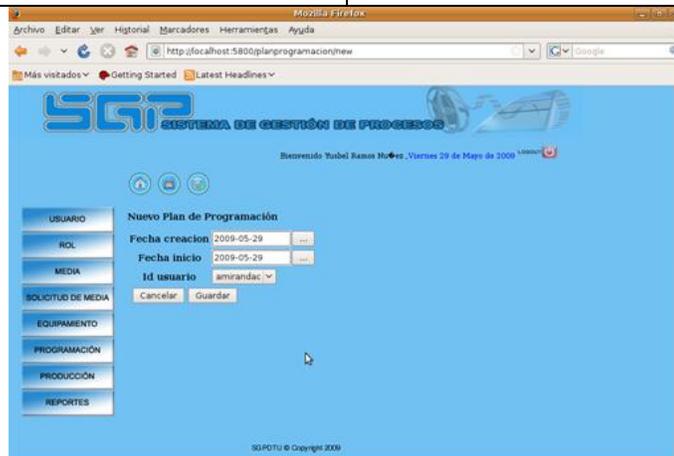
<b>Caso de Uso:</b>	Gestionar Plan de Programación.
<b>Actores:</b>	Administrador o Programador.
<b>Resumen:</b>	El administrador del sistema adiciona, modifica, elimina o busca un plan de programación.
<b>Precondiciones:</b>	Se debe haber autenticado como administrador o programador.

<b>Referencias</b>	R7, R7.1, R7.2, R7.3, R7.4.
<b>Prioridad</b>	Crítico.

**Flujo Normal de Eventos**

**Sección “Adicionar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción adicionar.	
2- El administrador entra los datos del nuevo plan de programación y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema guarda los datos.
	5- El sistema muestra un mensaje de confirmación.



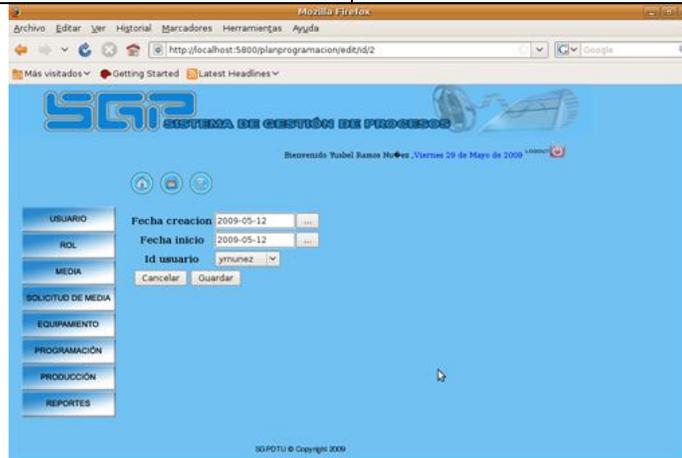
**Flujos Alternos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Modificar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona el plan de programación que se va a modificar.	
2- El administrador modifica los datos y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.

5- El sistema muestra un mensaje de confirmación.



### Flujos Alternos

#### Acción del Actor

#### Respuesta del Sistema

4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

### Sección "Eliminar"

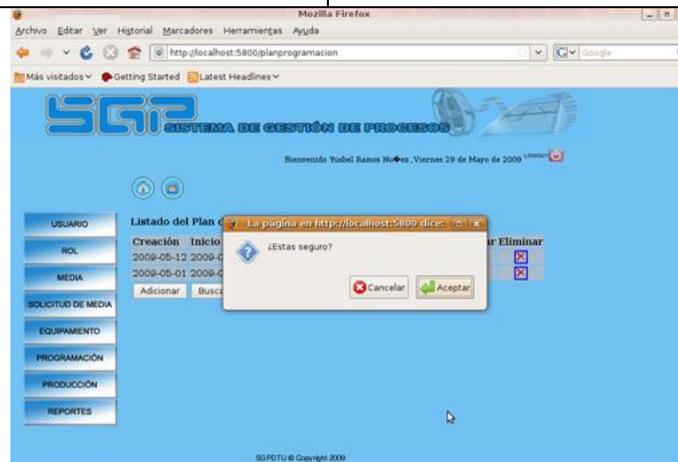
#### Acción del Actor

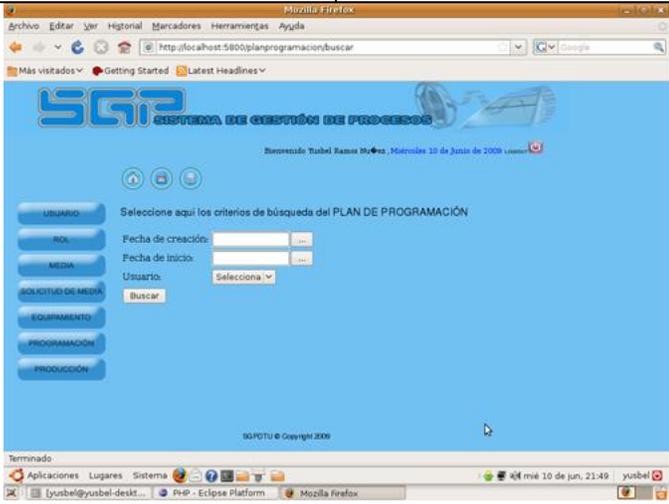
#### Respuesta del Sistema

1- El administrador selecciona el plan de programación a eliminar.

2- El sistema muestra un mensaje de confirmación.

3- El sistema elimina los datos del plan de programación seleccionado.



<b>Sección “Buscar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción buscar.	
2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos del plan de programación buscado.
	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3- El sistema muestra un mensaje de error que indica que no se encontró el plan de programación.
<b>Poscondiciones</b>	Se actualiza la tabla plan de programación o se encuentra o no el plan de programación buscado.

- ❖ R8 Gestionar solicitud de media.
  - R8.1 Registrar solicitud.
  - R8.2 Buscar solicitud.
  - R8.3 Modificar solicitud.

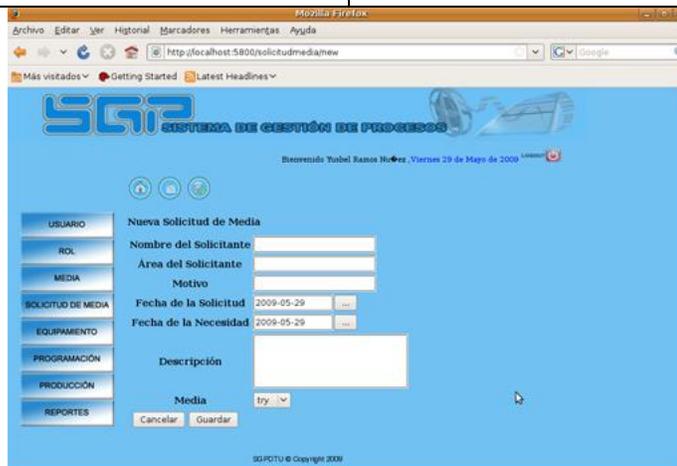
El requerimiento permitirá que el especialista de la DTU (persona encargada de gestionar todo lo referente a las medias) que debe interactuar con el sistema, registrar, o modificar una solicitud de media, así como también permitirá buscar cualquier solicitud de media existente.

<b>Caso de Uso:</b>	Gestionar Solicitud de Media.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El administrador del sistema adiciona, modifica o busca una solicitud de media.
<b>Precondiciones:</b>	Se debe haber autenticado como administrador.
<b>Referencias</b>	R8, R8.1, R8.2, R8.3.
<b>Prioridad</b>	Crítico.

**Flujo Normal de Eventos**

**Sección “Adicionar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción adicionar.	
2- El administrador entra los datos de la nueva solicitud de media y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema guarda los datos.
	5- El sistema muestra un mensaje de confirmación.



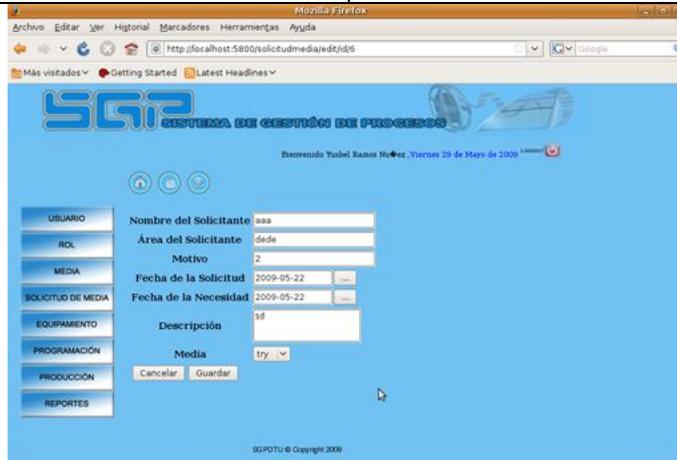
**Flujos Alternos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Modificar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la solicitud de	

media que va a modificar.	
2- El administrador modifica los datos y oprime el botón “Guardar”.	3- El sistema verifica los datos.
	4- El sistema actualiza los datos.
	5- El sistema muestra un mensaje de confirmación.



**Flujos Alternos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un mensaje de error que alerta que existen campos vacíos o con datos incorrectos.

**Sección “Buscar”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El administrador selecciona la opción buscar.	
2- El administrador inserta el criterio de búsqueda.	3- El sistema muestra los datos del plan de programación buscado.



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3- El sistema muestra un mensaje de error que indica que no se encontró el plan de programación.
<b>Poscondiciones</b>	Se actualiza la tabla solicitud de media o se encuentra o no la solicitud de media buscada.

### 2.3 Estándares de codificación.

El estándar de codificación es el documento que detalla los convenios que los desarrolladores siguen para escribir el código fuente de un sistema. Logrando que todos los programadores escriban el código con un mismo estilo, permitiendo así que el trabajo en equipo sea más efectivo y que se obtenga así homogeneidad en el código fuente, todo esto permite que un desarrollador que vea una parte del código logre saber, qué es lo que hace la aplicación.

*“Un estándar de codificación comprende los aspectos de la generación de código y repercute directamente en la legibilidad y la extensibilidad de cualquier proyecto de Software, haciendo que nuevos desarrolladores se acoplen rápidamente al proceso de desarrollo.” (8)*

La unión de las mejores prácticas de la programación y las sólidas técnicas de codificación con el objetivo de crear un código robusto, es un punto clave para lograr una buena calidad del software. La aplicación

continua de un estándar de codificación contribuye a la creación de un sistema de software fácil de entender y de mantener.

Reglas para codificar o definir un estándar de codificación.

1. Usar igual sangría en todo el código del sistema.
2. Evitar la tenencia de más de una clase por archivo.
3. No emplear variables o métodos multipropósito.
4. El tiempo de vida de las variables debe ser el más breve posible.
5. Redactar todos los nombres de las estructuras en un mismo idioma.
6. Manejar siempre las situaciones de error, preferentemente mediante rutinas de captura y tratamiento de excepciones.
7. Lograr el uso de las sentencias SWITCH-CASE en lugar de las sentencias IF-ELSE repetitivas.
8. Dotar a las variables y métodos de nombres objetivos, impidiendo así la ocurrencia de ambigüedad en el código y mejorando con ello la comprensión del mismo.
9. Comentar todos los métodos o funciones, así como las partes del código que pudieran ser difíciles de comprender.

Estándar de codificación para el Sistema de Gestión de Procesos de la Dirección de Televisión Universitaria.

1. Todas las etiquetas PHP deben estar completas (`<?php ?>`), no reducidas (`<? ?>`).
2. Los nombres de las variables y funciones tienen que ser siempre fáciles de leer, procurando que sean palabras en minúsculas con significado claro. Si realmente se necesita más de una palabra, éstas deben ponerse juntas, poniendo la inicial de cada palabra en mayúscula siempre que no sea la primera, procurando que su longitud sea la menor posible. Utilizar nombres en plural para arreglos o matrices de objetos.

Ejemplos:

`$ci`

`$tipoMedia`

`$user`

3. Los bloques de código deben estar siempre confinados por llaves (incluso si tienen una sola línea).

4. Las cadenas deben definirse utilizando comillas simples siempre que sea posible, para obtener un mejor rendimiento.
5. Todas las clases y funciones deben estar comentariadas, estos deben ser añadidos de manera tal que resulten prácticos, ya sea para explicar el flujo de eventos del código o el propósito de las funciones o variables.

### 2.4 Diagramas de componentes.

Los diagramas de componentes son los encargados de describir los componentes físicos del sistema así como sus relaciones. Estos incluyen: ficheros de código de fuente, ficheros ejecutables y otros ficheros de gran importancia para la implementación del sistema.

Los diagramas de componentes exponen las partes del software reusables, sus interfaces e interrelaciones; en muchos casos un diagrama de componentes se puede considerar como un diagrama de clases a gran escala. Cada uno de los componentes existentes en un diagrama ha de ser bien documentado en un diagrama de componentes más detallado, diagrama de clases o un diagrama de casos de uso.

Estos diagramas definen cómo los artefactos, las clases y otros elementos de bajo nivel se unifican para, crear así, componentes de alto nivel y sus respectivas conexiones entre ellos. Un componente es un artefacto de software compilado que funciona acoplado para ofrecer una funcionalidad requerida en el funcionamiento dentro de las restricciones delimitadas en el proceso de captura de requisitos.

A continuación se muestra el diagrama de componentes genéricos agrupados por paquetes:

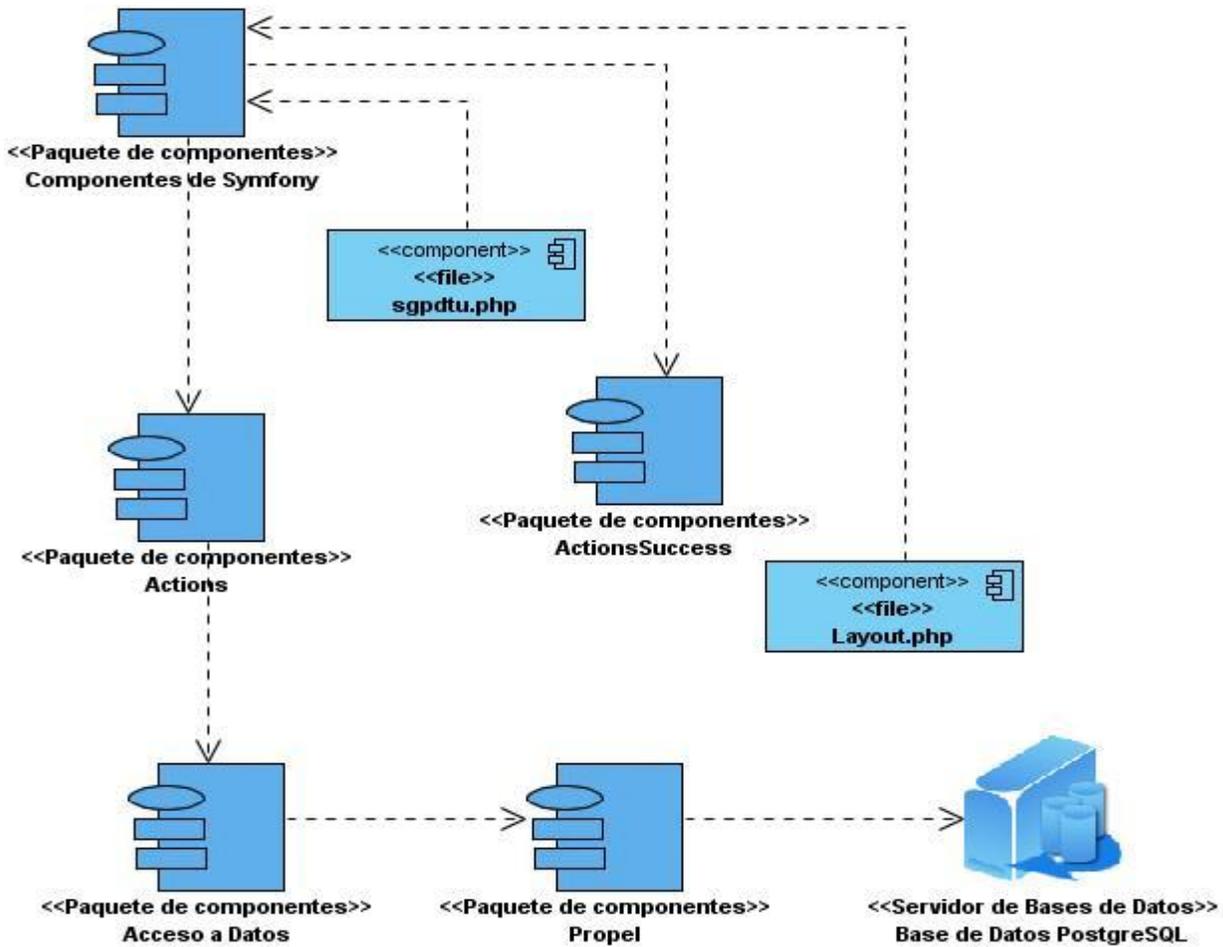


Figura: 5 Diagrama de componentes por paquetes.

En la representación del contenido individual de cada uno de los paquetes se obviaron los componentes que son archivos físicos, ya que estos no afectaban de manera directa la implementación, pues son parte del esquema del proyecto que genera el framework de forma automática.

A continuación se exponen los paquetes con los contenidos que complementan la solución.

