



Análisis y Diseño del módulo Mediador para un Sistema IPTV en la Universidad de las Ciencias Informáticas.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Yosiel Lorenzo Valero

Tutor: Ing. Víctor Frank Molina López

Ciudad de La Habana, Junio 2010
"Año 52 de la Revolución"



"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos..."

Dedicatoria

El presente trabajo de diploma se lo dedico primeramente a mi Papá y Mamá que a ellos les debo todo lo que soy, sus vidas las han dedicado a educarnos a mi hermana y a mí y pues aquí tienen el fruto de ese sacrificio, ya me tienen formado de ingeniero que era el que faltaba en ser un profesional.

También le dedico este trabajo a mi abuelo Roberto que no pudo verme como ingeniero pero estoy seguro que donde quiera que esté va estar orgulloso, como mismo lo estuvo cuando supo que estudiaría en la UCI.

Agradecimientos

En especial a mis padres y mi hermana que me han dado todo el apoyo, el amor y la fuerza para poder llegar hasta aquí.

A toda mi gran familia que de una forma u otra me han apoyado para que lograra ser Ingeniero.

A todos los profesores que he tenido a lo largo de mi vida, en las anteriores escuelas cursadas y en la UCI.

A mi tutor Víctor Frank por ayudarme mucho en la realización de este trabajo a pesar del poco tiempo que dispone.

A mi oponente Enrique Almeida por sus críticas muy constructivas y que además fue como un tutor para mí.

A Anay Chapman, presidenta del tribunal de tesis, pero también me ayudó como una tutora.

Al tribunal de tesis por sus críticas muy oportunas para mejorar el documento y la exposición.

A mis socios de la UCI: Pacheco, Diosbel, Serguey, Cinolkis, Falero, Pepe, Yumar.

A mis socios del barrio: Yankiel y Eiser.

A los profes Abelito y el Yoa por las ideas aportadas en la realización de este trabajo.

A Dayamí por toda la ayuda que me brindó en cada momento que necesité de ella.

A la Revolución, por darme la oportunidad de formarme en profesional.

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yosiel Lorenzo Valero

Ing: Víctor Frank Molina López

Datos de Contacto

Tutor(a)

Nombre y apellidos: Víctor Frank Molina López

Sexo: M **Institución:** UCI

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: vfmolina@uci.cu

Teléfono del trabajo:

Teléfono particular: 052252358

Categoría docente: Adiestrado

Grado científico: Adiestrado

Cargo del trabajador:

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas

Año de graduación: 2008

Institución donde se graduó: Universidad de las Ciencias Informáticas

La televisión en sus inicios era un derecho exclusivo de pequeños sectores de la sociedad, en la actualidad pocos rechazan este medio de difusión masiva. Desde los últimos años del siglo XX es completamente normal ser un televidente habitual, sin embargo, es difícil imaginar el volumen de trabajo que se debe realizar para desarrollar un sistema de televisión que simplemente haga llegar la señal a la pequeña pantalla. Un gran número de personas dedican sus días a desempeñarse en los más diversos roles en función de que ese fenómeno audiovisual llegue a distintos hogares. Existen varias formas de brindar televisión que han surgido con el transcurso del tiempo y con el avance de las tecnologías. Con estos avances se consiguen mejoras en la calidad del contenido audiovisual que recibe el televidente, además sistemas más avanzados ya logran que ocurra una interacción entre el televidente y los contenidos audiovisuales transmitidos en vivo.

En Cuba la Televisión (TV) es transmitida de forma analógica y más específicamente en la Universidad de las Ciencias Informáticas (UCI), se transmite TV analógica por cable. El usuario solamente puede realizar acciones tales como cambiar de canal y subir o bajar el volumen, con estas nunca modificará el contenido de lo que se transmite ya que son funciones propias que incluyen los dispositivos de televisión utilizados. El Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 9, debido a las características de la infraestructura que presenta la UCI, que cuenta con computadoras en casi todos los locales conectadas a una misma red, decidió crear un propio sistema de televisión más sofisticado que remplace el existente.

El objetivo del presente trabajo es diseñar uno de los módulos llamado Mediador. Luego será integrado con los demás subsistemas que conformarán la Plataforma de Televisión. Mediante este documento se presentan los resultados de la investigación realizada para lograr el diseño del módulo en cuestión, se realiza la correspondiente explicación de los conceptos y características más trascendentales que lo identifican. Además, se muestran los resultados de todos los diagramas obtenidos que conforman el análisis y diseño del Mediador.

| | |
|---|----|
| Introducción..... | 12 |
| Métodos Teóricos..... | 15 |
| Métodos Empíricos..... | 15 |
| Capítulo I: Fundamentación Teórica | 16 |
| 1.1 Introducción..... | 16 |
| 1.2 Televisión Digital (TVD, DTV)..... | 16 |
| 1.2.1 Televisión Digital Terrestre (TDT) | 17 |
| 1.2.2 Televisión Digital por Cable..... | 17 |
| 1.2.3 Televisión Digital por Satélite | 17 |
| 1.2.4 Televisión Digital por tecnologías IP | 17 |
| 1.3 Situación Problemática..... | 18 |
| 1.4 Middleware y su módulo Mediador | 18 |
| 1.4.1 Funciones de Mediador..... | 19 |
| 1.5 Proceso de funcionamiento de un Sistema IPTV..... | 23 |
| 1.6 Soluciones Existentes. | 23 |
| 1.6.1 Empresa Huawei..... | 24 |
| 1.6.2 Empresa Zhong Xing Telecommunication Equipment (ZTE por sus siglas en Inglés). | 24 |
| 1.6.3 Empresa IPTV Américas..... | 25 |
| 1.6.4 Empresa NetUp. | 26 |
| 1.7 Conclusiones parciales..... | 29 |
| Capítulo 2: Herramientas y Tecnologías Actuales | 31 |
| 2.1 Introducción | 31 |
| 2.2 Metodologías de desarrollo..... | 31 |
| 2.2.1 Metodologías ágiles. | 32 |
| 2.2.2 Metodologías pesadas. | 33 |
| 2.3 Selección de la metodología a utilizar..... | 35 |
| 2.4 Lenguaje Unificado de Modelado (UML)..... | 36 |
| 2.4.1 Tipo de modelos UML | 36 |
| 2.5 Herramientas CASE | 37 |
| 2.5.1 Rational Rose Enterprise | 39 |

| | |
|---|----|
| 2.5.2 Visual Paradigm..... | 40 |
| 2.6 Selección de la herramienta CASE..... | 41 |
| 2.7 Conclusiones parciales..... | 41 |
| Capítulo 3: Análisis y Diseño del Sistema | 42 |
| 3.1 Introducción..... | 42 |
| 3.2 Modelo de Dominio..... | 42 |
| 3.2.1 ¿Por qué modelo de dominio? | 42 |
| 3.2.2 Conceptos del modelo de dominio. | 43 |
| 3.2.3 Diagrama del Modelo del Dominio. | 44 |
| 3.2.4 Descripción del Modelo de Dominio | 44 |
| 3.3 Requisitos Funcionales | 45 |
| 3.4 Requisitos no Funcionales..... | 46 |
| 3.5 Modelo de Casos de Usos del Sistema | 48 |
| 3.5.1 Actores del Sistema | 48 |
| 3.5.2. Casos de Uso del Sistema..... | 48 |
| 3.6 Diagrama de Casos de Uso del Sistema | 50 |
| 3.7 Descripción de los Casos de Uso del Sistema..... | 51 |
| 3.8 Análisis..... | 55 |
| 3.8.1 Diagramas de Clases del Análisis..... | 55 |
| 3.8.1 Diagramas de Colaboración para cada Diagrama de Clases del Análisis. | 58 |
| 3.9 Arquitectura del Software. | 60 |
| 3.9.1 Patrones. | 61 |
| 3.9.2 Patrones de Arquitectura..... | 62 |
| 3.10 Diseño. | 65 |
| 3.10.1 Patrones de Diseño..... | 65 |
| 3.11 Diagramas de Clases del Diseño para cada Caso de Uso..... | 67 |
| 3.12 Conclusiones Parciales | 70 |
| Conclusiones | 72 |
| Recomendaciones | 73 |
| Bibliografía Citada..... | 74 |
| Bibliografía Consultada..... | 76 |

Índice de Figuras

| | |
|--|----|
| Figura 1: Interfaz NetUp Middleware..... | 27 |
| Figura 2: Interfaz del sistema IPTV | 28 |
| Figura 3: Interfaz del Sistema IPTV | 29 |
| Figura 4: Fases de RUP..... | 34 |
| Figura 5: Modelo de Dominio | 44 |
| Figura 6: Diagrama de Casos de Uso del Sistema..... | 50 |
| Figura 7: Diagrama de Clases del Análisis del Caso de Uso Verificar Terminal. | 56 |
| Figura 8: Diagrama de Clases del Análisis del Caso de Uso Gestionar Canal. | 56 |
| Figura 9: Diagrama de Clases del Análisis del Caso de Uso Gestionar Archivo VoD..... | 57 |
| Figura 10: Diagrama de Clases del Análisis del Caso de Uso Gestionar EPG..... | 57 |
| Figura 11: Diagrama de Colaboración del Caso de Uso Verificar Terminal..... | 58 |
| Figura 12: Diagrama de Colaboración del Caso de Uso Gestionar Canal..... | 59 |
| Figura 13: Diagrama de Colaboración del Caso de Uso Gestionar Archivo VoD..... | 59 |
| Figura 14: Diagrama de Colaboración del Caso de Uso Gestionar EPG..... | 60 |
| Figura 15: Fases e Hitos en RUP..... | 61 |
| Figura 16: Patrón Modelo Vista Controlador | 62 |
| Figura 17: Diagrama de Clases del Diseño del Caso de Uso Verificar Terminal. | 67 |
| Figura 18: Diagrama de Clases del Diseño del Caso de Uso Gestionar Canal. | 68 |
| Figura 19: Diagrama de Clases del Diseño del Caso de Uso Gestionar Archivo VoD..... | 69 |
| Figura 20: Diagrama de Clases del Diseño del Caso de Uso Gestionar EPG. | 70 |

Introducción

Desde hace varias décadas el impacto de la informática en la vida del hombre es cada vez más evidente. Para todos los países es una necesidad imperiosa adoptar las Tecnologías de la Informática y las Comunicaciones (TIC) para alcanzar sus perspectivas de desarrollo. La producción y consumo de aplicaciones informáticas está marcando pautas en las relaciones económicas a nivel internacional.

En Cuba, desde el triunfo de la Revolución se realizan ingentes esfuerzos por propiciar e incentivar el desarrollo de la informática. El ejemplo más fehaciente lo constituye la creación de la Universidad de las Ciencias Informáticas, institución forjada al calor de la Batalla de Ideas, y que posee un elevado compromiso social.

Dicha universidad está estructurada por varias facultades en su sede principal y tres facultades regionales, todas con diversas líneas de investigación y producción. Entre ellas está la facultad 9, en la cual se encuentra el Centro de Geoinformática y Señales Digitales (GEYSED). Este cuenta con un Departamento de Señales Digitales que se dedica a *investigar* y producir en el campo de la televisión. A la par que se forma personal para brindar productos y servicios informáticos en el área de video y sonido digital, se trabaja en el desarrollo de productos software en la rama de la televisión y la gestión de contenido audiovisual. Todas estas actividades se realizan con el objetivo de lograr la informatización de la sociedad cubana como primera prioridad y luego la comercialización como parte de la experiencia adquirida en el campo del video y sonido digital.

La televisión en la universidad es un medio de comunicación que juega un gran papel ya que mediante esta se logra formar profesionalmente a estudiantes y profesores; y logra mantener informado a todos, además de proporcionar entretenimiento y recreación a toda la comunidad universitaria. En todo el país se transmite televisión analógica y más específicamente en la UCI, la televisión se transmite de forma analógica por cable. El usuario solamente puede realizar acciones tales como cambiar de canal, subir o bajar el volumen, en resumen, acciones que nunca modificarán el contenido de lo que se transmite ya que estas son funciones propias que incluyen los dispositivos de televisión de este usuario.

Esto trae consigo que se estén analizando alternativas de televisión digital por el valor agregado que brinda a los usuarios: mejora la calidad de la imagen y el sonido, aumenta las opciones en cuanto a la cantidad de canales, la alta definición y las posibilidades de crear aplicaciones interactivas. Muchos

países desarrollados han implantado este tipo de tecnología como lo son: Estados Unidos, Japón, España, Inglaterra y Francia. También en América Latina otros como: México, Brasil y Argentina ya han alcanzado adoptar un estándar, siendo estas unas de las pocas naciones de Latinoamérica que lo han logrado. (1)

En la UCI se cuenta con un sitio web al cual tiene acceso toda la comunidad universitaria, este se llama **Inter-nos** y tiene como función principal contribuir con el proceso docente educativo en la universidad pues en el mismo se publican teleclases, materiales educativos, políticos y de apoyo a la docencia. Integra servicios de audio y video por la web usando un reproductor embebido y la visualización de estas medias online se realiza mediante la tecnología streaming, utilizada para acelerar la descarga y ejecución de estos contenidos audiovisuales en la web, ya que permite reproducir los archivos mientras se están descargando. Para este fin se emplea Windows Media Server, tecnología de Microsoft para proporcionar el servidor de streaming, el formato de video que utiliza es Windows Media Video (WMV). Está implementado usando el lenguaje de programación ASP, como servidor web Microsoft Internet Information Services (IIS), Windows Media Server como servidor de streaming y el sistema gestor de base de datos Microsoft SQL Server 2000. El proceso de publicación de medias se realiza de forma manual, desde el sitio se habilita el vínculo, pero se debe copiar el fichero a la carpeta indicada para que el servidor logre conformar el streaming y a pesar de que proporciona de alguna forma la interactividad entre el usuario y el contenido que el mismo reproduce, no favorece la completa interoperabilidad necesaria.

La forma de televisión digital que más se presta a aplicar en esta universidad es la televisión digital por IP, pues se cuenta con computadoras en casi todos los locales del centro que están conectadas a la red local existente. Con las potencialidades existentes, sería conveniente utilizar un sistema de televisión por IP (IPTV) en la UCI. El cual proporcionaría un sencillo uso de sus servicios, gran variedad de sus contenidos, soportando una mayor cantidad de canales. El nuevo sistema brindaría la posibilidad que se emitieran los contenidos de forma más rápida y con mayor calidad y cada usuario podría contar con una programación individual. Con todas estas posibilidades los usuarios lograrían una mayor interactividad con el sistema de televisión. Por tales razones surge en la universidad un grupo IPTV que tiene como misión realizar un sistema IPTV. Para su desarrollo es necesario la realización de diferentes subsistemas, como son:

Middleware¹, Head End (HE), Servidor de video bajo demanda, Guía Electrónica de Programación (EPG), Set Top Box (STB), Sistema de control de derechos (DRM).

El Middleware está compuesto por dos módulos que son el Mediador y el TV Portal, el primero es el encargado de gestionar los flujos de datos entre todos los subsistemas (2). El módulo Mediador tiene gran importancia dentro del Middleware ya que realiza gran parte de las funciones de este. El subsistema Middleware es imprescindible para conseguir brindar televisión por IP pues es responsable de gestionar, coordinar las actividades y llevar a cabo el intercambio de los recursos dentro del sistema IPTV, por estas razones surge la inmediata necesidad de desarrollar un Sistema Operativo Middleware sobre tecnologías libres para una Plataforma IPTV en la UCI, de ahí que se identifique el **problema a resolver**: Inexistencia de un Sistema Operativo Middleware sobre tecnologías libres para una Plataforma IPTV en la UCI.

La investigación se enmarca en el **objeto de estudio**: proceso de funcionamiento de un Sistema IPTV y el **campo de acción**: Sistemas Operativos para sistemas IPTV.

Como **idea a defender** se tiene que con el estudio de los procesos de gestión dentro de las plataformas IPTV, se podría realizar el análisis y diseño del módulo mediador del Middleware para un sistema de igual tipo en la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado se define como **objetivo general** diseñar el módulo Mediador de un Sistema Operativo para una Plataforma IPTV.

Para lograr este objetivo se desarrollan las siguientes **Tareas de la investigación**:

1. Valorar soluciones existentes de IPTV que contengan Sistemas Operativos.
2. Describir el proceso de gestión de todos los módulos existentes en un sistema IPTV.
3. Seleccionar las metodologías de Análisis y Diseño para la aplicación, que permita su creación con mayor facilidad y garantice la calidad de la misma.
4. Determinar las herramientas para el desarrollo de la aplicación y la elección de la plataforma en la que se desarrollará, fundamentando dicha elección.
5. Realizar el análisis y diseño del módulo Mediador del Sistema Operativo.

