



**Universidad de Las Ciencias Informáticas**

**Facultad 6**

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

**TÍTULO:**

*Desarrollo de un componente para el control de acceso a publicaciones de archivos multimedia en servidores de streaming libres.*

**AUTOR: Adaily Hernández Carballé.**

**TUTOR: Ing. Angel Dayán Marín Abreu.**

La Habana, Junio 2011

“Año del 53 de la Revolución”



*“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido...”*

*Che*

*A tata,*

*Para que donde quieras que estés, sepas que te llevo siempre en mi corazón.*

*A mis padres,*

*Por su apoyo y por ese amor tan grande que me han brindado siempre, los quiero.*

*A mi hermanita,*

*Porque es la fuerza que me inspira a ser cada día mejor.*

*A Sule y su familia,*

*Por su cariño y apoyo incondicional.*

*A Yoandrys,*

*Por su amor, su apoyo y comprensión.*

## Agradecimientos

---

Quiero agradecer primeramente a mi abuelita tata, que la vida me arrebató antes que pudiera verme convertida en la mujer que hoy soy, te agradezco por educarme, por tu ejemplo de mujer luchadora, por ese amor infinito que me brindaste. Abuela donde quieras que estés gracias.

A mis padres por haberme apoyado siempre, por todo el amor que me han dado. A ustedes que luchan cada día por mi porvenir, a ustedes les debo todo lo que soy, porque ustedes son mi guía y mi ejemplo a seguir en la vida. A mi hermanita que es la razón por la que luchó para ser mejor cada día, para que mi esfuerzo le sirva como guía para su futuro. A ustedes les quiero agradecer porque son mi razón de ser, gracias por su cariño infinito, los quiero mucho.

A mi compañero y amigo Yoandrys, por apoyarme en cada paso que he dado. Gracias por tu comprensión, por estar a mi lado, por brindarme tu mano en los momentos más difíciles y por guiar mis pasos durante todo el camino que hemos recorrido juntos para llegar hasta aquí. Gracias por tu amor, te quiero.

A Sule mi hermana del alma, mi amiga incondicional y a su familia por todo el amor y el apoyo que me han dado, a ustedes muchas gracias por ser tan especiales conmigo. También quiero agradecer a mi mami de proyecto, Ailín por su ayuda y su cariño cuando estuve tan lejos de mi familia.

A mis hermanitas de la universidad Yalili, Yuleimys, Diana, Nixys, Susy, a la Yami por su cariño y su amistad durante todo este tiempo. A las Kan̄kas (Lisandra, Ivelin, Ariadna, Saily, Kryсна, Glenda, Yissel, Bety), por esa amistad tan linda, por la alegría que transmiten y por el apoyo que me han brindado.

Quiero agradecer a los profesores del tribunal y en especial a mi tutor por apoyarme y guiarme, por sus consejos y recomendaciones. También quiero agradecer a Frank por estar siempre dispuesto a ayudarme ante cualquier duda. Agradezco a todas aquellas personas que de una forma u otra me han ayudado y me han apoyado durante toda mi carrera, a toda mi familia, mis amigos y profesores. A todos muchas gracias.

## *Declaración de autoría*

---

Declaro que soy la única autora de este trabajo y autorizo al centro GEYSED de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 24 días del mes de junio del año 2011.

Adaily Hernández Carballé

---

Angel D. Marín Abreu

---

**Datos generales del tutor**

**Nombre y apellidos:** Angel Dayán Marín Abreu.

**Categoría docente:** Adiestrado.

**Centro de trabajo:** Universidad de las Ciencias Informáticas.

**Cargo:** Profesor.

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

**Año de graduación:** 2009

**Institución en la que se graduó:** Universidad de las Ciencias Informáticas.

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

Debido a la gran cantidad de usuarios que acceden a diario a las aplicaciones existentes en la red el control de acceso es fundamental. La información que se encuentra publicada en los servidores para su transmisión debe ser protegida mediante un estricto control de acceso. Actualmente este control se realiza según las técnicas implementadas en los principales servidores tales como Darwin y Flumotion. Estos servidores carecen de una interfaz que permite una mayor flexibilidad, usabilidad e interacción con las aplicaciones que estos vinculan, lo que trae aparejado que la aplicación de dichas técnicas resulte un proceso muy complicado para los usuarios que administran dichos sistemas.

El presente trabajo tiene como objetivo la implementación de un sistema que vinculará las principales técnicas que se utilizan en estos servidores streaming. Que posea una interfaz gráfica, la cual posibilitará al usuario mayor flexibilidad y usabilidad sobre la implantación de las técnicas para el control de acceso. Para lograr este objetivo se realizó un estudio sobre las técnicas de control de acceso aplicadas por los servidores streaming. Se definieron además las metodologías y herramientas más apropiadas para el desarrollo del sistema, entre las que se encuentran RUP como metodología de desarrollo de software, PHP como lenguaje de programación, PostgreSQL como gestor de base de datos, Netbeans como IDE de desarrollo entre otros. Las pruebas realizadas al componente arrojaron resultados positivos, demostrando de esta forma que el mismo cumple con las funcionalidades definidas, logrando el objetivo principal de la investigación.

### **Palabras claves**

Control de acceso, flexibilidad, metodología, servidores streaming, usabilidad.

Due to the large number of users accessing applications on a daily basis on the network access control is essential. The information is posted on the servers for transmission must be protected by strict access control. Currently, this control is performed according to the techniques implemented in major servers such as Darwin and Flumotion. These servers do not have an interface that allows greater flexibility, usability and interaction with applications that link these, which brings the application of these techniques is a very complicated for users who manage those systems.

This work aims to implementing a system that will link the main techniques used in these streaming servers. That has a graphical interface, which will allow the user greater flexibility and usability on the implementation of techniques for access control. To achieve this, a study on access control techniques applied by the streaming server. It further defines the most appropriate methodologies and tools for system development, among which are RUP as software development methodology, PHP programming language, PostgreSQL as database manager, Netbeans IDE development among others. The component tests were positive, thus demonstrating that it meets the defined features, achieving the main objective of the investigation.

### **Keywords**

Access Control, flexibility, methodology, streaming servers, usability.



Introducción.....	1
Capítulo 1. Fundamentación teórica. ....	5
Introducción.....	5
1.1    Análisis del estado del arte.....	5
1.2    Aplicaciones Web.....	5
1.2.1    Ventajas del uso de aplicaciones web:.....	5
1.3    Tecnología streaming.....	6
1.4    Técnicas para el control de acceso.....	6
1.5    Servidores streaming.....	7
1.5.1    Darwin Streaming Server (DSS).....	7
1.5.2    Flumotion.....	9
1.6    IPtables como técnica de control de acceso a servidores unix.....	11
1.7    Proceso Unificado de Desarrollo de Software (RUP).....	12
1.8    Lenguaje de programación.....	14
1.9    El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.....	15
1.10    PostgreSQL como sistema gestor de bases de datos.....	15
1.11    Herramientas.....	16
1.12    Conclusiones.....	17
Capítulo 2. Características del sistema.....	18
Introducción.....	18
2.1    Modelo de Dominio.....	18
2.1.1    Descripción del modelo de dominio.....	18
2.1.2    Definición de las clases del modelo de dominio.....	18
2.1.3    Diagrama de clases del modelo de dominio.....	19

2.2	Requisitos .....	19
2.2.1	Requisitos Funcionales.....	20
2.2.2	Requisitos no Funcionales.....	21
2.3	Descripción del sistema propuesto.....	22
2.3.1	Actores del sistema.....	22
2.3.2	Casos de Uso del Sistema.....	22
2.3.3	Diagrama de Casos de Usos del Sistema.....	23
2.3.4	Descripción textual de Casos de Usos.....	23
2.4	Conclusiones.....	48
Capítulo 3. Análisis y diseño del sistema.....		50
Introducción.....		50
3.1	Modelo de Análisis.....	50
3.1.1	Diagrama de clases del análisis.....	50
3.2	Modelo de Diseño.....	53
3.2.1	Diagramas de clases del diseño.....	53
3.3	Patrones utilizados.....	57
3.3.1	Patrón Modelo-Vista-Controlador.....	57
3.3.2	Patrones GRASP.....	58
3.3.3	Patrones GOF.....	59
3.4	Diseño de la Base de datos.....	59
3.4.1	Diagrama de clases persistentes.....	59
3.4.2	Diagrama entidad relación.....	60
3.5	Conclusiones.....	60
Capítulo 4. Implementación y prueba del sistema.....		61
Introducción.....		61

4.1	Modelo de despliegue. ....	61
4.2	Modelo de Implementación. ....	61
4.3	Prueba.....	62
	Descripción de las variables CU. Gestionar usuario. ....	63
	4.3.1 Resultado de las pruebas.....	66
4.4	Conclusiones .....	66
5.1	Conclusiones .....	67
	Recomendaciones .....	68
	Bibliografía consultada .....	69
	Bibliografía citada.....	70

FIGURA 1: CAMINO QUE SIGUE UN PAQUETE EN EL KERNEL (12).....	11
FIGURA 2: FASES Y FLUJOS DE TRABAJO DE RUP (13).....	14
FIGURA 3: MODELO DE DOMINIO.....	19
FIGURA 4. DIAGRAMA DE CASOS DE USOS DEL SISTEMA.....	23
FIGURA 5. DESCRIPCIÓN CU AUTENTICAR USUARIO.....	25
FIGURA 6. DESCRIPCIÓN CU GESTIONAR USUARIOS.....	29
FIGURA 7: DESCRIPCIÓN CU GESTIONAR GRUPO DE USUARIOS.....	34
FIGURA 8: DESCRIPCIÓN CU GESTIONAR SERVIDOR.....	39
FIGURA 9: DESCRIPCIÓN CU GESTIONAR PERMISO.....	43
FIGURA 10: DESCRIPCIÓN CU GESTIONAR TIPO DE SERVIDOR.....	45
FIGURA 11: DESCRIPCIÓN CU GESTIONAR IP O RANGO IP.....	48
FIGURA 12: DCA AUTENTICAR USUARIO.....	51
FIGURA 13: DCA GESTIONAR USUARIO.....	51
FIGURA 15: DCA GESTIONAR SERVIDOR.....	52
FIGURA 16: DCA GESTIONAR TIPO DE SERVIDOR.....	52
FIGURA 17: DCA GESTIONAR IP O RANGO IP.....	52
FIGURA 18: DCD AUTENTICAR USUARIO.....	54
FIGURA 19: DCD GESTIONAR USUARIO.....	54
FIGURA 20: DCD GESTIONAR GRUPO DE USUARIO.....	55
FIGURA 21: DCD GESTIONAR SERVIDOR.....	55
FIGURA 22: DCD GESTIONAR TIPO DE SERVIDOR.....	56
FIGURA 23: DCD GESTIONAR PERMISO.....	56
FIGURA 24: DCD GESTIONAR IP O RANGO IP.....	57

## **Introducción**

La creación, búsqueda y obtención de información creada, expresada y transmitida por otros son acciones esenciales en la naturaleza humana. Desde sus antepasados, expresar pensamientos, ideas, emociones y dejar huella de sí mismo a través de la comunicación, han constituido necesidades básicas para los seres humanos. En sus ansias por comunicarse el hombre comenzó utilizando sus propios medios y a lo largo de su historia ha creado diversas formas para lograr su objetivo.

Esta búsqueda constante por satisfacer su necesidad de comunicación y los avances tecnológicos han sido el impulso fundamental para la instauración en el mundo de numerosos instrumentos y sistemas de comunicación, con mayor eficiencia, capacidad y potencia. Estos sistemas han permitido alcanzar tecnologías impresionantes a las cuales se tiene acceso diariamente.

En la actualidad los grandes avances tecnológicos en los campos de la informática y las comunicaciones han permitido el desarrollo de aplicaciones y tecnologías con diversos fines comunicativos. Internet, también conocida como red de redes, se ha convertido en el medio de comunicación por excelencia y de mayor crecimiento a través de la web. Gracias a su crecimiento vertiginoso cada día surgen nuevos servicios sobre las redes de datos. Actualmente unos de los que mayor demanda ha logrado es la transmisión de vídeo y sonido en la red y en base a lograr mayor eficiencia en esta transmisión se utiliza la tecnología streaming. El uso de esta tecnología permite la transmisión de video y audio en tiempo real (9). En el proceso de streaming no es necesario descargar el archivo en el ordenador del cliente para reproducirlo, sino que la media se reproduce a medida que se está recibiendo, y este a su vez se recibe a una velocidad adecuada para su reproducción. Esta tecnología ofrece una mejor experiencia para los usuarios y un mejor uso del ancho de banda y poder de procesamiento de los distribuidores.

Debido a la cantidad de usuarios que acceden a las aplicaciones publicadas en la red de redes, es necesario restringir el acceso, y de esta forma proteger la información que se maneja, así como el derecho de autor sobre cada una de estas aplicaciones. La seguridad de las aplicaciones es un elemento fundamental de lo que depende la protección del contenido y el control de acceso sobre una información determinada.

Cuba a pesar de ser un país bloqueado avanza con un programa de informatización de la sociedad que involucra a múltiples sectores y aúna diversos esfuerzos. Como parte de este programa surge la Universidad de las Ciencias Informáticas (UCI). Desde sus inicios la UCI tiene como objetivo ser la

primera ciudad digital del país. En base a lograr este objetivo se ha trabajado para ofrecer a sus estudiantes, profesores y trabajadores en general una amplia gama de servicios informatizados llevando la vanguardia en lo referente a desarrollo tecnológico en el país.

Actualmente la universidad está inmersa en el desarrollo de software para su exportación y en el proceso de informatización de la sociedad cubana, para ello cuenta con centros de desarrollo, los cuales tienen como principal objetivo contribuir a la formación de futuros profesionales en la informática y elevar la producción de software en la universidad y el país. El centro GEYSED de la facultad 6 en dicha universidad tiene dentro de sus proyectos de Investigación + Desarrollo e Innovación, el referente a **VideoWeb**, el cual enfoca sus productos a la transmisión de contenido multimedia a través de la web. El contenido publicado en el servidor para su transmisión debe ser protegido mediante un estricto control de acceso. Actualmente el control de acceso se realiza según las técnicas implementadas en los principales servidores y frameworks utilizados, ya sean servidores como Darwin Streaming Server, Flumotion o frameworks como Symfony entre otros. Estos servidores carecen de una interfaz que permita una mayor flexibilidad y usabilidad en las aplicaciones, esto trae consigo que aplicar dichas técnicas se convierta en un proceso engorroso para cualquier cliente. La situación problemática descrita arroja el siguiente **problema de la investigación**: ¿Cómo lograr mayor flexibilidad en el control de acceso a los archivos multimedia publicados en servidores de streaming libres? Para resolver dicho problema es necesario estudiar las técnicas para el control de acceso a información publicada en servidores. Para darle cumplimiento al mismo se definen el siguiente **objetivo** para la presente investigación: Desarrollar un componente empleando tecnología web que permita una mayor flexibilidad en el control de acceso a los archivos multimedia publicados en servidores de streaming libres. Este objetivo es aplicado al **campo de acción**, técnicas para el control de acceso a los archivos multimedia publicados en servidores de streaming libres dentro del departamento de señales digitales.

Se propone la **Idea a defender**

La implementación de un componente para el control de acceso a los archivos multimedia publicados en servidores de streaming libres permitirá una mayor flexibilidad y usabilidad en la restricción de accesos no autorizados a dichos archivos.

Durante la investigación y con el fin de lograr los objetivos del presente trabajo de diploma se plantean las siguientes **Tareas de la investigación**:

1. Definir y caracterizar las técnicas de control de acceso a la información publicada en servidores.
2. Definir y caracterizar iptable como técnica de control de acceso a servidores Unix.
3. Valorar las herramientas y tecnologías a utilizar en la construcción de la solución.
4. Seleccionar y argumentar la Metodología de Desarrollo de Software a usar en el proceso.
5. Identificar las funcionalidades que debe brindar el componente.
6. Realizar el Modelo de Casos de Uso del componente.
7. Diseñar el componente.
8. Implementar el componente.
9. Realizar pruebas al componente.

Dentro de los posibles resultados están:

- Documentación que recoja el resultado de la investigación realizada.
- Documentación UML de los artefactos resultantes del análisis y diseño.
- Componente para el control de acceso a los archivos multimedia publicados en un servidor de streaming libre desarrollado empleando tecnologías web.
- Manual de instalación y uso del componente.

Para el desarrollo de la presente investigación se tendrán en cuenta los siguientes métodos de la investigación científica.

Dentro de los **teóricos** se utilizaron los siguientes:

- **Analítico-Sintético** para enunciar y describir las características de las principales técnicas de control de acceso los servidores streaming libres.
- **Inductivo-deductivo** para llegar a un grupo de conclusiones particulares sobre lo que se quiere lograr gracias a la obtención de un conocimiento general acerca de las principales técnicas para el control de acceso.

Dentro de los **empíricos** se empleó el siguiente:

- **La entrevista** a miembros del centro GEYSED se efectuó con el objetivo de conocer la necesidad de desarrollar un componente para el control de acceso a los archivos multimedia publicados en un servidor streaming, además de conocer como se realiza el control de acceso a la información publicada en dichos servidores actualmente. Ver anexo 1.

