

**UNIVERSIDAD DE LAS CIENCIAS**

**INFORMÁTICAS**

Facultad 1



Título: **“Sistema de control de acceso para la plataforma IPTV-UCI”.**

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

**Autores: Elizabeth Zequeira Sánchez.**

**Guillermo Mesa Campo.**

**Tutor(es): Ing. Frank Miravalle Herrera.**

**Msc. Héctor Rodríguez Figueredo.**

**Ciudad de la Habana, Cuba**

**Junio, 2011**

**DECLARACIÓN DE AUTORÍA**

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

Para que así conste se firma la presente a los \_\_\_\_días del mes de \_\_\_\_ del año\_\_\_\_\_.

Elizabeth Zequeira Sánchez

Guillermo Mesa Campo

---

---

Firma del Autor

Firma del Autor

Frank Miravalle Herrera

Héctor Rodríguez Figueredo

---

---

Firma del Tutor.

Firma del Tutor.

**Ing. Frank Miravalle Herrera.**

Ingeniero en Ciencias Informáticas.

Categoría Científica: Ingeniero.

Años de Experiencia: 3 años.

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

Teléfono: 835-8000

**Msc.Héctor Rodríguez Figueredo.**

Licenciado en Cibernética Matemática.

Categoría Científica: Máster.

Años de Experiencia: 20 años.

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

Teléfono: 837-3671

### Resumen

Los medios de comunicación audiovisuales constituyen elementos de extrema importancia para el desarrollo, entretenimiento y culturización de las sociedades. De manera especial, la televisión ha tenido, históricamente, un poder de penetración cultural extraordinario influyendo en la formación de patrones de conducta en las distintas esferas de la sociedad. La televisión en la Universidad de las Ciencias Informáticas es una herramienta fundamental en el proceso de formación-producción. El soporte para la transmisión de multimedias y programas televisivos se realiza por cable coaxial, lo que trae consigo problemas en la calidad del servicio, un rápido deterioro y a la vez una exhaustiva tarea de soporte. Es por ello que se ha decidido implementar la transmisión de televisión utilizando el protocolo IP que utiliza la red de datos ya existente. El objetivo de la presente investigación es desarrollar un sistema de control que permita delimitar los niveles de acceso a los materiales audiovisuales constituyendo un elemento de vital importancia para las plataformas IPTV (por sus siglas en inglés *Internet Protocol Television*). Para el desarrollo de la aplicación se emplea la metodología RUP para guiar el proceso de desarrollo de software y UML como lenguaje de modelado. La aplicación se desarrolla en el IDE NetBeans, y se implementa en el lenguaje de programación PHP5. Esta aplicación permitirá la gestión del acceso a multimedias según tipos de usuarios, donde los roles estén definidos en la misma aplicación o extraídos de un directorio activo.

**Palabras Claves:** Autenticación, Autorización, rol, usuario.

**ÍNDICE**

Introducción..... - 7 -

CAPÍTULO 1: Fundamentación Teórica. .... - 11 -

    1.1 Conceptos asociados al dominio del problema ..... - 11 -

    1.2 Objeto de estudio..... - 13 -

        1.2.1 Surgimiento de IPTV ..... - 13 -

        1.2.2 IPTV en el Mundo..... - 14 -

        1.2.3. IPTV en la UCI: ..... - 16 -

        1.2.4 Control de acceso..... - 16 -

        1.2.5 Modelos de control de acceso ..... - 17 -

    1.3 Análisis de otras soluciones existentes ..... - 19 -

    1.4 Metodologías a utilizar, tendencias y tecnologías actuales. .... - 20 -

        1.4.1 Metodologías de desarrollo..... - 20 -

        1.4.2 Herramientas de modelado..... - 25 -

        1.4.3 Lenguaje de Programación ..... - 26 -

        1.4.4 Entorno de desarrollo..... - 28 -

        1.4.5 Sistema de control de versiones ..... - 29 -

        1.4.6 Servidor Web..... - 32 -

    1.5 Conclusiones parciales del capítulo ..... - 32 -

CAPÍTULO 2: Presentación de la solución propuesta..... - 33 -

    2.1 Entorno donde trabajará el sistema ..... - 33 -

        2.1.1 Conceptos principales del entorno..... - 33 -

        2.1.2 Características del servidor streaming Flumotion. .... - 34 -

        2.1.3 Eventos principales del entorno ..... - 35 -

        2.1.4 Diagrama de clases del modelo de dominio ..... - 36 -

        2.1.5 Glosario de Términos del Dominio..... - 37 -

    2.2 Requerimientos Funcionales ..... - 37 -

    2.3 Requerimientos No Funcionales ..... - 39 -

        2.3.1 Restricciones en el diseño y la implementación: ..... - 39 -

        2.3.2 Apariencia o interfaz externa: ..... - 39 -

        2.3.3 Seguridad: ..... - 39 -

        2.3.4 Rendimiento: ..... - 40 -

        2.3.5 Portabilidad: ..... - 40 -

        2.3.6 Requerimientos de Usabilidad:..... - 41 -

        2.3.7 Requerimientos de Soporte: ..... - 41 -

        2.3.8 Requerimientos Políticos:..... - 41 -

        2.3.9 Requerimientos de Software: ..... - 41 -

- 5 -

2.4 Descripción del Sistema Propuesto .....	- 42 -
2.4.1 Descripción de los actores .....	- 42 -
2.4.2 Casos de Uso del Sistema .....	- 43 -
2.4.2.1 Patrones de casos de uso utilizados .....	- 43 -
2.5 Conclusiones parciales del capítulo .....	- 55 -
<b>CAPÍTULO 3: Diseño de la solución propuesta.</b> .....	- 56 -
3.1 Diagramas de Clases .....	- 56 -
3.2 Diagramas de Secuencias .....	- 62 -
3.3 Principios de diseño .....	- 64 -
3.3.1 Estándares de la interfaz de la aplicación .....	- 64 -
3.4 Diseño de la Base de Datos .....	- 65 -
3.4.1 Diagrama Entidad Relación .....	- 65 -
3.4.2 Diagrama de Clases Persistentes .....	- 66 -
3.5 Estilo Arquitectónico .....	- 66 -
3.5.1 Ciclo de vida de MVC .....	- 67 -
3.5.2 Ventajas y Desventajas de MVC .....	- 68 -
3.6 Conclusiones parciales del capítulo .....	- 69 -
<b>CAPÍTULO 4: Implementación y pruebas.</b> .....	- 70 -
4.1 Modelo de Despliegue .....	- 70 -
4.2 Modelo de Implementación .....	- 71 -
4.3 Pruebas del sistema propuesto .....	- 72 -
4.4 Conclusiones parciales del capítulo .....	- 75 -
Conclusiones Generales .....	- 76 -
Recomendaciones .....	- 77 -
Referencias Bibliográficas .....	- 78 -
Glosario .....	- 81 -

### **Introducción**

La televisión y la radio como medios de comunicación constituyen elementos de extrema importancia para el desarrollo, entretenimiento y culturización de las sociedades. La televisión tiene un poder de penetración cultural extraordinario: influye en la formación y refuerzo de valores sociales, en la configuración de patrones de conducta, y ejerce un enorme poder de influencia sobre las opiniones y actitudes políticas de los miembros de la sociedad tal que, hoy en día se pueden ver generaciones completas que han sido educadas y entretenidas por este medio, el cual tiene uno de los más altos niveles de demanda a nivel mundial.

Desde sus inicios y hasta la actualidad, producto del avance tecnológico y la actividad competitiva entre los diferentes proveedores de servicios televisivos en el mundo, han surgido varios tipos de redes de transmisión de televisión, entre las que se encuentran: terrestre, por cable, por satélite y transmisión utilizando el protocolo de internet (por sus siglas en inglés IPTV).

Actualmente la televisión por cable es la que más usuarios abarca en el mundo debido a la alta competencia que satura el mercado y abarata los precios. La IPTV ha sido fruto del avance de las Tecnologías de la Informática y las Comunicaciones (TIC) y significa un avance cualitativo con respecto a la televisión cableada, ya que integra varios servicios, teniendo así el potencial suficiente para ser la más usada mundialmente, lo que no ocurre debido a que para que pueda desarrollarse de manera completa es necesario aumentar la velocidad de las conexiones actuales, sumado a esto, que su costo de adquisición es muy alto, aunque después de creada su infraestructura los gastos por mantenimiento son bajos. La IPTV es una de las formas de Televisión Digital más difundida a nivel mundial.

En Cuba se han dado algunos pasos en este aspecto, existen empresas que han comenzado a trabajar en el desarrollo de la IPTV, aunque de forma aislada como son la Empresa de Telecomunicaciones de Cuba SA.(ETECSA), TeleCable y COPEXTEL, a las cuales se les ha dificultado el trabajo por no tener un escenario de pruebas con las condiciones necesarias.

En la Universidad de las Ciencias Informáticas (UCI) la transmisión de multimedia y programas televisivos se realiza por cable coaxial, esto trae consigo problemas en

la calidad de este servicio ya que esta infraestructura es un poco antigua, de complejo mantenimiento lo que provoca problemas de deterioro en los mismos y a la vez una exhaustiva tarea de soporte.

En los últimos tiempos se ha hecho una gran inversión en aras de modernizar la red de datos de la universidad, renovándose todo el equipamiento de datos lo que posibilita que sea un escenario ideal para realizar pruebas para la implementación de IPTV.[1]

Se han dado los primeros pasos enfocados a la creación de una plataforma IPTV en la Universidad con la creación de un proyecto para el desarrollo de esta en una plataforma de software libre, esto permitiría un mejoramiento sustancial en la calidad de la gestión y la trasmisión de multimedias en la Universidad y constituye un polígono de prueba para su posterior expansión a lo largo del país.

Las multimedias constituyen una parte importante y sensible de la información que se gestiona en la UCI y es de vital importancia que esta información no sea accedida por usuarios no autorizados garantizando que sea utilizada solo con el objetivo con que fue expuesta. Para garantizar esto existen los sistemas de control que son utilizados para delimitar los niveles de acceso a los materiales audiovisuales.

Por todo lo antes expuesto se plantea como problema a resolver: ¿Cómo controlar el acceso de usuarios a las multimedias en la plataforma IPTV de la UCI?

El objetivo general del trabajo es desarrollar un sistema que controle el acceso a las multimedias publicadas en la plataforma IPTV de la UCI.

El objeto de estudio lo constituyen los sistemas de control de acceso.

Como campo de acción: Sistemas para el control de acceso a las multimedias en plataformas IPTV.

Asumiendo como idea a defender: El debido control informático sobre las multimedias en la plataforma IPTV de la UCI evitaría el acceso de personal no autorizado.

Del objetivo general se derivaron los siguientes objetivos específicos:

- Describir la evolución asociada a los sistemas de control de acceso.
- Definir los requisitos funcionales que tendrá la aplicación.
- Analizar y diseñar la aplicación para controlar el acceso a las multimedias en la plataforma IPTV de la UCI.

- Implementar la aplicación para controlar el acceso a las multimedias en la plataforma IPTV de la UCI.
- Realizar pruebas de caja negra a la aplicación de control de acceso a las multimedias en la plataforma IPTV de la UCI.

Para dar cumplimiento a los objetivos específicos se trazaron las siguientes tareas de investigación:

- Revisar la literatura existente acerca de los sistemas de control de acceso.
- Identificar las funcionalidades que conformarán la aplicación para controlar el acceso a las multimedias en la plataforma IPTV de la UCI.
- Definir la arquitectura de la aplicación para controlar el acceso a las multimedias en la plataforma IPTV de la UCI.
- Implementar la aplicación para controlar el acceso a las multimedias en la plataforma IPTV de la UCI.
- Realizar pruebas al sistema.

Aportes e impactos esperados:

- Manejar contenidos audiovisuales con mayor seguridad.
- Controlar el acceso a los contenidos de los diferentes usuarios.
- Lograr una protección de los contenidos que se encuentran en la plataforma IPTV de la UCI.

Métodos de investigación que fueron utilizados:

### **Métodos Teóricos:**

**Análisis histórico – lógico:** Se utiliza para hacer un estudio sobre la evolución y desarrollo que han tenido los sistemas de control de acceso en plataformas IPTV con el objetivo de lograr una mejor comprensión del objeto de estudio.

**Analítico – sintético:** Es usado para estudiar y analizar documentos y bibliografías de diferentes autores para poder realizar una amplia investigación sobre los elementos que se relacionan con el objeto de estudio.

**Hipotético deductivo** para el análisis y la definición de la hipótesis de la investigación que será verificada o probada en función del estudio de elementos particulares o de menor nivel de generalidad.

**Teórico Inductivo-Deductivo:** Es utilizado para determinar las funcionalidades que permiten algunos de los sistemas a través de los elementos reflejados en las imágenes de interfaces gráficas que publican en sus respectivas páginas web.

Modelación: Para crear el proceso de diseño mediante la abstracción de sus elementos fundamentales utilizando un lenguaje de modelado.

**Métodos empíricos:**

Entrevistas: consiste básicamente en una conversación planificada con personas bien informadas en determinados aspectos de una problemática específica. Se utiliza para conocer el estado del arte del tema en el mundo, Cuba y la UCI.

### **CAPÍTULO 1: Fundamentación Teórica.**

En el presente capítulo se abordarán conceptos asociados con el tema de la investigación con el fin de ofrecer un mejor entendimiento y establecer un vocabulario común. Se realiza además un estudio sobre el estado de IPTV en el mundo para conocer las tendencias actuales y futuras, las diferencias, beneficios y desventajas con respecto a otras tecnologías, así como su posicionamiento en el mercado. Se realiza además un estudio sobre los mecanismos y sistemas que existen en el mundo para controlar de alguna manera el acceso a multimedias.

#### 1.1 Conceptos asociados al dominio del problema

Televisión IP: La televisión por Internet es una variante del servicio de telecomunicación y entretenimiento que se ve día a día en los receptores de casa, su diferencia es que en lugar de que la información viaje por el aire como una señal radioeléctrica ésta se traslada en forma digital dentro de paquetes de datos junto al Internet.

La televisión IP es un conjunto de servicios multimedia (televisión, video, audio, texto, gráficos y datos) que son distribuidos por una red IP, los cuales deben poseer un nivel de calidad de servicio, seguridad, interactividad y fiabilidad.

También se puede referirse a la televisión IP utilizando la abreviatura TVIP o la abreviatura del término inglés IPTV. [2] [3]

Streaming: Se utiliza para aligerar la descarga y ejecución de audio y vídeo en la web, ya que permite escuchar y visualizar los archivos mientras se están descargando.

Existen dos formas de realizar streaming: en directo y bajo demanda. [4]

Streaming en directo: El vídeo y el audio se codifican en el acto, se retransmiten instantáneamente hacia Internet y se visualizan y escuchan en tiempo real. Es lo que ocurre con la videoconferencia y las difusiones en directo. En este caso, es el servidor el que se encarga de controlar la transmisión de los datos utilizando tecnología "push". [5]

**Streaming bajo demanda:** El contenido multimedia una vez grabado, se aloja en un servidor y puede ser visualizado bajo demanda en cualquier instante. En este caso es el cliente el que controla la transmisión y recepción del contenido multimedia utilizando tecnología “pull”. [6]

**Servidor streaming:** Se refiere a un sistema informático dedicado a la entrega de medios de transmisión o de un servicio de streaming en una web o servidor de aplicaciones. [7]

**Permisos:** Acciones o funciones que puede realizar un usuario dentro del sistema en dependencia del rol que desempeña. Así se podrá restringir o admitir el acceso de los usuarios a la información para su visualización, modificación y ejecución. Los permisos se dividen en tres tipos: escritura, lectura y ejecución. El de escritura permite al usuario escribir y modificar el archivo. El permiso de lectura permite leer el contenido del mismo. El permiso de ejecución admite al usuario ejecutar el archivo, si tiene algo para ejecutarse.

**Control de acceso:** Proceso de conceder permisos a usuarios o grupos de acceder a objetos, tales como ficheros o impresoras en la red. El control de acceso está basado en tres conceptos fundamentales: identificación, autenticación y autorización.

**Identificación:** Se denomina identificación a la acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece además que el usuario es responsable de las acciones que lleve a cabo en el sistema, lo que se relacione con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado.

**Autenticación:** Es la verificación de que el usuario que trata de identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión. Es el proceso de detectar y comprobar la identidad de una entidad de seguridad mediante el examen de las credenciales del usuario y la validación de las mismas consultando a una autoridad determinada.

**Autorización:** Es el procedimiento para determinar si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos de un sistema o aplicación. [8]

**Derechos de usuario:** Los derechos de usuario son tareas que un usuario puede llevar a cabo en un sistema o dominio del equipo, porque tiene permiso para ello. Existen dos tipos de derechos de usuario: el derecho de inicio de sesión y privilegios. Con los derechos de inicio de sesión se controla quién tiene autorización para iniciar sesión en un equipo ó sistema y el modo en que puede hacerlo, mientras que con los privilegios se supervisa el acceso a recursos de todo el sistema y se anulan grupos de permisos en objetos determinados. Estos derechos de usuario los asigna el administrador a usuarios individuales o a grupos como parte de la configuración de seguridad del sistema. [9]

**Protocolo:** Descripción formal de formatos de mensaje y de reglas que dos ordenadores deben seguir para realizar un intercambio exitoso de los mensajes. Un protocolo puede describir tanto detalles de bajo nivel entre máquinas o intercambios de alto nivel entre aplicaciones. [10]

**SGBD:** Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. [11]

### **1.2 Objeto de estudio**

**1.2.1 Surgimiento de IPTV:** Internet ha experimentado un enorme éxito como medio de comunicación: una grandísima parte de los hogares del primer mundo dispone ya de conexión. Hoy en día la red es un estándar para la comunicación, el trabajo y el entretenimiento, especialmente para el sector joven de la población. Si se atiende a la parcela del ocio no cabe duda de que muchas de las posibilidades de diversión pasan por Internet: descarga y disfrute de contenidos multimedia,

videojuegos en red, chats, etc. La tendencia no es otra que unificar todos los dispositivos de entretenimiento que pueden encontrarse en el hogar en un solo centro de ocio conectado a Internet.