¹Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

Para la realización de estas tareas se emplearon los diferentes **métodos de la investigación**:

Métodos Teóricos

- “Histórico – Lógico”: este método permitió determinar las características esenciales de las principales tecnologías IPTV existentes y realizar el estudio de la trayectoria real de determinados elementos que servirán de guía para la construcción del Sistema Operativo (Middleware) para la plataforma IPTV que se requiere.
- “Analítico - Sintético”: este método permitió realizar un análisis de la documentación de cada una de las herramientas que se utilizaron en el sistema diseñado.
- “Modelación”: Este método se utilizó en la modelación de diagramas teniendo en cuenta la metodología de desarrollo de software seleccionada.

Métodos Empíricos

- “Entrevista”: Se realizó la entrevista a 8 trabajadores de la Dirección de Televisión Universitaria (DTU) para conocer los antecedentes sobre la televisión y tener claro el dominio del problema, es decir, la población estuvo constituida por 8 miembros. De esta se tomaron 4 miembros como muestra para realizarle la entrevista, representando el 50% de la población. La técnica de muestreo realizada fue muestreo no probabilístico, exactamente muestreo intencional.

Capítulo I: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se describen los conceptos relacionados con el problema planteado anteriormente y con el objetivo general del trabajo investigativo realizado. Además, se abordan temas relacionados con el proceso de gestión de todos los módulos existentes en un sistema IPTV, se muestra de manera detallada la situación problemática expuesta y además se realiza un análisis de las principales soluciones existentes de IPTV para entender el funcionamiento de sus Sistemas Operativos.

Para una mejor comprensión del trabajo realizado se hace necesario el dominio de algunos conceptos tratados en el cuerpo del documento a los cuales se le hace referencia a continuación.

1.2 Televisión Digital (TVD, DTV)

La televisión digital (o DTV, de sus siglas en inglés: Digital TV) se refiere al conjunto de tecnologías de transmisión y recepción de imagen y sonido, a través de señales digitales. En contraste con la televisión analógica, la televisión digital codifica los datos de forma binaria, lo cual proporciona una mayor calidad de señal, evita interferencias o fallas en la transmisión. Brinda la posibilidad de crear aplicaciones interactivas y la capacidad de transmitir varias señales en un mismo canal asignado, gracias a la diversidad de formatos existentes. (3)

Las vías más importantes por las cuales los clientes tienen acceso a la Televisión Digital son:

- Televisión Digital Terrestre (TDT).
- Televisión Digital por Cable.
- Televisión Digital por Satélite.
- Televisión Digital por tecnologías IP (IPTV).

1.2.1 Televisión Digital Terrestre (TDT)

Es una plataforma que permite optimizar la transmisión de datos, requiere para la transmisión canales similares a los de la televisión analógica actual de un ancho de banda reducido y no deja canales en desuso entre una señal y otra, ya que se producen menos interferencias, mejora la calidad audiovisual, esta señal es transmitida mediante ondas hertzianas terrestres. (4)

1.2.2 Televisión Digital por Cable

La señal digital de televisión se distribuye por medio de redes híbridas de fibra óptica y cable coaxial. Junto con la señal de televisión digital, a través de estas redes se proporcionan otros servicios como radio, telefonía fija y acceso a Internet. (5)

1.2.3 Televisión Digital por Satélite

La señal digital se transmite a una amplia zona geográfica por medio de satélites de comunicaciones, en diferencia con la televisión terrestre, cuyas ondas no salen de la atmósfera. La transmisión de Televisión Digital vía Satélite se divide en dos tramos claramente diferenciados:

1. El enlace ascendente o uplink, mediante el cual el centro emisor envía las señales de televisión al satélite utilizando grandes antenas parabólicas (de 9 a 12 metros de diámetro).
2. El enlace descendente, o downlink, por medio del cual el satélite retransmite la señal de televisión recibida hacia su zona de cobertura sobre la superficie de la tierra, utilizando una banda de frecuencias desigual a la del enlace ascendente, para evitar interferencias. (6)

1.2.4 Televisión Digital por tecnologías IP

La señal de televisión es transmitida por medio de las Líneas de Suscripción Digital (por sus siglas en inglés xDSL) o por redes comunes de computadoras. Este servicio llega al cliente mediante su conexión a internet, pero con un ancho de banda reservado sobre la misma infraestructura, lo que implica una mayor velocidad en la transferencia de datos. (7)

La clave está en la personalización del contenido para cada cliente de manera individual. Esto permite el desarrollo de video bajo demanda o pago por evento. El usuario dispondrá de un equipo receptor conectado a su ordenador o a su televisión y a través de una guía podrá seleccionar los contenidos que

desea ver o descargar para almacenar en el receptor y de esta manera poder visualizarlos tantas veces como desee. (7)

Un sistema IPTV está compuesto por varios subsistemas como son:

- Middleware.
- Head End (HE).
- Servidor de video bajo demanda.
- Guía Electrónica de Programación (EPG).
- Set Top Box (STB).
- Sistema de control de derechos (DRM).

1.3 Situación Problemática

En la UCI, la forma de televisión digital más conveniente a aplicar es la Televisión Digital por IP, pues se cuenta con computadoras conectadas en casi todos los locales del centro. Un sistema IPTV provee facilidades de uso de sus servicios, una gran variedad de sus contenidos, soporta mayor cantidad de canales que otras formas de televisión. Por esta infraestructura y recursos con los cuales cuenta la UCI se reducen bastante los costos. Con este sistema se conseguiría que se emitan los contenidos de forma más rápida y con mayor calidad audiovisual, cada usuario podría contar con una programación individual. Con estas posibilidades los usuarios lograrían una mayor interactividad con la televisión. Para el desarrollo del sistema de televisión por IP en la UCI, se crea un proyecto IPTV en la facultad 9, que está encargado de diseñar e implementar cada uno de los subsistemas que integran la novedosa plataforma de TV. Los subsistemas necesitados son: Middleware, Head End (HE), Servidor de video bajo demanda, EPG, Set Top Box (STB), Sistema de control de derechos (DRM por sus siglas en Inglés); pudiéndose decir que ninguno de estos está al menos diseñado.

Dentro de ellos, el Middleware es el de mayor importancia ya que es el sistema operativo de la plataforma IPTV. Está compuesto por dos módulos que son el Mediador y el TV Portal, el primero gestiona los flujos de datos entre todos los subsistemas. Por la importancia que tiene el Mediador surge de inmediata necesidad desarrollar un Sistema Operativo Middleware sobre tecnologías libres para una Plataforma IPTV en la UCI.

1.4 Middleware y su módulo Mediador

El Middleware es el sistema encargado de gestionar la entrega de servicios de IPTV. Este define y coordina la forma en que el usuario interactúa con el servicio de IPTV, y además da soporte a la interacción de los distintos servidores de aplicaciones con el Head End. También podrá contar con un módulo de evaluación de servicios. El Middleware constituye una especie de servidor de portal que es accedido y utilizado por una aplicación “cliente” que se ejecuta en el STB. (2)

El Mediador es uno de los dos subsistemas con que cuenta el Middleware, este es el encargado de interactuar con los distintos sistemas de gestión y elementos de la red. Con diversos niveles de seguridad este permitirá a los operadores realizar las modificaciones al sistema para que esté configurado de acuerdo con las necesidades de la solución implementada.

1.4.1 Funciones de Mediador

El Mediador desempeña un papel de gran importancia dentro del Middleware ya que gestiona gran parte de las tareas que resuelve el Middleware en general, a continuación se explican todas las funciones de este módulo mediador.

Gestión de STB

El Mediador debe de realizar la administración de los STB de los clientes, controlando el acceso estos al sistema IPTV en función de su identificación. Además, asocia el STB al cliente correspondiente permitiendo el acceso solamente a los servicios con los que cuenta en el contrato. (2)

Cuando un STB es iniciado y se autoriza a acceder a la red, el Middleware le envía la información de la EPG la cual es almacenada en su memoria. En caso de modificación de la EPG, este es reenviado a todos los STBs para que se actualicen los mismos. (2)

Gestión de Servicios

El Mediador debe disponer de la capacidad de administrar y configurar servicios del sistema. Además, es el encargado de coordinar las distintas plataformas para la puesta en marcha y facturación de los servicios por lo que debe disponer de funciones que permitan al operador configurarlos o debe intermediar con algún sistema que realice esta función. Debe administrar:

- Canales en vivo.

- VoD: Video bajo demanda.
- PPV: (por sus siglas en inglés Pay Per View).
- PVR (por sus siglas en inglés Personal Video Recorder) y TSTV (por sus siglas en inglés Time Shifted TV). (2)

Gestión de la EPG

En el Mediador se deberá configurar el orden de los canales y la asignación de los mismos en los distintos canales de multicast de la red para que luego los mismos estén asociados a la información de la EPG.

A su vez, el Mediador soporta la posibilidad de importación automática de la programación en vivo futura desde fuentes externas de modo que sea integrada a la EPG. Esta información incluye los metadatos de la programación, esto hace que el mantenimiento de la EPG se realice de forma automática. (2)

Si se incluyen canales propios o locales de los cuales no se puede disponer un proveedor de la programación en un formato estándar para ser importado por la EPG es necesario recurrir a algún sistema manual para el ingreso de la programación futura. (2)

Gestión del contenido

El Mediador deberá realizar tareas de administración de los contenidos a demanda incorporados al sistema. Cada vez que se agregue una nueva película se deberá agregar información adecuada para su promoción, clasificación y búsqueda.

Las funciones principales para la gestión del contenido son:

- Configurar, crear e importar metadatos.
- Agregar posters y trailers de presentación de una nueva película.
- Facilidades para contar con la información de rating (¿cuánto gusta la misma?). (2)

Distribución del contenido

Por su alto consumo de la red, es conveniente almacenar el contenido bajo demanda lo más próximo posible a los clientes. Por lo tanto, todas las soluciones proponen un sistema con el cual se tengan varios servidores de video distribuidos en la red.

La propuesta general es la de disponer de un servidor de video con una capacidad de almacenamiento grande en el sitio central donde se guardan todas las películas disponibles y a su vez disponibilidad de

servidores de video de menor porte distribuidos por la red donde se almacenan las películas más vistas. Estas tareas se pueden resolver en forma automática.

El Middleware, en base a las estadísticas que dispone interactúa con los servidores de video para realizar en forma dinámica una correcta distribución de contenidos en la red. (2)

Gestión de PVR (por sus siglas en inglés Personal Video Recorder) **y TSTV** (por sus siglas en inglés Time Shifted TV)

El Mediador utiliza los recursos de los servidores de video para implementar las funcionalidades de PVR y TSTV. Ambas deben de ser definidas de manera muy cuidadosa para evitar problemas de exceso de tráfico en la red. (2)

Para el caso del PVR se pueden configurar determinados perfiles para ser comercializados, en estos cambiarán:

- Máximo de horas posibles a almacenar por usuario.
- Cantidad de veces que un contenido grabado podrá ser reproducido.
- Caducidad de almacenamiento.

En el caso del TSTV, se puede configurar la cantidad de minutos de un programa en vivo que se almacenará para luego observar de forma diferida, pero para evitar el posible peligro de congestión en la red algunos sistemas disponen de la posibilidad de limitar la cantidad de intentos simultáneos de utilización del TSTV. De otro modo, la posibilidad de no habilitar el uso de TSTV en algunos canales permitirá evitar este tipo de problemas. (2)

Gestión de suscriptores

El Mediador debe disponer de la facilidad de administración de clientes o debe intermediar con un sistema que realice esta función. De este modo se mantendrá toda la información de clientes almacenada para la normal operación del sistema. Esta información estará almacenada en una o varias bases de datos y deberá incluir como mínimo: datos administrativos del cliente, la asociación de cada cliente con los servicios o paquetes contratados y la cantidad y tipo de terminales autorizados, a su vez permitirá autenticar el cliente para el pago de cuentas o acceso parental e incluirá información que se utilizará para la identificación de los registros de uso de los servicios a demanda. (2)

Para realizar la administración de clientes se deberá incluir una interfaz por software y manual para:

- Realizar el abastecimiento de los clientes (alta, baja, bloqueo).
- Habilitar cantidad de STB permitidos por cliente.
- Permitir perfiles de clientes y permisos de acceso.
- Habilitar crédito y límites de gasto por cliente.
- Definir, reiniciar clave de los clientes (PIN).
- Activar o desactivar servicios.
- Importar/exportar información de los clientes de sistemas existentes. (2)

Mediación con el sistema de facturación

A través del Mediador se realiza la asignación de precios a las suscripciones mensuales, servicios adicionales y consumo de contenidos a demanda.

En la medida que los servicios son utilizados por un cliente, el Mediador registra en una base de datos todas las transacciones realizadas con la información de este a quien deberá cargarse ese uso. (2)

Gestión de Paquetes y Precios

El sistema de IPTV cuenta con la facilidad de crear paquetes de contenido o de servicios. Cada operador armará estos paquetes de acuerdo con su estrategia de negocio o marketing. (2)

Integración con el DRM

En caso que los diversos contenidos de una solución estén encriptados con más de un tipo de DRM, el Mediador será el responsable de hacer transparente al usuario final este problema. Esto puede ocurrir debido a que en general las exigencias sobre el uso del DRM la imponen los proveedores de contenidos y puede ser que se disponga de un DRM para las señales en vivo y otro para las señales de VoD. (2)

Reportes

El Mediador puede crear reportes de diversos tipos. En el caso de video en vivo se puede conocer exactamente la audiencia de los distintos canales a toda hora, determinar los picos de rating, conocer los picos de uso y calcular todo tipo de información respecto a hábitos de uso. Esta información no solo es de gran utilidad para realizar la planificación y el marketing del producto si no que puede ser comercializada a los proveedores de contenido de estas señales. (2)

Respecto a los contenidos bajo demanda, se puede monitorear la utilización de los diversos servidores de video y el funcionamiento del sistema automático de distribución de contenidos. Por otro lado, se puede identificar cuáles son los contenidos más vistos, asociarlo por ciudades, regiones o grupo de clientes. (2)

Es importante que el sistema de contenido a demanda disponga de la herramienta de auditoría. Los contenidos a demanda en general son distribuidos para ser comercializados con la modalidad de reparto de ingresos por lo que es fundamental para el proveedor de contenido poder realizar auditorías y verificar la cantidad de veces que fueron vistas sus películas en el sistema.

1.5 Proceso de funcionamiento de un Sistema IPTV.

Como objeto de estudio de este trabajo se tiene el proceso de funcionamiento de un sistema IPTV.

Entre los elementos o subsistemas que deben conformar un sistema IPTV se encuentran, Head End (HE), Subsistema de video a demanda (VoD), Middleware (MW), Subsistema de control de derechos (DRM), Set Top Box (STB). (2)

El Head End será el conjunto de elementos que cumplen las funciones de recibir las Señales en vivo y convertirlas al formato necesario para su transmisión por la red y su posterior recepción por los STB. El Subsistema de video bajo demanda tiene como ocupación almacenar y transmitir a la red videos y pistas de audio que podrán ser solicitados por los clientes para ser recibidos a demanda. La función del Middleware es soportar la entrega de servicios de IPTV. Este define y coordina la forma en que el usuario interactúa con el servicio de IPTV y soporta la interacción de los distintos servidores de aplicaciones con el Head End. El Middleware constituye una especie de servidor de portal que es accedido y utilizado por una aplicación cliente que se ejecuta en el STB, funciona de sistema operativo ya que se encarga de gestionar los flujos de datos entre todos los subsistemas. El DRM será el encargado de la inscripción de los contenidos de modo que no se violen los derechos de propiedad intelectual de los contenidos al ser transmitidos en la red. Los STBs serán los dispositivos que se encontraran conectados con el televisor del cliente. Deberán decodificar las señales para que se transporten como flujos de información multimedia sobre el protocolo IP para hacerla compatibles con una TV. Cada STB deberá disponer de un control remoto el cual será utilizado por los clientes para enviar las órdenes al sistema. (2)

1.6 Soluciones Existentes.

Se realizó un estudio del estado del arte de los sistemas IPTV a nivel mundial y nacional, aunque en Cuba todavía no se ha logrado implementar ningún sistema de este tipo. Solo algunas entidades aisladas han

dado algunos pasos distantes en este sentido, principalmente grupos de trabajo de la Empresa de Telecomunicaciones de Cuba (ETECSA), quienes lograron un cuasi-sistema IPTV en el hotel Montehabana transmitiendo por cableado eléctrico (PLC), pero muy lejos de los estándares internacionales existentes y pobres en cuanto a cantidad de aplicaciones e interactividad.

