

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 9**



**Diseño de un Sistema de Gestión de Procesos para la  
Dirección de Televisión Universitaria.**

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

**AUTORES:** Belayne Alfonso Quesada  
Yusney Marrero García

**TUTOR:** Ing. Yunier Albrecht Delgado

**Ciudad de la Habana, julio 2 de 2008  
“Año 50 de la Revolución”**

*“Lo que puedes hacer, o has soñado que puedes hacer, debes comenzar. La osadía lleva en sí, genio, poder y magia”*

*Johann Wolfgang von Goethe*

## DEDICATORIA

*De Belayne*

*A mi familia: por preocuparse y apoyarme en todo lo que necesito, por confiar y creer en mí.  
Especial a mi mamá: por ser mi razón de ser, por estar a mi lado en todos los momentos buenos y malos de mi vida, por saber comprenderme y aconsejarme cuando creía que iba por mal camino.*

*A mi papá: por estar pendiente de mí, por quererme y apoyarme como lo hace y darme la confianza y la fuerza suficiente para enfrentar la vida por muy dura que esta sea.  
A mi hermana que es mi alma gemela por siempre tenerme presente, por quererme como lo hace, por entenderme cuando nadie lo hace, por apoyarme aunque sea difícil incluso para ella, por aconsejarme cuando lo necesité y por regalarme todo ese amor que emana de su interior.*

*A mi sobrinito que es la ilusión y la alegría de cada uno de mis días.*

*De Yusney*

*A mi familia por haberme apoyado en todo momento, por todas las fiestas, las alegrías que pasamos juntos y por ayudarme siempre en lo que he necesitado.*

*A mi abuelita querida, por ser mi segunda madre, por haberme entregado tanto amor y cariño durante toda mi vida y por haber sido para mí fuente de inspiración y haberme dado tanta fuerza para continuar en todo momento, estés donde estés sé que estarás muy orgullosa de mí.*

*A mi adorada madre por ese infinito amor que siempre me ha brindado, por apoyarme y guiarme en todos los momentos de mi vida y ser mi razón de ser.*

*A mi papá por ser tan bueno y por vivir orgulloso de mí, por ayudarme cada día a ser una mejor persona y llegar hasta donde he llegado hoy.*

*A mi hermana por estar ahí siempre que la he necesitado, porque sé que me quieres mucho.*

## AGRADECIMIENTOS

*A la Universidad de las Ciencias Informáticas por recibirnos y ayudarnos a formarnos como futuros profesionales, a nuestro tutor, el Ing. Yunier Albrecht por aconsejarnos y ayudarnos en todo momento, al profesor Ing. Ramsés Ibarrola por sus recomendaciones siempre constructivas. A nuestras amistades por brindarnos su amistad y su ayuda, a nuestros compañeros de aula por compartir juntos tantos momentos buenos y malos en estos cinco años que indudablemente serán inolvidables para nosotros y a todos los que de una forma u otra nos han ayudado en la realización de este trabajo.*

*De Belayne*

*A mis padres, mi hermana y mi cuñado por ayudarme a realizar este sueño que también es de ellos, por el cariño y el amor con que siempre han sabido llenar mi alma.*

*A Barbarita que es como mi segunda madre por ayudarme en todo, por preocuparse siempre por mi futuro por hacerme sentir parte de su familia y darme tanto cariño desde pequeña.*

*A mi vecina FaFi y Maite por apoyarme y siempre estar ahí en los momentos mas importantes de mi familia, por aconsejarme cuando les parecía que algo estaba mal.*

*De Yusney*

*A toda mi familia por apoyarme y confiar en mí en todo momento, especialmente a mis padres, mi hermana y mi abuela por estar ahí siempre y no fallarme nunca.*

*A mis tíos y primos en fin a todos los motas por ayudarme de una forma u otra en todo lo que han podido y por preocuparse tanto por mí.*

*A Haydee por ayudarme siempre en todo lo que he necesitado.*

## DECLARACIÓN DE AUTORÍA

Belayne Alfonso Quesada y Yusney Marrero García declaramos que somos los únicos autores de este trabajo y autorizamos 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 firmamos la presente a los 2 días del mes de julio del año 2008.

---

Belayne Alfonso Quesada

---

Yusney Marrero García

---

Tutor: Ing. Yunier Albrecht Delgado.

## DATOS DE CONTACTO

Tutor: Ing. Yunier Albrecht Delgado.

Ingeniero en Ciencias Informáticas, 2007. Diploma de Oro. Profesor del Departamento de Técnicas de Programación, Facultad 9. Líder del Proyecto UCITeVe, Polo Productivo “Video y Sonido Digital”.

Dirección: Universidad de las Ciencias Informáticas, Edificio32 Apto 103

Teléfono Laboratorio: +53 -7 – 8358284 Teléfono Apto: +53 -7 – 8358902

E –mail: [yalbrecht@uci.cu](mailto:yalbrecht@uci.cu)

## RESUMEN

En los primeros años de la Universidad de las Ciencias Informáticas (UCI) debido a la convivencia de la gran cantidad de alumnos y profesores surgió la necesidad de contar con un medio que ayudara a difundir todo tipo de información a la gran comunidad universitaria, un medio que apoyara en cierta medida la docencia y que además contribuyera a la recreación del personal que reside en la universidad.

A raíz de esta necesidad se creó la Dirección de Televisión Universitaria (DTU) en la cual existe actualmente un gran flujo de información que genera diversos procesos que se realizan de forma manual enviando las orientaciones y actividades de la semana vía correo, guardando en agendas personales información que es constantemente consultada por lo que todo se hace con dificultad y provoca la introducción de errores por parte del hombre que pueden ser de pequeña o gran dimensión. Los problemas anteriormente expuestos necesitan una solución que facilite el trabajo de todos y a la vez satisfaga las necesidades de la comunidad universitaria. Por eso se nos plantea realizar el Diseño de un Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria. Este proyecto pudiera servir como base a otros de su tipo en cualquier productora de televisión tanto nacional como extranjera aunque se diferencien en algunos procesos en específicos. El objetivo de este trabajo diseñar un sistema de gestión de procesos para la DTU, que permita administrar los flujos de trabajo dentro de la estructura de la institución y que cubra todos los requerimientos necesarios para dar solución a los mismos.

## PALABRAS CLAVES

Sistema de Gestión de Procesos

## INDICE DE TABLAS Y FIGURAS

TABLA 1 DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO.....	41
TABLA 2 DESCRIPCIÓN DE LOS TRABAJADORES DEL NEGOCIO.....	43
TABLA 3 DESCRIPCIÓN TEXTUAL DEL CASO DE USO “PRODUCIR MEDIA”.....	45
TABLA 4 DESCRIPCIÓN TEXTUAL DEL CASO DE USO “ENTREGAR MEDIAS”.....	47
TABLA 5 DESCRIPCIÓN TEXTUAL DEL CASO DE USO “TRASMITIR PROGRAMACIÓN”.....	48
TABLA 6 DESCRIPCIÓN TEXTUAL DEL CASO DE USO “REALIZAR PLAN DE PROGRAMACIÓN”.....	49
TABLA 7 DESCRIPCIÓN TEXTUAL DEL CASO DE USO “REALIZAR PLAN DE PRODUCCIÓN”.....	49
TABLA 8 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	54
TABLA 9 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “AUTENTICAR USUARIOS”.....	59
TABLA 10 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR USUARIO”.....	62
TABLA 11 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR PLAN DE PRODUCCIÓN”.....	64
TABLA 12 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR REPORTE DE TRANSMISIÓN”.....	67
TABLA 13 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR MEDIA”.....	70
TABLA 14 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR PLAN DE PROGRAMACIÓN”.....	73
TABLA 15 DESCRIPCIÓN DETALLADA DEL CASO DE USO DEL SISTEMA “GESTIONAR DATOS DE EQUIPAMIENTO”.....	75
FIG. 1 ARQUITECTURA MODELO-VISTA-CONTROLADOR.....	15
FIG. 2 ARQUITECTURA POSTGRESQL.....	29
FIG. 3 FASES E HITOS DE UN PROYECTO (21).....	32
FIG. 4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	44
FIG. 5 PAQUETES DEL MODELO DE CASOS DE USOS DEL SISTEMA.....	54
FIG. 6 CASO DE USO ASOCIADO AL PAQUETE DE SEGURIDAD DEL SISTEMA.....	55
FIG. 7 CASOS DE USO ASOCIADOS AL PAQUETE ADMINISTRADOR DEL SISTEMA.....	55
FIG. 8 CASOS DE USO ASOCIADOS AL PAQUETE INFORMÁTICO DEL SISTEMA.....	56
FIG. 9 CASOS DE USO ASOCIADOS AL PAQUETE JEFE DPTO. PROGRAMACIÓN DEL SISTEMA.....	56
FIG. 10 CASOS DE USO ASOCIADOS AL PAQUETE TÉCNICO DE SISTEMA DEL SISTEMA.....	57
FIG. 11 CASO DE USO ASOCIADO AL PAQUETE ESPECIALISTA DEL SISTEMA.....	58
FIG. 12 CASO DE USO ASOCIADO AL PAQUETE PRODUCTOR DEL SISTEMA.....	58
FIG. 13 CASO DE USO ASOCIADO AL PAQUETE TRANSMISOR DEL SISTEMA.....	58
FIG. 14 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CASO DE USO “AUTENTICAR USUARIO”.....	78
FIG. 15 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CASO DE USO “GESTIONAR USUARIO”.....	78
FIG. 16 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CASO DE USO “MOSTRAR REPORTE DE TRANSMISIÓN”.....	78
FIG. 17 DIAGRAMA DE COLABORACIÓN “AUTENTICAR USUARIO”.....	79
FIG. 18 DIAGRAMA DE COLABORACIÓN “ADICIONAR PLAN DE PRODUCCIÓN”.....	79
FIG. 19 DIAGRAMA DE COLABORACIÓN “MODIFICAR REPORTE DE EQUIPAMIENTO ENTREGADO”.....	80
FIG. 20 DIAGRAMA DE COLABORACIÓN “MOSTRAR REPORTE DE TRANSMISIÓN”.....	80
FIG. 21 DIAGRAMA DE COLABORACIÓN “ELIMINAR PLAN DE PRODUCCIÓN”.....	81
FIG. 22 ESTRUCTURA DE LAS CLASES GENERADAS POR SYMFONY.....	83
FIG. 23 DIAGRAMA DE CLASES DEL DISEÑO CUS-AUTENTICAR USUARIO.....	85
FIG. 24 DIAGRAMA DE CLASES DEL DISEÑO CUS-GESTIONAR DATOS DE MEDIA.....	86
FIG. 25 DIAGRAMA DE CLASES DEL DISEÑO CUS-MOSTRAR REPORTE DE MATERIAL AUDIOVISUAL.....	87
FIG. 26 DIAGRAMA DE CLASES PERSISTENTES.....	90
FIG. 27 MODELO DE DATOS.....	91
FIG. 28 MODELO DE DESPLIEGUE.....	99



# INDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1</b> .....	<b>5</b>
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema.....	5
1.3 Objeto de Estudio.....	7
1.3.1 Descripción General.....	7
1.3.2 Descripción actual del dominio del problema.....	7
1.3.3 Situación Problemática.....	8
1.4 Análisis de otras soluciones existentes.....	9
1.5 Conclusiones.....	9
<b>CAPÍTULO 2</b> .....	<b>10</b>
2.1 Introducción.....	10
2.2 Las Aplicaciones Web.....	10
2.3 Modelo Cliente Servidor.....	12
2.4 Lenguajes de Programación Web.....	19
2.4.1 PHP.....	21
2.5 Servidores Web.....	23
2.5.1 Apache.....	24
2.6 Sistema de Gestión de Base de Datos.....	25
2.6.1 Postgre SQL.....	27
2.8 Lenguaje Unificado de Modelado (UML).....	33
2.9 Conclusiones.....	36
<b>CAPÍTULO 3</b> .....	<b>37</b>
3.1 Introducción.....	37
3.2 Modelo de Negocio.....	37
3.2.1 Procesos del Negocio.....	37
3.2.2 Reglas del Negocio.....	39
3.2.3 Actores del Negocio.....	40
3.2.4 Trabajadores del Negocio.....	41
3.2.5 Diagrama de Casos de Uso del Negocio.....	44
3.2.6 Descripción textual de los casos de Uso del Negocio.....	44
3.2.6.1 Caso de Uso “Producir Media”.....	44
3.2.6.2 Caso de Uso “Entregar Medias”.....	46
3.2.6.3 Caso de Uso “Trasmitir Programación”.....	47
3.2.6.4 Caso de Uso “Realizar Plan de Programación.”.....	48
3.2.6.5 Caso de Uso “Realizar Plan de Producción.”.....	49

3.3	Requerimientos Funcionales.....	49
3.4	Requerimientos No Funcionales.....	51
3.5	Descripción del Sistema Propuesto.....	53
3.5.1	Descripción de los actores del sistema.....	53
3.5.2.1	Paquete Seguridad.....	55
3.5.2.2	Paquete Administrador.....	55
3.5.2.3	Paquete Informático.....	56
3.5.2.4	Paquete Jefe Dpto. Programación.....	56
3.5.2.5	Paquete Técnico de Sistema.....	57
3.5.2.6	Paquete Especialista.....	58
3.5.2.7	Paquete Productor.....	58
3.5.2.8	Paquete Transmisor.....	58
3.5.3	Descripción Textual de los Casos de Uso del Sistema.....	59
3.5.3.1	Caso de Uso “Autenticar Usuarios”.....	59
3.5.3.2	Caso de Uso “Gestionar Usuarios”.....	60
3.5.3.3	Caso de Uso “Gestionar Plan de Producción”.....	62
3.5.3.4	Caso de Uso “Gestionar Reporte de Transmisión”.....	65
3.5.3.5	Caso de Uso “Gestionar Datos de Media”.....	68
3.5.3.6	Caso de Uso “Gestionar Plan de Programación”.....	70
3.5.3.7	Caso de Uso “Gestionar Datos de equipamiento”.....	73
3.6	Conclusiones.....	76
<b>CAPÍTULO 4</b>	<b>.....</b>	<b>77</b>
4.1	Introducción.....	77
4.2	Diagramas de clases del análisis.....	77
4.3	Diagramas de Interacción (Colaboración).....	79
4.4	Modelo del Diseño.....	81
4.5	Diagrama de Clases del Diseño.....	81
4.6	Patrones de diseño en Symfony.....	87
4.7	Diagrama de Clases Persistentes.....	89
4.8	Modelo de Datos.....	91
4.7	Descripción de las tablas del Modelo de Datos y sus atributos.....	92
4.8	Modelo de Despliegue.....	99
4.9	Conclusiones.....	100
<b>CONCLUSIONES</b>	<b>.....</b>	<b>101</b>
<b>RECOMENDACIONES</b>	<b>.....</b>	<b>102</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>.....</b>	<b>103</b>
<b>GLOSARIO</b>	<b>.....</b>	<b>106</b>

## INTRODUCCIÓN

En la actualidad el manejo de información es la premisa fundamental para lograr los objetivos propuestos en cada unas de las esferas de la sociedad, esta situación en muchos casos tiende a ser voluminosa, desorganizada y compleja en cuanto a su interpretación y se hace muy lento el procesamiento de la misma de forma manual.

Por todos es conocida la gran capacidad que poseen los ordenadores modernos, los cuáles son capaces de realizar cálculos millones de veces más rápido que la poderosa mente humana, aprovechando estas funcionalidades y utilizándolas en nuestros intereses se logran realizar sistemas automatizados capaces de gestionar la información de una forma rápida y eficiente.

Debido a la escasez de profesores existente en nuestra universidad desde sus inicios, incluso a las necesidades de utilizar los medios puestos en nuestra universidad para dar apoyo a todo el proceso docente educativo y dar cobertura informática a los estudiantes, surge la Dirección de Televisión Universitaria (DTU) desde el primer año de vida de la Universidad de las Ciencias Informáticas.

La información que se procesa en la DTU no queda exenta de este mal, numerosos son los procesos y la información que se gestiona a diario en cada uno de los departamentos de dicha institución, cada uno de estos pasa a través de un flujo de trabajo que es diferente para cada uno de los 17 servicios que hasta ahora se brindan.

Partiendo del conocimiento de toda la información que se procesa en cada uno de los servicios que se presta se realizó un estudio investigativo en el área problemática, se identificaron elementos y se presentaron soluciones para el diseño del sistema de gestión de los procesos que se generan en la DTU. Al realizar esta investigación se utilizaron varios métodos de análisis donde se presentaron soluciones adecuadas para el manejo de la información.

Como parte de esta gran ciudad digital y debido al gran volumen de información que se gestiona a diario de forma deficiente no es posible que estos servicios no estén al margen del desarrollo cotidiano e incremental de la UCI por lo que se hace muy necesario la automatización de los procesos de

gestión de todos los servicios prestados en dicha dirección, logrando con esto un mejor funcionamiento de la misma debido a los grandes problemas que existen actualmente con la gestión de los procesos que allí se llevan a cabo.

Con el aumento de los servicios que presta la DTU y la complejidad de los mismos es por lo que se nos plantea la necesidad del diseño de un sistema de gestión para los procesos de la DTU de la UCI, facilitando de esta forma una mejor organización de todos los materiales así como la información que se gestiona a diario en dicha dirección.

Este proyecto pudiera servir como base a otros de su tipo en cualquier productora de televisión tanto nacional como extranjera aunque se diferencien en algunos procesos en específicos.

Lo esperado de este trabajo es presentar un diseño para los procesos de la DTU que cubra todos los requerimientos necesarios para dar solución a los mismos.

En general el **problema** que surge de esta investigación a raíz de las situaciones expuestas es: la inexistencia de un sistema automatizado que de respuesta a las necesidades de gestión de los procesos que se llevan a cabo en la DTU lo que lleva de manera natural a la conclusión de la necesidad del diseño de un sistema que gestione los procesos de la dirección y contribuya a aumentar la calidad en los servicios en este ámbito, lo cual es el propósito de este trabajo.

El **objeto de estudio** lo constituyen los procesos que se desarrollan en la DTU para de esta forma garantizar mejores servicios.

EL **campo de acción** que abarca este trabajo es la automatización de la gestión de los procesos que surgen como resultado de todos los flujos de trabajo que se generan en la DTU.

Lo planteado anteriormente lleva a la formulación de los siguientes objetivos:

### **Objetivo General:**

Diseñar un sistema de gestión de procesos para la DTU, que permita administrar los flujos de trabajo dentro de la estructura de la institución.

A partir de un análisis del objetivo general se derivan los siguientes objetivos específicos:

**Objetivos Específicos:**

- Realizar un estudio del arte de los sistemas de gestión de procesos para productoras de TV.
- Realizar un análisis de los procesos que rigen los flujos de trabajo en la DTU.
- Analizar y diseñar un sistema de gestión de procesos para la DTU.
- Proponer herramientas de software libre que se puedan utilizar en el desarrollo del sistema al concluir el diseño.

Para cumplir los objetivos anteriormente expuestos se desarrollaron las siguientes **tareas**:

- Investigar sobre sistemas de gestión.
- Investigar sobre los sistemas de gestión para productoras de TV existentes.
- Estudio de procesos dentro de la DTU.
- Capturar y clasificar los requisitos funcionales y no funcionales.
- Identificar, clasificar y describir casos de uso del sistema.
- Estructurar el modelo de sistema.
- Estructurar modelo de diseño.
- Investigar sobre las herramientas de software libre que se puedan utilizar en el desarrollo del sistema, para proponerlas al concluir el diseño.
- Proponer la Plataforma de desarrollo a utilizar.

El documento está estructurado en tres capítulos, que incluye todo lo relacionado con el trabajo investigativo realizado, así como el análisis y el diseño del sistema que se propone.

**Capítulo I.** Fundamentación Teórica, incluye el análisis de la información existente acerca del tema a tratar a nivel internacional, nacional, de la Universidad y se describe el objeto de estudio en que se desarrolla al sistema.

**Capítulo II.** Tendencias y tecnologías actuales, incluye el análisis de la información existente acerca de las tendencias y tecnologías actuales que existen en el mundo.

**Capítulo III.** Presentación de la Solución Propuesta. Se describe el negocio aclarando los principales procesos y a partir de ese análisis proponer una solución al problema.

**Capítulo IV.** Análisis y diseño del sistema. Se describe el análisis y el diseño propuesto para el sistema.

# CAPÍTULO 1

## Fundamentación Teórica.

### 1.1 Introducción

En este capítulo se tratará aspectos generales de la DTU. Brindaremos una explicación detallada sobre los procesos que se gestionan en ella en cada uno de los departamentos. Se introduce a los problemas que se presentan en el departamento con el manejo de la información allí generada y se analiza las variantes posibles para la solución de dichos problemas. A nivel internacional y nacional se buscan y comparan alternativas.

### 1.2 Conceptos asociados al dominio del problema

**Televisión (TV):** transmisión instantánea de imágenes, tales como fotos o escenas, fijas o en movimiento, por medios electrónicos a través de líneas de transmisión eléctricas o radiación electromagnética (ondas de radio).

**Producción televisiva:** técnicas empleadas para crear un programa de televisión.

**Subprocesos:** son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso.

**Sistema:** Estructura organizativa, procedimientos, procesos y recursos necesarios para implantar una gestión determinada, como por ejemplo la gestión de la calidad, la gestión del medio ambiente o la gestión de la prevención de riesgos laborales. Normalmente están basados en una norma de reconocimiento internacional que tiene como finalidad servir de herramienta de gestión en el aseguramiento de los procesos.

**Web:** sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (*el web, un web*) se refiere a un sitio web entero, en cambio si se

utiliza en femenino (*la web, una web*) se refiere a una página web concreta dentro del sitio web.