Los datos obtenidos constituyen la base para lograr el desarrollo de un software que sea de gran utilidad, para ello han sido seleccionados para entrevistar a líderes y jefes de líneas del proyecto VideoWeb. La población escogida son 4 integrantes del proyecto VideoWeb del Centro GEYSED. De ellos se toman como muestra 2 individuos para ser entrevistados, lo que representa el 50 % de la población. Se realiza una técnica de muestreo no probabilístico, específicamente muestreo intencional.



## Capítulo 1. Fundamentación teórica.

### Introducción

En el presente capítulo se realizará un análisis sobre algunas de las tecnologías y herramientas que servirán de sustento a la investigación. Se realizarán una serie de análisis de las características y funcionalidades que permitirán llevar a cabo las bases para la solución propuesta. Además se abordarán algunos conceptos asociados al lenguaje de programación, sistema gestor de base de datos, la metodología de desarrollo entre otros.

#### 1.1 Análisis del estado del arte

Luego de haber realizado un minucioso estudio no se encontró un componente que reúna las principales técnicas de control de acceso a los contenidos publicados en servidores de streaming existentes. Actualmente el control de acceso se realiza mediante las técnicas implementadas por los propios servidores, las cuales no cuentan con una interfaz que le posibilite al usuario la configuración y aplicación de las mismas. Con la implementación del sistema que se propone se daría solución a lo planteado con anterioridad, ya que brindaría la posibilidad de integrar las técnicas de control de acceso principales y más utilizadas, lo que daría como resultado una aplicación estándar para los principales servidores streaming libres y a la vez su interfaz le daría una mayor flexibilidad y usabilidad a la aplicación a la hora de ser usados por los diferentes usuarios.

#### 1.2 Aplicaciones Web

Debido al crecimiento vertiginoso de Internet, la mayoría de las empresas están aprovechando las indudables ventajas que ofrecen las aplicaciones web, término que se utiliza para una serie de sistemas de software que cumplen con algunas características propias y que está relacionado con Internet. Una aplicación web es aquella que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Un conjunto de páginas generadas, según una secuencia de parámetros introducidos por el usuario (1).

##### 1.2.1 Ventajas del uso de aplicaciones web:

- No requieren instalarse en los clientes, el uso de esta tecnología permite el aprovechamiento de muchas de las características de Internet.
- Son fáciles de usar (no requieren conocimientos avanzados de computación).

- Alta disponibilidad, ya que se pueden realizar consultas desde cualquier parte del mundo donde se tenga acceso a Internet y a cualquier hora.
- Facilitan la comunicación entre gobierno, instituciones educativas, empresas, asociaciones y personas físicas, con el propósito de establecer una relación aún más estrecha entre ellos.
- Particularmente en el aspecto comercial, fomentan una mayor comunicación entre clientes y empresas estableciendo un modelo de operación del negocio más orientado al cliente.

### **1.3 Tecnología streaming.**

La tecnología streaming permite la visualización y escucha de contenidos multimedia (audio + video) sin necesidad de descargarlo para la computadora del cliente. Permite además distintas formas de transmisión del contenido, ya sea transmisión en tiempo real o bajo demanda. Los datos son procesados en la computadora del cliente como un flujo continuo y constante, mostrando la información transmitida antes de que el archivo se haya descargado en su totalidad, como una transmisión de televisión o de radio (9).

El proceso de streaming se puede dividir en dos categorías, en función de cómo se obtiene la información que se va a difundir:

- Streaming en directo.
- Streaming bajo demanda.

Para la transmisión de radio y televisión por Internet se necesita difundir la información en vivo, aunque al mismo tiempo pueden brindar acceso bajo demanda a los archivos de sus programas grabados.

### **1.4 Técnicas para el control de acceso.**

Las técnicas para el control de acceso permiten garantizar la seguridad de los ficheros almacenados en un servidor. Estas definen una serie de reglas sobre la interacción y acceso a los archivos de datos, que pueden ser archivos de audio, video, imagen, etc. La integración de estas técnicas a sistemas de gestión permite asignar responsabilidades específicas a usuarios o grupos de usuarios, también puede definir la forma en la que los usuarios acceden a los archivos a través de la red.

Los principales tipos de restricciones son:

Limitación de acceso en función de direcciones IP o dominio. Sólo los usuarios de un dominio u organización tendrán acceso a la información.

Limitación de acceso por nombres de usuario y claves de acceso. Sólo los usuarios que conozcan una clave de acceso válida pueden acceder a la información.

### 1.5 Servidores streaming.

Para publicar medias no basta con solo subir un archivo a un servidor y permitir que las personas accedan al mismo desde el navegador o algún servicio implementado. En determinados casos como la puesta en marcha de una radio o la transmisión de un evento en directo, se hace imprescindible disponer de un servidor de streaming al cual mandar la señal y con ella, la enviará a todos los clientes. Existen servidores especiales preparados para transmitir streaming, que pueden ofrecer importantes prestaciones, entre los que se encuentran Darwin Streaming Server y Flumotion.

#### 1.5.1 Darwin Streaming Server (DSS).

Es un servidor que permite el envío de información, utilizando el formato QuickTime, a través de los protocolos RTP y RTSP, que son estándares en la industria para la transmisión de streaming. Se puede configurar el acceso de los clientes sobre los archivos multimedia para hacer el streaming mediante varias opciones generalmente siguiendo los siguientes indicadores:

1. Protección de contenido por configuración de contraseña.
2. Limitando el acceso a las medias a través de un fichero de configuración llamado *qtaccess*.
3. Mediante la encriptación para la comunicación Web con herramientas del servidor de streaming.
4. Mediante el streaming por el puerto 80.

La autenticación de los usuarios es el método tradicional para establecer el control de acceso a determinado objetivo, en este caso ficheros de medias. Dentro de las técnicas que soporta Darwin Streaming Server están:

Basic Authentication (.htaccess)

**HTTP Digest Access Authentication.** Ésta última viene configurada por defecto y es más segura. Consiste en un método que implementan los servidores web para el control por credenciales con usuarios que usan el protocolo Http. Usando Basic Access Authentication, permite al usuario establecer la seguridad sin tener que cifrar las contraseñas. Esto puede verse como una debilidad ya que trae como consecuencia que sea posible tener acceso al fichero y ver las contraseñas de los usuarios.

**Digest access Authentication** es una técnica que utiliza cifrado MD5, en una dirección para prevenir el criptoanálisis y ataques posteriores a los ficheros de medias.

También permite establecer las listas de control de acceso para administrar el servidor de streaming.

Para el control sobre los archivos, el fichero de acceso debe estar presente en el directorio donde estarán las medias. En el caso que el fichero de acceso no esté dentro del directorio, todos los usuarios tendrán acceso a todas las medias presentes en el directorio.

Para establecer el control de acceso:

1. Para crear nuevos usuarios y contraseñas a través de una consola dependiendo del sistema operativo se debe escribir el comando `qtpasswd`, este comando se instala con el servidor.
2. Se debe crear el fichero de acceso `qtaccess` y copiarlo dentro del directorio donde están las medias que desea proteger.
3. Si desea deshabilitar la autenticación para un directorio de medias debe eliminar el fichero de acceso, o cambiarle el nombre, por ejemplo si se llama `qtaccess` debería poner `qtaccess.disabled`.

Crear un archivo de Acceso.

Se crea el archivo `qtaccess`, el cual contiene información sobre los usuarios y los grupos de usuarios autorizados para ver los medios de comunicación en el directorio en el que se almacena los archivos sobre los cuales se solicita el acceso. Teniendo en cuenta que un directorio de almacenamiento puede contener varios directorios, se puede crear para cada directorio su propio fichero de acceso.

Cuando los usuarios solicitan acceder a un archivo determinado, el servidor verifica el fichero de acceso (`qtaccess`) para comprobar el acceso del usuario a determinada media. En caso de no existir

tal fichero, el servidor busca en el directorio y si no encuentra el fichero de acceso entonces se hace una búsqueda y el primer archivo que encuentra lo utiliza para determinar si el usuario está autorizado para acceder a ese archivo solicitado.

Ésta técnica utilizada por el Darwin Streaming Server trabaja de la misma manera que lo hace el servidor Apache para establecer el control de usuario mediante fichero de acceso.

Otra forma de abordar el control de acceso y mejorar la seguridad lo permite hacer mediante Secure Sockets Layer (SSL), por sus siglas en inglés. Este es un protocolo que provee la comunicación encriptada mediante Internet. Antes de instalar la seguridad mediante SSL debe tener instalado en el servidor un certificado de seguridad válido para la versión del servidor, a la hora de habilitar este protocolo se debe tener en cuenta que el navegador a utilizar tenga soporte para dicho protocolo.

La otra opción que tiene el Darwin Streaming Server para proteger las medias es a través del uso de Firewall mediante el bloqueo de paquetes a través de la red (10).

### 1.5.2 Flumotion.

Flumotion puede transmitir los contenidos audiovisuales tanto en vivo como bajo demanda en una gran variedad de formatos de audio y video, aunque su versión libre solo permite transmitir los contenidos en formato ogg con codificación de audio vorbis y de video theora.

Este servidor cuenta con una interfaz de usuario simple que permite publicar contenidos rápidamente y de forma sencilla e intuitiva, luego solo es necesario ubicar un vínculo al contenido publicado en el sitio web, sin embargo estos contenidos publicados desde esta interfaz son a manera de prueba, son temporales y solo pueden ser accedidos mientras no se reinicie el servidor.

Flumotion permite que el procesamiento sea distribuido a través de numerosas computadoras, por lo que la plataforma se puede escalar para manipular más usuarios y streaming tanto en vivo como bajo demanda. Su arquitectura de código abierto lo hace más eficiente y más flexible que otros sistemas, haciendo mejor uso del hardware.

#### **Control de acceso a las medias mediante Flumotion Streaming Server:**

**Flumotion** se implementa en forma de componentes, es decir las distintas funcionalidades son agrupadas por componentes, algunos de ellos proveen diferentes funcionalidades como por ejemplo la de control de flujo y administración, que aunque no están involucradas directamente en la transmisión del flujo de datos, juegan un papel importante en el control sobre esos datos.

Algunos ejemplos de estos componentes son:

**Bouncer:** Los componentes de tipo "Bouncer" implementan el mecanismo de autenticación, estos reciben las peticiones de autenticación de otros componentes o administradores y verifican si la acción solicitada tiene permisos. La autenticación se implementa utilizando el fichero `httpasswd`, aunque también puede hacerlo mediante LDAP con usuario y contraseña o algún otro mecanismo. Para instancias del componente "`httpasswdcrypt-bouncer`" controlado por el fichero `httpasswd` se usa el formato de contraseñas encriptadas.

- Mediante la línea de comando puede crear contraseñas utilizando el siguiente comando:  
***htpasswd -d -c passwords someone.***

**Flumotion:** Un sistema flumotion consiste en unos cuantos procesos trabajando juntos. Un proceso trabajador (`worker`) crea procesos para los componentes (`components`), y los procesos administradores (`managers`) le dicen a los trabajadores qué hacer. La interfaz de administración de flumotion se conecta al administrador, el cual por su parte controla a los trabajadores indicándoles qué componentes iniciar o detener.

Después de iniciado el sistema, el administrador manda a iniciar un componente de tipo Bouncer utilizado para autenticar los trabajadores y los componentes.

El directorio "`/etc/flumotion`" contiene directorios para los administradores (`managers`) y los trabajadores (`workers`) los cuales contienen archivos XML. Estos archivos pueden estar en subdirectorios y pueden ser divididos en archivos XML separados. El directorio de los administradores "`/etc/flumotion/managers`" contiene un subdirectorio para cada administrador (usualmente solo uno), cada uno de estos subdirectorios contiene un archivo llamado "`planet.xml`". Este archivo `planet.xml` contiene un nodo XML para el planeta, que a su vez contiene nodos para el administrador, la atmósfera y los flujos del planeta, los cuales a su vez contienen nodos para los componentes.

En el nodo del administrador se puede especificar el nombre de host, el puerto y el protocolo de transporte que se utilizará para la comunicación entre el administrador y los trabajadores, pero también se pueden utilizar los valores por defecto. Siempre se debe utilizar el protocolo `ssl` establecido por defecto para lograr conexiones seguras. Especificando "`localhost`" como nombre de host para el manager se puede restringir el acceso a los trabajadores que están corriendo en la misma máquina, esto no es apropiado para un sistema flumotion real.

## Componentes de Flumotion utilizados para el control de acceso:

- **ip-bouncer:** Restringir el acceso por usuario y para direcciones o rangos IP.
- **ical-bouncer:** Permite el acceso al streaming mediante un evento de un archivo de calendario (iCalendar).
- **bouncer-trivial:** Siempre acepta las autenticaciones requeridas (11).

### 1.6 IPTables como técnica de control de acceso a servidores unix.

IPTables es un sistema de firewall vinculado al kernel de Linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. IPTables está integrado con el kernel, es parte del sistema operativo. Para ponerlo en marcha, se aplican reglas y para ello se ejecuta el comando iptables, con el que se añade, borran, o crean reglas. Por ello iptables no es un simple script de shell en el que se van ejecutando las reglas de firewall.

Las reglas de firewall están a nivel de kernel, al cual le llegan paquetes sobre los que tiene que decidir qué hacer. En dependencia de para quien este enviado el paquete se hacen consultas a dichas reglas anteriormente definidas y decide qué hacer con cada paquete.

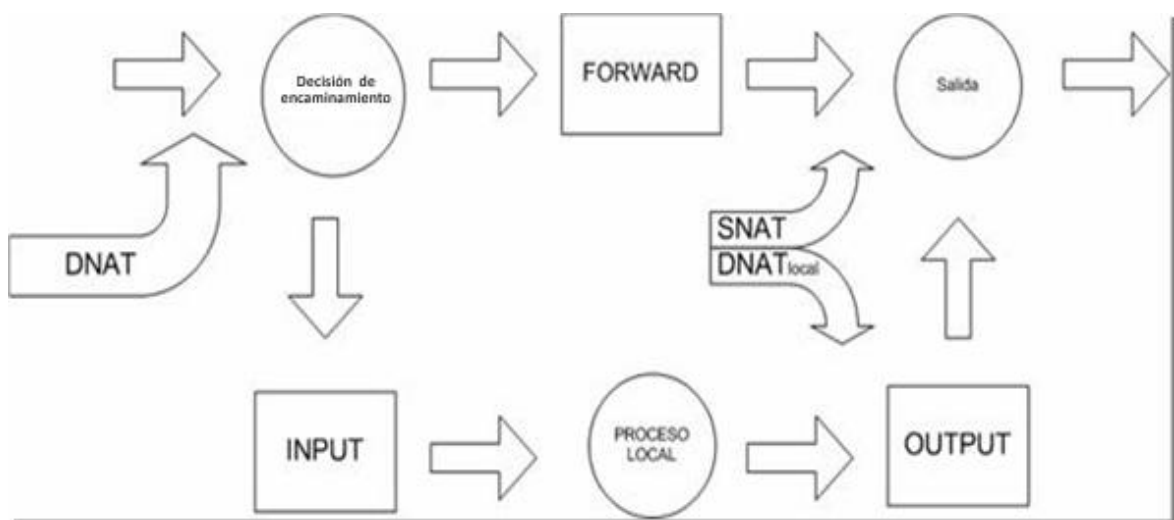


Figura 1: Camino que sigue un paquete en el kernel (12).

Para los paquetes o datagramas, según el protocolo que van a la propia máquina se aplican las reglas INPUT y OUTPUT, y para filtrar paquetes que van a otras redes o máquinas se aplican simplemente reglas FORWARD.

INPUT, OUTPUT y FORWARD son los tres tipos de reglas de filtrado. Pero antes de aplicar esas reglas es posible aplicar reglas de NAT: estas se usan para hacer redirecciones de puertos o cambios en las IPs de origen y destino, también antes del NAT se pueden aplicar reglas de tipo MANGLE, que están destinadas a modificar los paquetes.

Reglas en iptables:

- MANGLE
- NAT: reglas PREROUTING, POSTROUTING
- FILTER: reglas INPUT, OUTPUT, FORWARD (12).

### 1.7 Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Unificado de Desarrollo de Software (RUP en sus siglas en inglés correspondientes a *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP es un proceso de ingeniería de software planteado por Kruchten en 1996 que tiene como objetivo producir un software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido. RUP cubre el ciclo de vida y desarrollo de software (13).

El proceso unificado es un proceso de desarrollo de software y su ciclo de vida está caracterizado por:

- Dirigido por casos de usos.
- Centrado en la arquitectura.
- Iterativo e incremental.

RUP consta de cuatro fases o etapas, además el ciclo de vida de esta metodología contiene flujos de trabajos. Ver figura 2.

#### Fases

**1. Comienzo o Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.



**2. Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

**3. Construcción:** Se completa el desarrollo del sistema y se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario.

**4. Transición:** Se garantiza que esté listo para ser utilizado. Puede implicar reparación de errores. Se capacitan a los usuarios y brinda soporte técnico.