Las empresas que se mencionan a continuación constituyen la vanguardia de esta tecnología en el mundo, es por eso que son tomadas como referentes en esta rama.

1.6.1 Empresa Huawei.



A la vanguardia en cuanto al desarrollo de la tecnología IPTV se encuentra la gran empresa china **Huawei**. Las soluciones de IPTV que esta empresa ha desarrollado uniformemente son: Red Futura y Hogar Digital, para ofrecer a los usuarios la experiencia de una amplia gama de servicios. Además, para ayudar a los proveedores de contenido, a los proveedores de servicios y los operadores de red a proporcionar una gama de aplicaciones cada vez más desarrollada. Huawei ofrece una solución completa que consta de cinco capas: red doméstica, red portadora, distribución del contenido, soporte de servicios y operación de contenido. Gracias a su arquitectura de servicios abiertos, la solución proporciona posibilidades de integración de servicios E2E (end-to-end por sus siglas en inglés). (8)

Huawei ha estado operando en 13 países en América Latina. Esta empresa ha ganado cada vez más presencia a nivel internacional, sus productos y soluciones se utilizan en más de 100 países y atienden las necesidades de más de mil millones de usuarios de todo el mundo. (9)

Las soluciones desarrolladas por esta empresa ofrecen servicio a nivel metropolitano, es decir, son grandes soluciones para ciudades, con un valor de adquisición bastante elevado por lo que no sería conveniente hacer la compra de una de estas para su despliegue en la UCI.

1.6.2 Empresa Zhong Xing Telecommunication Equipment (ZTE por sus siglas en Inglés).



ZTE中兴

La **Zhong Xing** Telecommunication Equipment Company Limited, más conocida por su acrónimo ZTE, es una de las grandes empresas que marchan a la vanguardia en cuanto al desarrollo de tecnologías IPTV. Como miembro principal de la Norma de Comunicación de la Asociación de China (CCSA), ZTE ha asumido un papel preponderante en la elaboración de una serie de normas de IPTV, sola o conjuntamente con socios industriales. Además, ZTE es un patrocinador de AVS Alianza de la Industria de China, que ha definido un estándar de la industria de IPTV líder junto con AVS sistemas de prueba de IPTV. Ha ganado la mayor cuota de mercado en China (más del 50%), con su IPTV "Eye-Will" plataforma y servicios desplegados, por los dos mayores proveedores de la red fija de línea, China Telecom y China Netcom, en más de quince provincias. Sirve a más de 800.000 usuarios residenciales con servicios de IPTV. Ha proporcionado ricas características y rentables soluciones de IPTV para operadores internacionales en diez países, incluyendo Colombia, Tailandia, Bielorrusia, entre otros lo que propicia que se haya convertido en un actor global importante en todos los ámbitos de las telecomunicaciones. (10)

Las soluciones brindadas por esta compañía también son diseñadas para brindarle servicios a un gran número de usuarios residenciales, es decir, a grandes provincias y no sería rentable adquirir una de sus soluciones ya que donde se desea aplicar es en la UCI, la cual cuenta con una infraestructura bastante pequeña en comparación con una gran ciudad, por lo que con su compra se incurriría en un gasto monetario innecesario.

1.6.3 Empresa IPTV Américas.



Un ejemplo que se encuentra a la vanguardia en el continente americano es **IPTV Américas** esta empresa fue diseñada con el único propósito de facilitar la implementación de los servicios de IPTV en América Latina. Sus fundadores tienen una extensa experiencia en el desarrollo de la industria de la TV, incluyendo el lanzamiento y operación del primer satélite dedicado a la televisión de

pago en América Latina (PAS-1), el lanzamiento y distribución de varios canales de TV paga y de iniciar varias operaciones de cable. IPTV Américas tiene más de 100 años de experiencia combinada en la industria de los medios latinoamericanos. (11)

Una de las soluciones que ha presentado esta empresa es CA/DRM IPTV Américas, con esta se puede monitorear y controlar la red y su contenido. Además, puede ofrecer servicios como música digital, pay-per-view, guía electrónica de programas y otros servicios de valor agregado. (11)

IPTV Américas le ofrece a las compañías operadoras de servicio una solución a costos razonables. IPTV Américas ofrece una alineación de canales compuesta de aproximadamente 60 canales básicos, 15 canales Premium, y 20 canales digitales de música, además ofrece servicios adicionales para facilitar la compra del middleware, de los servicios de acceso condicionales y equipo relacionados tales como las cajas digitales (STBs). (11)

Esta solución pudiera ser muy conveniente instalarla en Cuba ya que brinda sus servicios a muchos países latinos y sus precios son bastante asequibles, pero el principal problema es que su cabecera IPTV está centralizada en Miami con acceso satelital y vía fibra óptica a toda Latinoamérica, pero Cuba no puede contar con ninguno de estos servicios ya que los Estados Unidos con el bloqueo que tiene impuesto lo deja excluido de poder ser un cliente más.

1.6.4 Empresa NetUp.



Otra gran empresa que se encuentra a la vanguardia en cuanto a las tecnologías IPTV es la compañía rusa NetUp que fue creada en el año 2001, ha logrado un gran desarrollo en el ámbito de las telecomunicaciones y brinda soluciones IPTV, en este poco tiempo logró crear un sistema completo de hardware y software para ofrecer la televisión interactiva.

Uno de sus productos más avanzados es NetUp IPTV complejo, este involucra un sistema Middleware, siendo este la parte más importante del hardware-software. Con este se logra además de la interfaz de usuario gráfica más rápida, que el servidor esté menos ocupado, amplias posibilidades de personalización, otra ventaja indiscutible es que con este se puede utilizar una red de suscriptores de varios tipos de consolas de diferentes fabricantes. Las siguientes imágenes muestran la interfaz del sistema IPTV NetUp Middleware. (12)



Figura 1: Interfaz NetUp Middleware. (12)



Figura 2: Interfaz del sistema IPTV. (12)



Figura 3: Interfaz del Sistema IPTV. (12)

1.7 Conclusiones parciales.

Con el desarrollo de este capítulo se realizó un análisis de los elementos teóricos, lográndose el entendimiento de una serie de conceptos sobre los sistemas IPTV y su forma de funcionamiento que sirven como base para dar solución al problema científico y a los objetivos planteados en el estudio realizado. Además, la descripción general de la situación problemática con la que se consigue entender mejor el problema que existe en la universidad con respecto a construcción de un propio sistema IPTV.

Fueron mostrados ejemplos de soluciones existentes a nivel internacional en el campo de los sistemas IPTV que contienen sistemas operativos Middleware. Quedando claro que es necesaria la realización de un propio sistema IPTV para la UCI ya que se ahorraría una gran suma de dinero que cuesta adquirir una

de esas y que estas soluciones existentes son mayormente para aplicar en grandes zonas residenciales con un gran número de usuarios. El país no se encuentra en condiciones de realizar esos gastos, pudiendo desarrollarse una solución de igual tipo en la UCI y que además luego este sistema propio podría ser comercializado.

Capítulo 2: Herramientas y Tecnologías Actuales

2.1 Introducción

En el presente capítulo son analizadas las tendencias y tecnologías actuales a nivel mundial, se escoge la metodología más adecuada para lograr el desarrollo de la propuesta de solución según las características del proyecto. Además, se estudia el lenguaje de modelado que se va a utilizar para lograr el diseño del sistema y se realiza un análisis de las herramientas CASE y se selecciona la más conveniente.

2.2 Metodologías de desarrollo

Una tarea tan importante como el proceso de desarrollo del software exige de una rigurosa organización y planificación, con el objetivo de lograr esto surgen las Metodologías de desarrollo. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el propósito de hacerlo más eficiente y predecible. Esto lo consiguen llevando a cabo un proceso bien detallado con un fuerte peso en la planificación inspirado por otras disciplinas de la ingeniería. Todo proceso de desarrollo de un proyecto puede presentar muchos riesgos y ser difícil de controlar; pero al utilizar una metodología de desarrollo, se conseguirá la satisfacción de los clientes y de los propios desarrolladores del producto final.

En dos grandes grupos se dividen las metodologías de desarrollo del software, las metodologías pesadas y las ágiles. Si se aplica una de las pesadas, obliga a forzar al cliente a que tome la mayoría de las decisiones al inicio del desarrollo del producto por lo que el coste de cambio de una decisión tomada puede llegar a ser muy elevado, las ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles.

La diferencia más visible entre estas es que las metodologías ágiles son menos orientadas al documento, exigiendo menor cantidad de documentación para cada tarea dada. De muchas maneras son más bien orientadas al código pues se basan en la idea de que la parte más significativa de la documentación es el código fuente.

2.2.1 Metodologías ágiles.

❖ **Programación Extrema** (por sus siglas en Inglés XP).

Se plantea que XP es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software. La Programación Extrema fue pensada por Kent Beck, como proceso de creación de software diferente del convencional. (13)

Objetivos de XP

Son muy sencillos los objetivos de XP: primero la satisfacción del cliente y segundo potenciar al máximo el trabajo en grupo. Esta metodología se propone dar al comprador el software que él necesita y en el tiempo que lo necesita, por lo que se debe responder rápidamente a las necesidades de este, incluso cuando ocurran cambios al final de ciclo de la programación. Tanto los jefes de proyecto, clientes y desarrolladores, forman parte del equipo y están implicados en el desarrollo del software. (13)

Bases de XP

La metodología XP se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Esta metodología apuesta por iteraciones cortas que generan software que el cliente puede ver. Lo que se busca en definitiva es la reducción de costes. (13)

Otra metodología ágil que puede ser analizada es la siguiente:

❖ **Desarrollo Guiado por la Funcionalidad** (por sus siglas en Inglés FDD).

Se podría considerar que FDD (por las siglas en inglés de Feature Driven Development) está en el medio entre una metodología ágil y una pesada, aunque ciertamente se asemeja más a las ligeras. Está pensada para proyectos con un tiempo de desarrollo relativamente corto, menores de un año. Está basada en un proceso iterativo y estas iteraciones son cortas de aproximadamente dos semanas que producen un software funcional, el cual el cliente y la dirección de la empresa pueden examinar. Cada iteración se define en término de funcionalidades (por eso su nombre) que son pequeñas partes del sistema con significado para el cliente. (14)

El proyecto que siga esta metodología estará fraccionado en cinco fases:

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.

- Plan de entregas sobre la base de las funcionalidades a implementar.
- Diseño basado en las funcionalidades.
- Implementación basada en las funcionalidades.

Todo el trabajo se realiza en grupo, siempre cuenta con un responsable, que habitualmente tiene mayor experiencia, que dice la última palabra en caso de el equipo no llegar a un acuerdo. (14)

Las funcionalidades de cada entrega se dividen entre los distintos subgrupos del equipo y se implementan. El código escrito (las clases) tiene propietario, o sea, solamente quien lo crea puede modificarlo, esto no ocurre en XP. Por eso es que en un subgrupo deben estar todos los propietarios de las clases implicadas, un desarrollador puede pertenecer a varios subgrupos. También son tomadas como parte del proceso de implementación, la preparación y ejecución de pruebas, las revisiones de código y la integración de las partes que componen el software. (14)

2.2.2 Metodologías pesadas.

Dentro de este tipo de metodologías de desarrollo, debe ser analizada la siguiente:

❖ RUP

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del puro análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (15)

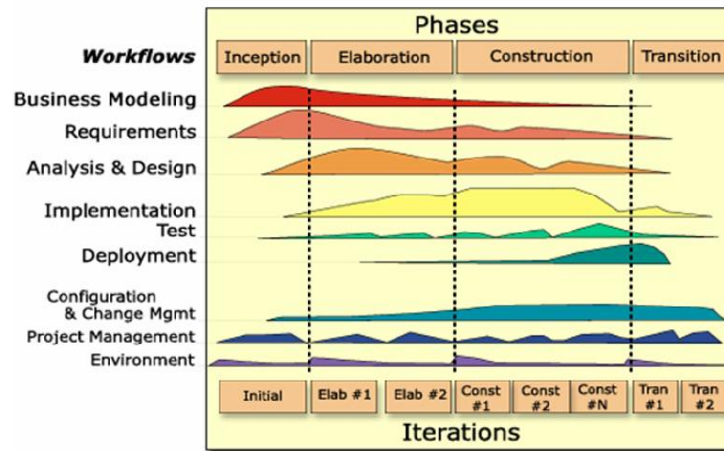


Figura 4: Fases de RUP.

Características principales de RUP

Centrado en los modelos: los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema. (15)

Guiado por los Casos de Uso: los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba. (15)

Centrado en la arquitectura: los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar. (15)

Iterativo e incremental: durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo. (15)

Beneficios que aporta RUP

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.

- Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web, pruebas y calidad. También se proporciona guías para desarrollar en plataformas IBM WebSphere y Microsoft Web Solution para acelerar el desarrollo de los proyectos.
- Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.
- No solo garantiza que los proyectos abordados serán ejecutados íntegramente sino que además evita desviaciones importantes respecto a los plazos.
- Permite una definición acertada del sistema en un inicio para hacer innecesarias las reconstrucciones parciales posteriores. (15)

2.3 Selección de la metodología a utilizar

Luego de realizar un análisis de algunas metodologías de desarrollo existentes para lograr definir cuál sería la más conveniente a utilizar según las características del presente trabajo. Se tiene que podría convenir utilizar una metodología ágil ya que el tema en cuestión es poco conocido, el cliente estará estrechamente vinculado con el equipo de trabajo, por lo que se contará con la disponibilidad del mismo.

Pero como no se va a realizar la implementación del sistema que será analizado en este trabajo, es decir, solo se llegará hasta las fases de análisis y diseño, no conviene utilizar una metodología ágil ya que estas llevan a la vez la investigación más el desarrollo y no generan la documentación suficiente

para que en el futuro a la hora de implementar el sistema exista una buena documentación con todos los artefactos bien definidos. Por tanto, es más conveniente la utilización de la metodología pesada RUP por sus características y los beneficios que esta proporciona que va más allá del puro análisis y diseño orientado a objetos generando técnicas que soportan el ciclo completo de desarrollo de software.

Además, permite una definición acertada del sistema en un inicio para hacer innecesarias las reconstrucciones parciales posteriores, evita desviaciones importantes respecto a los plazos. Por todos estos beneficios que proporciona RUP es escogido para la realización del trabajo propuesto.

2.4 Lenguaje Unificado de Modelado (UML)

Es un lenguaje para el desarrollo de software orientado a objetos, su propósito es visualizar, especificar, construir y documentar proyectos de software. (16)

Hoy día, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. (16)

En otros términos, así como en la construcción de un edificio se realizan planos previos a su construcción; en el desarrollo de software se deben realizar diseños en UML previos a la codificación de un sistema. Aunque UML es un lenguaje, éste posee más características visuales que programáticas, facilitando así la participación e intercomunicación de los integrantes de un equipo multidisciplinario en el proceso, siendo estos integrantes los analistas, diseñadores, especialistas de área y desde luego los programadores. (16)

2.4.1 Tipo de modelos UML

Estáticos:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de componentes.
- Diagrama de despliegue.

Dinámicos:

- Diagrama de estados.
- Diagrama de actividad.
- Diagrama de interacción.