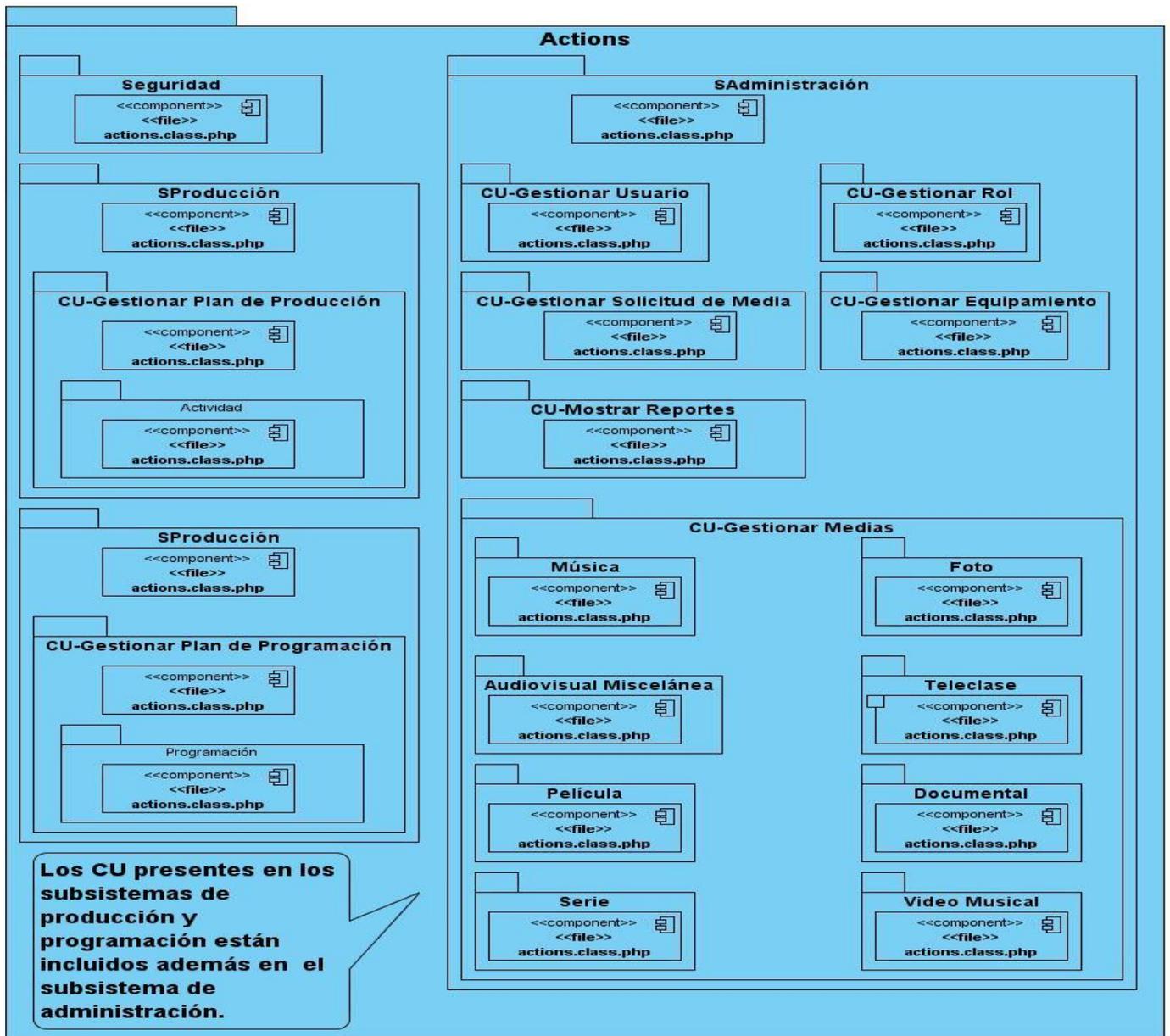


Figura 6: Paquete Actions.



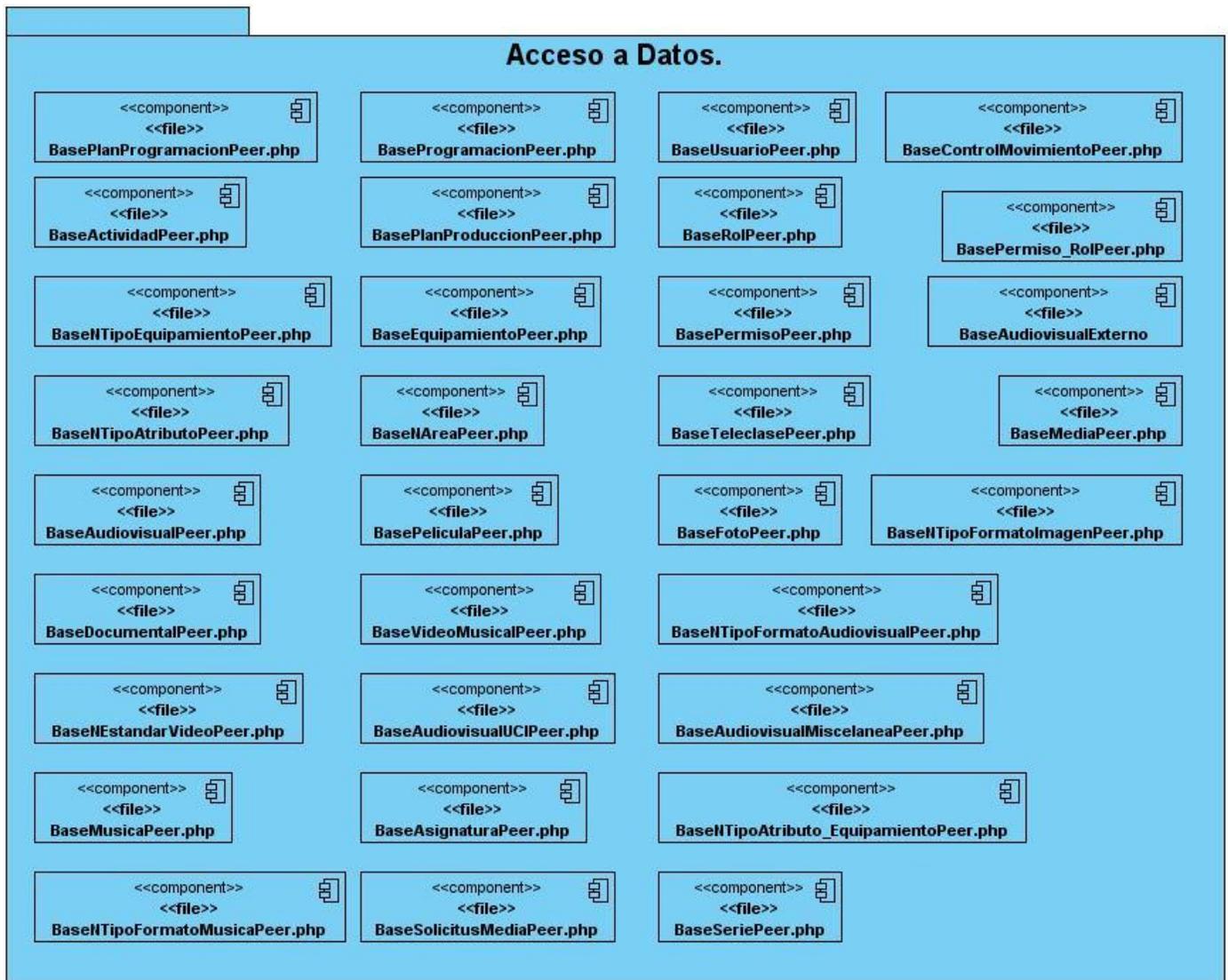


Figura 8: Paquete de acceso a datos.

En el caso particular del paquete de acceso a datos, se decidió representar sólo aquellas clases que aportaban información para las relaciones, para cada una de las tablas de la Base de Datos se generarían cuatro archivos físicos como se muestran a continuación.

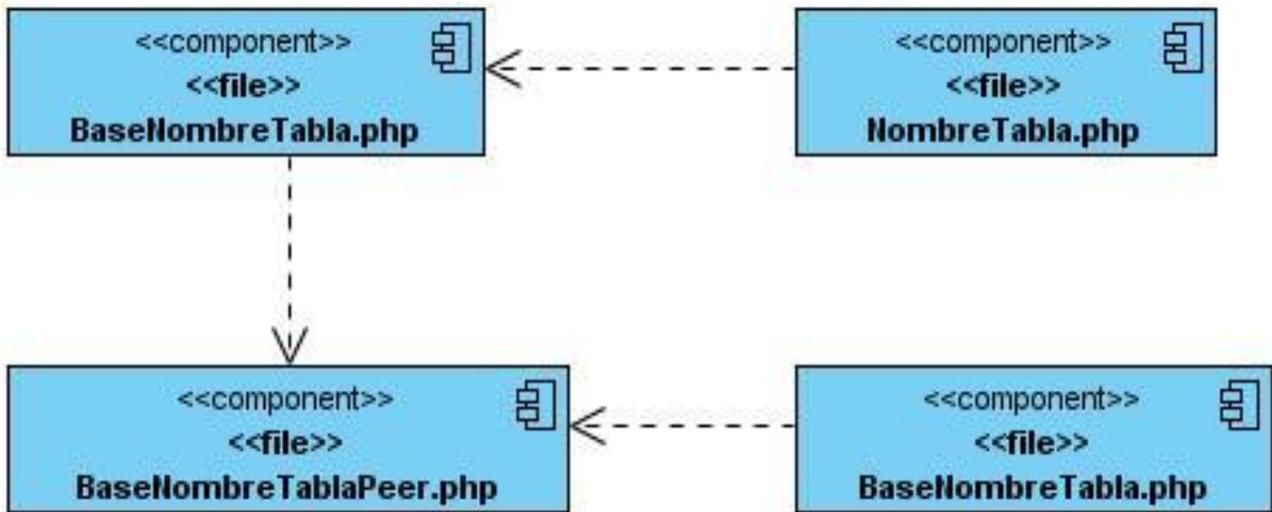


Figura 9: Archivos generados por symfony a partir de la Base de Datos.

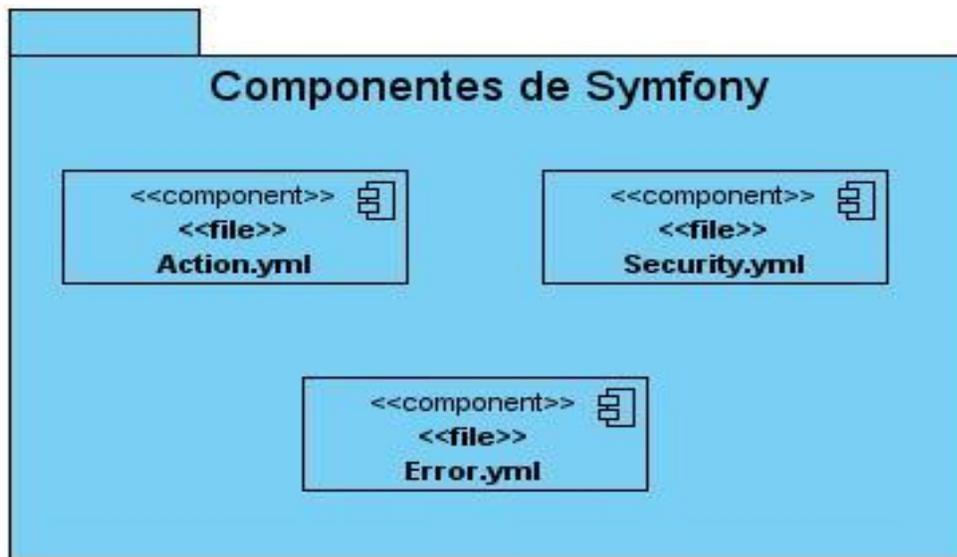


Figura 10: Paquete de symfony.

A continuación se muestran los diagramas de componentes para cada uno de los casos de usos descritos en el epígrafe 2.2.

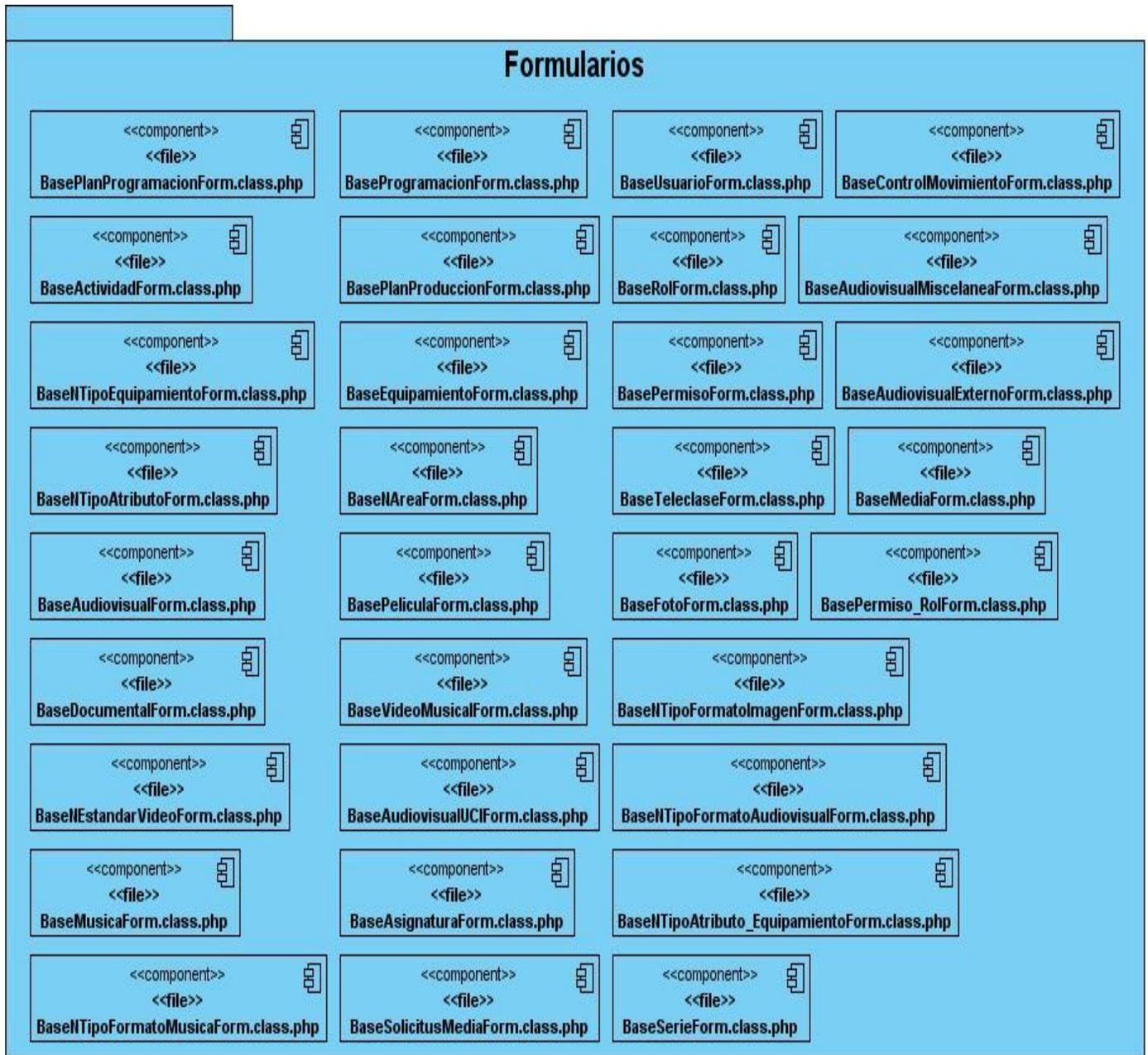


Figura 11: Formularios generados por Symfony.

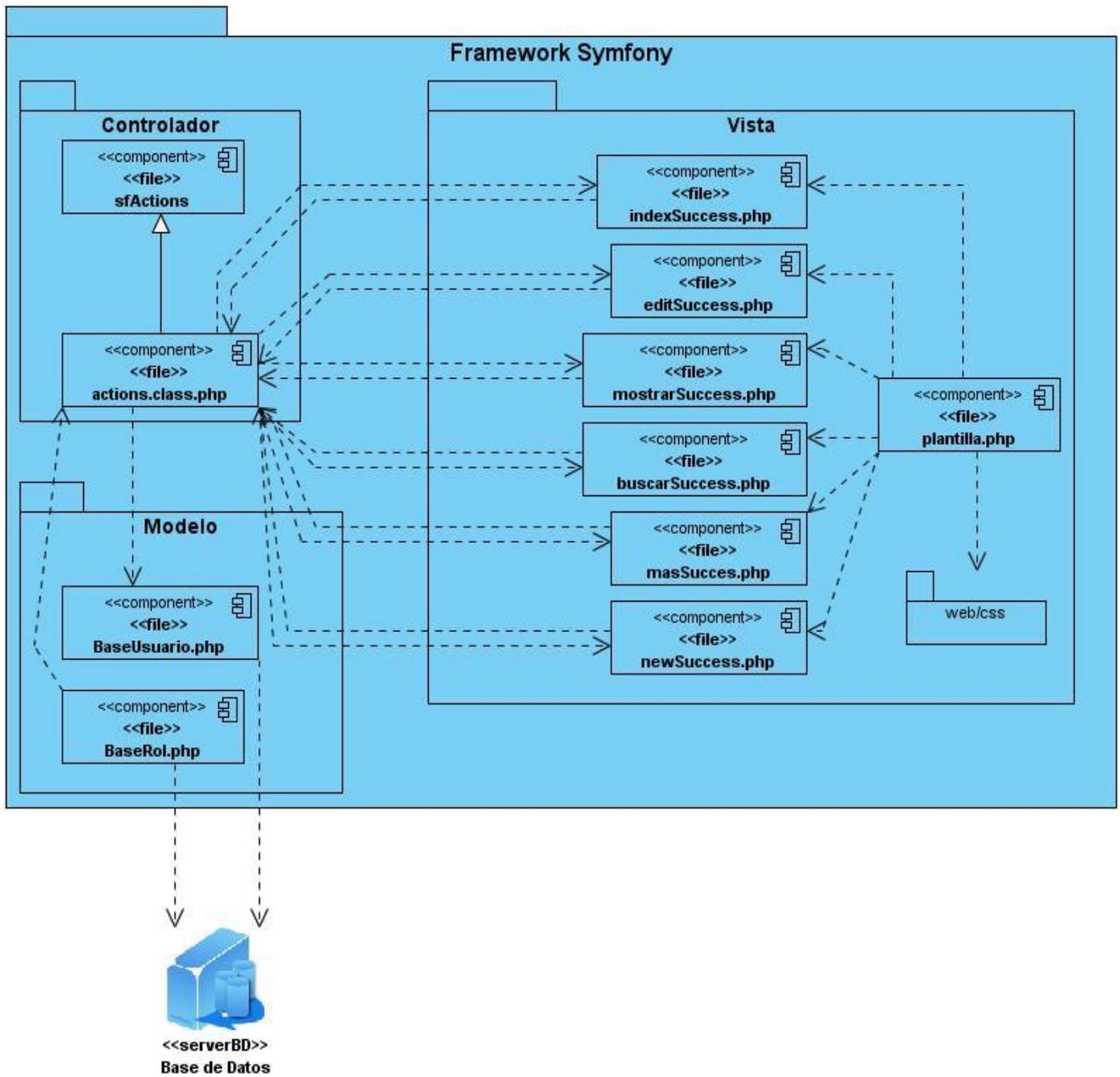


Figura 12: Diagrama de componentes del caso de uso Gestionar Usuarios.