### **Flujos de trabajo**

- Modelado del negocio.
- Análisis de requisitos.
- Análisis y diseño.
- Implementación.
- Pruebas.
- Distribución.
- Gestión de configuración y cambios.
- Gestión del proyecto.
- Gestión del entorno

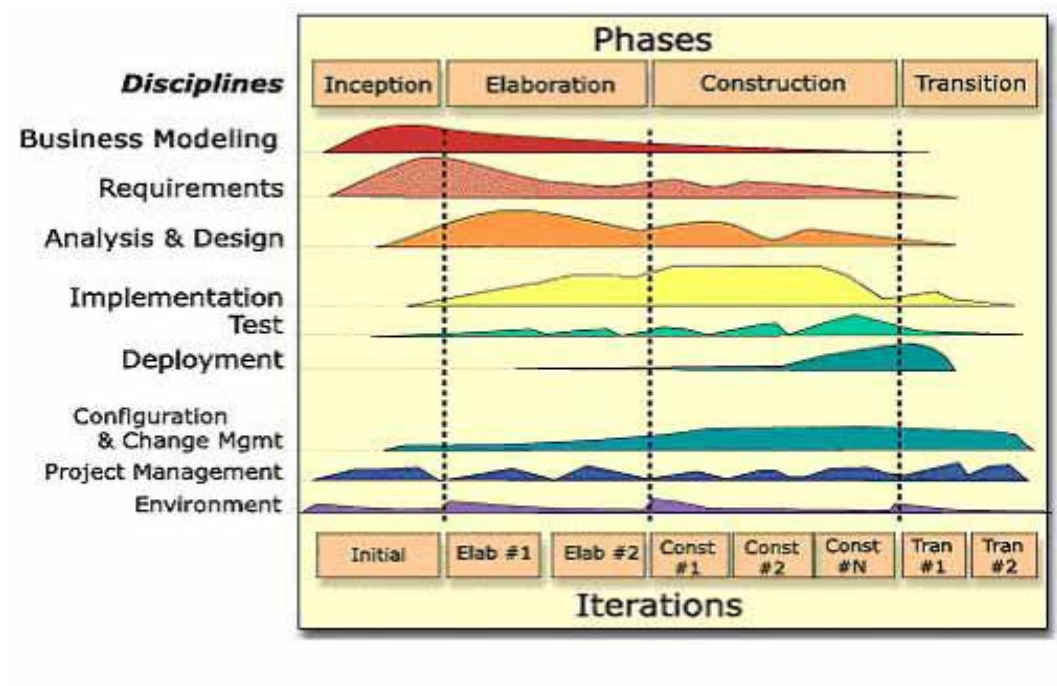


Figura 2: Fases y flujos de trabajo de RUP (13).

## 1.8 Lenguaje de programación.

PHP (Personal Home Page) es el acrónimo de Hypertext Preprocessor. Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor diseñado específicamente para la web. Es un lenguaje de programación gratuito y multiplataforma, con una gran librería de funciones y mucha documentación. Es un lenguaje completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. Se destaca por su alta capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, principalmente con MySQL y PostgreSQL. Permite aplicar técnicas de programación orientada a objetos.

PHP cuenta con un motor de plantillas denominado SMARTY que permite separar la lógica de la programación de la presentación, es decir, el código PHP del código HTML. Por tanto se puede modificar uno sin afectar el otro. SMARTY es sumamente rápido, entre otros aspectos porque cada plantilla se compila sólo una vez y sólo recompila aquellas que han sido modificadas. Permite que una misma plantilla sea utilizada por varias páginas PHP que muestren el contenido en el mismo formato, independientemente de que los procedimientos usados para obtener la información a mostrar sean diferentes.

Otra de las ventajas de utilizar PHP, es poder usar las librerías de acceso a datos (AD) que permitan realizar una abstracción de las base de datos y que muestran mayor rapidez en el procesamiento de sentencias SQL. Posee además características como la gestión de sesiones, generación automática del código SQL, simulación de SELECT LIMIT para todas las bases de datos y monitorización del rendimiento. PHP tiene una de las comunidades más grandes en Internet, por lo que es fácil encontrar ayuda, documentación, artículos, noticias, y demás recursos.

### **1.9 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.**

Entre los lenguajes de modelado que define OMG (Object Management Group) uno de los más conocidos y usado es sin duda UML (Unified Modelling Language), este es uno de los estándares más utilizado para especificar y documentar cualquier sistema de forma precisa. UML es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño orientado a objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

### **1.10 PostgreSQL como sistema gestor de bases de datos.**

PostgreSQL es un sistema gestor de bases de datos relacional orientado a objeto que se deriva del proyecto POSTGRES surgido en la Universidad de Berkeley California, dirigido por el profesor Michael Stonebraker, distribuida bajo la licencia BSD. PostgreSQL constituye un potente gestor de base de datos de código abierto muy avanzado, ofrece un control de concurrencia multiversión (MVCC por sus siglas en inglés) lo que mejora las operaciones de bloqueo y las transacciones en sistemas multiusuario. Soporta casi toda la sintaxis SQL, incluyendo subconsultas, transacciones, tipos y funciones definidas.

PostgreSQL se destaca por una amplia lista de prestaciones que le permiten competir con cualquier SGBD comercial entre las que se incluyen:

- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc.
- 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 (14).

Todas estas características hacen que PostgreSQL ofrezca hoy día un amplio y muy útil conjunto de funciones. La conectividad, la velocidad y seguridad, además de su rica interacción con el lenguaje de programación a utilizar para el desarrollo del componente propuesto, hacen que sea compatible y mayor beneficio su uso.

### 1.11 Herramientas.

Visual Paradigm (15) para sustentar la documentación y como modelador visual de la notación UML (Unified Modeling Language), para la confección de los diagramas que se ilustran en este documento. Es una herramienta CASE<sup>1</sup> profesional que soporta el ciclo de vida completo del desarrollo de software. También brinda un diseño centrado en casos de uso y enfocado al negocio lo que genera un software de mayor calidad. Posee una disponibilidad de múltiples versiones, además de integrarse en los principales IDEs incluyendo NetBeans que es el utilizado para el desarrollo del

---

<sup>1</sup> Las **herramientas CASE** (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

componente (16). NetBeans es una herramienta que impulsa el desarrollo de la tecnología servidor. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación.

### **1.12 Conclusiones**

Cada día surgen nuevas tecnologías y herramientas para la realización de aplicaciones de software, lo que trae consigo que las exigencias de mercado y la necesidad de nuevas soluciones aumenten, es por esto que los desarrolladores se ven obligados a mantenerse actualizados sobre el tema. En el presente capítulo se hace un minucioso estudio para conocer qué tecnología y herramientas son las más apropiadas para el desarrollo de un componente que garantice el control de acceso a la información que se encuentra publicada en los servidores streaming. Una vez concluido el estudio que dio a conocer, las tecnologías, herramientas y metodología que serán utilizadas a lo largo del desarrollo del componente propuesto, se realizó un análisis de las principales técnicas de control de acceso utilizadas por los servidores streaming. Se definió PHP como el lenguaje de programación a utilizar para el desarrollo del sistema, UML como lenguaje de modelado, el cual brinda soporte para visualizar, construir y documentar el componente. Rational Unified Process (RUP) como proceso de desarrollo de software que junto con el UML, constituye la metodología estándar más utilizada y efectiva para el análisis, implementación y documentación de un sistema.

### **Capítulo 2. Características del sistema.**

#### **Introducción**

En este capítulo se realiza un análisis del sistema propuesto para dar solución a la problemática existente. Debido a que los procesos del negocio no se encuentran definidos con claridad se decide realizar un Modelo de Dominio, el mismo abarca las definiciones y las relaciones entre los conceptos relacionados con el entorno que enmarca el sistema. Se definen y plantean los requisitos funcionales y no funcionales, además de la descripción textual de los casos de uso del sistema.

#### **2.1 Modelo de Dominio.**

El modelo de dominio ayuda a comprender los conceptos con los que trabajará el usuario y con los que debe trabajar la aplicación, brinda una representación visual del entorno de la aplicación. Es un diagrama donde se representan los objetos reales relacionados con la aplicación y las relaciones entre ellos, además de exponer los conceptos existentes en el negocio.

##### **2.1.1 Descripción del modelo de dominio.**

El usuario es el encargado de definir y aplicar las técnicas para realizar el control de acceso, para ello el sistema verifica los datos del mismo ya sea nombre de usuario y contraseña, a través de grupos de usuarios o IP y rangos de IP, el sistema dependiendo del tipo de servidor streaming, verifica la validez de estos datos en una Base de Datos o fichero de configuración. Esto le permite al usuario acceder a las diferentes funcionalidades de la aplicación para restringir el acceso a las medias que se encuentran almacenadas en el servidor.

##### **2.1.2 Definición de las clases del modelo de dominio.**

**Usuario:** es la persona que interactúa con la aplicación para definir y aplicar las diferentes técnicas para el control de acceso a las medias.

**Servidor streaming:** en este servidor es donde estarán almacenadas las medias a las cuales se le debe restringir el acceso.

**Medias:** son los archivos de audio y video que se encuentran almacenados en el servidor.

**Archivo de audio:** un contenedor de multimedia que guarda archivos de audio que pueden ser música, un discurso etc.

**Archivo de video:** un contenedor de multimedia que guarda archivos de video como son películas, series, videos musicales entre otros.

**Base de Datos:** es la entidad donde será almacenada de manera sistemática y estructurada la información.

**Fichero:** es utilizado por los servidores de streaming para almacenar los nombres de usuario y contraseñas de los usuarios autorizados a acceder a un directorio dentro del servidor.

**Control de acceso:** es la entidad que se encarga de realizar las funcionalidades para definir y aplicar las técnicas para realizar el control de acceso.

### 2.1.3 Diagrama de clases del modelo de dominio.

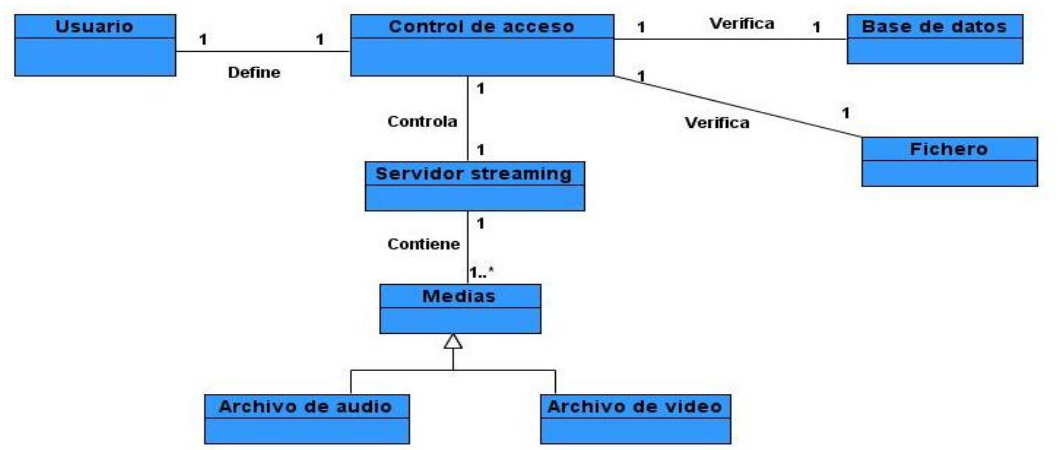


Figura 3: Modelo de dominio

## 2.2 Requisitos

La IEEE define un requisito como: condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

El objetivo fundamental del flujo de trabajo de Requisitos es guiar el desarrollo hacia el sistema correcto, los requisitos pueden dividirse en dos:

**Requisitos Funcionales:** definen las funciones, capacidades o condiciones que el sistema debe ser capaz de realizar, no son más que una descripción de las necesidades de un producto. (17)

**Los Requisitos no Funcionales:** son aquellos requisitos que surgen con las necesidades del usuario y definen las restricciones y propiedades del sistema (18).

### **2.2.1 Requisitos Funcionales.**

**RF 1-Insertar usuarios:** el sistema debe permitir insertar un nuevo usuario.

**RF 2-Modificar usuarios:** el sistema debe permitir modificar los datos de un usuario existente.

**RF 3-Eliminar usuarios:** el sistema debe permitir eliminar un usuario existente.

**RF 4-Insertar grupo de usuarios:** el sistema debe permitir a los usuarios autorizados adicionar un nuevo grupo de usuarios.

**RF 5-Modificar grupo de usuarios:** el sistema debe permitir a los usuarios autorizados modificar un grupo de usuarios existente.

**RF 6-Eliminar grupo de usuarios:** el sistema debe permitir a los usuarios autorizados eliminar un grupo de usuarios existente.

**RF 7-Insertar permiso:** el sistema debe permitir a los usuarios autorizados insertar un permiso.

**RF 8-Modificar permiso:** el sistema debe permitir a los usuarios autorizados modificar un permiso existente.

**RF 9-Eliminar permiso:** el sistema debe permitir a los usuarios autorizados eliminar un permiso existente.

**RF 10-Insertar servidor:** el sistema debe permitir al administrador insertar los datos de un nuevo servidor.

**RF 11-Modificar servidor:** el sistema debe permitir modificar los datos de un servidor existente.

**RF 12-Eliminar servidor:** el sistema debe permitir eliminar un servidor existente.

**RF 13-Insertar un IP o rango IP:** el sistema debe permitir al administrador insertar un IP o rango de IP nuevo.

**RF 14-Eliminar IP o rango IP:** el sistema debe permitir eliminar un IP o rango existente.

**RF 15-Insertar tipo de servidor:** el sistema debe permitir al administrador insertar el tipo de servidor al que se le aplicarán las técnicas para el control de acceso.



**RF 16-Eliminar tipo de servidor:** el sistema debe permitir al administrador eliminar un tipo de servidor existente.

**RF 17-Autenticar usuario:** el sistema debe permitir la autenticación de los usuarios.

### **2.2.2 Requisitos no Funcionales.**

Con el objetivo de que la aplicación posea una mayor calidad se proponen una serie de requisitos no funcionales, que le brindarán a la misma mayor eficacia, a la vez que la harán más confiable, rápida y atractiva para los usuarios.

#### **Usabilidad.**

- La aplicación debe brindar la posibilidad de ser utilizada por personas que posean conocimientos básicos de informática.
- Debe tener un alto grado de flexibilidad y estar disponible las 24 horas al día.

#### **Requisitos de Software.**

Para las PC's clientes.

1. Navegador web Mozilla Firefox 1.5 o superior.

Para los servidores.

- Como sistema operativo GNU/Linux.
- Servidor web el Apache 2.0.
- PostgreSQL como Sistema Gestor de Base de Datos.

#### **Restricciones de diseño e implementación.**

- El lenguaje de programación que se debe utilizar es PHP.
- El patrón arquitectónico que se debe emplear en el desarrollo es el modelo-vista-controlador.
- Debe estar habilitada la librería PDO.
- En el servidor el usuario www-data debe tener los permisos de súper usuario.

#### **Confiabilidad.**

- Permitir la autenticación segura de los usuarios.

### Interfaz.

- Debe tener una interfaz amigable y con apariencia profesional.
- La interfaz debe tener un diseño sencillo y ser de fácil comprensión para el usuario, a la vez que es intuitiva e interactiva.

### 2.3 Descripción del sistema propuesto.

El sistema está destinado a controlar el acceso a los archivos multimedia que se encuentran publicados en un servidor streaming, el cual reúne las principales técnicas de control de acceso que utilizan los principales servidores streaming, a su vez cuenta con una interfaz que brinda mayor flexibilidad y usabilidad a los usuarios a la hora de definir y aplicar las técnicas para el control de acceso.

#### 2.3.1 Actores del sistema

El actor es aquella persona que interactúa con el sistema, es quien lleva a cabo los casos de usos. Es un rol que cumple un usuario, puede ser un ser humano, software o máquina que interactúe con el sistema. (18)

Por las características del sistema se definieron los siguientes actores:

**Usuario:** Es la persona que interactúa con la aplicación.

**Administrador:** Es un usuario autenticado que tiene un rol que lo identifica, realiza las principales funcionalidades del sistema.

#### 2.3.2 Casos de Uso del Sistema.

1. Autenticar usuario.
2. Gestionar usuarios.
3. Gestionar servidor.
4. Gestionar tipo de servidor.
5. Gestionar IP o rango de IP.
6. Gestionar grupos de usuarios.
7. Gestionar permisos.