El paradigma de la televisión por Internet, conocido como IPTV, surge como una alternativa avanzada al modelo tradicional. Consiste en el mantenimiento de canales que emiten contenido multimedia a través de la red. Internet permite la comunicación bidireccional entre emisor y receptor de forma natural, lo que incrementa en gran medida la libertad de elección de este último. Además tiene otras ventajas como la posibilidad de hacer "tracking", es decir, desplazarnos a lo largo de la duración del programa, almacenar el programa en el dispositivo de forma local y participar activamente y en tiempo real a través de Internet.

Esta idea está planteada desde 1995, pero su puesta en práctica no ha sido posible hasta hace bien poco, debido principalmente a su necesidad de redes con gran ancho de banda y mínima latencia que lleguen hasta los hogares. Este escollo ha sido ya superado en buena medida gracias a dos motivos principales: el gran aumento de las prestaciones de los elementos de conmutación IP (routers) y la aparición de forma comercial de técnicas que permiten ajustar y garantizar la calidad del servicio.

Las tecnologías necesarias para la implantación a gran escala de la IPTV están aún en una fase muy temprana. Es por ello que existe un gran número de líneas de investigación abiertas en distintos campos: redes, codificación de vídeo y audio, calidad del servicio y protección del contenido. La meta es obtener una tecnología de difusión mundial que sustituya a la televisión tradicional. [12]

### **1.2.2 IPTV en el Mundo**

Según investigación realizada por la consultoría Point Topic a pedido de DSL Fórum el número de suscriptores de IPTV ha crecido en un 179% en 12 meses, de julio del año pasado hasta el mismo período de este año.

La base de clientes cambió de 2,95 para 8,22 millones. El mayor crecimiento ha sido en Europa, de 1,5 para 4,9 millones en un año. El América 660 mil nuevos

## Capítulo 1: Fundamentación Teórica

---

clientes adicionados y en la región Asia/Pacífico 1,18 millón de suscriptores conquistados.

En el 2007 el número de suscriptores de IPTV en China creció en un 20,3% alcanzando los 736 mil usuarios.

Pero China aún permanece detrás de Europa, que posee más de 4 millones de usuarios.

Francia es el país con el mayor número de suscriptores, pero Hong Kong - cuando es separado de China - posee más de 800 mil usuarios. Comparado con otros, el país asiático tiene la mayor penetración en la base de suscriptores de banda ancha, sólo PCCW tiene más de 700 mil suscriptores.

Las suscripciones al servicio siguen aumentando en cada región. Se confirma una vez más que en 2014 serán más de 65 millones los abonados a esta tecnología. Esta proyección ha sido dada a conocer por la firma de investigaciones y consultorías Strategy Analytics.

Durante el año 2009 los usuarios de IPTV representaron 4,9% de los suscriptores de televisión por cable, frente a sólo un 0,01% a finales de 2003. Así mismo el número de abonados de IPTV creció un 46,9% entre el año 2008 y el año 2009, mientras los ingresos aumentaron un 69%, llegando a US\$ 8,7 billones.

En cuanto a suscriptores por región geográfica, Europa occidental lleva la delantera frente a otras regiones con 13,2 millones de suscriptores, seguida por Asia / Pacífico y América del Norte, con 9,6 millones y 6 millones respectivamente.[13]

**La tabla 1 muestra los 10 países con mayor penetración y suscriptores:**

País (miles)	Suscriptores	Penetración
Francia	1.410	10,17%
Hong Kong	920	55,93%
China	721	1,29%
España	428	6,07%
Japón	304	1,14

Taiwan	269	6,04%
Italia	212	2,46%
Holanda	167.5	3,26%
Bélgica	150	6,39%
Suecia	145	6.11%

**Tabla 1: Suscriptores de IPTV con mayor penetración.**

### **1.2.3. IPTV en la UCI:**

En la UCI se está llevando a cabo la tarea de implementar una plataforma IPTV para perfeccionar los servicios que brinda la dirección de televisión universitaria (DTU), ofreciendo grandes ventajas tanto para la dirección como para la comunidad universitaria, eliminando problemas que existían en la trasmisión y recepción de materiales audiovisuales.

La protección del contenido es una prioridad fundamental para las plataformas IPTV, para evitar la visualización y descarga de las medias por usuarios no autorizados. Una plataforma IPTV que quiera tener éxito deberá sin duda ofrecer un mecanismo de protección fiable y seguro para garantizar que los contenidos que emite sólo sean manipulados por los clientes con facultades para ello por lo que se propone implementar un mecanismo para controlar el acceso a las medias de la plataforma en la Universidad de la Ciencias Informáticas.

### **1.2.4 Control de acceso:**

La informática es un largo proceso a través de la historia y desde sus inicios se ha tratado de controlar el acceso a los recursos de información. La tecnología utilizada para este control de accesos ha evolucionado con los propios sistemas de información protegidos.

En un mundo cada vez más dependiente de las redes, es vital no sólo proteger el acceso a los recursos por los 'malos', sino también evitar una manipulación accidental con fatales consecuencias.

La información puede existir en muchas formas, impresa o escrita en papel, almacenada digitalmente, transmitida por correo postal o utilizando medios digitales, presentada en imágenes o expuesta en una conversación. Cualquiera sea la forma que adquiere la información es un recurso que, como el resto de los activos importantes tiene un gran valor, siendo a veces incalculable, por contener

la “vida” de una organización; es por eso que debe ser debidamente protegida independientemente de los medios por los cuales se distribuye o almacena. En esta “sociedad de la información” persisten las razones y motivos para mantener mecanismos de control de acceso sobre las áreas (seguridad física) y la información (seguridad lógica) que se desea proteger. La seguridad física describe las medidas que previenen o detienen a intrusos antes de que accedan a una instalación, recurso o información almacenada en medios físicos. Los controles de acceso físico tienen una gran importancia puesto que si alguien que desee atacar un sistema tiene acceso físico al mismo, el resto de las medidas de seguridad implantadas se convierten en inútiles. Muchos ataques serían entonces triviales, como por ejemplo los de denegación de servicio; si es posible apagar un servidor que proporciona un servicio es evidente que nadie podrá utilizarlo. Luego de ver como un sistema puede verse afectado por la falta de seguridad física, es importante recalcar que la mayoría de los daños que puede sufrir una organización no será sobre los medios físicos sino contra la información. La seguridad de la información, es la preservación de la confidencialidad, integridad y disponibilidad y se logra implementando un conjunto adecuado de controles, que abarcan políticas, prácticas, procedimientos, estructuras organizacionales y funciones del software. Entre las medidas de protección de la información esenciales para una institución se encuentra el control de acceso lógico a los recursos.

### **1.2.5 Modelos de control de acceso:**

Existen tradicionalmente dos tipos básicos de controles de acceso con filosofías diametralmente opuestas:

**1.2.5.1 Modelo de control de acceso obligatorio (MAC):** En este modelo es el sistema quién protege los recursos, comparando las etiquetas del sujeto que accede frente al recurso accedido, o sea, la autorización para que un sujeto acceda a un objeto depende de los niveles de seguridad que tengan, ya que estos indican que permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto. Todos los sujetos y objetos del sistema tienen una etiqueta de seguridad.

**1.3.5.2 Modelo de control de acceso discrecional (DAC):** Este modelo se ha venido usando profusamente en sistemas operativos de propósito general con

clasificación de seguridad TCSEC (Information Technology Security Evaluation Criteria) C1 o superior, y en virtualmente todos los sistemas de bases de datos y sistemas de comunicaciones de propósito comercial. En este modelo un usuario bien identificado, típicamente el creador o propietario del recurso, decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema. Lo esencial es que el propietario del recurso puede cederlo a un tercero.

En sus inicios estos sistemas eran excesivamente simples, al permitir un conjunto limitado de operaciones posibles sobre un recurso (rwx por propietario, grupo o resto de usuarios, como en Unix), pero rápidamente se añadieron las famosas listas de control de accesos (ACLs), listas de usuarios y grupos con sus permisos específicos.

Los modelos DAC y MAC son inadecuados para cubrir las necesidades de la mayor parte de las organizaciones. El modelo DAC es demasiado débil para controlar el acceso a los recursos de información de forma efectiva, en tanto que el MAC es demasiado rígido. Desde los 80 se ha propuesto el modelo de control de accesos basado en roles (RBAC), como intento de unificar los modelos clásicos DAC y MAC, consiguiendo un sistema donde el sistema impone el control de accesos, pero sin las restricciones rígidas impuestas por las etiquetas de seguridad.

**1.2.5.3 Control de acceso basado en Roles (RBAC)** En este modelo los usuarios son asignados a uno o varios roles mientras que los permisos y privilegios se asignan a estos roles. Por tanto, las políticas de control de accesos basado en roles regulan el acceso de los usuarios a la información en términos de sus actividades y funciones de trabajo (roles), representándose así de forma natural la estructura de las organizaciones. Este modelo permite la construcción jerárquica de estas políticas de acceso, por herencia o especialización, por ello tiene el potencial de reducir la complejidad y el coste de la administración de seguridad en entornos heterogéneos.

Dada la alta integración entre los roles y las responsabilidades de los usuarios, se pueden seguir los principios del mínimo privilegio y de la separación de responsabilidades. Estos principios son vitales para alcanzar el objetivo de integridad, al requerir que a un usuario no se le otorguen mayores privilegios que

los necesarios para efectuar su trabajo. El uso de RBAC para administrar los privilegios de los usuarios en un sistema o aplicación es ampliamente aceptado como una práctica recomendada. Algunos sistemas como Microsoft Active Directory, SELinux, Solaris, Oracle DBMS, PostgreSQL, y otros, lo implementan de alguna forma. A continuación se lista un conjunto de principios básicos a tener en cuenta en la implementación de RBAC:

- Un usuario tiene acceso a los objetos de acuerdo al rol que tenga asignado.
- Los roles son definidos en base a funciones de trabajo.
- Los permisos son definidos en función de la autoridad y responsabilidad que se asume en una función de trabajo.
- Las operaciones sobre un objeto son invocadas de acuerdo a los permisos.

[14]

### **1.3 Análisis de otras soluciones existentes.**

En el mundo se han implementado algunos sistemas para controlar el acceso a las multimedias, así como protegerlas del uso indebido. A continuación se muestran algunos ejemplos de estos programas con sus principales características:

**Helix Security Manager:** Es un sistema de seguridad implementado para proporcionar un acceso seguro a los contenidos disponibles para el streaming o descargar a través del servidor Helix, Proxy de Helix, Helix Mobile Server y Helix Mobile proxy, utiliza tokens únicos para cada cliente que establece comunicación solicitando la entrega de un contenido, asegurando que sólo los usuarios autorizados puedan tener acceso a la media solicitada. Helix Security Manager impide enlace de alta elevación y garantiza la autenticación basada en usuarios autorizados para que sólo estos tengan acceso a su contenido. Helix Security Manager funciona con LDAP de Active Directory y la autenticación de los sistemas de Shibboleth y soporta todos los RealMedia, Windows Media, Quick Time, MPEG-4 y los formatos de medios basados en Flash. [15]

**Windows Media Rights Manager:** Permite a los proveedores de contenido ofrecer canciones, vídeos y otros contenidos de medios digitales a través de Internet en un formato protegido, el archivo cifrado. Windows Media Rights Manager ayuda a proteger los medios de comunicación digitales (tales como canciones y vídeos) por un embalaje de archivos multimedia digitales. Un archivo empaquetado de medios contiene una versión de un archivo multimedia que se ha cifrado y se cerró con

una "clave". El resultado es un archivo empaquetado que sólo puede ser usado por una persona que haya obtenido una licencia. [16]

**CoreMedia DRM:** Es una completa infraestructura DRM que apoya los procesos de distribución que protegen el valor de los servicios de contenido. Esto incluye la protección y la entrega de contenido y gestión y entrega de licencias. CoreMedia DRM es compatible con una variedad de modelos de negocios diferentes, incluyendo pago por descarga, pago por uso, suscripción, Vista previa, Promoción, superdistribución, la Participación y el Contenido en movimiento. CoreMedia DRM proporciona un componente de prestación de servicios que actúe como punto de acceso único para todos los artículos contenidos protegidos y no protegidos. Puede ser fácilmente escalable para implementar portales de alto tráfico de descarga. [17]

### **1.4 Metodologías a utilizar, tendencias y tecnologías actuales.**

Para la realización de una aplicación informática se deben definir los tipos de tecnologías a utilizar, así como las herramientas y las metodologías que serán de mayor utilidad para su implementación. A continuación se explicarán brevemente los detalles de cada una de estas tecnologías seleccionadas para el desarrollo e implementación de la aplicación.

#### **1.4.1 Metodologías de desarrollo.**

Hoy en día a los desarrolladores de grandes aplicaciones se les dificulta realizar una planificación organizada del trabajo, mantener las políticas del trabajo en equipo y a la vez asumir los adelantos en la informatización de los procesos productivos. Es por ello que organizaciones y empresas requieran cada vez más de aplicaciones confiables y de mejor calidad tanto para su desarrollo como para su mantenimiento. Para contribuir con las necesidades de estas organizaciones y empresas la comunidad desarrolladora de software ha estandarizado un conjunto de procedimientos ya definidos con el fin de complementar una metodología de trabajo a seguir para el desarrollo de productos software.

##### **1.4.1.1 Extreme Programming(XP)**

Metodología más destacada dentro de los procesos ágiles de desarrollo de software, utilizada para el desarrollo de proyectos cortos. Sólo se requiere de un equipo de desarrollo pequeño y en pareja, trabajando en el mismo local físico. Se

centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Características de XP, la metodología se basa en:

- Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir.
- Prefabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

En todas las iteraciones del ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

### 1.4.1.2 Rational Unified Process

*Rational Unified Process* (por sus siglas en inglés RUP) es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de

organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

En RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba, constituyendo un elemento integrador y una guía del trabajo.

RUP es un proceso bien definido, estructurado y adaptable específicamente a cada proyecto. Se define a partir de tres características esenciales: estar dirigido por los Casos de Uso, centrado en la arquitectura y ser iterativo e incremental.

RUP es un proceso iterativo e incremental que se encarga de dividir el trabajo en partes más pequeñas o en mini proyectos, permitiendo que se logre un equilibrio entre la arquitectura y casos de uso durante cada mini proyecto. En cada iteración del proyecto o mini proyecto como se ha definido se obtiene un incremento, provocando un aumento del producto.

Este proceso de desarrollo está definido por 4 fases fundamentales:

- Inicio: el objetivo en esta fase es determinar la visión del proyecto.
- Elaboración: en esta fase el objetivo es determinar la arquitectura óptima.
- Construcción: en esta fase el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición: el objetivo es llegar a obtener la liberación del proyecto.

Cada fase del RUP concluye con un hito bien definido, en el cual se deben tomar determinadas decisiones y alcanzar las metas clave antes de pasar a la siguiente fase. Los hitos para cada una de las fases son:

- Inicio: Visión de los objetivos.
- Elaboración: Prototipo de la arquitectura.
- Construcción: Capacidad operacional inicial.
- Transición: Liberación del producto.

## Capítulo 1: Fundamentación Teórica

---

- Los artefactos son productos tangibles del proyecto que son producidos, modificados
- y usados por las actividades. Pueden ser modelos, elementos dentro del modelo,

código fuente y ejecutables.

RUP utiliza como lenguaje de modelado: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*).

Utiliza herramientas de modelado visual que facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. Ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrolladores se crea una documentación también común, de forma tal que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

UML se ha convertido en un estándar para representar y modelar la información con la que se trabaja en las fases de análisis y especialmente en la fase de diseño. La selección de una metodología de desarrollo para lograr un producto con mayor calidad, es una de las dificultades que presentan los desarrolladores de software actualmente. El Proceso Unificado de Software y Programación Extrema (XP por sus siglas en inglés) son los procesos más utilizados y mejor documentados a nivel mundial. En la Tabla 2 se establece una comparación entre estas dos metodologías para determinar cuál de ellas es la más indicada para el desarrollo de la aplicación.

	<b>Proceso Unificado de Software(RUP)</b>	<b>Programación Extrema(XP)</b>
Ventajas	<ul style="list-style-type: none"><li>-Mayor documentación.</li><li>- Verificar calidad de software.</li><li>-Configuración y control de cambios.</li><li>-Es modelado, guiado por casos de uso.</li></ul>	<ul style="list-style-type: none"><li>-Comunicación con el cliente.</li><li>-Ligero.</li><li>-Simplicidad.</li></ul>

	-Es centrado en arquitectura, guiado por riesgos.	
Carga de trabajo	<ul style="list-style-type: none"> <li>-Es un proceso pesado, basado en mucha documentación.</li> <li>-Existen diferentes elementos de planificación con los que se controla el desarrollo de software.</li> <li>-Define qué artefactos son necesarios para poder realizar una actividad y qué artefactos se deberán crear durante dicha actividad.</li> </ul>	<ul style="list-style-type: none"> <li>-Es un proceso ligero que no presenta demasiadas tareas organizativas sobre los desarrolladores y pobre en cuanto a la documentación.</li> <li>-En el desarrollo de este proceso es mas importante la entrega al cliente del software que necesita que las funcionalidades que quedan por implementar.</li> <li>-Los cambios se acuerdan directamente con el representante del cliente, con los cuales se ajusta el plan de iteraciones y el de liberación y se toman nuevas direcciones en el desarrollo.</li> </ul>
Principal objetivo	-Mantener el control.	- Código.
Pruebas	-Después de terminar la fase de transición.	- Desde el comienzo del proyecto.
Miembros	-Los grupos de trabajo.	- El cliente es un miembro del equipo de trabajo
Desarrollo	-Guiado por la terminación de cada fase.	- Guiado por las pruebas.