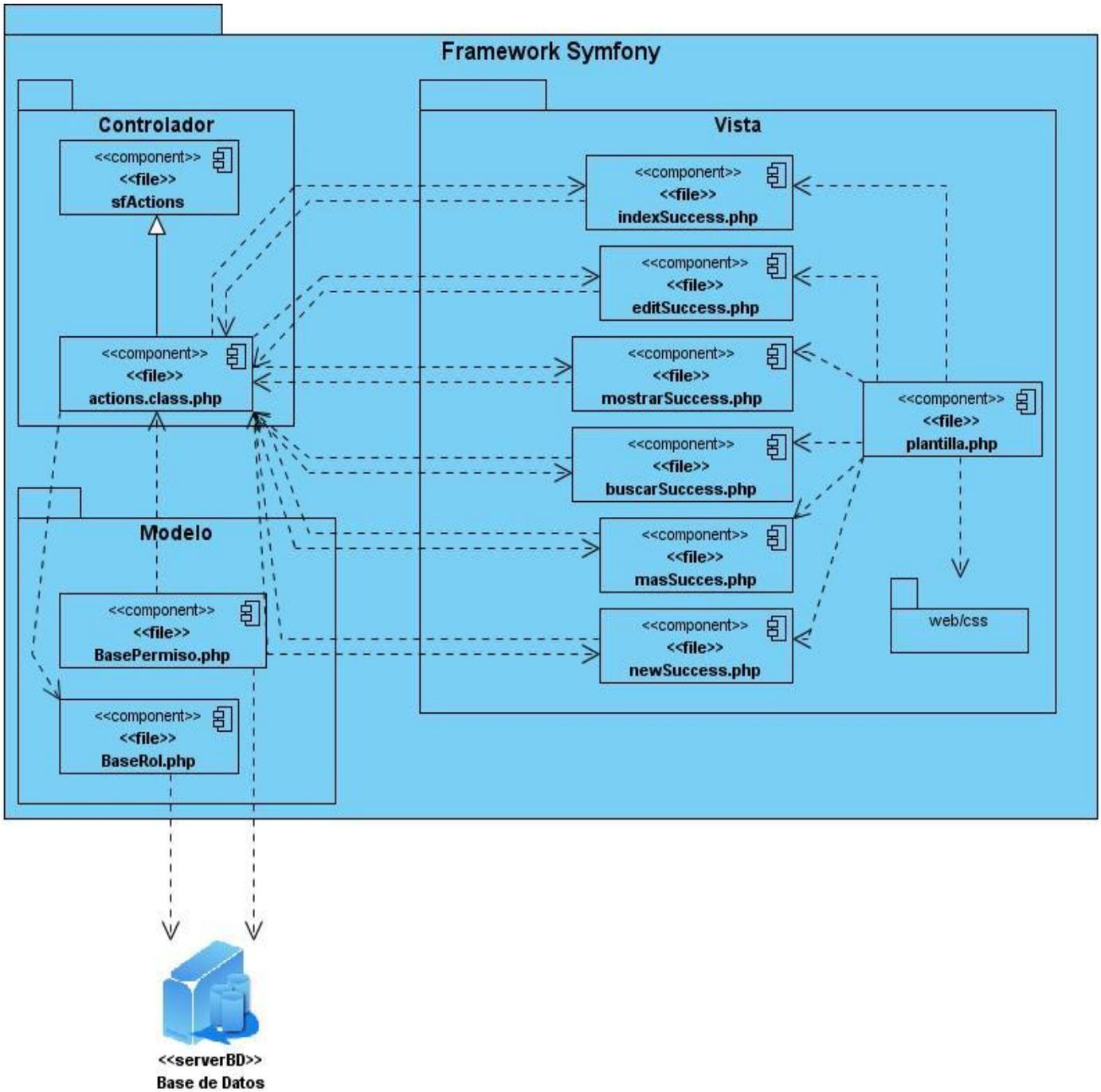


Figura 13: Diagrama de componentes del caso de uso Gestionar Rol.

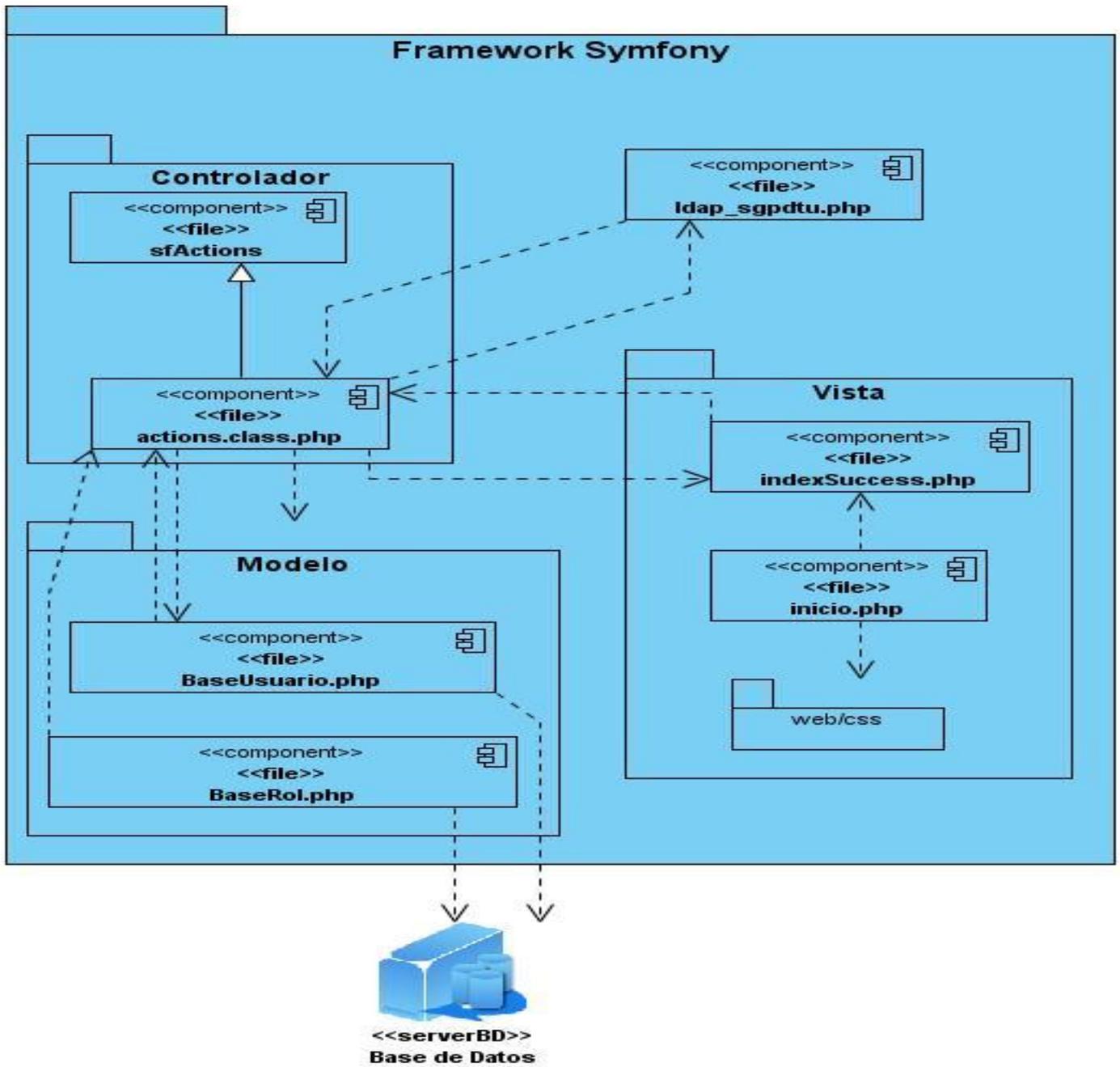


Figura 14: Diagrama de componentes del caso de uso Autenticar.

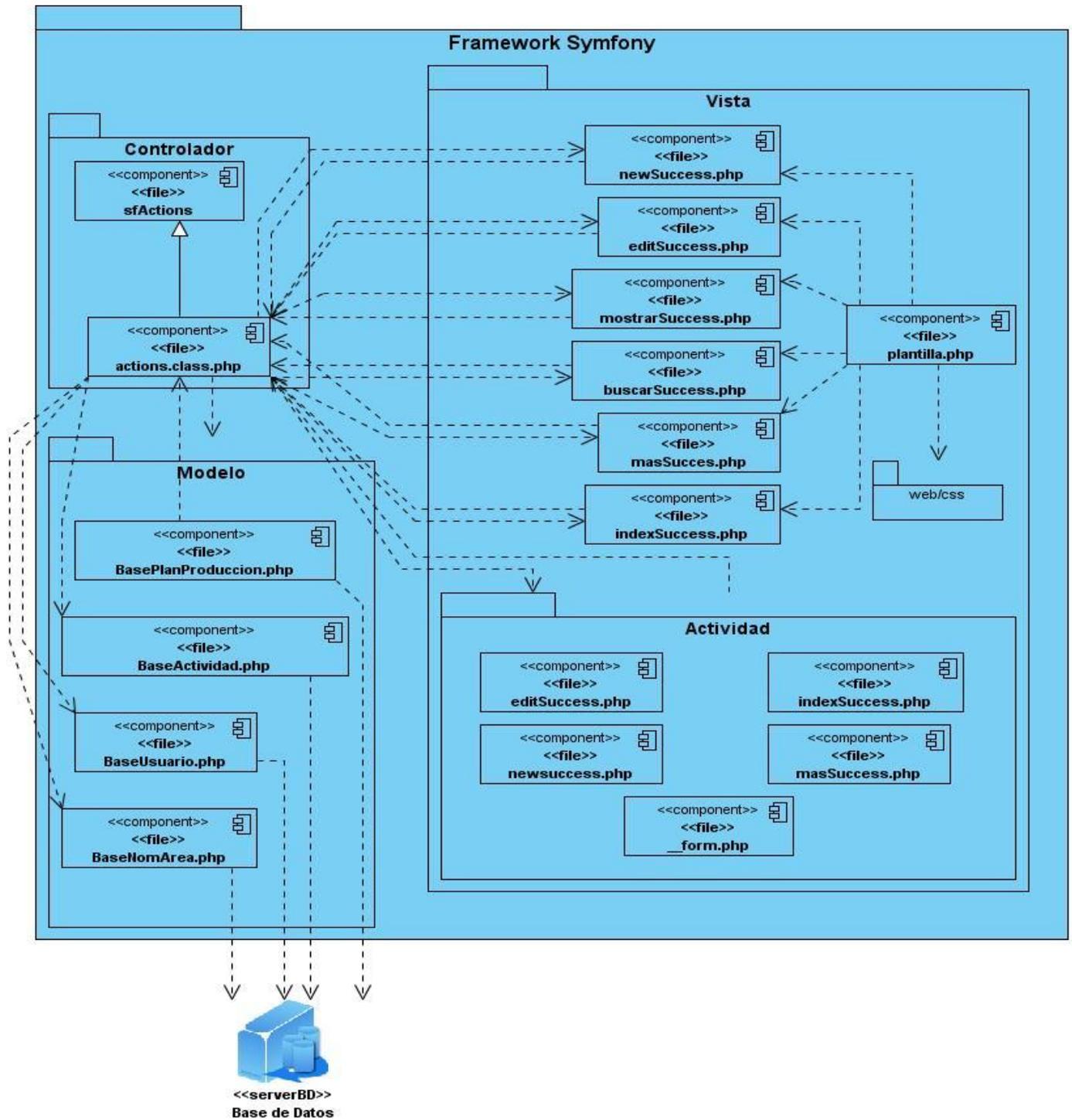


Figura 15: Diagrama de componentes del caso de uso Gestionar Plan de Producción.

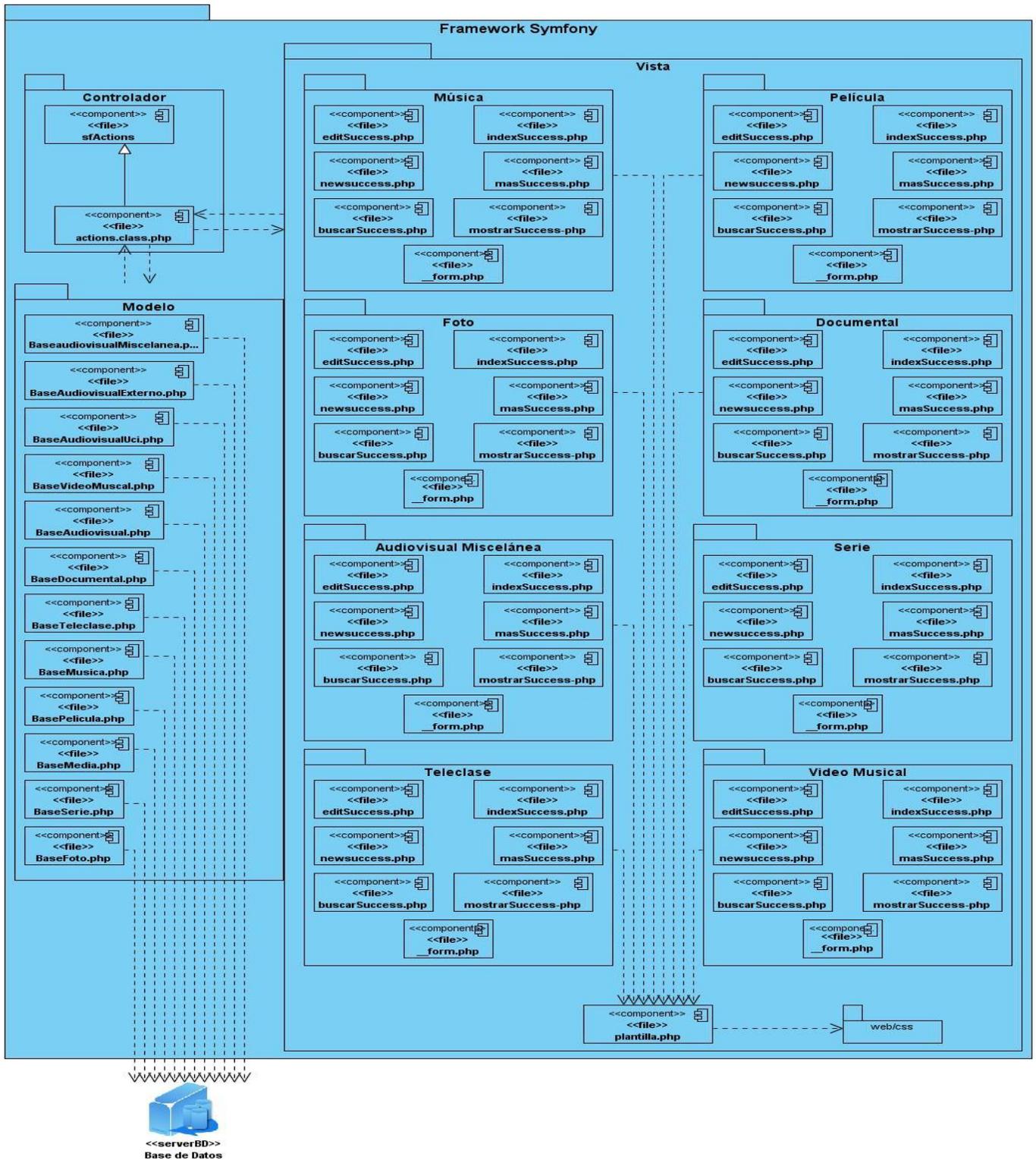


Figura 16: Diagrama de componentes del caso de uso Gestionar Media.

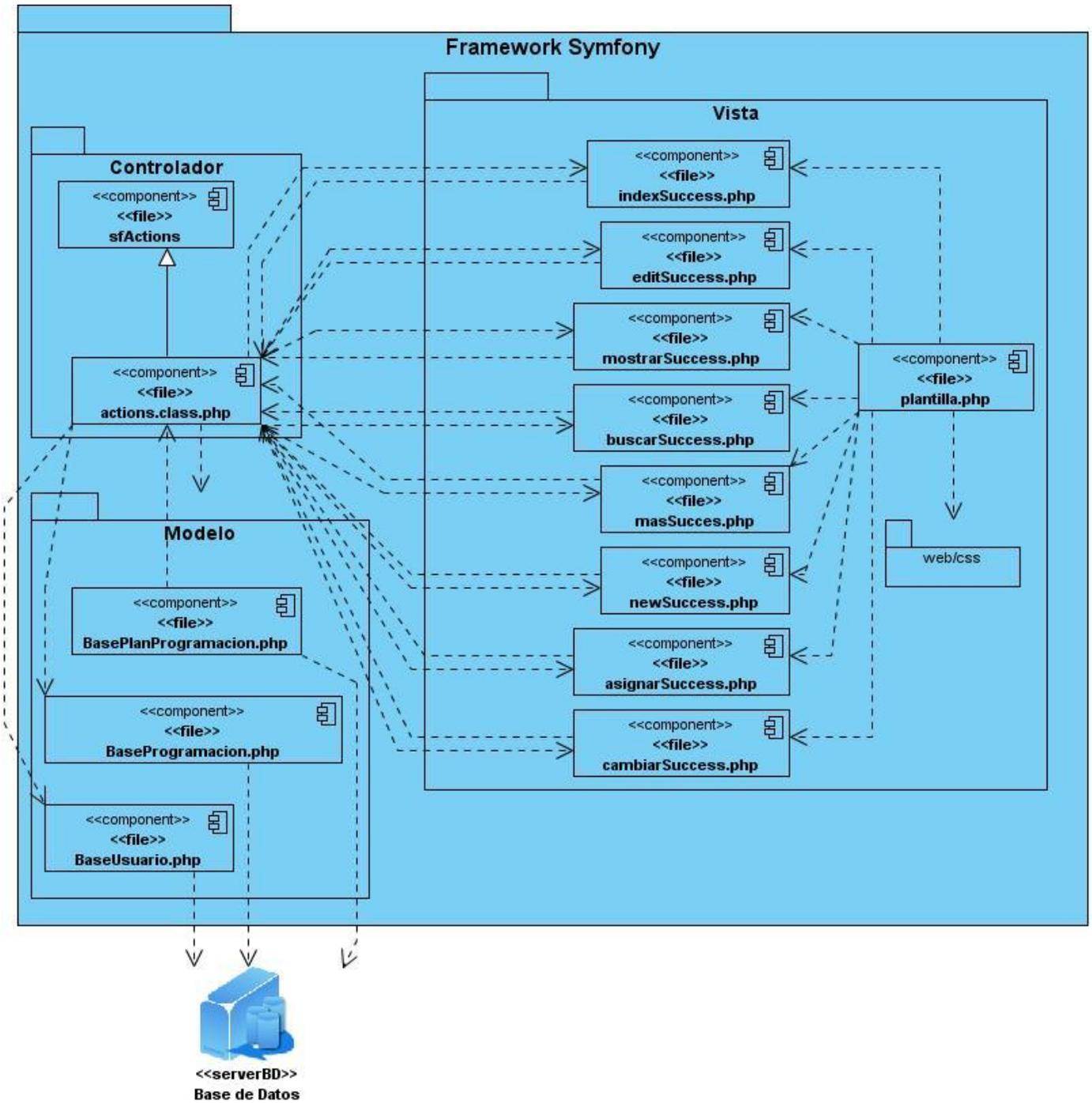


Figura 17: Diagrama de componentes del caso de uso Gestionar Equipamiento.