Algunas de las funciones de UML son las siguientes:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (16)

Es importante destacar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, sino un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Además, describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

2.5 Herramientas CASE

Las herramientas de Ingeniería de Software Asistida por Ordenador (por sus siglas en inglés (CASE), Computer Aided Software Engineering), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. (17)

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información. Para mejorar la calidad y la

productividad de los sistemas de información a la hora de construir software se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos. (17)

De una forma esquemática se puede decir que una herramienta CASE se compone de los siguientes elementos:

- Repositorio (diccionario) donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.
- Metamodelo (no siempre visible), que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.
- Carga o descarga de datos, son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.
- Comprobación de errores, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.
- Interfaz de usuario, que consta de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías. (17)

2.5.1 Rational Rose Enterprise

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Siendo la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. (18) Rational Rose posee una serie de características fundamentales:

- Capacidad de Análisis de calidad de código.
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase. (18)

Permite a los arquitectos y diseñadores producir modelos independientes de lenguaje en Unified Modeling Language (UML) de arquitectura de software, necesidades de negocios, activos re-utilizables y comunicaciones en nivel de gerencia, posee una experiencia de diseño única, su desarrollo es dirigido por modelo con soporte de UML, permite el soporte a múltiples modelos para Model-Driven Architecture (MDA), se ejecuta tanto independiente o integrado a Microsoft Visual Studio .NET. Crea arquitecturas independientes de plataforma que se pueden implementar en plataformas Java™ y .NET. Sus estándares son definibles para usuarios para crear y personalizar estándares de diseño arquitectónico. Las referencias de modelo cruzado y de realización de versiones hasta el nivel de diagrama y clase permiten estructuración para adecuarse a cualquier proyecto. Mantiene la posibilidad de rastreo entre modelos de análisis, diseño e implementación. La modelación es de forma libre y permite publicación e informes en Web. (18)

2.5.2 Visual Paradigm

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Provee soporte para la generación de código, tiene integración con diversos Entornos de Desarrollo Integrados (por sus siglas en inglés (IDEs), Integrated Development Environment), como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate. Tiene dentro de sus características que es portable y posee gran facilidad de uso. Su diseño se centra en casos de uso y se enfoca al negocio que genera un software de mayor calidad, además tiene disponibilidad en múltiples plataformas y se puede contar con la versión libre Visual Paradigm for UML Community Edition que soporta la realización de todos los diagramas necesarios. (19)

Visual Paradigm tiene diversas características como son:

- Producto de calidad.
- Soporta aplicaciones web.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Visual Paradigm ofrece servicios tales como:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas. (19)

2.6 Selección de la herramienta CASE

Después de realizar un profundo análisis sobre las herramientas CASE se llega a la conclusión que la herramienta que se utilizará para el modelado de este trabajo es Visual Paradigm, ya que esta cuenta con características que facilitan y aseguran el trabajo a realizarse, como son: que se cuenta con una versión libre, es una herramienta multiplataforma, está orientada a la creación de diseños usando el paradigma de programación orientada a objetos y además provee soporte para la generación de código.

2.7 Conclusiones parciales

En este capítulo se realizó un análisis de algunas metodologías de desarrollo y fue seleccionada la metodología pesada RUP ya que según las características del sistema que se requiere es la más conveniente para la confección del mismo. Fue seleccionado como lenguaje de modelado el UML. Como resultado del análisis realizado a varias herramientas CASE se obtuvo la selección de Visual Paradigm.

Capítulo 3: Análisis y Diseño del Sistema

3.1 Introducción

En este capítulo, se describe la solución al problema que dio origen a la investigación, se describen los principales conceptos y se presenta el modelo de dominio del sistema en desarrollo. Son enumerados los requisitos funcionales y no funcionales que guían el desarrollo del trabajo, así como se determinan y describen los actores y casos de uso del sistema.

3.2 Modelo de Dominio

El modelo de dominio es una de las vías que brinda el Proceso Unificado para la comprensión del contexto del sistema, capturando los tipos más importantes de objetos o eventos dentro del entorno donde este se encontrará. Los objetos del dominio representan “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (15)

El modelo de dominio se describe mediante diagramas de UML, en los cuales se muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. (15)

El objetivo fundamental del modelo de dominio es comprender y describir objetos de un determinado entorno y sus principales relaciones, que en principio guiarán la construcción de un sistema específico. Posibilitando, a los involucrados en el desarrollo del producto ya sean clientes, desarrolladores e interesados, utilizar un vocabulario común el cual permita capturar correctamente los requisitos y por tanto obtener el producto deseado.

3.2.1 ¿Por qué modelo de dominio?

Se realiza el modelo de dominio debido primeramente a que no existe un negocio definido, por lo cual, no se pueden determinar los procesos y roles del proceso de negocio, haciéndose engorroso y poco exacto la descripción de los mismos. Teniendo en cuenta además de lo anteriormente expuesto que no están definidas las actividades y procesos a automatizar, que el proceso en desarrollo está altamente centrado en herramientas y tecnologías informáticas, se determina que no es viable llevar a cabo un modelo de

negocio y se hace necesario realizar la descripción de la aplicación mediante un modelo de dominio o modelo de conceptos del dominio, el cual agrupa los conceptos fundamentales de objetos relacionados con el dominio, sus atributos y sus relaciones.

3.2.2 Conceptos del modelo de dominio

Terminal: hace función de un usuario que va a interactuar con la EPG, ya sea mediante un STB o una computadora.

EPG: es quien le muestra al usuario toda la información que este desea e interactúa con el mediador para solicitar todos los flujos de datos.

Servidor VoD: contiene toda la información disponible para satisfacer las solicitudes del servicio de VoD.

Head End: contiene la información disponible para satisfacer las solicitudes de canales que se transmiten en vivo.

DRMCAS: contiene la información de los terminales que forman parte del sistema.

Controlador: es la clase del mediador con la cual interactúa la EPG para hacer solicitudes y esta clase controlador delega las tareas a las distintas clases gestoras contenidas en el mediador.

Gestor de Servicios: gestiona la información que le solicita controlador, interactuando con las clases: Servidor VoD, Head End y DRMCAS.

Gestor de EPG: gestiona la información que le solicita la clase controlador, pidiéndole los parámetros a la clase Gestor de Servicios.

3.2.3 Diagrama del Modelo del Dominio.

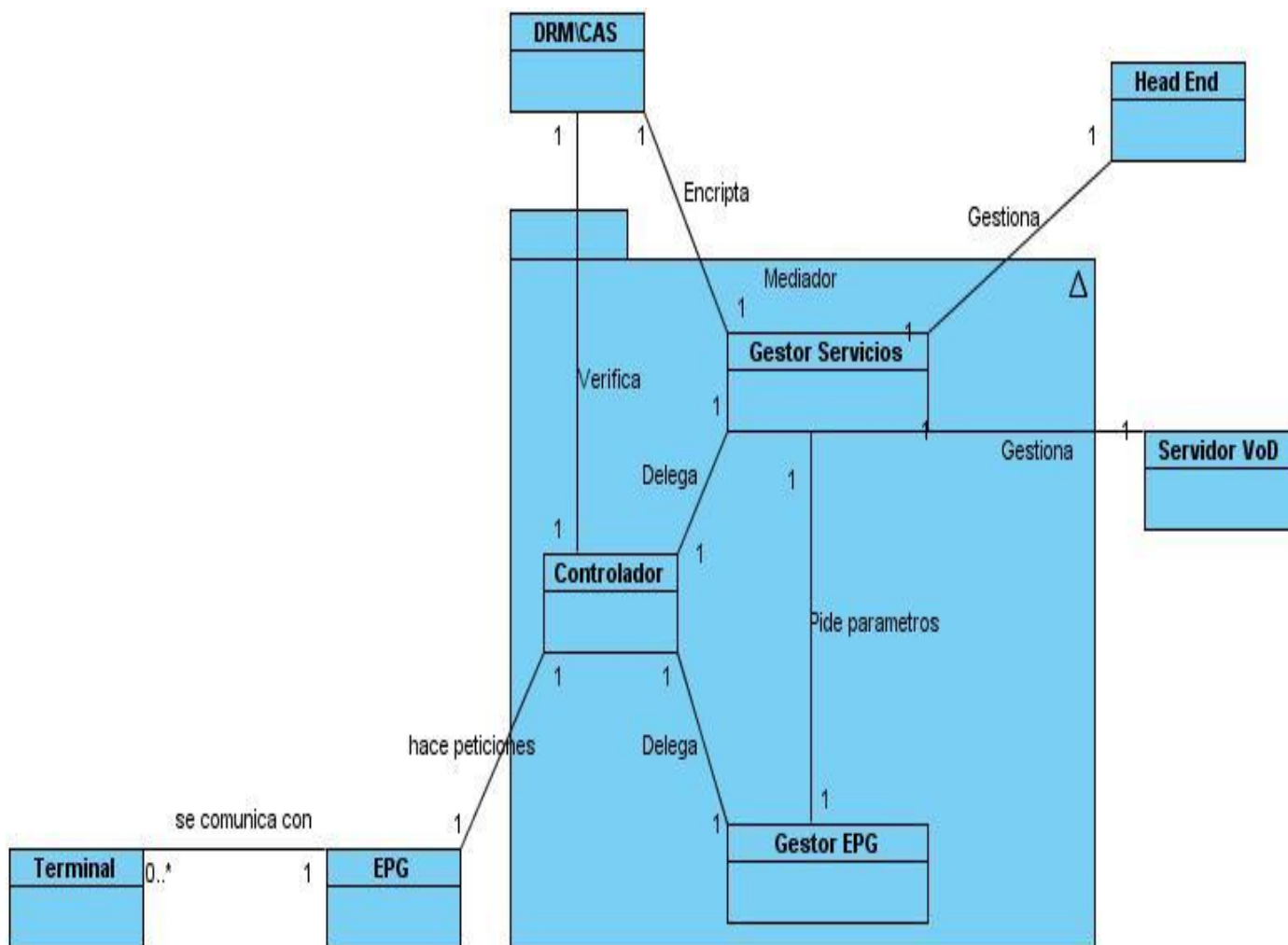


Figura 5: Modelo de Dominio

3.2.4 Descripción del Modelo de Dominio

El Terminal se comunica con la EPG, entonces esta es quien hace las peticiones al Mediator de forma general. El Mediator está compuesto por Controlador, Gestor de Servicios, Gestor de EPG. Las clases con quien interactuará el mediador para gestionar los servicios son: DRM\CAS, Head End y Servidor VoD. La EPG interactúa directamente con Controlador, si hace una solicitud de verificar un usuario, Controlador

verifica esto con DRM/CAS Sistema de Acceso Condicional (por sus siglas en inglés CAS), este le devuelve la información a controlador y seguidamente controlador a EPG. Una vez que el usuario ya está autenticado, la EPG comprueba si tiene EPGs personalizadas (ocurre internamente en la EPG), si no tiene le debe enviar una solicitud a Controlador, el mismo delega al gestor de EPG, entonces Gestor de EPG solicita todos los parámetros al Gestor de servicios y él se los devuelve, el Gestor de EPG crea la EPG de acuerdo a sus parámetros, se la envía controlador para que sea devuelta a la EPG y esta actualiza la vista del terminal con la EPG gestionada.

La EPG le solicita un canal determinado que se transmite a Controlador, este interactúa con el Gestor de Servicios el cual gestiona el flujo solicitado mediante el subsistema Head End, cuando ya tiene el flujo del canal lo encripta mediante el DRM y luego se lo devuelve a la EPG.

La EPG solicita a Controlador un servicio de VoD, luego este interactúa con Gestor de servicios para que gestione el flujo pedido con el Servidor VoD, cuando Controlador ya tiene la información solicitada, entonces mediante el DRM lo encripta y luego lo devuelve a la EPG.

3.3 Requisitos Funcionales

Los requisitos funcionales describen las funciones que el software debe ejecutar, son conocidos en ocasiones como capacidades.

Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Estos requisitos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema. A continuación los mismos.

R1. Comprobar acceso del Terminal:

El subsistema debe atender a la solicitud de la EPG de verificar si un terminal puede entrar al sistema, haciendo esto apoyándose en la información contenida en el DRM/CAS.

R2. Agregar canal:

El subsistema debe permitir que se asigne un determinado canal cuando este haya sido solicitado al Mediador mediante la EPG.

R3. Gestionar Archivo Multimedia para Video bajo Demanda:

El subsistema debe permitir que mediante una solicitud de la EPG le sea asignado un archivo multimedia para el servicio de VoD.

R4. Gestionar EPG:

El subsistema debe permitir que se gestione una EPG personalizada de acuerdo con los parámetros solicitados.

3.4 Requisitos no Funcionales.

Los requisitos no funcionales, son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto agradable, usable, rápido o confiable. Seguidamente son definidos:

RNF1 Rendimiento

El sistema debe soportar al menos 1000 conexiones concurrentes para garantizar el servicio de Video bajo Demanda, el tiempo de respuesta está dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información. El servidor debe estar directamente conectado al backbone (red principal de la universidad), para evitar que el ancho de banda necesario para la transmisión de video se vea afectado por otros servicios.

RNF2 Seguridad

Solo podrán solicitarle servicios al sistemas los terminales que estén registrados, por ello lo primero que se realiza es verificar si el terminal es usuario del sistema, además este debe ser del dominio uci.cu. Por tanto, ningún otro usuario que no forme parte del sistema podrá acceder a este.

RNF3 Soporte

El sistema deberá ser objeto de una revisión y mantenimiento cada 2 meses como mínimo para garantizar su correcto funcionamiento de forma continua. En el futuro se le podrán agregar nuevas funcionalidades que gestionen más servicios y aumente la calidad del servicio.

RNF4 Requerimientos de Hardware

Para las PCs clientes:

- Tarjeta de red.
- Mínima memoria RAM de 256 MB.
- Mínimo disco duro de 100 MB.
- Procesador a frecuencia de 1024 MHz como mínimo.

Para los servidores:

- Tarjeta de red.
- Tarjeta de video y de captura.
- El Servidor de BD con 2GB de RAM y 80GB de disco duro como mínimo.
- Procesador a frecuencia de 3 GHz como mínimo.

Para los Televisores:

- Un Set Top Box para cada televisor.

RNF5 Portabilidad

El sistema será multiplataforma y su puesta en marcha debe ser compatible con los sistemas operativos Windows y Linux.

RNF6 Legales

La mayoría de las herramientas utilizadas para el desarrollo se encuentran bajo estándares y plataformas libres. Como producto, los componentes del Mediador para un sistema IPTV se distribuyen amparados bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

3.5 Modelo de Casos de Usos del Sistema

Un caso de uso es un pedazo o fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante, los casos de uso representan los requisitos funcionales y guían el diseño, la implementación y pruebas en el proceso de desarrollo.

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. El modelo de casos de uso permite que los desarrolladores y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. (20) El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada usuario se representa mediante uno o más actores.

Los actores son el entorno del subsistema, comunicándose con este mediante el envío y recepción de mensajes. Por su parte los casos de uso especifican el subsistema y están diseñados para cumplir los deseos del usuario cuando utiliza el subsistema. (20)

3.5.1 Actores del Sistema

Los actores representan terceros fuera del sistema que colaboran con el sistema. Al identificar los actores del sistema se identifica el entorno externo del sistema.

| Actor | Descripción |
|-------|--|
| EPG | Es quien interactúa con el Mediador, inicializa todos los Casos de Uso |

3.5.2. Casos de Uso del Sistema

Los casos de uso son el componente clave del modelado. Su propósito es ilustrar como un sistema permite a un actor cumplir una meta, ilustrando todos los posibles caminos apropiados que ellos pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar. Los casos de uso son los siguientes:

| | |
|-------------|---|
| CU-1 | Verificar Terminal |
| Actor | EPG |
| Descripción | La EPG le envía al Mediador los datos del terminal que desea entrar al sistema, y este verifica con el DRM/CAS si este puede entrar al sistema. |
| Referencias | RF 1 |

| | |
|-------------|---|
| CU-2 | Gestionar Canal |
| Actor | EPG |
| Descripción | La EPG le solicita al Mediador un determinado canal que se transmite en vivo para que este sea visualizado. |
| Referencias | RF 2 |

| | |
|-------------|---|
| CU-3 | Gestionar Archivo VoD |
| Actor | EPG |
| Descripción | La EPG le solicita al Mediador un determinado archivo del servidor de VoD y este le gestiona y proporciona este servicio. |

| | |
|-------------|------|
| Referencias | RF 3 |
|-------------|------|

| | |
|-------------|---|
| CU-4 | Gestionar EPG |
| Actor | EPG |
| Descripción | La EPG le solicita al Mediador que gestione un determinado flujo de datos con los cuales se construirá un EPG personalizada, la cual modificará la EPG que existía anteriormente y será mostrada al Terminal. |
| Referencias | RF 4 |

3.6 Diagrama de Casos de Uso del Sistema

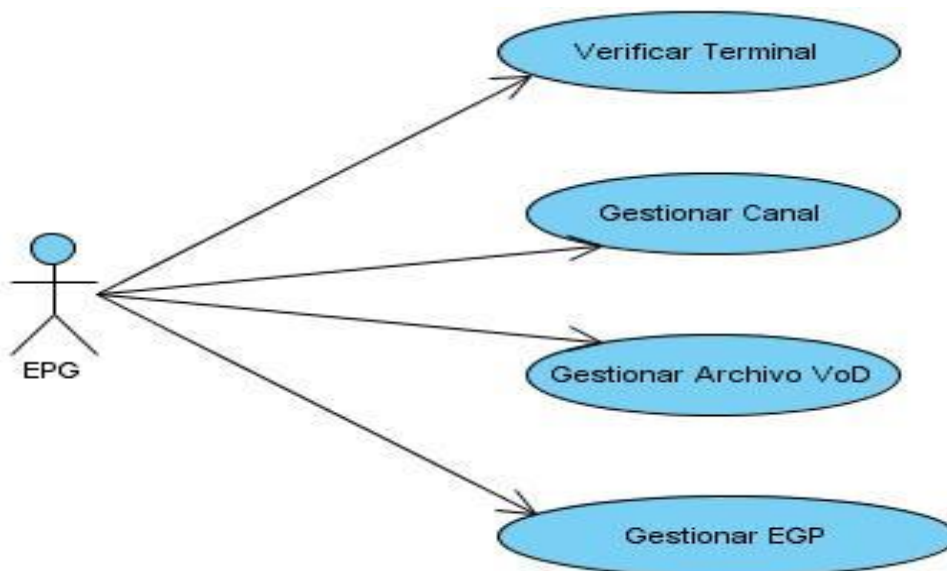


Figura 6: Diagrama de Casos de Uso del Sistema.