**Tabla2 Comparación entre RUP y XP.**

### ¿Por qué RUP?

RUP lejos de ser un sistema con pasos firmemente establecidos, es un conjunto de metodologías lo suficientemente flexibles como para adaptar el proceso de desarrollo al contexto y a las necesidades de la organización cliente. El proceso

iterativo de RUP permite que en cada iteración se analice la opinión de los clientes y la estabilidad y eficacia del producto. Esta es una gran ventaja para cualquier proyecto, pues admite un incesante refinamiento del producto, así como una continua mitigación de los riesgos involucrados. Esto evidencia su capacidad de asegurar la calidad del producto final. RUP proporciona muchas ventajas sobre XP, le da énfasis en los requisitos y el diseño.

La ventaja principal de RUP es que se basa todo en las mejores prácticas que se han intentado y se han probado en el campo.

RUP ha sido seleccionada como metodología para el desarrollo de la aplicación teniendo en cuenta todas las ventajas antes mencionadas, haciendo hincapié en que se necesita un estudio bien documentado ya que la aplicación es un módulo para la seguridad de una plataforma para la gestión y transmisión de IPTV en la UCI, sin experiencia previa, la cual pudiera ir enriqueciéndose producto a futuras necesidades. [18][19]

### **1.4.2 Herramientas de modelado.**

En las últimas décadas se ha trabajado en el área de desarrollo de sistemas para encontrar técnicas que permitan incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software. Hoy en día la tecnología *Computer Aided Software Engineering* (CASE, por sus siglas en inglés) reemplaza al papel y al lápiz por el ordenador para transformar la actividad de desarrollar software en un proceso automatizado.

#### **1.4.2.1 Rational Rose.**

Es una de las herramientas más poderosas de software para el modelado visual mediante UML el cual permite realizar el análisis y diseño de sistemas basados en objetos, además de utilizarse para modelado de sistemas antes de proceder a construirlos. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Esta herramienta mantiene la consistencia de los modelos del sistema software y el chequeo de las sintaxis UML. Genera la documentación automáticamente como también los códigos a partir de los modelos. Presenta ingeniería inversa y calidad de análisis del código. [20]

### 1.4.2.2 Visual Paradigm 6.4.

Es una herramienta para visualizar y diseñar elementos de software, para ello hace uso del lenguaje de modelado UML. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Facilita la interoperabilidad con otras herramientas CASE como el Rational Rose y puede ser integrada con herramientas java. Posee además gran facilidad de uso.

Para el desarrollo de la aplicación se selecciona la herramienta de modelado Visual Paradigm debido a que es multiplataforma y brinda muchas facilidades para el diseño de los diagramas necesarios y su documentación. Apoya a un conjunto de idiomas tanto en la generación de código como en la ingeniería inversa de los mismos, permitiendo la importación y exportación de archivos XML de diferentes versiones. Emplea una rápida respuesta y con poca memoria, requisito que le permite manejar grandes y complicadas estructuras de un proyecto en una forma muy eficiente y, sin embargo, sólo requiere de una configuración de escritorio.

Visual Paradigm acelera el desarrollo de aplicaciones, ya que sirve de puente visual entre arquitectos, analistas y diseñadores de sistemas de información, haciendo el trabajo más fácil y dinámico. [21]

### 1.4.3 Lenguaje de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora, además permite a uno o más programadores especificar de manera precisa sobre qué datos de una computadora desea operar. Todo esto consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

### PHP 5.3.2

PHP 5.3.2 es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML.

¿Cómo Funciona PHP?

A diferencia de Java o JavaScript que se ejecutan en el navegador PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap).

Al ser PHP 5.3.2 un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- 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 reparen rápidamente.

- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se puede realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

### JavaScript

Lenguaje de programación que ha permitido el gran desarrollo de la web, ha sido el avance más significativo en el logro de páginas web dinámicas y exactas en cuanto a posición y presentación de su contenido, es un lenguaje robusto y a la vez ligero, el cual a pesar de ser no orientado a objetos permite implementar varias de las características de este paradigma de programación. JavaScript es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del ratón (mouse), utilización de teclas, cargas de páginas, entre otros. Este es un lenguaje interpretado, no requiere compilación y es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera y Mozilla Firefox.[22]

Para el desarrollo de la aplicación se decidió utilizar como lenguaje de programación PHP 5.3.2 ya que brinda facilidades para el trabajo en el servidor, ofrece la posibilidad de crear páginas web dinámicas de manera rápida y sencilla y posee las librerías necesarias para la implementación del sistema.

#### 1.4.4 Entorno de desarrollo

##### NetBeans 6.9M1.

El NetBeans es un Entorno Integrado de Desarrollo (IDE), siendo un producto libre y gratuito sin restricciones de uso. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. A continuación se describen algunas de sus características:

- **Creación de Proyectos PHP**

NetBeans provee una estructura para los proyectos que se pueden crear junto a este IDE, propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. NetBeans posee además un sistema para examinar todo los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

### ➤ Editor de Código Fuente

El editor de PHP, es mucho más ágil y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee la última versión de PHP, la 5.3.

### ➤ Depuración de PHP

NetBeans integra muy bien la utilización Xdebug, gracias a esto se puede inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código.

El IDE de NetBeans para PHP también ofrece la línea de comandos de depuración: La salida del programa PHP aparece en una pantalla de línea de comandos en el IDE de sí mismo y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.

### ➤ Integración con Sistemas de Control de Versiones

Una de las condiciones necesarias para los proyectos es la posibilidad de contar con la integración de sistemas de control de versiones, tales como SVN, CVS, Mercurial y Git.

Desde el editor es posible realizar la administración de estos sistemas versionados, sus commit, branch, importar, exportar, revert, clonar, etc. [23]

## 1.4.5 Sistema de control de versiones

### **Subversion.**

Subversion es un sistema de control de versiones libre y de código fuente abierto. Maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Ésto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada por la pérdida de ese conducto único.

Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros, código fuente, vídeo digital y más allá. [24]

### 1.4.6 Gestor de Base de Datos

#### PostgreSQL 8.4.

PostgreSQL 8.4 es un potente motor de bases de datos. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. [19]

A continuación se enumeran las principales características de este gestor de bases de datos:

- **DBMS Objeto-Relacional**

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.

- **Altamente Extensible**

PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.

- **Soporte SQL Comprensivo**

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

- **Integridad Referencial**

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

### ➤ **API Flexible**

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++, y Pike.

### ➤ **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.

### ➤ **Cliente/Servidor**

PostgreSQL 8.4 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.

- Su administración se basa en usuarios y privilegios.
- Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc. Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.
- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida. [25][26][27]

Se determinó utilizar PostgreSQL 8.4 para el desarrollo de la aplicación por las ventajas que brinda en el trabajo con bases de datos en plataformas libres y por haber sido definido por el cliente como sistema gestor de base de datos.

### 1.4.6 Servidor Web

#### Apache2

Apache es el servidor web más usado en sistemas Linux. Los servidores web se usan para servir páginas web solicitadas por equipos cliente. Los clientes normalmente solicitan y muestran páginas web mediante el uso de navegadores web como Firefox, Opera o Mozilla.

El protocolo más comúnmente utilizado para ver páginas Web es el Hyper Text Transfer Protocol (HTTP). Protocolos como el Hyper Text Transfer Protocol sobre Secure Sockets Layer (HTTPS), y File Transfer Protocol (FTP), un protocolo para subir y descargar archivos, también son soportados.

Algunas características de este servidor son:

- Estabilidad: Ha probado su estabilidad y funcionamiento en una inmensa cantidad de proyectos.
- Independencia de la plataforma: Está disponible en variedad de plataformas.
- Código abierto: Esto permite por ejemplo, producir binarios para plataformas en las que no hay un binario oficial o compilarlo para un mejor rendimiento en una plataforma específica. [28][29]

Luego de hacer un análisis de las metodologías y tecnologías necesarias y actuales se decide utilizar NetBeans 6.9M1 como IDE, haciendo uso de lenguaje de programación Php 5.3.2. Como metodología de desarrollo se ha definido utilizar RUP y como herramienta de modelado Visual Paradigm 3.4. Se utilizará como Sistema Gestor de Base de Datos(SGBD) PostgreSQL 8.4.

### 1.5 Conclusiones parciales del capítulo

Con el desarrollo de este capítulo se logró una mejor comprensión de los conceptos relacionados con el dominio del problema, se investigó sobre los mecanismos de control de acceso existentes definiendo el RBAC como mecanismo a utilizar ya que es una solución factible para garantizar la seguridad en el sistema, se estudió la situación actual de la IPTV en el mundo y en la UCI, elementos fundamentales para darle solución al problema planteado y por último se definieron las herramientas a utilizar para la implementación de la aplicación teniendo en cuenta que estas son de suma importancia en el éxito de un producto.

### **CAPÍTULO 2: Presentación de la solución propuesta.**

En este capítulo se realiza un levantamiento de los requisitos tanto funcionales como no funcionales del problema. Incluye una detallada descripción del sistema propuesto. Se confecciona el modelo de dominio y los casos de uso con su descripción permitiendo una mejor comprensión del escenario sobre el que se encuentra el sistema. Se muestran también los prototipos de interfaz de usuario correspondientes.

#### **2.1 Entorno donde trabajará el sistema**

##### **2.1.1 Conceptos principales del entorno**

**Directorio de personas (LDAP):** Protocolo Ligero de Acceso a Directorios, por sus siglas en inglés *Lightweight Directory Access* permite a un cliente buscar a través de grandes bases de datos. Hace referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. [30]

**Rol:** Función o papel que desempeña una persona o grupo en cualquier actividad.

**Usuario:** aquella persona que utiliza un dispositivo o un ordenador y realiza múltiples operaciones con distintos propósitos.

**Encriptación:** Encriptación es el proceso de tomar los datos (llamado *texto limpio*) y una clave criptográfica y producir un texto encriptado, que son unos datos sin significado para cualquiera que no conozca la clave. [31]

**SSL:** Siglas de *Secure Socket Layer*. Es un protocolo que proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

**Flumotion:** Servidor de streaming compatible con los formatos más utilizados en Internet (Flash, Windows Media, MP3, H264, AAC, AAC+...) que incluye la adquisición en su diseño descentralizado. Flumotion Streaming Server permite una gran escalabilidad y su administración es sencilla. Este software utiliza una tecnología homogénea que funciona sobre Linux. [32]

**Componente:** Etiqueta XML que se encuentra en el archivo de configuración del servidor streaming Flumotion, mediante ellos se ejecutan todas las funcionalidades del servidor , cada componente se encarga de una funcionalidad en específico.

**Bouncer:** Es el componente que se encarga de la seguridad en el servidor streaming Flumotion.

**Planet:** Archivo XML que se encarga de la configuración del servidor streaming Flumotion.

**Fichero XML:** Ficheros de texto, que en principio está en código Unicode, pero se pueden usar otros alfabetos.

### 2.1.2 Características del servidor streaming Flumotion.

Flumotion puede transmitir los contenidos audiovisuales tanto en vivo como bajo demanda en una gran variedad de formatos de audio y video.

Además soporta un amplio rango de dispositivos de entrada gracias a su uso de Linux, Gstreamer y otras aplicaciones de código abierto. Cuenta con una interfaz de usuario simple que permite publicar contenidos rápidamente y de forma sencilla e intuitiva. Permite que el procesamiento sea distribuido a través de numerosas computadoras, por lo que la plataforma se puede escalar para manipular mas usuarios y streaming tanto en vivo como bajo demanda. Su arquitectura de código abierto lo hace mas eficiente y mas flexible que otros sistemas, haciendo mejor uso del hardware.

Flumotion se configura para sus diferentes funciones a través de un fichero XML llamado planet ubicado en etc/flumotion/manager/default/planet. En este fichero se encuentran componentes con propiedades específicas, el componente bouncer es el encargado de la seguridad, existen tres tipos: htpasswdcrypt, ical-bouncer e ip-bouncer para controlar el acceso mediante usuarios, fechas o direcciones ip respectivamente. Los bouncer se pueden asignar a cualquier componente dentro del archivo de configuración, principalmente a los que permiten el streaming bajo demanda y en directo con el objetivo de controlar el acceso a las publicaciones. La forma mas común es la restricción por usuario, para esto a cada bouncer de tipo htpasswdcrypt se le asigna un fichero en el cual se almacenan los usuarios y las contraseñas de forma encriptada de forma tal que solo podrán acceder a las

publicaciones los usuarios que se encuentren en el fichero relacionado con el bouncer que se le asignó a dichas publicaciones. La administración de usuarios y contraseñas se realiza a través del comando htpassword.

### 2.1.3 Eventos principales del entorno

Inicialmente aparece una interfaz principal donde el usuario tiene la posibilidad de autenticarse en el sitio, luego este le brinda la posibilidad de ejecutar las funcionalidades implementadas en el sistema tales como: Gestionar Usuarios, Gestionar Grupo, Gestionar Publicaciones, Gestionar la relación usuario-grupo, Gestionar la relación grupo- publicación, Gestionar Permisos, Gestionar Ficheros, Buscar Elementos, Actualizar los grupos desde el LDAP y también podrá salir de la sesión cuando lo desee.

El Sistema de control de acceso para la plataforma IPTV-UCI está diseñado para administrar los principales elementos de forma rápida y segura que intervienen en el proceso que realiza el servidor streaming Flumotion para restringir el acceso a las multimedias que el mismo publica. El servidor es configurable en su totalidad, dicha configuración se lleva a cabo desde un archivo de tipo xml, el cual permitirá ejecutar todas y cada una de las funcionalidades que brinda el servidor, esta configuración se lleva a cabo a través de los componentes que no son más que nodos xml <component> donde cada uno está asociado a una funcionalidad en específico, los bouncer son los componentes encargados de la seguridad del resto de los componentes y del servidor en general. Un bouncer puede ser asignado a uno o muchos componentes y cada componente puede tener asignado uno o varios bouncer, el Sistema de control de acceso para la plataforma IPTV-UCI permitirá almacenar usuarios y grupos en una base de datos, los grupos pueden ser locales o los existentes en el LDAP de la universidad. Los usuarios son locales y solo podrán formar parte de los grupos de igual distinción. El sistema podrá gestionar tanto usuarios como grupos, además permitirá realizar las distribuciones de los usuarios en los diferentes grupos, brindará además la posibilidad de gestionar las diferentes publicaciones que la plataforma brinda. Además el sistema será capaz de cargar todas las publicaciones que se encuentren activas en el fichero de configuración y gestionarlas de igual manera. También se podrán definir qué grupos podrán visualizar una publicación en específico y asignarle a un componente tantos bouncer como se deseen, así como eliminarlo, estas operaciones se efectuarán directamente en el archivo de configuración del servidor

streaming Flumotion. Los bouncer pueden ser de tres tipos: htpasswdcrypt, ical-bouncer e ip-bouncer, el htpasswdcrypt va a tener asociado un fichero físico que es donde se guardarán los usuarios y las contraseñas encriptados; cuando se le asigna dicho bouncer a un componente encargado de transmitir un flujo de video tanto en vivo como bajo demanda solo podrán acceder a este flujo los usuarios que se encuentren en el fichero asociado a él. La aplicación brindará la posibilidad de gestionar estos ficheros, quienes estarán asociados unívocamente con un grupo de forma tal que en este fichero se encuentren los usuarios del grupo al que está relacionado el fichero. El sistema también es capaz de cargar a partir de los bouncer existentes de tipo htpasswdcrypt los ficheros y guardarlos en la base de datos para su posterior gestión.

### 2.1.4 Diagrama de clases del modelo de dominio

El Modelo de Dominio ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar una aplicación, da una idea de manera general de los conceptos (objetos) significativos en el dominio del problema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de dominio se debe concebir como un diccionario visual de abstracciones que será utilizado en fases posteriores y cuya función principal es ayudar a comprender el problema a tratar, es una de las vistas del modelo de requerimientos. Las clases en el modelo del dominio son las abstracciones principales del sistema y el modelo se representa como un diagrama de clases con las abstracciones identificadas durante el análisis.

La siguiente figura representa el modelo de dominio para el "Sistema de control de acceso para la plataforma IPTV-UCI". La clase "usuario" representa a los usuarios que se gestionarán en el servidor, quienes están asignados a ningún o varios grupos, simbolizados por la clase "grupo", a la vez un grupo puede tener ningún o varios usuarios. Un grupo tiene permisos, esta clase se refiere a la disponibilidad que van a tener las multimedias para un grupo. La clase "multimedia" se refiere a los archivos multimedia que serán solicitados por un usuario, estos se encuentran dentro de los diferentes directorios asociados al servidor streaming Flumotion representados por las clases "directorio" y "servidor streaming" respectivamente.

Dicho servidor puede tener ningún o varios directorios, y estos a la vez pueden contener ningún o varios directorios.

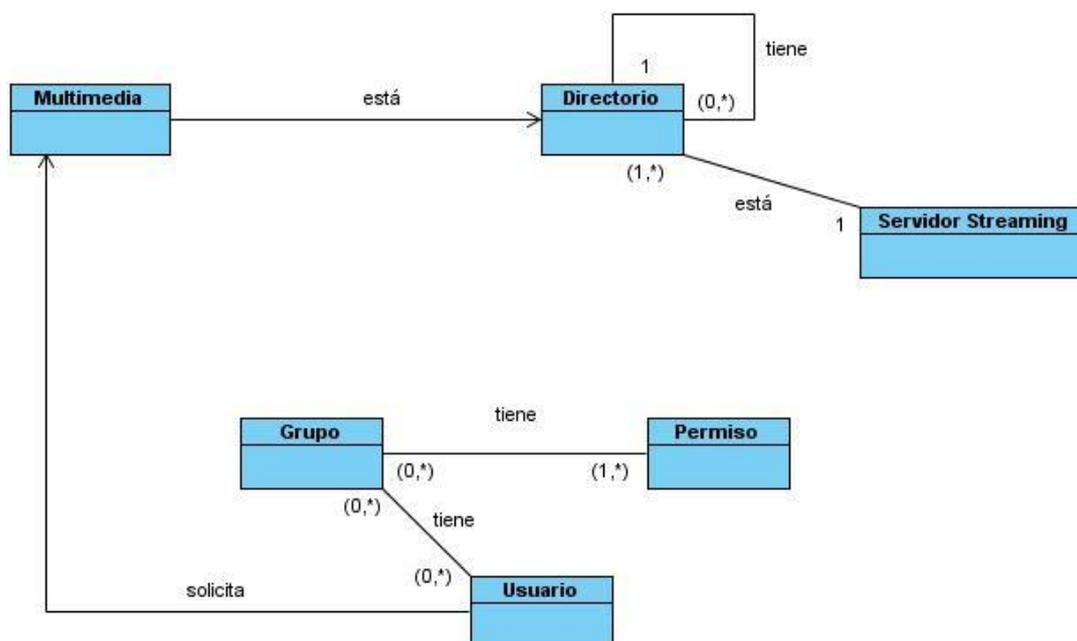


Figura 1. Modelo de Dominio

### 2.1.5 Glosario de Términos del Dominio

#### Directorio

Un directorio es un sistema que la computadora utiliza para organizar ficheros a partir de información específica. Los directorios pueden estar organizados jerárquicamente de forma que los ficheros aparezcan de distintas maneras, como por ejemplo el orden en que fueron creados, por orden alfabético, según el tipo, etc.