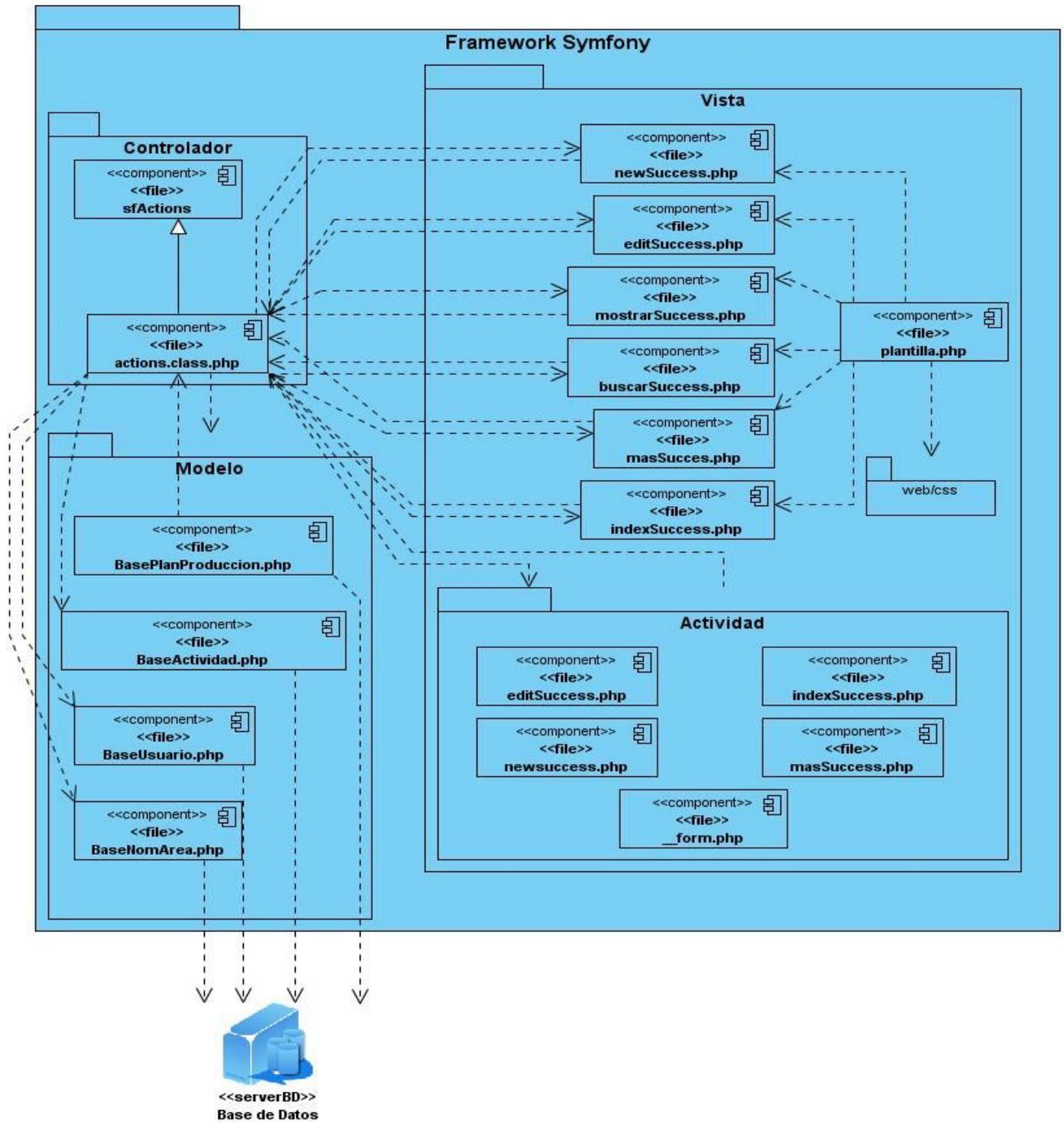


Figura 18: Diagrama de componentes del caso de uso Gestionar Plan de Programación.

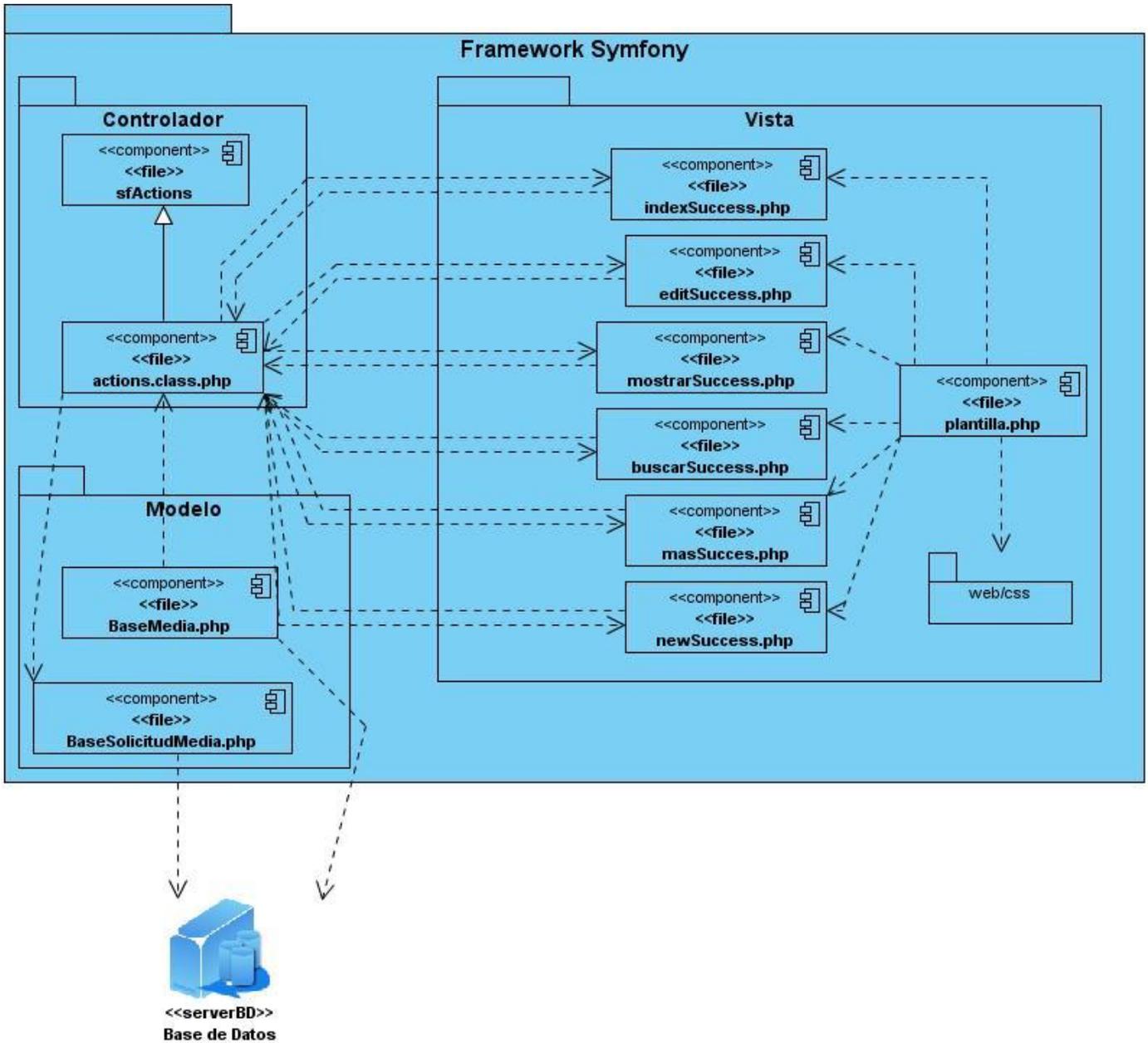


Figura 19: Diagrama de componentes del caso de uso Gestionar Solicitud de Medias.

## 2.5 Código fuente.

Los actions son la columna vertebral de las aplicaciones desarrolladas en symfony, puesto que engloban toda la lógica del sistema, comprueban la entereza de las peticiones y preparan los datos requeridos por la capa de presentación.

Estas utilizan para su desarrollo la capa del modelo y definen variables que serán utilizadas en la vista. La clase que representa las actions de un módulo se encuentra en el archivo actions.class.php, en el directorio actions/ del módulo.

### 2.5.1 Módulo Seguridad (Caso de uso Autenticar).

```
<?php
/** seguridad actions * @package proyecto * @subpackage seguridad
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 2692 2006-11-15 21:03:55Z fabien $ */
class seguridadActions extends sfActions{
public function executeIndex(){
public function executeAutenticar(sfWebRequest $request) {
    if($request->getMethod() == sfRequest::POST){
        {
            $c = new Criteria();
            $c->add(UsuarioPeer::USUARIO,$request->getParameter('usuario'));
            $user = UsuarioPeer::doSelect($c);
            $usuario;
            $idrol = 1;
            $rol = new Rol();
            foreach ($user as $u){
                $usuario = $u; }
            if(count($user)>0){
                $dominio = new sgpdtu_ldap();
                $us = $request->getParameter('usuario');
                $p = $request->getParameter('pass');
                If($dominio->login($us,$p)){
                    $this->getUser()->setAuthenticated(true);
                    $this->getUser()->setAttribute('usuario',$dominio->search($us));
                    $ccc = new Criteria();
                    $ccc->add(RolPermisoPeer::ID_ROL,$usuario->getIdRol());
                    $datos = RolPermisoPeer::doSelect($ccc);
                    $arreglo = array();
                    for($i=0;$i<count($datos);$i++){
                        $permiso = $datos[$i];
                        $arreglo[$i] = $permiso->getIdPermiso();
                        $this->getUser()-
                    >addCredential(PermisoPeer::retrieveByPK($permiso->getIdPermiso()->getNombre()));}
```

```

        $this->getUser()->setAttribute('rol',$arreglo);
        $this->redirect('sadministracion/index'); }
    else{
        $this->getUser()->setFlash('autenticar', 'Lo sentimos '.$request-
>getParameter('usuario').' falló la autenticación con el dominio. ');
        $this->redirect('seguridad/index'); }}
    else
    {
        $this->getUser()->setFlash('autenticar', 'Lo sentimos '.$request-
>getParameter('usuario').' pero usted no tiene acceso al sistema. ');
        $this->redirect('seguridad/index');}}}
public function executeCredencial(){
    $this->getUser()->setAuthenticated(false);
    $this->getUser()->clearCredentials();
    $this->redirect('seguridad/index');}
public function executeError404(){}

```

### 2.5.2 Módulo Administración (Caso de uso Gestionar Usuario).

```

<?php
/** * user actions. * @package proyecto * @subpackage user
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
function FancyTable($header,$data){
//Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont("",'B');
    //Cabecera
    $this->Ln(15);
    $w=array(40,35,40);
    $this->SetX(45);
    $x1 = $this->GetX();
    $y1 = $this->GetY();
    for($i=0;$i<count($header);$i++)
        $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
    $this->Ln();
    //Restauración de colores y fuentes
    $this->SetFillColor(255,255,255);
    $this->SetTextColor(0);
    $this->SetFont("");
    //Datos

```

```

$as=0;
$this->SetX(45);
foreach($data as $row) {
    $this->Line($this->GetX(),$this->GetY(),160,$this->GetY());
    $as=0;
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(85);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(120);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'R',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($as);
    $this->SetX($a); }
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->Line($x1,$y1,$x1,$this->GetY());
$this->Line(160,$y1,160,$this->GetY());
$this->Cell(array_sum($w),0,"T");}}
class userActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->usuario_list = UsuarioPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new UsuarioForm(); }
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new UsuarioForm();
    $this->processFormnuevo($request, $this->form);
    $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($usuario = UsuarioPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object usuario does not exist (%s).', $request->getParameter('id')));
    $this->form = new UsuarioForm($usuario);
    $this->usuario = new Usuario();
    $this->usuario = UsuarioPeer::retrieveByPK($request->getParameter('id'));}

```

```

public function executeUpdate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
    $this->forward404Unless($usuario = UsuarioPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object usuario does not exist (%s).', $request->getParameter('id')));
    $this->form = new UsuarioForm($usuario);
    $this->processFormmodificar($request, $this->form);
    $this->setTemplate('edit');}

public function executeDelete(sfWebRequest $request){
    $request->checkCSRFProtection();
    $this->forward404Unless($usuario = UsuarioPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object usuario does not exist (%s).', $request->getParameter('id')));
    $user = $usuario->getUsuario();
    $usuario->delete();
    $this->getUser()->setFlash('eliminar', 'El usuario '.$user.' ha sido eliminado correctamente. ');
    $this->redirect('user/index');}

protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $usuario = new Usuario();
        $a = $form->getValue('usuario');
        $usuario->setUsuario($a);
        $a = $form->getValue('contrasenna');
        $usuario->setContrasenna($a);
        $a = $form->getValue('id_rol');
        $usuario->setIdRol($a);
        $usuario->save();
        $this->getUser()->setFlash('nuevo', 'El usuario '.$usuario->getUsuario().' ha sido
insertado correctamente. ');
        $this->redirect('user/index');} }

protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $usuario = UsuarioPeer::retrieveByPK($form->getValue('id'));
        $a = $form->getValue('usuario');
        $usuario->setUsuario($a);
        $a = $form->getValue('contrasenna');
        $usuario->setContrasenna($a);
        $a = $form->getValue('id_rol');
        $usuario->setIdRol($a);
        $usuario->save();
    }
}

```

```
        $this->getUser()->setFlash('modificar', 'El usuario '.$usuario->getUsuario().' ha sido
modificado correctamente.');
```

```
        $this->redirect('user/index');} }
public function executeMas(sfWebRequest $request){
    $this->usuario = new Usuario();
    $this->usuario = UsuarioPeer::retrieveByPK($request->getParameter('id'));
    $this->roles = RolPeer::retrieveByPK($this->usuario->getIdRol());}
public function executeBuscar(){}
public function executeMostrar($request){
    $si = 0;
    $c=new Criteria();
    if($request->getParameter('usuario')!=") {
        $c->add(UsuarioPeer::USUARIO,$request->getParameter('usuario'));
        $si = 1;}
    if($request->getParameter('idrol')>0) {
        $c->add(UsuarioPeer::ID_ROL,$request->getParameter('idrol'));
        $si = 1;}
    if($si == 1){
        $this->datos=UsuarioPeer::doSelect($c);
        if(count($this->datos)>0) {
            $this->esta = 1;}
        else {
            $this->esta = 0;}}
    else{
        $this->esta = 0;}}
public function executeImprimir(){
    $con = $this->getUser()->getAttribute('imprimir');
    $pdf=new YPDF();
    //Títulos de las columnas
    $header=array('Id','Usuario','Rol');
    //Carga de datos
    $data[][]=array();
    $i=0;
    foreach($con as $c){
        $data[$i][0] = utf8_decode($c->getId());
        $data[$i][1] = utf8_decode($c->getUsuario());
        $data[$i][2] = utf8_decode(RolPeer::retrieveByPK($c->getIdRol())->getNombre());
        $i++;}

    $pdf->SetFont('Arial',"",14);
    $pdf->AddPage();
    $pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg',10,0,30);
```

```
$pdf->SetX(50);
$pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"false,");
$pdf->FancyTable($header,$data);
$pdf->Output(); }}
```

### 2.5.3 Módulo de Administración (Caso de uso Gestionar Rol).

```
<?php
/** * rol actions. * @package proyecto * @subpackage rol
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
//Colores, ancho de línea y fuente en negrita
$this->SetFillColor(25,185,255);
$this->SetTextColor(0);
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->SetFont('', 'B');
//Cabecera
$this->Ln(15);
$w=array(25,35);
$this->SetX(55);
$x1 = $this->GetX();
$y1 = $this->GetY();
for($i=0;$i<count($header);$i++)
    $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
$this->Ln();
//Restauración de colores y fuentes
$this->SetFillColor(255,255,255);
$this->SetTextColor(0);
$this->SetFont("");
//Datos
$as=0;
$this->SetX(55);
foreach($data as $row) {
    $this->Line($this->GetX(),$this->GetY(),115,$this->GetY());
    $as=0;
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
}
```

```

$this->SetY($b);
$this->SetX(80);
$this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
if($as<$this->GetY()) {
    $as = $this->GetY(); }
$this->SetY($as);
$this->SetX($a); }
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->Line($x1,$y1,$x1,$this->GetY());
$this->Line(115,$y1,115,$this->GetY());
$this->Cell(array_sum($w),0,"T");}}
class rolActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->rol_list = RolPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new RolForm();}
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new RolForm();
    $this->processFormnuevo($request, $this->form);
    $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($rol = RolPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object rol does not exist (%s).', $request->getParameter('id')));
    $this->form = new RolForm($rol);
    $this->rol = new Rol();
    $this->rol = RolPeer::retrieveByPK($request->getParameter('id'));}
public function executeUpdate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
    $this->forward404Unless($rol = RolPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object rol does not exist (%s).', $request->getParameter('id')));
    $this->form = new RolForm($rol);
    $this->processFormmodificar($request, $this->form);
    $this->setTemplate('edit');}
public function executeDelete(sfWebRequest $request){
    $request->checkCSRFProtection();
    $this->forward404Unless($rol = RolPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object rol does not exist (%s).', $request->getParameter('id')));
    $nombre = $rol->getNombre();
    $rol->delete();
    $this->getUser()->setFlash('eliminar', 'El rol '.$nombre.' ha sido eliminado correctamente.');
```

```

        $this->redirect('rol/index');}
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $rol = new Rol();
        $a = $form->getValue('nombre');
        $rol->setNombre($a);
        $rol->save();
        $this->getUser()->setFlash('nuevo', 'El rol '.$rol->getNombre().' ha sido insertado
correctamente.');
```

```

        $this->redirect('rol/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $rol = RolPeer::retrieveByPK($form->getValue('id'));
        $a = $form->getValue('nombre');
        $rol->setNombre($a);
        $rol->save();
        $this->getUser()->setFlash('modificar', 'El rol '.$rol->getNombre().' ha sido modificado
correctamente.');
```

```

        $this->redirect('rol/index');} }
public function executeMas(sfWebRequest $request){
    $this->rol = new Rol();
    $this->rol = RolPeer::retrieveByPK($request->getParameter('id'));}
public function executeBuscar(){ }
public function executeMostrar($request) {
    $si = 0;
    $c=new Criteria();
    if($request->getParameter('nombre')!=""){
        $c->add(RolPeer::NOMBRE,$request->getParameter('nombre'));
        $si = 1;}
    if($si==1){
        $this->datos = RolPeer::doSelect($c);
        if(count($this->datos)>0){
            $this->esta = 1;}
        else {
            $this->esta = 0;}}
    else{
        $this->esta = 0;} }
public function executeImprimir(){
```

```

$con = $this->getUser()->getAttribute('imprimir');
$pdf=new YPDF();
//Títulos de las columnas
$header=array('Id','Nombre');
//Carga de datos
$data[][]=array();
$i=0;
foreach($con as $c){
    $data[$i][0] = utf8_decode($c->getId());
    $data[$i][1] = utf8_decode($c->getNombre());
    $i++;}
$pdf->SetFont('Arial',"",14);
$pdf->AddPage();
$pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg',10,0,30);
$pdf->SetX(50);
$pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"false,");
$pdf->FancyTable($header,$data);
$pdf->Output(); }}