3.7 Descripción de los Casos de Uso del Sistema

| | | |
|--|--|--|
| CU-1: | Verificar Terminal | |
| Actor: | (EPG) | |
| Propósito: | Verificar los datos del terminal enviados por la EPG para comprobar si tiene acceso al sistema. | |
| Resumen: | El Caso de Uso se inicia cuando la EPG envía los datos del terminal al Mediador para comprobar el acceso de este al sistema y termina cuando se verifican los datos con el DRM/CAS y se le devuelve la información a la EPG. | |
| Referencia: | RF 1 | |
| Precondición: | Existan los datos del terminal en el DRM/CAS | |
| Poscondición: | | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. La EPG envía los datos que son password y dirección IP. | 2. El sistema verifica si los datos están registrados en el DRM. | |
| | 3. El sistema asigna los privilegios del Terminal y los envía a la EPG. Se finaliza el caso de uso. | |
| Flujo Alterno de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |

| | |
|------------------|--|
| | 2.1 Si los datos no se encuentran registrados da por terminada la sesión del terminal. |
| Prioridad | Crítico |

| | |
|----------------------|---|
| CU-2: | Gestionar Canal |
| Actor: | (EPG) |
| Propósito: | Obtener un canal que se transmite en vivo para que sea visualizado por la EPG. |
| Resumen: | El caso de uso se inicia cuando la EPG le solicita al Mediador un canal que se transmite en vivo para luego esta visualizarlo. Lo solicita mediante el parámetro (idCanal). |
| Referencia: | RF 2 |
| Precondición: | Que existan canales en el subsistema Head End. |
| Poscondición: | Que se gestione el canal para su visualización. |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. La EPG solicita un canal que se transmite en vivo mediante el parámetro (idCanal). | 2. El sistema solicita el (idCanal) demandado en el subsistema Head End. |
| | 3. El sistema asigna el (idCanal) para que sea visualizado. Termina el caso de uso. |

| | |
|-----------|----------------|
| Prioridad | Crítico |
|-----------|----------------|

| | |
|----------------------|--|
| CU-3: | Gestionar Archivo VoD |
| Actor: | (EPG) |
| Propósito: | Obtener un archivo del servidor de VoD para ser visualizado de esta forma. |
| Resumen: | El caso de uso se inicia cuando la EPG le solicita al Mediador un archivo del servicio de VoD para que este sea visualizado. La solicitud se hace mediante el parámetro (idVoD). |
| Referencia: | RF 3 |
| Precondición: | Que exista el archivo solicitado en el servidor de VoD. |
| Poscondición: | Que se obtenga el archivo para su visualización. |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. La EPG solicita un archivo de VoD. Mediante el parámetro (idVoD) | 2. El sistema solicita el archivo demandado en el subsistema Servidor de VoD. |
| | 3. El sistema asigna el archivo para que sea visualizado. Termina el caso de uso. |
| Prioridad | Crítico |

| | |
|----------------------|--|
| CU-4: | Gestionar EPG |
| Actor: | (EPG) |
| Propósito: | Obtener los datos solicitados para conformar la EPG personalizada. |
| Resumen: | El caso de uso se inicia cuando la EPG le solicita al Mediador un flujo de datos con los cuales se construirá una EPG personalizada, la cual modificará la EPG que existía anteriormente y será mostrada al Terminal. Los datos solicitados serán mediante los parámetros (idCanal) e (idVoD). |
| Referencia: | RF 4 |
| Precondición: | Que no exista la EPG personalizada para quien la requiere. |
| Poscondición: | Que se obtenga el flujo de datos requerido para conformar la EPG personalizada. |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1. La EPG hace la solicitud de (idCanal) e (idVoD), estos son los parámetros que conformarán la EPG Personalizada. | 2. El sistema solicita los parámetros demandados a Gestor de EPG. |
| | 3. El sistema devuelve la nueva EPG personalizada con los parámetros solicitados. Termina el caso de uso. |
| Prioridad | Crítico |

3.8 Análisis.

Las actividades del análisis son desarrolladas con el objetivo de facilitar la entrada al diseño, por lo que son un paso inicial y una primera aproximación conceptual para aumentar el nivel de especificidad en aras de garantizar el cubrimiento de los requisitos funcionales y obtener una visión de qué hace el sistema.

(20) El modelo de análisis está compuesto por artefactos como: clases del análisis y realizaciones de casos de uso del análisis. Las clases del análisis van a representar abstracciones de conceptos, en las cuales deben incluirse atributos y operaciones a un nivel alto. La realización de casos de uso del análisis describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos en interacción. Para ello RUP propone que para cada caso de uso:

- Se construya un diagrama de clases que muestre las clases de análisis participantes.
- Se construya alguno de los tipos de diagrama de interacción para representar el flujo en términos de objetos del análisis. (20)

3.8.1 Diagramas de Clases del Análisis

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación-composición, generalización-especialización y tipos asociativos. RUP propone clasificar a las clases en: interfaz, control y entidad. (20)

A continuación se muestran los diagramas de clases del análisis, correspondiente a los casos de uso del sistema.

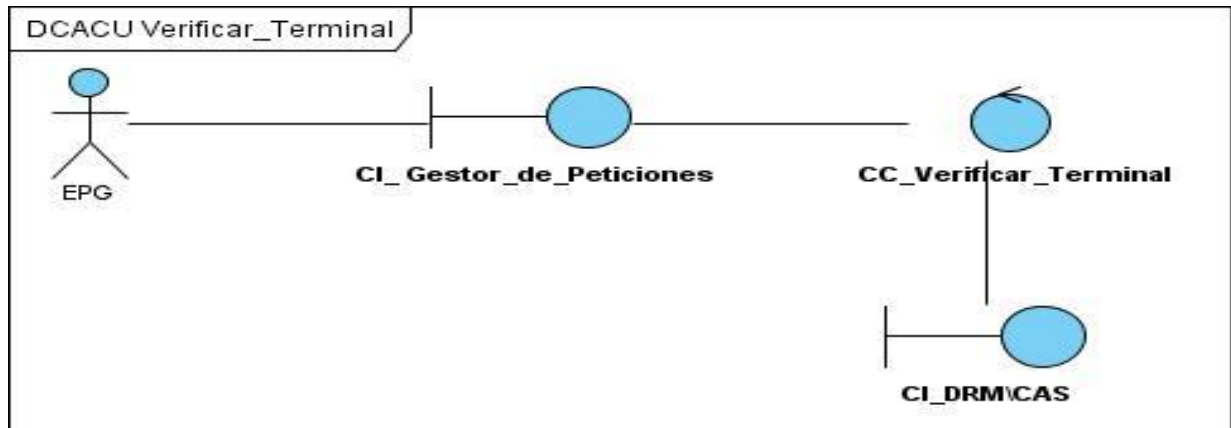


Figura 7: Diagrama de Clases del Análisis del Caso de Uso Verificar Terminal.

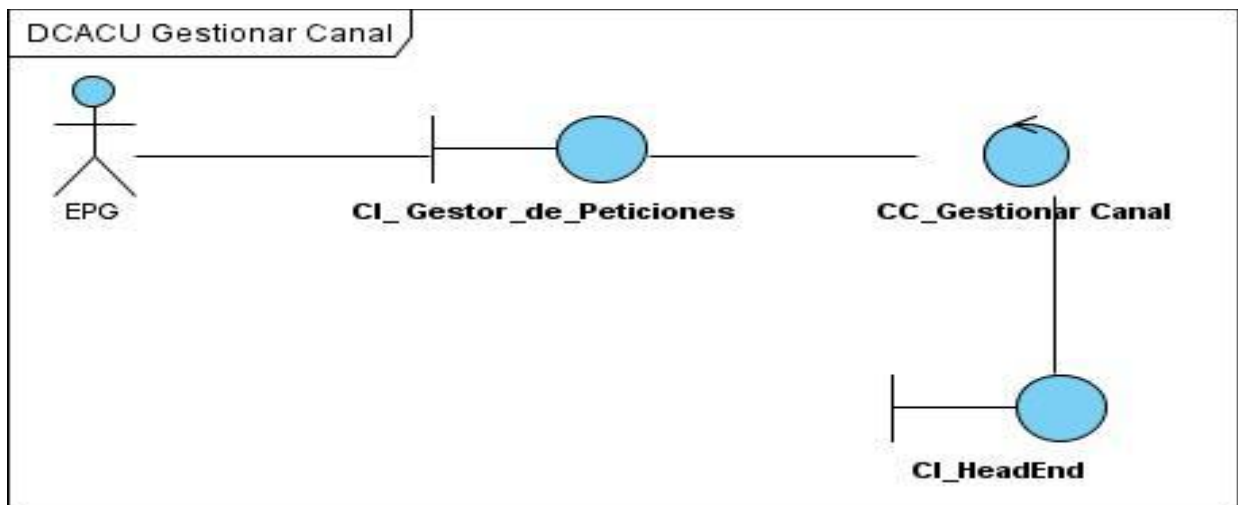


Figura 8: Diagrama de Clases del Análisis del Caso de Uso Gestionar Canal.

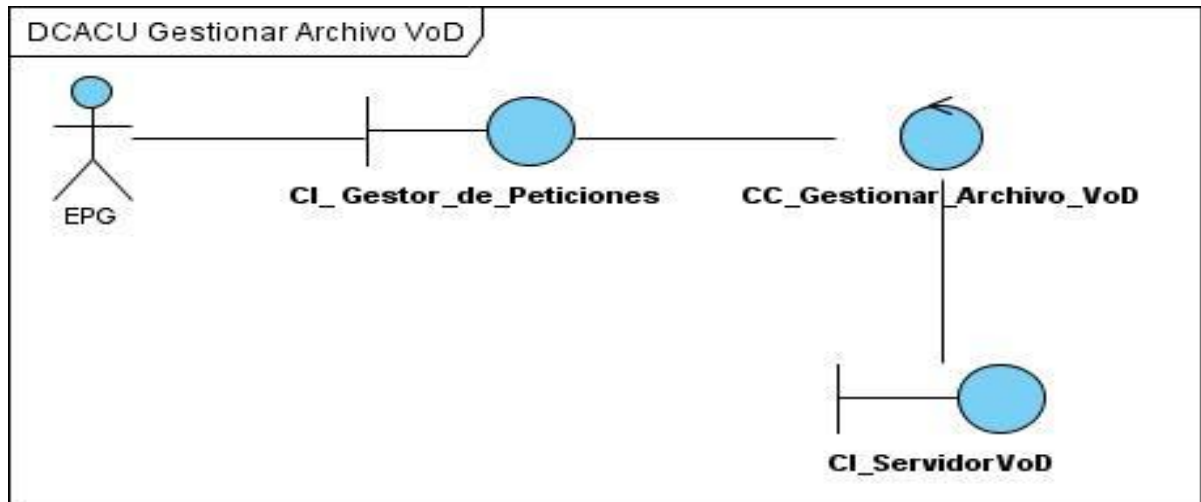


Figura 9: Diagrama de Clases del Análisis del Caso de Uso Gestionar Archivo VoD.

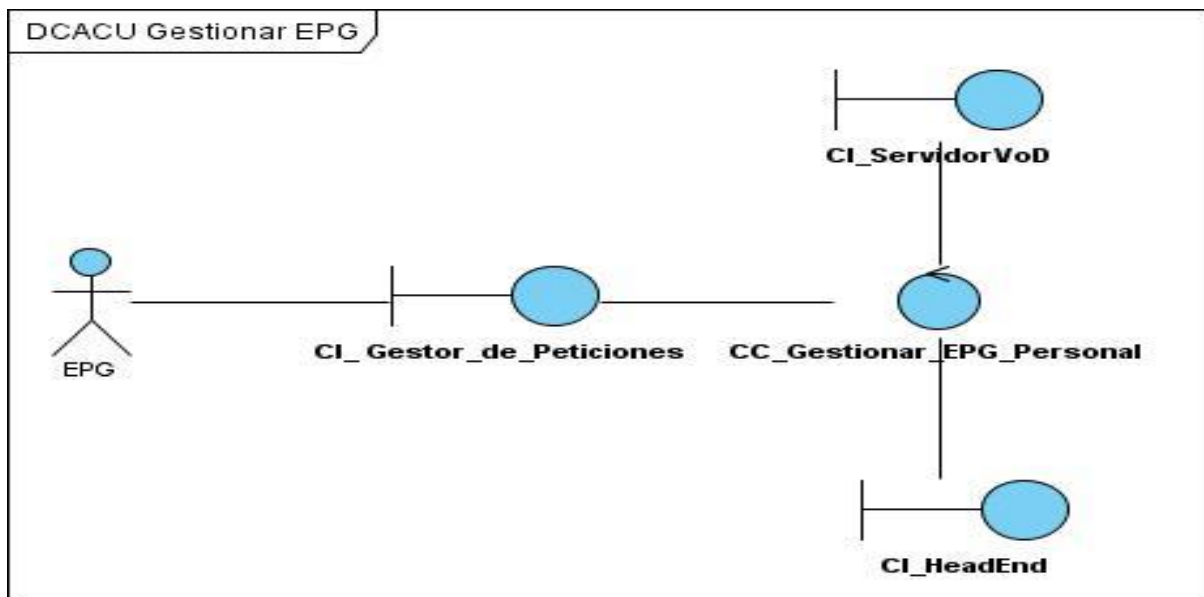


Figura 10: Diagrama de Clases del Análisis del Caso de Uso Gestionar EPG.

3.8.1 Diagramas de Colaboración para cada Diagrama de Clases del Análisis.

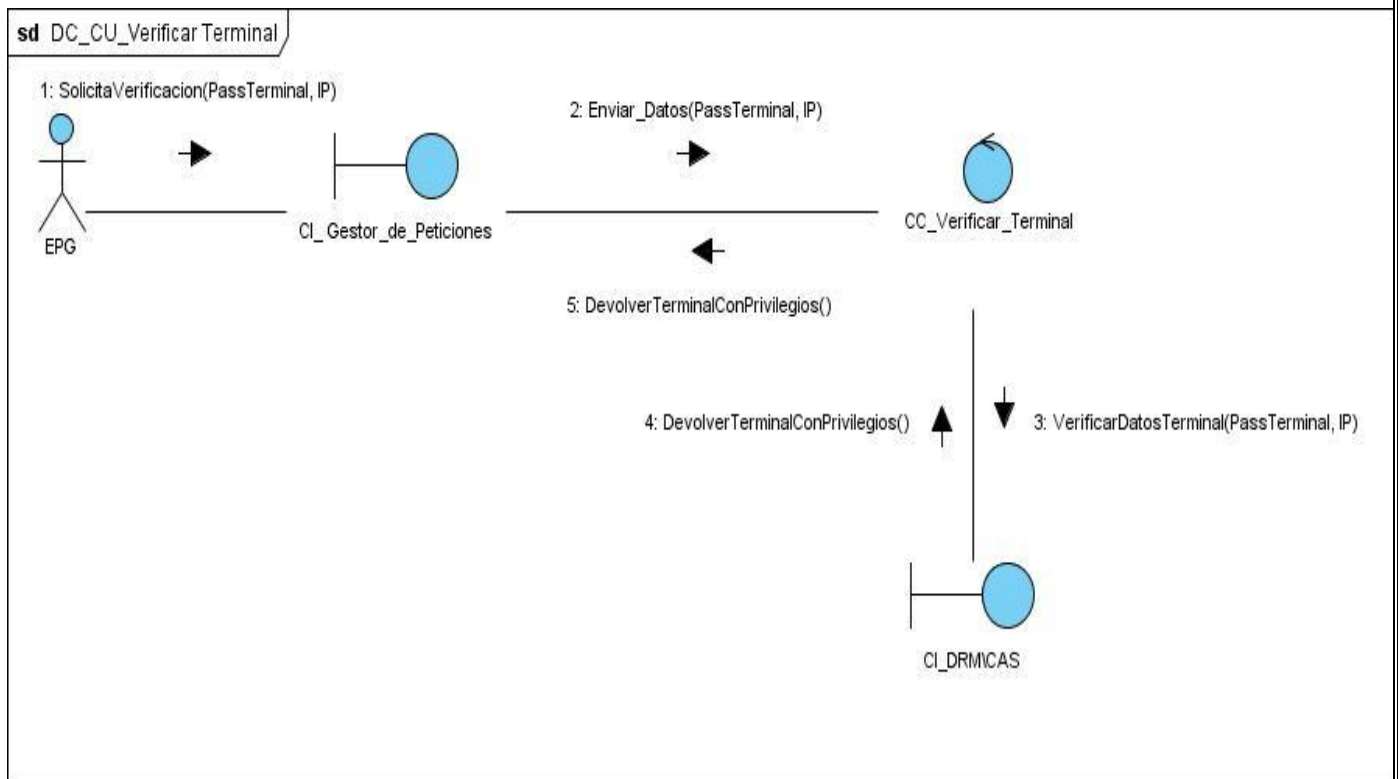


Figura 11: Diagrama de Colaboración del Caso de Uso Verificar Terminal.