### 2.3.3 Diagrama de Casos de Usos del Sistema.

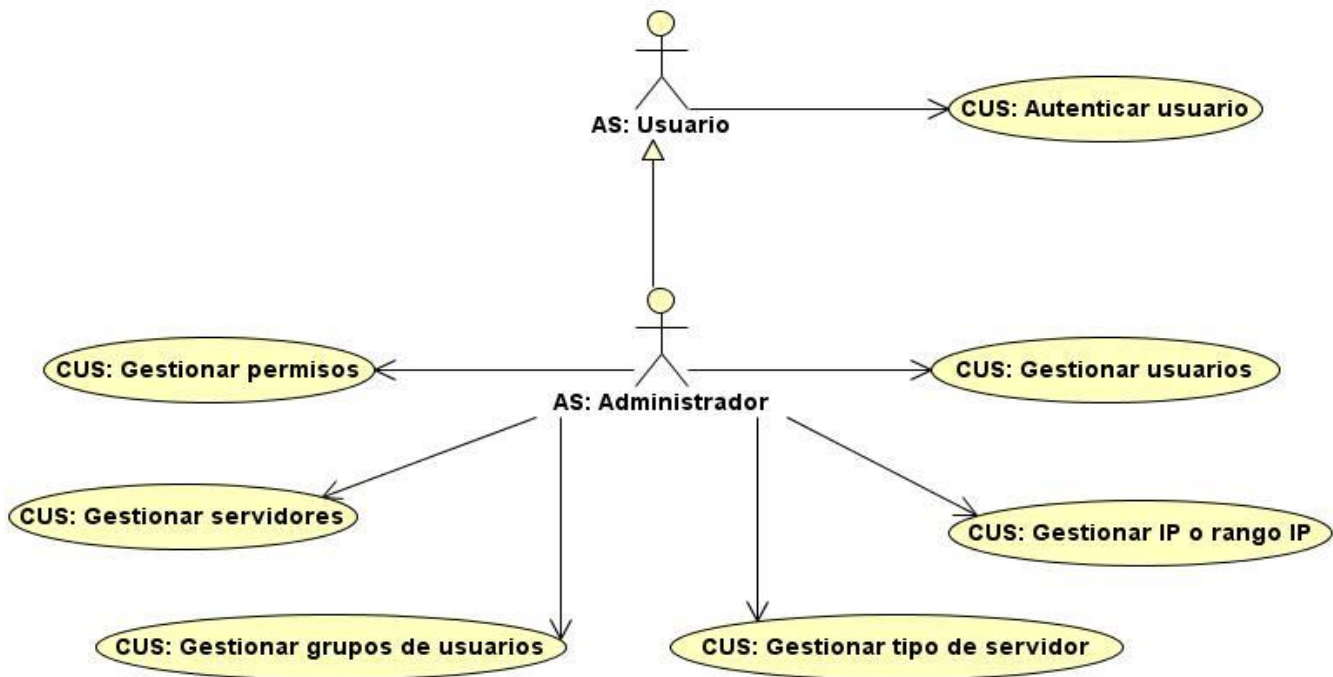



Figura 4. Diagrama de Casos de Usos del Sistema.

### 2.3.4 Descripción textual de Casos de Usos.

<b>Caso de Uso:</b>	<b>Autenticar usuario.</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario accede a la aplicación y para ello introduce su usuario y contraseña, si los datos del mismo son correctos, tendrá acceso a las funcionalidades que brinda la aplicación, finalizando así el caso de uso, en caso de no estar correctos los datos el sistema muestra un mensaje advirtiendo al usuario.
<b>Precondiciones:</b>	Debe ser administrador del sistema.
<b>Referencias</b>	RF 17
<b>Prioridad</b>	Crítico

## Capítulo 2. Características del sistema.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El caso de uso se inicia cuando el usuario accede a la aplicación.	1.1-El sistema muestra una interfaz para acceder a la aplicación.
2-El usuario introduce su nombre de usuario y contraseña y presiona el botón para entrar.	2.2-El sistema verifica que los datos sean correctos.
	2.3-El sistema le da acceso a la aplicación dependiendo del permiso del usuario y finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.2- El sistema muestra un mensaje en caso que existan campos vacíos y retorna a la página de inicio y finaliza el caso de uso.  2.2-El sistema muestra un mensaje advirtiendo que el usuario debe ser administrador, retorna a la página de inicio y finaliza el caso de uso.
Prototipo de Interfaz	
	

## Capítulo 2. Características del sistema.

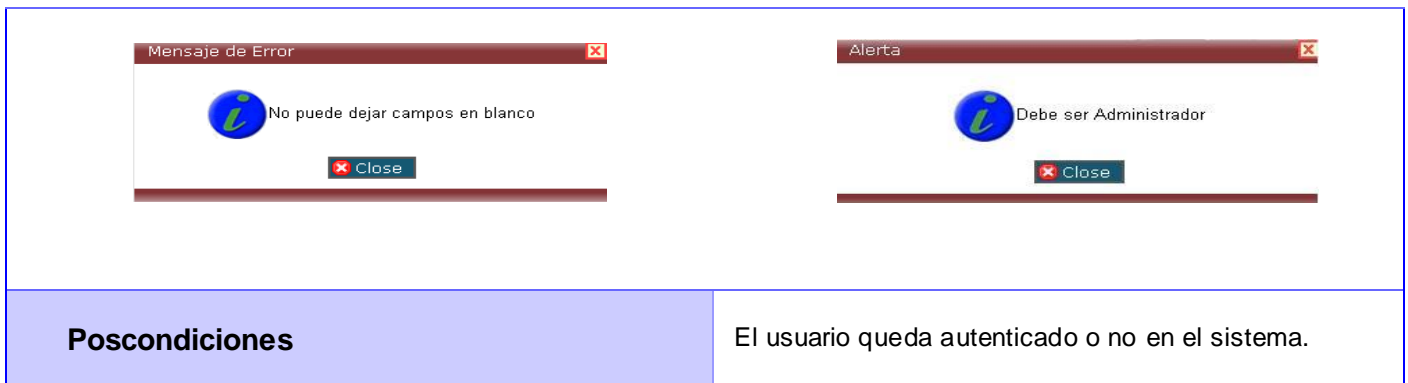


Figura 5. Descripción CU Autenticar Usuario.



<b>Caso de Uso:</b>	<b>Gestionar usuarios.</b>
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema decide gestionar usuarios y permite insertar, eliminar, modificar los datos de un usuario existente, como son el nombre, el usuario, la contraseña y seleccionar el grupo al cual va a pertenecer y finaliza el caso de uso.
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente y debe existir un grupo en el sistema.
<b>Referencias</b>	RF1, RF2, RF3.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El caso de uso comienza cuando el administrador escoge la opción Gestionar usuarios.	1.1-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar un usuario.  - Para insertar un usuario ir a la sección

## Capítulo 2. Características del sistema.


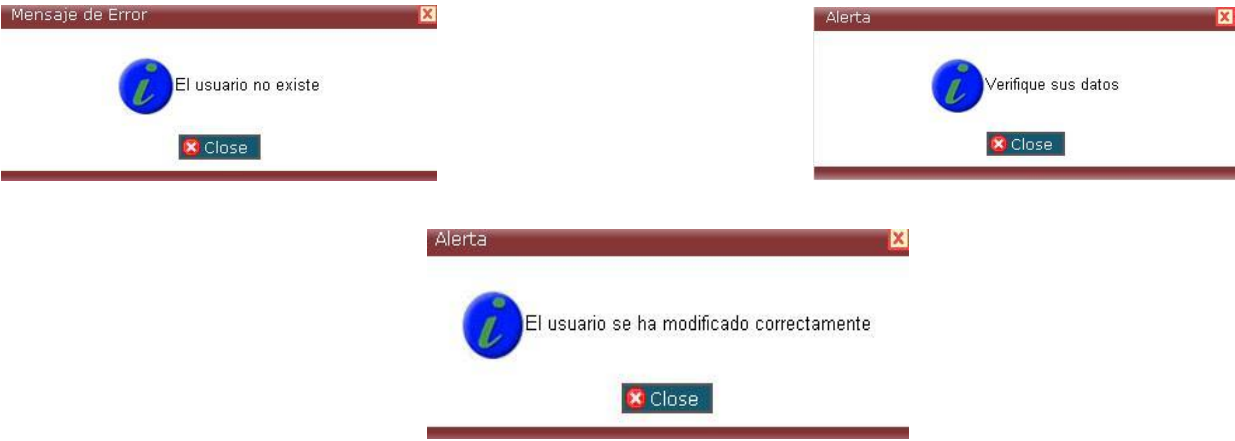
---

	<p><b>“Insertar usuario”.</b></p> <ul style="list-style-type: none"><li>- Para modificar los datos de un usuario ir a la sección <b>“Modificar usuario”</b>.</li><li>- Para eliminar un usuario ir a la sección <b>“Eliminar usuario”</b>.</li></ul>
<b>Sección “Insertar usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2-El usuario escoge la opción Insertar usuario.	2.2-El sistema muestra la interfaz para insertar los datos.
3-El administrador del sistema entra los datos del usuario para luego solicitar su adición al sistema.	3.1- Verifica que no existan campos en blanco y que los datos son correctos.  3.2-Verifica que no exista el usuario.  3.3-Guarda los datos del nuevo usuario.  3.4-Muestra mensaje indicando que el usuario se ha creado y guardado correctamente y de esta forma finaliza el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1- Muestra un mensaje de error en caso que existan campos en blanco, o que los datos sean incorrectos.  3.2-Muestra en mensaje de error advirtiendo que el usuario ya existe.

## Capítulo 2. Características del sistema.

<b>Poscondiciones</b>	Se ha creado o no, un nuevo usuario en el sistema.
<b>Prototipo de Interfaz</b>	
	
	
<b>Sección "Modificar Usuario"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2-El usuario escoge la opción Modificar usuario.	2.1-El sistema muestra una interfaz introducir usuario que se desea modificar.
3-El administrador introduce el usuario que quiere modificar.	3.1-El sistema verifica la existencia del usuario. 3.2-El sistema muestra los datos del usuario seleccionado.
4-Modifica los parámetros deseados y solicita modificar al usuario.	4.1-Verifica que los datos sean correctos.

## Capítulo 2. Características del sistema.

	4.2-Se actualizan los datos del usuario en la Base de Datos y finaliza el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1-Se emite un mensaje de error porque no existe el usuario.
	4.1-El sistema muestra un mensaje de error cuando los datos no son correctos.
<b>Poscondiciones</b>	Se ha modificado o no un usuario existente en el sistema.
<b>Prototipo de Interfaz</b>	
 <p>Nombre <input type="text" value="admin"/> Usuario <input type="text" value="admin"/> Contraseña <input type="password" value="....."/> Grupo <input type="text" value="Administrador"/> <input type="button" value="Modificar"/></p>	
 <p>Mensaje de Error: El usuario no existe. Close</p> <p>Alerta: Verifique sus datos. Close</p> <p>Alerta: El usuario se ha modificado correctamente. Close</p>	
<b>Sección "Eliminar usuario".</b>	



## Capítulo 2. Características del sistema.



Acción del Actor	Respuesta del Sistema
2-El usuario selecciona la opción eliminar usuario.	2.1-El sistema muestra una interfaz para introducir el usuario que desea eliminar.
3-El administrador selecciona introduce el usuario que desea eliminar.	3.1-El sistema verifica que el usuario existe. 3.2-Se elimina el usuario, se actualiza la Base de Datos y finaliza el CU.
<b>Flujos alternos</b>	
Acción del Actor	Respuesta del Sistema
	3.1- Se emite mensaje de error porque no existe el usuario.
<b>Poscondiciones</b>	Se ha eliminado o no un usuario existente en el sistema.
<b>Prototipo de Interfaz</b>	
	
	

Figura 6. Descripción CU Gestionar usuarios.

<b>Caso de Uso:</b>	<b>Gestionar grupos de usuarios.</b>
---------------------	--------------------------------------

## Capítulo 2. Características del sistema.

<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema decide insertar, eliminar o modificar los datos relacionados con un grupo de usuarios, como el nombre del grupo, además de seleccionar el permiso y el servidor a cual va a pertenecer dicho grupo.
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente y deben existir un servidor y un permiso creado en el sistema.
<b>Referencias</b>	RF 4, RF 5, RF 6.
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El caso de uso se inicia cuando se escoge la opción Gestionar grupos de usuarios.	1.1-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar un grupo de usuarios.  - Para insertar un grupo de usuarios ir a la sección <b>“Insertar grupo”</b> .  - Para modificar los datos de un grupo de usuarios ir a la sección <b>“Modificar grupo”</b> .  - Para eliminar un grupo de usuarios ir a la sección <b>“Eliminar grupo”</b> .

### Sección “Insertar grupo”

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

## Capítulo 2. Características del sistema.

2-El administrador del sistema selecciona Insertar grupo de usuarios.	2.1-El sistema muestra una interfaz para introducir los datos.
3-El administrador del sistema entra los datos del nuevo grupo, para luego solicitar la creación del mismo.	3.1-Verifica que los datos estén correctos. 3.2-Comprueba que el grupo no exista. 3.3-Guarda los datos del nuevo grupo. 3.4-Muestra un mensaje indicando que el grupo se ha creado y guardado correctamente y finaliza el caso de uso.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.1-El sistema muestra un mensaje cuando los datos estén incorrectos o existan campos vacíos. 3.2-Muestra un mensaje advirtiéndole que el grupo existe.
Poscondiciones	Se ha creado o no un grupo en el sistema.

### Prototipo de Interfaz

El prototipo de interfaz muestra un formulario con los siguientes elementos:

- Un campo de texto etiquetado como "Grupo".
- Un menú desplegable etiquetado como "Permiso" con la opción "Seleccionar" visible.
- Un menú desplegable etiquetado como "Servidor" con la opción "Seleccionar" visible.
- Un botón de acción etiquetado como "Insertar" con un icono de checkmark verde.

## Capítulo 2. Características del sistema.



### Sección "Modificar grupo"

Acción del Actor	Respuesta del Sistema
2-Selecciona Modificar grupo de usuarios.	2.1-Muestra una interfaz donde se puede introducir el nombre del grupo que se desea modificar.
3-El administrador introduce el grupo que quiere modificar.	3.1-El sistema verifica la existencia del grupo 3.2-El sistema muestra los datos del grupo seleccionado.
4-Modifica los parámetros deseados y solicita modificar grupo.	4.2-Se actualizan los datos del grupo y se finaliza el caso de uso.
<b>Poscondiciones</b>	Se ha modificado o no un grupo de usuarios en el sistema.

### Prototipo de Interfaz

## Capítulo 2. Características del sistema.

Grupo

Permiso Administracion Servidor Flumotion1

Modificar

**Alerta** ✕

El grupo no existe

✕ Close

**Alerta** ✕

Verifique el nombre del grupo

✕ Close

**Alerta** ✕

Se ha modificado el grupo

✕ Close

### Flujos alternos

Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe el grupo.

### Sección “Eliminar grupo”.

Acción del Actor	Respuesta del Sistema
2-El administrador selecciona Eliminar grupo de usuarios.	2.1-Se muestra una interfaz introducir el grupo que se desea eliminar.
3-Introduce el grupo que desea eliminar.	3.1-Verifica que el grupo existe.  3.2-Se elimina el grupo, se actualiza la Base de Datos y finaliza el CU.

## Capítulo 2. Características del sistema.



Flujos alternos	
Acción del Actor	Respuesta del Sistema
	3.2-Se emite mensaje de error porque no existe el grupo.
Poscondiciones	Se ha eliminado o no un grupo existente en el sistema.
Prototipo de Interfaz	
	
	

Figura 7: Descripción CU Gestionar grupo de usuarios.

<b>Caso de Uso:</b>	<b>Gestionar servidor.</b>
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema desea gestionar un servidor, el mismo puede insertar, eliminar o modificar los datos relacionados con un servidor, como son el servidor, la política por defecto, el puerto y el protocolo para el script de iptable, así como la dirección del fichero y el tipo de servidor que será.
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente y debe existir en el sistema algún tipo de servidor almacenado.

## Capítulo 2. Características del sistema.

<b>Referencias</b>	RF 10, RF 11, RF 12.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-Escoje la opción Gestionar servidor.	2-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar los datos de un servidor.  <ul style="list-style-type: none"><li>- Para insertar un servidor ir a la sección <b>“Insertar servidor”</b>.</li><li>- Para modificar los datos de un servidor ir a la sección <b>“Modificar servidor”</b>.</li><li>- Para eliminar un servidor ir a la sección <b>“Eliminar servidor”</b>.</li></ul>
<b>Sección “Insertar servidor”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2-El administrador selecciona Insertar servidor.	2.1-Se muestra una interfaz donde el administrador debe entrar los datos del servidor.
3-El administrador del sistema entra los datos del servidor para luego solicitar su adición.	3.1-Verifica que los datos sean correctos.  3.2-Verifica que no exista.  3.3-Guarda los datos del servidor.  3.4-Muestra mensaje indicando que el servidor ha sido creado y se ha guardado correctamente,

## Capítulo 2. Características del sistema.

de esta forma finaliza el caso de uso.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.1-Muestra un mensaje cuando los datos son incorrectos o existen campos en blanco.  3.2-Muestra mensaje advirtiendo el servidor existe.
<b>Poscondiciones</b>	Se ha creado o no un nuevo servidor en el sistema.

### Prototipo de Interfaz

Formulario de modificación de servidor con los siguientes campos:

- Servidor:
- Protocolo:
- Puerto:
- Dirección de fichero:
- Tipo:
- Política por defecto:  ACEPTAR  DENEGAR

Botón:

Mensaje de Error



Debe entrar todos sus datos

Close

Alerta



Verifique sus datos

Close

Mensaje de Error



El servidor ya existe

Close

Alerta



Inserto correctamente en la BD

Close

### Sección "Modificar servidor"



## Capítulo 2. Características del sistema.

Acción del Actor	Respuesta del Sistema
2-El administrador selecciona Modificar servidor.	2.1-Muestra una interfaz para seleccionar el nombre del servidor que desea eliminar.
3-El administrador selecciona el servidor que quiere modificar.	3.1-El sistema verifica que el servidor existe. 3.2-El sistema muestra los datos del servidor.
4-Modifica los parámetros deseados y solicita modificar servidor.	4.1-Verifica que esté correcta la información. 4.2-Se actualizan los datos del servidor y finaliza el caso de uso.

### Flujos alternos

Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe el servidor. 4.1-Muestra un mensaje porque los datos no son correctos.
Poscondiciones	Se han modificado o no los datos de un servidor existente en el sistema.