```

#### 2.5.4 Módulo de Administración (Caso de uso Gestionar Media (Foto)).

```

<?php
/** * foto actions.* @package proyecto * @subpackage foto
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
    //Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont("','B');
    //Cabecera
    $this->Ln(15);
    $w=array(18,23,30,33,27,40,26);
    $this->SetX(8);
    $x1 = $this->GetX();
    $y1 = $this->GetY();
    for($i=0;$i<count($header);$i++)
        $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
    $this->Ln();
}
}

```

```

//Restauración de colores y fuentes
$this->SetFillColor(255,255,255);
$this->SetTextColor(0);
$this->SetFont("");
//Datos
$as=0;
$this->SetX(8);
foreach($data as $row) {
    $as=0;
    $this->Line($this->GetX(),$this->GetY(),205,$this->GetY());
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(26);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(49);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(79);
    $this->MultiCell($w[3],6,$row[3],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(112);
    $this->MultiCell($w[4],6,$row[4],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(139);
    $this->MultiCell($w[5],6,$row[5],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(179);
    $this->MultiCell($w[6],6,$row[6],'LR',1,'L',false);
}

```

```

if($as<$this->GetY()) {
    $as = $this->GetY();
    $this->SetY($as);
    $this->SetX($a); }
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->Line($x1,$y1,$x1,$this->GetY());
$this->Line(26,$y1,26,$this->GetY());
$this->Line(49,$y1,49,$this->GetY());
$this->Line(79,$y1,79,$this->GetY());
$this->Line(112,$y1,112,$this->GetY());
$this->Line(139,$y1,139,$this->GetY());
$this->Line(179,$y1,179,$this->GetY());
$this->Line(205,$y1,205,$this->GetY());
$this->Cell(array_sum($w),0,"T");}}
class fotoActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->foto_list = FotoPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new FotoForm();}
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new FotoForm();
    $this->processFormnuevo($request, $this->form);
    $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($foto = FotoPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object foto does not exist (%s).', $request->getParameter('id')));
    $this->form = new FotoForm($foto);
    $this->foto = new Foto();
    $this->foto = FotoPeer::retrieveByPK($request->getParameter('id'));
    $this->media = new Media();
    $this->media = MediaPeer::retrieveByPK($request->getParameter('id'));
    $this->formato = NomTipoFormatoImagenPeer::retrieveByPK($this->foto->getIdFormato());
    $this->tipo = $this->formato->getNombre();}
public function executeUpdate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
    $this->forward404Unless($foto = FotoPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object foto does not exist (%s).', $request->getParameter('id')));
    $this->form = new FotoForm($foto);
    $this->processFormmodificar($request, $this->form);
    $this->setTemplate('edit');}

```

```

public function executeDeleted(sfWebRequest $request){
    $request->checkCSRFProtection();
    $this->forward404Unless($foto = FotoPeer::retrieveByPk($request->getParameter('id')),
sprintf('Object foto does not exist (%s).', $request->getParameter('id')));
    $media = MediaPeer::retrieveByPK($request->getParameter('id'));
    $foto = FotoPeer::retrieveByPK($request->getParameter('id'));
    $titulo = $media->getTitulo();
    $media->delete();
    $foto->delete();
    $this->getUser()->setFlash('eliminar', 'La foto '.$titulo.' ha sido eliminado correctamente.');
```

```

    $this->redirect('foto/index');}
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $media = new Media();
        $a = $form->getValue('titulo');
        $media->setTitulo($a);
        $a = $form->getValue('fecha_produccion');
        $media->setFechaProduccion($a);
        $a = $form->getValue('coddvd');
        $media->setCoddvd($a);
        $a = $form->getValue('estado');
        $media->setEstado($a);
        $a = $form->getValue('fuente');
        $media->setFuente($a);
        $a = $form->getValue('palabras_claves');
        $media->setPalabrasClaves($a);
        $media->save();
        /*Hasta aqui lo de media*/
        $llave = $media->getId();
        $foto = new Foto();
        $foto->setId($llave);
        $b = $form->getValue('fotografo');
        $foto->setFotografo($b);
        $b = $form->getValue('alto');
        $foto->setAlto($b);
        $b = $form->getValue('ancho');
        $foto->setAncho($b);
        $b = $form->getValue('tematica');
        $foto->setTematica($b);
        $b = $form->getValue('nombre_carpeta');
```

```

    $foto->setNombreCarpeta($b);
    $b = $form->getValue('responsabilidad');
    $foto->setResponsabilidad($b);
    $b = $form->getValue('periodista_asociado');
    $foto->setPeriodistaAsociado($b);
    $b = $form->getValue('color');
    $foto->setColor($b);
    $b = $form->getValue('personalidades');
    $foto->setPersonalidades($b);
    $b = $form->getValue('lugar');
    $foto->setLugar($b);
    $b = $form->getValue('comentario');
    $foto->setComentario($b);
    $b = $form->getValue('id_formato');
    $foto->setIdFormato($b);
    $foto->save();
    $this->getUser()->setFlash('nuevo', 'La foto '.$media->getTitulo().' ha sido insertada
correctamente.');
```

```

    $this->redirect('foto/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $media = MediaPeer::retrieveByPK($request->getParameter('id'));
        $a = $form->getValue('titulo');
        $media->setTitulo($a);
        $a = $form->getValue('fecha_produccion');
        $media->setFechaProduccion($a);
        $a = $form->getValue('coddvd');
        $media->setCoddvd($a);
        $a = $form->getValue('estado');
        $media->setEstado($a);
        $a = $form->getValue('fuente');
        $media->setFuente($a);
        $a = $form->getValue('palabras_claves');
        $media->setPalabrasClaves($a);
        $media->save();
        /*Hasta aqui lo de media*/
        $llave = $media->getId();
        $foto = FotoPeer::retrieveByPK($llave);
        $b = $form->getValue('fotografo');
        $foto->setFotografo($b);
    }
}

```

```

        $b = $form->getValue('alto');
        $foto->setAlto($b);
        $b = $form->getValue('ancho');
        $foto->setAncho($b);
        $b = $form->getValue('tematica');
        $foto->setTematica($b);
        $b = $form->getValue('nombre_carpeta');
        $foto->setNombreCarpeta($b);
        $b = $form->getValue('responsabilidad');
        $foto->setResponsabilidad($b);
        $b = $form->getValue('periodista_asociado');
        $foto->setPeriodistaAsociado($b);
        $b = $form->getValue('color');
        $foto->setColor($b);
        $b = $form->getValue('personalidades');
        $foto->setPersonalidades($b);
        $b = $form->getValue('lugar');
        $foto->setLugar($b);
        $b = $form->getValue('comentario');
        $foto->setComentario($b);
        $b = $form->getValue('id_formato');
        $foto->setIdFormato($b);
        $foto->save();
        $this->getUser()->setFlash('nuevo', 'La foto '.$media->getTitulo().' ha sido insertada
correctamente.');
```

```

        $this->redirect('foto/index');} }
public function executeMas(sfWebRequest $request){
    $this->foto = new Foto();
    $this->foto = FotoPeer::retrieveByPK($request->getParameter('id'));
    $this->media = new Media();
    $this->media = MediaPeer::retrieveByPK($request->getParameter('id'));
    $this->formato = NomTipoFormatoImagenPeer::retrieveByPK($this->foto->getIdFormato());
    $this->tipo = $this->formato->getNombre();}
public function executeBuscar(){ }
public function executeMostrar($request) {
    $si = 0;
    $c=new Criterio();
    if($request->getParameter('fotografo')!="){
        $c->add(FotoPeer::FOTOGRAFO,$request->getParameter('fotografo'));
        $si = 1;}
    if($request->getParameter('alto')!=") {
        $c->add(FotoPeer::ALTO,$request->getParameter('alto'));

```

```

        $si = 1;}
if($request->getParameter('ancho')!="){
    $c->add(FotoPeer::ANCHO,$request->getParameter('ancho'));
    $si = 1;}
if($request->getParameter('tematica')!="){
    $c->add(FotoPeer::TEMATICA,$request->getParameter('tematica'));
    $si = 1;}
if($request->getParameter('nombre_carpeta')!="){
    $c->add(FotoPeer::NOMBRE_CARPETA,$request->getParameter('nombre_carpeta'));
    $si = 1;}
if($request->getParameter('responsabilidad')!="){
    $c->add(FotoPeer::RESPONSABILIDAD,$request->getParameter('responsabilidad'));
    $si = 1;}
if($request->getParameter('periodista_asociado')!="){
    $c->add(FotoPeer::PERIODISTA_ASOCIADO,$request-
>getParameter('periodista_asociado'));
    $si = 1;}
if($request->getParameter('color')!="){
    $c->add(FotoPeer::COLOR,$request->getParameter('color'));
    $si = 1;}
if($request->getParameter('personalidades')!="){
    $c->add(FotoPeer::PERSONALIDADES,$request->getParameter('personalidades'));
    $si = 1;}
if($request->getParameter('lugar')!="){
    $c->add(FotoPeer::LUGAR,$request->getParameter('lugar'));
    $si = 1;}
if($request->getParameter('comentario')!=") {
    $c->add(FotoPeer::COMENTARIO,$request->getParameter('comentario'));
    $si = 1;}
if($request->getParameter('id_formato')!=0) {
    $c->add(FotoPeer::ID_FORMATO,$request->getParameter('id_formato'));
    $si = 1;}
if($si==1){
    $this->datos=FotoPeer::doSelect($c);
    if(count($this->datos)>0) {
        $this->esta = 1;}
    else {
        $this->esta = 0;}}
    else{
        $this->esta = 0;} }
public function executeImprimir() {
    $con = $this->getUser()->getAttribute('imprimir');

```

```

$pdf=new YPDF();
//Títulos de las columnas
$header=array('Id','DVD',utf8_decode('Fotógrafo'),'Alto X Ancho','Periodista','Lugar','Formato');
//Carga de datos
$data[][]=array();
$i=0;
foreach($con as $c) {
    $data[$i][0] = utf8_decode($c->getId());
    $data[$i][1] = utf8_decode(MediaPeer::retrieveByPK($c->getId()->getCoddvd());
    $data[$i][2] = utf8_decode($c->getFotografo());
    $data[$i][3] = utf8_decode($c->getAlto().' X '.$c->getAncho());
    $data[$i][4] = utf8_decode($c->getPeriodistaAsociado());
    $data[$i][5] = utf8_decode($c->getLugar());
    $data[$i][6] = utf8_decode(NomTipoFormatoImagenPeer::retrieveByPK($c->getIdFormato())-
>getNombre());
    $i++;}
$pdf->SetFont('Arial',"",14);
$pdf->AddPage();
$pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg',10,0,30);
$pdf->SetX(50);
$pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"",false,"");
$pdf->FancyTable($header,$data);
$pdf->Output(); }}

```

### 2.5.5 Módulo de Administración (Caso de uso Gestionar Solicitud de Media).

```

<?php
/** * solicitudmedia actions. * @package proyecto * @subpackage solicitudmedia
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
    //Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont("",'B');
    //Cabecera
    $this->Ln(15);
    $w=array(12,30,36,38,42,35);
    $x1 = $this->GetX();

```

```

$y1 = $this->GetY();
for($i=0;$i<count($header);$i++)
    $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
$this->Ln();
//Restauración de colores y fuentes
$this->SetFillColor(255,255,255);
$this->SetTextColor(0);
$this->SetFont("");
//Datos
$as=0;
foreach($data as $row) {
    $as=0;
    $this->Line($this->GetX(),$this->GetY(),203,$this->GetY());
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(22);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(52);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(88);
    $this->MultiCell($w[3],6,$row[3],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(126);
    $this->MultiCell($w[4],6,$row[4],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(168);
    $this->MultiCell($w[5],6,$row[5],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
}

```

```

        $this->SetY($as);
        $this->SetX($a);}
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->Line($x1,$y1,$x1,$this->GetY());
    $this->Line(22,$y1,22,$this->GetY());
    $this->Line(52,$y1,52,$this->GetY());
    $this->Line(88,$y1,88,$this->GetY());
    $this->Line(126,$y1,126,$this->GetY());
    $this->Line(168,$y1,168,$this->GetY());
    $this->Line(203,$y1,203,$this->GetY());
    $this->Cell(array_sum($w),0,"T");}}
class solicitudmediaActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->solicitud_media_list = SolicitudMediaPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new SolicitudMediaForm(); }
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new SolicitudMediaForm();
    $this->processFormnuevo($request, $this->form);
    $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($solicitud_media = SolicitudMediaPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object solicitud_media does not exist (%s).', $request->getParameter('id')));
    $this->form = new SolicitudMediaForm($solicitud_media);
    $this->solicitudmedia = new SolicitudMedia();
    $this->solicitudmedia = SolicitudMediaPeer::retrieveByPK($request->getParameter('id'));}
public function executeUpdater(sfWebRequest $request){
    if($request->getParameter('id_media')==0) {
        $this->redirect('solicitudmedia/index');}
    else{
        $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
        $this->forward404Unless($solicitud_media =
SolicitudMediaPeer::retrieveByPk($request->getParameter('id')), sprintf('Object solicitud_media does not
exist (%s).', $request->getParameter('id')));
        $this->form = new SolicitudMediaForm($solicitud_media);
        $this->processFormmodificar($request, $this->form);
        $this->setTemplate('edit');} }
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));}

```

```

if ($form->isValid()){
    $solicitudmedia = new SolicitudMedia();
    $a = $form->getValue('nombre_solicitante');
    $solicitudmedia->setNombreSolicitante($a);
    $a = $form->getValue('area_solicitante');
    $solicitudmedia->setAreaSolicitante($a);
    $a = $form->getValue('motivo');
    $solicitudmedia->setMotivo($a);
    $a = $form->getValue('fecha_solicitud');
    $solicitudmedia->setFechaSolicitud($a);
    $a = $form->getValue('fecha_necesidad');
    $solicitudmedia->setFechaNecesidad($a);
    $a = $form->getValue('descripcion');
    $solicitudmedia->setDescripcion($a);
    $a = $form->getValue('id_media');
    $solicitudmedia->setIdMedia($a);
    $solicitudmedia->save();
    $this->getUser()->setFlash('nuevo', 'La solicitud de media de '.$solicitudmedia-
>getNombreSolicitante().' ha sido insertada correctamente. ');
    $this->redirect('solicitudmedia/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $solicitudmedia = SolicitudMediaPeer::retrieveByPK($form->getValue('id'));
        $a = $form->getValue('nombre_solicitante');
        $solicitudmedia->setNombreSolicitante($a);
        $a = $form->getValue('area_solicitante');
        $solicitudmedia->setAreaSolicitante($a);
        $a = $form->getValue('motivo');
        $solicitudmedia->setMotivo($a);
        $a = $form->getValue('fecha_solicitud');
        $solicitudmedia->setFechaSolicitud($a);
        $a = $form->getValue('fecha_necesidad');
        $solicitudmedia->setFechaNecesidad($a);
        $a = $form->getValue('descripcion');
        $solicitudmedia->setDescripcion($a);
        $a = $form->getValue('id_media');
        $solicitudmedia->setIdMedia($a);
        $solicitudmedia->save();
        $this->getUser()->setFlash('modificar', 'La solicitud de media de '.$solicitudmedia-
>getNombreSolicitante().' ha sido modificada correctamente. ');
    }
}

```

```

        $this->redirect('solicitudmedia/index');} }
public function executeMas(sfWebRequest $request){
    $this->solicitudmedia = new SolicitudMedia();
    $this->solicitudmedia = SolicitudMediaPeer::retrieveByPK($request->getParameter('id'));
    $this->media = MediaPeer::retrieveByPK($this->solicitudmedia->getIdMedia());
    $this->titulo = $this->media->getTitulo();
public function executeBuscar(){ }
public function executeMostrar($request){
    $si = 0;
    $c = new Criteria();
    if($request->getParameter('nombre_solicitante')!="){
        $c->add(SolicitudMediaPeer::NOMBRE_SOLICITANTE,$request-
>getParameter('nombre_solicitante'));
        $si = 1;}
    if($request->getParameter('area_solicitante')!="){
        $c->add(SolicitudMediaPeer::AREA_SOLICITANTE,$request-
>getParameter('area_solicitante'));
        $si = 1;}
    if($request->getParameter('motivo')!="){
        $c->add(SolicitudMediaPeer::MOTIVO,$request->getParameter('motivo'));
        $si = 1;}
    $fecha = date("Y-m-d G:i",time());
    if($request->getParameter('fecha_solicitud')!="){
        $c->add(SolicitudMediaPeer::FECHA_SOLICITUD,$request-
>getParameter('fecha_solicitud'));
        $si = 1;}
    if($request->getParameter('fecha_necesidad')!="){
        $c->add(SolicitudMediaPeer::FECHA_NECESIDAD,$request-
>getParameter('fecha_necesidad'));
        $si = 1;}
    if($request->getParameter('descripcion')!=") {
        $c->add(SolicitudMediaPeer::DESCRIPCION,$request->getParameter('descripcion'));
        $si = 1;}
    if($request->getParameter('id_media')>0){
        $c->add(SolicitudMediaPeer::ID_MEDIA,$request->getParameter('id_media'));
        $si = 1;}
    if($si == 1){
        $this->datos=SolicitudMediaPeer::doSelect($c);
        if(count($this->datos)>0) {
            $this->esta = 1;}
        else {
            $this->esta = 0;}}

```

```

else{
    $this->esta = 0;} }
public function ejecutelImprimir() {
    $con = $this->getUser()->getAttribute('imprimir');
    $pdf=new YPDF();
    //Títulos de las columnas
    $header=array('Id','Solicitante','Motivo','Fecha Solicitud','Fecha Necesidad','Media');
    //Carga de datos
    $data[][]=array();
    $i=0;
    foreach($con as $c){
        $data[$i][0] = utf8_decode($c->getId());
        $data[$i][1] = utf8_decode($c->getNombreSolicitante());
        $data[$i][2] = utf8_decode($c->getMotivo());
        $data[$i][3] = utf8_decode($c->getFechaSolicitud());
        $data[$i][4] = utf8_decode($c->getFechaNecesidad());
        $data[$i][5] = utf8_decode(MediaPeer::retrieveByPK($c->getIdMedia()->getIdMedia());->getTitulo());
        $i++;}
    $pdf->SetFont('Arial',"",14);
    $pdf->AddPage();
    $pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg',10,0,30);
    $pdf->SetX(50);
    $pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"false,");
    $pdf->FancyTable($header,$data);
    $pdf->Output(); }}

```

### 2.5.6 Módulo de Administración (Caso de uso Gestionar Equipamiento).

```

<?php
/** * equipamiento actions.* @package proyecto * @subpackage equipamiento
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
    //Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont('','B');
    //Cabecera
    $this->Ln(15);