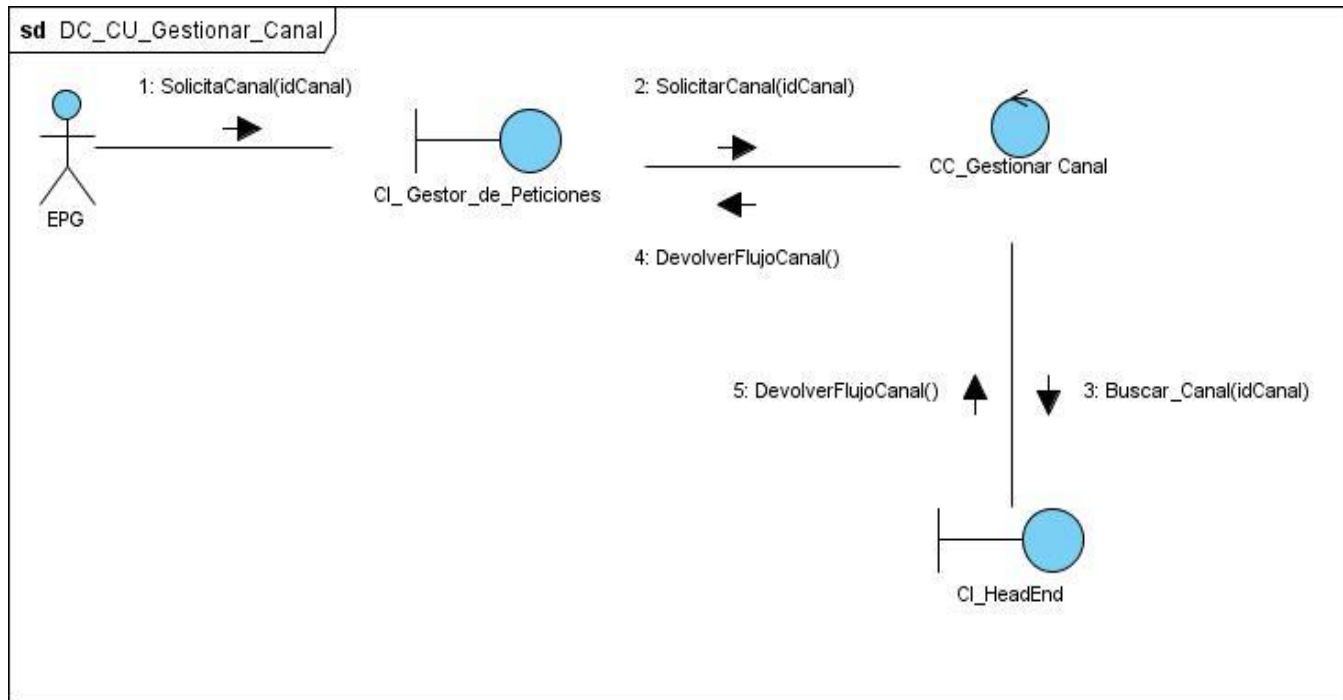


Figura 12: Diagrama de Colaboración del Caso de Uso Gestionar Canal.

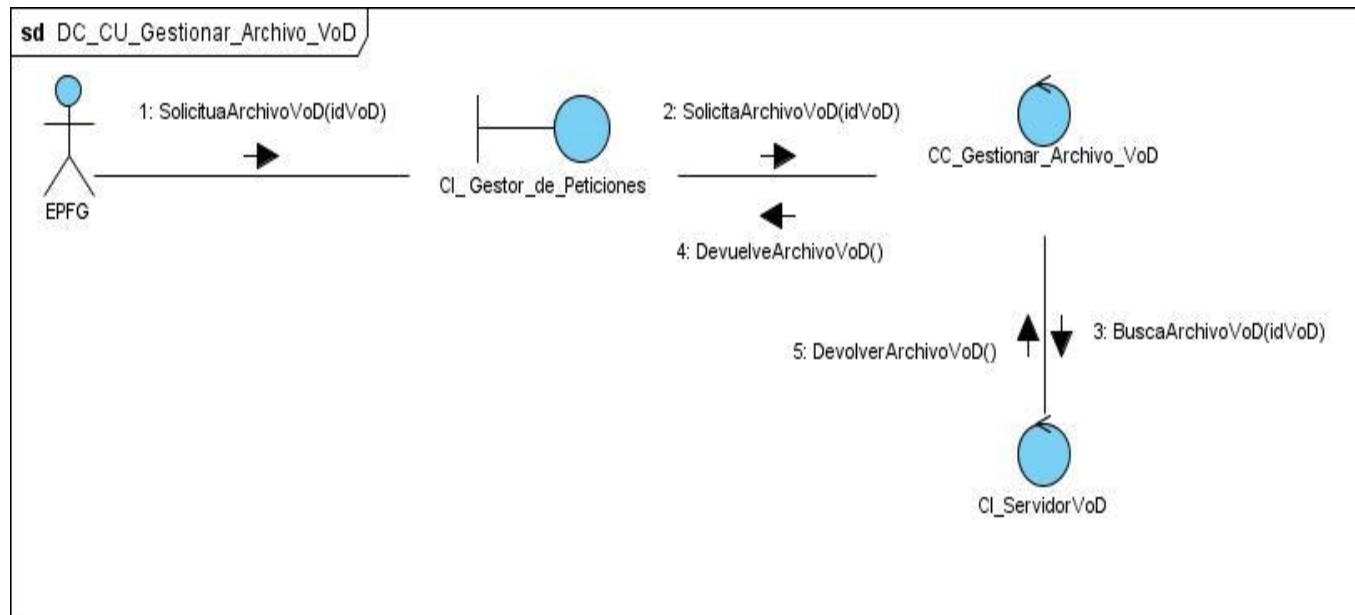


Figura 13: Diagrama de Colaboración del Caso de Uso Gestionar Archivo VoD.

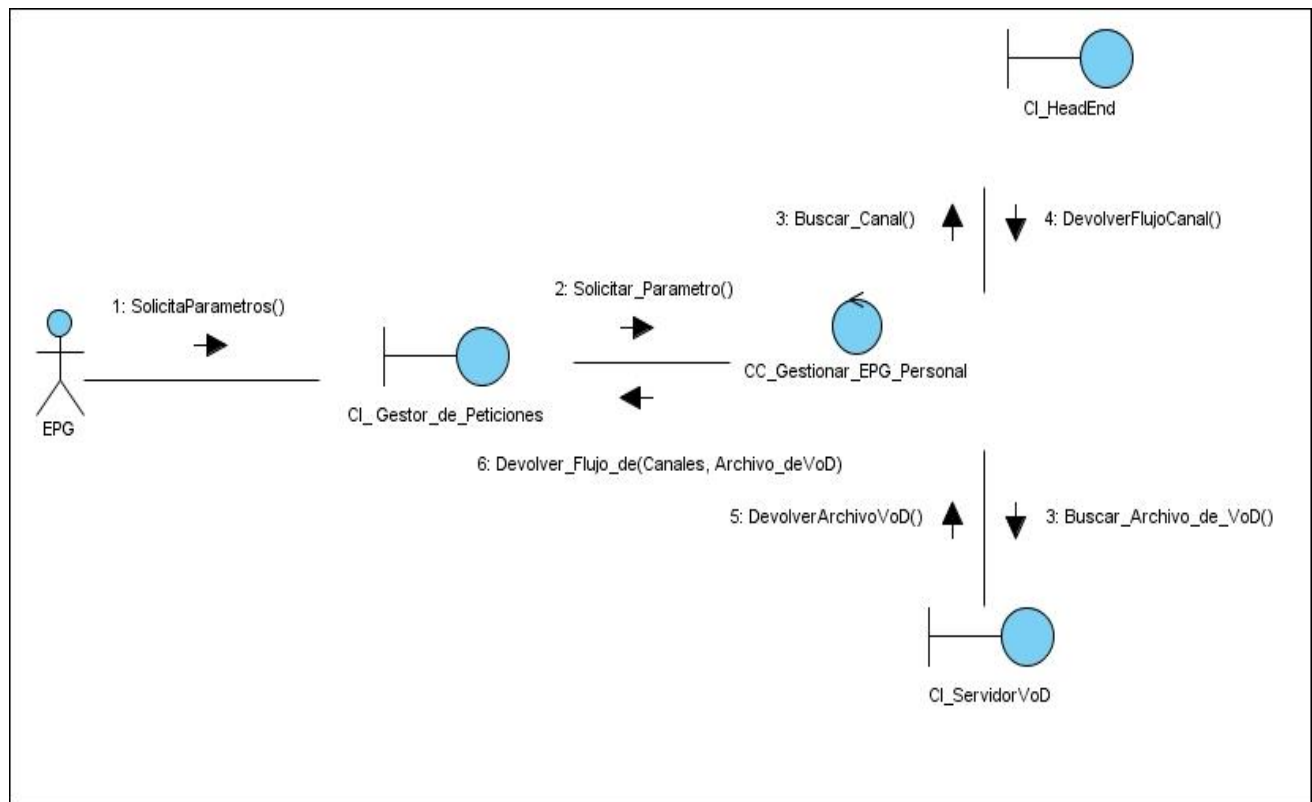


Figura 14: Diagrama de Colaboración del Caso de Uso Gestionar EPG.

3.9 Arquitectura del Software

Siguiendo la definición que ofrece el Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers (IEEE), por sus siglas en Inglés), la Arquitectura de Software es la organización fundamental de un sistema representado por sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución.

Durante el desarrollo de un software es común encontrar diferentes puntos de vistas entre los equipos de trabajo que asumirán la concepción y construcción del sistema, la arquitectura es el instrumento encargado para unificar de manera sencilla estas diferencias. Proporciona una visión global del sistema contribuyendo a lograr un desarrollo armónico e incremental que deberá prevalecer durante el ciclo de

vida del proyecto, sirviendo de guía para evitar diferentes riesgos asociados con la implementación de la aplicación, que de convertirse en errores encarecerían el costo del producto final.

En los primeros años del siglo XXI se pueden observar diferentes metodologías de desarrollo de software, por lo que se hace complicado definir una visión unificada sobre el diseño arquitectónico. Vale destacar sobre este aspecto que sin importar el carácter de la metodología usada siempre existirá una fase donde se implantará la arquitectura base y por otra parte se podrá observar la dependencia existente entre la arquitectura y los casos de usos. En la presente investigación se utiliza RUP como metodología de desarrollo para la conceptualización y diseño del sistema, por lo que la arquitectura se establece en la fase de elaboración, constituyendo el hito de la misma, como se puede observar en la siguiente imagen.

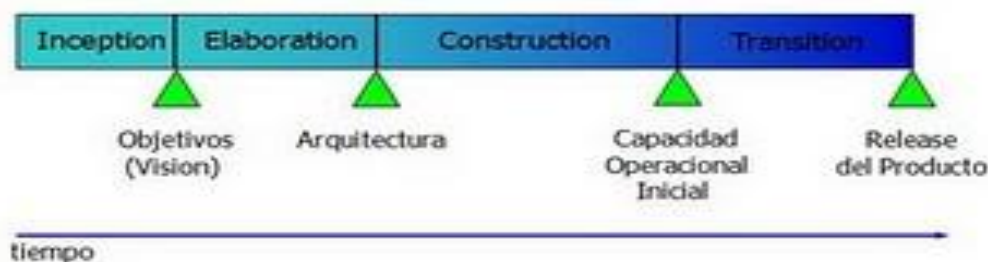


Figura 15: Fases e Hitos en RUP.

3.9.1 Patrones.

Según G. Booch: “Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles.” (21) Los patrones están basados en la experiencia acumulada por los desarrolladores, con el objetivo de describir la solución más factible a problemas comunes, lo cual ayudará a no cometer errores consumados con anterioridad.

Se pueden agrupar en diferentes categorías según el nivel de abstracción:

Patrones arquitecturales: aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.

Patrones de diseño: aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software. Idiomas: patrones de bajo nivel, específicos para un lenguaje de programación o entorno concreto.

3.9.2 Patrones de Arquitectura.

Entre los patrones de arquitectura más conocidos se encuentran el Modelo Vista Controlador (MVC), Arquitectura Orientada a Objetos (AOO) y Arquitectura en Capas. Pero en ocasiones de acuerdo con la funcionalidad del sistema que se desea desarrollar resulta difícil identificar cual es el idóneo para lograr alcanzar las metas propuestas.

Seguidamente se explica en qué consisten algunos de estos patrones de arquitectura.

Patrón Modelo-Vista-Controlador (MVC)

Es un patrón de AS que separa los datos, interfaz de usuario, y la lógica de control de una aplicación en tres componentes distintos.

Modelo: Es la representación específica de los datos e información con que el sistema interactúa.

Vista: Es la encargada de mostrar la información e interactuar con el usuario.

Controlador: Responde a eventos, usualmente acciones de usuario e invoca cambios en el modelo y probablemente en la vista.

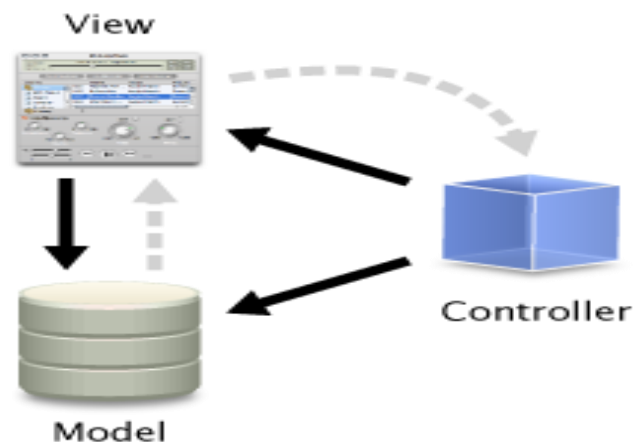


Figura 16: Patrón Modelo Vista Controlador. (22)

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual.

Desde el punto de vista de la complejidad este patrón introduce nuevos niveles de dirección por lo que aumenta ligeramente la complejidad de la solución. Como consecuencia de las frecuentes

actualizaciones, el hecho de desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas. Si el modelo experimenta cambios frecuentes, por ejemplo, podrían desbordar las vistas con una amplia actualización de requerimientos. (22)

Arquitecturas Orientadas a Objetos (AOO)

Los componentes de este estilo son los objetos, o más bien instancias de los tipos de datos abstractos. En la caracterización clásica de David Garlan y Mary Shaw, los objetos representan una clase de componentes que ellos llaman managers, debido a que son responsables de preservar la integridad de su propia representación. Un rasgo importante de este aspecto es que la representación interna de un objeto no es accesible desde otros objetos. En la semblanza de estos autores curiosamente no se establece como cuestión definitoria el principio de herencia. Ellos piensan que, a pesar de que la relación de herencia es un mecanismo organizador importante para definir los tipos de objeto en un sistema concreto, ella no posee una función arquitectónica directa. En particular, en dicha concepción la relación de herencia no puede concebirse como un conector, puesto que no define la interacción entre los componentes de un sistema. Además, en un escenario arquitectónico la herencia de propiedades no se restringe a los tipos de objeto, sino que puede incluir conectores e incluso estilos arquitectónicos enteros. (23)

Se puede decir que en las AOO los componentes del estilo se basan en principios orientados a objetos donde se puede realizar encapsulamiento, herencia y polimorfismo. Las unidades de modelado, diseño e implementación, los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. En cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases. En tantos componentes, los objetos interactúan a través de invocaciones de funciones y procedimientos.