### Prototipo de Interfaz

---

<b>Servidor</b>	<input type="text" value="1"/>
<b>Protocolo</b>	<input type="text" value="tcp"/>
<b>Puerto</b>	<input type="text" value="5800"/>
<b>Dirección de fichero</b>	<input type="text" value="home/adaily/Escritorio/"/>
<b>Tipo</b>	<input type="text" value="apache"/> ▼
ACEPTAR <input checked="" type="radio"/>	DENEGAR <input type="radio"/>
<input type="button" value="Modificar"/>	

---

## Capítulo 2. Características del sistema.



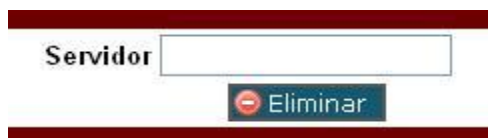
### Sección "Eliminar servidor".

Acción del Actor	Respuesta del Sistema
2-Selecciona Eliminar servidor.	2.1-Muestra muestra una interfaz para introducir el servidor que se desea eliminar.
3-Introduce el servidor.	3.1-Verifica que el servidor existe. 3.2-Se elimina el servidor y finaliza el CU.

### Flujos alternos

Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe el servidor.
Poscondiciones	Se ha eliminado o no un servidor existente en el sistema.

### Prototipo de Interfaz



## Capítulo 2. Características del sistema.



Figura 8: Descripción CU Gestionar servidor

<b>Caso de Uso:</b>	<b>Gestionar permisos.</b>
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema desea gestionar los permisos que se le otorgan a los grupos de usuarios, el mismo puede insertar, eliminar o modificar un permiso y de esta forma finaliza el caso de uso.
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente.
<b>Referencias</b>	RF 7, RF 8, RF 9.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-Escoje la opción Gestionar permisos.	2-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar un permiso.  - Para insertar un permiso ir a la sección <b>“Insertar permiso”</b> .  - Para modificar un permiso ir a la sección <b>“Modificar permiso”</b> .

## Capítulo 2. Características del sistema.

- Para eliminar un permiso ir a la sección “Eliminar permiso”.

### Sección “Insertar permiso”

Acción del Actor	Respuesta del Sistema
2-El caso de uso se inicia cuando el administrador selecciona Insertar permiso.	2.1-Se muestra una interfaz donde el administrador debe entrar los datos del permiso.
3-El administrador del sistema entra los datos del permiso, para luego solicitar su adición al mismo.	3.1-Comprueba que no exista el permiso. 3.2-Guarda los datos del nuevo permiso. 3.3-Muestra mensaje indicando que el permiso ha sido insertado y se ha guardado correctamente, finalizando de esta forma el caso de uso.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.1-Muestra mensaje advirtiendo que el permiso existe.
Poscondiciones	Se ha creado o no un nuevo permiso en el sistema.

### Prototipo de Interfaz

Permiso

## Capítulo 2. Características del sistema.



### Sección "Modificar permiso"

Acción del Actor	Respuesta del Sistema
2-Selecciona Modificar permiso.	2.1-Muestra una interfaz para introducir el permiso a modificar.
3-El administrador introduce el permiso que desea modificar.	3.1-El sistema verifica que el permiso existe 3.2-El sistema muestra los datos del permiso seleccionado.
4-Modifica los parámetros deseados y solicita modificar.	4.2-Se actualizan los datos y se finaliza el caso de uso.

### Flujos alternos

Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe el permiso.
Poscondiciones	Se ha modificado o no un permiso existente en el sistema.

### Prototipo de Interfaz

## Capítulo 2. Características del sistema.



### Sección "Eliminar permiso".

Acción del Actor	Respuesta del Sistema
2-Selecciona Eliminar permiso.	2.1-Muestra una interfaz para introducir el permiso.
3-Introduce el permiso que desea eliminar.	3.1-Verifica que el permiso existe.  3.2-Se elimina el permiso, se actualiza la base de datos y finaliza el CU.

### Flujos alternos

Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe el permiso.
Poscondiciones	Se ha eliminado o no un permiso existente en el sistema.

### Prototipo de Interfaz

## Capítulo 2. Características del sistema.

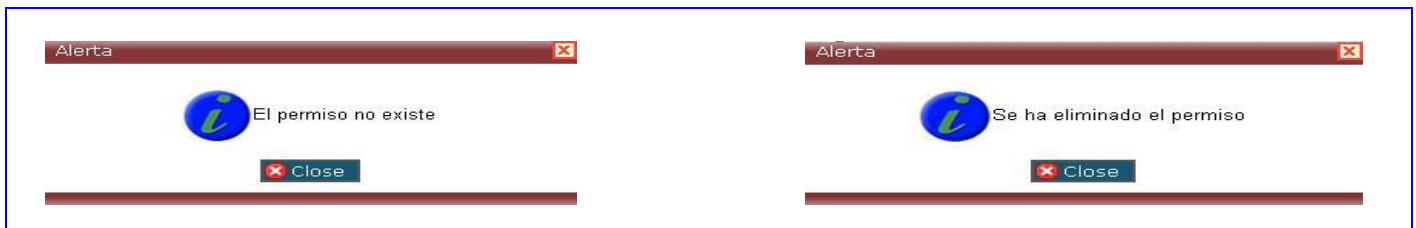


Figura 9: Descripción CU Gestionar permiso.

<b>Caso de Uso:</b>	<b>Gestionar tipo de servidor.</b>	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema desea gestionar un tipo de servidor, puede insertar, eliminar o modificar un tipo de servidor en el sistema.	
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente.	
<b>Referencias</b>	RF 15, RF 16.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1-El caso de uso se inicia cuando el administrador escoge la opción Gestionar tipo de servidor.	2-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar los datos de un tipo de servidor. <ul style="list-style-type: none"> <li>- Para insertar un servidor ir a la sección <b>“Insertar tipo servidor”</b>.</li> <li>- Para eliminar un servidor ir a la sección <b>“Eliminar tipo de servidor”</b>.</li> </ul>	
<b>Sección “Insertar tipo de servidor”</b>		

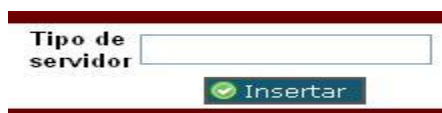
## Capítulo 2. Características del sistema.

Acción del Actor	Respuesta del Sistema
2-El administrador selecciona Insertar tipo servidor.	2.1-Se muestra una interfaz donde el administrador debe entrar el tipo de servidor.
3-El administrador del sistema entra el tipo de servidor.	3.1-Verifica que no exista el tipo de servidor. 3.2-Guarda el tipo de servidor. 3.3-Muestra mensaje indicando que ha sido creado un nuevo tipo de servidor y de esta forma finaliza el caso de uso.

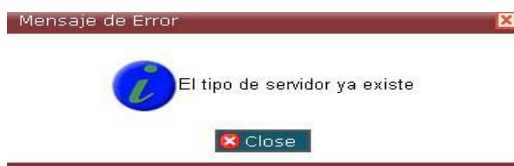
### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.1-Muestra mensaje advirtiendo el tipo de servidor ya existe.
Poscondiciones	Se ha creado o no un nuevo tipo de servidor en el sistema.

### Prototipo de Interfaz



Tipo de servidor



### Sección "Eliminar tipo servidor".



## Capítulo 2. Características del sistema.


Acción del Actor	Respuesta del Sistema
2-Selecciona Eliminar tipo servidor.	2.1-Muestra muestra una interfaz para introducir el tipo de servidor que se desea eliminar.
3-Introduce el tipo.	3.1-Verifica que existe. 3.2-Se elimina el tipo de servidor y finaliza el CU.
<b>Flujos alternos</b>	
Acción del Actor	Respuesta del Sistema
	3.1-Se emite mensaje de error porque no existe ese tipo de servidor.
Poscondiciones	Se ha eliminado o no un tipo de servidor existente en el sistema.
<b>Prototipo de Interfaz</b>	
	
	

Figura 10: Descripción CU Gestionar tipo de servidor.

<b>Caso de Uso:</b>	<b>Gestionar IP o rango IP.</b>
<b>Actores:</b>	Administrador

## Capítulo 2. Características del sistema.

<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema desea gestionar un IP o rango IP, el mismo puede insertar, eliminar o modificar un IP o rango IP con el objetivo de crear un script de IPtable, además de seleccionar el servidor al que pertenece dicho IP o rango IP.
<b>Precondiciones:</b>	El administrador debe haberse autenticado previamente y en el sistema debe haber almacenado un servidor.
<b>Referencias</b>	RF 13, RF 14.
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1-El caso de uso se inicia cuando el administrador escoge la opción Gestionar un IP o rango IP.	2-El sistema muestra la interfaz donde el administrador puede insertar, eliminar o modificar un IP o rango IP.  - Para insertar un servidor ir a la sección <b>“Insertar IP o rango IP”</b> .  - Para eliminar un servidor ir a la sección <b>“Eliminar IP o rango IP”</b> .

### Sección “Insertar IP o rango IP”

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2-El administrador selecciona Insertar IP o rango IP.	2.1-Se muestra una interfaz donde el administrador debe entrar el IP o rango IP.
3-El administrador del sistema entra el IP o rango IP.	3.1-Verifica que el IP o rango IP esté correcto.

## Capítulo 2. Características del sistema.

- 3.2-Verifica que no exista el IP o rango IP.
- 3.3-Guarda el IP o rango IP genera el script de Iptable.
- 3.4-Muestra mensaje indicando que ha sido creado IP o rango IP y de esta forma finaliza el caso de uso.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	<ul style="list-style-type: none"><li>3.1-El sistema muestra un mensaje advirtiendo que el IP o rango IP no son correctos.</li><li>3.2-Muestra mensaje advirtiendo que el IP o rango IP ya existe.</li></ul>
Poscondiciones	Se ha creado o no un nuevo IP o rango IP en el sistema.

### Prototipo de Interfaz

IP o Rango IP   
Ejemplo IP: 192.168.1.0  
Rango IP: 192.168.1.0/24


Servidor

Alerta

 El IP o Rango IP no es válido

Mensaje de Error


 El tipo IP o rango IP ya existe




**Sección “Eliminar IP o rango IP”.**

Acción del Actor	Respuesta del Sistema
2-Selecciona Eliminar IP o rango IP.	2.1-Muestra muestra una interfaz para seleccionar el IP o rango IP que se desea eliminar.
3-Selecciona el IP o rango IP.	3.2-Se elimina el IP o rango IP y finaliza el CU.
<b>Poscondiciones</b>	Se ha eliminado o no un IP o rango IP existente en el sistema.

**Prototipo de Interfaz**





**Figura 11: Descripción CU Gestionar IP o rango IP.**

### 2.4 Conclusiones.

En este capítulo se realizó un análisis del entorno que enmarca el sistema donde se confeccionó un modelo de dominio que permite conocer los objetos reales relacionados con el entorno del sistema y las relaciones entre ellos, así como los conceptos fundamentales del entorno. Además se delimitaron

## *Capítulo 2. Características del sistema.*

---

los requisitos funcionales y no funcionales del sistema, los cuales brindan una visión más clara sobre las características que debe cumplir el sistema propuesto, así como sus restricciones.

### **Capítulo 3. Análisis y diseño del sistema.**

#### **Introducción**

Este capítulo se centra en el análisis y diseño del sistema, los cuales constituyen uno de los mecanismos más importantes dentro del proceso unificado de desarrollo de software. El modelo de análisis ofrece una especificación más precisa de los requisitos, los estructura de modo que facilita su preparación, comprensión, modificación y en general su mantenimiento. Este es considerado como una primera aproximación al modelo del diseño.

#### **3.1 Modelo de Análisis**

El modelado de análisis según Roger Pressman “Es la primera aproximación técnica de un sistema”. El análisis permite estructurar el sistema además de su arquitectura, no es más que una aproximación al modelo del diseño.

Durante el análisis se evalúan los requerimientos descritos en el proceso de captura de requisitos, refinándolos y estructurándolos, con el objetivo de obtener una mejor comprensión.

##### **3.1.1 Diagrama de clases del análisis.**

Los diagramas de clases se utilizan para mostrar la estructura estática del sistema modelado. Pueden contener clases, interfaces, paquetes, relaciones e incluso instancias, como objetos o enlaces. Los diagramas de clases son una potente herramienta de diseño, ayudando a los desarrolladores a planificar, establecer la arquitectura y estructura del sistema.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP clasifica a las clases en:

- Clase Interfaz: «boundary»: Indica que la clase constituye la frontera o interfaz entre el sistema modelado y el exterior. Se aplica a formularios, informes, interfaces y similares
- Clase Control: «control»: Indica que la clase se dedica a labores de coordinación. Estas clases son fácilmente reconocibles porque de ellas suelen partir muchos mensajes a otras clases.
- Clase Entidad: «entity»: Contienen la información significativa del sistema (entidades) y generalmente son guardadas en almacenamiento externo, son definidas como clases persistentes.

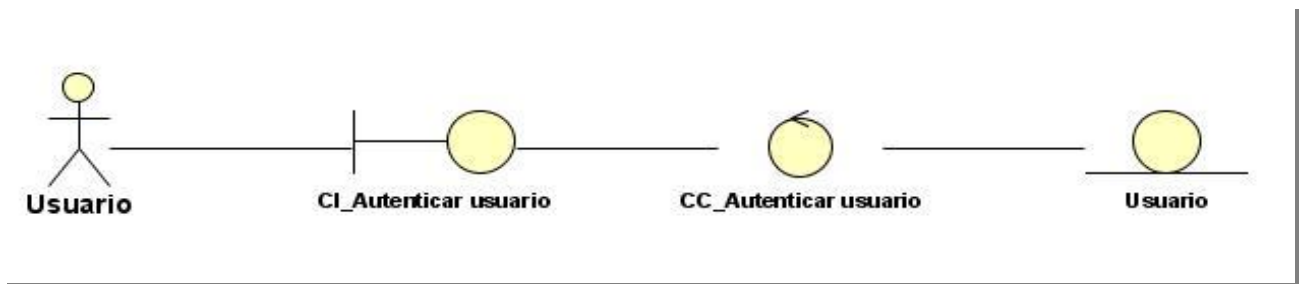


Figura 12: DCA Autenticar usuario.

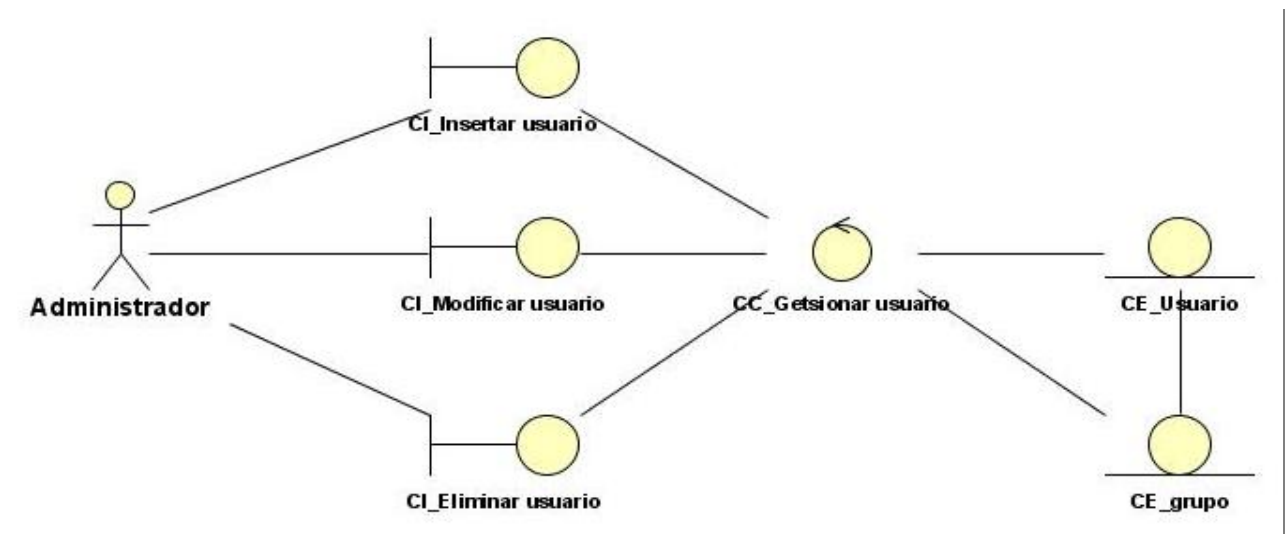


Figura 13: DCA Gestionar usuario.