```

```

$w=array(12,30,36,38,42,35);
$x1 = $this->GetX();
$y1 = $this->GetY();
for($i=0;$i<count($header);$i++)
    $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
$this->Ln();
//Restauración de colores y fuentes
$this->SetFillColor(255,255,255);
$this->SetTextColor(0);
$this->SetFont("");
//Datos
$as=0;
foreach($data as $row) {
    $as=0;
    $this->Line($this->GetX(),$this->GetY(),203,$this->GetY());
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(22);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(52);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(88);
    $this->MultiCell($w[3],6,$row[3],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(126);
    $this->MultiCell($w[4],6,$row[4],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(168);
    $this->MultiCell($w[5],6,$row[5],'LR',1,'L',false);
}

```

```

if($as<$this->GetY()) {
    $as = $this->GetY(); }
    $this->SetY($as);
    $this->SetX($a); }
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->Line($x1,$y1,$x1,$this->GetY());
$this->Line(22,$y1,22,$this->GetY());
$this->Line(52,$y1,52,$this->GetY());
$this->Line(88,$y1,88,$this->GetY());
$this->Line(126,$y1,126,$this->GetY());
$this->Line(168,$y1,168,$this->GetY());
$this->Line(203,$y1,203,$this->GetY());
$this->Cell(array_sum($w),0,"T");}}
class equipamientoActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->equipamiento_list = EquipamientoPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new EquipamientoForm();}
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new EquipamientoForm();
    $this->processFormnuevo($request, $this->form);
    $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($equipamiento = EquipamientoPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object equipamiento does not exist (%s).', $request->getParameter('id')));
    $this->form = new EquipamientoForm($equipamiento);
    $this->equipamiento = new Equipamiento();
    $this->equipamiento = EquipamientoPeer::retrieveByPK($request->getParameter('id'));
    $this->tipoequipamiento = NomTipoEquipamientoPeer::retrieveByPK($this->equipamiento-
>getIdTipoEquipamiento());
    $this->tipo = $this->tipoequipamiento->getNombre();}
public function executeUpdate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
    $this->forward404Unless($equipamiento = EquipamientoPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object equipamiento does not exist (%s).', $request->getParameter('id')));
    $this->form = new EquipamientoForm($equipamiento);
    $this->processFormmodificar($request, $this->form);
    $this->setTemplate('edit');}
public function executeDelete(sfWebRequest $request){
    $request->checkCSRFProtection();}

```

```

        $this->forward404Unless($equipamiento = EquipamientoPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object equipamiento does not exist (%s).', $request->getParameter('id')));
        $equipamiento->delete();
        $this->redirect('equipamiento/index');}
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $equipamiento = new Equipamiento();
        $a = $form->getValue('no_serie');
        $equipamiento->setNoSerie($a);
        $a = $form->getValue('no_inventario');
        $equipamiento->setNoInventario($a);
        $a = $form->getValue('descripcion_medio');
        $equipamiento->setDescripcionMedio($a);
        $a = $form->getValue('observaciones');
        $equipamiento->setObservaciones($a);
        $a = $form->getValue('fecha_registro');
        $equipamiento->setFechaRegistro($a);
        $a = $form->getValue('id_tipo_equipamiento');
        $equipamiento->setIdTipoEquipamiento($a);
        $equipamiento->save();
        $this->redirect('equipamiento/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $equipamiento = new Equipamiento();
        $equipamiento = EquipamientoPeer::retrieveByPK($request->getParameter('id'));
        $a = $form->getValue('no_serie');
        $equipamiento->setNoSerie($a);
        $a = $form->getValue('no_inventario');
        $equipamiento->setNoInventario($a);
        $a = $form->getValue('descripcion_medio');
        $equipamiento->setDescripcionMedio($a);
        $a = $form->getValue('observaciones');
        $equipamiento->setObservaciones($a);
        $a = $form->getValue('fecha_registro');
        $equipamiento->setFechaRegistro($a);
        $a = $form->getValue('id_tipo_equipamiento');
        $equipamiento->setIdTipoEquipamiento($a);
        $equipamiento->save();
    }
}

```

```

        $this->redirect('equipamiento/index'); } }
public function executeMas(sfWebRequest $request){
    $this->equipamiento = new Equipamiento();
    $this->equipamiento = EquipamientoPeer::retrieveByPK($request->getParameter('id'));
    $this->usuario = new Usuario();
    $this->usuario = UsuarioPeer::retrieveByPK($this->equipamiento->getIdUsuario());
    $this->tiene = 0;
    $this->user = "";
    if(count($this->usuario)){
        $this->tiene=1;}
    if($this->tiene == 1){
        $this->user = $this->usuario->getUsuario();}
    $this->tipoequipamiento = new NomTipoEquipamiento();
    $this->tipoequipamiento = NomTipoEquipamientoPeer::retrieveByPK($this->equipamiento-
>getIdTipoEquipamiento());
    $this->tipo = $this->tipoequipamiento->getNombre();}
    public function executeBuscar(){ }
public function executeMostrar($request) {
    $c = new Criterias();
    $i = 0;
    if($request->getParameter('no_serie')!="){
        $c->add(EquipamientoPeer::NO_SERIE,$request->getParameter('no_serie'));
        $i = 1;}
    if($request->getParameter('no_inventario')!="){
        $c->add(EquipamientoPeer::NO_INVENTARIO,$request->getParameter('no_inventario'));
        $i = 1;}
    if($request->getParameter('descripcion_medio')!=") {
        $c->add(EquipamientoPeer::DESCRIPCION_MEDIO,$request-
>getParameter('descripcion_medio'));
        $i = 1;}
    if($request->getParameter('observaciones')!="){
        $c->add(EquipamientoPeer::OBSERVACIONES,$request->getParameter('observaciones'));
        $i = 1;}
    if($request->getParameter('fecha_registro')!="){
        $c->add(EquipamientoPeer::FECHA_REGISTRO,$request-
>getParameter('fecha_registro'));
        $i = 1;}
    if($request->getParameter('id_equipamiento')!=0){
        $c->add(EquipamientoPeer::ID_TIPO_EQUIPAMIENTO,$request-
>getParameter('id_equipamiento'));
        $i = 1;}
    if($request->getParameter('id_usuario')!=0){

```

```

        $c->add(EquipamientoPeer::ID_USUARIO,$request->getParameter('id_usuario'));
        $i = 1;}
    if($i == 1){
        $this->datos=EquipamientoPeer::doSelect($c);
        if(count($this->datos)>0) {
            $this->esta = 1;}
        else {
            $this->esta = 0;}}
    else{
        $this->esta = 0; } }
public function executeImprimir() {
    $con = $this->getUser()->getAttribute('imprimir');
    $pdf=new YPDF();
    //Títulos de las columnas
    $header=array('Id','No Serie','No Inventario',utf8_decode('Descripción'),'Fecha Registro','Usuario');
    //Carga de datos
    $data[][]=array();
    $i=0;
    foreach($con as $c){
        $data[$i][0] = utf8_decode($c->getId());
        $data[$i][1] = utf8_decode($c->getNoSerie());
        $data[$i][2] = utf8_decode($c->getNoInventario());
        $data[$i][3] = utf8_decode($c->getDescripcionMedio());
        $data[$i][4] = utf8_decode($c->getFechaRegistro());
        if($c->getUsuario()!=""){
            $data[$i][5] = utf8_decode(UsuarioPeer::retrieveByPK($c->getIdUsuario()->getUsuario());)
        }
        else{
            $data[$i][5] = 'No Asignado';}
        $i++;}
    $pdf->SetFont('Arial',"",14);
    $pdf->AddPage();
    $pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg',10,0,30);
    $pdf->SetX(50);
    $pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"",false,"");
    $pdf->FancyTable($header,$data);
    $pdf->Output(); }
public function executeAsignar($request){
    $this->idequipo = $this->getRequest()->getParameter('id');
    $this->datos = UsuarioPeer::doSelect(new Criteria());}
public function executeCambiar($request){
    $this->idequipo = $this->getRequest()->getParameter('id');
    $this->datos = UsuarioPeer::doSelect(new Criteria());}

```

```
public function executeAsignar1($request){
    $usuario = $request->getParameter('usuario');
    if($usuario>0){
        $equipamiento = new Equipamiento();
        $equipamiento = EquipamientoPeer::retrieveByPK($request->getParameter('id'));
        $equipamiento->setIdUsuario($usuario);
        $equipamiento->save();
        $this->redirect('equipamiento/index');}
    else{
        $equipamiento = new Equipamiento();
        $equipamiento = EquipamientoPeer::retrieveByPK($request->getParameter('id'));
        $equipamiento->setIdUsuario(NULL);
        $equipamiento->save();
        $this->redirect('equipamiento/index');} } }
```

### 2.5.7 Módulo de Administración (Caso de uso Gestionar Plan de Programación).

```
<?php
/** * planprogramacion actions.* @package proyecto * @subpackage planprogramacion
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
    //Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont('', 'B');
    //Cabecera
    $this->Ln(15);
    $w=array(22,42,40,30);
    $this->SetX(30);
    $x1 = $this->GetX();
    $y1 = $this->GetY();
    for($i=0;$i<count($header);$i++)
        $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
    $this->Ln();
    //Restauración de colores y fuentes
    $this->SetFillColor(255,255,255);
    $this->SetTextColor(0);
    $this->SetFont("");}
```

```

//Datos
$as=0;
$this->SetX(30);
foreach($data as $row) {
    $as=0;
    $this->Line($this->GetX(),$this->GetY(),134,$this->GetY());
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(52);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(94);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY();}
    $this->SetY($b);
    $this->SetX(134);
    $this->MultiCell($w[3],6,$row[3],'LR',1,'L',false);
    if($as<$this->GetY()){
        $as = $this->GetY(); }
    $this->SetY($as);
    $this->SetX($a); }
$this->SetLineWidth(.3);
$this->SetDrawColor(128,0,0);
$this->Line($x1,$y1,$x1,$this->GetY());
$this->Line(52,$y1,52,$this->GetY());
$this->Line(94,$y1,94,$this->GetY());
$this->Line(134,$y1,134,$this->GetY());
$this->Cell(array_sum($w),0,"','T');}}
class planprogramacionActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->plan_programacion_list = PlanProgramacionPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new PlanProgramacionForm();}
public function executeCreate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post'));
    $this->form = new PlanProgramacionForm();}

```

```

        $this->processFormnuevo($request, $this->form);
        $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
    $this->forward404Unless($plan_programacion =
PlanProgramacionPeer::retrieveByPk($request->getParameter('id')), sprintf('Object plan_programacion
does not exist (%s).', $request->getParameter('id')));
    $this->form = new PlanProgramacionForm($plan_programacion);
    $this->planprogramacion = new PlanProgramacion();
    $this->planprogramacion = PlanProgramacionPeer::retrieveByPK($request-
>getParameter('id'));}
public function executeUpdate(sfWebRequest $request){
    $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
    $this->forward404Unless($plan_programacion =
PlanProgramacionPeer::retrieveByPk($request->getParameter('id')), sprintf('Object plan_programacion
does not exist (%s).', $request->getParameter('id')));
    $this->form = new PlanProgramacionForm($plan_programacion);
    $this->processFormmodificar($request, $this->form);
    $this->setTemplate('edit');}
public function executeDelete(sfWebRequest $request){
    $request->checkCSRFProtection();
    $this->forward404Unless($plan_programacion =
PlanProgramacionPeer::retrieveByPk($request->getParameter('id')), sprintf('Object plan_programacion
does not exist (%s).', $request->getParameter('id')));
    $plan_programacion->delete();
    $this->redirect('planprogramacion/index');}
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){
        $planprogamacion = new PlanProgramacion();
        $a = $form->getValue('fecha_creacion');
        $planprogamacion->setFechaCreacion($a);
        $a = $form->getValue('fecha_inicio');
        $planprogamacion->setFechaInicio($a);
        $a = $form->getValue('id_usuario');
        $planprogamacion->setIdUsuario($a);
        $planprogamacion->save();
        $this->redirect('planprogramacion/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
    $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
    if ($form->isValid()){

```

```

        $planprogramacion = PlanProgramacionPeer::retrieveByPK($request-
>getParameter('id'));
        $a = $form->getValue('fecha_creacion');
        $planprogramacion->setFechaCreacion($a);
        $a = $form->getValue('fecha_inicio');
        $planprogramacion->setFechaInicio($a);
        $a = $form->getValue('id_usuario');
        $planprogramacion->setIdUsuario($a);
        $planprogramacion->save();
        $this->redirect('planprogramacion/index');} }

public function executeBuscar(){ }
public function executeMostrar($request) {
    $si = 0;
    $c= new Criteria();
    if($request->getParameter('fecha_creacion')!=''){
        $c->add(PlanProgramacionPeer::FECHA_CREACION,$request-
>getParameter('fecha_creacion'));
        $si = 1;}
    if($request->getParameter('fecha_inicio')!='){
        $c->add(PlanProgramacionPeer::FECHA_INICIO,$request->getParameter('fecha_inicio'));
        $si = 1;}
    if($request->getParameter('id_usuario')!=') {
        $c->add(PlanProgramacionPeer::ID_USUARIO,$request->getParameter('id_usuario'));
        $si = 1;}
    if($si == 1){
        $this->datos=PlanProgramacionPeer::doSelect($c);
        if(count($this->datos)>0) {
            $this->esta = 1;}
        else {
            $this->esta = 0;}}
    else{
        $this->esta = 0;}
    $this->fecha = $request->getParameter('fecha_creacion'); }
public function executeImprimir() {
    $con = $this->getUser()->getAttribute('imprimir');
    $pdf=new YPDF();
    //Títulos de las columnas
    $header=array('Id','utf8_decode('Fecha Creación'),'Fecha Inicio','Usuario');
    //Carga de datos
    $data[][]=array();
    $i=0;
    foreach($con as $c){

```

```

$data[$i][0] = utf8_decode($c->getId());
$data[$i][1] = utf8_decode($c->getFechaCreacion());
$data[$i][2] = utf8_decode($c->getFechaInicio());
$data[$i][3] = utf8_decode(UsuarioPeer::retrieveByPK($c->getIdUsuario())->getUsuario());
$i++;}
$pdf->SetFont('Arial', "", 14);
$pdf->AddPage();
$pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg', 10,0,30);
$pdf->SetX(50);
$pdf->Cell(6,0,'Reporte del sistema del '.date('d/m/Y h:i:s A').'.',0,0,"",false,"");
$pdf->FancyTable($header,$data);
$pdf->Output(); }

public function executeMas(sfWebRequest $request){
    $c=new Criteria();
    $c->add(ProgramacionPeer::ID_PLAN,$request->getParameter('id'));
    $this->datos=ProgramacionPeer::doSelect($c);} }

```

### 2.5.8 Módulo de Administración (Caso de uso Gestionar Plan de Producción).

```

<?php
/** * planproduccion actions.* @package proyecto * @subpackage planproduccion
 * @author Yusbel Ramos Núñez
 * @version SVN: $Id: actions.class.php 12474 2008-10-31 10:41:27Z fabien $ */
class YPDF extends FPDF{
//Tabla coloreada
function FancyTable($header,$data){
    //Colores, ancho de línea y fuente en negrita
    $this->SetFillColor(25,185,255);
    $this->SetTextColor(0);
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->SetFont("",'B');
    //Cabecera
    $this->Ln(15);
    $w=array(22,42,40,30);
    $this->SetX(30);
    $x1 = $this->GetX();
    $y1 = $this->GetY();
    for($i=0;$i<count($header);$i++)
        $this->Cell($w[$i],7,$header[$i],1,0,'C',1);
    $this->Ln();
    //Restauración de colores y fuentes
    $this->SetFillColor(255,255,255);

```

```

$this->SetTextColor(0);
$this->SetFont("");
//Datos
$as=0;
$this->SetX(30);
foreach($data as $row){
    $as=0;
    $this->Line($this->GetX(),$this->GetY(),134,$this->GetY());
    $b=$this->GetY();
    $a=$this->GetX();
    $this->MultiCell($w[0],6,$row[0],'LR',1,'R',false);
    $as = $this->GetY();
    $this->SetY($b);
    $this->SetX(52);
    $this->MultiCell($w[1],6,$row[1],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(94);
    $this->MultiCell($w[2],6,$row[2],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($b);
    $this->SetX(134);
    $this->MultiCell($w[3],6,$row[3],'LR',1,'L',false);
    if($as<$this->GetY()) {
        $as = $this->GetY(); }
    $this->SetY($as);
    $this->SetX($a); }
    $this->SetLineWidth(.3);
    $this->SetDrawColor(128,0,0);
    $this->Line($x1,$y1,$x1,$this->GetY());
    $this->Line(52,$y1,52,$this->GetY());
    $this->Line(94,$y1,94,$this->GetY());
    $this->Line(134,$y1,134,$this->GetY());
    $this->Cell(array_sum($w),0,"','T');}}
class planproduccionActions extends sfActions{
public function executeIndex(sfWebRequest $request){
    $this->plan_produccion_list = PlanProduccionPeer::doSelect(new Criteria());}
public function executeNew(sfWebRequest $request){
    $this->form = new PlanProduccionForm();}
public function executeCreate(sfWebRequest $request){

```

```

        $this->forward404Unless($request->isMethod('post'));
        $this->form = new PlanProduccionForm();
        $this->processFormnuevo($request, $this->form);
        $this->setTemplate('new');}
public function executeEdit(sfWebRequest $request){
        $this->forward404Unless($planproduccion = PlanProduccionPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object plan_produccion does not exist (%s).', $request->getParameter('id')));
        $this->form = new PlanProduccionForm($planproduccion);
        $this->planproduccion = new PlanProduccion();
        $this->planproduccion = PlanProduccionPeer::retrieveByPK($request->getParameter('id'));}
public function executeUpdate(sfWebRequest $request){
        $this->forward404Unless($request->isMethod('post') || $request->isMethod('put'));
        $this->forward404Unless($planproduccion = PlanProduccionPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object plan_produccion does not exist (%s).', $request->getParameter('id')));
        $this->form = new PlanProduccionForm($planproduccion);
        $this->processFormmodificar($request, $this->form);
        $this->setTemplate('edit');}
public function executeDelete(sfWebRequest $request){
        $request->checkCSRFProtection();
        $this->forward404Unless($plan_produccion = PlanProduccionPeer::retrieveByPk($request-
>getParameter('id')), sprintf('Object plan_produccion does not exist (%s).', $request->getParameter('id')));
        $plan_produccion->delete();
        $this->redirect('planproduccion/index');}
protected function processFormnuevo(sfWebRequest $request, sfForm $form){
        $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
        if ($form->isValid()){
                $planproduccion = new PlanProduccion();
                $a = $form->getValue('fecha_creacion');
                $planproduccion->setFechaCreacion($a);
                $a = $form->getValue('fecha_inicio');
                $planproduccion->setFechaInicio($a);
                $a = $form->getValue('id_usuario');
                $planproduccion->setIdUsuario($a);
                $planproduccion->save();
                $this->redirect('planproduccion/index');} }
protected function processFormmodificar(sfWebRequest $request, sfForm $form){
        $form->bind($request->getParameter($form->getName()), $request->getFiles($form-
>getName()));
        if ($form->isValid()){
                $planproduccion = PlanProduccionPeer::retrieveByPK($request-
>getParameter('id'));