#### Multimedia

Integración en un mismo soporte digital de diferentes “medios” o tipos de información: texto, imágenes, vídeo, sonido.

### 2.2 Requerimientos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A continuación se muestra un listado de estos requisitos.

1. Solicitar datos de autenticación

2. Validar datos de autenticación.
3. Gestionar usuario
  - 3.1 Adicionar usuario
  - 3.2 Mostrar usuario
  - 3.3 Modificar usuario
  - 3.4 Eliminar usuario
4. Gestionar grupos locales
  - 4.1 Adicionar grupo
  - 4.2 Mostrar grupo
  - 4.3 Modificar grupo
  - 4.4 Eliminar grupo
5. Gestionar publicación
  - 5.1 Adicionar publicación
  - 5.2 Mostrar publicación
  - 5.3 Modificar publicación
  - 5.4 Eliminar publicación
6. Gestionar usuario\_grupo
  - 6.1 Adicionar usuario\_grupo
  - 6.2 Mostrar usuarios\_grupos
  - 6.3 Modificar usuario\_grupo
  - 6.4 Eliminar usuario\_grupo
7. Gestionar grupo\_publicación
  - 7.1 Adicionar grupo\_publicación
  - 7.2 Mostrar grupo\_publicación
  - 7.3 Modificar grupo\_publicación
  - 7.4 Eliminar grupo\_publicación
8. Gestionar permisos
  - 8.1 Asignar permisos
  - 8.2 Mostrar permisos
  - 8.3 Eliminar permisos
9. Gestionar Fichero
  - 9.1 Adicionar fichero
  - 9.2 Mostrar fichero
  - 9.3 Modificar fichero

- 9.4 Eliminar fichero
- 10. Actualizar grupos de LDAP
- 11. Buscar elemento
- 12. Cerrar sesión

### **2.3 Requerimientos No Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales. [33]

#### **2.3.1 Restricciones en el diseño y la implementación:**

Para crear la aplicación se utilizará:

- La Implementación se realizará utilizando el lenguaje de programación PHP5.3.2 como módulo del servidor web Apache 2 y como sistema gestor de base de datos PostgreSQL 8.4.
- Para el desarrollo del sistema debe ser utilizada la metodología RUP, usando como lenguaje de modelación UML y como herramienta Case para llevarlo su modelado Visual Paradigm 3.4.
- Se utilizará como IDE para la implementación del sistema Netbeans 6.9M1.
- Se utilizará Flumotion como servidor streaming.
- Para la implementación de la aplicación Web deberá utilizarse el estilo Modelo-Vista-Controlador.

#### **2.3.2 Apariencia o interfaz externa:**

- El sistema debe poseer una interfaz amigable al usuario.
- La interfaz gráfica del programa deberá ser lo más simple, legible e interactiva posible para facilitar la navegación dentro del sistema y que permita una fácil comprensión y utilización del mismo.

#### **2.3.3 Seguridad:**

- Cada usuario realizará operaciones en la aplicación en dependencia de sus privilegios o niveles de acceso.

- Sólo el administrador acreditado con su usuario y contraseña podrá realizar modificaciones en el sistema.
- Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto plano, como es el caso de las contraseñas. Se guardará encriptada esta información en la base de datos utilizando para ello crypt como algoritmo de encriptación.
- La seguridad del sistema debe estar regida por las Políticas de Seguridad Informática de la Universidad de las Ciencias Informáticas.
- El acceso al Sistema debe estar restringido por el uso de claves asignadas al administrador. Sólo podrán ingresar al sistema las personas que estén registradas.
- Respecto a la confidencialidad, el sistema debe estar en capacidad de rechazar accesos no autorizados a la información y proveer los servicios requeridos por los usuarios legítimos del sistema.
- Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

### **2.3.3.1 Confiabilidad:**

- Todas las salidas del sistema tienen que tener el 100% de veracidad y precisión.
- Deben existir sistemas de respaldo eléctrico en los locales de los servidores para mantener la disponibilidad de los servicios.

### **2.3.4 Rendimiento:**

- El sistema deberá responder en el mínimo de tiempo posible ante las solicitudes por parte de los usuarios y en el procesamiento de la información.

### **2.3.5 Portabilidad:**

- Al sistema se debe acceder desde cualquier plataforma.

### 2.3.6 Requerimientos de Usabilidad:

- El sistema no puede ser accedido directamente, sino a través de una interfaz diseñada para estos propósitos.
- Los mensajes de error deben ser reportados por la propia aplicación en la medida de las posibilidades y no por el Sistema Operativo.
- El software será usado por estudiantes, profesores y trabajadores de la Universidad de las Ciencias Informáticas.
- Deberá visualizarse en los navegadores más utilizados (Mozilla Firefox, Internet Explorer, Opera).

### 2.3.7 Requerimientos de Soporte:

- El uso de este software no necesita una preparación previa para que el usuario pueda navegar en el mismo.

### 2.3.8 Requerimientos Políticos:

- La aplicación debe cumplir con lineamientos, políticos y/o regulaciones de la Universidad de las Ciencias Informáticas.

### 2.3.9 Requerimientos de Software:

- La aplicación debe ejecutarse en plataformas libres.
- La aplicación se realizará en ambiente Web.

Para ejecutar la aplicación en el servidor:

- Debe existir un servidor de Base de Datos con soporte para PostgreSQL 8.4.
- El servidor debe tener instalado PHP 5.3.2 como módulo del servidor web Apache 2.
- Navegador Mozilla, Firefox o Internet Explorer.
- El servidor debe tener instalado el Servidor Streaming Flumotion.

Para ejecutar la aplicación en el cliente:

- Cualquier navegador web Firefox, Opera ,Internet Explorer.

### 2.3.10 Requerimientos de Hardware:

- Debe estar soportado por una red de datos a una velocidad de 100/1Gbps en la que sus equipos implementen protocolos de multidifusión y QoS, permitiendo aplicar niveles de calidad de servicio impidiendo que el flujo IP no compita con otros servicios soportando en una misma infraestructura de red.
- Un servidor o máquina con requerimientos mínimos de 1Gb de memoria Ram, 100 Gb o superior de HDD y un Micro Pentium 4 o superior, esta configuración garantizará la velocidad del procesamiento de las tareas de la aplicación y del almacenamiento de la base de datos de los usuarios y grupos del sistema.

### 2.3.11 Escalabilidad:

- El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible; para ello deben incorporarse aspectos de reutilización de componentes.
- El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

## 2.4 Descripción del Sistema Propuesto

### 2.4.1 Descripción de los actores

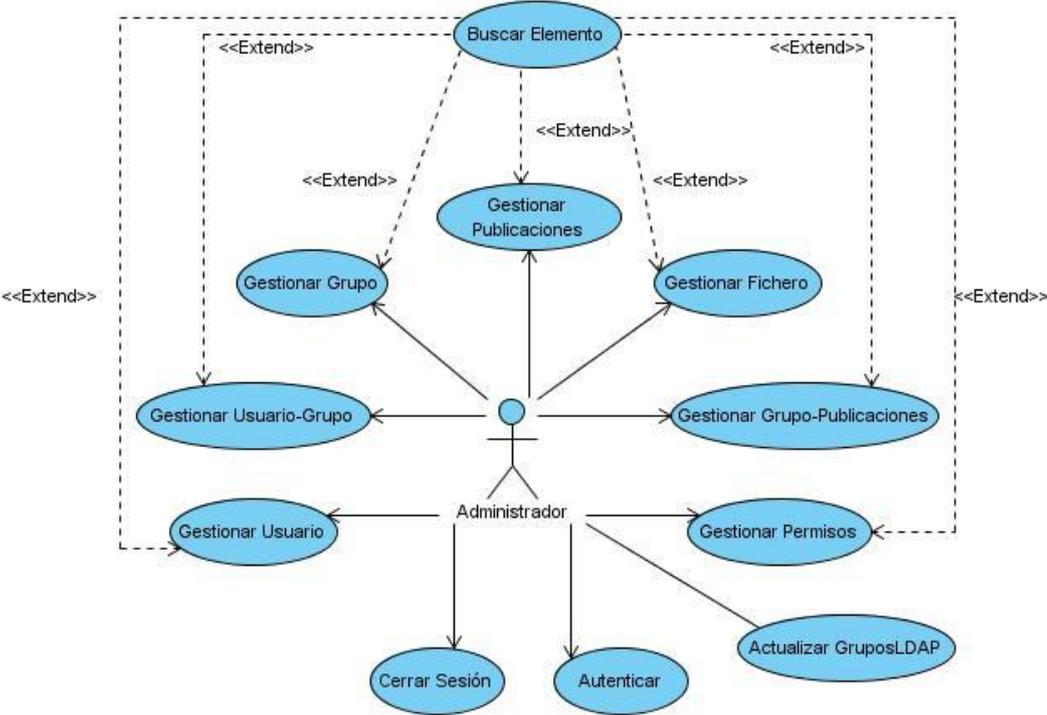
Actor	Justificación
Administrador	Es la persona que accede al sistema para hacer uso de los servicios que este le proporciona.

	Tiene acceso a todas las funcionalidades del sistema.
--	---

**2.4.2 Casos de Uso del Sistema**

Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Un caso de uso es iniciado por un actor. A partir de ese momento, ese actor, junto con otros actores, intercambian datos o control con el sistema, participando de ese caso de uso.

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario y permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. En la figura 2 se presenta el diagrama de casos de uso del sistema.



**Figura 2. Diagrama de Casos de Uso del Sistema**

**2.4.2.1 Patrones de casos de uso utilizados**

Los patrones de casos de uso son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso.

A continuación se describen los patrones de casos de uso utilizados en la modelación del sistema:

### **Concordancia (Commonality). Adición**

En este patrón, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

### **Patrón CRUD**

Este patrón está presente en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés *Create, Read, Update, Delete*. El patrón CRUD Completo consiste en un caso de uso para administrar la información (CRUD Información), nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y eliminar o dar de baja. [34]

#### **2.4.2.2 Descripción de los casos de uso del sistema**

Las descripciones de casos de uso son reseñas textuales del caso de uso, explican los procesos o actividades que tienen lugar en el caso de uso.

A continuación se muestran las descripciones textuales de los casos de uso del sistema:

#### **CUS-1: Caso de uso: Autenticar**

<b>Caso de uso:</b>	Autenticar
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el actor solicita entrar al

	sistema, ingresa sus datos, se comprueba su autenticidad y finaliza el caso de uso.
<b>Referencia:</b>	R1,R2
<b>Precondiciones:</b>	Sitio disponible
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El actor introduce los datos para su autenticación: - Usuario - Contraseña	1.1- El sistema verifica que no existan campos en blanco.
	1.1- El sistema verifica si los datos introducidos por el usuario son correctos.
	1.2- El sistema finaliza el caso de uso, mostrándole al usuario la interfaz correspondiente con sus privilegios.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1- Si existe algún campo en blanco el sistema muestra un mensaje de error.
	1.2- Si los datos de autenticación no son correctos el sistema muestra un mensaje de error y le permite al usuario volver a indicar los datos para su autenticación.
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> El usuario queda autenticado.	

**CUS-2** Caso de uso: Gestionar usuario

<b>Caso de uso:</b>	Gestionar Usuario
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide

	<p>mostrar los usuarios existentes hasta el momento y realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Crear usuario</li> <li>- Modificar Usuario</li> <li>- Eliminar Usuario</li> </ul>
<b>Referencia:</b>	R3(R3.1,R3.2,R3.3,R3.4)
<b>Precondiciones:</b>	El usuario debe estar autenticado
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1- El actor selecciona la acción que desea realizar:</p> <ul style="list-style-type: none"> <li>- Crear usuario</li> <li>- Modificar Usuario</li> <li>- Eliminar Usuario</li> </ul>	<p>Dependiendo de la acción seleccionada, el sistema hace lo siguiente:</p> <ul style="list-style-type: none"> <li>- Si el administrador selecciona la opción crear un nuevo usuario, ir a Sección “Crear usuario”.</li> <li>- Si el administrador selecciona la opción modificar usuario, ir a Sección “Modificar usuario”.</li> <li>- Si el administrador selecciona la opción eliminar usuario, ir a Sección “Eliminar usuario”.</li> </ul>
<b>Sección “Adicionar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>1.1-El sistema muestra un formulario con los campos necesarios para adicionar los datos. Los campos son los siguientes:</p> <ul style="list-style-type: none"> <li>-nombre</li> <li>-usuario</li> <li>-password</li> <li>-correo electrónico</li> </ul>
<p>2- El actor inserta los datos necesarios para adicionar el usuario.</p>	<p>2.1- El sistema verifica que los datos sean válidos.</p> <p>2.2- El sistema verifica que no hayan</p>

	campos en blanco.
	2.3- El sistema adiciona el usuario a la BD.
<b>Flujos Alternos Sección “Adicionar Usuario”</b>	
	2.1- Si los datos introducidos son incorrectos, el sistema muestra un mensaje de error.
	2.2- Si existen campos en blanco, el sistema muestra un mensaje de error.
	2.3- Si el usuario existe el sistema muestra un mensaje de error.
<b>Sección “Modificar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema busca los datos del usuario seleccionado.
	1.2 El sistema muestra el perfil relacionado con el usuario a modificar, con los datos que posee.
2- El actor modifica los datos.	2.1-El sistema valida los datos introducidos. 2.2- El sistema guarda los nuevos datos del usuario. 2.3- El sistema modifica los datos en los ficheros donde se encuentre el usuario.
<b>Flujos Alternos Sección “Modificar Usuario”</b>	
	2.1- Si los datos introducidos son incorrectos, el sistema muestra un mensaje de error.
	2.2- Si el usuario existe, el sistema muestra un mensaje de error.
	2.3- Si algún campo está en blanco, el sistema muestra un mensaje de error.

<b>Sección “Eliminar Usuarios”</b>	
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>
	1.1- El sistema elimina el usuario seleccionado de la BD y de los ficheros donde se encuentre.
<b>Flujos Alternos Sección “Eliminar Usuario”</b>	
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> Dependiendo de la acción del actor:	
<ul style="list-style-type: none"> <li>- Se crea un usuario.</li> <li>- Se modifica un usuario.</li> <li>- Se muestran todos los usuarios.</li> <li>- Se elimina un usuario.</li> </ul>	

### CUS-3 Caso de uso: Gestionar Grupo

<b>Caso de uso:</b>	Gestionar Grupo local
<b>Actores:</b>	Administrador
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el administrador decide mostrar los grupos existentes hasta el momento y realizar alguna de las siguientes operaciones</p> <ul style="list-style-type: none"> <li>- Crear grupo local</li> <li>- Eliminar grupo local</li> <li>- Modificar grupo local</li> </ul>
<b>Referencia:</b>	R4(R4.1, R4.2, R4.3, R4.4)
<b>Precondiciones:</b>	El usuario debe estar autenticado
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1- El actor selecciona la acción que desea realizar:</p> <ul style="list-style-type: none"> <li>- Crear grupo local</li> <li>- Modificar grupo local</li> <li>- Eliminar grupo local</li> </ul>	<p>Dependiendo de la acción seleccionada, el sistema hace lo siguiente:</p> <ul style="list-style-type: none"> <li>- Si el administrador selecciona la opción crear un nuevo grupo, ir a Sección “Crear grupo”.</li> <li>- Si el administrador selecciona la opción modificar grupo, ir a Sección</li> </ul>

	<p>“Modificar grupo”.</p> <ul style="list-style-type: none"> <li>- Si el administrador selecciona la opción eliminar grupo, ir a Sección “Eliminar grupo”.</li> </ul>
<b>Sección “Adicionar Grupo”</b>	
Acción del Actor:	Respuesta del Sistema:
	<p>1.1- El sistema muestra el campo necesario para adicionar los datos. El campo es el siguiente:</p> <ul style="list-style-type: none"> <li>-Nombre</li> </ul>
2- El actor inserta los datos necesarios para adicionar el grupo.	2.1- El sistema verifica que el campo no esté en blanco.
	2.2- El sistema adiciona el grupo a la BD.
<b>Flujos Alternos Sección “Adicionar Grupo”</b>	
	2.1- Si el campo está en blanco, el sistema muestra un mensaje de error.
	2.2- Si el grupo existe el sistema muestra un mensaje de error.
<b>Sección “Modificar Grupo”</b>	
Acción del Actor:	Respuesta del Sistema:
	1.1 El sistema busca los datos del grupo seleccionado.
	1.2 El sistema muestra el perfil relacionado con el grupo a modificar, con los datos que posee.
	2.1-El sistema verifica que el grupo no exista.
2- El actor modifica los datos.	2.2- El sistema guarda los nuevos datos del grupo en la BD y modifica el fichero asociado en caso que exista.
<b>Flujos Alternos Sección “Modificar Grupo”</b>	
	2.1- Si el grupo existe, el sistema

	muestra un mensaje de error.
	2.2- Si el campo está en blanco, el sistema muestra un mensaje de error.
<b>Sección “Eliminar Grupo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1- El sistema verifica si el grupo a eliminar no tiene ningún fichero asignado.
	1.2- El sistema elimina el grupo.
<b>Flujos Alternos Sección “Eliminar Grupo”</b>	
	1.1- El sistema elimina el fichero asociado al grupo.
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> Dependiendo de la acción del actor:	
	<ul style="list-style-type: none"> <li>- Se adiciona un grupo.</li> <li>- Se muestran todos los grupos.</li> <li>- Se modifica un grupo.</li> <li>- Se elimina un grupo.</li> </ul>