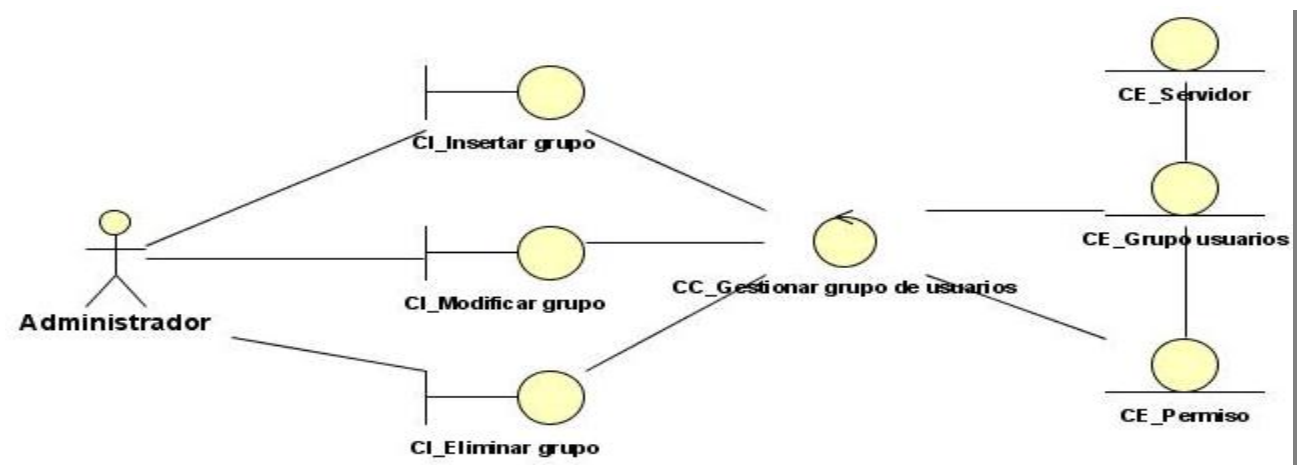


Figura 14. DCA Gestionar grupo de usuarios.

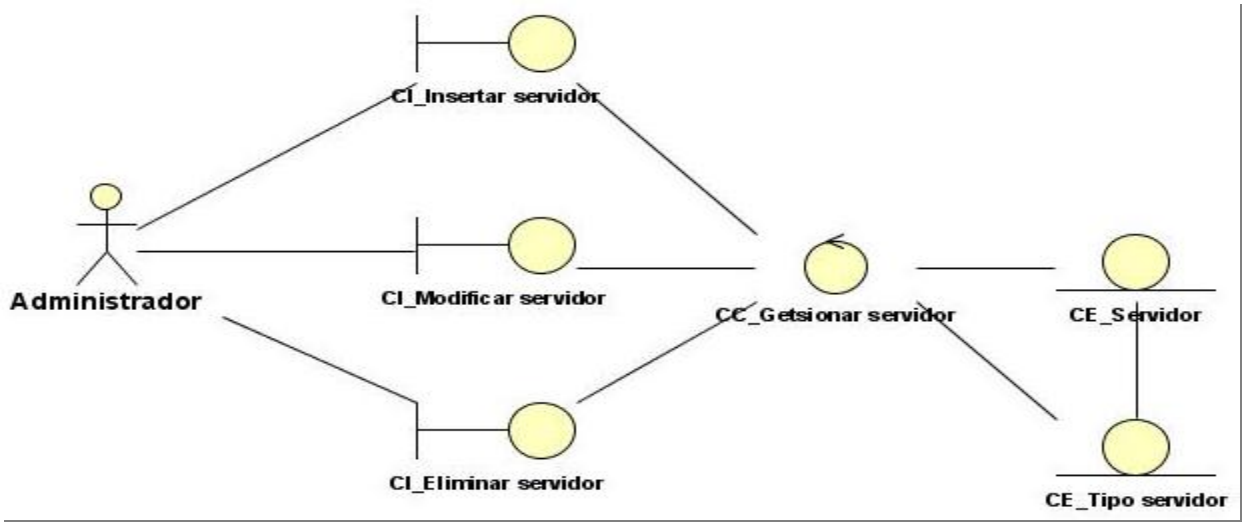


Figura 14: DCA Gestionar servidor.

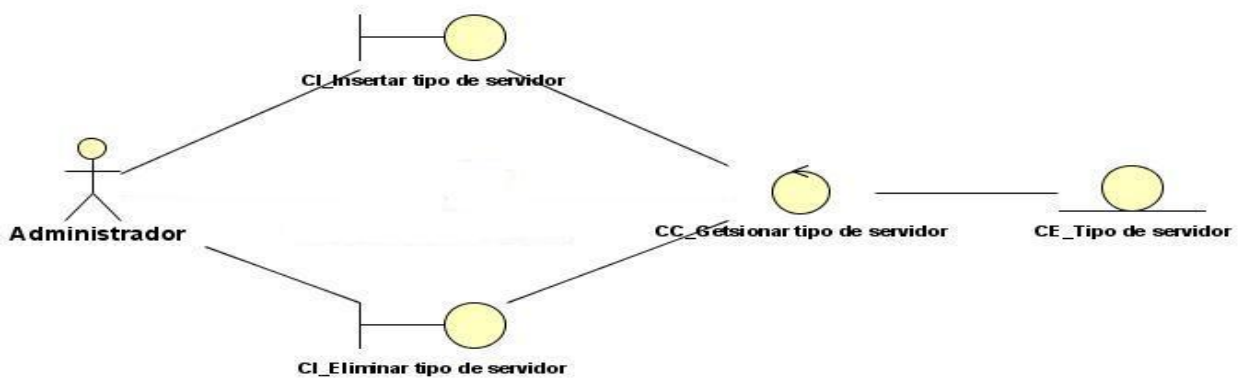


Figura 15: DCA Gestionar tipo de servidor.

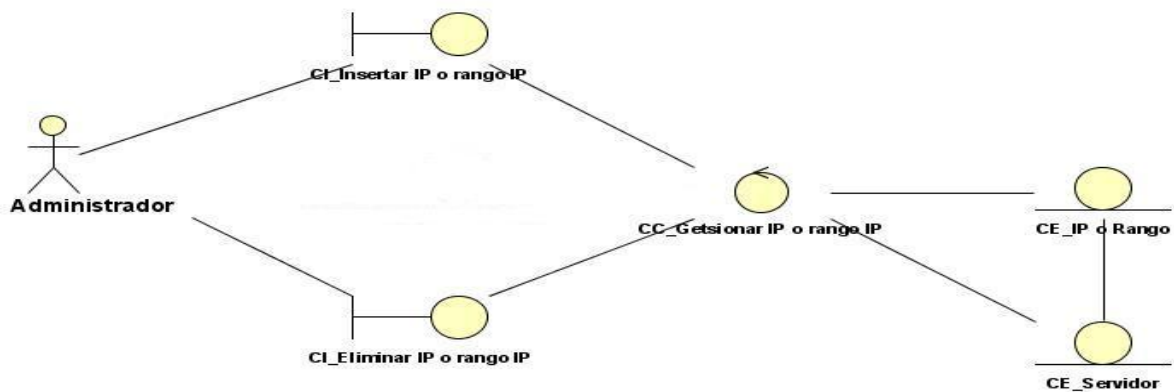


Figura 16: DCA Gestionar IP o rango IP.



### **3.2 Modelo de Diseño.**

Roger Pressman expresó en su libro “**Ingeniería de Software: Un Enfoque práctico**” que el diseño de software es un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software.

El diseño intenta preservar la estructura definida en el modelo de análisis. Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación.

La esencia de esta fase es la elaboración de diagramas de interacción, los cuales describen gráficamente el comportamiento del sistema, es decir, muestra el flujo de control a través de muchos objetos. Los objetos interactúan para realizar colectivamente los servicios ofrecidos por las aplicaciones.

#### **3.2.1 Diagramas de clases del diseño.**

Los diagramas de clases del diseño se modelaron a partir de varias clases de UML las cuales se utilizan para el modelado de aplicaciones Web, las clases “**Server Page**” representarían el código del servidor, las clases “**Client Page**” representarían el código puramente cliente, y una clase, “**Form**” representaría los formularios presentes en el código cliente.

A continuación se muestran los diagramas de clases del diseño los cuales se modelaron separados por CU.

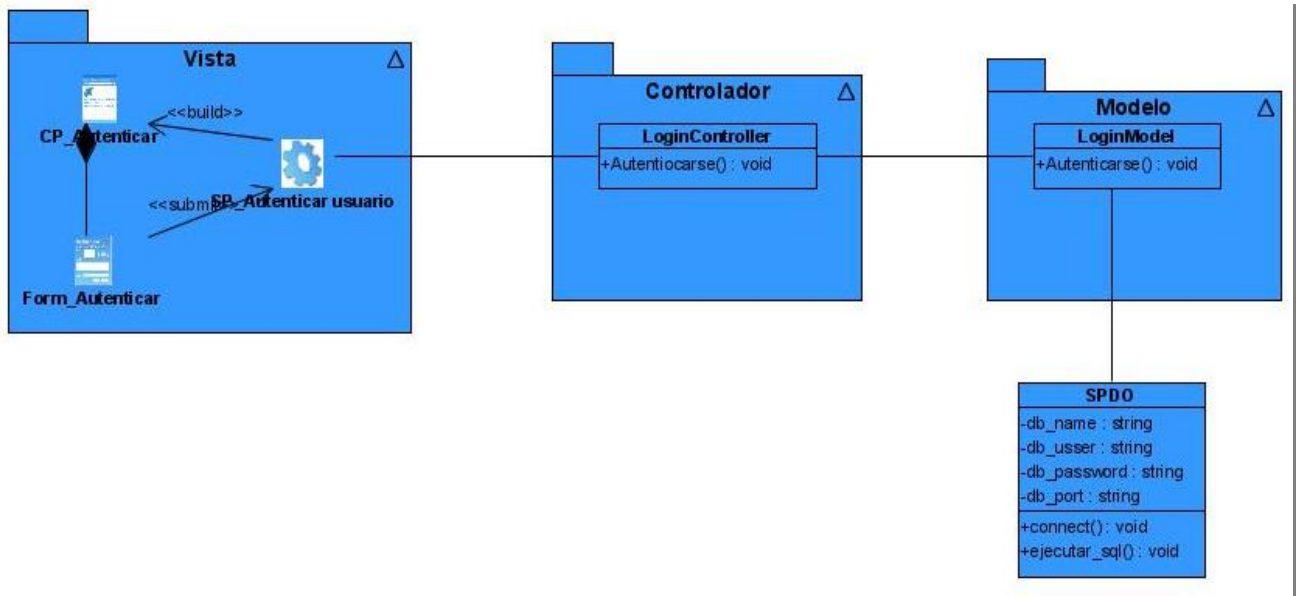


Figura 17: DCD Autenticar usuario.

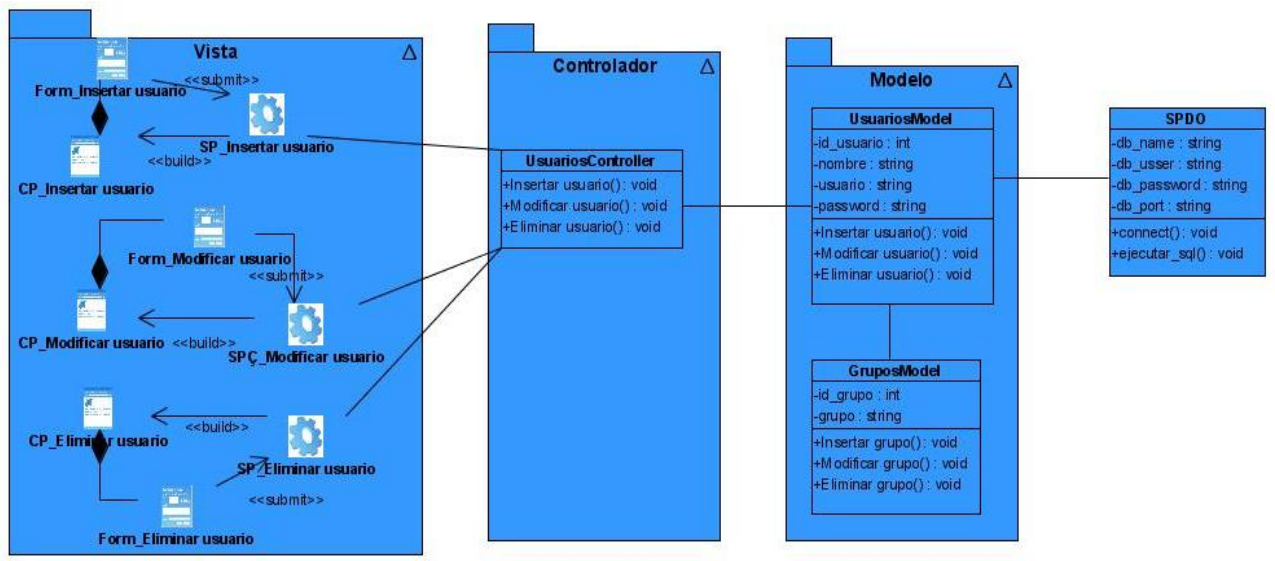


Figura 18: DCD Gestionar usuario.

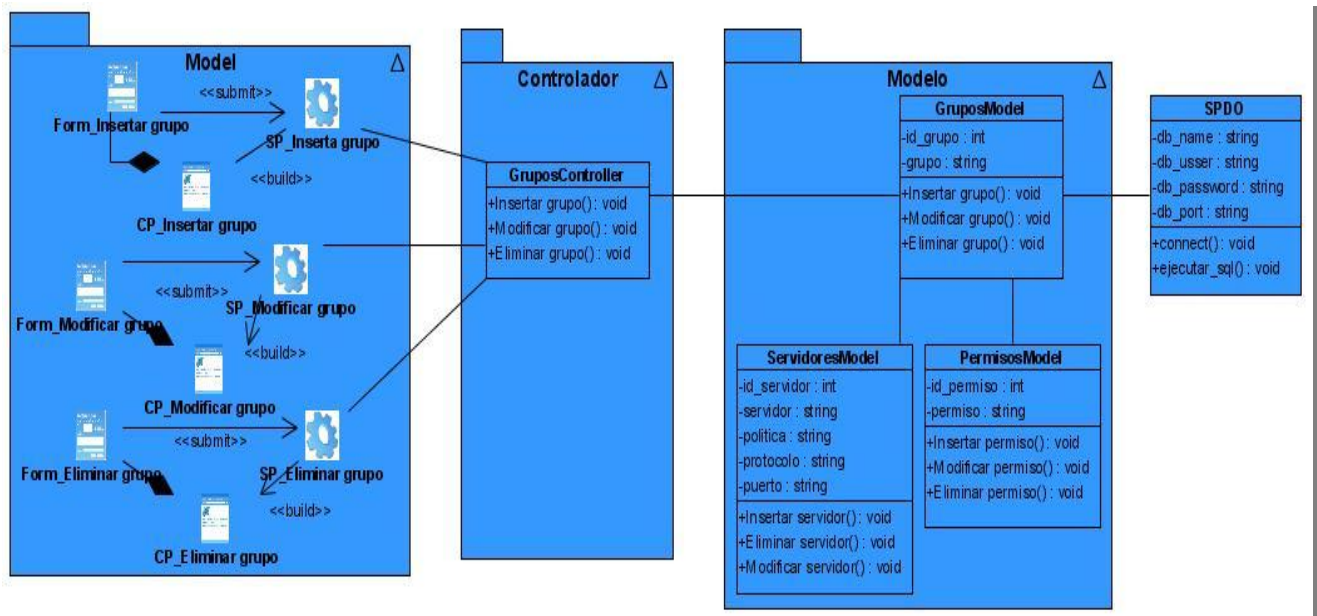


Figura 19: DCD Gestionar grupo de usuario.

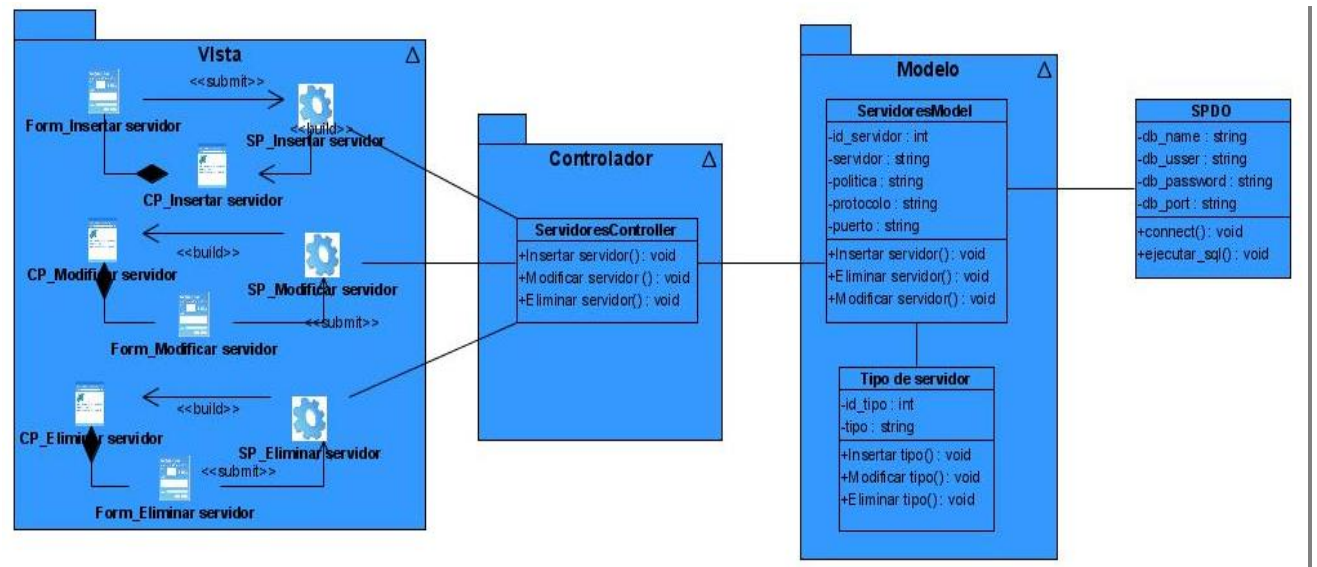


Figura 20: DCD Gestionar servidor.