```

```

        $a = $form->getValue('fecha_creacion');
        $planproduccion->setFechaCreacion($a);
        $a = $form->getValue('fecha_inicio');
        $planproduccion->setFechaInicio($a);
        $a = $form->getValue('id_usuario');
        $planproduccion->setIdUsuario($a);
        $planproduccion->save();
        $this->redirect('planproduccion/index');} }

public function executeBuscar(){ }
public function executeMostrar($request) {
    $si = 0;
    $c=new Criteria();
    if($request->getParameter('fecha_creacion')!=""){
        $c->add(PlanProduccionPeer::FECHA_CREACION,$request-
>getParameter('fecha_creacion'));
        $si = 1;}
    if($request->getParameter('fecha_inicio')!=""){
        $c->add(PlanProduccionPeer::FECHA_INICIO,$request->getParameter('fecha_inicio'));
        $si = 1;}
    if($request->getParameter('id_usuario')!=0){
        $c->add(PlanProduccionPeer::ID_USUARIO,$request->getParameter('id_usuario'));
        $si = 1;}
    $this->datos=PlanProduccionPeer::doSelect($c);
    if($si == 1){
    if(count($this->datos)>0) {
        $this->esta = 1;}
    else {
        $this->esta = 0;}}
    else{
        $this->esta = 0;} }
public function executeImprimir() {
    $con = $this->getUser()->getAttribute('imprimir');
    $pdf=new YPDF();
    //Títulos de las columnas
    $header=array('Id',utf8_decode('Fecha Creación'),'Fecha Inicio','Usuario');
    //Carga de datos
    $data[][]=array();
    $i=0;
    foreach($con as $c){
        $data[$i][0] = utf8_decode($c->getId());
        $data[$i][1] = utf8_decode($c->getFechaCreacion());
        $data[$i][2] = utf8_decode($c->getFechaInicio());
    }
}

```

```
$data[$i][3] = utf8_decode(UsuarioPeer::retrieveByPK($c->getIdUsuario())->getUsuario());  
$i++;}  
$pdf->SetFont('Arial', "", 14);  
$pdf->AddPage();  
$pdf->Image('/home/yusbel/workspace/proyecto/web/images/logo.jpg', 10, 0, 30);  
$pdf->SetX(50);  
$pdf->Cell(6, 0, 'Reporte del sistema del ' . date('d/m/Y h:i:s A') . ', 0, 0, "", false, "");  
$pdf->FancyTable($header, $data);  
$pdf->Output(); }  
public function executeMas(sfWebRequest $request){  
    $c=new Criteria();  
    $c->add(ActividadPeer::ID_PLAN,$request->getParameter('id'));  
    $this->datos=ActividadPeer::doSelect($c);} }
```

### 2.6 Diagrama de despliegue.

En el siguiente Modelo de Despliegue se verán los elementos de configuración del procesamiento así como las conexiones entre ellos; además, se detalla la distribución física del sistema o sea, cómo se distribuye la funcionalidad entre los nodos (PCs) mostrando la configuración del funcionamiento del sistema, se puede afirmar que el diagrama de despliegue muestra la disposición física de los distintos nodos que se encuentran en la composición de un sistema y la distribución de los programas ejecutables sobre estos nodos.

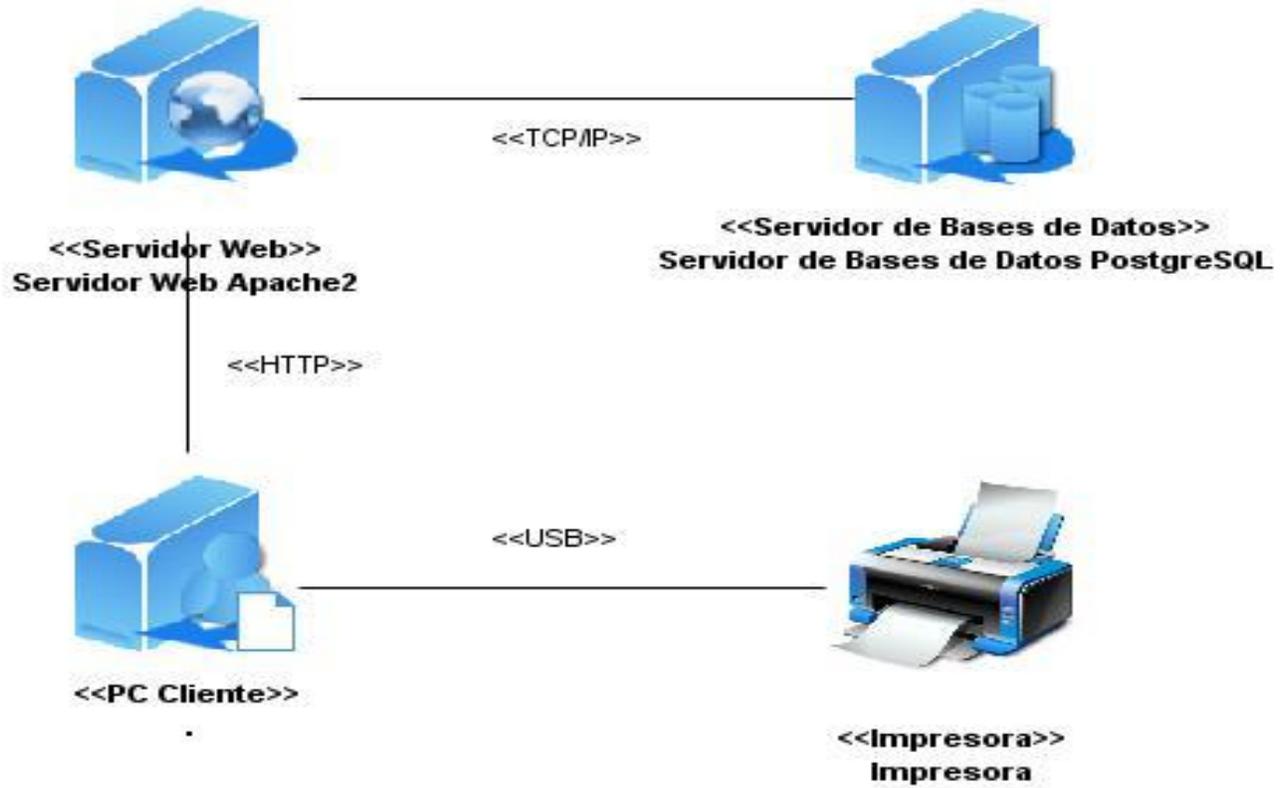


Figura: 20 Diagrama de Despliegue.

## 2.7 Conclusiones.

El capítulo describe las características que desde el punto de vista de funcionalidad debe cumplir el sistema. En el mismo se detalla parte de la arquitectura del sistema mediante los diferentes diagramas que conforman la vista de implementación. En él se define el estándar de codificación para la

implementación del sistema, así como se especifica el código PHP de las principales funcionalidades desarrolladas.

Además se enuncian elementos como un servidor de aplicación y otro de base de datos, los ordenadores clientes y por ultimo una impresora que describen los elementos arquitectónicos que guían la implantación de la solución propuesta, garantizando su correcto funcionamiento.

## CAPÍTULO III. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

---

### 3.1 Introducción.

Uno de los mayores retos que afronta la informática en toda su magnitud es la calidad del software, por lo que el proceso correspondiente a las pruebas es indiscutiblemente uno de los aspectos a tener en cuenta para evaluar el estado de la calidad en un sistema informático.

La elaboración de un software con calidad, involucra la utilización de metodologías o procedimientos estándares para todas las fases de desarrollo del software, ya sea el análisis, diseño, implementación y prueba. Logrando que estas permitan llevar sobre un mismo camino la filosofía de trabajo, en aras de obtener una mayor confianza y facilidad de prueba, en el momento que se enaltezca la productividad, tanto para el desarrollo como para el control de la calidad de la aplicación.

En el proceso de desarrollo del software, los riesgos de errores son incalculables. Estos pueden estar dados por una mala especificación de los requisitos funcionales, mal uso de las estructuras de datos, errores al enlazar módulos, entre otros. Las pruebas para la validación de los resultados no es un proceso que deba realizarse una vez desarrollado el software, sino que debe ejecutarse en cada una de las fases del desarrollo.

Un objetivo primordial a la hora de medir la eficacia de las pruebas, es la determinación de cuando se han realizado suficientes pruebas. Si cada vez que se procesa el software persisten errores, se hace necesaria la continuidad de las pruebas. En caso de producirse modificaciones en el software, entonces se tendrá que probar nuevamente las partes afectadas por las modificaciones. Por lo anteriormente dicho, se desarrolló este capítulo, con el objetivo de especificar las pruebas que serán hechas al software en su etapa de implementación y describir los resultados de las mismas.

## 3.2 Casos de pruebas.

Los casos de pruebas deben demostrar que el producto desarrollado, satisface realmente las peticiones del cliente, tal y como se describe en la especificación de los requerimientos y los casos de uso. Además es válido destacar la necesidad de elaborar un buen caso de prueba, porque de esta forma se detectarán errores posibles de la aplicación, permitiendo así que el software llegue libre de defectos a las manos del cliente.

### 3.2.1 Caso de prueba Autenticar.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos usuario y contraseña y presiona el botón entrar.	El sistema da al usuario acceso al sistema dándole acceso según el rol que este posea.	El usuario que está accediendo existe en la base de datos y su usuario y contraseña son válidos para el dominio UCI.
El usuario introduce un usuario que no es usuario del sistema y presiona el botón entrar.	El sistema muestra un mensaje indicándole al usuario que ese nombre de usuario no tiene acceso al sistema.	El usuario con el cual el usuario está tratando de entrar al sistema no es usuario del mismo.
El usuario entra un usuario del sistema y una contraseña que no es la del dominio UCI y oprime el botón entrar.	El sistema muestra un mensaje indicándole al usuario que ese nombre de usuario y esa contraseña no coinciden para el dominio UCI.	El usuario entra un usuario del sistema y una contraseña que no es la del dominio UCI.

### 3.2.2 Caso de prueba Gestionar Usuario.

#### 3.2.2.1 Adicionar usuario.

Tabla 8: Caso de prueba adicionar usuario.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos del usuario y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de usuarios.	El usuario que se está tratando de agregar no existe en la base de datos.
El usuario introduce un usuario introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que ese nombre de usuario ya existe.	El usuario que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra el usuario, la contraseña y no selecciona el rol del usuario y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra el usuario o la contraseña o no selecciona el rol del usuario.
El usuario se encuentra en la interfaz de adicionar usuario, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

#### 3.2.2.2 Modificar usuario.

Tabla 9: Caso de prueba modificar usuario.

Entrada	Resultado	Condiciones
El usuario modifica los datos del usuario seleccionado y presiona guardar.	El sistema muestra en el listado de los usuarios actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y

		debe de haber introducidos los datos correctamente.
El usuario no modifica los datos del usuario seleccionado y presiona guardar.	El sistema no modifica los datos del usuario y muestra el listado de los usuarios.	El usuario no debe haber modificado los campos del usuario seleccionado.
El usuario se encuentra en la interfaz modificar usuario, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.2.3 Detalles del usuario

Tabla 10: Caso de prueba detalles de usuario.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos del usuario.	El sistema muestra una tabla con los datos del usuario.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.2.4 Eliminar Usuario.

Tabla 11: Caso de prueba eliminar usuario.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar un usuario.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación del usuario.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.3 Caso de prueba Gestionar Rol.

#### 3.2.3.1 Adicionar rol.

Tabla 12: Caso de prueba adicionar rol.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos del rol y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de roles.	El rol que se está tratando de agregar no existe en la base de datos.
El usuario introduce un rol introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que ese nombre de rol ya existe.	El rol que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra el nombre del rol y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra el nombre del rol.
El usuario se encuentra en la interfaz de adicionar rol, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

#### 3.2.3.2 Modificar rol.

Tabla 13: Caso de prueba modificar rol.

Entrada	Resultado	Condiciones
El usuario modifica los datos del rol seleccionado y presiona guardar.	El sistema muestra en el listado de los roles actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los

		datos correctamente.
El usuario no modifica los datos del rol seleccionado y presiona guardar.	El sistema no modifica los datos del rol y muestra el listado de los roles.	El usuario no debe haber modificado los campos del rol seleccionado.
El usuario se encuentra en la interfaz modificar rol, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.3.3 Detalles del rol.

Tabla 14: Caso de prueba detalles del rol.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos del rol.	El sistema muestra una tabla con los datos del rol.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.3.4 Eliminar rol.

Tabla 15: Caso de prueba eliminar rol.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar un rol.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación del rol.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.4 Caso de prueba Gestionar Media.

#### 3.2.4.1 Adicionar foto.

Tabla 16: Caso de prueba adicionar foto.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos de la foto y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de las fotos.	La foto que se está tratando de agregar no existe en la base de datos.
El usuario introduce un título de media introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que ese título de media ya existe.	El título de media que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra el título de la media, la fecha de producción, el código del DVD, el estado del DVD, el alto, el ancho, la temática, el nombre de la carpeta contenedora, no selecciona el formato y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra el título de la media o la fecha de producción o el código del DVD o el estado del DVD o el alto o el ancho o la temática o el nombre de la carpeta contenedora o no selecciona el formato de la imagen.
El usuario se encuentra en la interfaz de adicionar foto, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.4.2 Modificar foto.

Tabla 17: Caso de prueba modificar foto.

Entrada	Resultado	Condiciones
El usuario modifica los datos de la foto seleccionada y presiona guardar.	El sistema muestra en el listado de las fotos actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos de la foto seleccionada y presiona guardar.	El sistema no modifica los datos de la foto y muestra el listado de las fotos.	El usuario no debe haber modificado los campos de la foto seleccionado.
El usuario se encuentra en la interfaz modificar foto, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.4.3 Detalles de la foto.

Tabla 18: Caso de prueba detalles de la foto.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos de la foto.	El sistema muestra una tabla con los datos de la foto.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

**3.2.4.4 Eliminar foto.**

Tabla 19: Caso de prueba eliminar foto.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar una foto.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación de la foto.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

**3.2.5 Caso de prueba Gestionar Solicitud de Media.**

**3.2.5.1 Adicionar solicitud de media.**

Tabla 20: Caso de prueba adicionar solicitud de media.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos de la solicitud de media y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de las solicitudes de media.	El usuario entra los datos correctamente.
El usuario no entra el nombre del solicitante, el área del solicitante, el motivo, la fecha de la solicitud, la fecha de la necesidad y no selecciona la media a solicitar y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra el nombre del solicitante o el área del solicitante o el motivo o la fecha de la solicitud o la fecha de la necesidad o no selecciona la media a solicitar.
El usuario se encuentra en la interfaz de adicionar solicitud de media, e introduce números en algunos campos de texto y	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

viceversa.		
------------	--	--

### 3.2.5.2 Modificar solicitud de media.

Tabla 21: Caso de prueba modificar solicitud de media.

Entrada	Resultado	Condiciones
El usuario modifica los datos de la solicitud de media seleccionada y presiona guardar.	El sistema muestra en el listado de las solicitudes de medias actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos de la solicitud de media seleccionada y presiona guardar.	El sistema no modifica los datos de la solicitud de media y muestra el listado de las solicitudes de medias.	El usuario no debe haber modificado los campos de la solicitud de media seleccionada.
El usuario se encuentra en la interfaz modificar solicitud de media, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.5.3 Detalles de una solicitud de media.

Tabla 22: Caso de prueba detalles de la solicitud de media.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos de la solicitud de media.	El sistema muestra una tabla con los datos de la solicitud de media.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.6 Caso de prueba Gestionar Equipamiento.

#### 3.2.6.1 Adicionar equipamiento.

Tabla 23: Caso de prueba adicionar equipamiento.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos del equipamiento y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de los equipamientos.	El número de serie que se está tratando de agregar no existe en la base de datos.
El usuario introduce un número de serie introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que ese número de serie ya existe.	El número de serie que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra el número de serie y no selecciona el tipo de equipamiento y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra el número de serie o no selecciona el tipo de equipamiento.
El usuario se encuentra en la interfaz de adicionar equipamiento, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

#### 3.2.6.2 Modificar equipamiento.

Tabla 24: Caso de prueba modificar equipamiento.

Entrada	Resultado	Condiciones
El usuario modifica los datos del	El sistema muestra en el listado	El usuario debe estar autenticado

equipamiento seleccionado y presiona guardar.	de los equipamientos actualizado.	en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos del equipamiento seleccionado y presiona guardar.	El sistema no modifica los datos del equipamiento y muestra el listado de los equipamientos.	El usuario no debe haber modificado los campos del equipamiento seleccionado.
El usuario se encuentra en la interfaz modificar equipamiento, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.6.3 Detalles del equipamiento.

Tabla 25: Caso de prueba detalles del equipamiento.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos del equipamiento.	El sistema muestra una tabla con los datos del equipamiento.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.6.4 Asignar equipamiento.

Tabla 26: Caso de prueba asignar equipamiento.

Entrada	Resultado	Condiciones
El usuario introduce selecciona el usuario al cual se le asignara el equipamiento y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de los equipamientos.	Seleccionar un usuario.
El usuario no selecciona el	El sistema muestra el listado de	El usuario no selecciona el

usuario y oprime el botón guardar.	los equipamientos.	usuario.
------------------------------------	--------------------	----------

### 3.2.6.5 Cambiar asignación del equipamiento.

Tabla 27: Caso de prueba cambiar asignación del equipamiento.

Entrada	Resultado	Condiciones
El usuario modifica la asignación del equipamiento seleccionado y presiona guardar.	El sistema muestra en el listado de los equipamientos actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica la asignación del equipamiento seleccionado y presiona guardar.	El sistema no modifica los datos del equipamiento y muestra el listado de los equipamientos.	El usuario no debe haber modificado los campos del equipamiento seleccionado.

### 3.2.6.6 Eliminar equipamiento.

Tabla 28: Caso de prueba eliminar equipamiento.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar un equipamiento.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación del equipamiento.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

## 3.2.7 Caso de prueba Gestionar Plan de Programación.

### 3.2.7.1 Adicionar plan de programación.

Tabla 29: Caso de prueba adicionar plan de programación.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos del plan de programación y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de los planes de programación.	La fecha de inicio que se está tratando de agregar no existe en la base de datos.
El usuario introduce un plan de programación introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que esa fecha de inicio ya existe.	La fecha de inicio que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra la fecha de creación, la fecha de inicio y no selecciona el usuario y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra la fecha de creación o la fecha de inicio o no selecciona el usuario.
El usuario se encuentra en la interfaz de adicionar plan de programación, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.7.2 Modificar plan de programación.