**Información Audiovisual:** Contenido de todas las tipologías audiovisuales (teleclases, audiovisuales varios, filmes, series, fotos, música, videos musicales, documentales, programas, efectos de sonido).

**Solicitudes de Información Audiovisual:** Solicitudes que realizan las entidades externas en función de sus actividades, así como las solicitudes que se le hacen a las instituciones afines en correspondencia con las necesidades del CGIA para satisfacer las demandas.

**Solicitudes de Información Audiovisual Aprobadas:** Solicitudes aprobadas por los directivos de la DTU, que se envían al CGIA.

**Información Audiovisual Solicitada:** Respuesta de su existencia o no, en el fondo del CGIA.

**Información Audiovisual Recuperada:** Si existe en el fondo del CGIA, se revisa y está en buen estado, se entrega a la entidad que la solicitó.

**Información Audiovisual a Editar:** Si existe en el fondo del CGIA, se revisa y no está en buen estado, se solicita edición o reedición, a Producción o al Proyecto UCITeVe.

**Información Audiovisual Docente:** Teleclases y materiales complementarios docentes.

**Desiderata:** Listados de audiovisuales que no existen en el fondo del CGIA, surgen a partir de estudios de necesidades y las solicitudes.

**Información Audiovisual Editada:** Se le ha aplicado todos los procesos de producción y está listo para procesar.

**Información Audiovisual Procesada:** Audiovisuales descargados del servidor, analizados y procesados por los especialistas.



**Información Audiovisual en DVD:** Audiovisuales quemados en DVD y almacenados en el Fondo del CGIA, listos para ser consultados y recuperados para prestar servicios.

### **1.3 Objeto de Estudio**

#### **1.3.1 Descripción General**

Desde hace muchos años la televisión ha sido uno de los medios informativos más importantes en el desarrollo de muchas esferas de la vida del hombre. Este medio ha servido para difundir, enseñar y educar a todos los que tienen el privilegio de disfrutarla.

Unido al desarrollo de la informática y las comunicaciones este medio ha alcanzado grandes logros, incluso llegando a convertirse en un medio de comunicación por excelencia.

Desde que surgió la informática en Cuba, se ha ido desarrollando cada vez más la tecnología, pues poco a poco se ha ido introduciendo el tema de las computadoras en la vida social e intelectual de la población.

En Cuba la televisión es un medio muy utilizado actualmente en la gran batalla de ideas en la que está inmerso el pueblo, en medio de la cual surge la Universidad de las Ciencias Informáticas (UCI). Debido a la gran matrícula del centro la cual fue creciendo año tras año, y a la necesidad de contar con un sistema propio que se encargara de todo lo relacionado con tele clases, mantener informada a la comunidad universitaria, recoger y archivar los eventos y las visitas mas importantes a la universidad entre otras tareas se crea la Dirección de Televisión Universitaria (DTU).

#### **1.3.2 Descripción actual del dominio del problema**

La Dirección de Televisión de la UCI esta dividida en 3 grandes departamentos: Tecnología, Informativo y el Departamento de Programación, entre los cuales existe un gran flujo de actividades que son las que hacen funcionar correctamente la DTU. En el departamento de Tecnología se lleva un control de todos los medios asignados a la dirección, desde cables, micrófonos, cámaras, hasta las computadoras con las que trabaja el personal de la dirección. El departamento Informativo es el encargado de recibir todas las informaciones provenientes de la dirección de la Universidad, de las

facultades, de los estudiantes pertenecientes al Movimiento de Código y Letra y de los estudiantes y trabajadores de forma general que envían Avisos, Reportajes, Artículos, Noticias entre otras Informaciones y estas son publicadas en Intranet y en señal 3 fundamentalmente.

A su vez el departamento de Programación esta dividido en 3 Áreas, esta el Área de Programación que es el encargado a partir de la efemérides de la semana, de las películas de estreno descargadas por los estudiantes del proyecto UCITeVe y los materiales del archivo conformar la programación del canal 7 semanalmente, otra de las áreas es el CGIA la cual brinda servicios a la FEU, las facultades, la dirección de la Universidad, así como a los mismo departamentos de la Dirección, aquí se brindan servicios como digitalización, préstamos y cualquier material audiovisual que se necesite para cualquier entidad en la Universidad aquí se les facilita, cuando alguna entidad llega al CGIA hace su solicitud del material que necesita, si este se encuentra y no hay que hacerle ninguna modificación se le entrega al usuario, si lleva algún procesos de edición se envía a producción o al proyecto UCITeVe en dependencia del tipo de material, aquí se le hace el proceso que necesite, se le entrega al cliente y en caso que sea algún material que se haya tenido que hacer se quema y va a formar parte del fondo del CGIA, en caso que el lo que se pida no se encuentra, esto pasa por un procesos de desiderata y se procede a la gestión del mismo en entidades afines a la Universidad y la otra de las áreas es la de producción que es la encargada de realizar la producción de las medias que se solicitan a la dirección así como realizar procesos de edición, digitalización y conversión a otras que lo requieran.

### **1.3.3 Situación Problemática**

Existen aspectos deficientes que han generado la situación problemática que se pretende resolver con este trabajo, actualmente la DTU cuenta con su propia televisora o productora de televisión, donde se producen diferentes tipo de materiales, presta servicios como la digitalización de materiales, búsqueda y recopilación de información audiovisual, prestamos de materiales entre otros servicios tanto dentro como fuera de la universidad. Toda esta información es muy voluminosa y va aumentando a medida que pasa el tiempo con el crecimiento y desarrollo de la universidad por lo que se hace muy engorroso trabajar con tan grandes volúmenes de información. Además de esta deficiencia tenemos algunas otras como por ejemplo la distribución del plan de trabajo se manda diariamente por correo lo cual es un problema pues algún que otro trabajador puede tener el buzón de su correo lleno y no recibir este mensaje, los planes de producción, que es el documento en el que se planifican todas las actividades de la dirección y el plan de programación que es donde se refleja la planificación de la programación

del canal 7 o mas conocido como Canal Cultural, se realizan en un documento Excel, siendo muy engorroso este proceso y la búsqueda de cualquier información que se necesite en estos documentos son procesos muy engorrosos y a veces se necesita de mucho tiempo, el cual en ocasiones es muy corto y dificulta el horario de las actividades previstas.

#### **1.4 Análisis de otras soluciones existentes**

Como resultado de investigaciones realizadas, y exhaustivas búsquedas en internet de algún sistema de este tipo en el mundo no se encontró ningún sistema con estas características, solamente se encontró uno que fue el resultado de una tesis del curso anterior, el cual es un sistema de gestión de procesos pero solamente de una de las áreas de uno de los 3 departamentos principales de la DTU, este sistema no es posible utilizarlo pues es para una sola área de uno de los departamento y este trabajo esta dirigido a todos los departamentos de la DTU, después de implementado dicho sistema debido a cambios que surgieron en la dirección se modificaron los flujos de trabajo de ese departamento por lo que ese sistema nunca se llegó a utilizar.

#### **1.5 Conclusiones**

A modo de de conclusión, en este capítulo se ha hecho un análisis profundo sobre la situación existente en la Dirección de Televisión Universitaria actualmente, se ha visto como el trabajo manual que aquí se hace con tan grandes volúmenes de información hace engorroso y demora muchos de los procesos que aquí se gestiona a diario y la importancia de la humanización de trabajo como parte de la informatización de la universidad, tarea en la que se encuentra enfrascada la universidad en estos momentos.

## CAPÍTULO 2

# Tendencias y tecnologías actuales a desarrollar.

### 2.1 Introducción

En este capítulo se evidencia el estudio realizado de las tendencias y tecnologías actuales que se puedan emplear, de las cuales se brinda una breve descripción, teniendo como premisa proponer que la futura implementación de este sistema se realice utilizando herramientas libres.

### 2.2 Las Aplicaciones Web

Una aplicación de escritorio se ejecuta en la PC cliente. Para iniciarla y hacerla funcionar se requiere estar delante del ordenador que tiene instalada dicha aplicación. Entre las ventajas que presenta este tipo de aplicación es que su ejecución no requiere habitualmente de comunicaciones con el exterior, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales y como principal desventaja podemos decir que su acceso se limita a la PC donde se instalan. Son dependientes del sistema operativo que utilice tu ordenador y de sus capacidades (video, memoria, etc.).

Una aplicación Web por el contrario reside en un servidor, y su ejecución requiere disponer de un PC con conexión a Internet, un navegador como Internet Explorer, Mozilla Firefox, Opera, etc. y por supuesto que la aplicación este funcionando en el servidor que la aloja estas proporcionan movilidad, dado que puedes ejecutarlas desde cualquier ordenador con conexión a Internet. La información que manejan se accede a través de Internet, motivo por el cual son especialmente interesantes para desarrollar aplicaciones multiusuario basadas en la compartición de información. El cliente o usuario que utiliza la aplicación no necesita tener un ordenador de grandes prestaciones para trabajar con ella como aspectos negativos podemos plantear que la comunicación constante con el servidor que ejecuta la aplicación establece una dependencia con una buena conexión a Internet. Además, el servidor debe

tener las prestaciones necesarias para ejecutar la aplicación de manera fluida, no sólo para un usuario sino para todos los que la utilicen de forma concurrente.

En la actualidad existen infinitas posibilidades de acceder a la información desde casi cualquier sitio del mundo gracias al Internet, lo cual representa un desafío para todas aquellas personas que desarrollan la Web pues los avances tecnológicos demandan cada día aplicaciones más rápidas, ligeras y robustas.

La Web hoy día es un medio muy flexible y económico para todo lo que respecta a la comunicación, el acceso a información y servicios, negocios y comercio, entretenimiento, etc.

Desde las primeras tecnologías básicas: HTML y HTTP hasta nuestros días han emergido tecnologías como CGI, Java, Javascript, ASP, JSP, PHP, Flash, J2EE, XML, por citar algunas de las más conocidas, que permiten una Web mejor, más amplia, más potente, más flexible, o más fácil de mantener.(1)

Una Aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero.

La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Existen dos tipos de aplicaciones Web según su entorno de funcionamiento. Este puede ser una Intranet o por el contrario una extranet. En el primer caso, la aplicación se aloja y está sólo disponible dentro de la red interna de la empresa y en el segundo, lo está en cualquier punto con acceso a Internet. Entonces si no se dispone de una buena conexión y/o se prefiere tener las aplicaciones dentro de la organización se puede hacer sin renunciar a las ventajas de las tecnologías de Internet.

### **Ventajas de las Aplicaciones Web:**

Multiplataforma: Con un solo programa, un único ejecutable, nuestras aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.

Actualización instantánea: Debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

Suave curva de aprendizaje: Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.

Fácil de integrar con otros sistemas: Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.

Acceso móvil: El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

El desarrollo de aplicaciones Web está siendo utilizado en muchas organizaciones, ésta situación va ir creciendo indefinidamente. Es por ello que día a día se requieran más programadores capacitados para desarrollos basados en el Word Wide Web (WWW).(2)

Después de haber analizado el sistema que se quiere desarrollar y las necesidades de la DTU así como el tipo de servicios que se requiere preste el sistema y por lo antes expuesto se decide que lo más conveniente sería una aplicación Web pues con esta se resolvería la situación problemática planteada.

### **2.3 Modelo Cliente Servidor.**

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.(3)

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo

datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura.(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.(5)

Ventajas de la arquitectura cliente/servidor:

El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.

Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.(2)

**Arquitectura Modelo-Vista-Controlador.**

La arquitectura Modelo-Vista-Controlador (MVC), es la implementación de la arquitectura en tres capas más extendida. Actualmente la arquitectura MVC es la alternativa más utilizada en la creación de páginas web.

**¿Qué es un patrón?**

Un patrón o modelo de programación, es una solución a un problema que constantemente aparece en el desarrollo programas. Este trata de describir con detalle la solución del problema en forma de plantilla, aunque no siempre de esta forma se encuentra.

Existen varias clasificaciones de patrones de software como los de creación, estructurales y comportamiento, que ayudan a resolver problemas de desarrollo.

Uno de los patrones que más se utilizan en desarrollo web es el modelo MVC (Model, View, Controller) o Modelo, Vista y Controlador.

### **El Patrón Modelo-Vista-Controlador (MVC).**

El modelo MVC es uno de los patrones que más se utiliza en la construcción de aplicaciones web. Este permite construir en tres capas de aplicación, permitiendo la separación del código entre cada una de las capas, y ayudando tanto a desarrolladores como a diseñadores a cooperar y mantener el código fuente más fácilmente, además permite que el software sea fácil de modificar.

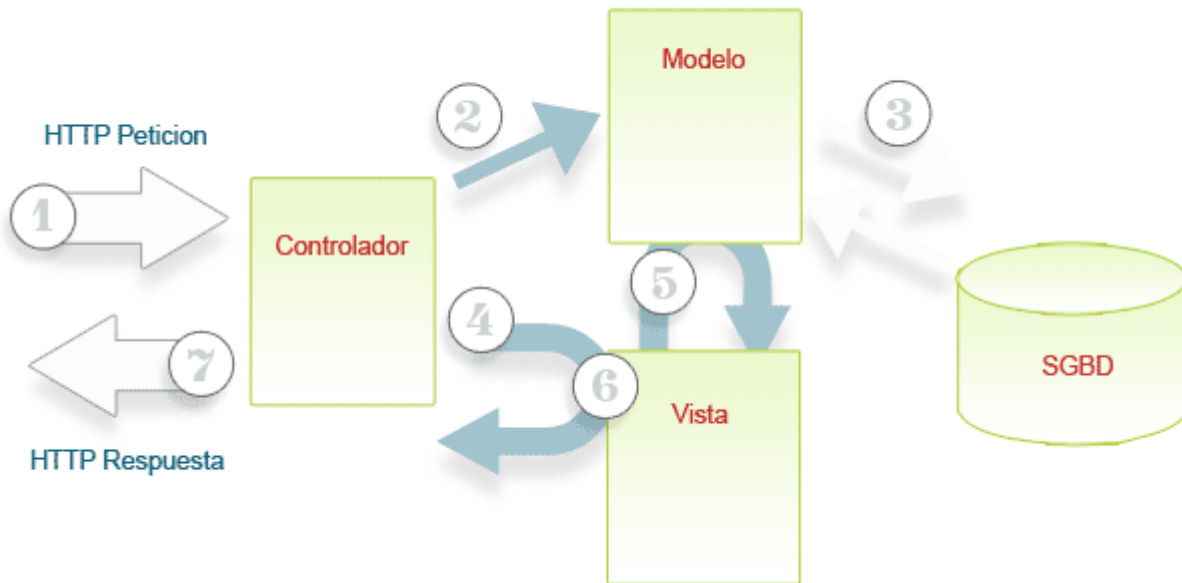
- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada. El modelo debe de preservar la integridad de los datos.
- **Vista (View):** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. Interactúa con la interfaz de usuario.
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos, e interpreta las operaciones del usuario; codificando los movimientos, pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests") para el modelo o la vista. Es el que debe de controlar los eventos.

### **Ventajas**

Ya se han ido comentando las ventajas de este patrón, pero ahora se recopilarán en este punto.

- Se consiguen 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).





**Fig. 1 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

El Modelo Vista Controlador no es una cosa reciente. Surgió en los años 70 al empezar a hacer aplicaciones medianamente grandes que manejaban muchos datos. Es el llamado “Modelo\* 2” en la jerga Sun.

*\*Aquí la palabra Modelo no tiene que ver con el Modelo de Modelo Vista Controlador sino con el nombre de la arquitectura.*

### **¿Por qué usarlo entonces?**

Cuando en un proyecto medianamente grande se desee cambiar algo esto podría resultar muy engorroso en cambio el MVC básicamente te permitirá poder cambiar las Capas de presentación sin que afecten a la de acceso de datos y viceversa.

Hay pequeñas diferencias en como implementan el MVC distintos frameworks, pero los fundamentos son los mismos.

También se pueden usar motores de plantillas para implementar el MVC, pero en realidad no se necesita nada más que el PHP puro.

### **¿Qué es un framework?**

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Para nuestra investigación se realizó un análisis de varios frameworks entre los que se encuentran: Aurora, CakePHp, Cumbia y Symfony, teniendo en cuenta varios criterios como por ejemplo: utilización de patrón MVC, compatibilidad con diferentes bases de datos, uso de PHP 5, uso de plantillas, curva de aprendizaje suave, buena documentación, una comunidad activa y facilidad de instalación, teniendo en cuenta estos factores y teniendo en cuenta que hoy en nuestra Universidad existe una tendencia a utilizar este framework se ha seleccionado para proponer en nuestra solución a Symfony, por todas las ventajas y funcionalidades que este posee las cuales se presentan a continuación:

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

Symfony está desarrollado completamente con PHP 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.

A continuación se muestran algunas de sus características.

### **Características de Symfony**

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- 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.

### **Automatización de Características de Proyectos Web**

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.

### Entorno de desarrollo y herramientas

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:

- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas ("test-driven development").
- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

### 2.4 Lenguajes de Programación Web

En la actualidad existen varios lenguajes de programación para desarrollar en la Web, estos han ido surgiendo debido a las tendencias y necesidades de las diferentes plataformas. Primero surgieron los lenguajes estáticos para Web y luego con el desarrollo de la tecnología y el aumento de la demanda de usuarios se crearon los lenguajes dinámicos para de esta forma permitir la interacción con los usuarios y la utilización de sistemas de bases de datos.

Los lenguajes de programación Web se clasifican en dos partes fundamentales:

- Los lenguajes del lado del Servidor.
- Los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor podemos encontrar entre los más sobresalientes por el auge que estos han tenido, algunos como PERL, ASP, PHP, Java, etc. Estos se caracterizan por desarrollar

la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información entre otras características. Del lado del cliente se encuentra principalmente el Java Script, encargado de aportar dinamismo a la aplicación en los navegadores.

### **Lenguaje HTML**

Desde el surgimiento de Internet se han publicado sitios Web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios Web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C) Los archivos pueden tener las extensiones (htm, html).(6)

### **Lenguaje Javascript**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas Web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de nuestras páginas Web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).(6)

### **Lenguaje PHP**

Es un lenguaje de programación utilizado para la creación de sitio Web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).(6)

Ventajas:

- 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. (6)

### **Seguridad:**

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, es capaz 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.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.(6)

Seguidamente se analizará el lenguaje propuesto para la realización de este sistema.

#### **2.4.1 PHP**

PHP (Hypertext Preprocessor) es un lenguaje de programación de alto nivel y ejecutado en el lado del servidor. El PHP originalmente fue diseñado por Rasmus Lerdorf en 1994 como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.(7)

A mediados de 1997 se hizo otra gran contribución al lenguaje cuando se volvió a programar el analizador sintáctico, esta vez por los programadores Andi Gutmans y Zeev Suraski, en ella se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales, creando así las bases de PHP 3. Una de sus mejores características era su extensibilidad. Además de proveer a los usuarios finales de una sólida infraestructura para muchísimas bases de datos, protocolos y APIs, las características de extensibilidad de PHP 3.0 atrajeron a docenas de desarrolladores a unirse y enviar nuevos módulos de extensión. Sin duda, ésta fue la clave del enorme éxito de PHP 3.0.(7)

El nuevo motor, apodado 'Motor Zend', otra reescritura de Suraski and Gutmans, se introdujo por primera vez a mediados de 1999. PHP 4.0, basado en este motor, y acoplado con un gran rango de nuevas características adicionales, fue oficialmente liberado en Mayo de 2000, casi dos años después que su predecesor, PHP 3.0. Además de la mejora de ejecución de esta versión, PHP 4.0 incluía otras características clave como el soporte para la mayoría de los servidores Web, sesiones HTTP, buffers de salida, formas más seguras de controlar las entradas de usuario y muchas nuevas construcciones de lenguaje.(7)

Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre las que se pueden mencionar InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML (eXtensible Markup Language) y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+.(8)

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes.(8)

Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.(8)



PHP es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson y NASA.(8)

Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.(8)

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el Zend Engine II (o Zend Engine 2) que incluye todas las ventajas que provee el nuevo Zend Engine 2 como:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM, etc.).
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones.

Los diseñadores de PHP5 han realizado un cambio radical en el tratamiento de las variables objeto: en PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador '&' ni en las asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de potenciales errores. La principal novedad en las clases de PHP5 es la inclusión de modificadores de control de acceso para implementar la encapsulación piedra angular en la programación orientada a objetos de la que adolecía PHP4.(9)

### **2.5 Servidores Web**

Un servidor Web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP).(10)

Servidor Web: Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario.(11)

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

**Algunos servidores Web importantes son:**

- Apache.
- Internet Information Server.

### 2.5.1 Apache

Apache es un servidor HTTP, de código abierto y licenciamiento libre, que funciona en Linux, sistemas operativos derivados de Unix™, Windows, Novell NetWare y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continua siendo el servidor HTTP más utilizado, siendo además el servidor *de facto* contra el cual se realizan las pruebas comparativas y de desempeño para otros productos competidores. Apache es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por Apache Software Foundation.(12)

Apache es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP). Entre sus características se destacan:(13)

- Multiplataforma
- Es un servidor de Web conforme al protocolo HTTP/1.1

- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías: (14)

Módulos Base: Módulo con las funciones básicas del Apache.

Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.

Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.(15)

### **2.6 Sistema de Gestión de Base de Datos**

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

Para plasmar los tres niveles en el enfoque o modelo de datos seleccionado, es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que desempeñan los SGBD, ya reseñados, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Se han señalado como componentes de un sistema ideal de gestión de bases de datos los siguientes (FROST, 1989: 90):(16)

1. Un lenguaje de definición de esquema conceptual.
2. Un sistema de diccionario de datos.
3. Un lenguaje de especificación de paquetes de entrada/salida.
4. Un lenguaje de definición de esquemas de base de datos.
5. Una estructura simétrica de almacenamiento de datos.
6. Un módulo de transformación lógica a física.
7. Un subsistema de privacidad de propósito general.
8. Un subsistema de integridad de propósito general
9. Un subsistema de reserva y recuperación de propósito general.
10. Un generador de programas de aplicación.
11. Un generador de programas de informes.
12. Un lenguaje de consulta de propósito general.