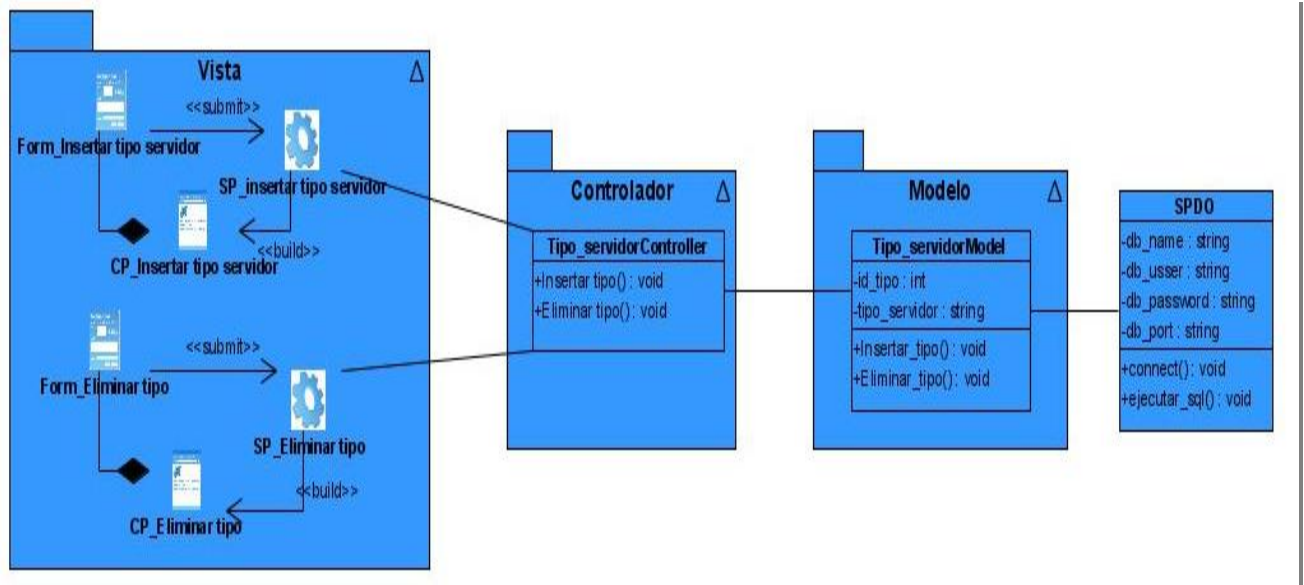


Figura 21: DCD Gestionar tipo de servidor.

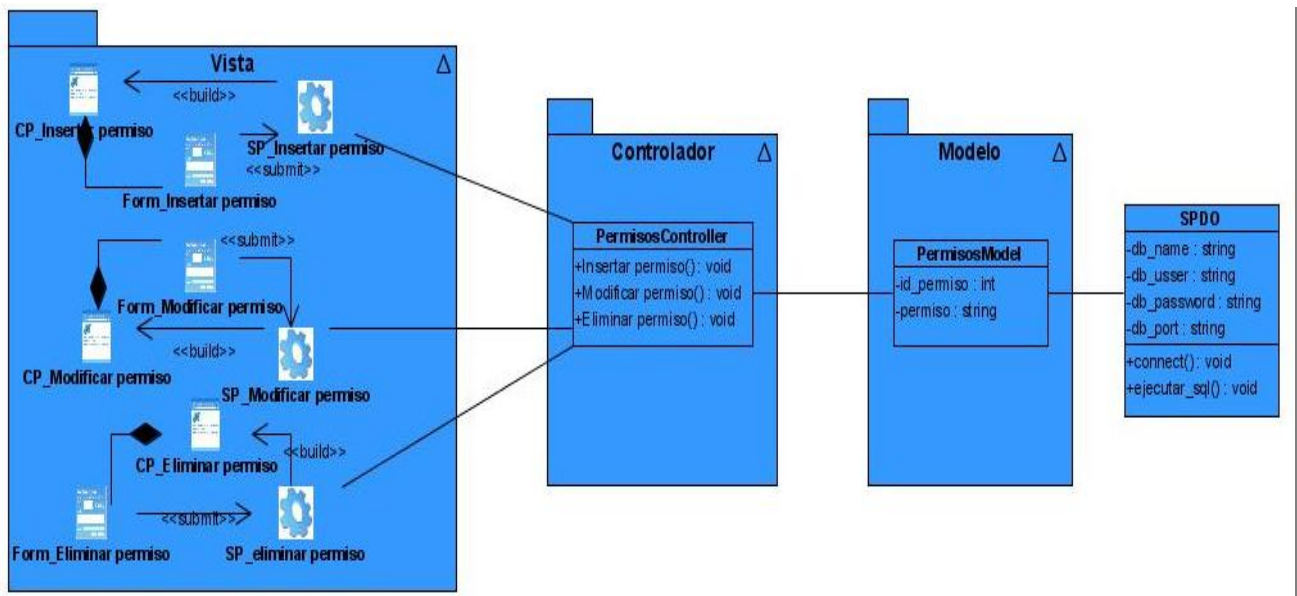


Figura 22: DCD Gestionar permiso.

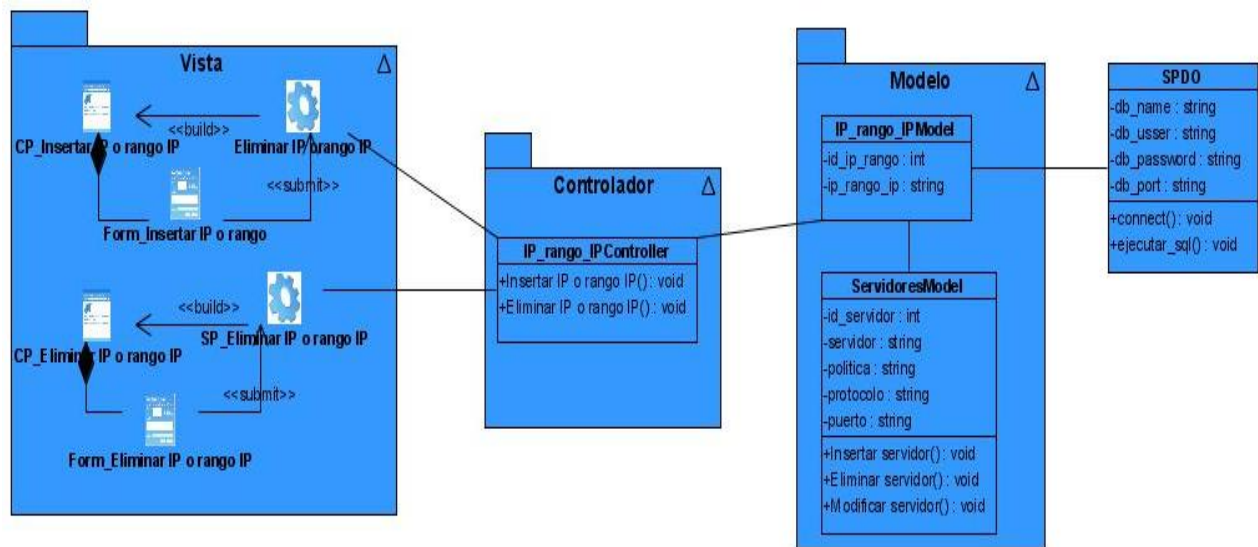


Figura 23: DCD Gestionar IP o rango IP.

### 3.3 Patrones utilizados.

El uso de patrones constituye una base en la búsqueda de soluciones a problemas que surgen durante el desarrollo de software.

Según el arquitecto Christopher Alexander<sup>2</sup>, “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”.

#### 3.3.1 Patrón Modelo-Vista-Controlador.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Modelo:** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

**Vista:** Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

<sup>2</sup> Reconocido arquitecto que creó y validó junto a Sarah Ishikawa y Murray Silverstein el término Lenguaje de Patrón.

**Controlador:** Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. En una aplicación web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista para la interacción con los usuarios.

Este patrón fue utilizado ya que permite una separación de conceptos, de forma tal que el desarrollo de la aplicación esté estructurado de una mejor forma, también facilita la programación de manera paralela e independiente en las diferentes capas, además de brindarle una mayor escalabilidad. Permite crear nuevos tipos de datos si la aplicación lo requiere pues el funcionamiento de las otras capas es independiente y facilita el tratamiento de errores.

### 3.3.2 Patrones GRASP<sup>3</sup>.

Son patrones que resaltan la importancia de captar principios para la programación, a la hora de diseñar eficazmente un software orientado a objetos. Los patrones GRASP describen entre otros elementos la asignación de responsabilidades a objetos, expresados en forma de patrones.

Entre los patrones GRASP utilizados están:

- **Creador:** este patrón asigna responsabilidades relacionadas con la creación de objetos y en la solución propuesta se utiliza al asignarle la responsabilidad a las controladoras de crear un objeto de las clases modelos para el posterior acceso a las funciones implementadas en el modelo.
- **Bajo acoplamiento,** permite que el diseño de clases sea más independientes, lo que posibilita reducir el impacto que puedan tener los cambios así como la reutilización. Este elemento se observa cuando se le otorga responsabilidades relacionadas con el control del flujo de eventos del sistema a clases específicas como por ejemplo las validaciones.
- **Controlador,** es un patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, de tal forma es la que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en

---

<sup>3</sup> GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades).

el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

### 3.3.3 Patrones GOF.

El libro **Design Patterns** (Patrones de diseño) fue escrito por el grupo Gang of Tour (GOF) o también conocido como “pandilla de los cuatros” en este libro se recogen 23 patrones de diseño comunes, este grupo estaba compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlisodes. Estos patrones se dividen en 3 grupos, patrones creacionales, estructurales y estructurales.

Entre los patrones GOF utilizados están:

- **Singleton** (Solitario), en el diseño de clases es necesario aplicar la solución del patrón que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases, un ejemplo se puede observar en las clases controladoras en las cuales para acceder a los métodos del modelo se crea una sola instancia de estas clases.
- **Fachada**, es un patrón de diseño de tipo estructural que consiste en crear una interfaz que facilite el acceso a un conjunto de interfaces o funcionalidades en la aplicación que se propone se observa en el menú de acceso.

## 3.4 Diseño de la Base de datos.

### 3.4.1 Diagrama de clases persistentes.

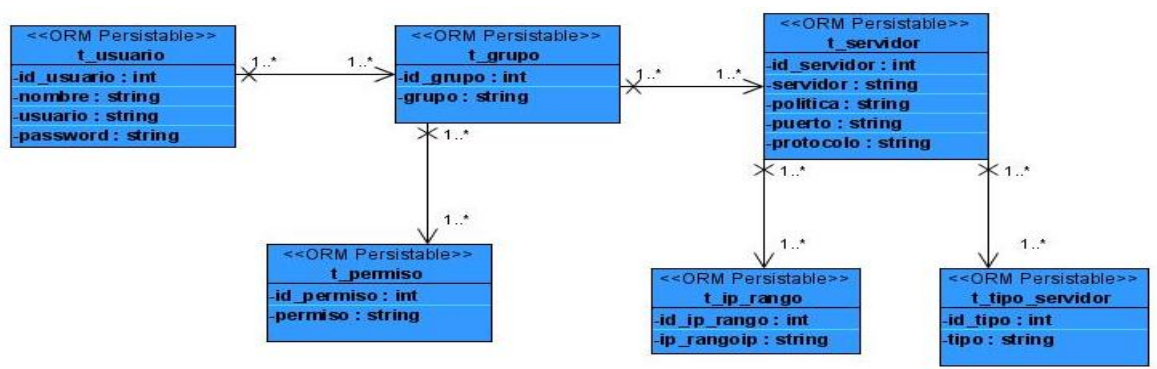


Figura 31: Diagrama de clases persistentes.

## 3.4.2 Diagrama entidad relación.

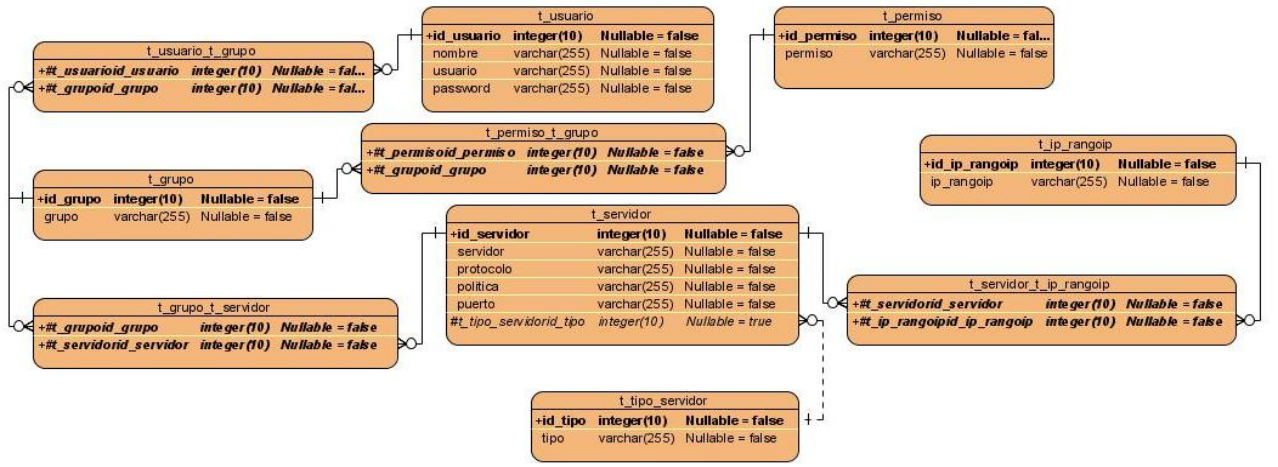


Figura 32: Diagrama Entidad Relación.

## 3.5 Conclusiones.

Con el desarrollo del análisis y diseño de la propuesta de solución se sentaron las bases para el correcto modelado de los artefactos que genera este flujo de trabajo en términos de diagramas, los que permiten lograr un mayor entendimiento sobre la propuesta del componente para la gestión del control de acceso a servidores streaming libres. Con la realización del diagrama de clases persistentes se logra una representación física de la información necesaria para la realización del sistema propuesto.



## Capítulo 4. Implementación y prueba del sistema.

### Introducción

En este capítulo se muestra el flujo de trabajo de implementación, el cual forma parte de la fase de construcción, durante este flujo se describe como los elementos del modelo de diseño se implementan en términos de componentes y cómo estos se organizan en el modelo de despliegue.

#### 4.1 Modelo de despliegue.

El modelo de despliegue describe la distribución física de los componentes lógicos desarrollados, así como las conexiones entre ellos. En el mismo están representados dos nodos, uno es la PC cliente que serán aquellas pc de los usuarios los cuales podrán acceder mediante el protocolo http al servidor donde estarán las siguientes aplicaciones: y el segundo es el servidor web, sobre el cual estará ejecutándose el componente para el control de acceso, el servidor streaming donde estarán almacenadas las medias y el servidor de base datos.

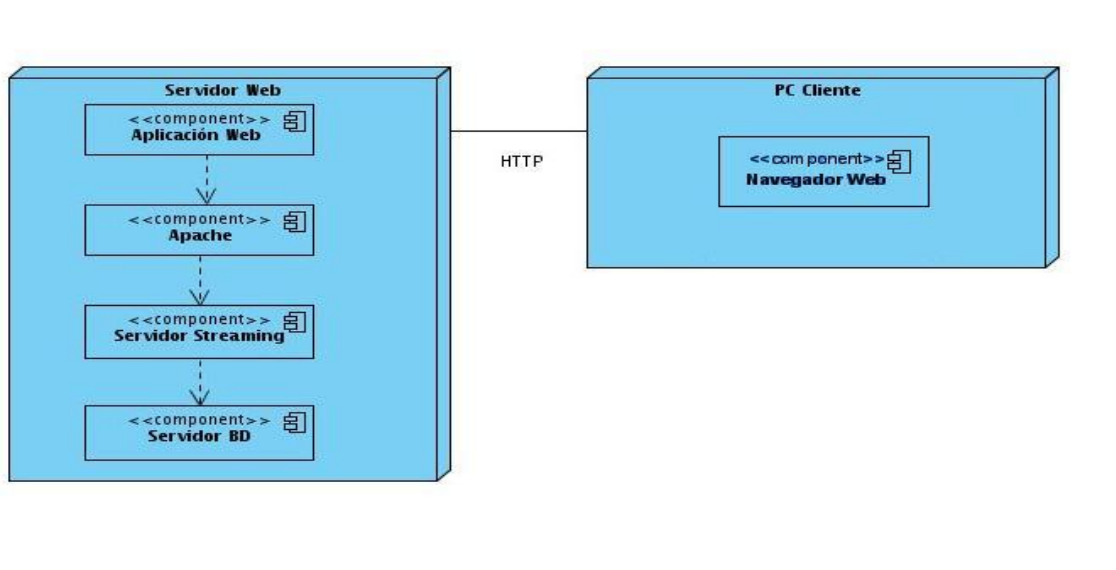


Figura 33: Modelo de despliegue.

#### 4.2 Modelo de Implementación.

El modelo de implementación describe cómo se implementan los elementos del modelo de diseño en términos de componentes. El diagrama de componentes, muestra la dependencia entre los distintos componentes de software basados en los casos de usos planteado anteriormente.