**CUS-5** Caso de uso: Gestionar usuario\_grupo

<b>Caso de uso:</b>	Gestionar usuario_grupo
<b>Actores:</b>	Administrador
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el administrador decide mostrar los grupos con los usuarios asignados existentes hasta el momento y realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Asignarle un usuario a un grupo</li> <li>- Eliminar un usuario de un grupo</li> <li>- Modificar un usuario de un grupo</li> </ul>
<b>Referencia:</b>	R6(R6.1, R6.2, R6.3, R6.4)
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema.
<b>Flujo Normal de Eventos</b>	

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El actor selecciona la acción que desea realizar: <ul style="list-style-type: none"> <li>- Asignar un usuario a un grupo</li> <li>- Eliminar un usuario de un grupo</li> <li>- Modificar relación de un usuario a un grupo</li> </ul>	Dependiendo de la acción seleccionada, el sistema hace lo siguiente: <ul style="list-style-type: none"> <li>- Si el administrador selecciona la opción Asignar un usuario a un grupo, ir a Sección "Adicionar usuario_grupo".</li> <li>- Si el administrador selecciona la opción modificar una relación usuario-grupo, ir a Sección "Modificar usuario_grupo".</li> <li>- Si el administrador selecciona la opción eliminar un usuario de un grupo, ir a Sección "Eliminar usuario_grupo".</li> </ul>
<b>Sección "Adicionar Usuario_Grupo"</b>	
	1.1- El sistema muestra una interfaz que permitirá seleccionar los grupos al cual el usuario será asignado.
	1.2- El sistema asigna el usuario al grupo.
<b>Flujos Alternos Sección "Adicionar Usuario_Grupo"</b>	
<b>Sección "Modificar Usuario_Grupo"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1- El sistema muestra una interfaz para seleccionar el usuario o grupo que desea modificar
2- El actor selecciona el nuevo grupo o usuario.	2.1- El sistema modifica la relación usuario-grupo en la BD y en los ficheros donde se encuentre.
<b>Flujos Alternos Sección "Modificar Usuario_Grupo"</b>	
	2.1- Si la relación ya existe el sistema muestra un mensaje de error.
<b>Sección "Eliminar Usuario_Grupo"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1- El sistema elimina la relación

	usuario-grupo seleccionada.
<b>Flujos Alternos Sección “Eliminar Usuario_Grupo”</b>	
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> Dependiendo de la acción del actor:	
<ul style="list-style-type: none"> <li>- Se asigna un usuario a un grupo.</li> <li>- Se muestran todas las relaciones usuario- grupo.</li> <li>- Se modifica una la relación usuario- grupo.</li> <li>- Se elimina una relación usuario- grupo.</li> </ul>	

**CUS-7** Caso de uso: Gestionar permisos

<b>Caso de uso:</b>	Gestionar permisos
<b>Actores:</b>	Administrador
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el administrador decide mostrar los componentes con los bouncers asignados existentes hasta el momento y realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Asignarle un bouncer a un componente</li> <li>- Eliminar un bouncer de un componente</li> </ul>
<b>Referencia:</b>	R8(R8.1, R8.2, R8.3)
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema.
<b>Flujo Normal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1- El actor selecciona la acción que desea realizar:</p> <ul style="list-style-type: none"> <li>- Asignar un bouncer a un componente</li> <li>- Eliminar un bouncer de un componente</li> </ul>	<p>Dependiendo de la acción seleccionada, el sistema hace lo siguiente:</p> <ul style="list-style-type: none"> <li>- Si el administrador selecciona la opción Asignar un bouncer a un componente, ir a Sección “Adicionar bouncer_componente”.</li> <li>- Si el administrador selecciona la opción eliminar un bouncer de un componente, ir a Sección “Eliminar bouncer_componente”.</li> </ul>
<b>Sección “Adicionar Permiso”</b>	
	1.2- El sistema muestra una interfaz

	para que el actor indique el componente al cual se le asignará uno o varios bouncer.
	1.3- El sistema asigna el bouncer al componente.
<b>Flujos Alternos Sección “Adicionar Permiso”</b>	
<b>Sección “Eliminar Permiso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2- El actor selecciona el bouncer y el componente del que desea eliminarlo.	2.1- El sistema elimina la relación bouncer-componente.
<b>Flujos Alternos Sección “Eliminar Permiso”</b>	
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> Dependiendo de la acción del actor:	
<ul style="list-style-type: none"> <li>- Se asigna un bouncer a un componente.</li> <li>- Se muestran todas los bouncers y componentes.</li> <li>- Se elimina una relación bouncer- componente.</li> </ul>	

**CUS-9** Caso de uso: Actualizar Grupos del LDAP

<b>Caso de uso:</b>	Actualizar Grupos LDAP
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide actualizar los grupos desde el LDAP.
<b>Referencia:</b>	R10
<b>Precondiciones:</b>	El usuario debe estar autenticado
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El actor selecciona la opción Gestionar usuario.	1.1- El sistema muestra la interfaz que contiene la información de los grupos.
2- El actor selecciona la opción Actualizar.	2.1- El sistema muestra un formulario para introducir los datos para autenticarse como usuario dominio.
3-El actor introduce su usuario y contraseña del dominio.	3.1- El sistema verifica si los datos introducidos por el usuario son correctos.
	3.2- Actualiza la BD con los grupos del

	LDAP.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1- Si los datos de autenticación no son correctos el sistema muestra y un mensaje de error.
<b>Prioridad: Crítico</b>	
<b>Poscondiciones:</b> Queda actualizada la información de los grupos que se encuentran en el LDAP.	

#### **CUS-10** Caso de uso: Buscar Elemento

<b>Caso de uso:</b>	Buscar elemento
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide mostrar buscar algún elemento de los que se encuentran guardados en la BD.
<b>Referencia:</b>	R11
<b>Precondiciones:</b>	El usuario debe estar autenticado
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El actor selecciona la opción Gestionar elemento.	1.1- El sistema muestra la interfaz que contiene la información de los elementos.
2- El actor selecciona la opción Buscar.	2.1- El sistema muestra un formulario para introducir un parámetro de búsqueda.
3- El actor inserta los datos necesarios para buscar el elemento.	3.1- El sistema muestra el elemento que se corresponde con el parámetro introducido por el actor.
<b>Flujos Alternos</b>	
	3.1- Si el elemento no existe el sistema muestra un mensaje.

#### **CUS-11** Caso de uso: Cerrar Sesión

<b>Caso de uso:</b>	Cerrar Sesión	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador desea finalizar sesión.	
<b>Referencia:</b>	R12	
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El actor selecciona la opción finalizar sesión desde cualquier interfaz del sistema.	1.1- El sistema cierra la sesión.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<b>Prioridad:</b> Auxiliar		
<b>Postcondiciones:</b> El usuario finaliza su sesión		

### ***2.5 Conclusiones parciales del capítulo***

En el contenido del presente capítulo se identificaron los requisitos funcionales de la aplicación especificando así las principales funcionalidades que debe poseer el sistema, además se definieron los requisitos no funcionales, aspecto esencial para que el software tenga la calidad requerida en términos de usabilidad, apariencia e interoperabilidad y de esta forma se convierta en un producto atractivo para los clientes. Se definieron, describieron y modelaron los casos de uso permitiendo detallar de forma concreta el funcionamiento del sistema logrando así un mejor entendimiento del mismo.

### CAPÍTULO 3: Diseño de la solución propuesta.

En este capítulo se modelan los diagramas de clase y de secuencias que satisfacen a cada uno de los requisitos definidos en el Capítulo II. Se muestran además el modelo de datos el cual brinda una representación gráfica del diseño de la base de datos del sistema y el diagrama de clases persistentes correspondiente al diagrama entidad relación. Se abordará acerca del estilo arquitectónico a utilizar, exponiendo sus principales características, ventajas y desventajas.

#### 3.1 Diagramas de Clases

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. A continuación se muestran los diagramas de clases de diseño de cada uno de los casos de uso del sistema de control de acceso para la plataforma IPTV-UCI.