**Por tanto un Sistema Gestor de Base de Datos debe permitir:**

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

**Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son:(17)**

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.

- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

Los Sistemas Gestores de Base de Datos más usados son Oracle, MySQL, SQL Server de Microsoft, PostgreSQL entre otros.

**Oracle:** requiere de una licencia para poderlo utilizar, es decir, es necesario pagar para poder utilizarlo.

**Microsoft SQL Server:** no es multiplataforma, solo puede ser utilizado con el sistema operativo Windows que está patrocinado por la compañía Microsoft.

**PostgreSQL:** soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.

Como SGBD se seleccionó el PostgreSQL por las ventajas que ofrece y porque se requiere sea un software libre.

### 2.6.1 Postgre SQL

PostgreSQL es un potente sistema de base de datos relacional libre (open source, su código fuente está disponible) liberado bajo licencia BSD. Tiene más de 15 años de activo desarrollo y arquitectura probada que se ha ganado una muy buena reputación por su confiabilidad e integridad de datos.(18)

Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.(18)

El desarrollo de PostgreSQL es realizado por un equipo de desarrolladores (voluntarios en su mayoría) dispersos alrededor del mundo y comunicados vía Internet. Este es un proyecto de la comunidad y no es controlado por ninguna compañía.(18)

### Principales características (18)

- Soporta casi toda la sintaxis SQL tiene soporte total para foreign keys, joins, views, triggers, y stored procedures (en múltiples lenguajes).
- Integridad transaccional, obedece completamente a la especificación ACID.
- Acceso concurrente multiversión, MVCC Control de Concurrencia Multi-Versión (Multi- Version Concurrency Control), no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe. Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Mediante el uso de VCC, PostgreSQL evita el problema de que procesos lectores estén esperando a que se termine de escribir. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- Write Ahead Logging (WAL): La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.
- Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido. Además de en C, C++ y, Java.

- Interfaces con lenguajes de programación. La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, Pike, etc.
- Herencia de tablas.
- Incluye la mayoría de los tipos de datos SQL92 y SQL99 (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP), soporta almacenamiento de objetos grandes binarios, además de tipos de datos y operaciones geométricas.
- Puntos de recuperación a un momento dado, tablespaces, replicación asincrónica, transacciones jerarquizadas (savepoints), salvadas en línea.
- Un sofisticado analizador/optimizador de consultas.
- Soporta juegos de caracteres internacionales, codificación de caracteres multibyte.

PostgreSQL funciona con una arquitectura Cliente/Servidor, un proceso servidor (postmaster) y una serie de aplicaciones cliente que realizan solicitudes de acciones contra la base de datos a su proceso servidor. Por cada una de estas aplicaciones cliente, el proceso postmaster crea un proceso postgres.

(18)

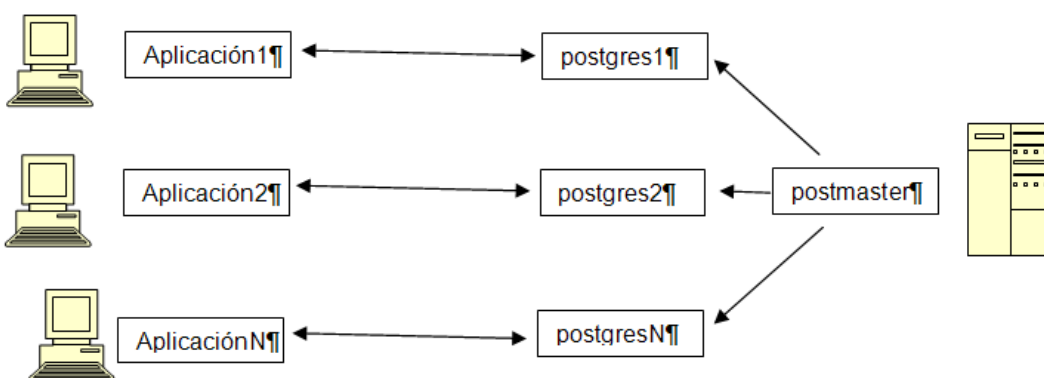


Fig. 2 Arquitectura PostgreSQL

## 2.7 Proceso de Desarrollo de Software.

Para un mejor desarrollo de la propuesta que presenta este trabajo, se propone utilizar como metodología el Proceso Unificado de Modelado (RUP), debido a las características que posee y las facilidades que aporta a todo el proceso.

El Rational Unified Process (RUP) es un proceso de ingeniería de software que mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo.(19)

RUP es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software.

### **Las características del proceso unificado de modelado son:**

Guiado/Manejado por casos de uso: La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.(20)

Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.(20)

Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.(20)



Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollen y maduran sus componentes.(20)

Utilización de un único lenguaje de modelado: UML es adoptado como único lenguaje de modelado para el desarrollo de todos los modelos.(20)

Proceso Integrado: Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.(20)

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación:

### **Fase 1: Preparación Inicial (“Inicio”)**

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.(20)

### **Fase 2: Preparación Detallada (“Elaboración”)**

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.(20)

### **Fase 3: Construcción (“Construcción”)**

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.(20)

**Fase 4: Transición (“Transición”)**

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.(20)



Fig. 3 Fases e Hitos de un Proyecto (21)

**Hitos**

Cada fase finaliza con un hito. Cada hito se determina por la disponibilidad de un conjunto de artefactos, es decir un conjunto de modelos o documentos que han sido desarrollados hasta alcanzar un estado predefinido.

Los hitos tienen muchos objetivos. El más crítico es que los directores deben tomar ciertas decisiones antes de que el trabajo continúe con la siguiente fase.

Los hitos también permiten controlar la dirección y progreso del trabajo.

Al final se obtiene un conjunto de datos a partir del seguimiento del tiempo y esfuerzo consumidos en cada fase. Estos datos son útiles para las estimaciones en futuros proyectos.(21)

**RUP define nueve Flujos de Trabajo:(22)**

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Prueba
6. Distribución
7. Gestión de configuración y cambios
8. Gestión del proyecto
9. Gestión del entorno

Y el flujo de trabajo (workflow) entre ellas en base a los llamados diagramas de actividad. El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado (artefactos en la jerga de RUP) que se espera de ellos.

RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal.(22)

**2.8 Lenguaje Unificado de Modelado (UML)**

Cualquier rama de ingeniería o arquitectura ha encontrado útil desde hace mucho tiempo la representación de los diseños de forma gráfica. Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico.(23)

Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: *Unified Modeling Language*). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño.(23)

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de des-pliegue.(23)

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía *Rational* fundada por Booch (dos reputados investiga-dores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (*Object Modelling Tool*). El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a *Rational* y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML.(23)

El Lenguaje de Modelado Unificado UML (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. (24)

UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar.

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Los principales beneficios de UML son:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema. Umbrello UML Modeller soporta los siguientes tipos de diagramas:(25)

- Diagrama de casos de uso que muestra a los actores (otros usuarios del sistema), los casos de uso (las situaciones que se producen cuando utilizan el sistema) y sus relaciones.
- Diagrama de clases que muestra las clases y la relaciones entre ellas.
- Diagrama de secuencia muestra los objetos y sus múltiples relaciones entre ellos.
- Diagrama de colaboración que muestra objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.
- Diagrama de estado muestra estados, cambios de estado y eventos en un objeto o en parte del sistema.
- Diagrama de actividad que muestra actividades, así como los cambios de una a otra actividad junto con los eventos que ocurren en ciertas partes del sistema.

- Diagrama de componentes que muestra los componentes de mayor nivel de la programación (cosas como Kparts o Java Beans).
- Diagrama de implementación que muestra las instancias de los componentes y sus relaciones.
- Diagrama de relaciones de entidad que muestra los datos y las relaciones y restricciones entre ellos.

El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo.

## **2.9 Conclusiones**

Por todo lo que se ha investigado a lo largo de este capítulo se ha decidido que para un mejor cumplimiento de nuestros objetivos es conveniente realizar el diseño de una aplicación web con una arquitectura cliente/servidor, utilizando como lenguaje de programación web el PHP, un servidor web Apache, como gestor de base de datos Postgre SQL y para su modelado se decidió utilizar como Lenguaje para el modelado el Lenguaje Unificado de Modelado (UML).

## CAPÍTULO 3

### Presentación de la solución propuesta.

#### 3.1 Introducción

En este capítulo se describe el negocio, aclarando sus principales procesos y datos de información manipulable resultantes, para a partir de ese análisis proponer una solución presentando un sistema que cumpla con los requisitos funcionales que respondan a las necesidades de informatización que se planteen en el modelado del negocio.

#### 3.2 Modelo de Negocio

Cuando se comienza a desarrollar un sistema para una organización, empresa e institución es necesario comprender todos los procesos que en estas tienen lugar. Cada uno de estos procesos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la organización. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y reglas de negocio.

##### 3.2.1 Procesos del Negocio.

Un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente o el mercado, estos son de gran importancia ya que definen los límites del proceso de modelado posterior.

##### Selección de los procesos.

En esta etapa se seleccionan los procesos relevantes y claves de la organización, que son las bases del diseño que se propone la organización y se finaliza con la clasificación de estos.

Tiene como objetivos:

1. Seleccionar todos los procesos relevantes y claves en la Dirección de Televisión Universitaria.
2. Clasificar los procesos de dicha dirección.

Identificación de los procesos relevantes de la DTU.

En esta fase se procedió a la identificación de los procesos relevantes de la DTU. Que son aquellos procesos de alcance globales, que normalmente suele atravesar las delimitaciones funcionales. Para completar el proceso fue necesaria la intervención de varios miembros de la organización.

Se determinaron los procesos atendiendo a:

- Clientes de la DTU.
- La naturaleza de sus actividades.
- Su estrategia.

Para la selección se tuvieron en cuenta las siguientes premisas. (Nogueira, 2004):

- El nombre asociado a cada proceso debe ser representativo de lo que conceptualmente representa o pretende representar.
- El mayor número de las actividades desarrolladas en la DTU deben estar incluidas en algunos de los procesos listados.
- Se deben identificar procesos que aporten valor al producto.
- Los procesos deben estar enfocados a la satisfacción tanto del cliente interno como externo.

Estos procesos claves se clasifican en dos grupos:

- Procesos claves estratégicos: Son aquellos procesos claves que su mayor importancia fue otorgada al impacto de los procesos, con los objetivos estratégicos.
- Procesos claves operativos: Son aquellos procesos claves que su mayor puntuación fue otorgada al peso y a la repercusión con el cliente.

El resto de los procesos relevantes también se clasifican en:



- Procesos Estratégicos.
- Procesos de Apoyo.

Teniendo en cuenta todo esto, se analizaron los procesos que se gestionan en la DTU, los servicios que se prestan y sus principales clientes llegando a la conclusión que:

Como Procesos Claves Estratégicos:

- **Realizar Plan de Producción:** Plan donde se recogen todas las solicitudes de producción llegadas a la DTU, y se confecciona un plan con la planificación de las actividades a realizar en la dirección durante toda la semana.
- **Realizar Plan de Programación:** En este plan se recoge la planificación de la programación del canal 7 de toda la semana, especificándolo por días y horarios con los materiales asociados.

Como Proceso Clave Operativo:

- **Entregar Medias:** Cuando un Representante de una entidad necesita una media se dirige a la dirección de la DTU, si es aprobada pasa la solicitud al CGIA donde se entrega la media.

Y como procesos de Apoyo

1. **Producir Media:** Cuando llega una solicitud para producir una media a la dirección de la DTU si esta es aprobada se envía al departamento de producción el cual planifica y lleva a cabo la producción de la misma.
2. **Trasmitir Programación:** Proceso de transmitir la programación programada anteriormente en el plan de programación.

#### 3.2.2 Reglas del Negocio.

- Todas las personas interesadas en una media deben dirigirse a la dirección de la DTU para hacer su solicitud.

- Todas las personas interesadas en que se le produzca una media deben dirigirse a la dirección de la DTU para llenar la planilla de solicitud.
- Solamente el productor hace el plan de producción
- El Jefe de Dpto. Técnico es quien asigna el equipamiento especializado de televisión a los grupos de trabajo y el equipamiento informático a un usuario.
- Cuando se entrega el equipamiento de trabajo especializado de TV a un grupo de trabajo el Jefe de Dpto. Técnico hace un reporte con las características de cada equipo entregado.
- Cuando se devuelve el equipamiento de trabajo especializado de TV, el Jefe Dpto. Técnico agrega al reporte el estado en que se devolvió cada equipo.
- Siempre que se produce algún material que sea de interés para la institución este se quema y se guarda en un DVD en el fondo del CGIA.
- Siempre que se reedite algún material Audiovisual, el mismo es quemado en un DVD y guardado en el fondo del CGIA.
- Diariamente se va a generar un reporte con las incidencias ocurridas durante el proceso de transmisión.

#### 3.2.3 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor.

Una vez identificados los procesos del negocio es necesario encontrar los actores que están involucrados en su realización.

Actores del negocio y su justificación:

Actores del Negocio	Justificación
---------------------	---------------

<p>Ciente</p>	<p>El cliente que puede ser la misma DTU, así como un representante de la FEU, Teleformación, Facultad, Dirección UCI, Consejo Universitario, es el que inicia la producción de una media y la solicitud de una media, es el que solicita y recibe los servicios brindados del cual se beneficia toda o parte de la comunidad universitaria.</p>
<p>Productor</p>	<p>El productor es el que inicia la Asignación de equipamiento de trabajo especializado de TV, el que realiza el Plan de Producción.</p>
<p>Jefe de Servicios Técnicos</p>	<p>El Jefe de Servicios Técnicos es el que inicia la transmisión de la programación, es el que solicita el servicio el cual beneficia a toda la comunidad universitaria.</p>
<p>Director(a) de la DTU.</p>	<p>El (la) Director(a) de la DTU es quien indica la construcción del plan de programación y de producción respectivamente de la semana.</p>

**Tabla 1 Descripción de los actores del negocio.**

### 3.2.4 Trabajadores del Negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

Trabajadores del negocio y su justificación:

Trabajadores	Justificación
Director DTU.	Es la que dirige y representa los intereses de la DTU.
Productor.	Es el encargado de realizar el plan de producción de la semana.
Jefe Dpto. Técnico.	Es el encargado de asignar el equipamiento especializado de TV y enviar a los transmisores lo que se va a transmitir.
Realizador.	Es el encargado de hacer el guión del material que se va a filmar.
Digitalizador.	Es el encargado de digitalizar los materiales.
Editor.	Es el encargado de editar los materiales.
Especialista Superior del CGIA.	Es el encargado de recibir las solicitudes de medias que son aprobadas por la dirección de la DTU y distribuir el trabajo correspondiente.
Grupo de trabajo.	Son los que en conjunto realizan la grabación de un material.
Técnico de servicios del CGIA.	Es el encargado de buscar y recuperar información audiovisual en el fondo del CGIA.
Líder de UCITeVe.	Es el encargado de dirigir el proyecto UCITeVe.
Estudiantes de UCITeVe	Son los encargados de llevar a cabo procesos de edición en materiales docentes.
Jefe Dpto. Programación	Es el encargado de conformar la programación de la semana.

Especialista del CGIA.	Es el encargado de analizar y procesar la información audiovisual.
Gestor del CGIA.	Es el encargado de gestionar las medias que son solicitadas a la DTU y no se encuentran en la misma.
Representante de institución afín.	Es el quien brinda la media solicitada por el gestor del CGIA.
Transmisor.	Es el encargado de transmitir la programación.

**Tabla 2 Descripción de los trabajadores del negocio**

### 3.2.5 Diagrama de Casos de Uso del Negocio

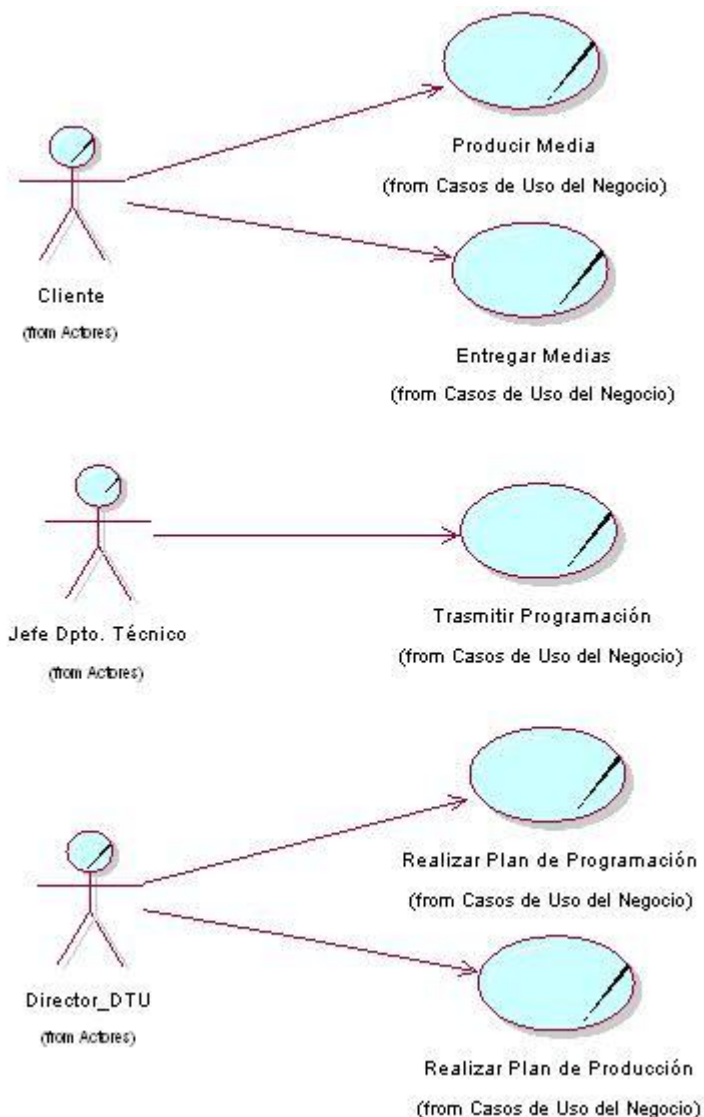


Fig. 4 Diagrama de Casos de Uso del Negocio

### 3.2.6 Descripción textual de los casos de Uso del Negocio.

#### 3.2.6.1 Caso de Uso "Producir Media"

<b>Nombre del Caso de Uso</b>	Producir Media.
<b>Actores</b>	Cliente.(Inicia)
<b>Propósito</b>	Realizar la producción de una media.
<b>Resumen</b>	El caso de uso se inicia cuando llega un cliente a la DTU y hace la solicitud para la producción de una

	media.
<b>Curso normal de los eventos</b>	
<b>Acciones del actor</b>	<b>Respuestas del proceso del negocio</b>
1. El Cliente llega a la DTU y llena la planilla de solicitud para la producción de una media.	2. La directora de la DTU informa al productor el material solicitado.
	3. El productor incluye esta solicitud en el plan de producción.
	4. El productor envía el plan de producción al Jefe Dpto. Técnico.
	5. El jefe de Dpto. Técnico recibe el plan de producción y revisa el equipamiento que se necesita.
	6. El Jefe de Dpto. Técnico asigna el equipamiento al grupo de trabajo que se encargara de dicho material y genera un reporte con los equipos entregados.
	7. El Grupo de trabajo realiza el proceso de filmación del material.
	8. El Grupo de trabajo una vez concluida la filmación devuelven el equipamiento que les fue entregado al Jefe de Dpto. Técnico.
	9. El Jefe de Dpto. Técnico revisa cada equipo devuelto y agrega al reporte el estado en que se devolvieron.
	10. El realizador lleva el material grabado al área de digitalización.
	11. El digitalizador digitaliza el material, y selecciona los planos a guardar y los envía al Especialista Superior del CGIA, y los planos para editar se envían a las PCs de edición.
	12. El editor con la anuencia del realizador lleva a cabo el montaje del audio y el video.
	13. El editor exporta el material en el formato correspondiente al destino del mismo.
	14. El editor entrega el material terminado al Cliente que lo solicito y si es de interés para la DTU lo envía al Especialista Superior del CGIA para almacenarlo en el fondo.
15. El Cliente recibe el material terminado.	
<b>Curso alternativo de los eventos</b>	
<b>Acciones del actor</b>	
<b>Prioridad</b>	Crítico

Tabla 3 Descripción textual del caso de uso "Producir Media"