Patrón en Capas

El patrón en capas es definido como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (24)

En un patrón en capas, los conectores se definen mediante los protocolos que determinan las formas de la interacción. Los diagramas de sistemas clásicos dibujan las capas en adyacencia, sin conectores, flechas ni interfaces; en algunos casos se suele representar la naturaleza jerárquica del sistema en forma de círculos concéntricos. Las restricciones topológicas del patrón pueden incluir una limitación, más o menos rigurosa, que exige a cada capa operar sólo con capas adyacentes, y a los elementos de una capa entenderse sólo con otros elementos de la misma; se supone que si esta exigencia se relaja, el patrón deja de ser puro y pierde algo de su capacidad heurística. (25)

El objetivo primordial del patrón en capas es dividir, fraccionar y llegar a separar la Presentación (donde muestras y obtienes datos), de la Lógica de Negocio (donde realizas operaciones) y con todo esto se obtendría independencia, por lo que si hay cambios de arquitectura se puede acceder a los datos o cambiar el negocio o modificar la presentación y solo sería en esa fracción de la capa.

La Capa de la Presentación: esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús y gráficos en general. La Capa de **Lógica de Negocio:** esta capa reúne todos los aspectos del software que tienen que automatizar o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. Esta capa también recibe el nombre de la capa de la Lógica de la Aplicación.

La Capa de Acceso a Datos: la capa de Acceso a Datos es el puente entre la capa de Lógica de Negocio y el sistema de base de datos. Encapsula la lógica de acceso a datos. Aquí se encuentran componentes que hacen transparente el acceso a la base de datos. Este es el lugar idóneo para implementar los objetos de acceso a datos, permitiendo los mismos ingresar, obtener, actualizar y eliminar información del sistema de bases de datos.

Capa de Seguridad: esta es una capa especial y su objetivo es lograr que el sistema alcance una seguridad total. Esta capa está relacionada con la administración del sistema porque es aquí donde se definen los permisos que puede tener cada usuario para acceder a la aplicación. Además, se definen las variables de sesión para cada uno de los usuarios. (25)

La arquitectura en capas define el patrón en capas como una organización jerárquica, lo que posibilita un diseño basado en niveles de abstracción creciente, posibilitando a los implementadores, fraccionar un

problema en una secuencia de pasos incrementales. Este estilo de desarrollo en varios niveles, facilita que en caso que ocurra algún cambio, sólo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles. Los sistemas o arquitecturas en capas constituyen uno de los patrones que aparecen con mayor frecuencia.

Para la construcción del módulo a diseñar se decidió utilizar el patrón en capas, pero solo existirá 1 tier (1 nivel). Este tipo de diseño es monolítico, no presenta interfaz, y es comúnmente utilizado en sistemas heredados, se define la utilización de una sola capa en la que se encontrarán todas las clases que conformaran el diseño. Se propone esta arquitectura porque no se cuenta con una interfaz gráfica, ni con acceso a datos, además brinda como ventaja que se consiga una independencia, es decir, que cuando este módulo sea integrado al sistema IPTV general, se puede actualizar o modificar este nivel sin afectar la aplicación en su conjunto.

3.10 Diseño.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea, cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. En el diseño se modela el sistema y se define una arquitectura que soporte todos los requisitos. Concretamente se puede definir como propósitos del diseño:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

3.10.1 Patrones de Diseño

Para la realización del diseño del Mediador, se utilizaron los Patrones Generales de Software para Asignación de Responsabilidades (GRASP), por sus siglas en inglés (Responsability Assignment Software

Patterns). Se considera que más que patrones son una serie de "Buenas Prácticas" de aplicación, recomendables en el diseño de software (26). Entre los patrones GRASP se encuentran los siguientes:

Alta cohesión: propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, soporta mayor capacidad de reutilización.

Bajo acoplamiento: es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Este patrón se pone de manifiesto en la mayoría de las clases porque cada una es capaz de realizar sus responsabilidades sin la utilización de las demás.

Alta cohesión y bajo acoplamiento: pueden aparecer separados, aunque están íntimamente ligados, de hecho, si se aumenta mucho la cohesión del sistema software, se tiene un alto acoplamiento entre las clases, y por el contrario si reduce mucho el acoplamiento, se verá mermada la cohesión.

Experto: el GRASP de Experto en información es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo, lo cual permite que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

Creador: ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que: tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase. Este patrón brinda soporte de bajo acoplamiento, lo cual supone menos dependencias entre clases.

Controlador: este patrón funciona como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos y la que los envía a las distintas clases según el método llamado. (26)

3.11 Diagramas de Clases del Diseño para cada Caso de Uso.

Las siguientes imágenes muestran los diagramas de clases del diseño para cada caso de uso del sistema:

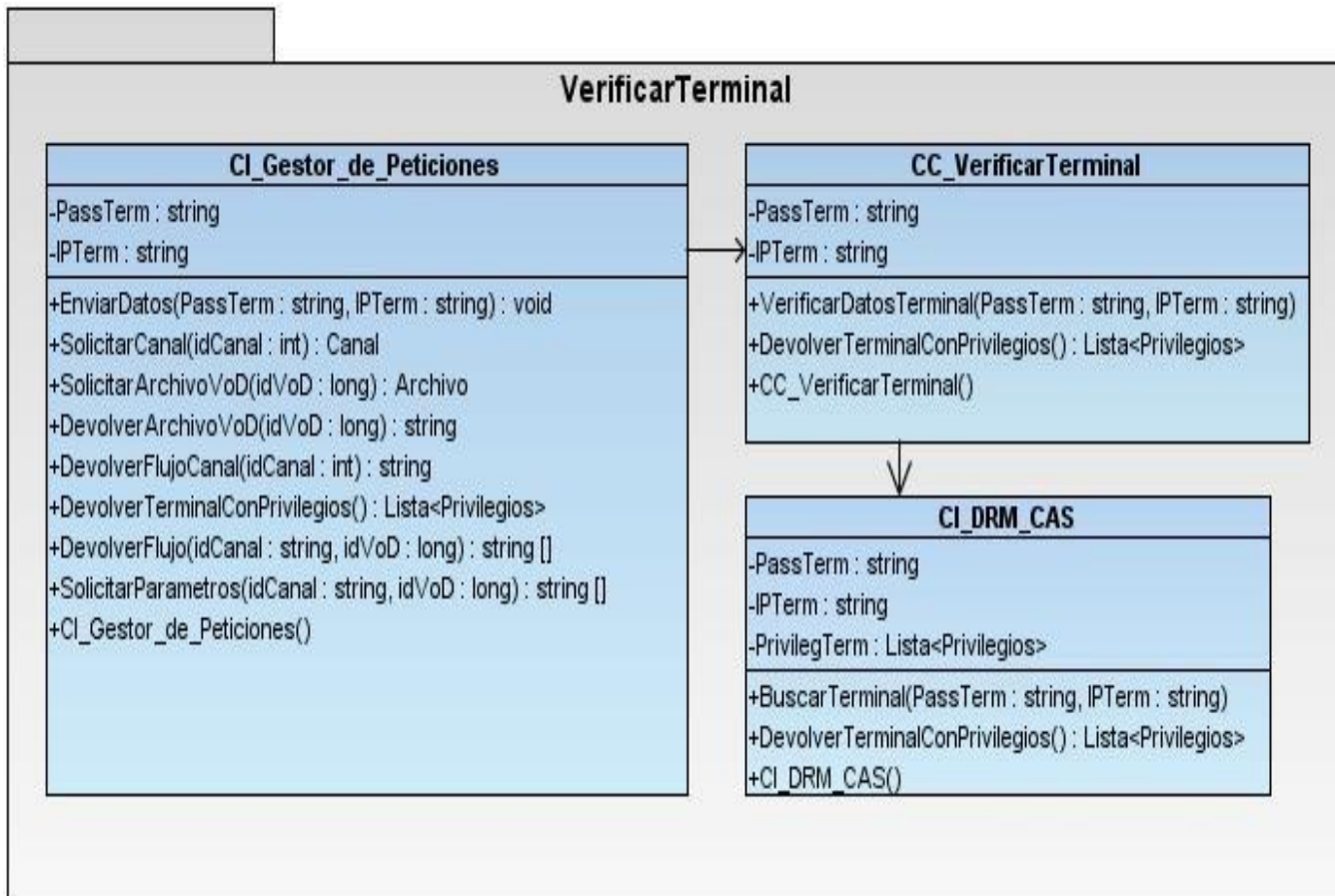


Figura 17: Diagrama de Clases del Diseño del Caso de Uso Verificar Terminal.

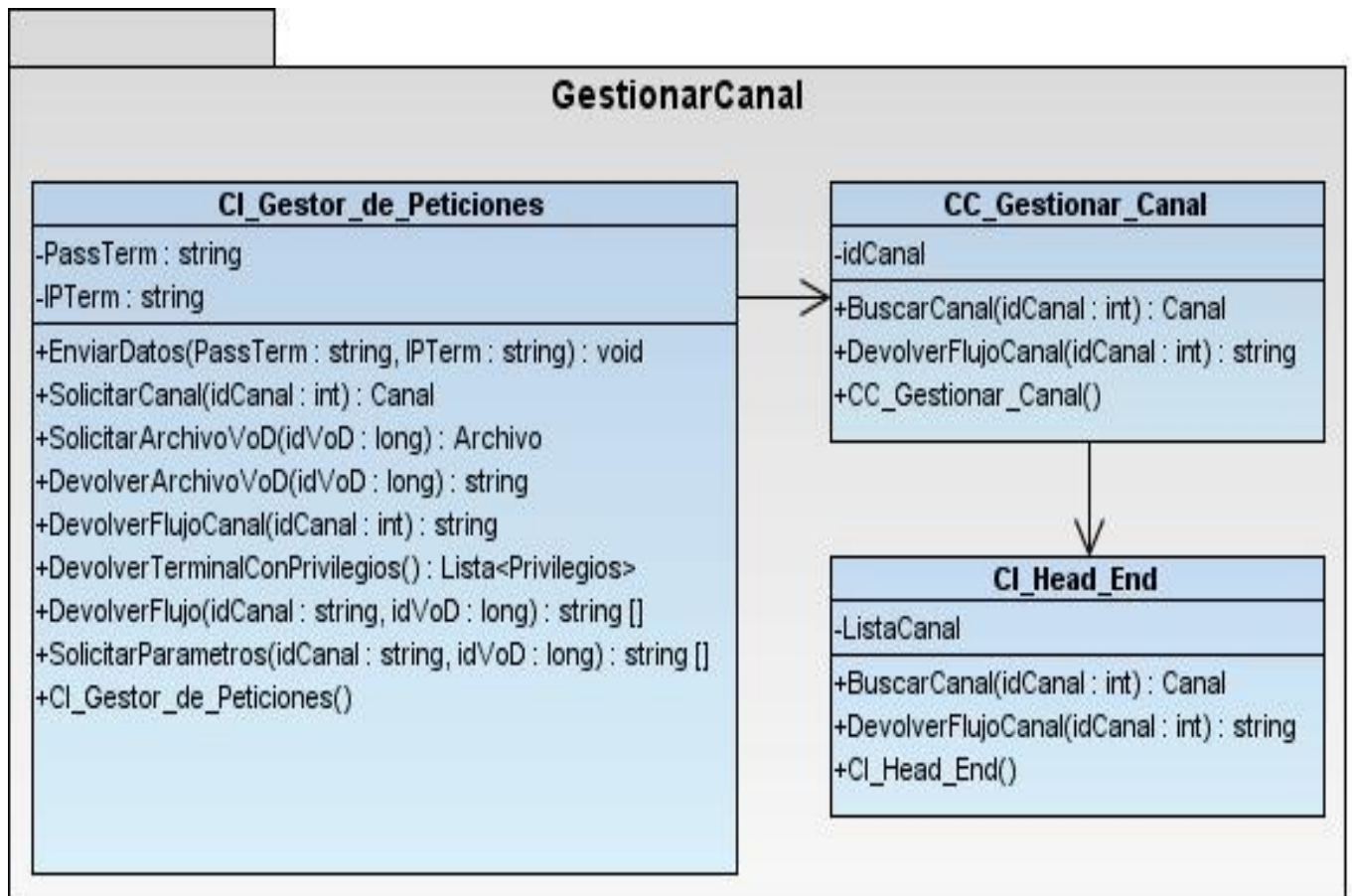


Figura 18: Diagrama de Clases del Diseño del Caso de Uso Gestionar Canal.

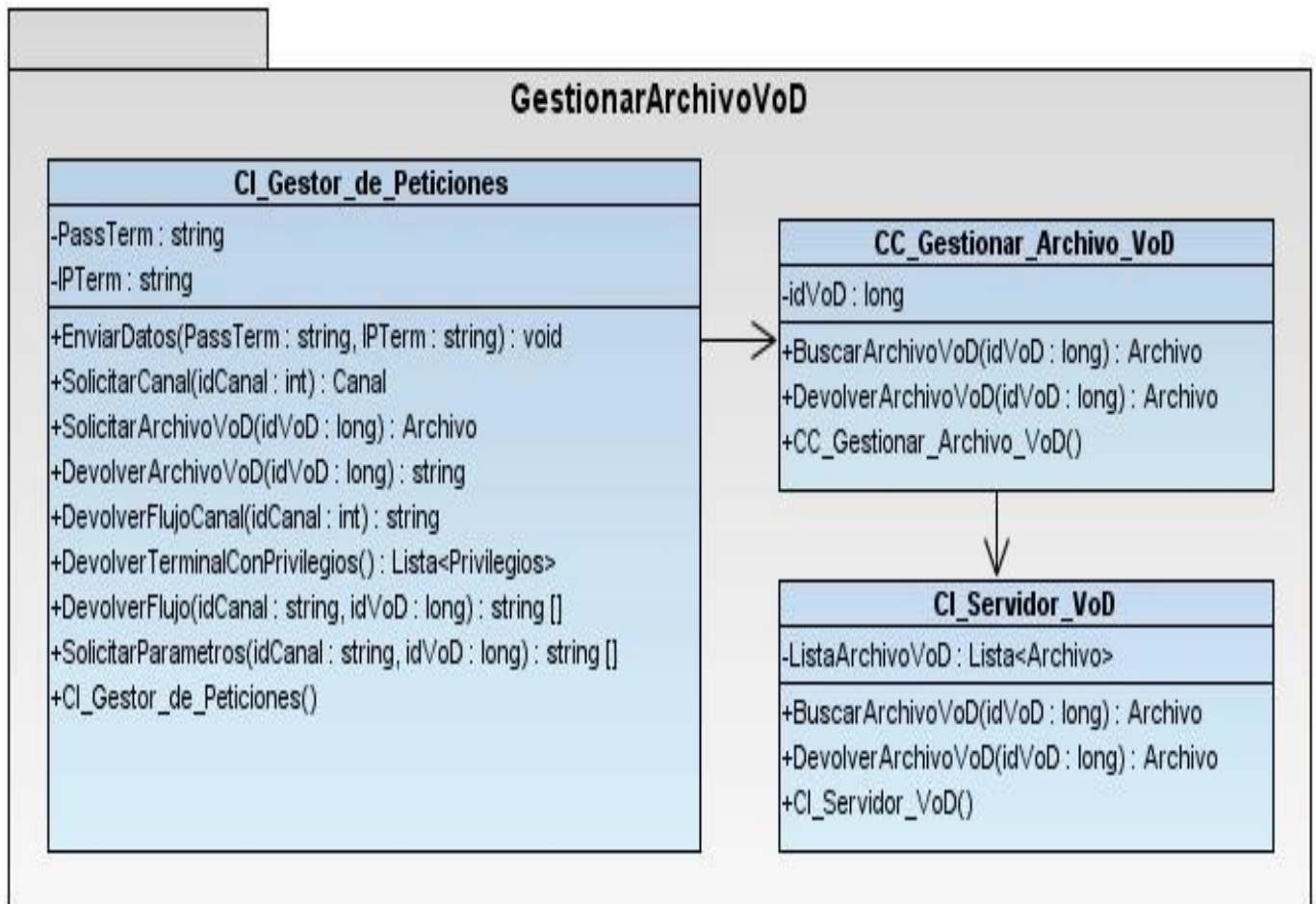


Figura 19: Diagrama de Clases del Diseño del Caso de Uso Gestionar Archivo VoD.