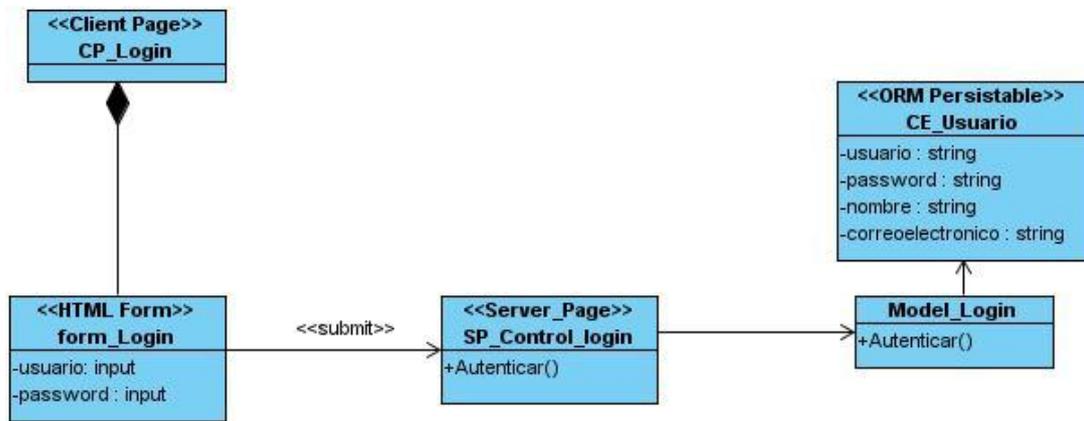


Figura 3. Diagrama de clases caso de uso Autenticar

### Capítulo 3: Diseño de la solución propuesta

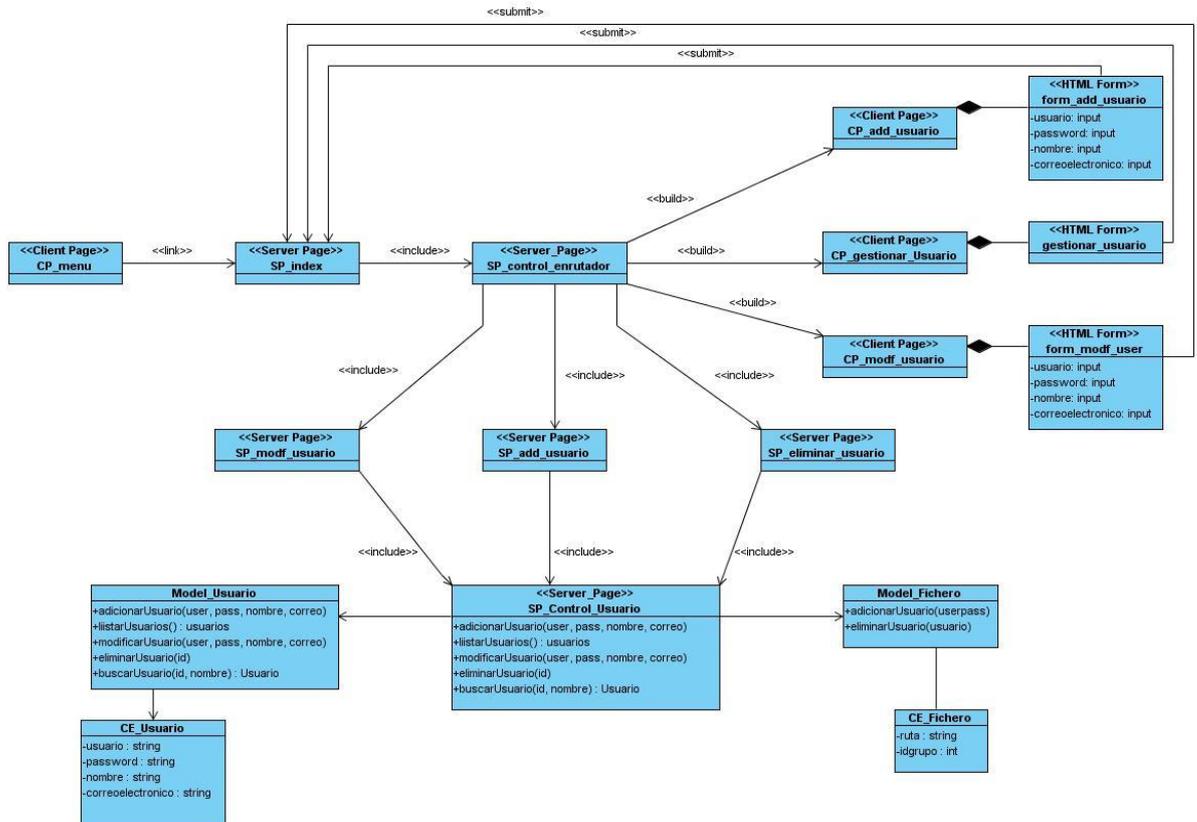


Figura 4. Diagrama de clases caso de uso Gestionar Usuario

## Capítulo 3: Diseño de la solución propuesta

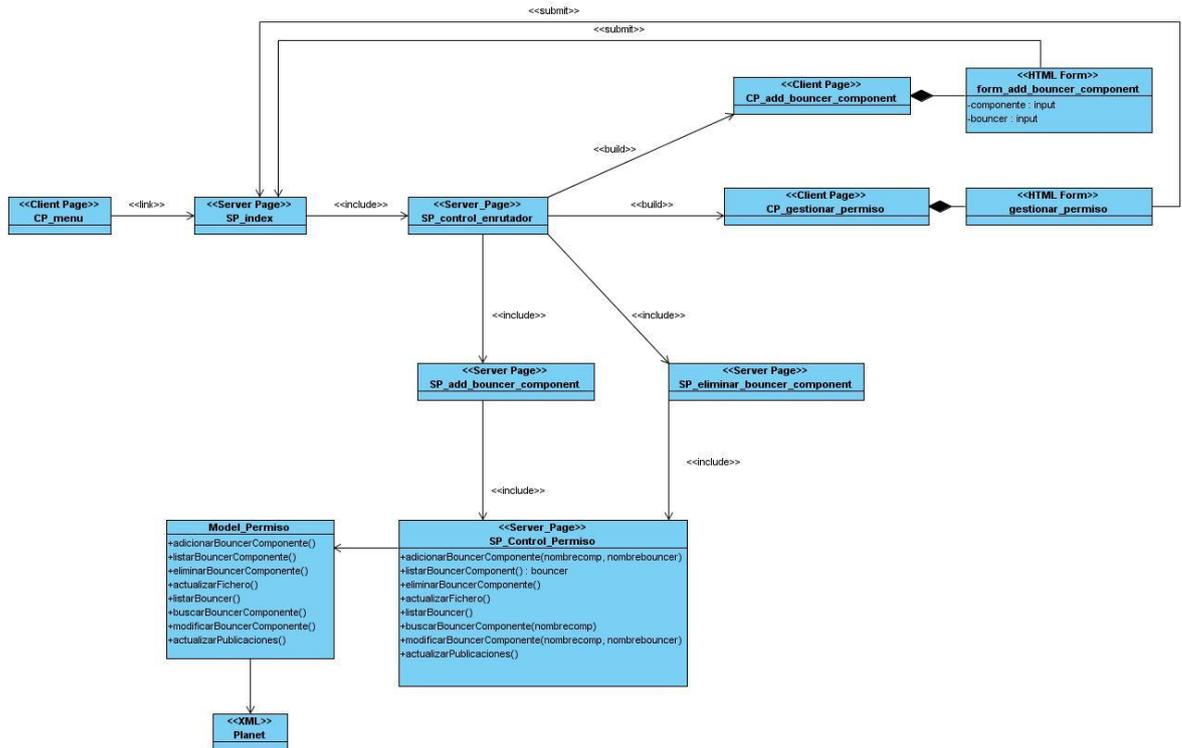


Figura 5. Diagrama de clases caso de uso Gestionar permisos

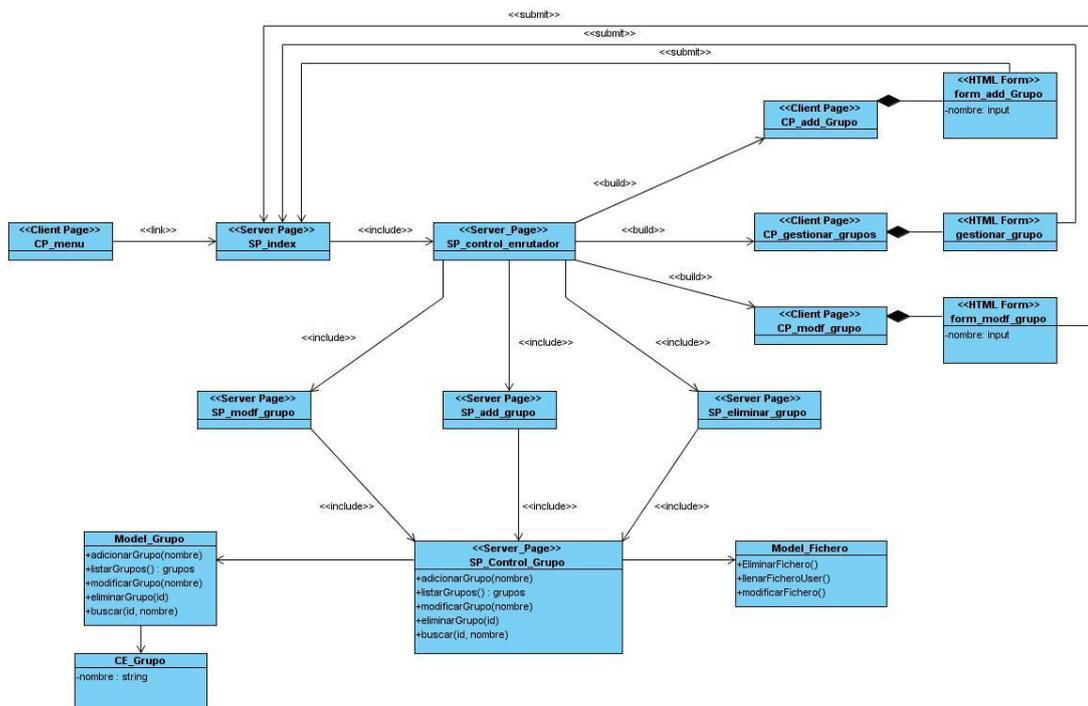


Figura 6. Diagrama de clases caso de uso Gestionar Grupo

## Capítulo 3: Diseño de la solución propuesta

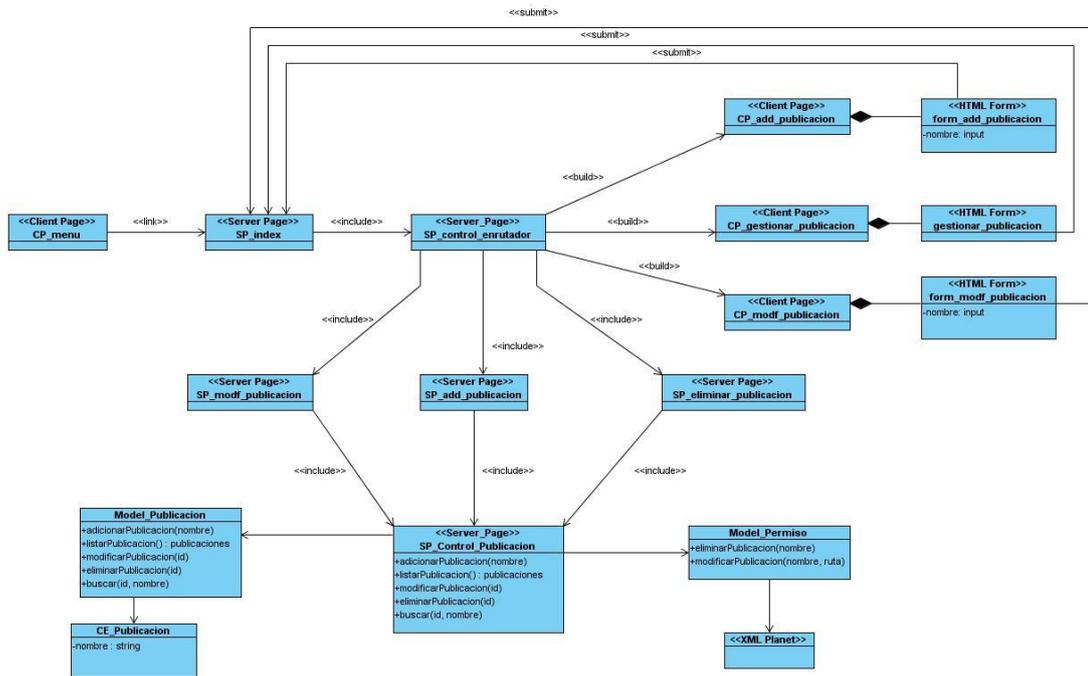


Figura 7. Diagrama de clases caso de uso Gestionar Publicaciones

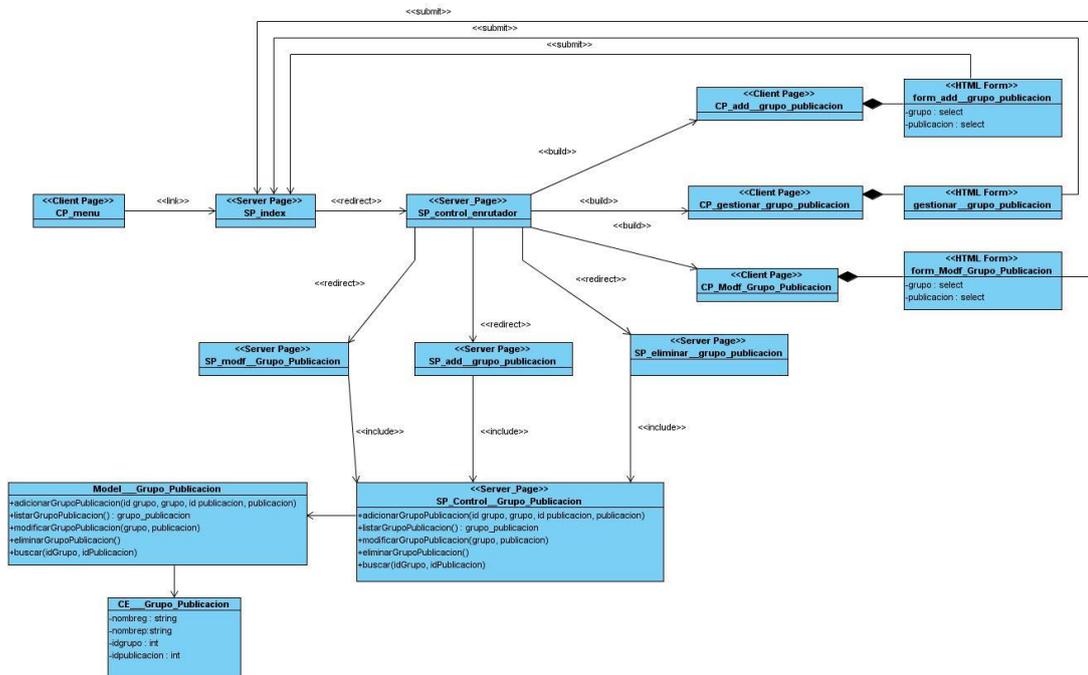


Figura 8. Diagrama de clases caso de uso Grupo- Publicación

### Capítulo 3: Diseño de la solución propuesta

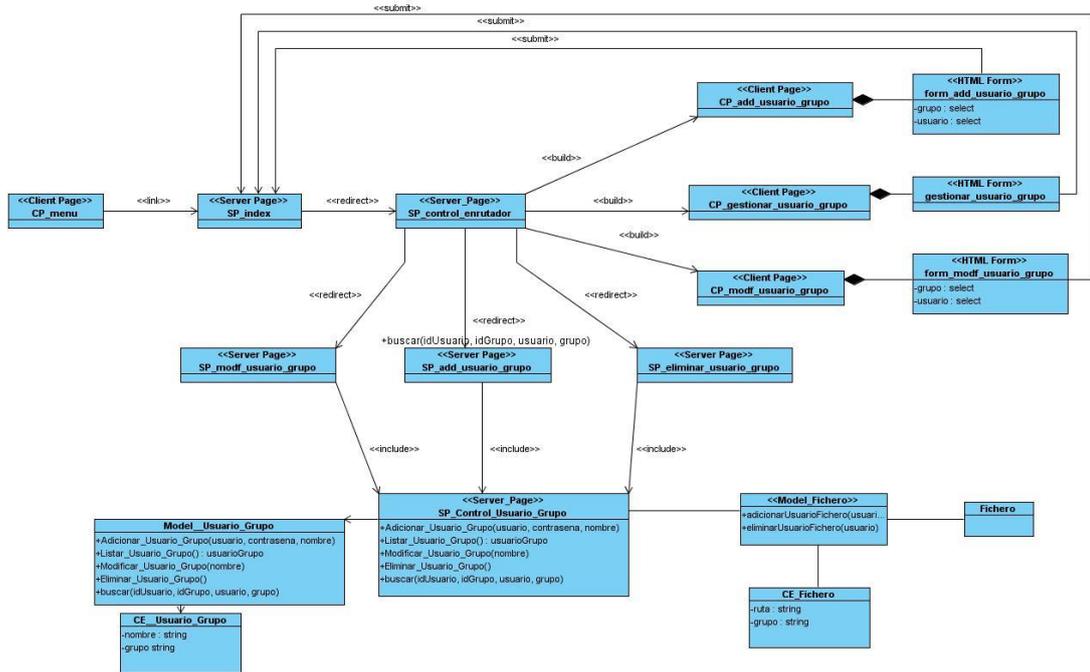


Figura 9. Diagrama de clases caso de uso Gestionar Usuario- Grupo

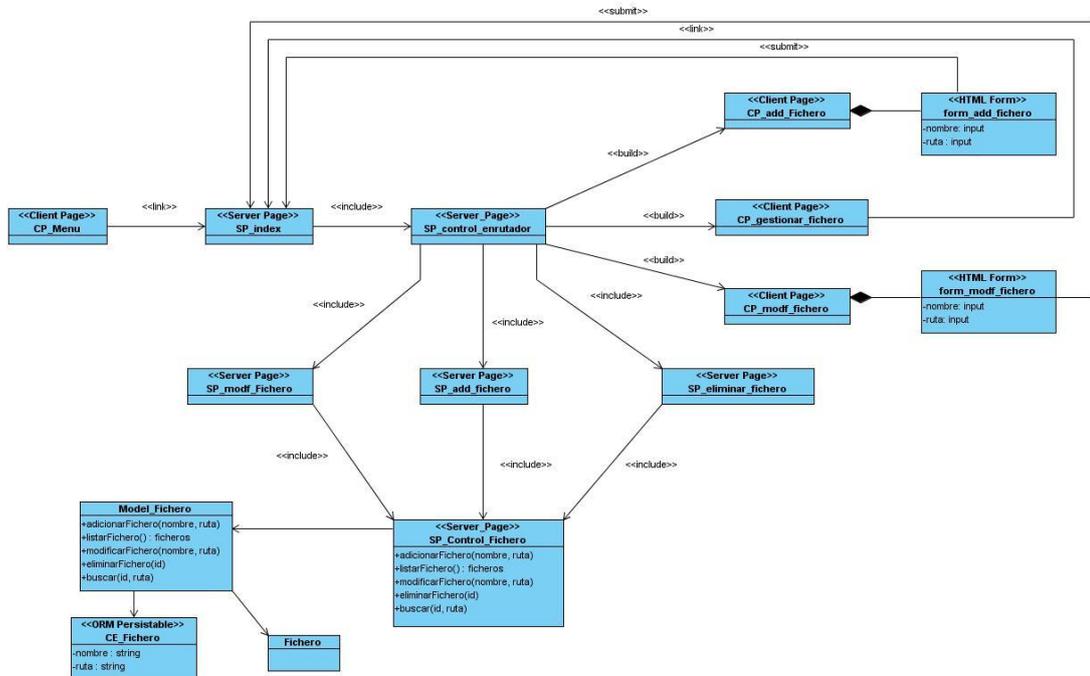


Figura 10. Diagrama de clases caso de uso Gestionar Fichero

### Capítulo 3: Diseño de la solución propuesta

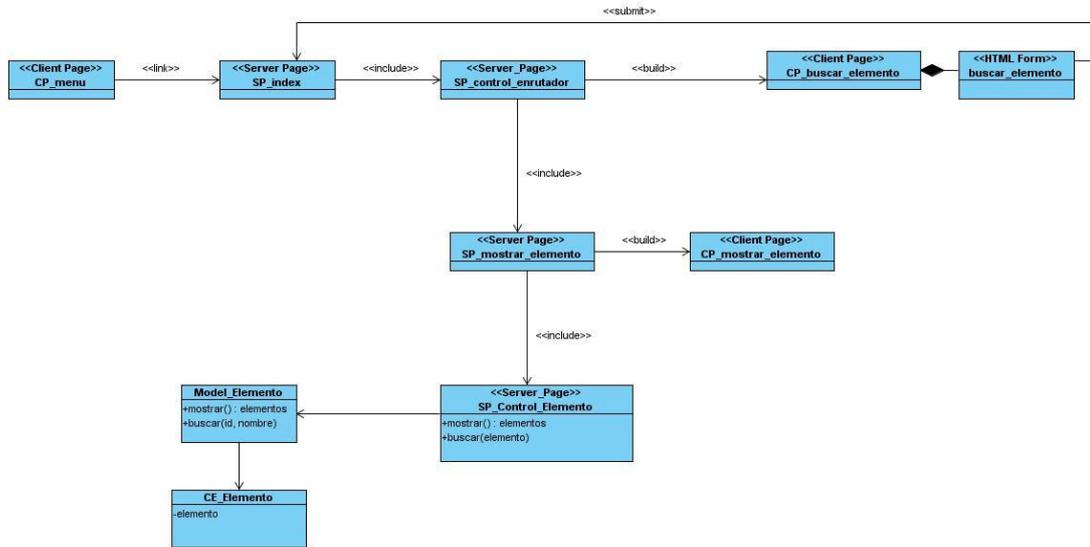


Figura 11. Diagrama de clases caso de uso Buscar Elemento

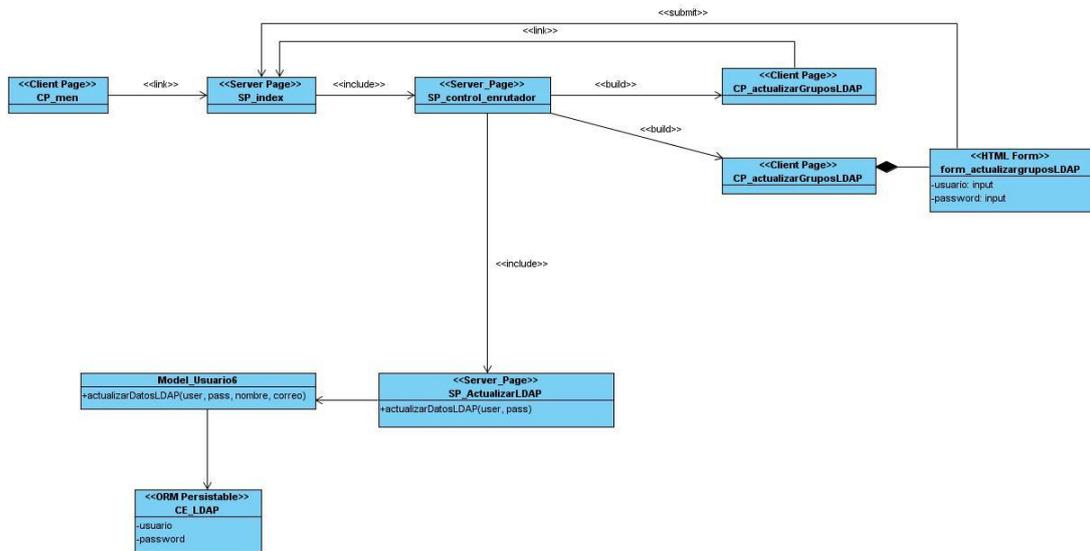


Figura 12. Diagrama de clases caso de uso Actualizar Grupos de LDAP

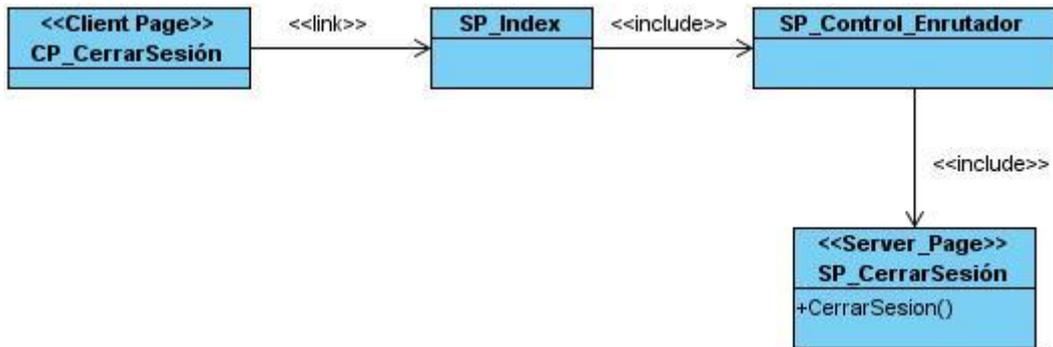


Figura 13. Diagrama de clases caso de uso Cerrar Sesión

### 3.2 Diagramas de Secuencias

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir cómo se comunican los objetos entre sí. El diagrama de secuencias proporciona un camino a partir de los escenarios para describir las operaciones en una forma más detallada.

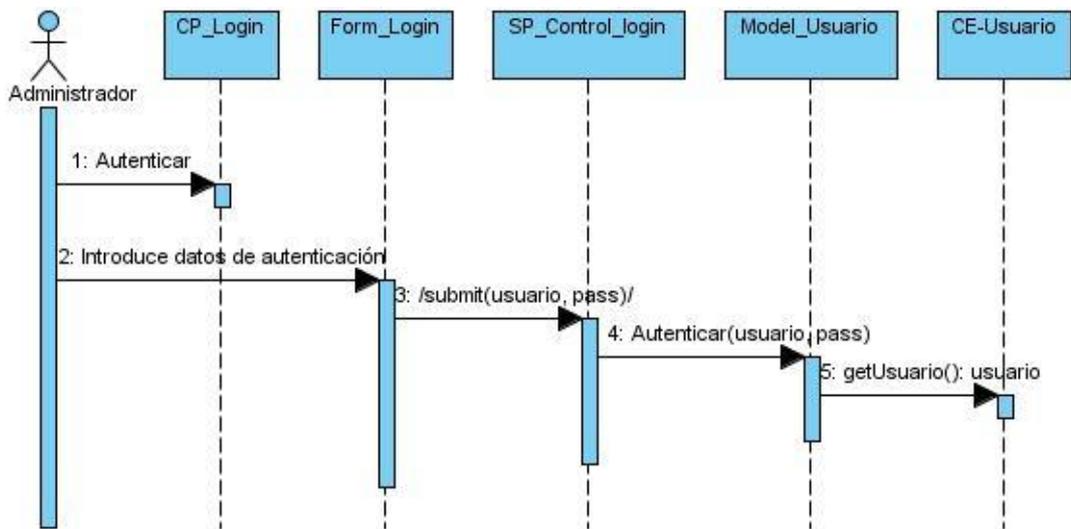


Figura 14. Diagrama de secuencias Caso de Uso Autenticar Usuario.