## 3.2.6.2 Caso de Uso "Entregar Medias"

<b>Nombre del Caso de Uso</b>	Entregar Medias	
<b>Actores</b>	Cliente (Inicia)	
<b>Propósito</b>	Permitir que la DTU brinde la Información Audiovisual solicitada.	
<b>Resumen</b>	El caso de uso inicia cuando alguno de estos actores solicita a la DTU Información Audiovisual.	
<b>Curso normal de los eventos</b>		
<b>Acciones del actor</b>	<b>Respuestas del proceso del negocio</b>	
1. El Cliente solicita información audiovisual.	2. La directora de la DTU recepciona y evalúa la solicitud.	
	3. La directora de la DTU envía al Especialista Superior del CGIA la solicitud aprobada.	
	4. El Especialista Superior del CGIA informa el material audiovisual solicitado.	
	5. El Técnico de Servicios del CGIA busca y recupera la Información Audiovisual en el fondo del CGIA.	
	6. Si la Información Audiovisual está lista, el Técnico de Servicios del CGIA envía al Especialista Superior del CGIA la información Audiovisual solicitada.	
	7. El Especialista Superior del CGIA envía la Información Audiovisual recuperada al Cliente.	
8. El Cliente recibe la Información.		
<b>Curso alternativo de los eventos</b>		
<b>*a</b>		
<b>Acciones del actor</b>		
	6. Si la Información Audiovisual no esta lista (necesita reedición).	
	7. El Especialista Superior del CGIA valora a partir de la clasificación de la Información Audiovisual en dependencia de su tipología.	
	8. Si es Conferencia o Teleclase el Especialista Superior del CGIA envía al Líder de UCITeVe la Información Audiovisual a reeditar.	
	9. El estudiante de UCITeVe reedita, comprime la Información Audiovisual.	
	10. El estudiante de UCITeVe copia la Información Audiovisual editada en el servidor del CGIA.	
	11. El estudiante de UCITeVe envía confirmación al Especialista del CGIA.	
	12. El Especialista del CGIA analiza y procesa la Información Audiovisual.	
	13. El Especialista del CGIA quema y almacena	



	en DVD en el fondo del CGIA la Información Procesada.
	14. El Especialista del CGIA ofrece la Información Procesada al Cliente.
15. El Cliente recibe la Información Audiovisual.	
<b>*a.1</b>	
<b>Acciones del actor</b>	
	8. Si es otro tipo de material, el Especialista General del CGIA envía la Información Audiovisual a reeditar al Productor.
	9. El productor realiza el proceso de reedición y envía la Información Audiovisual al Especialista del CGIA.
	10. El Especialista del CGIA digitaliza si es necesario y lo copia para el Servidor del CGIA.
	11. Regresa al Paso 12 del Curso Alternativo de los eventos *a.
<b>*b</b>	
<b>Acciones del actor</b>	
	6. Si la Información Audiovisual no es encontrada se hace la <i>desiderata</i> y se envía al Desarrollo de Colecciones.
	7. El Gestor hace las solicitudes de Información Audiovisual al representante de una Institución Afín.
	8. El representante de la Institución Afín entrega al Gestor del CGIA la Información Audiovisual solicitada.
	9. Regresa al Paso 10 del Curso Alternativo de los eventos *a.1
<b>Prioridad</b>	<b>Crítico</b>

Tabla 4 Descripción textual del caso de uso "Entregar Medias"

## 3.2.6.3 Caso de Uso "Trasmitir Programación"

<b>Nombre del Caso de Uso</b>	Transmitir Programación
<b>Actores</b>	Jefe Dpto. Técnico.
<b>Propósito</b>	Transmitir los materiales programados en el plan de programación.
<b>Resumen</b>	El Caso de Uso se inicia cuando el Jefe de Servicios Técnicos envía a Transmisión el plan de programación, a partir de ahí se transmiten los materiales.
<b>Curso normal de los eventos</b>	
<b>Acciones del actor</b>	<b>Respuestas del proceso del negocio</b>

1. El Jefe Dpto. Técnico envía a transmisión el plan de programación y pone a disposición los materiales a transmitir.	2. El transmisor recibe el plan de programación y los materiales a transmitir y programa la transmisión.
	3. El transmisor genera un reporte con las incidencias en el proceso de transmisión.
	4. El transmisor envía el reporte al Jefe de Dpto. Técnico.
5. El Jefe de Dpto. Técnico recibe el reporte y archiva el reporte.	
<b>Curso alternativo de los eventos</b>	
<b>Acciones del actor</b>	
<b>Prioridad</b>	Crítico

Tabla 5 Descripción textual del caso de uso "Trasmitir Programación"

## 3.2.6.4 Caso de Uso "Realizar Plan de Programación."

<b>Nombre del Caso de Uso</b>	Realizar Plan de Programación.	
<b>Actores</b>	Director(a) DTU.(Inicia)	
<b>Propósito</b>	Realizar el plan de Programación de la semana.	
<b>Resumen</b>	El caso de uso se inicia cuando el (la) director(a) de la DTU solicita al departamento de programación la confección del plan de programación de la semana.	
<b>Curso normal de los eventos</b>		
<b>Acciones del actor</b>	<b>Respuestas del proceso del negocio</b>	
1. El (la) director(a) de la DTU indica al departamento de programación la confección del plan de programación.	2. El Jefe Dpto. Programación hace la propuesta de la programación de la semana.	
	3. El Jefe Dpto. Programación se reúne en trabajo de mesa con la FEU, Vicerrector de Extensión Universitaria donde ahí queda conformada y aprobada la programación de la semana.	
	4. El Jefe Dpto. Programación envía el plan de programación al Jefe Dpto. Técnico.	
	5. El Jefe Dpto. Programación envía confirmación al director(a) de la DTU que el plan de programación esta conformado.	
6. El (la) director(a) de la DTU recibe la confirmación.		
<b>Curso alternativo de los eventos</b>		
<b>Acciones del actor</b>		

<b>Prioridad</b>	Crítico
------------------	---------

Tabla 6 Descripción textual del caso de uso "Realizar Plan de Programación"

## 3.2.6.5 Caso de Uso "Realizar Plan de Producción."

<b>Nombre del Caso de Uso</b>	Realizar Plan de Producción.	
<b>Actores</b>	Director(a) DTU.(Inicia)	
<b>Propósito</b>	Realizar el plan de Producción de la semana.	
<b>Resumen</b>	El caso de uso se inicia cuando el (la) director(a) de la DTU solicita al departamento de producción la confección del plan de producción de la semana según las solicitudes de producción recibidas.	
<b>Curso normal de los eventos</b>		
<b>Acciones del actor</b>	<b>Respuestas del proceso del negocio</b>	
1. El (la) director(a) de la DTU indica al departamento de producción la confección del plan de producción.	2. El productor revisa las solicitudes de producción recibidas y realiza el plan de producción de la semana.	
	3. El productor envía el plan de producción a todo el personal de su departamento y al Jefe Dpto. Técnico.	
	4. El productor envía confirmación al director(a) de la DTU que el plan de programación fue confeccionado.	
5. Recibe la confirmación.		
<b>Curso alternativo de los eventos</b>		
<b>Acciones del actor</b>		
<b>Prioridad</b>	Crítico	

Tabla 7 Descripción textual del caso de uso "Realizar Plan de Producción"

## 3.3 Requerimientos Funcionales.

## R1 Gestionar Usuario

R 1.1 Adicionar usuario.

R 1.2 Modificar usuario.

R 1.3 Eliminar usuario.

## 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.

**R3 Autenticar usuario.**

**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 Visualizar plan de Producción.

**R5 Gestionar equipamiento.**

R 5.1 Adicionar equipamiento.

R 5.2 Modificar datos equipamiento.

R 5.3 Baja Técnica equipamiento

**R6 Permitir asignar equipamiento informático a un usuario.**

R 6.1 Modificar Asignación.

R 6.2 Eliminar Asignación.

**R7 Mostrar reporte de equipamiento.**

**R8 Asignar equipamiento de trabajo especializado de TV.**

**R9 Mostrar reporte del equipamiento en trabajo.**

**R10 Elaborar reporte de incidencias en transmisión.**

**R11 Gestionar Medias.**

R 11.1 Adicionar Medias.

R 11.2 Modificar Medias.

R 11.3 Eliminar Medias.

**R12 Gestionar Plan de Programación.**

R 12.1 Adicionar plan de programación.

R 12.2 Modificar plan de programación.

R 12.3 Eliminar plan de programación.

R 12.4 Visualizar plan de programación.

**R13 Mostrar Reporte de un Material.**

**R14 Permitir hacer solicitud de Media.**

**R15 Permitir hacer solicitud de Producción.**

**R16 Mostrar Reporte de Transmisión.**

#### **R17 Permitir Imprimir los Reportes.**

#### **3.4 Requerimientos No Funcionales**

##### **Requisitos de Interfaz**

La Aplicación dispondrá de una elevada integración con un Portal Web para la gestión de procesos en la DTU, homogeneizando el diseño con el propósito de facilitar la comprensión y el acceso a informaciones importantes y orientativas sobre los diferentes procesos que aquí se generan.

La interfaz de usuario del Sistema, será amigable a los usuarios finales, cumpliendo con los requisitos siguientes:

- Las ventanas del Sistema contendrán los datos de forma clara y bien estructurada, y al mismo tiempo permitirán la interpretación correcta e inequívoca de la información.
- El diseño de la interfaz de usuario del Sistema buscará la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
- El Sistema usará una norma que permita la distinción visual entre los elementos de la ventana a través del uso de colores, así como otras técnicas.
- Su funcionamiento será intuitivo, y requerirá de información mínima.

##### **Requisitos de Usabilidad**

- El sistema deberá trabajar de manera automática. En caso de ser necesario, podrá ser administrado de manera remota desde las regiones o dirección general, siempre que se garantice la seguridad necesaria.
- Permitir deberá permitir la conexión concurrente, por sus características de software, desde todos los puestos de la dirección de televisión. Posibilitar el incremento de usuarios concurrentes.
- El sistema deberá validar que la información introducida sea correcta. Además debe validarse que el tipo de dato se corresponda con lo previsto para el campo.

- El sistema deberá verificar, de existir la conectividad y las garantías de acceso, los datos introducidos contra los datos del dominio UCI y los datos de la base de datos del propio sistema del usuario que se esta registrando.
- El sistema debe proveer un servicio Web que permita a otros sistemas, en desarrollo o por desarrollar, interactuar con él.

#### **Requisitos de Rendimiento**

El sistema debe tener una similitud en sus páginas y estar poco cargado, posibilitando que el sistema devuelva las respuestas de una manera eficiente, siendo más sencillo de entender y usar por el usuario.

#### **Requisitos de Soporte**

Para el desarrollo de este sistema se pretende realizar a través de servicios Web, lo cual le propicia una alta portabilidad y uso desde distintas plataformas.

#### **Requisitos de Portabilidad**

Permitir que el sistema se ejecute sobre el Sistema Operativo Linux, Windows 98 o superior.

#### **Requisitos de Seguridad**

Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoria (AAA).

- **Confiability:** La información manejada por el sistema está protegida de acceso no autorizado. El sistema debe prevenir posibles fallos y/o errores y presentar facilidades para una rápida recuperación en dichos casos.
- **Integridad:** Que la información sea modificada (incluyendo su creación y borrado) sólo por personal autorizado. Se permitirá la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.
- **Disponibilidad:** Los usuarios autorizados tendrán acceso a la información en todo momento, se debe lograr balancear la carga de acceso entre múltiples servidores, disminuyendo los tiempos de respuesta.

#### **Requisitos de Software del Sistema**

Las computadoras donde se instalara el software deben tener instalado:

- a. Windows XP Profesional ó GNU/Linux.

- b. Navegador Internet Explorer (IE), Mozilla.

La PC que alojará la aplicación y la Base de Datos debe tener instalado:

- a. Un servidor Apache con la versión 5 de PHP
- b. Symfony como framework
- c. PostgreSQL 8.2 como SGBD

### Requisitos de Hardware del sistema

Las computadoras que utilizarán el software deberán tener 256 Ram, Pentium 4 o superior con un mínimo de espacio de disco duro de 10 Gb.

## 3.5 Descripción del Sistema Propuesto.

### 3.5.1 Descripción de los actores del sistema

Actores del Sistema	Justificación
Administrador	Es el que va a administrar la aplicación.
Productor	Es el que confecciona semanalmente el plan de Producción.
Técnico de Sistema	Es el que gestiona la información de los materiales tanto informáticos como especializados de TV dentro de la dirección, además es quien gestiona los reportes correspondientes a dichos materiales.
Transmisor	Es el que transmite la programación y gestiona el reporte de transmisión.
Especialista	Es el encargado de gestionar los datos de las medias pertenecientes al fondo del CGIA.
Informático	Es quién recoge las solicitudes de Medias y de Producción.

Jefe Dpto. Prog	Es el encargado de confeccionar el Plan de Programación.
-----------------	--