Tabla 30: Caso de prueba modificar plan de programación.

Entrada	Resultado	Condiciones
El usuario modifica los datos del plan de programación seleccionado y presiona guardar.	El sistema muestra en el listado de los planes de programación actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.

El usuario no modifica los datos del plan de programación seleccionado y presiona guardar.	El sistema no modifica los datos del plan de programación y muestra el listado de los planes de programación.	El usuario no debe haber modificado los campos del plan de programación seleccionado.
El usuario se encuentra en la interfaz modificar plan de programación, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.7.3 Detalles del plan de programación.

Tabla 31: Caso de prueba detalles del plan de programación.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos del plan de programación.	El sistema muestra una tabla con los datos del plan de programación.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.7.4 Adicionar programación a un plan de programación.

Tabla 32: Caso de prueba adicionar programación a un plan de programación.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos de la programación y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado del plan de programación al cual pertenece la programación añadida.	La programación que se está tratando de agregar no existe en la base de datos.
El usuario introduce una programación introducida	El sistema muestra un mensaje indicándole al usuario que esa	La programación que el usuario está tratando de agregar ya

anteriormente y presiona el botón guardar.	programación ya existe.	existe en la base de datos.
El usuario no entra la fecha, el canal, no selecciona la media y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra la fecha o el canal o no selecciona la media.
El usuario se encuentra en la interfaz de adicionar programación, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.7.5 Modificar programación de un plan de programación.

Tabla 33: Caso de prueba modificar programación de un plan de programación.

Entrada	Resultado	Condiciones
El usuario modifica los datos de la programación seleccionada y presiona guardar.	El sistema muestra el listado de las programaciones del plan asociado actualizado.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos de la programación seleccionada y presiona guardar.	El sistema no modifica los datos de la programación y muestra el listado de las programaciones del plan asociado.	El usuario no debe haber modificado los campos de la programación seleccionada.
El usuario se encuentra en la interfaz modificar programación, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.7.6 Detalles de programación de un plan de programación.

Tabla 34: Caso de prueba detalles de programación de un plan de programación.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos de la programación.	El sistema muestra una tabla con los datos de la programación.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.7.7 Eliminar programación de un plan de programación.

Tabla 35: Caso de prueba eliminar programación de un plan de programación.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar una programación.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación de la programación.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.7.8 Eliminar plan de programación.

Tabla 36: Caso de prueba eliminar plan de programación.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar un plan de programación.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación del plan de programación.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.8 Caso de prueba Gestionar Plan de Producción.

#### 3.2.8.1 Adicionar plan de producción.

Tabla 37: Caso de prueba adicionar plan de producción.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos del plan de producción y presiona el botón guardar.	El sistema muestra un mensaje de confirmación al usuario y el listado de los planes de producción.	El plan de producción que se está tratando de agregar no existe en la base de datos.
El usuario introduce un plan de producción introducido anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que ese plan de producción ya existe.	El plan de producción que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra la fecha de creación, la fecha de inicio, no selecciona el usuario y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra la fecha de creación o la fecha de inicio o no selecciona el usuario.
El usuario se encuentra en la interfaz de adicionar plan de producción, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

#### 3.2.8.2 Modificar plan de producción.

Tabla 38: Caso de prueba modificar plan de producción.

Entrada	Resultado	Condiciones
El usuario modifica los datos del	El sistema muestra en el listado	El usuario debe estar autenticado

plan de producción seleccionado y presiona guardar.	de los planes de producción actualizado.	en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos del plan de producción seleccionado y presiona guardar.	El sistema no modifica los datos del plan de producción y muestra el listado de los planes de producción.	El usuario no debe haber modificado los campos del plan de producción seleccionado.
El usuario se encuentra en la interfaz modificar plan de producción, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.8.3 Detalles del plan de producción.

Tabla 39: Caso de prueba detalles del plan de producción.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos del plan de producción.	El sistema muestra una tabla con los datos del plan de producción.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.8.4 Adicionar actividad a un plan de producción.

Tabla 40: Caso de prueba adicionar actividad de un plan de producción.

Entrada	Resultado	Condiciones
El usuario introduce correctamente los datos de la actividad y presiona el botón	El sistema muestra un mensaje de confirmación al usuario y el listado de actividades del plan	La actividad que se está tratando de agregar no existe en la base de datos.

guardar.	asociado.	
El usuario introduce una actividad introducida anteriormente y presiona el botón guardar.	El sistema muestra un mensaje indicándole al usuario que esa actividad ya existe.	La actividad que el usuario está tratando de agregar ya existe en la base de datos.
El usuario no entra la fecha, el lugar, la actividad, el transporte, no selecciona el área y oprime el botón guardar.	El sistema muestra un mensaje de error solicitando que los campos obligatorios no pueden estar vacíos.	El usuario no entra la fecha o el lugar o la actividad o el transporte o no selecciona el área.
El usuario se encuentra en la interfaz de adicionar actividad, e introduce números en algunos campos de texto y viceversa.	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

### 3.2.8.5 Modificar actividad de un plan de producción.

Tabla 41: Caso de prueba modificar actividad de un plan de producción.

Entrada	Resultado	Condiciones
El usuario modifica los datos de la actividad seleccionada y presiona guardar.	El sistema muestra en el listado de las actividades asociadas al plan actualizadas.	El usuario debe estar autenticado en el sistema, tener las credenciales correspondientes y debe de haber introducidos los datos correctamente.
El usuario no modifica los datos de la actividad seleccionada y presiona guardar.	El sistema no modifica los datos de la actividad y muestra el listado de las actividades asociadas al plan.	El usuario no debe haber modificado los campos de la actividad seleccionada.
El usuario se encuentra en la interfaz modificar actividad, e introduce números en algunos	El sistema muestra un mensaje de error (caracteres inválidos).	El usuario introduce caracteres no válidos al formulario.

campos de texto y viceversa.		
------------------------------	--	--

### 3.2.8.6 Detalles de actividad de un plan de producción.

Tabla 42: Caso de prueba detalles de una actividad de un plan de producción.

Entrada	Resultado	Condiciones
El usuario presiona sobre la imagen que representa ver los datos de la actividad.	El sistema muestra una tabla con los datos de la actividad.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.8.7 Eliminar actividad de un plan de producción.

Tabla 43: Caso de prueba eliminar actividad de un plan de producción.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar una actividad.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación de la actividad.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### 3.2.8.8 Eliminar plan de producción.

Tabla 44: Caso de prueba eliminar plan de producción.

Entrada	Resultado	Condiciones
El usuario selecciona la imagen que representa eliminar un plan de producción.	El sistema muestra un mensaje de confirmación para que el usuario confirme o cancele la eliminación del plan de producción.	El usuario debe estar autenticado en el sistema y poseer las credenciales correspondientes.

### **3.3 Conclusiones.**

Durante el desarrollo de este capítulo se comprobó la validez del sistema desarrollado como solución, mediante la realización de pruebas de unidad.

Se certificó el cumplimiento de los requerimientos funcionales solicitados por el cliente.

Probando de esta manera el cumplimiento del objetivo general de la investigación, de implementar un sistema de gestión de procesos para la DTU de la UCI.

## CONCLUSIONES

---

Los sistemas de gestión de información brindan salidas eficaces al procesamiento de flujos de datos. Durante el desarrollo de este trabajo ha sido su objetivo general la implementación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, el cual ha sido cumplido a cabalidad. El resultado ha sido la aplicación web Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, el cuál será de gran utilidad a la DTU debido a que con la utilización de este se podrán manipular de una forma más fluida y organizada los siempre crecientes flujos de información que diariamente se gestionan en dicha dirección.

Para su implementación se utilizaron herramientas libres, como el framework Symfony, el cuál implementa la arquitectura modelo-vista-controlador así como las mejores prácticas de patrones arquitectónicos y de la programación, como lenguaje de programación PHP5 debido a sus características. Además de poseer gran popularidad en lo que a lenguajes de programación web se refiere. El servidor Web Apache2 para el soporte de la aplicación. Todo ello le proporciona a la aplicación la posibilidad de ser desplegada en diferentes sistemas operativos.

Con la implantación de este sistema se pretende: organizar de manera coherente y secuencial los flujos de trabajos dentro de la DTU para lograr así una mayor eficiencia los servicios que ofrece dicha entidad. Eliminar el trabajo con los documentos tradicionales (papeles) posibilitará que la comunicación entre los trabajadores de la dirección sea más rápida, garantizando con esto un ahorro de materiales de oficina, así como una mejor gestión de la información que se manipula.

Durante la implementación del sistema, se siguieron los principios descritos para su desarrollo, esperando que el mismo logre una buena protección de los datos, cumpliendo así con el vital principio de seguridad informática asegurando la integridad de los datos en todo momento.

## RECOMENDACIONES

---

Se recomienda:

- Continuar el estudio con el objetivo de añadir nuevas funcionalidades al sistema, para mejorar los servicios existentes en la dirección y otros que permitan gestionar los nuevos servicios que la dirección pueda comenzar a brindar.
- Proponer la implementación de sistemas similares en las diferentes televisoras de nuestro país, adaptándolos a las características y flujos de trabajo de cada una de ellas.

## REFERENCIAS BIBLIOGRÁFICAS

---

1. **Chuck Musciano, Bill Kenedy.** *HTML. La Guía Completa* [En Línea] McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V. 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg01313.pdf> ].
2. **Duque Méndez, N. D.** *Conceptos de arquitectura Cliente/Servidor* [En Línea] 2009-01-23 [Disponible en: [http://www.it.uc3m.es/mcfp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcfp/docencia/si/material/1_cli-ser_mcfp.pdf) ].
3. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.* 2009-01-23. [Disponible en: <http://www.librosweb.es/symfony>].
4. **Gil, R.** *Sistemas de Información.* [En Línea] 2009-01-23. [Disponible en: [http://www.it.uc3m.es/mcfp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcfp/docencia/si/material/1_cli-ser_mcfp.pdf) ].
5. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software* [En Línea]. En español. Madrid: Addison Wesley, Pearson Educación. S.A, 2000. pág. 464. Vol. I. 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>].
6. **Junta de Castilla y León.** *Glosario* 2009-01-21[Disponible en: <http://www.eclap.es/contenidos/calidad/GLOSARIO/GLOSARIO.doc>].
7. **Josefa Risoto.** *Gestión de la documentación audiovisual en Televisión Valenciana* [En Línea]. ISSN 1695-5498. 2009-01-23. [Disponible en: <http://www.hipertext.net/web/pag211.htm> ].
8. **KToon Documentation.** *Estándar de Codificación.*2009-04-17 .[Disponible en: [http://ktoon.toonka.com/documentation/index.php?title=Manual:Est%E1ndar\\_de\\_Codificaci%F3n](http://ktoon.toonka.com/documentation/index.php?title=Manual:Est%E1ndar_de_Codificaci%F3n) ].
9. **Kabir, Mohammed.** *La Biblia del Servidor Apache 2.0.* [En línea] 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg01737.pdf> ].

10. **Larman, Craig.** *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos.* [En Línea] La Habana: Félix Varela, 2004. Vol. I. 2009-01-23[Disponible en: <http://bibliodoc.uci.cu/pdf/reg00061.pdf> ].
11. **Mehdi Achour, Friedhelm Betz, Antony Dovgal.** *PHP.* [En línea] Grupo de documentación de PHP, 2009-01-23. [Disponible en: <http://www.php.net/manual/es/index.php> ].
12. **Valdés, D. P.** *Los diferentes lenguajes de programación para la Web.* [En Línea] 2009-01-23 [Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/> ].

## BIBLIOGRAFÍA

---

1. **Blog de Programación.** *Un framework para php symfony*. [Disponible en: [http://www.blogdeprogramacion.blogspot.com/2007/10/un\\_framework\\_para\\_php\\_symfony.html](http://www.blogdeprogramacion.blogspot.com/2007/10/un_framework_para_php_symfony.html) ].
2. **Chuck Musciano, Bill Kenedy.** *HTML. La Guía Completa* [En Línea] McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V. 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg01313.pdf> ].
3. **Departamento de Ingeniería de software UCI.** *Flujo de Implementación*. La Habana : s.n., 2006-2007.
4. **Duque Méndez, N. D.** *Conceptos de arquitectura Cliente/Servidor* [En Línea] 2009-01-23 [Disponible en: [http://www.it.uc3m.es/mcfp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcfp/docencia/si/material/1_cli-ser_mcfp.pdf) ].
5. **Enrique Hernández Orallo.** DISCA Lenguaje Unificado de Modelado (UML). [Disponible en: <http://www.disca.upv.es> ].
6. **Espina, Eduardo.** *Seguridad en la Web*. [Disponible en: <http://www.espina.info/papers/seguridadenlaweb.pdf> ].
7. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva*. 2009-01-23. [Disponible en: <http://www.librosweb.es/symfony>].
8. **Factoría de Internet S.L.** *Validar los datos de los formularios*. [Disponible en: <http://www.webtaller.com/manual-asp/validar-datos-formularios.php> ].
9. **Foundation, The Eclipse.** *PDT Project*.2009-03-22. [Disponible en: <http://www.eclipse.org/pdt/>].
10. **Gil, R.** *Sistemas de Información*. [En Línea] 2009-01-23. [Disponible en: [http://www.it.uc3m.es/mcfp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcfp/docencia/si/material/1_cli-ser_mcfp.pdf) ].

11. **Grupo\_Desarrollo\_Symfony**. Proyecto ASKEET. *Proyecto ASKEET*. [Disponible en: <http://trac.symfony-project.com/browser/doc/branches/1.0/askeet/es>].
12. **Hernández Meléndrez, E.** *Cómo escribir una tesis*. 2006.
13. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El Proceso Unificado de Desarrollo de Software* [En Línea]. En español. Madrid: Addison Wesley, Pearson Educación. S.A, 2000. pág. 464. Vol. I. 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>].
14. **Josefa Risoto**. *Gestión de la documentación audiovisual en Televisión Valenciana* [En Línea]. ISSN 1695-5498. 2009-01-23. [Disponible en: <http://www.hipertext.net/web/pag211.htm> ].
15. **Junta de Castilla y León**. *Glosario* 2009-01-21[Disponible en: <http://www.eclap.es/contenidos/calidad/GLOSARIO/GLOSARIO.doc>].
16. **Kabir, Mohammed**. *La Biblia del Servidor Apache 2.0*. [En línea] 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg01737.pdf> ].
17. **KToon Documentation**. *Estándar de Codificación*.2009-04-17 .[Disponible en: [http://ktoon.toonka.com/documentation/index.php?title=Manual:Est%E1ndar\\_de\\_Codificaci%F3n](http://ktoon.toonka.com/documentation/index.php?title=Manual:Est%E1ndar_de_Codificaci%F3n) ].
18. **Larman, Craig**. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. [En Línea] La Habana: Félix Varela, 2004. Vol. I. 2009-01-23. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00061.pdf> ].
19. **Ltd, Zend Technologies**. *Eclipse PHP Development Tools*. [Disponible en: <http://www.zend.com/en/community/pdt>]
20. **Maestros del Web**. *Los diferentes lenguajes de programación para la web*. [Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>].

21. **MATEU, C.** *Desarrollo de aplicaciones Web*. vol. Primera edición [Disponible en: [http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/desarrollo-de-aplicaciones-web/desarrollo-de-aplicaciones-web/XP06\\_M2108\\_01497.pdf](http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/desarrollo-de-aplicaciones-web/desarrollo-de-aplicaciones-web/XP06_M2108_01497.pdf) ]
22. **Mehdi Achour, Friedhelm Betz, Antony Dovgal.** *PHP*. [En línea] Grupo de documentación de PHP, 2009-01-23. [Disponible en: <http://www.php.net/manual/es/index.php> ].
23. **ProgramacionWeb.net.** *MVC - Modelo Vista Controlador*. [Disponible en: <http://www.programacionweb.net/articulos/articulo/?num=505> ].
24. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas*. s.l. : Prentice Hall.
25. **Stallings, William.** *Web Security*. [Disponible en: <http://www.microsoft.com/technet/archive/security/prodtech/windows/iis/chaptr14.mspx?mfr=true> ].
26. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, y Jouni Ahto.** *Manual de PHP*. 2001.
27. *Symfony en pocas palabras* [Disponible en : [http://www.librosweb.es/symfony/capitulo1/symfony\\_en\\_pocas\\_palabras.html](http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html) ]
28. **Universidad de Valencia.** *Introducción a RUP*. [Disponible en: <http://www.pid.dsic.upv.es/C1/Materiales/Documentos%20Disponibles/Introducci%B3n%20d%20RUP.doc> ]
29. **Valdés, D. P.** *Los diferentes lenguajes de programación para la Web*. [En Línea] 2009-01-23 [Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/> ].

## GLOSARIO DE TÉRMINOS.

---

**Aplicación:** Es un tipo de programa informático o software que está diseñado para facilitar al usuario algún tipo de trabajo, lo cual lo diferencia de otros tipos de programas como los sistemas operativos, lenguajes de programación o utilidades.

**Aplicación Web:** Aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

**Caso de uso:** En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

**E-mail:** Es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente (también denominados mensajes electrónicos o cartas electrónicas) mediante sistemas de comunicación electrónicos.

**Framework:** En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

**FTP:** Protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

**Gopher:** Servicio de Internet consistente en el acceso a la información a través de menús.

**Helper:** Herramientas que proporcionan una manera más rápida de crear controles en las plantillas.

**Hipertexto:** Nombre que recibe el texto que en la pantalla de una computadora te conduce a otro texto relacionado.

**Implementación:** La fase de implementación es conocida también como fase de codificación, pues supone todo el proceso de escribir el código software necesario que hará posible que el sistema finalmente implementado cumpla con las especificaciones.

**Internet Explorer:** Navegador Web producido por Microsoft para el sistema operativo Windows.

**Layout:** Término inglés, se le denomina así a la plantilla que contiene el código HTML común a todas las páginas o a una parte de ellas.

**Mixin:** Clase que ofrece cierta funcionalidad para ser heredada por una subclase, pero no está ideada para ser autónoma.

**Motores de búsqueda:** Sistema informático que indexa archivos almacenados en servidores web gracias a su «*spider*» (o Web crawler).

**Mozilla Firefox:** Navegador de Internet, libre y de código abierto descendiente de Mozilla Application Suite, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos.

**Outlook:** Programa de agenda ofimática y cliente de e-mail de Microsoft, y forma parte de la suite Microsoft Office.

**Peer:** Clases objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados.

**Plug-in:** Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

**Script:** Programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script.

**Social Networking:** Describe todas aquellas herramientas diseñadas para la creación de espacios que promuevan o faciliten la conformación de comunidades e instancias de intercambio social.

**Software libre:** Software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido. En inglés free software.

**Subversion:** Software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos.

**Word:** Software destinado al procesamiento de textos.