## Capítulo 3: Diseño de la solución propuesta

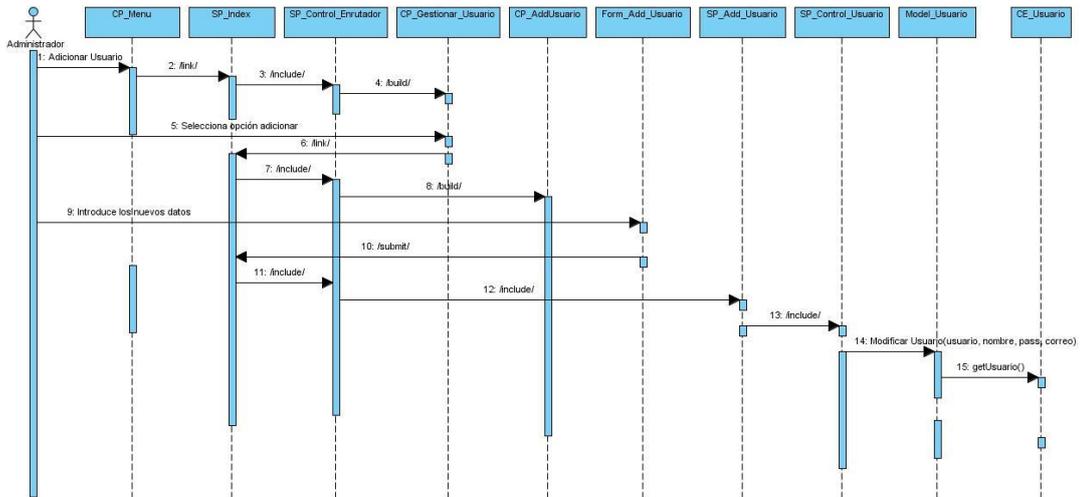


Figura 15. Diagrama de secuencias Caso de Uso Gestionar Usuario. Sección Adicionar Usuario.

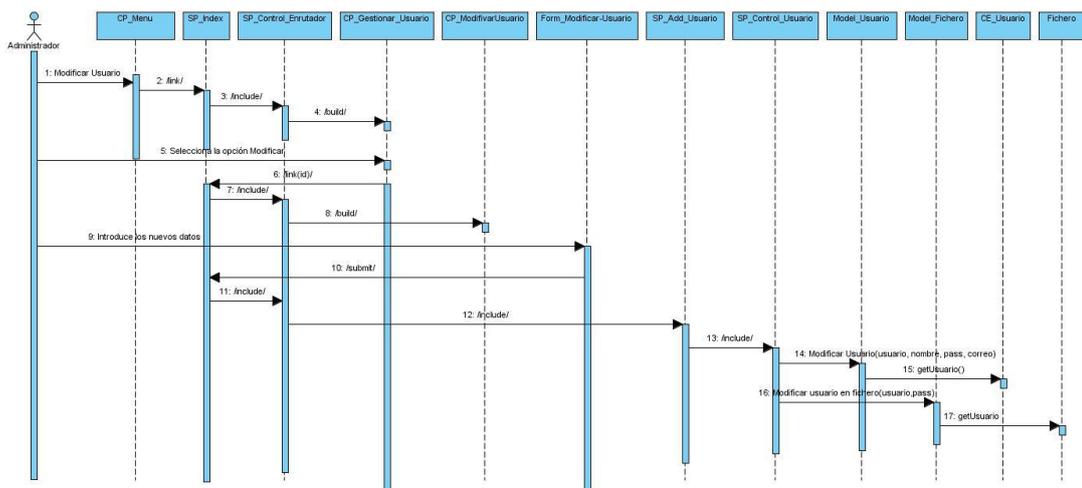


Figura 16. Diagrama de secuencias Caso de Uso Gestionar Usuario. Sección Modificar Usuario.

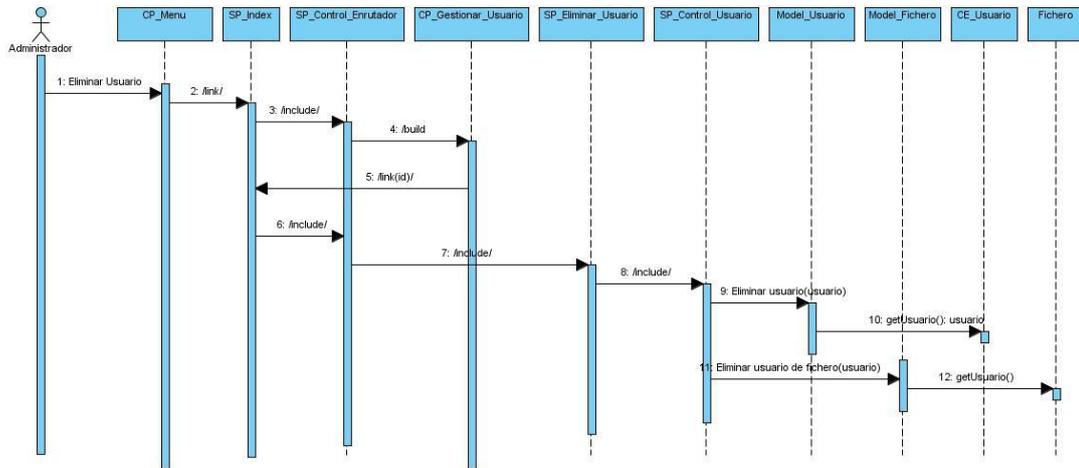


Figura 17. Diagrama de secuencias Caso de Uso Gestionar Usuario. Sección Eliminar Usuario.

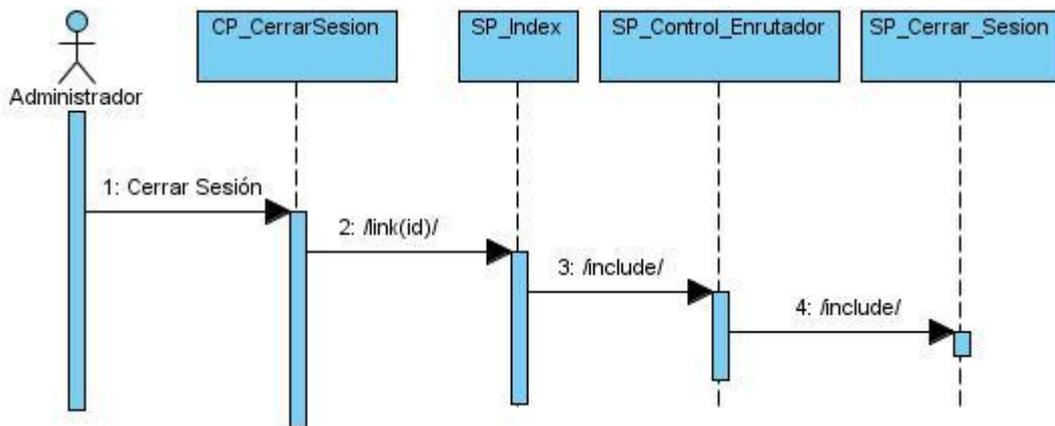


Figura 18. Diagrama de secuencias Caso de Uso Cerrar Sesión.

### 3.3 Principios de diseño

#### 3.3.1 Estándares de la interfaz de la aplicación

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta cómodo, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades suaves y refrescantes. El

vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios. Los mensajes de error son pequeños y en español. El fondo de las páginas es de color blanco para mayor frescura de la vista. Todo esto se ha hecho con el objetivo de que el uso del sitio brinde comodidad y confort al usuario.

### 3.4 Diseño de la Base de Datos

#### 3.4.1 Diagrama Entidad Relación

El modelo entidad-relación o diagrama E/R está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas a través de un esquema gráfico empleando los terminología de entidades; una entidad es cualquier objeto distinguible relevante en el mundo en cuestión. Estas entidades poseen un número indeterminado de propiedades, que son "trozos" de información que describen a las entidades de uno u otro modo. Cada una de las entidades tiene una identidad, o sea, son identificables de forma única. [35]

A continuación se presenta el diagrama entidad relación del sistema para el control de acceso para la plataforma IPTV-UCI.

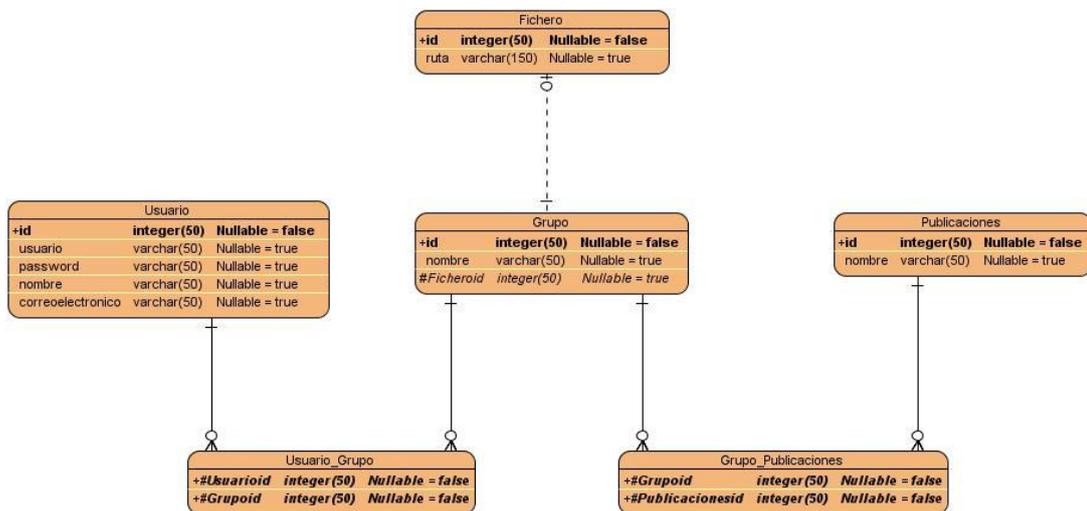


Figura 19. Diagrama Entidad Relación

Este modelo especifica la existencia de cuatro entidades: Usuario, Grupo, Publicaciones y Fichero. Un usuario tiene uno o muchos grupos y un grupo puede tener ningún o muchos usuarios. Un grupo está relacionado con una o muchas

publicaciones y una publicación se relaciona con uno o muchos grupos. Para cada una de las entidades existe una propiedad que las identifica de forma única y que se corresponde con la clave primaria (en este caso es el identificador) de cada una de las tablas en la implementación relacional. Las entidades tienen otras propiedades que las describen y que se corresponden con los distintos campos de la tabla. Producto de la relación usuario-grupo y grupo-publicaciones se genera una nueva tabla por cada relación con los identificadores respectivos de las tablas que contienen esta relación. Un fichero está relacionado con sólo un grupo y este puede tener asociado ningún o un grupo.

### 3.4.2 Diagrama de Clases Persistentes

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo. A continuación se muestra este diagrama.

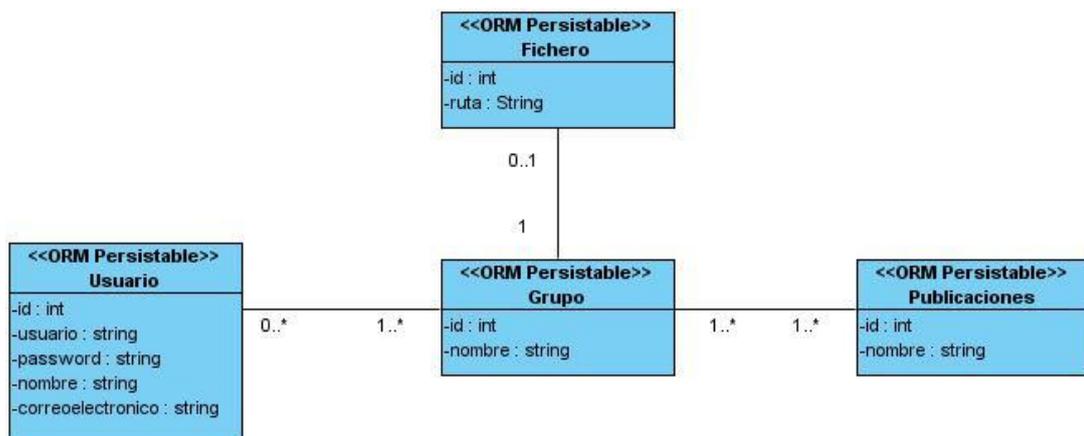


Figura 20. Diagrama de clases persistentes

### 3.5 Estilo Arquitectónico

El patrón de diseño **Modelo Vista Controlador** o **MVC** (por sus siglas en inglés) describe una forma, muy utilizada en el Web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera,

facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en 3 estratos:

Modelo, Vista y Controlador, los cuales serán explicados en breve:

**El Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

**La Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

**El Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

### 3.5.1 Ciclo de vida de MVC

El ciclo de vida de MVC es normalmente representado por las 3 capas presentadas anteriormente y el usuario. El siguiente diagrama representa el ciclo de vida de manera sencilla:

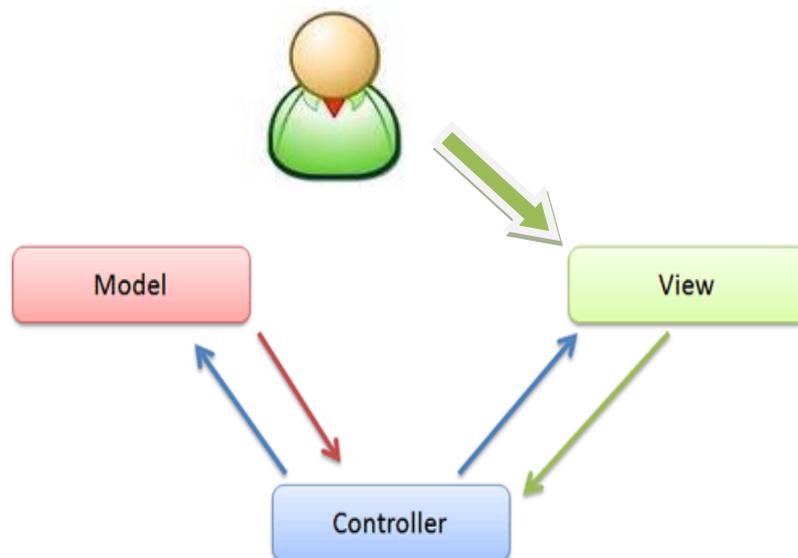


Figura 21. Modelo Vista Controlador

La figura muestra un sencillo ejemplo con MVC, en la que el flujo de control comienza cuando el usuario hace una solicitud al controlador con información sobre lo que el usuario desea realizar. Entonces el Controlador decide a quién debe delegar la tarea y es aquí donde el Modelo empieza su trabajo. En esta etapa, el Modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el Controlador. Una vez que termina su labor, le regresa al Controlador la información resultante de sus operaciones, el cual a su vez redirige a la Vista. La Vista se encarga de transformar los datos en información visualmente entendible para el usuario. Finalmente, la representación gráfica es transmitida de regreso al Controlador y éste se encarga de transmitírsela al usuario. El ciclo entero puede empezar nuevamente si el usuario así lo requiere.

#### 3.5.2 Ventajas y Desventajas de MVC

Este modelo de arquitectura presenta varias ventajas:

- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Reutilización de los componentes.

Las desventajas de seguir el planteamiento de MVC son:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos. [36]

**3.6 Conclusiones parciales del capítulo**

En el presente capítulo se presentaron los diagramas de clases del diseño y diagramas de secuencias, brindando una descripción precisa del flujo de eventos que sigue el sistema al ejecutar cada una de las funcionalidades. Además se diseñó la Base de datos obteniendo el diagrama de clases persistentes a partir del mismo y se definió el Modelo Vista Controlador como estilo arquitectónico a utilizar por poseer elementos que lo hacen ser el ideal para el desarrollo del sistema.

## CAPÍTULO 4: Implementación y pruebas.

En este capítulo se presentan el diagrama de despliegue y el diagrama de componentes correspondiente al sistema de control de acceso para la plataforma IPTV- UCI, mostrándose los protocolos de interconexión utilizados por los elementos físicos que intervienen en el funcionamiento del sistema. Se muestran además las pruebas realizadas al sistema para comprobar que se cumple cabalmente con cada uno de los requisitos funcionales garantizando la calidad del software.

### 4.1 Modelo de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes de hardware y software de un sistema informático.

El diagrama de despliegue para el Sistema de control de acceso para la plataforma IPTV-UCI representa 3 nodos principales. El nodo procesador PC Cliente, el nodo procesador Servidor de Aplicaciones y el nodo procesador Servidor de Base de datos.

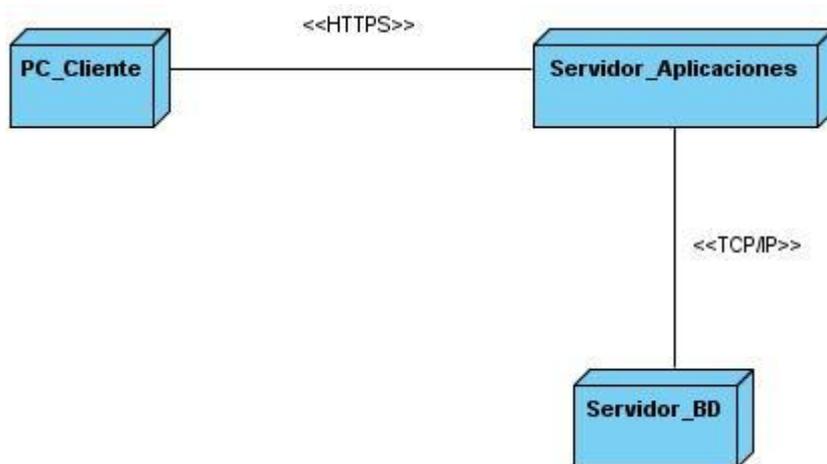


Figura 22. Diagrama de despliegue.