Tabla 8 Descripción de los actores del sistema

3.5.2 Casos de Uso del Sistema.

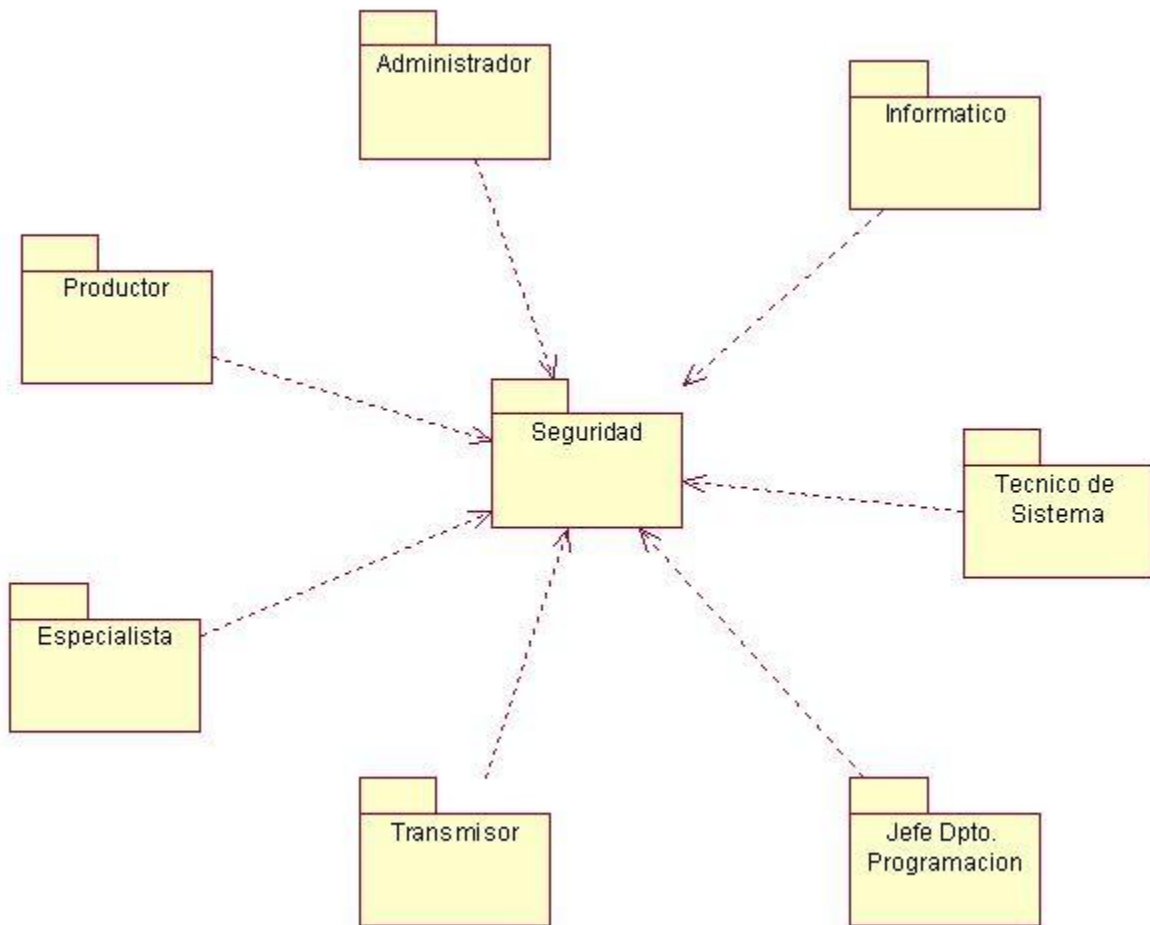


Fig. 5 Paquetes del Modelo de Casos de Usos del Sistema



3.5.2.1 Paquete Seguridad.

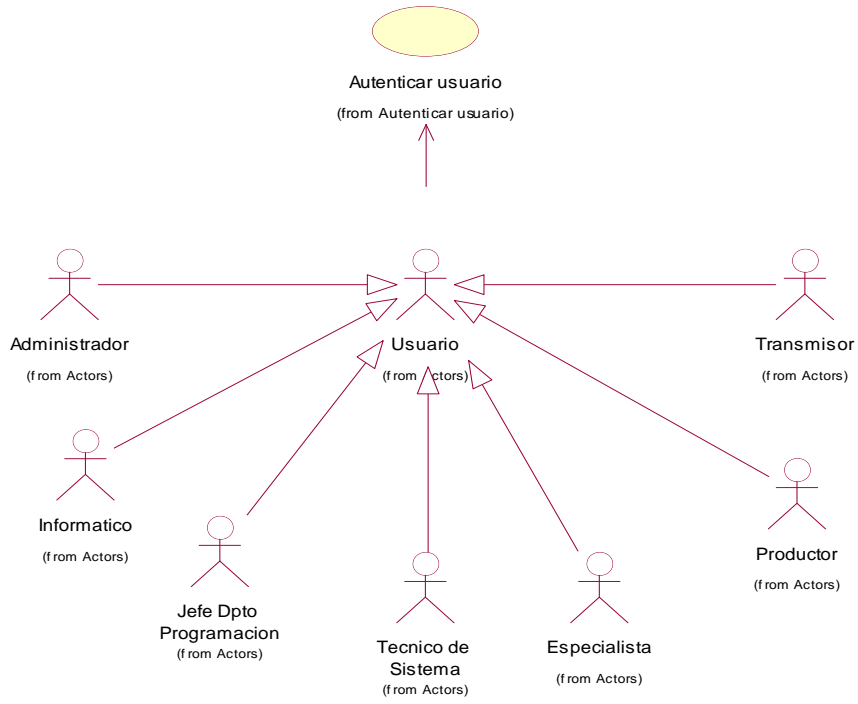


Fig. 6 Caso de uso asociado al Paquete de Seguridad del Sistema

3.5.2.2 Paquete Administrador

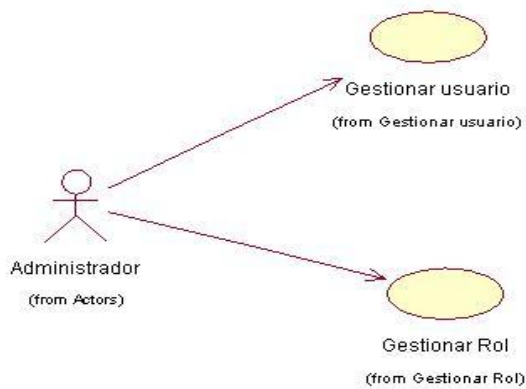


Fig. 7 Casos de uso asociados al Paquete Administrador del Sistema

### 3.5.2.3 Paquete Informático

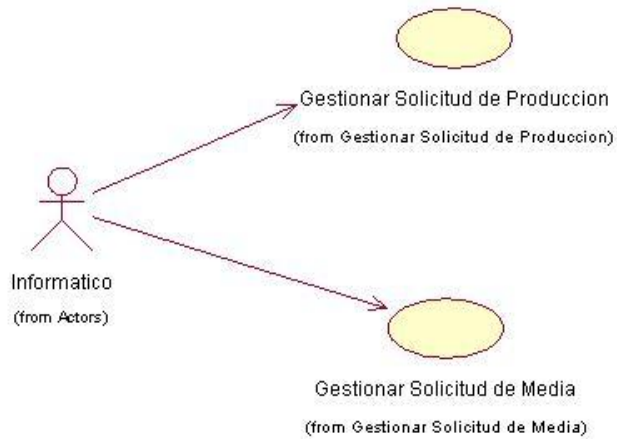


Fig. 8 Casos de uso asociados al Paquete Informático del Sistema

### 3.5.2.4 Paquete Jefe Dpto. Programación

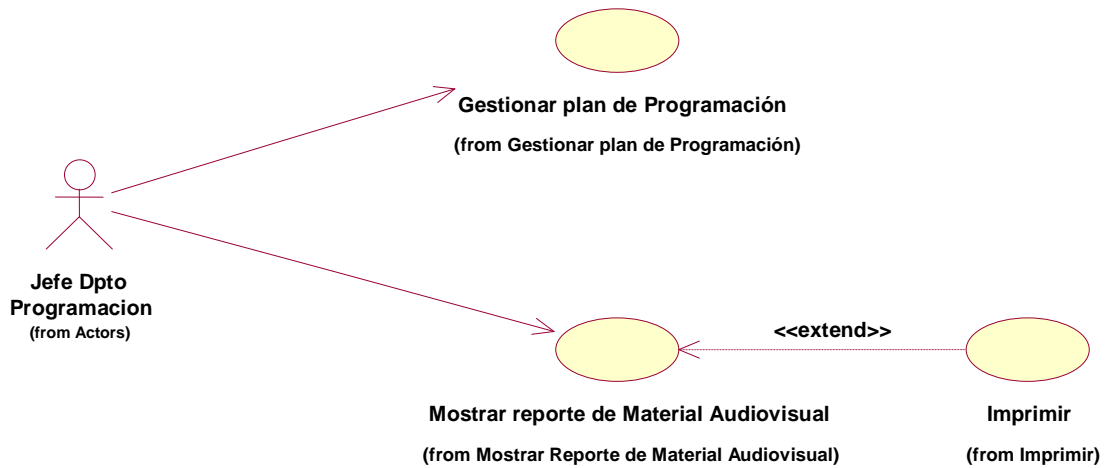


Fig. 9 Casos de uso asociados al Paquete Jefe Dpto. Programación del Sistema

3.5.2.5 Paquete Técnico de Sistema

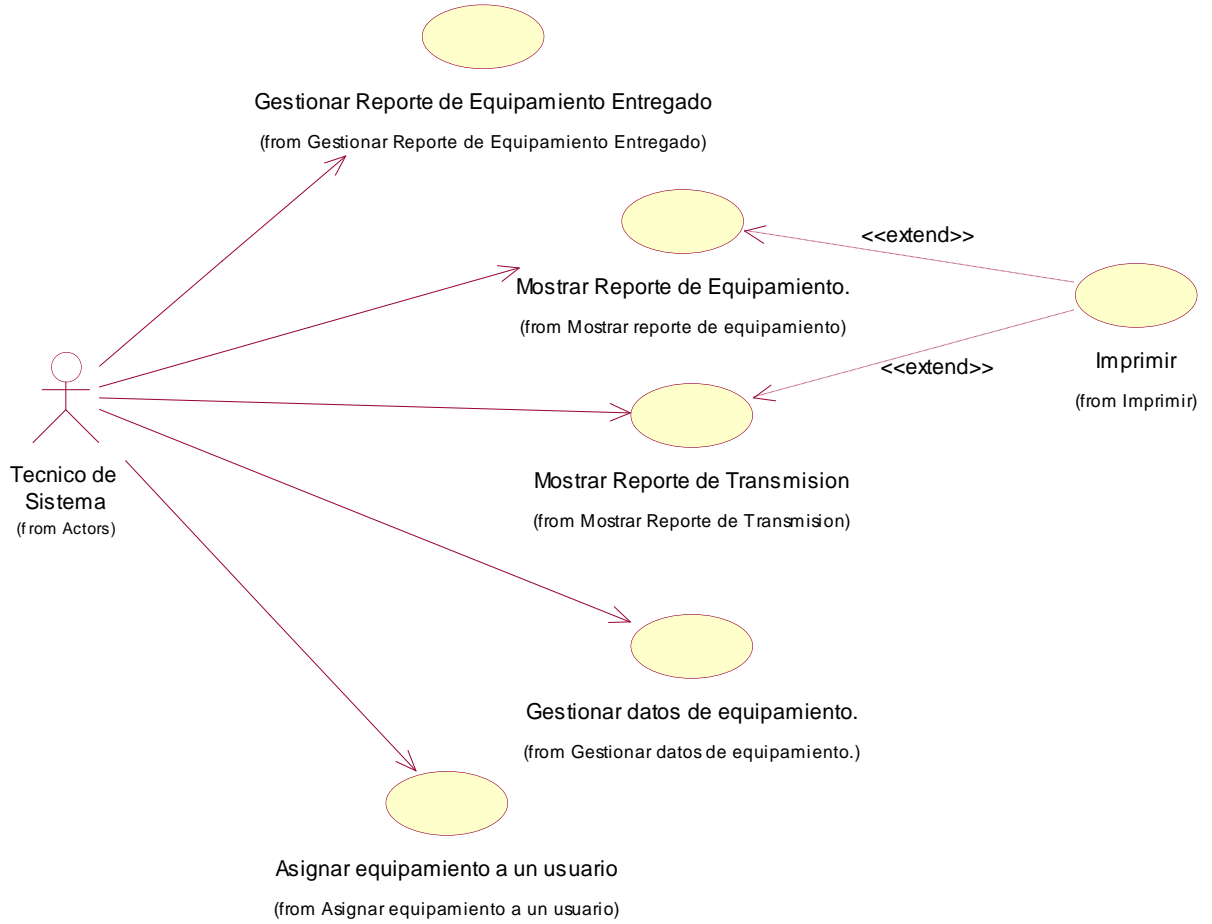


Fig. 10 Casos de uso asociados al Paquete Técnico de Sistema del Sistema

### 3.5.2.6 Paquete Especialista



Fig. 11 Caso de uso asociado al Paquete Especialista del Sistema

### 3.5.2.7 Paquete Productor



Fig. 12 Caso de uso asociado al Paquete Productor del Sistema

### 3.5.2.8 Paquete Transmisor



Fig. 13 Caso de uso asociado al Paquete Transmisor del Sistema

## 3.5.3 Descripción Textual de los Casos de Uso del Sistema.

## 3.5.3.1 Caso de Uso "Autenticar Usuarios"

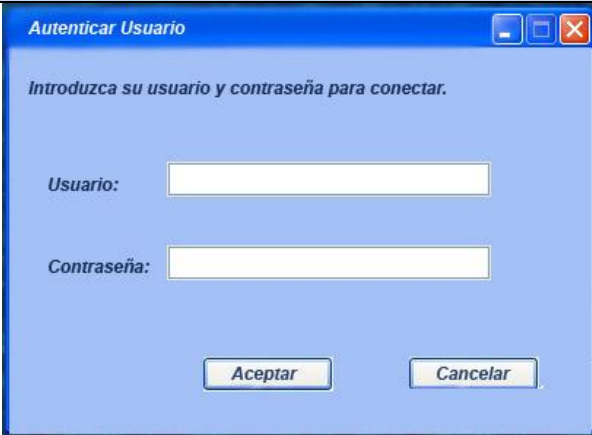
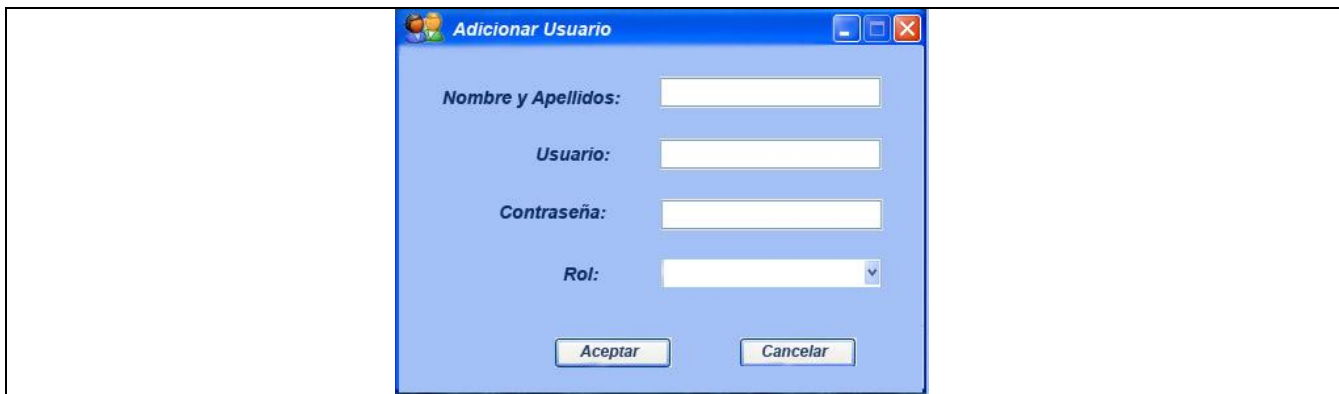
<b>Nombre del Caso de Uso</b>	Autenticar Usuarios.	
<b>Actores</b>	Usuario (Inicia).	
<b>Propósito</b>	Permitir a los usuarios autenticarse en el sistema.	
<b>Resumen</b>	El caso de uso comienza cuando el usuario accede al sistema para poder navegar en las páginas a las cuales tiene permiso, mostrándole el sistema la opción de autenticarse.	
<b>Referencias</b>	R3	
<b>Precondiciones</b>	El usuario debe estar registrado en el sistema.	
<b>Pos condiciones</b>	Para las siguientes acciones: Autenticar: El usuario esta autenticado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario accede al sistema para entrar en él.	2. El sistema muestra una interfaz para entrar los datos de autenticación	
3. El usuario entra los datos para autenticarse.	4. El sistema valida que los datos entrados sean correctos	
	5. El sistema muestra una ventana de bienvenida al usuario.	
<b>Curso alternativo de los eventos</b>		
5. Emite un mensaje indicando que alguno de los datos que el usuario entro están incorrectos.		
<b>Prototipo</b>		
		
<b>Prioridad:</b>	Crítico	

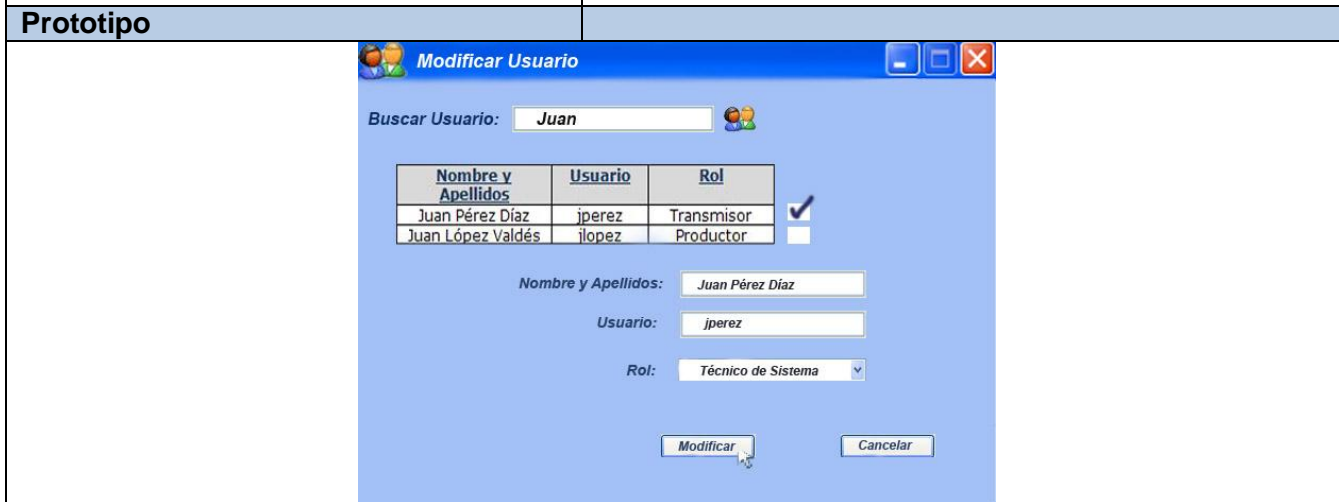
Tabla 9 Descripción detallada del Caso de Uso del sistema "Autenticar Usuarios"

## 3.5.3.2 Caso de Uso "Gestionar Usuarios"

<b>Nombre del Caso de Uso</b>	Gestionar Usuarios.
<b>Actores</b>	Administrador (Inicia).
<b>Propósito</b>	Permitir al administrador gestionar los usuarios del sistema.
<b>Resumen</b>	El caso de uso comienza cuando el administrador selecciona la opción Gestionar Usuarios, y el sistema le muestra un conjunto de acciones que puede realizar sobre el usuario, de estas acciones puede seleccionar adicionar, modificar o eliminar usuarios.
<b>Referencias</b>	R1
<b>Precondiciones</b>	El administrador debe haberse autenticado en el sistema. Debe existir al menos un rol.
<b>Pos condiciones</b>	Para las siguientes acciones: Adicionar: Se adiciona un nuevo usuario en el sistema. Modificar: Se modifica un usuario del sistema. Eliminar: Se elimina un usuario del sistema.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona adicionar, modificar o eliminar usuarios.	2. El sistema ejecuta alguna de las siguientes opciones: a) Si selecciona la opción de adicionar usuario ir a la sección de "Adicionar Usuario". b) Si selecciona la opción de modificar usuario ir a la sección de "Modificar Usuarios". c) Si selecciona la opción de eliminar usuario ir a la sección de "Eliminar Usuario".
<b>Sección "Adicionar Usuario"</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
3. El administrador entra los datos necesarios para adicionar un nuevo usuario y especifica el rol asignado.	4. El sistema valida los datos y muestra el resultado de la validación.
5. El administrador pulsa el botón "Adicionar" para adicionar un usuario al sistema.	6. Se inserta un nuevo usuario en el sistema.
<b>Curso alternativo de los eventos</b>	
	4 Emite un mensaje indicando que el usuario no existe en el dominio.
<b>Prototipo</b>	



Sección "Modificar Usuario"	
Acciones del Actor	Respuesta del sistema
3. El Administrador entra los datos para buscar el usuario que desea modificar.	4. El sistema muestra una interface donde aparece el usuario y la opción de modificar.
5. El Administrador entra los nuevos datos y pulsa el botón "Modificar".	6. El sistema modifica los datos del usuario y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.



Sección "Eliminar Usuario"	
Acciones del Actor	Respuesta del sistema
3. El Administrador entra los datos para buscar el usuario que desea eliminar.	4. El sistema muestra una interface donde aparece el usuario y la opción de eliminar.
5. El administrador pulsa el botón "Eliminar".	6. El sistema elimina al usuario y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

Curso alternativo de los eventos	
	6. El sistema muestra un mensaje indicando que el usuario no existe dentro del sistema.

**Prototipo**




Tabla 10 Descripción detallada del caso de Uso del sistema "Gestionar Usuario".

3.5.3.3 Caso de Uso "Gestionar Plan de Producción"

Nombre del Caso de Uso	<b>Gestionar Plan de Producción.</b>	
Actores	Productor (Inicia).	
Propósito	Permitir al productor gestionar el plan de producción.	
Resumen	Comienza cuando el productor selecciona la opción "Plan de Producción", mostrándole el sistema una interface con las opciones para gestionar el plan de producción.	
Referencias	R4	
Precondiciones	El productor debe haberse autenticado en el sistema.	
Pos condiciones	Para las siguientes acciones: Adicionar: Se adiciona un nuevo plan de producción en el sistema. Modificar: Se modifica un nuevo plan de producción del sistema. Eliminar: Se elimina un nuevo plan de producción del sistema.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El productor selecciona la opción "Plan de Producción".	2. El sistema muestra una interfaz donde brinda la posibilidad de crear, modificar o eliminar el plan de producción.	
3. El productor solicita crear, modificar o eliminar el plan de producción.	4. El sistema ejecuta alguna de las siguientes acciones: a) Si se selecciona la opción de crear PProd ir a la sección "Crear PProd". b) Si se selecciona la opción de modificar PProd ir a la sección "Modificar PProd". c) Si se selecciona la opción de eliminar PProd ir a la sección "Eliminar PProd".	
<b>Sección "Crear PProd"</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
5. El productor entra los datos necesarios para	6. El sistema valida los datos insertados y crea	



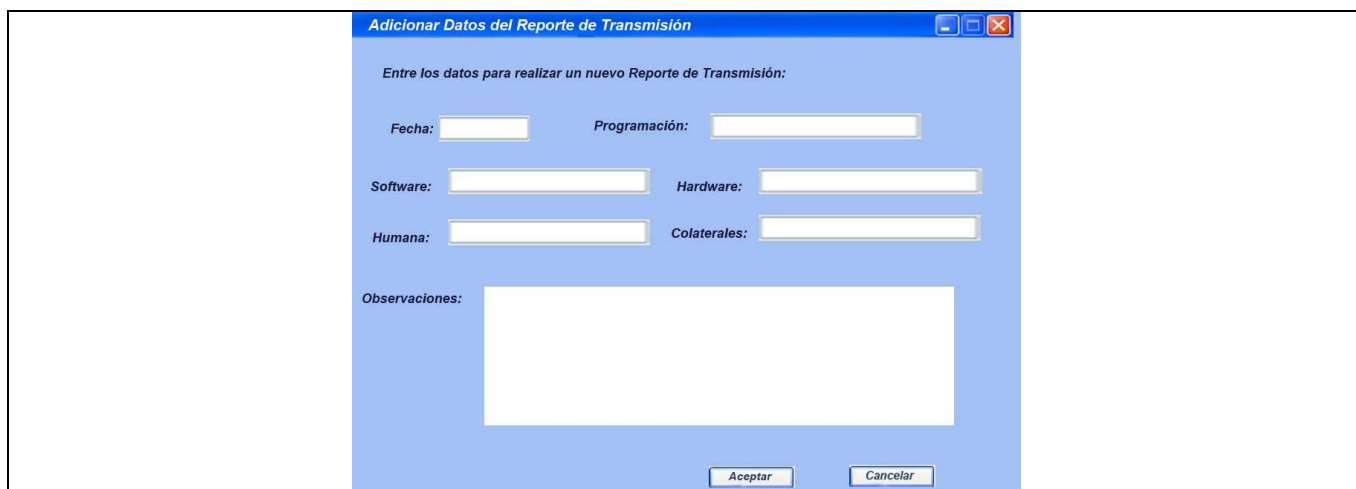
<p>crear el PProd.</p>	<p>el PProd almacenando los datos en la base de datos. 7. Posteriormente el sistema muestra un mensaje indicando que el PProd se ha creado con éxito.</p>
<p>Curso alternativo de los eventos</p>	
	<p>6. El sistema muestra un mensaje indicando llenar los campos obligatorios correctamente.</p>
<p><b>Prototipo</b></p>	
	
<p><b>Sección "Modificar PProd"</b></p>	
<p><b>Acciones del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>5. El productor entra los datos para buscar el PProd que desea modificar.</p>	<p>6. El sistema muestra una interface donde aparece el PProd y la opción de modificar.</p>
<p>7. El productor entra los nuevos datos y pulsa el botón "Modificar".</p>	<p>8. El sistema modifica los datos del PProd y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.</p>
<p>Curso alternativo de los eventos</p>	
	<p>6. El sistema muestra un mensaje indicando que los campos obligatorios deben ser llenados correctamente.</p>
<p><b>Prototipo</b></p>	

<b>Sección "Eliminar PProd"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
5. El productor entra los datos para buscar el PProd que desea eliminar.	6. El sistema muestra una interface donde aparece el PProd y la opción de eliminar.
7. El productor pulsa el botón "Eliminar".	8. El sistema elimina el PProd y muestra un mensaje indicando que el PProd ha sido eliminado.
<b>Curso alternativo de los eventos</b>	
	8. El sistema muestra un mensaje indicando que el PProd no existe dentro del sistema.
<b>Prototipo</b>	
<b>Prioridad:</b>	<b>Crítico</b>

Tabla 11 Descripción detallada del caso de Uso del sistema "Gestionar Plan de Producción".

## 3.5.3.4 Caso de Uso "Gestionar Reporte de Transmisión"

<b>Nombre del Caso de Uso</b>	Gestionar Reporte de Transmisión.	
<b>Actores</b>	Transmisor (Inicia).	
<b>Propósito</b>	Permitir al transmisor gestionar el reporte de transmisión.	
<b>Resumen</b>	Comienza cuando el transmisor selecciona la opción "Reporte de Transmisión", mostrándole el sistema una interface con las opciones para su gestión.	
<b>Referencias</b>	R10	
<b>Precondiciones</b>	El trasmisor debe haberse autenticado en el sistema.	
<b>Pos condiciones</b>	Para las siguientes acciones: Crear: Se crea un nuevo Reporte de Transmisión en el sistema. Modificar: Se modifica un nuevo Reporte de Transmisión del sistema. Eliminar: Se elimina un Reporte de Transmisión del sistema. Visualizar: Se visualiza un Reporte de Transmisión	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El transmisor selecciona la opción " Reporte de Transmisión ".	2. El sistema muestra una interfaz donde brinda la posibilidad de crear, modificar, eliminar o visualizar el Reporte de Transmisión.	
3. El transmisor solicita crear, modificar, eliminar o visualizar el Reporte de Transmisión.	4. El sistema ejecuta alguna de las siguientes acciones: a) Si se selecciona la opción de crear RT ir a la sección "Crear RT". b) Si se selecciona la opción de modificar RT ir a la sección "Modificar RT". c) Si se selecciona la opción de visualizar RT ir a la sección "Visualizar RT". d) Si se selecciona la opción de eliminar RT ir a la sección "Eliminar RT".	
<b>Sección "Crear RT"</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
5. El transmisor entra los datos necesarios para crear el RT.	6. El sistema valida los datos insertados y crea el RT almacenándolo en la base de datos. 7. Posteriormente el sistema muestra un mensaje indicando que el RT se ha creado con éxito.	
<b>Curso alternativo de los eventos</b>		
	6. El sistema muestra un mensaje indicando llenar los campos obligatorios correctamente.	
<b>Prototipo</b>		



**Sección "Visualizar RT"**

5. El transmisor entra los datos para buscar el REA que desea visualizar y pulsa el botón "Buscar".

5. El sistema muestra la interfaz que visualiza el último RT con todos sus datos.

**Prototipo**



**Sección "Modificar RT"**

**Acciones del Actor**

5. El transmisor entra los datos para buscar el RT que desea modificar.

7. El transmisor entra los nuevos datos y pulsa el botón "Modificar".

**Respuesta del Sistema**

6. El sistema muestra una interface donde aparece el RT y la opción de modificar.

8. El sistema modifica los datos del RT y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

**Curso alternativo de los eventos**

6. El sistema muestra un mensaje indicando que los campos obligatorios deben ser llenados correctamente.

**Prototipo**

<b>Sección "Eliminar RT"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	5. El sistema muestra la interface para eliminar RT.
6. El transmisor pulsa el botón "Eliminar RT".	7. El sistema elimina el RT y muestra un mensaje indicando que el RT ha sido eliminada.
<b>Curso alternativo de los eventos.</b>	
<b>Prototipo</b>	
<b>Prioridad:</b>	<b>Crítico</b>

Tabla 12 Descripción detallada del caso de Uso del sistema "Gestionar Reporte de Transmisión"

## 3.5.3.5 Caso de Uso "Gestionar Datos de Media"

<b>Nombre del Caso de Uso</b>	Gestionar Datos de Media.
<b>Actores</b>	Especialista (Inicia).
<b>Propósito</b>	Permitir al Especialista gestionar una media.
<b>Resumen</b>	El caso de uso comienza cuando el Especialista selecciona la opción de gestionar media, y el sistema le muestra un conjunto de acciones que puede realizar sobre una media, de estas acciones puede seleccionar adicionar, modificar o eliminar los datos de una media.
<b>Referencias</b>	R11
<b>Precondiciones</b>	El Especialista debe haberse autenticado en el sistema.
<b>Pos condiciones</b>	Para las siguientes acciones: Adicionar: Se adiciona unos nuevos Datos de Media en el sistema. Modificar: Se modifica unos Datos de Media del sistema. Eliminar: Se elimina unos Datos de Media del sistema.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Especialista selecciona el tipo de media que desea Gestionar.	2. El sistema le muestra una interfaz con las posibles acciones a realizar sobre la media.
3. El Especialista selecciona adicionar, modificar o eliminar una media.	4. El sistema ejecuta alguna de las siguientes opciones: a) Si selecciona la opción de adicionar media ir a la sección de "Adicionar Media". b) Si selecciona la opción de modificar media ir a la sección de "Modificar Media". c) Si selecciona la opción de eliminar media ir a la sección de "Eliminar Media".
<b>Sección "Adicionar Media"</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
5. El Especialista entra los datos necesarios para adicionar una nueva media.	6. El sistema valida los datos y los almacena en la base de datos
7. El Especialista pulsa el botón "Adicionar" para adicionar una media al sistema.	8. El sistema muestra un mensaje indicando que la media fue adicionada correctamente.
<b>Curso alternativo de los eventos</b>	
	6 Emite un mensaje indicando que la media ya existe en la Base de Datos.
<b>Prototipo</b>	

Sección "Modificar Media"	
Acciones del Actor	Respuesta del sistema
5. El Especialista entra los datos para buscar la media que desea modificar los datos.	6. El sistema muestra una interface donde aparece los datos de la media y la opción de modificar.
7. El Especialista entra los nuevos datos y pulsa el botón "Modificar".	8. El sistema modifica los datos de la media y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

**Prototipo**

Sección "Eliminar Media"	
Acciones del Actor	Respuesta del sistema
4. El Especialista entra los datos para buscar la media que desea eliminar.	5. El sistema muestra una interface donde aparece la media y la opción de eliminar.
6. El Especialista elimina la media buscada.	7. El sistema elimina la media y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.
Curso Alternativo de los eventos	
	7. El sistema muestra un mensaje indicando que la media no existe en la Base de Datos.
Prototipo	



Tabla 13 Descripción detallada del caso de Uso del sistema "Gestionar Media"

## 3.5.3.6 Caso de Uso "Gestionar Plan de Programación"

<b>Nombre del Caso de Uso</b>	Gestionar Plan de Programación.
<b>Actores</b>	Jefe Dpto. Programación (Inicia).
<b>Propósito</b>	Permitir al Jefe Dpto. Programación gestionar el plan de programación.
<b>Resumen</b>	Comienza cuando el Jefe Dpto. Programación selecciona la opción "Plan de Programación", mostrándole el sistema una interface con las opciones para su gestión.
<b>Referencias</b>	R12
<b>Precondiciones</b>	El Jefe Dpto. Programación debe haberse autenticado en el sistema.
<b>Pos condiciones</b>	Para las siguientes acciones: Crear: Se adiciona un nuevos Plan de Programación en el sistema. Modificar: Se modifica un Plan de Programación del sistema. Eliminar: Se elimina un Plan de Programación del sistema. Visualizar: Se visualiza un Plan de Programación del sistema.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Jefe Dpto. Programación selecciona la opción "Plan de Programación".	2. El sistema muestra una interfaz donde brinda la posibilidad de crear, modificar, eliminar o visualizar el plan de programación.
3. El Jefe Dpto. Programación solicita crear, modificar, eliminar o visualizar el plan de programación.	4. El sistema ejecuta alguna de las siguientes acciones: a) Si se selecciona la opción de crear PProg ir a la sección "Crear PProg". b) Si se selecciona la opción de modificar PProg ir a la sección "Modificar PProg". c) Si se selecciona la opción de visualizar PProg ir a la sección "Visualizar PProg".



	d) Si se selecciona la opción de eliminar PProg ir a la sección "Eliminar PProg".
--	---

**Sección "Crear PProg"**

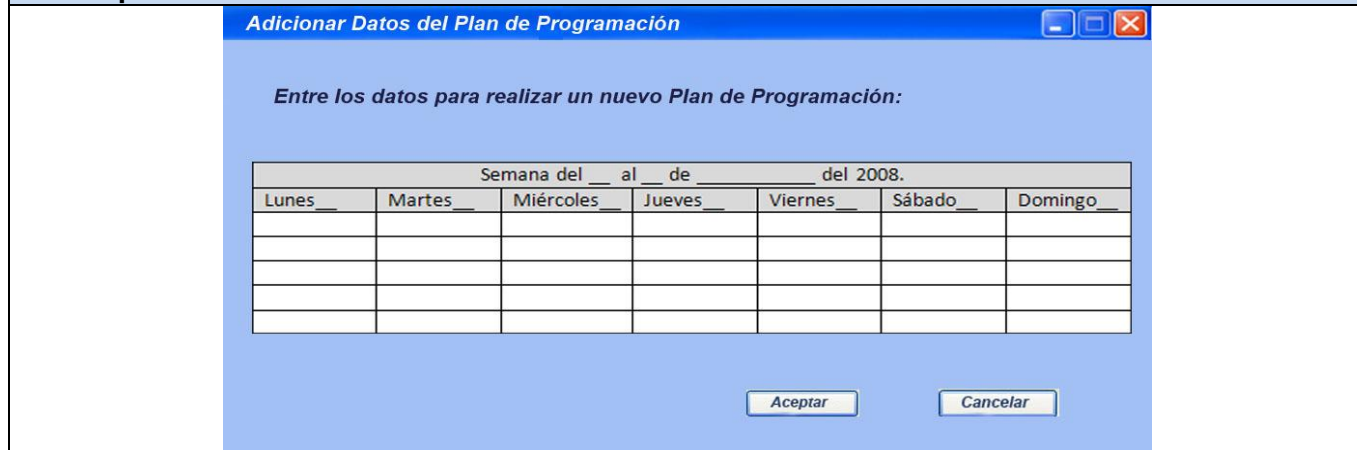
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
---------------------------	------------------------------

5. El Jefe Dpto. Programación entra los datos necesarios para crear el PProg.	6. El sistema valida los datos insertados y crea el PProg almacenando los datos en la base de datos. 7. Posteriormente el sistema muestra un mensaje indicando que el PProg se ha creado con éxito.
---	--

**Curso alternativo de los eventos**

	6. El sistema muestra un mensaje indicando llenar los campos obligatorios correctamente.
--	--

**Prototipo**



**Sección "Visualizar PProg"**

	5. El sistema muestra la interfaz que visualiza el último PProg con todos sus datos.
--	--

**Prototipo**



**Sección "Modificar PProg"**

<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
---------------------------	------------------------------

### 3: "PRESENTACION DE LA SOLUCION PROPUESTA"

5. El Jefe Dpto. Programación entra los datos para buscar el PProg que desea modificar los datos.	6. El sistema muestra una interface donde aparece los datos del PProg y la opción de modificar.
7. El Jefe Dpto. Programación entra los nuevos datos y pulsa el botón "Modificar".	8. El sistema modifica los datos del PProg y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

#### Curso alternativo de los eventos

	6. El sistema muestra un mensaje indicando que los campos obligatorios deben ser llenados correctamente.
--	--

#### Prototipo

#### Sección "Eliminar PProg"

Acciones del Actor	Respuesta del Sistema
5. El Jefe Dpto. Programación entra los datos para buscar el PProg que desea eliminar.	6. El sistema muestra una interface donde aparece el PProg y la opción de eliminar.
7. El Jefe Dpto. Programación elimina el PProg buscado.	8. El sistema elimina el PProg y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

#### Curso alternativo de los eventos

	6. El sistema muestra un mensaje indicando que el PProg no existe en la Base de Datos.
--	--

#### Prototipo

Semana del 7 al 14 de marzo de 2008						
Lunes 7	Martes 8	Miércoles 9	Jueves 10	Viernes 11	Sábado 12	Domingo 13
Gray Anat.	Musical	Héroes	Musical	Animados	Musical	Documental
Shrek	Billy Elliot	El Rey y Yo	Armagedón	Constantine	Pretty Woman	El cocodrilo
Documental	Animados	Documental	Animados	Musical	Héroes	Musical


Eliminar      Cancelar

Prioridad: Crítico

Tabla 14 Descripción detallada del caso de Uso del sistema "Gestionar Plan de Programación"

## 3.5.3.7 Caso de Uso "Gestionar Datos de equipamiento"

<b>Nombre del Caso de Uso</b>	Gestionar Datos de equipamiento.
<b>Actores</b>	Técnico de Sistema (Inicia).
<b>Propósito</b>	Permitir al Técnico de Sistema gestionar los datos de un equipamiento.
<b>Resumen</b>	El caso de uso comienza cuando el Técnico de Sistema selecciona la opción "Gestionar datos de equipamiento", y el sistema le muestra una interfaz donde el Técnico de Sistema va a seleccionar el tipo de equipamiento que desea gestionar.
<b>Referencias</b>	R5
<b>Precondiciones</b>	El Técnico de Sistema debe haberse autenticado en el sistema.
<b>Pos condiciones</b>	Para las siguientes acciones: Adicionar: Se adicionan los datos de un nuevo equipamiento. Modificar: Se modifican los datos de un equipamiento. Eliminar: Se eliminan los datos de un equipamiento. Dar Baja: Se le da baja a un equipamiento.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Técnico de Sistema selecciona el tipo de equipamiento que desea gestionar.	2. El sistema muestra una interfaz donde el Técnico de Sistema puede seleccionar las opciones Adicionar, Modificar, Eliminar o dar Baja a un equipamiento.
1. El Técnico de Sistema selecciona adicionar, modificar, eliminar o dar baja a un equipamiento.	2. El sistema ejecuta alguna de las siguientes opciones: a) Si selecciona la opción de adicionar equipamiento ir a la sección de "Adicionar Datos Equipamiento".

	<p>b) Si selecciona la opción de modificar equipamiento o dar baja a un equipamiento ir a la sección de "Modificar Datos Equipamiento".</p> <p>c) Si selecciona la opción de eliminar equipamiento ir a la sección de "Eliminar Datos Equipamiento".</p>
<b>Sección "Adicionar Datos Equipamiento".</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
5. El Técnico de Sistema entra los datos necesarios para adicionar los datos de un nuevo equipamiento.	6. El sistema valida los datos y muestra el resultado de la validación.
7. El Técnico de Sistema pulsa el botón "Adicionar" para adicionar los datos de un equipamiento al sistema.	8. Se insertan los datos de un nuevo equipamiento en el sistema.
<b>Curso alternativo de los eventos</b>	
	6 Emite un mensaje indicando que los datos no son correctos.
<b>Prototipo</b>	
	
<b>Sección "Modificar Datos Equipamiento".</b>	
<b>Acciones del Actor</b>	<b>Respuesta del sistema</b>
5. El Técnico de Sistema entra los datos para buscar los datos del equipamiento que desea modificar.	6. El sistema muestra una interface donde aparecen los datos del equipamiento y la opción de modificar datos.
7. El Técnico de Sistema entra los nuevos datos y pulsa el botón "Modificar Datos".	8. El sistema modifica los datos del equipamiento y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.
<b>Prototipo</b>	

**Sección "Eliminar Datos Equipamiento".**

Acciones del Actor	Respuesta del sistema
5. El Técnico de Sistema entra los datos para buscar el equipamiento que desea eliminar.	6. El sistema muestra una interface donde aparecen los datos del equipamiento y la opción de eliminar.
7. El Técnico de Sistema pulsa el botón "Eliminar".	8. El sistema elimina los datos del equipamiento y muestra un mensaje indicando que la operación se ha realizado satisfactoriamente.

**Curso alternativo de los eventos**

	8. El sistema muestra un mensaje indicando que no existe equipamiento con esos datos.
--	---

**Prototipo**

**Prioridad:** Crítico

Tabla 15 Descripción detallada del caso de Uso del sistema "Gestionar datos de equipamiento"

#### **3.6 Conclusiones.**

Durante el desarrollo de este capítulo se ha hecho un análisis de los procesos y casos de uso del negocio, se han planteado los requerimientos, tanto funcionales como no funcionales y se ha hecho una descripción detallada del sistema propuesto, se han descrito los casos de uso del sistema así como sus actores y trabajadores, logrando así una visión mas clara de el sistema que se desea proponer.

## CAPÍTULO 4

# “Análisis y Diseño del Sistema”

### 4.1 Introducción

El análisis y diseño del sistema son uno de los mecanismos más importantes dentro del proceso unificado del desarrollo de un software. Se consideran más cercanos al acceso de datos y a la arquitectura física de la aplicación.

#### Modelo de Análisis

El modelo de análisis ofrece una especificación más precisa de los requisitos, los estructura de modo que facilita su preparación, su comprensión, su modificación y en general su mantenimiento. Este es tomado como una primera aproximación al modelo del diseño. Por su parte, este se encargará de moldear el sistema y buscar una forma de arquitectura que de vida a los requerimientos del sistema, que incluya componentes de código que se compilan e integran en versiones ejecutables del mismo.

Durante el análisis se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos, buscando la forma de conseguir una comprensión más precisa y una descripción que sea fácil de mantener, que ayude a estructurar el sistema entero incluyendo la arquitectura.

En el modelo de clases del análisis se definen varios artefactos, tal es el caso de las clases de interfaz, entidad y control.

Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del modelo del diseño.

### 4.2 Diagramas de clases del análisis.

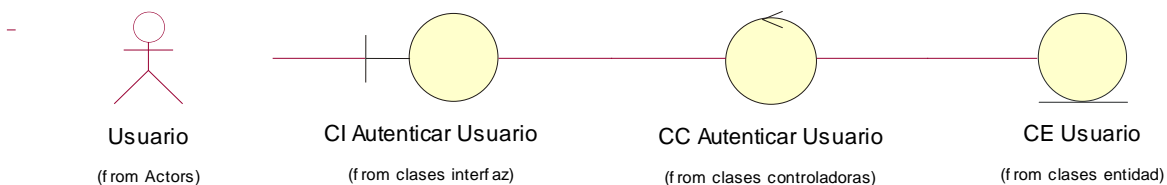


Fig. 14 Diagrama de clases del análisis del Caso de uso “Autenticar Usuario”

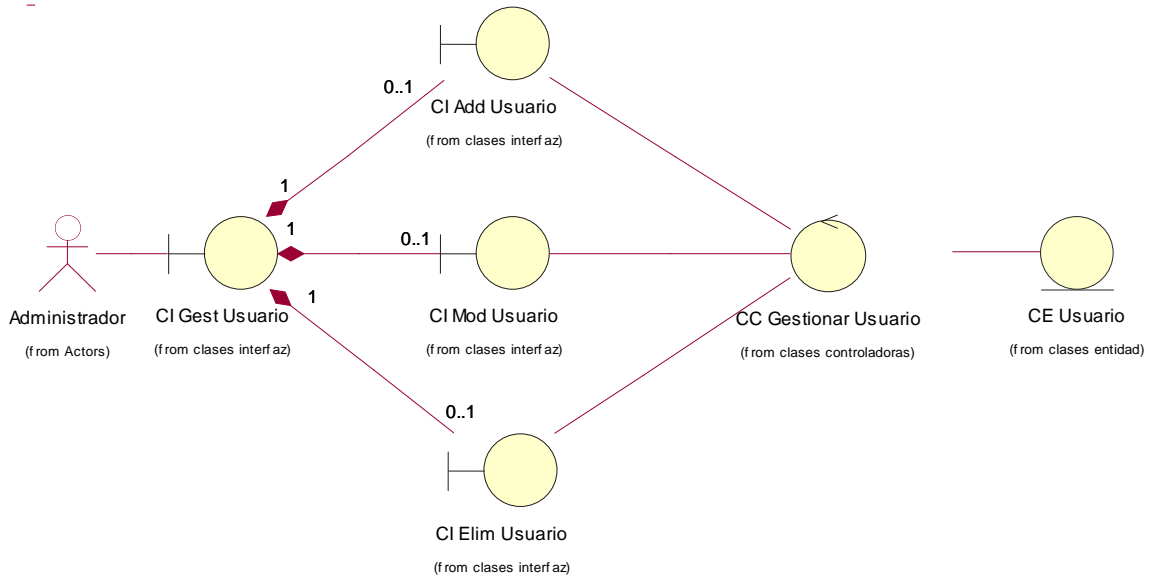


Fig. 15 Diagrama de clases del análisis del Caso de uso “Gestionar Usuario”

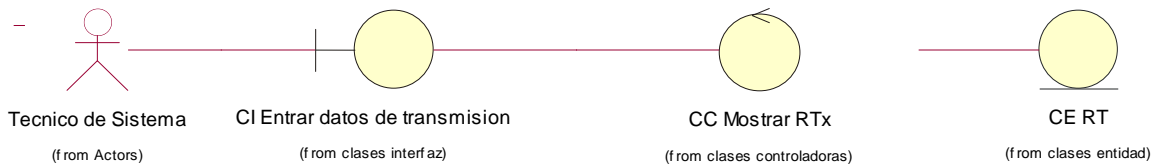


Fig. 16 Diagrama de clases del análisis del Caso de uso “Mostrar Reporte de Transmisión”



### 4.3 Diagramas de Interacción (Colaboración)

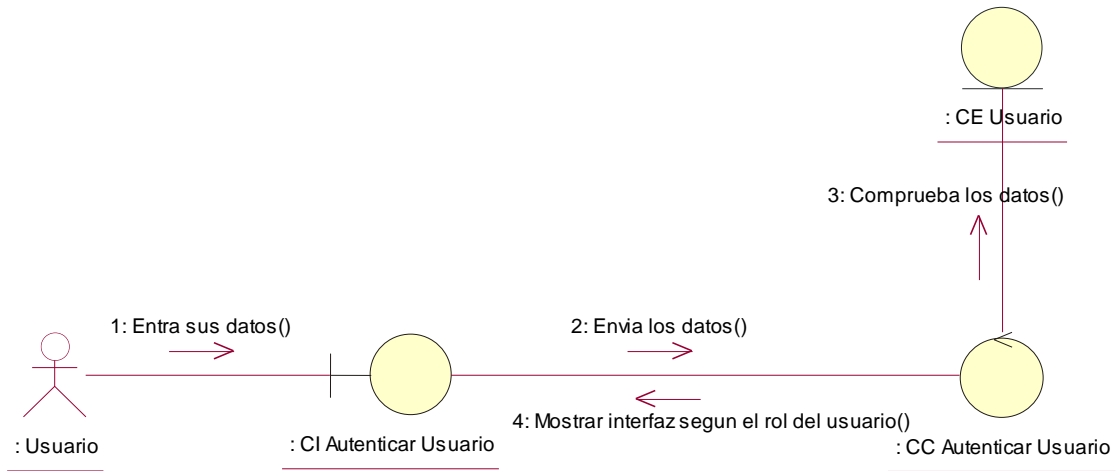


Fig. 17 Diagrama de colaboración “Autenticar Usuario”

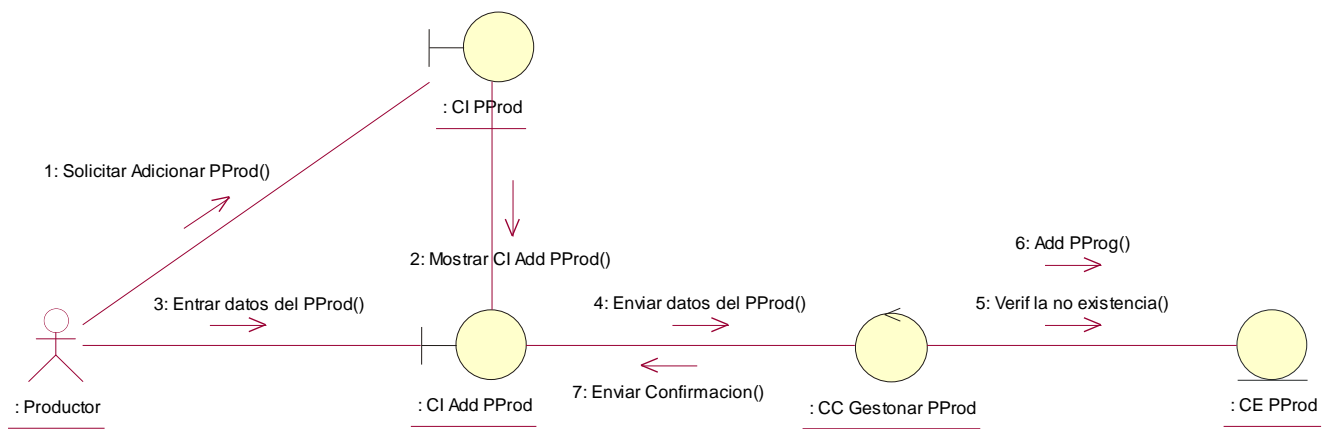


Fig. 18 Diagrama de colaboración “Adicionar Plan de Producción”

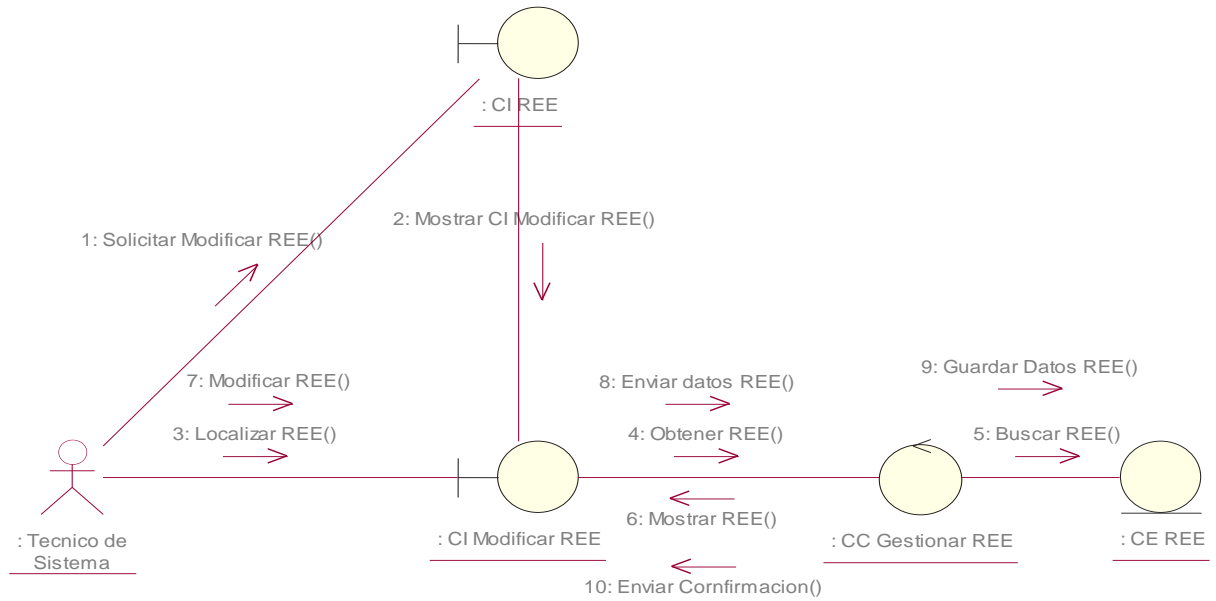


Fig. 19 Diagrama de colaboración "Modificar reporte de Equipamiento Entregado"

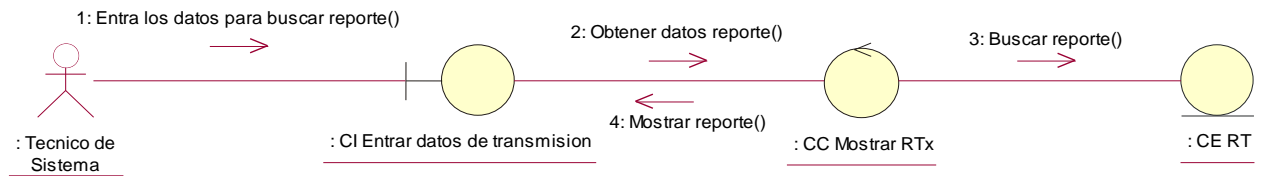


Fig. 20 Diagrama de colaboración "Mostrar Reporte de Transmisión"

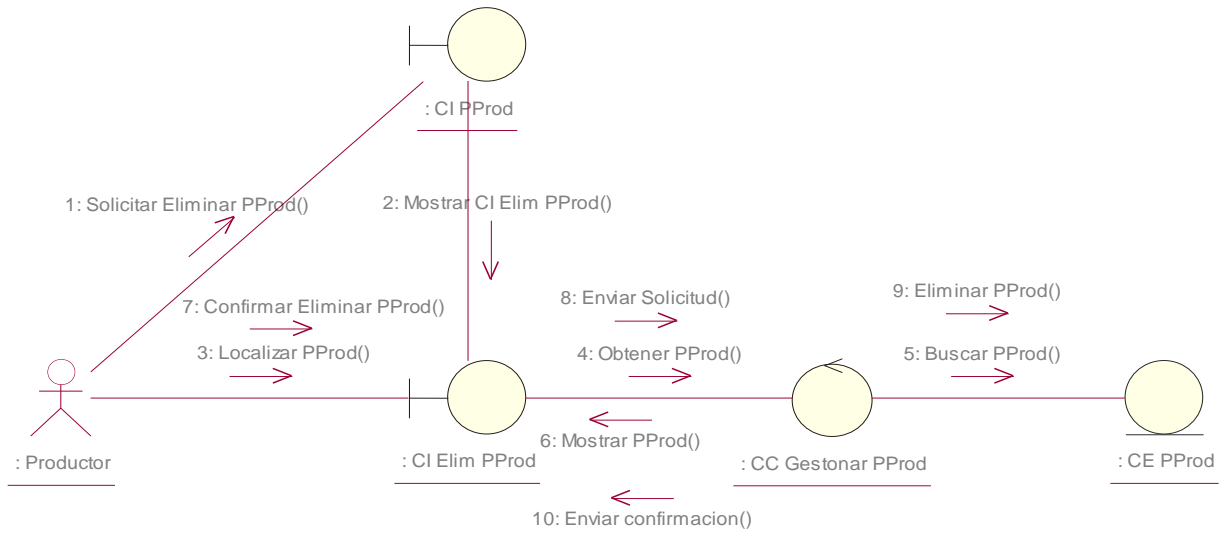


Fig. 21 Diagrama de colaboración “Eliminar Plan de Producción”

#### 4.4 Modelo del Diseño

Al modelar las aplicaciones Web se puede generar a partir de un proceso creativo, la solución a un nivel determinado de abstracción; pero además esto facilita el manejo de todo lo que pueda ser cambiado puesto que se hace más sencillo cambiar un modelo que lo que se haya generado a través del código fuente.

El modelo de diseño intenta preservar la estructura definida por el modelo de análisis. Es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto a otras restricciones del entorno de implementación tienen su impacto en el sistema que se desarrolla. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación.

Para esta investigación se ha seleccionado un framework para llevar a cabo la construcción de la solución, esto implica una reestructuración del diseño en función de la implementación propia de la arquitectura que él realiza.

#### 4.5 Diagrama de Clases del Diseño

Para el caso de los diagramas de diseño, se decidió representar solamente las clases y extensiones web que tengan que ver directamente con la construcción de la solución, esto debido a la gran complejidad de la solución, incluyendo las partes fundamentales del framework que ayudarán a una mejor comprensión de los diagramas que se presentan a continuación.

Según la arquitectura MVC que implementa Symfony el diseño queda estructurado de la siguiente manera:

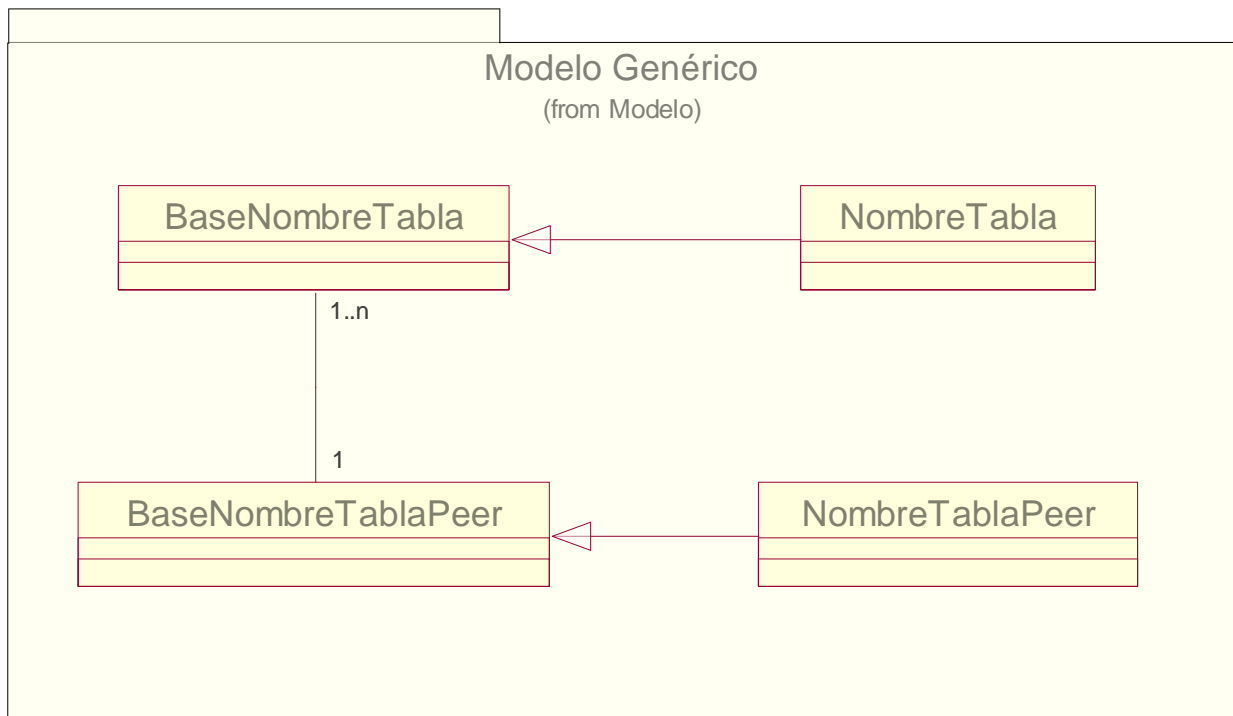
### El modelo deberá fundamentalmente:

- Acceder a los datos persistentes, a la capa de almacenamiento de datos.
- Definir las reglas de negocio.

Cuando se trabaja con Symfony, este genera automáticamente todas las clases del modelo según el diseño de los datos implementado sobre algunos de los gestores que son compatibles con él, en este caso con PostgreSQL, traduce por cada tabla real de la Base de Datos cuatro clases: *NombreTabla*, *NombreTablaPeer*, *BaseNombreTabla*, *BaseNombreTablaPeer* las cuales se encargan en conjunto de realizar todo el acceso a los datos y la lógica de la aplicación.

- ***NombreTabla* y *NombreTablaPeer***: Pudiera decirse que son la parte más importante de la aplicación, se encargan en su conjunto de toda la lógica del negocio.
- ***BaseNombreTabla***: Contiene todos los atributos definidos en la tabla y un conjunto de métodos ya implementados que gestionan el acceso a los datos y aceleran y simplifican el trabajo del equipo.
- ***BaseNombreTablaPeer***: Contiene un conjunto de métodos estáticos que complementan el acceso y la lógica de los datos.

Según lo expuesto anteriormente el modelo quedaría representado como se muestra en la siguiente figura:



**Fig. 22 Estructura de las clases generadas por Symfony**

Con el objetivo de disminuir la complejidad de los diagramas de diseño, y siguiendo lo señalado anteriormente, se decidió representar solamente en el modelo las clases **BaseNombreTabla** pues son las que proporcionan información acerca de los datos en la BD ya que a través de ellas se realizan las relaciones entre las tablas. Las representaciones de algunas de las relaciones entre estas clases según los Casos de Uso se encuentran en el Anexo 9.

**El Controlador:** Esta capa se divide en dos partes fundamentales

- Las acciones
- El controlador frontal.

Las acciones: Se encargan de obtener los resultados del modelo y define variables para la vista. Constituyen métodos con el nombre *executeNombreAccion* de la clase *NombreModuloAction* que a su vez hereda de la clase *sfActions*, es importante decir que el prefijo *execute* de las acciones es obligatorio.

El Controlador Frontal es el único punto de entrada a la aplicación, representa un archivo .php, carga la configuración y determina la acción a ejecutarse, además, todas las peticiones son manejadas por él que ayudado por los demás componentes del framework realizan todo el trabajo transparente al programador. El archivo .php es nombrado en esta investigación frontend.php.

**La Vista:** Contiene tres partes fundamentales

- El layout (conocido como plantilla global)
- El complemento de las acciones (llamado plantilla)
- Las páginas con sus formularios.

El layout almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla.

El complemento de las acciones o plantilla se encarga de, con los resultados de la acción, se inserta en el body de la layout y genera finalmente la página web resultado de la petición de un usuario. Esta plantilla tiene el mismo nombre que el sufijo de la acción correspondiente y termina en la palabra Success.

**Propel:** que también es un proyecto de software libre, es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5. Propel está completamente integrado en Symfony.

La librería Propel se encarga de la generación automática de las clases de la capa del modelo en función de la estructura de datos de la aplicación, es decir, crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario

La abstracción de la base de datos es completamente invisible al programador, ya que la realiza otro componente específico llamado Creole. Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración.

Propel se utiliza para el Mapeo de Objetos Relacional (ORM). Proporciona persistencia para los objetos y un servicio de consultas.

Symfony utiliza Propel como ORM y Propel utiliza Creole como capa de abstracción de bases de datos. Estos 2 componentes externos han sido desarrollados por el equipo de Propel, y están completamente integrados en Symfony, por lo que se pueden considerar una parte más del framework.

A continuación se muestran algunos de los diagramas de clases del diseño:

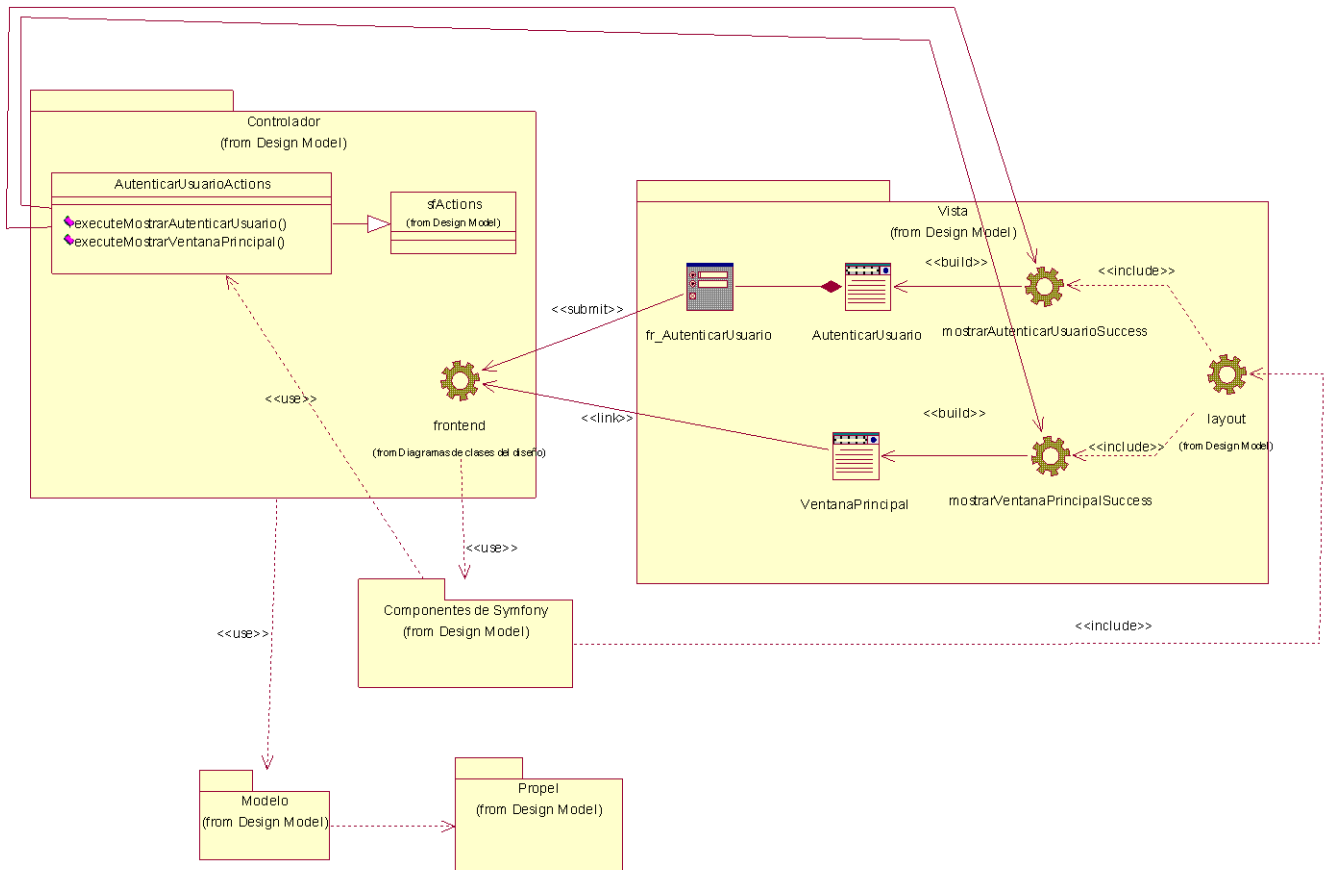


Fig. 23 Diagrama de Clases del Diseño CUS-Autenticar Usuario





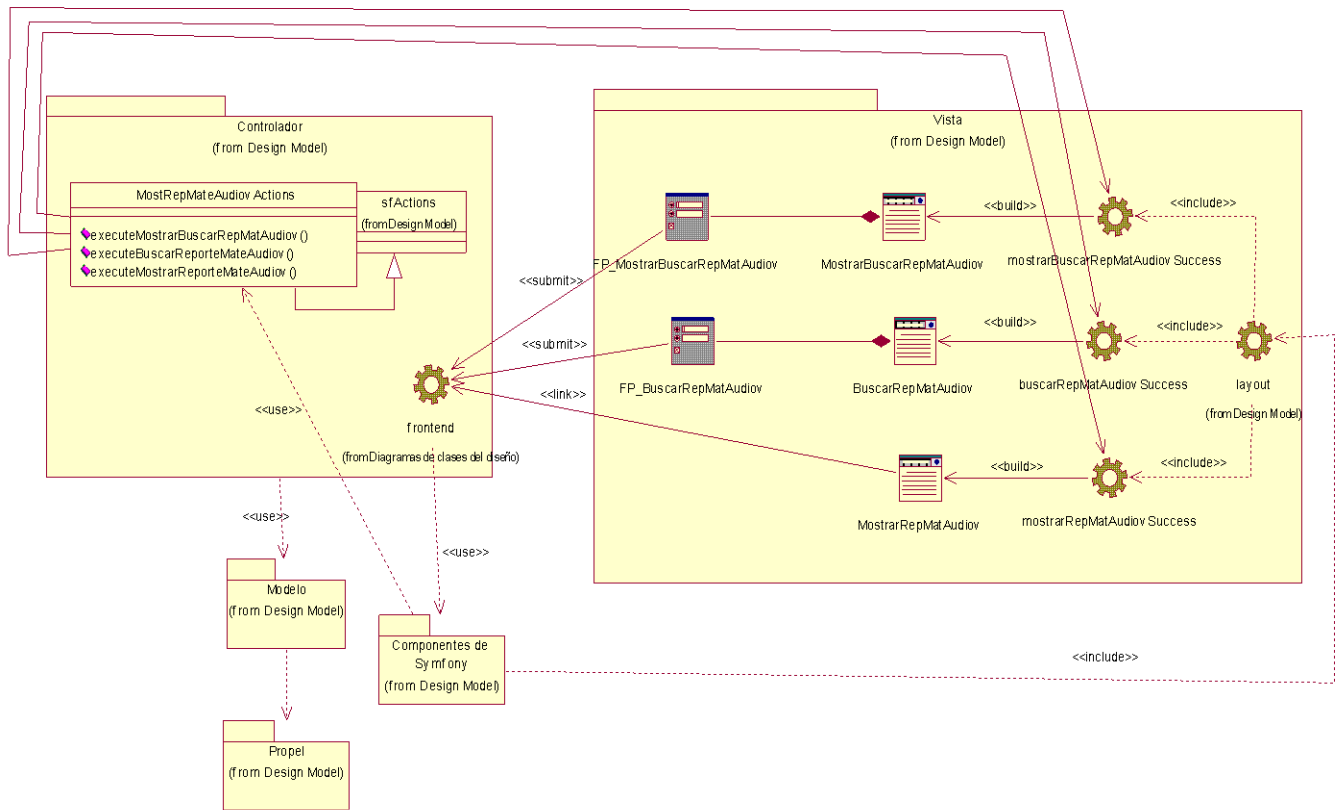


Fig. 25 Diagrama de Clases del Diseño CUS-Mostrar Reporte de Material Audiovisual

#### 4.6 Patrones de diseño en Symfony

Una de las definiciones mas aceptadas sobre patrones de diseño fue enunciada por Cristopher Alexander cuando dijo: "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución un millón de veces, sin hacer lo mismo dos veces".

Las metodologías orientadas a objetos tienen como su máxima el hecho de no reinventar la rueda en cada nuevo proyecto. Por tanto los patrones se convierten en esenciales para la reutilización del software. El objetivo principal que persiguen los patrones es crear un lenguaje común entre los Diseñadores y Desarrolladores para comunicar experiencias sobre los problemas y sus soluciones.

Los patrones de diseño se refieren a las descripciones de comunicación de las clases y los objetos que pueden personalizarse para resolver un problema de diseño general en un contexto particular.

Symfony, es un framework que sigue las mejores prácticas y patrones de diseño para la web, utiliza tanto patrones GRASP como patrones GoF, por ejemplo:

### **Patrones GRASP**

#### Alta Cohesión

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

#### Bajo Acoplamiento

La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases, esto implica disminución de modificaciones frente a grandes cambios.

#### Creador

En las acciones se crean los objetos de las clases que representan las entidades, esto evidencia que la clase Actions es "creadora" de dichas entidades.

#### Controlador

Todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado.

Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

### **Patrones GoF**

En la categoría Creacionales:

#### Singleton (Instancia única)

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a sfContext::getInstance().

En una acción, el método `getContext()`, un objeto muy útil que guarda una referencia a todos los objetos del núcleo de `Symfony Abstract Factory` (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

En la categoría Estructurales:

Decorator (Envoltorio)

Añade funcionalidad a una clase, dinámicamente. El archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado Composite (Objeto compuesto)

Composite (Objeto compuesto)

Permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

## 4.7 Diagrama de Clases Persistentes

El Diagrama de Clases se puede usar para modelar la estructura lógica de la base de datos, con clases representando tablas, y atributos de clase representando columnas. Las clases persistentes, que se definen en estos diagramas de clases, y sus atributos; hacen referencia directamente a las entidades lógicas y a sus atributos. A continuación se muestra el diagrama de clases persistentes y el modelo de datos generado una vez que se realizó el diagrama.





#### 4.7 Descripción de las tablas del Modelo de Datos y sus atributos.

<b>Nombre: T_Usuario</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de los usuarios.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombre	Varchar (255)	Nombre del usuario
apellidos	Varchar (255)	Apellidos del usuario
usuario	Varchar (255)	Identificador de la tabla
contraseña	Varchar (255)	Contraseña del usuario
Id_rol	SmallInt	Llave foránea de la tabla T_Rol para saber el rol que tiene asignado.

<b>Nombre: T_Equipamiento</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos comunes para los equipamientos.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
no_serie	Varchar (255)	Es el identificador de cada equipamiento

<b>Nombre: T_Equip Inf</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos del equipamiento informático.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
administrador	Varchar (255)	Nombre del administrador
HDD	Varchar (255)	Información sobre el disco duro
ram	Varchar (255)	Memoria Ram
motherboard	Varchar (255)	Información sobre motherboard
tarjeta_video	Varchar (255)	Información sobre tarjeta video
estado	Varchar (255)	Estado del equipamiento
tipo_material	Varchar (255)	Tipo de equipamiento
no_serie	Varchar (255)	Identificador de la tabla

<b>Nombre: T_Equip Esp de TV</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de los equipamientos especializados de televisión.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombre_equipo	Varchar (255)	Nombre del equipo
estado	Varchar (255)	Estado del equipo
observaciones	Varchar (255)	Algo que se desee añadir sobre el equipamiento.
no_serie	Varchar (255)	Identificador de la tabla

<b>Nombre: T_Equipamiento_Usuario</b>		
<b>Descripción:</b> en esta tabla se almacenan los identificadores del usuario y el equipamiento que le fue asignado		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
usuario	Varchar (255)	Identificador del usuario
no_serie	Varchar (255)	Numero de serie de equipamiento

<b>Nombre: T_PProd</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos un plan de producción		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
semana	Varchar (255)	Es el identificador de la tabla
usuario	Varchar (255)	Ususer del usuario que creo el plan de producción
fecha	DateTime	Fecha en que se confecciono el plan
responsable	Varchar (255)	Nombre de la persona que confecciono el plan
list_activ	SMALLINT	Lista de las actividades con todos los datos pertenecientes a cada una

<b>Nombre: T_PProg</b>		
<b>Descripción:</b> en esta tabla se almacenan de cada material incluido en el plan de programación		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
semana	Varchar (255)	Es el identificador de la tabla
usuario	Varchar (255)	Ususer del usuario que creo el plan de producción
observaciones	Varchar (255)	Algo que se desee añadir sobre el algún material programado o sobre el plan en general
responsable	Varchar (255)	Nombre de la persona que confecciono el plan de programación
list_materiales	Varchar (255)	Lista con los materiales programados en cada día con sus datos correspondientes

<b>Nombre: T_Solic Prod</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de las solicitudes de producción		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

Id_sol_prod	Int	Es el identificador de la tabla
nombre_solicitante	Varchar (255)	Nombre del solicitante
area_pertenece	Varchar (255)	Área a la que pertenece el solicitante
tipología_material	Varchar (255)	Tipología del material solicitado
motivo	Varchar (255)	Motivo por el que hace la solicitud
tema	Varchar (255)	Tema de la producción
usuario	Varchar (255)	Usuario que hace la solicitud
descripción	Varchar (255)	Descripción de la solicitud
fecha entrega	DateTime	Fecha que necesita que se le entregue
fecha solicitud	DateTime	Fecha de la solicitud

<b>Nombre: T_REE</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de un reporte de equipamiento entregado		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_reporte	Int	Es el identificador de la tabla
fecha	DateTime	Fecha del reporte
responsable	Varchar (255)	Responsable del reporte
actividad asociada	Varchar (255)	Actividad para la que se destinó el material
material_entregado	Varchar (255)	Materiales entregado
serie_material	Varchar (255)	Serie de cada material
observaciones	Varchar (255)	Otros datos de interés
usuario	Varchar (255)	Llave foránea, Usuer del usuario que hace el reporte

<b>Nombre: T_Solic_Media</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de las solicitudes de medias.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_sol_media	Int	Es el identificador de la tabla
nombre_solicitante	Varchar (255)	Nombre del solicitante
area_pertenece	Varchar (255)	Área a la que pertenece
motivo	Varchar (255)	Motivo por el que necesita la media
media_solicitada	Varchar (255)	Media solicitada
descripción	Varchar (255)	Otros datos de interés
usuario	Varchar (255)	Llave foránea, Usuer del usuario que hace el reporte
fecha_solicitud	DateTime	Fecha de la solicitud
fecha_necesita	DateTime	Fecha para la que necesita la media

<b>Nombre: T_RT</b>
---------------------



<b>Descripción:</b> en esta tabla se almacenan los datos de los reportes de transmisión		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fecha	DateTime	Es el identificador de la tabla
programación	Varchar (255)	Indica si el error fue de programación
software	Varchar (255)	Indica si el error fue de software
hardware	Varchar (255)	Indica si el error fue de hardware
humano	Varchar (255)	Indica si el error fue humano
colaterales	Varchar (255)	Indica si el error fue algún daño colateral
observaciones	Varchar (255)	Otros datos de interés
usuario	Varchar (255)	Llave foránea, Usset del usuario que hace el reporte

<b>Nombre: T_Media</b>		
<b>Descripción:</b> en esta tabla de almacenan los datos comunes para lo diferentes tipos de medias		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
código	Varchar (255)	Es el identificador de la tabla
código_dvd	Varchar (255)	Código del Dvd donde se encuentra la media
titulo	Varchar (255)	Titulo de la media
fuelle	Varchar (255)	Fuente de la que proviene
estado	Varchar (255)	Estado en que se encuentra
palabra_clave	Varchar (255)	Palabra clave para identificarla
fecha_producción	DateTime	Fecha en que se produjo.

<b>Nombre: T_Usuario_Media</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de las medias asignadas con el usuario correspondiente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
usuario	Varchar (255)	Es el identificador de la tabla
código	Varchar (255)	Llave foránea, código de la media entregada

<b>Nombre: T_Rol</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de los roles		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombre	Varchar (255)	Nombre del rol
id_rol	Int	Es el identificador de la tabla

<b>Nombre: T_Permiso</b>		
<b>Descripción:</b> en esta tabla se almacenan lo datos de los permisos		

Atributo	Tipo	Descripción
nombre	Varchar (255)	Es el identificador de la tabla
Id_rol	Int	Llave foránea identificador del rol asociado

Nombre: T_Rol_Permission		
Descripción:		
Atributo	Tipo	Descripción
id_rol	Int	Llave foránea y primaria identificador del rol asociado
nombre	Varchar (255)	Llave foránea y primaria Es el nombre del permiso

Nombre: T_audiovisuales_varios		
Descripción: en esta tabla se almacenan los datos de los audiovisuales		
Atributo	Tipo	Descripción
dpto._responsable	Varchar (255)	Departamento responsable
clasificacion	Varchar (255)	Clasificación del audiovisual
personalidad	Varchar (255)	Nombre
cargo_personalidad	Varchar (255)	Cargo de personalidad
formato	Varchar (255)	Formato del audiovisual
fecha_conmemorativa	DateTime	Motivo de la fecha
duración	Varchar (255)	Duración del audiovisual
lugar	Varchar (255)	Lugar donde se realizo
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

Nombre: T_Documental		
Descripción: en esta tabla se almacenan los datos de un documental		
Atributo	Tipo	Descripción
director	Varchar (255)	Director del documental
duración	Varchar (255)	Duración del documental
formato	Varchar (255)	Formato del documental
sinopsis	Varchar (255)	Breve resumen del documental
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

Nombre: T_Efectos_Sonido		
Descripción: en esta tabla se almacenan los datos de los efectos de sonido		
Atributo	Tipo	Descripción
categoría	Varchar (255)	Categoría de los efectos.
canales	Varchar (255)	Canales de los efectos.
veloc_muestra	Varchar (255)	Velocidad de muestra.

veloc_transm	Varchar (255)	Velocidad transmitida.
formato	Varchar (255)	Formato del efecto de sonido
entrada_BD	Varchar (255)	Entregado en la BD.
observaciones	Varchar (255)	Otros datos de interés.
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Foto</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de una imagen.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
formato	Varchar (255)	Formato de la imagen.
autor	Varchar (255)	Autor de la imagen.
dimensiones	Varchar (255)	Dimensiones de la imagen.
periodista_asociado	Varchar (255)	Periodista de esa imagen.
resolución	Varchar (255)	Resolución de la imagen.
material_asociado	Varchar (255)	Material para el que se va a utilizar la imagen.
color	Varchar (255)	Color de la imagen.
temática	Varchar (255)	Temática de la imagen.
observaciones	Varchar (255)	Otros datos de interés.
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Multimedia</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de una multimedia.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
autores	Varchar (255)	Autores de la multimedia.
entrada_base_datos	DateTime	Fecha de entrada a la BD.
clasificación	Varchar (255)	Clasificación de las multimedias
idioma	Varchar (255)	Idioma de la multimedia
formato	Varchar (255)	Formato de la multimedia
isbn	Varchar (255)	
Edición	Varchar (255)	Edición de la multimendia
Tipología_material	Varchar (255)	Tipo del material que se va a escoger.
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Musica</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de la música		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
formato	Varchar (255)	Formato de música.
interprete	Varchar (255)	Interprete de la música.
canales	Varchar (255)	Canales de música.
disco	Varchar (255)	Disco donde esta la música.
genero	Varchar (255)	Tipo de género de música.
velocidad_transmision	Varchar (255)	Velocidad de transmisión

velocidad_muestra	Varchar (255)	Velocidad de muestra
país	Varchar (255)	País del que proviene la música
responsabilidad_mencion	Varchar (255)	
casa_disquera	Varchar (255)	Casa disquera de la música
tamaño_muestra	Varchar (255)	Tamaño de la muestra
referencia	Varchar (255)	Dirección física de la música.
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Pelicula</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de la película		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
director	Varchar (255)	Director de la película
genero	Varchar (255)	Genero de la película
duración	Varchar (255)	Duración de la película
formato	Varchar (255)	Formato de la película
país	Varchar (255)	País de la película
reparto	Varchar (255)	Actores que trabajan en la película
sinopsis	Varchar (255)	Breve resumen de la película
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Programa</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de un programa.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
director	Varchar (255)	Director del programa
duración	Varchar (255)	Duración del programa
tema	Varchar (255)	Tema del programa
formato	Varchar (255)	Formato en el que se encuentra el programa
código	Varchar (255)	Es el identificador de la tabla

<b>Nombre: T_Serie</b>		
<b>Descripción:</b> en esta tabla se almacenan los datos de la serie		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
director	Varchar (255)	Director de la serie
genero	Varchar (255)	Genero de la serie
formato	Varchar (255)	Formato en el que se encuentra la serie
rango_cap	Varchar (255)	Cantidad de capítulos de la serie
temporada	Varchar (255)	Temporada de la serie
reparto	Varchar (255)	Actores que trabajan en la serie

sinopsis	Varchar (255)	Breve resumen de la serie
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

Nombre: T_Teleclase		
Descripción: en esta tabla se almacenan los datos una teleclase		
Atributo	Tipo	Descripción
profesor	Varchar (255)	Profesor que imparte la teleclase
formato	Varchar (255)	Formato en que esta la teleclase
semestre	Varchar (255)	Semestre del curso en que se impártela teleclase
curso	Varchar (255)	Curso en el que se creo
duración	Varchar (255)	Duración de la teleclase
no_conf	Varchar (255)	Numero de la conferencia
créditos	Varchar (255)	Quienes ayudaron a su realización
observaciones	Varchar (255)	Otros datos de interés
código	Varchar (255)	Es el identificador de la tabla

#### 4.8 Modelo de Despliegue

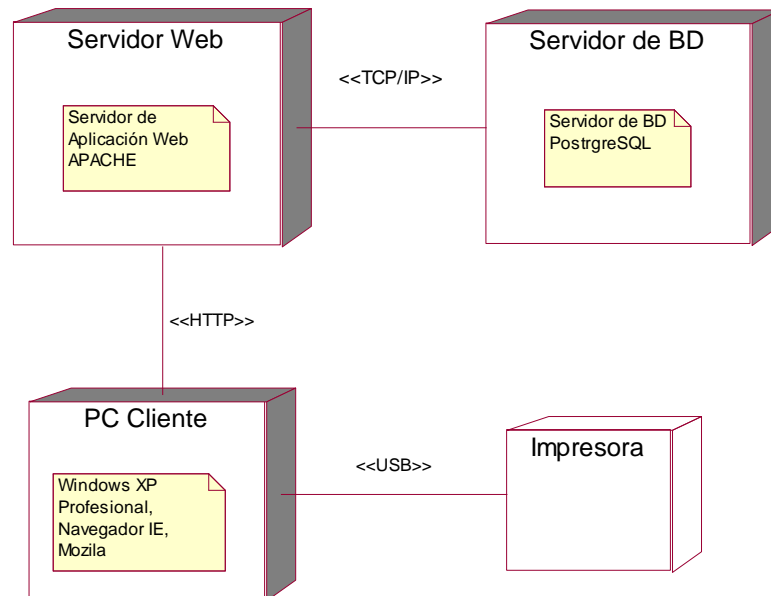


Fig. 28 Modelo de Despliegue

#### **4.9 Conclusiones**

En el presente capítulo se mostraron los elementos más importantes del análisis y el diseño de el sistema que se propone en esta investigación, se abordó una breve explicación sobre el funcionamiento y la construcción de los diagramas utilizando la notación que se emplea en Symphony y su implementación del patrón Modelo-Vista-Controlador, así como una explicación detallada de las tablas que forman el modelo de datos.

## CONCLUSIONES

Los Sistemas de Gestión de la Información presentan soluciones eficaces al procesamiento de datos, ha sido objetivo de este trabajo proponer el diseño de uno de ellos, enfocado fundamentalmente al procesamiento de información en la Dirección de Televisión Universitaria.

El resultado ha sido el Diseño de un Sistema de Gestión de Procesos para la DTU, el cuál será de gran utilidad a la Dirección de Televisión Universitaria ya que se podrán manipular de una forma más rápida y organizada los grandes volúmenes de información que diariamente se gestiona en dicha dirección.

El desarrollo de este trabajo de tesis esta orientado al diseño de una herramienta que gestione la información que se maneja en la dirección. Para su futura implementación se proponen herramientas libres, se decidió utilizar el framework Symfony, el cuál implementa la arquitectura modelo-vista-controlador, como lenguaje de programación PHP 5 utilizando un gestor de base de datos Postgre SQL y Apache como servidor Web. Todo esto brinda la posibilidad que la aplicación sea desplegada en los sistemas operativos Linux y Windows.

Con el estudio realizado a lo largo de todas las etapas de desarrollo y el sistema que se ha logrado diseñar, se cumple con el objetivo general del proyecto: diseñar un sistema de gestión de procesos para la DTU, que permita administrar los flujos de trabajo dentro de la estructura de la institución, reafirmando así la utilidad y validez de emplear las tecnologías informáticas para apoyar las labores que se desarrollan en todas las esferas.

Con el diseño de este sistema se pretende:

Organizar los flujos de trabajos dentro de la DTU para realizar con mayor eficiencia los servicios que brinda dicha entidad. Eliminar todo tipo de documentos a la hora de las comunicaciones entre los trabajadores de la dirección y con esto un ahorro de materiales y una mejor gestión de la información que estos manipulan.

Se realizó el análisis y diseño del sistema, se han seguido los principios básicos de diseño descritos para el desarrollo del mismo y se espera lograr una buena protección de los datos consecuentes con el nivel de seguridad requerido en la Dirección de Televisión Universitaria.

## RECOMENDACIONES

Se recomienda:

- Continuar con la implementación del sistema y de esta forma poder ver los resultados de este trabajo que aquí se concluye.
- Continuar el estudio con el objetivo de añadir nuevos requerimientos y funcionalidades al sistema, para mejorar los servicios existentes en la dirección y otros que permitan gestionar los nuevos servicios que la dirección comenzará a brindar.
- Proponer el diseño e implementación de sistemas similares en las diferentes televisoras de nuestro país, adaptándolo a las características y flujos de trabajo de cada una de ellas.



## REFERENCIAS BIBLIOGRÁFICAS

1. CASTELLS, P. *La Web semántica*. publicado el: 5 marzo del 2008 de última actualización: 5 marzo del 2008. 13 p. Disponible en: <http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf>
2. Y. VALIDO, Y. M. *SAIMM: Sistema de Apoyo Integral a la Misión Milagro*. Aplicación Web, Instituto Superior Politécnico “José Antonio Echeverría”, 2005.
3. GIL, R. *Sistemas de Información*. publicado el: 5 de marzo del 2008 de última actualización: 5 de marzo del 2008. 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).
4. L.MÉNDEZ, A. T. *Sistema de Promoción y Gestión Comercial para la oficina de Transferencia Tecnológica de la Universidad de Cienfuegos*. Instituto Superior Politécnico “José Antonio Echeverría”, 2005.
5. DUQUE MÉNDEZ, N. D. *Conceptos de arquitectura Cliente/Servidor*. publicado el: 4 de abril del 2008 de última actualización: 4 de abril del 2008. 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).
6. VALDÉS, D. P. *Los diferentes lenguajes de programación para la Web*. publicado el: 02 de noviembre del 2007 de 2007, última actualización: 02 de noviembre del 2007. [Consultado el: 4 de abril del 2008]. Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
7. ALVAREZ, M. A. *Breve Historia de PHP*. publicado el: 3 de abril del 2008 de 2001, última actualización: 3 de abril del 2008. Disponible en: <http://www.desarrolloweb.com/articulos/436.php>.
8. S, C. V. D. H. *¿Qué es el PHP?* publicado el: 4 de abril del 2008 de 2001, última actualización: 4 de abril del 2008. Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro>.
9. TERRERO, E. D. *Sistema Automatizado para la Gestión de Información de la Misión Milagro*. Universidad de las Ciencias Informáticas, 2006.
10. MATEU, C. *Desarrollo de aplicaciones Web*. publicado el: 3 de abril del 2008 de 2004, última actualización: 3 de abril del 2008. [Consultado el: 3 de Abril del 2008]. 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).
11. DATOS, A. D. P. D. *Glosario de términos sobre Internet y Spam\_archivos*. publicado el: 3 de abril del 2008 de 2004, última actualización: 3 de abril del 2008. Disponible en: <https://www.agpd.es/index.php?idSeccion=541>

12. DUEÑAS, J. B. *Configuración básica de apache*. publicado el: 27 de marzo del 2008 de 2007, última actualización: 27 de marzo del 2008. Disponible en: [http://linux.cienfuegos.jovenclub.cu/index.php?option=com\\_content&task=view&id=16&Itemid=7](http://linux.cienfuegos.jovenclub.cu/index.php?option=com_content&task=view&id=16&Itemid=7)
13. BARRAPUNTO. *El servidor de web Apache*. publicado el: 2 de Marzo de 2008 de última actualización: 2 de Marzo de 2008. Disponible en: <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/c20.html>.
14. YUNIER ALBRECHT, R. T. *Sistema Integrado para la Gestión Estomatológica*. Universidad de las Ciencias Informáticas, 2007.
15. DESARROLLOWEB. *Arquitectura del servidor Apache*. publicado el: 2 de Marzo de 2008 de última actualización: 2 de Marzo de 2008. Disponible en: <http://www.desarrolloweb.com/articulos/1112.php>.
16. JESÚS TRAMULLAS, K. *Introducción a la documática*. publicado el: 5 de marzo de 2008 de 1997, 2000, última actualización: 5 de marzo de 2008. Disponible en: <http://tramullas.com/documatica/2-4.html>.
17. ORTIZ, A. *error500*. publicado el: 22 de marzo 2008 de 2006, última actualización: 22 de marzo 2008. Disponible en: [http://www.error500.net/garbagecollector/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos](http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_datos).
18. MEDINA, J. M. A. *Administración SGBD PostgreSQL*. publicado el: 23 de marzo 2008 de 2006, última actualización: 23 de marzo 2008. Disponible en: [http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/SI\\_TARGES/manual.pdf](http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/SI_TARGES/manual.pdf).
19. *Calidad*. publicado el: 18 de marzo 2008 de última actualización: 18 de marzo 2008. Disponible en: <http://www.ne.com.co/html/esp/calidad.html>
20. *Proceso Unificado para Desarrollo de Software (RUP)*. 2001.
21. TOROSSI, G. *El Proceso Unificado de Desarrollo de Software* publicado el: 19 de marzo 2008 de última actualización: 19 de marzo 2008. Disponible en: <http://www.ecomchaco.com.ar/UTN/disenodesistemas/apuntes/oo/ApunteRUP.pdf>.
22. MOLPECERES, A. *Proceso de desarrollo: RUP, XP y FDD*. publicado el: 19 de marzo 2008 de 2002, última actualización: 19 de marzo 2008. Disponible en: <http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF>.
23. *EL LENGUAJE UNIFICADO DE MODELADO (UML)*. . publicado el: 10 de Marzo de 2008 de última actualización: 10 de Marzo de 2008. Disponible en: <http://www.fi-b.unam.mx/pp/profesores/carlos/aydoo/uml.html>.
24. VAGO, E. R. D. *EL DESARROLLO DE SISTEMAS DE INFORMACIÓN EMPLEANDO EL LENGUAJE DE MODELADO UNIFICADO UML*. publicado el: 10 de Marzo de 2008 de última

actualización: 10 de Marzo de 2008. Disponible en:

[http://html.rincondelvago.com/uml\\_5.html](http://html.rincondelvago.com/uml_5.html).

25. *Introducción a UML* publicado el: 10 de marzo 2008 de última actualización: 10 de marzo 2008. Disponible en: <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>.

## GLOSARIO

**CGIA:** Centro de Gestión de Información Audiovisual.

**DTU:** Dirección de Televisión Universitaria.

**UCI:** Universidad de las Ciencias Informáticas.

**FEU:** Federación Estudiantil Universitaria.

**UML:** Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

**WEB:** Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

**SGBD:** Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

**RUP:** Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

**PHP:** Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

**APACHE:** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

**Arquitectura Cliente/Servidor:** es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

**HTTP:** HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

**CUN:** Caso de Uso del Negocio.

**CUS:** Caso de Uso del Sistema.

**IDEs:** Integrated Development Environment. Entorno Integrado de Desarrollo.

**XHTML:** eXtensible HyperText Markup. Es un lenguaje de descripción pensado para destronar al HTML como estándar para hacer páginas web. Es la versión XML de HTML.