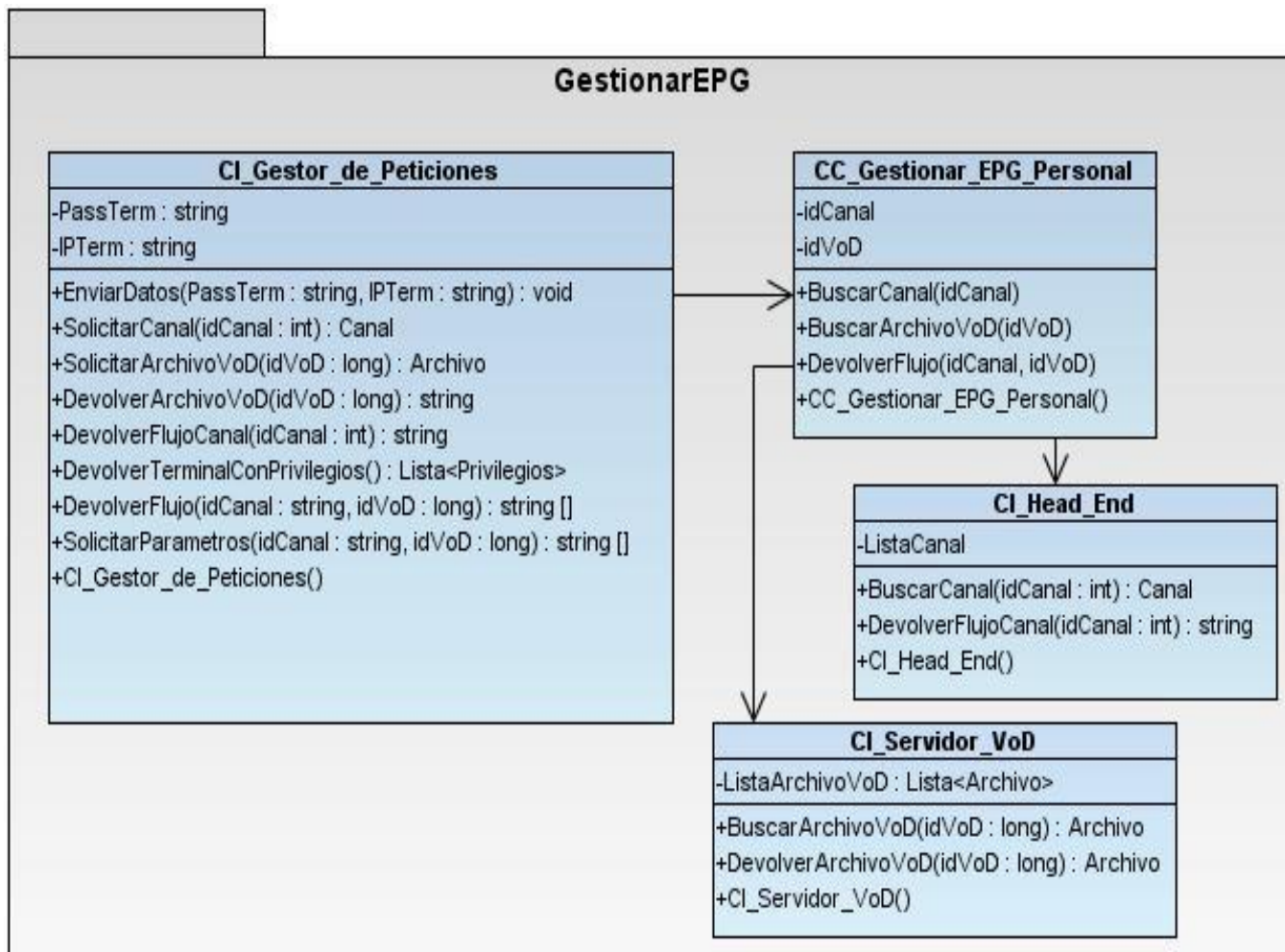


Figura 20: Diagrama de Clases del Diseño del Caso de Uso Gestionar EPG.

3.12 Conclusiones Parciales

Con la realización de este capítulo se logró un mejor entendimiento del problema planteado y se conformó un correcto modelado del dominio del sistema a diseñar. Fueron identificados los requerimientos tanto funcionales como no funcionales evidenciando las principales funciones que debe presentar el sistema. Fueron identificados los actores y casos de usos que surgen a partir de los requerimientos del sistema, además se describe detalladamente cada caso de uso del sistema. Se realizó el diagrama de clases del análisis y diagrama de colaboración para cada caso de uso. Además,

se definió una arquitectura propia que se ajusta a las características del sistema. El estudio de los patrones de arquitectura y diseño contribuyó a la correcta realización de los diagramas de clases del diseño construidos por cada caso de uso identificando bien las relaciones existentes y los métodos y atributos de cada clase.

Conclusiones

Al finalizar este trabajo de diploma que contiene el análisis y diseño del módulo Mediador para un Sistema IPTV en la UCI, se puede arribar a las siguientes conclusiones:

- Con el estudio de los sistemas IPTV existentes que presentan las empresas que llevan la delantera en este campo, fue comprobado que todas sus soluciones cuentan con un Middleware.
- Se determinaron las principales funcionalidades del Mediador que garantizarán un correcto funcionamiento del sistema IPTV.
- Con la metodología, herramientas y lenguaje de modelado escogido se garantizó un modelado estable y organizado de los procesos para formar la solución propuesta.
- Se logró construir el diseño del módulo Mediador.
- Con la modelación y diseño del sistema propuesto se garantiza que se logre una correcta implementación del mismo.

De esta forma, se da cumplimiento al objetivo general trazado para la elaboración del trabajo de diploma, obteniendo como resultado final el análisis y diseño del módulo Mediador del subsistema Middleware para un sistema IPTV en la UCI.

Recomendaciones

- Realizar los restantes flujos de trabajo que propone (RUP).
- Continuar investigando sobre posibles funcionalidades a incorporar al subsistema con nuevos requerimientos que puedan surgir.
- Integrar este módulo diseñado con los demás subsistemas para formar el sistema IPTV.

Bibliografía Consultada

1. **Martínez, Yarisel Rodríguez.** *Diseño de un Sistema Gestor de Guía Electrónica de Programación.* Habana : Universidad de las Ciencias Informáticas, 2009.
2. **ITU e-Learnig Centre.** *Guía Práctica de utilización de la Plataforma ITU.* China : s.n., 2004-2005.
3. **Link World Network S.L.** CANALTOPDIGITAL.COM . *CANALTOPDIGITAL.COM* . [En línea] Link World Network S.L. [Citado el: 05 de Diciembre de 2009.] http://www.canaltopdigital.com/tv_digital/.
4. **Gobierno de España. Ministerio de Industria, Turismo y Comercio.** Televisión Digital Terrestre. *Televisión Digital.* [En línea] Ministerio de Industria, Turismo y Comercio. [Citado el: 05 de Diciembre de 2009.] <http://www.televisiondigital.es/Terrestre/Que/Paginas/Que.aspx>.
5. **Gobierno de España. Ministerio de Industria, Turismo y Comercio .** Televisión Digital por Cable. *Televisión Digital.* [En línea] Ministerio de IndustriaTurismo y Comercio. [Citado el: 5 de Diciembre de 2009.] <http://www.televisiondigital.es/Cable/Paginas/TV%20Cable.aspx>.
6. **Gobierno de España. Ministerio de Industria, Turismo y Comercio.** Televisión Digital Vía Satelite. *Televisión Digital.* [En línea] Ministerio de Industria, Turismo y Comercio. [Citado el: 5 de Diciembre de 2009.] <http://www.televisiondigital.es/Satelite/Paginas/TVSatelite.aspx>.
7. **Gobierno de España. Ministerio de Industria, Turismo y Comercio .** Televisión Digital. <http://www.televisiondigital.es>. [En línea] Ministerio de Industria, Turismo y Comercio . [Citado el: 05 de Diciembre de 2009.] <http://www.televisiondigital.es/ADSL/Paginas/tvadsl.aspx>.
8. **Huawei Technologies Co., Ltd.** Huawei. [En línea] Huawei Technologies Co., Ltd, 2009. [Citado el: 6 de 12 de 2009.] <http://www.huawei.com/es/catalog.do?id=1690>.
9. **Huawei Technologies Co., Ltd.** Huawei. [En línea] Huawei Technologies Co., Ltd, 2009. [Citado el: 6 de 12 de 2009.] <http://www.huawei.com/es/catalog.do?id=1690>.
10. **ZTE Corporation.** ZTE Corporation. *ZTE Corporation.* [En línea] ZTE, 1998-2010 . [Citado el: 03 de 01 de 2010.] <http://www.zte.com.cn/>.
11. **IPTV-Américas.** IPTV Américas. *IPTV-Américas.* [En línea] 2006. [Citado el: 25 de 09 de 2009.] <http://www.iptv-americas.tv/solutions.php?leng=E>.

Bibliografía

12. **NetUP - Soluciones IPTV.** NetUP. *NetUP*. [En línea] NetUP - Soluciones IPTV, 2010. [Citado el: 03 de Enero de 2010.] <http://www.netup.tv/ru-RU/index.php>.
13. **Don Wells.** Extreme Programming. *Extreme Programming*. [En línea] 2009. [Citado el: 25 de Febrero de 2010.] <http://www.extremeprogramming.org/>.
14. **Ambyssoft Inc.** Agile Modeling. *Agile Modeling*. [En línea] 2005-2009. [Citado el: 25 de Febrero de 2010.] <http://www.agilemodeling.com/essays/fdd.htm>.
15. **Ivar Jacobson, Grady Booch, James Rumbaugh.** <http://bibliodoc.uci.cu>. *El Proceso Unificado de desarrollo de Software*. [En línea] 2000. [Citado el: 20 de 03 de 2010.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
16. **Object Management Group.** Object Management Group. *sitio Web de Object Management Group*. [En línea] 1997-2010. [Citado el: 10 de Febrero de 2010.] <http://www.uml.org/>.
17. **madritel.** madritel. [En línea] 15 de Julio de 2003. [Citado el: 20 de 12 de 2009.] <http://web.madritel.es/personales3/edcollado/ingsw/tema2/tema2.htm..>
18. **IBM.** *Sitio web de IBM*. [En línea] [Citado el: 11 de Febrero de 2010.] <http://www.ibm.com/software/awdtools/developer/rose/enterprise/index.html>.
19. **Visual Paradigm International .** Visual Paradigm. *Visual Paradigm*. [En línea] Visual Paradigm International , 2010. [Citado el: 26 de Febrero de 2010.] <http://www.visual-paradigm.com/product/vpum/>.
20. **Jacobson, I, Booch, G y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de Software*. Habana : Felix Varela, 2004. pág. 435.
21. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000. 8478290362.
22. **Rondón, Yoandri Quintana.** *Desarrollo de la arquitectura del proyecto Captura y*. Habana : s.n., 2009.
23. **Kiccillof, Carlos Reynoso – Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : s.n., 2004. s.n.
24. **Shaw, David Garlan y Mary.** *An Introduction to Software Architecture*. New Jersey V.Ambriola and G.Tortora : World Scientific , 1994.
25. **Corporation, Microsoft.** *Integration Patterns (Patterns & Practices)*. s.l. : s.l. : Microsoft Press, 2004. 073561850X.
26. **Larman, Craig.** *UML y Patrones*. La Habana : Felix Varela, 2004.

Bibliografía Citada

1. **Martínez, Yarisel Rodríguez.** *Diseño de un Sistema Gestor de Guía Electrónica de Programación.* Habana : Universidad de las Ciencias Informáticas, 2009.
2. **ITU e-Learnig Centre.** *Guía Práctica de utilización de la Plataforma ITU.* China : s.n., 2004-2005.
3. **Link World Network S.L.** CANALTOPDIGITAL.COM . *CANALTOPDIGITAL.COM* . [En línea] Link World Network S.L. [Citado el: 05 de Diciembre de 2009.] http://www.canaltopdigital.com/tv_digital/.
4. **Gobierno de España. Ministerio de Industria, Turismo y Comercio.** Televisión Digital Terrestre. *Televisión Digital.* [En línea] Ministerio de Industria, Turismo y Comercio. [Citado el: 05 de Diciembre de 2009.] <http://www.televisiondigital.es/Terrestre/Que/Paginas/Que.aspx>.
5. **Gobierno de España. Ministerio de Industria, Turismo y Comercio .** Televisión Digital por Cable. *Televisión Digital.* [En línea] Ministerio de IndustriaTurismo y Comercio. [Citado el: 5 de Diciembre de 2009.] <http://www.televisiondigital.es/Cable/Paginas/TV%20Cable.aspx>.
6. **Gobierno de España. Ministerio de Industria, Turismo y Comercio.** Televisión Digital Vía Satelite. *Televisión Digital.* [En línea] Ministerio de Industria, Turismo y Comercio. [Citado el: 5 de Diciembre de 2009.] <http://www.televisiondigital.es/Satelite/Paginas/TVSatelite.aspx>.
7. **Gobierno de España. Ministerio de Industria, Turismo y Comercio .** Televisión Digital. <http://www.televisiondigital.es>. [En línea] Ministerio de Industria, Turismo y Comercio . [Citado el: 05 de Diciembre de 2009.] <http://www.televisiondigital.es/ADSL/Paginas/tvadsl.aspx>.
8. **Huawei Technologies Co., Ltd.** Huawei. [En línea] Huawei Technologies Co., Ltd, 2009. [Citado el: 6 de 12 de 2009.] <http://www.huawei.com/es/catalog.do?id=1690>.
9. **Huawei Technologies Co., Ltd.** Huawei. [En línea] Huawei Technologies Co., Ltd, 2009. [Citado el: 6 de 12 de 2009.] <http://www.huawei.com/es/catalog.do?id=1690>.
10. **ZTE Corporation.** ZTE Corporation. *ZTE Corporation.* [En línea] ZTE, 1998-2010 . [Citado el: 03 de 01 de 2010.] <http://www.zte.com.cn/>.
11. **IPTV-Américas.** IPTV Américas. *IPTV-Américas.* [En línea] 2006. [Citado el: 25 de 09 de 2009.] <http://www.iptv-americas.tv/solutions.php?leng=E>.

Bibliografía

12. **NetUP - Soluciones IPTV.** NetUP. *NetUP*. [En línea] NetUP - Soluciones IPTV, 2010. [Citado el: 03 de Enero de 2010.] <http://www.netup.tv/ru-RU/index.php>.
13. **Don Wells.** Extreme Programming. *Extreme Programming*. [En línea] 2009. [Citado el: 25 de Febrero de 2010.] <http://www.extremeprogramming.org/>.
14. **Ambyssoft Inc.** Agile Modeling. *Agile Modeling*. [En línea] 2005-2009. [Citado el: 25 de Febrero de 2010.] <http://www.agilemodeling.com/essays/fdd.htm>.
15. **Ivar Jacobson, Grady Booch, James Rumbaugh.** <http://bibliodoc.uci.cu>. *El Proceso Unificado de desarrollo de Software*. [En línea] 2000. [Citado el: 20 de 03 de 2010.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
16. **Object Management Group.** Object Management Group. *sitio Web de Object Management Group*. [En línea] 1997-2010. [Citado el: 10 de Febrero de 2010.] <http://www.uml.org/>.
17. **madritel.** madritel. [En línea] 15 de Julio de 2003. [Citado el: 20 de 12 de 2009.] <http://web.madritel.es/personales3/edcollado/ingsw/tema2/tema2.htm..>
18. **IBM.** *Sitio web de IBM*. [En línea] [Citado el: 11 de Febrero de 2010.] <http://www.ibm.com/software/awdtools/developer/rose/enterprise/index.html>.
19. **Visual Paradigm International .** Visual Paradigm. *Visual Paradigm*. [En línea] Visual Paradigm International , 2010. [Citado el: 26 de Febrero de 2010.] <http://www.visual-paradigm.com/product/vpum/>.
20. **Jacobson, I, Booch, G y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de Software*. Habana : Felix Varela, 2004. pág. 435.
21. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000. 8478290362.
22. **Rondón, Yoandri Quintana.** *Desarrollo de la arquitectura del proyecto Captura y*. Habana : s.n., 2009.
23. **Kiccillof, Carlos Reynoso – Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : s.n., 2004. s.n.
24. **Shaw, David Garlan y Mary.** *An Introduction to Software Architecture*. New Jersey V.Ambriola and G.Tortora : World Scientific , 1994.
25. **Corporation, Microsoft.** *Integration Patterns (Patterns & Practices)*. s.l. : s.l. : Microsoft Press, 2004. 073561850X.
26. **Larman, Craig.** *UML y Patrones*. La Habana : Felix Varela, 2004.