**Pc Cliente:** Representa la máquina que empleará el usuario que interactuará con el Sistema.

**Servidor de Aplicaciones:** No es más que un servidor en una red de computadores que ejecuta ciertas aplicaciones, es un software que ayuda al servidor Web a procesar las páginas que contienen scripts o etiquetas del lado del servidor. Cuando se solicita al servidor una página de este tipo, el servidor Web

pasa la página al servidor de aplicaciones para su procesamiento antes de enviarla al navegador.

**Servidor de Base de Datos:** Se utiliza para bases de datos con múltiples usuarios. Las bases de datos están situadas en un servidor y se puede acceder a ellas desde terminales o equipos con un programa (llamado cliente) que permita el acceso a la base o bases de datos. Estos gestores de base de datos permiten que varios usuarios hagan operaciones sobre ella al mismo tiempo: uno puede hacer una consulta al mismo tiempo que otro, situado en lugares diferentes introduciendo datos en la base.

El nodo PC Cliente estará conectado mediante el Protocolo de Transferencia de Hipertexto (HTTP) al nodo procesador que representa al Servidor de Aplicaciones. La conexión entre el Servidor de Aplicaciones y el Servidor de Base de datos se realizará mediante el protocolo de comunicación TCP/IP.

### ***4.2 Modelo de Implementación***

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Los componentes son grupos de piezas de software clases que representan todo el sistema, sus relaciones y son responsables por las operaciones dentro del sistema. El diagrama de componentes muestra el empaquetamiento físico del código. Esto puede ser una clase, un archivo comprimido, un paquete, un ejecutable, librerías compartidas, bases de datos, entre otras. Cuando un componente colabora con otro, esta colaboración es representada como una dependencia entre el componente cliente y el componente servicio.

En la siguiente imagen se representa el diagrama de componentes referente al sistema para el control de acceso para la plataforma IPTV-UCI.

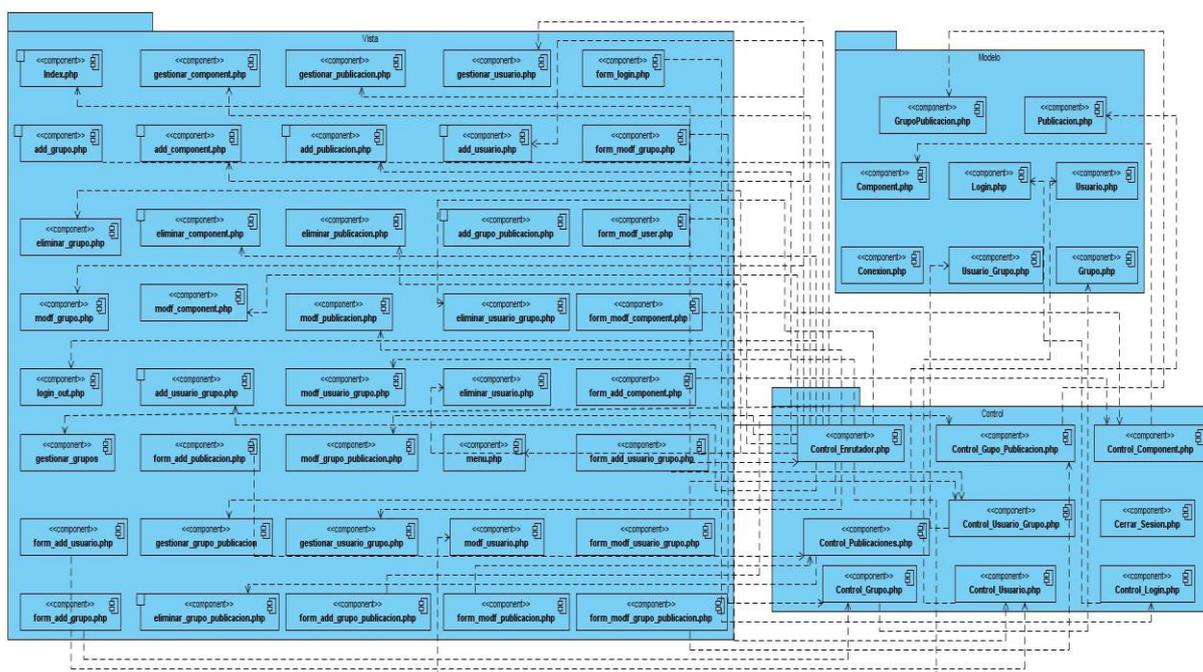


Figura 23. Diagrama de componentes.

### 4.3 Pruebas del sistema propuesto

“Las pruebas de software constituyen un pilar indispensable para evaluar y determinar la calidad de un software. Concretamente se puede definir pruebas de software como:

- El proceso de ejecución de un programa con la intención de descubrir errores previos a la entrega al usuario final.
- Una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones específicas, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente”. [37]

El proceso de pruebas permite determinar la calidad de un producto de software, su objetivo es medir el grado en que el software cumple con los requerimientos definidos. Las pruebas se pueden realizar basándose en dos esquemas diferentes: demostrar a través de pruebas de caja blanca que las operaciones internas se ajustan a lo especificado y que los componentes internos andan bien, o mediante las pruebas de caja negra, conociendo la función del programa, intentar demostrar que las funciones del programa están correctas. Para la realización de las pruebas al Sistema de control de acceso para la plataforma IPTV-UCI se utilizó la prueba de caja negra, mediante la cual se comprueba que el sistema desarrollado cumple con las

## Capítulo 4: Implementación y pruebas

funciones específicas para las cuales ha sido creado, no enfoca su atención a cómo se generan las respuestas del sistema, se basa en el análisis de los datos de entrada y en los de salida sin importar su funcionamiento interno, esto se detalla en los casos de prueba que se muestran a continuación:

Caso de Prueba del caso de uso Autenticar:

### Descripción General.

El caso de uso se inicia cuando el administrador solicita entrar al sistema ingresando su usuario y contraseña en el formulario correspondiente, al oprimir el botón “Entrar” disfruta de las funcionalidades que el sistema le brinda según sus privilegios.

### Condiciones de Ejecución.

Para iniciar este caso de uso sólo es necesario que el sitio esté disponible.

Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Autenticar Usuario	EC 1.1: Autenticarse.	<p><i>El administrador introduce el usuario.</i></p> <p><i>El administrador introduce la contraseña.</i></p> <p><i>El administrador inicia la sesión.</i></p>
	EC 1.2: Usuario y/o contraseña incorrecta.	<p><i>El administrador introduce el usuario.</i></p> <p><i>El administrador introduce la contraseña.</i></p> <p><i>El administrador inicia la sesión.</i></p> <p><i>No se inicia la sesión.</i></p>

Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Usuario	Campo de texto	No	En este campo se debe introducir el usuario administrador establecido.

[2]	Password	<i>Campo de texto</i>	No	<i>En este campo se debe introducir la contraseña correspondiente al usuario administrador que se introdujo en el campo "usuario".</i>
-----	----------	-----------------------	----	--

### Matriz de Datos

#### SC 1 Autenticar Usuario

Escenario	Usuario	Password	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
<i>EC1.1: Autenticarse.</i>	guille	*****	Iniciar la sesión	Satisfactorio.	<i>Llenar formulario Iniciar sesión</i>
<i>EC 1.2: Usuario y/o contraseña incorrecta.</i>	ely	*****	No se inicia la sesión mostrando	<i>Insatisfactorio.</i>	<i>Llenar el formulario. Iniciar sesión.</i>

*Para consultar los casos de prueba de los restantes casos de uso consultar los anexos.*

Luego de establecer el método de prueba a utilizar, fue posible evaluar las funcionalidades del sistema y detectar los errores de las mismas. Durante esta etapa cada funcionalidad generada durante la implementación fue sometida a diferentes pruebas para validar el funcionamiento de la aplicación. Se detectaron ocho no conformidades, las cuales fueron solucionadas por el equipo de desarrollo en su totalidad.

#### **4.4 Conclusiones parciales del capítulo**

En el presente capítulo se mostró cómo están relacionados los componentes del sistema físicamente utilizando como representación gráfica el modelo de despliegue y el modelo de implementación que ejemplifica cómo está estructurado el sistema en términos de componentes. Además se exponen los resultados obtenidos de las pruebas que se le realizan al sistema obteniendo resultados satisfactorios ya que el 100% de las no conformidades detectadas fueron resueltas.

### Conclusiones Generales

En vista a darle solución al problema científico planteado en la investigación se considera que se le ha dado cumplimiento a todos los objetivos propuestos quedando como resultado el sistema de control de acceso para la plataforma IPTV-UCI. Para esto se llevaron a cabo una serie de tareas que tributaron a alcanzar el resultado esperado logrando arribar a las siguientes conclusiones:

- Se elaboró el basamento conceptual del tema, logrando una amplia documentación relacionada con los términos necesarios para una mejor comprensión del contenido.
- Se estudió el estado de la IPTV en el mundo y en la UCI.
- Se analizaron las características de los sistemas de control de acceso existentes para las plataformas IPTV.
- Se realizó el análisis y diseño de la aplicación permitiendo establecer una arquitectura que da soporte a la implementación y que responde a los requisitos funcionales y no funcionales establecidos.
- Se analizó el estilo arquitectónico utilizado en la solución planteada.
- El sistema de control de acceso para la plataforma IPTV-UCI cumple con el objetivo fundamental del presente trabajo, ya que constituye un mecanismo para controlar el acceso a las multimedias publicadas en la plataforma IPTV de la UCI.

**Recomendaciones**

- Utilizar el sistema de control de acceso como parte de la solución de seguridad para la plataforma IPTV que se desarrolla en la universidad.
- Continuar el desarrollo de la herramienta con el objetivo de incorporar otras funcionalidades que incorporen otros criterios de seguridad.

**Referencias Bibliográficas**

1. Entrevistas realizadas a los compañeros de la Dirección de Televisión universitaria(DTU) y de la Dirección de Gestión Tecnológica(DGT) de la UCI.
2. Televisión IP, casi una realidad - Tecnología - CNNExpansion.com. [En línea] [Citado el: 20 de Noviembre de 2010.] <<http://www.cnnexpansion.com/tecnologia/2010/09/10/television-ip-en-mexico-una-realidad>>.
3. Definiciones rapidas:¿Que es Television IP? - Servicios de Streaming - Webtv - IPTV - Television Online - Multistream.tv [En línea] [Citado el: 22 de Noviembre de 2010.] <<http://www.multistream.tv/definiciones-television-ip.aspx>>.
4. Qué es Streaming. [En línea] [Citado el: 25 de Noviembre de 2010.] <<http://www.desarrolloweb.com/articulos/482.php>>.
5. IR: Protocolo de Streaming. [En línea] [Citado el: 25 de Noviembre de 2010.]<<http://docencia.etsit.urjc.es/moodle/mod/forum/discuss.php?d=11608>>.
6. RTSP. [En línea] [Citado el: 27 de Noviembre de 2010.] <<http://ldc.usb.ve/~poc/RedesII/Grupos/G10/transmision.htm#streaming>>.
7. P3.pdf (application/pdf Object). [En línea] [Citado el: 28 de Noviembre de 2010.] <<http://www.grc.upv.es/docencia/tdm/practicas/P3.pdf>>.
8. Control de acceso - EcuRed. [En línea] [Citado el: 23 de Noviembre de 2010.] <[http://www.ecured.cu/index.php/Control\\_de\\_acceso](http://www.ecured.cu/index.php/Control_de_acceso)>.
9. Asignación de derechos de usuario [En línea] [Citado el: 24 de Noviembre de 2010.] <<http://www.microsoft.com/spain/technet/recursos/articulos/secmod51.ms>>
10. Definición de PROTOCOLO. Diseño páginas web y diseño web a su medida. Streaming de video. [En línea] [Citado el: 4 de enero de 2011.] <<http://www.hooping.net/glossary/protocolo-96.aspx>>.
11. ESIDEA GRUPO TECNOLÓGICO S.L. - ¿Que es el Streaming? [En línea] [Citado el: 16 de Marzo de 2011.] <[http://www.esidea.com/index.asp?pagina\\_e=noticias&mas=155](http://www.esidea.com/index.asp?pagina_e=noticias&mas=155)>.
12. teleco.com.br. [En línea] [Citado el: 2 de Diciembre de 2010.]

- <[http://www.teleco.com.br/es/es\\_IPTV\\_mundo.asp](http://www.teleco.com.br/es/es_IPTV_mundo.asp)>.
13. Qué es el control de acceso. [En línea] [Citado el: 3 de Diciembre de 2010.] <<http://www.iec.csic.es/criptonomicon/autenticacion/control.html>>.
  14. End-To-End Streaming Media Delivery - Helix, a product from RealNetworks [En línea] [Citado el: 5 de Diciembre de 2010.] <<http://www.realnetworks.com/helix/index.aspx>>.
  15. Microsoft Windows Media - Architecture of Windows Media Rights Manager [En línea] [Citado el: 6 de Diciembre de 2010.] <<http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx>>.
  16. Mauritius- CoreMedia DRM [En línea] [Citado el: 14 de Diciembre de 2010.] <<http://mauritius.coremedia.com/coremedia/drm/-/610/5948/-/index.html>>.
  17. Grady, Booch. *The Unified Modeling Language*.
  18. Rational Unified Process es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable [En línea] [Citado el: 10 de Enero de 2011.] <<http://www.rational.com.ar/herramientas/rup.html>>.
  19. LinuxCentro.net - Características de PHP. [En línea] [Citado el: 10 de Enero de 2011.] <<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>>.
  20. Rational Software Modeler, una herramienta de diseño y modelado visual basada en UML para documentación y comunicación con diferentes vistas de un sistema. [En línea] [Citado el: 12 de Enero de 2011.] <<http://www.rational.com.ar/herramientas/softwaremodeler.html>>.
  21. is1-p01-trans.pdf (application/pdf Objeto). [En línea] [Citado el: 12 de Enero de 2011.] <<http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/practicas-1/is1-p01-trans.pdf>>.
  22. Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] [Citado el: 17 de Enero de 2011.] <<http://www.usabilidadweb.com.ar/postgre.php>>.
  23. NetBeans IDE 6.9 | CódigoAdicto.Com. [En línea] [citado el 15 Junio 2011]. [Citado el: 18 de Enero de 2011.] <<http://www.codigoadicto.com/software-netbeans-ide-6-9/>>.
  24. Subversion – EcuRed [En línea] [Citado el: 1 de Diciembre de 2010.]

- <<http://www.ecured.cu/index.php/Subversion>>.
25. PostgreSQL - Guía Ubuntu. [En línea] [Citado el: 6 de Enero de 2011.]  
<<http://www.guia-ubuntu.org/index.php?title=PostgreSQL>>.
26. Prestaciones | Dataprix. [En línea] [Citado el: 6 de Enero de 2011.]  
<<http://www.dataprix.com/12-prestaciones>>.
27. PostGreSQL. [En línea] [Citado el: 6 de Enero de 2011.]  
<[http://danielpecos.com/docs/mysql\\_postgres/x15.html#AEN30](http://danielpecos.com/docs/mysql_postgres/x15.html#AEN30)>.
28. HTTPD - Servidor web Apache2. [En línea] [Citado el: 20 de Enero de 2011.] <<https://help.ubuntu.com/6.10/ubuntu/serverguide/es/httpd.html>>.
29. Ventajas y Desventajas de Apache Web Server « cerocerouno [En línea] [Citado el: 20 de Enero de 2011.]  
<<http://www.cerocerouno.com.ar/?p=228>>.
30. Glosario [En línea] [Citado el: 8 de Febrero de 2011.]  
<<http://olea.org/traduccion/guia-usuario-evolution-1.2/apx-gloss.html>>.
31. Glosario. [En línea] [Citado el: 10 de Enero de 2011.]  
<<http://artemisa.unicauca.edu.co/~dparedes/java/security1dot2/summary/glossary.html>>.
32. Flumotion. [En línea] [Citado el: 14 de Enero de 2011.]  
<<http://www.flumotion.com/esp/index.php>>
33. Jacobson, Ivar, Grady Rumbaugh Booch, and James. *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*.
34. Patrones de caso de uso. [En línea] [Citado el: 15 de Febrero de 2011.]  
<[http://www.ecured.cu/index.php/Patrones\\_de\\_Casos\\_de\\_Uso](http://www.ecured.cu/index.php/Patrones_de_Casos_de_Uso)>
35. Base de Datos para el Control e Inventario del Mobiliario y Personal de una empresa. [En línea] [Citado el: 14 de Marzo de 2011.]  
<<http://issuu.com/tepuyes/docs/amanda-edyan>>.
36. MVC - Modelo Vista Controlador - Articulos Otros - Programación Web. [En línea] [Citado el: 16 de Marzo de 2011.]  
<<http://www.programacionweb.net/articulos/articulo/?num=505>>.
37. Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado del Software. s.l. : Addison Wesley, 2000. [Online]

### Glosario

**CASE:** Ingeniería de software asistida por computadora. (*Computer Aided Software Engineering*).

**UML:** Lenguaje Unificado de Modelado (*Unified Modeling Language*)

**Bouncer:** Es el componente que se encarga de la seguridad en el Servidor Streaming flumotion.

**Escalabilidad:** Grado en el que una aplicación informática o componente puede ser ampliado en tamaño, volumen o número de usuarios y aún pueda funcionar correctamente.

**HDD:** *Hard Disk Device*, forma inglesa de denominar al disco duro.

**HTTPS:** Protocolo seguro de transferencia de hipertexto (*Hypertext Transfer Protocol Secure*), es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

**API:** Una interfaz de programación de aplicaciones o API (por sus siglas en inglés de *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

**Gstreamer:** Es un framework multimedia libre multiplataforma escrito en el lenguaje de programación C. GStreamer permite crear aplicaciones multimedia, como de vídeo, sonido, codificación, etc. Por ejemplo, con GStreamer se puede reproducir música o realizar tareas más complejas como mezclar audio y vídeo.

**XML:** Lenguaje de marcas extensible (*eXtensible Markup Language*)