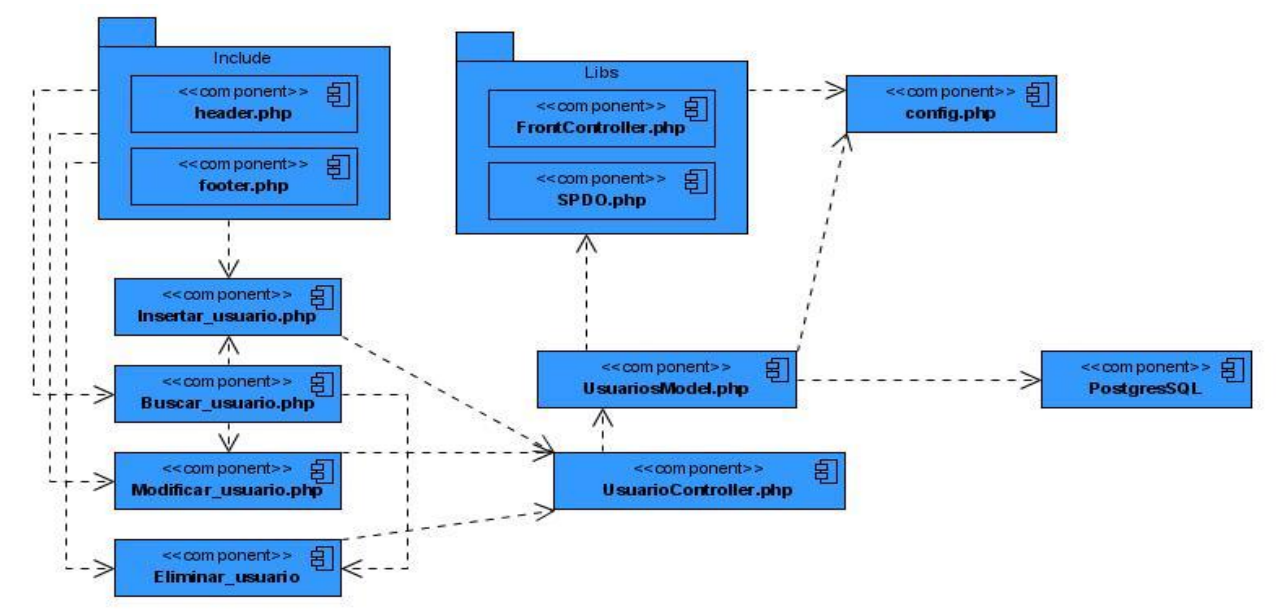


Figura 34: Diagrama de componente CU Gestionar usuario

### 4.3 Prueba

Con el objetivo de conocer el funcionamiento de la aplicación, verificar que las operaciones internas se ajustan a las especificaciones planteadas y comprobar de forma adecuada los componentes internos, se realizaron pruebas de caja negra. Esta prueba se realiza sobre la interfaz del sistema para comprobar las funcionalidades del mismo y verificar sus entradas y respuestas. Este proceso de pruebas está basado en la descripción de los diferentes casos de uso.

#### Descripción General del caso de uso Gestionar Usuario.

Comienza cuando el administrador accede al sistema para gestionar usuarios.

#### Condiciones de Ejecución.

Se debe haber autenticado como administrador.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Registrar usuario.	EC 1.1: Registrar usuario con éxito.	El Administrador accede a la funcionalidad Registrar usuario y el sistema muestra una interfaz para introducir los datos del nuevo usuario que se desea registrar. Cuando el administrador pulsa en el botón insertar quedan almacenados los datos del nuevo

## *Capítulo 4. Implementación y prueba del sistema.*

		usuario.
	EC 1.2: Registrar usuario falla.	El sistema emite un mensaje de error cuando existen campos en blanco, en que caso de que los datos estén incorrectos, en blanco o advirtiéndolo que el usuario ya existe.
SC 2: Modificar usuario.	EC 2.1: Modificar usuario con éxito.	El Administrador accede a la funcionalidad Modificar usuario y el sistema muestra una interfaz, donde debe introducir el nombre del usuario que desea modificar. Se muestra una interfaz donde se muestran los datos del usuario. Cuando el administrador modifica los datos que desea pulsa en el botón modificar y de esta forma quedan almacenados los nuevos datos del usuario.
	EC 2.2: Modificar usuario falla.	El sistema emite un mensaje de error cuando existen campos en blanco, cuando los datos estén incorrectos o cuando el usuario no existe.
SC 3: Eliminar usuario.	EC 3.1: Eliminar usuario con éxito.	El Administrador accede a la funcionalidad Eliminar usuario y el sistema muestra una interfaz para introducir el usuario que desea eliminar. Cuando el administrador pulsa en el botón eliminar se eliminan el usuario.
	EC 3.2: Eliminar usuario falla.	El sistema emite un mensaje de error cuando el campo está en blanco o advirtiéndolo que el usuario no existe.

### Descripción de las variables CU. Gestionar usuario.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Este campo solo letras, refiriéndose al nombre del
2	Usuario	Campo de texto	No	Este campo permite una cadena de caracteres, refiriéndose al nombre de

## *Capítulo 4. Implementación y prueba del sistema.*

3	Contraseña	Campo de texto	No	Este campo permite una cadena de caracteres, refiriéndose a la contraseña
4	Grupo	Lista desplegable	No	Permite seleccionar el grupo de usuarios.
5	Insertar	Botón	No	Permite insertar un usuario.
6	Buscar	Botón	No	Permite buscar un usuario.
7	Eliminar	Botón	No	Permite eliminar un usuario existente.

Matriz de datos CU. Gestionar usuarios

Escenario	Variable 1 Nombre	Variable 2 Usuario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Insertar usuario con éxito.	(Adaily Hernandez)V	(ady) V	El sistema muestra una interfaz donde se introducen los datos relacionados con el usuario que se desea insertar	Satisfactorio	<p>Una vez que el administrador se ha identificado como tal en el sistema.</p> <p>1- El administrador accede al sistema para insertar un usuario</p> <p>1.1 El sistema muestra una interfaz para introducir los datos del usuario (nombre, usuario, contraseña), además de seleccionar el grupo al cual va a pertenecer.</p> <p>2- Luego de haber introducidos los datos correspondientes pulsa en el botón insertar.</p> <p>2.1 En el sistema se registran los datos del nuevo usuario</p>
EC 1.2: Insertar usuario falla	(Adaily11)I	V	El sistema muestra un mensaje de error indicando que se deben verificar los datos.	Satisfactorio	
	V	(ady)I	El sistema muestra un mensaje de error indicando que el usuario ya existe	Satisfactorio	
	(nada)I	V	El sistema muestra un mensaje de error indicando que se deben entrar todos los datos	Satisfactorio	

## *Capítulo 4. Implementación y prueba del sistema.*

Escenario	Variable 1 Nombre	Variable 2 Usuario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 2.1: Modificar usuario con éxito.		(ady) V	El sistema muestra una interfaz con los datos del usuario introducido.	Satisfactorio	<p>Una vez que el administrador se ha identificado como tal en el sistema.</p> <p>1- El administrador accede al sistema para modificar un usuario</p> <p>1.2 El sistema muestra una interfaz para introducir el usuario que desea modificar.</p> <p>2- Luego de haber introducido el nombre de usuario se muestra una interfaz con los datos del usuario introducido y se pulsa el botón modificar.</p> <p>2.1 En el sistema se registran los datos del usuario.</p>
	(Pepe) V	(pepito) V	El sistema muestra un mensaje indicando que se ha modificado el usuario con éxito.		
EC 2.2: Modificar usuario falla		(vacío) V	El sistema muestra un mensaje de error indicando que se debe introducir el usuario.	Satisfactorio	
	(Pepe1) V	(pepito) V	El sistema muestra un mensaje indicando que deben verificar los datos.		
		(ady1) I	El sistema muestra un mensaje de error indicando que el usuario no existe	Satisfactorio	

Escenario	Variable 1 Nombre	Variable 2 Usuario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 2.1: Eliminar usuario con éxito.		(pepito) V	El sistema muestra un mensaje indicando que el usuario ha sido eliminado	Satisfactorio	<p>Una vez que el administrador se ha identificado como tal en el sistema.</p> <p>3- El administrador accede al</p>

## Capítulo 4. Implementación y prueba del sistema.

EC 2.2: Eliminar usuario falla.		(vacio) V	El sistema muestra un mensaje de error indicando que se debe introducir el usuario.	Satisfactorio	sistema para eliminar un usuario  1.3 El sistema muestra una interfaz para introducir el
		(ady) I	El sistema muestra un	Satisfactorio	1.4 usuario que desea
			mensaje de error indicando que el usuario no existe		1.5 eliminar.  4- Luego de haber introducido el nombre de usuario se pulsa el botón eliminar.  2.1 En el sistema elimina los datos del usuario.

### 4.3.1 Resultado de las pruebas

El resultado de las pruebas realizadas al software fue satisfactorio, se comprobó que las respuestas del sistema son las esperadas y que el mismo cumple con las especificidades planteadas y a su vez coinciden con las descripciones de los casos de uso planteados con anterioridad. Con las pruebas aplicadas se comprobó que el sistema cumple con los objetivos propuestos.

### 4.4 Conclusiones

Durante la realización de este capítulo se realizó el flujo de trabajo de implementación, en el cual se describieron los elementos del diseño en términos de componentes, también quedó reflejada la distribución física de los componentes lógicos desarrollados durante la construcción del sistema, así como las conexiones entre ellos. Se realizaron las pruebas de caja negra donde quedaron definidos los diseños de casos de pruebas para comprobar las funcionalidades del sistema además de verificar las entradas y respuestas del mismo cumpliendo con los resultados esperados.

## 5.1 Conclusiones

Esta investigación tiene un papel fundamental, porque a partir de ella se deriva la implementación y aplicación de un Componente para el control de acceso a publicaciones de archivos multimedia en servidores de streaming libres, a raíz de su modelación, se garantizará la fiabilidad e integridad de la información.

- Logró concebirse, modelarse y especificarse un sitio Web para la gestión del control de acceso a servidores streaming libres.
- Se diseñó una base de datos capaz de almacenar de manera organizada la información que se manipula con la realización de las diferentes funcionalidades que brinda el sistema.
- Se logró documentar todo el proceso de desarrollo.

Con el presente trabajo se demuestra que la investigación no es actividad providencial de hombres particulares, es una fuerza motriz que influye positivamente en el desarrollo de todas y cada una de las esferas de nuestra vida económica y social, por lo tanto, también constituye un elemento de cambio en cuanto a la seguridad se refiere. Hágase de ella, un instrumento del quehacer cotidiano.

### Recomendaciones

Una vez vencidos los objetivos de esta investigación, y teniendo en cuenta las experiencias obtenidas a lo largo de su desarrollo se recomienda a la institución:

- Seguir trabajando en el desarrollo del sistema, si son añadidos a éste otros requerimientos funcionales, pues se cuenta con las especificaciones necesarias para comenzar a materializar la implantación del mismo.
- Generalizar el campo de aplicación del Componente para el control de acceso a publicaciones de archivos multimedia en servidores de streaming libres a otros departamentos que utilicen estas tecnologías.

Con la culminación de todas las etapas de desarrollo de software se demuestra, una vez más, que las Tecnologías de la Información y de la Comunicación (TIC), se están convirtiendo en un aspecto fundamental para el logro de una mayor organización y eficiencia; y el desarrollo de *software* se está convirtiendo en un producto muy importante en una sociedad con una población altamente entrenada, educada y capacitada.



### **Bibliografía consultada**

1. **Aguilar, Ing. Violena Hernández y Jorrín., Ing. Michael González.** *Proceso de pruebas de caja negra basado en la descripción de casos de uso.* Habana, Cuba : s.n.
2. **Rumbaugh, James, Jacobson, Ivar y Booch., Grady.** *El lenguaje unificado de modelado. Manual de referencia.*
3. **Pressman., Roger.** *Ingeniería de Software: Un enfoque práctico.*
4. **AUMAILLE., Benjamin.** *J2EE desarrollo de aplicaciones web.*
5. **Pelechano., Vicente.** *Servicios Web. Estándares,Extensiones y Perspectivas de Futuro.* Valencia : s.n.
6. **García., Rodrigo.** *Curso de doctorado Web Semántica: Tecnologías semánticas aplicadas a la definición de QoS.* Oviedo : s.n.
7. **J.J. Merelo, J.G. Castellano, P.A. Castillo, G. Romero.** *Algoritmos genéticos distribuidos usando SOAP.* .
8. W3C. . [En línea] <http://www.w3.org/TR/wsdl>..
9. W3C. [En línea] <http://www.w3.org/XML/>.
10. W3C. [En línea] <http://www.w3.org/TR/soap/>.
11. OASIS. [En línea] [www.oasis-open.org/committees/uddi-spec/](http://www.oasis-open.org/committees/uddi-spec/).
12. **Berrocal, Jose Luis Alonzo.** *Servidor de tutoriales y recursos de videos mediante streaming bajo demanda.* Universidad de Salamanca, 2009. : s.n.
13. **Apple Computer, Inc.** *Administrator's Guide.* .
14. **Murray Cumming, Thomas Vander Stichele,Christian Fredrik Kalager Schaller.** *Flumotion manual.* 2008.
15. **Pello Xabier Altadill Izura, Ingeniero Informático por la UPV-EHU.** *IPTABLES Manual práctico.* .
16. **Kruchten, Philippe.** *The Rational Unified.* . 2001.
17. **Ginestà, Marc Gibert y Mora, Oscar Pérez.** *Bases de datos en PostgreSQL.* .
18. **Altamirano, Alfonso Valdez.** *Ubicuos.* [En línea] <http://ubicuos.com>.
19. **María Sierra, Univ. Cantabria – Fac. de Ciencias.** *Trabajando con Visual Paradigm for UML.*

### Bibliografía citada

1. **AUMAILLE., Benjamin.** *J2EE desarrollo de aplicaciones web.*
2. **Pelechano., Vicente.** *Servicios Web. Estándares,Extensiones y Perspectivas de Futuro.* Valencia : s.n.
3. **García., Rodrigo.** *Curso de doctorado Web Semántica: Tecnologías semánticas aplicadas a la definición de QoS.* Oviedo : s.n.
4. **J.J. Merelo, J.G. Castellano, P.A. Castillo, G. Romero.** *Algoritmos genéticos distribuidos usando SOAP.* .
5. W3C. . [En línea] <http://www.w3.org/TR/wsdl/>.
6. W3C. [En línea] <http://www.w3.org/XML/>.
7. W3C. [En línea] <http://www.w3.org/TR/soap/>.
8. OASIS. [En línea] [www.oasis-open.org/committees/uddi-spec/](http://www.oasis-open.org/committees/uddi-spec/).
9. **Berrocal, Jose Luis Alonzo.** *Servidor de tutoriales y recursos de videos mediante streaming bajo demanda.* Universidad de Salamanca, 2009. : s.n.
10. **Apple Computer, Inc.** *Administrator's Guide.* .
11. **Murray Cumming, Thomas Vander Stichele,Christian Fredrik Kalager Schaller.** *Flumotion manual.* 2008.
12. **Pello Xabier Altadill Izura,** Ingeniero Informático por la UPV-EHU. *IPTABLES Manual práctico.* .
13. **Kruchten, Philippe.** *The Rational Unified.* . 2001.
14. **Ginestà, Marc Gibert y Mora, Oscar Pérez.** *Bases de datos en PostgreSQL.* .
15. **Altamirano, Alfonso Valdez.** Ubicuos. [En línea] <http://ubicuos.com>.
16. **María Sierra, Univ. Cantabria – Fac. de Ciencias.** *Trabajando con Visual Paradigm for UML.*
17. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* Madrid.
18. **Anna. C Grimán, María Pérez, Luis. E Mendoza.** *Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales.* Caracas, Venezuela.

### Glosario de términos

- **Apache:** Es el servidor web, disponible sin costo alguno, distribuido con licencia Open Source. Es compatible con sistemas operativos basados en Unix y con Windows.
- **Firewall:** Es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.
- **HTML: (*HyperText Markup Language*)** es el lenguaje de marcado predominante para la elaboración de páginas web.
- **HTTP: (*Hypertext Transfer Protocol*)** Protocolo de transmisión del hipertexto.
- **Internet:** Red de computadoras alrededor de todo el mundo que comparten información unas con otras por medio de páginas o sitios.
- **Kernel:** núcleo del sistema operativo.
- **PHP: (*Hypertext Pre-processor*)** Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recurrente que significa
- **RTP: (*Real-time Transport Protocol*)** utilizado para la transmisión de información en tiempo real.
- **RTSP: (*Real Time Streaming Protocol*)** establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de video.
- **Streaming:** Consiste en la transmisión y distribución de audio o video por Internet.
- **SQL: (*Structured Query Language*)** Lenguaje de Consulta Estructurado es un lenguaje declarativo de acceso a bases de datos relacionales
- **SSL: (*Secure Sockets Layer*).** Protocolo diseñado por la empresa Netscape para proveer comunicaciones encriptados.
- **UML: (*Unified Modeling Language*)** Lenguaje Unificado de Modelado. Es el lenguaje de modelado de sistema de software más conocido en la actualidad.
- **WWW: (*World Wide Web*)** Sistema de navegador web para extraer elementos de información llamados documentos o páginas web.
- **XML: (*Extensible Markup Language*)** un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium.