



Universidad de las Ciencias Informáticas
Facultad 9

Análisis de una solución BSS/CRM para un sistema IPTV.

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

Autor:

Jean Luis Luque Ramírez.

Tutor:

Ing. Yoandrys S. Pacheco Jeréz.

Ciudad de La Habana, Junio de 2010.

“Año 52 de la Revolución”

DEDICATORIA

A mi mamá, por ser ejemplo de tesón y fuerza para vivir, guía de mis primeros pasos, y por ser la mejor madre del mundo.

A mis hermanos, Yadián y Lois, y al bebé que está por nacer, para que les sirva de impulso e inspiración.

A mi novia Nayda, por todo el apoyo, la confianza, el optimismo, por ser más de lo que un día soñé, te amo princesita linda.

A mis abuelos Nena y Nito, y en especial a los que ya no están, Antonio y Juana.

AGRADECIMIENTOS

A mi tutor, por el apoyo.

A los profesores de los que aprendí algo, y a los que aunque sea intentaron enseñarme.

A Dios o al destino, por la oportunidad de hacerme ingeniero.

A mi mamá y a mi novia, otra vez, por la confianza, y por sobrevivir a mis cambios de humor. Si no fuera por ustedes no hubiera podido levantarme otra vez...

A todos los que interactuaron conmigo todos estos años, a los que lo agradecen y a los que no, gracias. A la gente del "mundo", al Chino, Yasmany, Friky. A la gente del zoológico que fue mi apartamento de 5to, al Yano, al Mellao y al Bin. A los compas del doble, a Badboy el "father". A la gente de la mechadera en las escaleras, a Gilberto el que más barras hacía.

A la persona que no puedo mencionar pero a la cual debo hoy poder estar aquí atacándome con estos agradecimientos.

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente del mismo.

Para que así conste firmo la presente a los __ días del mes de junio del 2010.

Jean Luis Luque Ramírez.

Autor

Yoandrys S. Pacheco Jeréz.

Tutor

DATOS DE CONTACTO

TUTOR: Ing. Yoandrys Silverio Pacheco Jeréz.

Ingeniero en Ciencias Informáticas

Graduado en la Universidad de las Ciencias Informáticas en Julio de 2007

Profesor Instructor – UCI

Facultad 9

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 119, Apto: 119106

Teléfono Oficina: +53 – 7 – 8373780 Teléfono Apto: +53 – 7 – 8372752 E-mail: ypachecoj@uci.cu

RESUMEN

El avance vertiginoso de las Tecnologías de la Información y las Comunicaciones y el cada vez más rápido crecimiento del volumen de información que sobre distintos temas existe actualmente en el mundo, conducen a una mayor utilización de los sistemas de apoyo al negocio (Business Support Systems, o BSS por sus siglas en inglés). No se trata únicamente de que las aplicaciones funcionen internamente de una manera correcta, sino de lograr que el usuario final se vea como centro del negocio, pues es él precisamente el medidor que expresará en niveles de satisfacción el uso de estas tecnologías. Se aconseja utilizar a la par de los sistemas BSS, filosofías de desarrollo de software en las que el usuario sea el principal objeto a complacer en todo momento, como la CRM (Customer Relationship Management, o gestión de las relaciones con el cliente).

En el campo de la televisión que se ofrece usando protocolo IP (IPTV), estas necesidades se hacen cada vez más evidentes, dado que el mercado es uno de los más competentes y variables en el campo de las telecomunicaciones, y donde el servicio personalizado a los clientes es el que se impone. En la Universidad de las Ciencias Informáticas (UCI) se ha creado un Grupo de Desarrollo de Componentes con el cual se pretende abordar, investigar, e implementar soluciones relativas a estos temas avanzados de los cuales en Cuba y en la propia Universidad se conoce muy poco, pero existen investigaciones e interés en estas tecnologías, debido a las prestaciones que puede ofrecer en principio a la comunidad universitaria, y en el futuro la posibilidad de extenderse a otros espacios. Por estas razones se decide realizar el análisis correspondiente al componente BSS/CRM de un sistema IPTV, donde el desarrollo e implantación del mismo estaría condicionado por los recursos e infraestructura tecnológica con que se cuente en la universidad.

Este trabajo de diploma está estructurado en cuatro capítulos. El **Capítulo 1**, contempla la fundamentación teórica de esta investigación, en la cual son expuestos los principales conceptos y argumentos que esclarecen el **objeto de estudio** de la misma. El **Capítulo 2** trata acerca de las metodologías de desarrollo, lenguaje de modelado y herramientas de apoyo y mantenimiento que se utilizan en la actualidad a nivel internacional para el desarrollo de software así como la selección de las que se utilizarán para el desarrollo de la documentación ingenieril del componente en cuestión. En el **Capítulo 3** se presenta el análisis del sistema y en el **Capítulo 4** se expone el diseño de la solución propuesta.

PALABRAS CLAVE

Sistema de Apoyo al Negocio (BSS), CRM, sistemas IPTV, Tecnologías de la Información y las Comunicaciones (TIC).

Índice de Contenido

Introducción	1
CAPÍTULO 1	5
Fundamentación Teórica.	5
1.1 Introducción.....	5
1.2 Breve historia de la televisión.....	5
1.3 Sistemas de televisión digital.....	6
1.3.1 Definición de televisión digital (DTV, Digital Television.).....	7
1.3.2 Tipos de televisión digital (DTV).	7
1.4 Ventajas de la DTV sobre la TV analógica.....	7
1.5 Objeto de estudio.....	8
1.5.1 Definición de IPTV.....	8
1.5.2 Diferencias entre IPTV y DTV.....	8
1.5.3 Diferencias entre IPTV y la TV sobre Internet.	8
1.6 Conceptos asociados al dominio del problema.	9
1.6.1 Calidad de servicio (QoS).	9
1.6.2 Calidad de experiencia (QoE).	10
1.6.3 Latencia.....	10
1.6.4 Multidifusión (Multicast).	10
1.6.5 Streaming.....	12
1.6.6 Zapping.....	12
1.6.7 Códec.	12
1.6.8 Jitter.....	12
1.7 Descripción de algunos protocolos utilizados en IPTV.....	13
1.7.1 Protocolo de Internet (IP, Internet Protocol).....	13
1.7.2 Protocolo de Datagramas de Usuario (UDP, User Datagram Protocol).....	13
1.7.3 Protocolo de Transporte en tiempo Real (RTP, Real-time Transport Protocol).....	14
1.7.4 Protocolo de Control para el Transporte en tiempo Real (RTCP).....	14
1.7.5 RTSP (Real Time Streaming Protocol).	15
1.7.6 Protocolo de Gestión de Grupos de Internet (IGMP, Internet Group Management Protocol).	15
1.8 Principales servicios de IPTV.....	15
1.8.1 Servicio de Broadcast TV.....	16
1.8.2 Servicio de Video bajo Demanda (VoD, Video on Demand).	16
1.9 Elementos de un sistema IPTV.....	17
1.9.1 Head End.....	17
1.9.2 Servidores de Video.....	18
1.9.3 Middleware.....	18
1.9.4 Sistema de Control de Derechos y de Acceso Condicional (DRM/CAS, Digital Rights Management/Conditional Access System).....	19
1.9.5 Set Top Box (STB).....	19
1.9.6 Home Gateways.....	20
1.9.7 Servidores para la gestión de la red y de las direcciones IP de los STB.	20
1.10 Principales funcionalidades de un sistema IPTV.....	20
1.10.1 Guía Electrónica de Programación (EPG, Electronic Program Guide).....	21
1.10.2 Grabación de Video personal (PVR, Personal Video Recorder).....	21
1.10.3 Pagar por Ver (PPV, Pay Per View).	21

1.10.4 Time-Shift TV (TSTV).	22
1.11 Características de la red de acceso de los suscriptores.....	22
1.11.1 Acceso ADSL.	22
1.11.2 Normas de acceso ADSL.....	23
1.12 Arquitectura de referencia de una red IPTV según ATIS.	24
1.13 Situación problemática: las soluciones BSS/CRM en sistemas IPTV.	24
1.14 Análisis de soluciones existentes.	27
1.15 Situación de Cuba ante los retos de la TV Digital e IPTV.	29
1.16 Conclusiones Parciales.....	29
CAPÍTULO 2	31
Tendencias y tecnologías actuales.....	31
Introducción.	31
2.1 Metodologías de desarrollo de software.	31
2.1.1 Metodologías tradicionales de desarrollo.	32
MSF (Microsoft Solution Framework).....	32
RUP (Rational Unified Process).....	33
2.1.2 Metodologías ágiles de desarrollo.	34
XP (Xtreme Programming).	34
SCRUM.	36
Crystal Clear.	36
ASD (Adaptive Software Development).....	37
2.2 Lenguajes de modelado.....	38
2.2.1 Lenguaje Unificado de Modelado (UML).	38
2.3 Herramientas CASE.	39
2.4 Observaciones del estudio de las tendencias y tecnologías.	41
2.5 Conclusiones Parciales.	42
CAPÍTULO 3	43
Análisis del sistema.....	43
Introducción.	43
3.1 El modelo de dominio en el desarrollo de software.	43
3.1.1 Modelo de dominio BSS/CRM.	44
3.1.2 Diagrama de clases del modelo de dominio.	44
3.1.3 Definición de las clases del modelo de dominio.	44
3.1.4 Breve descripción del diagrama.	45
3.2 Requerimientos.....	46
3.2.1 Requerimientos funcionales.	46
3.2.2 Requerimientos no funcionales.....	47
3.3 Descripción del sistema. Modelo de Casos de Uso del Sistema.	49
3.3.1 Modelo de Casos de Uso del Sistema.	50
3.4 Descripción textual de Casos de Uso del sistema.	50
3.4.1 Gestionar Usuarios.	50
3.4.2 Autenticar Usuario.....	54
3.4.3 Gestionar Cuentas.	54
3.4.4 Gestionar Servicios.	56
3.4.5 Gestionar Historial de Programas.....	58
3.4.6 Ofertar Programación.....	58

3.5 Diagramas de clases del análisis.	59
3.5.1 DCA CU Autenticar Usuario.	59
3.5.2 DCA CU Gestionar Usuario.	60
3.5.3 DCA CU Gestionar Cuentas.	60
3.5.4 DCA CU Gestionar Servicios.	60
3.5.5 DCA CU Gestionar Historial de Programas.	60
3.5.6 DCA CU Ofertar Programación.	60
3.6 Diagramas de colaboración del análisis.	60
3.6.1 Diagrama de Colaboración CU Autenticar Usuario.	60
3.6.2 Diagrama de Colaboración CU Gestionar Usuario.	60
3.6.3 Diagrama de Colaboración CU Gestionar Cuentas.	61
3.6.4 Diagrama de Colaboración CU Gestionar Servicios.	61
3.6.5 Diagrama de Colaboración CU Gestionar Historial de Programas.	61
3.6.6 Diagrama de Colaboración CU Ofertar Programación.	62
Conclusiones Parciales.	62
CAPÍTULO 4	63
Diseño del sistema.	63
Introducción.	63
4.1 El diseño de software.	63
4.2 Descripción de la arquitectura del sistema.	63
4.2.1 Patrón de arquitectura MVC.	64
4.3 Patrones de diseño.	65
4.4 Diagramas de clases del diseño.	66
4.4.1 Realización del CU Autenticar Usuario.	67
4.4.2 Realización del CU Gestionar Usuario.	67
4.4.3 Realización del CU Gestionar Cuentas.	67
4.4.4 Realización del CU Gestionar Servicios.	67
4.4.5 Realización del CU Gestionar Historial de Programas.	67
4.4.6 Realización del CU Ofertar Programación.	67
Conclusiones Parciales.	67
Conclusiones.	69
Recomendaciones.	70
Referencias Bibliográficas.	71

Índice de Figuras.

Figura 1: Diferencias entre el flujo de unidifusión y de multidifusión.	12
Figura 2: Formato de un paquete IP.	13
Figura 3: Cabecera UDP.	14
Figura 4: Cabecera RTP.	14
Figura 5: Capas en RTCP.	15
Figura 6: Arquitectura básica de un sistema típico VoD.	17
Figura 7: EPG de Microsoft TV.	21
Figura 8: Arquitectura de red IPTV según ATIS.	24
Figura 9: Fases y Flujos de Trabajo según RUP.	33
Figura 10: Las prácticas se refuerzan entre sí.	36
Figura 11: Familia de Crystal Methods.	37

Figura 12: Diagrama de clases del modelo de dominio.	44
Figura 13: Diagrama de Casos de Uso del Sistema.....	50
Figura 14: DCA CU Autenticar Usuario.....	59
Figura 15: DCA CU Gestionar Usuario.....	60
Figura 16: DCA CU Gestionar Cuentas.	60
Figura 17: DCA CU Gestionar Servicios.	60
Figura 18: DCA CU Gestionar Historial de Programas.	60
Figura 19: DCA CU Ofertar Programación.....	60
Figura 20: Diagrama de Colaboración CU Autenticar Usuario.	60
Figura 21: Escenario Insertar Usuario.....	60
Figura 22: Escenario Eliminar Usuario.	61
Figura 23: Escenario Modificar usuario.	61
Figura 24: Escenario Cambiar Contraseña.	61
Figura 25: Escenario Reactivar Usuario.	61
Figura 26: Escenario Suspende Usuario.	61
Figura 27: Escenario Ordenar Servicio.	61
Figura 28: Escenario Cancelar Servicio.	61
Figura 29: Diag. Colab. CU Gestionar Historial de Programas.....	62
Figura 30: Diag. Colab. CU Ofertar Programación.....	62
Figura 31: Estructura del patrón MVC.	64
Figura 32: DCD CU Autenticar Usuario.	67
Figura 33: DCD CU Gestionar Usuarios.	67
Figura 34: DCD CU Gestionar Cuentas.	67
Figura 35: DCD CU Gestionar Servicios.	67
Figura 36: DCD CU Gestionar Historial de Programación.....	67
Figura 37: DCD CU Ofertar Programación.....	67

Índice de Tablas.

Tabla 1: Normas de acceso xDSL.	23
Tabla 2: Situaciones factibles de despliegue de servicios.	23
Tabla 3: Descripción de los actores del sistema.	50
Tabla 4: Descripción del CU Gestionar Usuario.	53
Tabla 5: Descripción del CU Autenticar Usuario.	54
Tabla 6: Descripción del CU Gestionar Cuentas.	56
Tabla 7: Descripción del CU Gestionar Servicios.....	58
Tabla 8: Descripción del CU Gestionar Historial de Programas.....	58
Tabla 9: Descripción del CU Ofertar Programación.	59

Introducción

La televisión es uno de los inventos que ha cambiado la forma en que la sociedad y la cultura han evolucionado en los últimos 60 años, debido en gran parte a su poderoso efecto, entrelazando y creando un mundo desde una visión unificada. En 1969 ARPANET¹ fue creada, y con ella comenzó una nueva fase en las comunicaciones. Más adelante, en 1983, el protocolo núcleo de ARPANET migró de NCP (del inglés Network Control Protocol), a TCP/IP (del inglés Transfer Control Protocol/Internet Protocol) y de esta forma nació Internet.

Ambos, la televisión e Internet, han revolucionado la forma en que se vive. En la actualidad se dispone tanto de canales de televisión transmitiendo diversos contenidos 24 horas al día, como de Internet, facilitando tanto las comunicaciones como el comercio. Varias áreas comunes de estos trascendentales inventos tecnológicos al fin se han unido para dar vida a la televisión por IP, o IPTV como generalmente se le conoce. (1)

IPTV puede ser visto como ofertas comerciales brindadas por proveedores de servicios que están muy cerca de los consumidores, ofertas que incluyen un número de canales con una apariencia similar a la televisión tradicional. En realidad, es una nueva tecnología que permite una mayor flexibilidad para gestionar contenido, y facilitar interacción directa con las fuentes de contenido, mejorando la retroalimentación y la planificación. La interacción o experiencia del consumidor o suscriptor es mejorada considerablemente permitiendo un mayor control sobre los diferentes contenidos inmediatamente disponibles, así como comunicación hacia ambos lados, dígame entre consumidor y proveedor de contenidos. (1)

Hoy día, las comunicaciones se encuentran en un proceso de migración hacia la convergencia de servicios, evidentemente, proveer servicios de voz, datos y telefonía de manera aislada es un error que muchas compañías han pagado caro. A esa tecnología que incluye proveer esos tres tipos de servicios se le conoce como “triple play”. El principal requerimiento para estos servicios es disponer de un ancho de banda lo suficientemente grande como para ofrecer fiabilidad y competitividad para el despliegue de los mismos.

Los retos crecientes en los actuales sistemas de apoyo a las empresas demandan confiables y escalables estándares de tecnología de servidores de computadoras con rendimiento en ascenso para manejar la complejidad de estos sistemas, el número de suscriptores, los servicios, y la

¹ Advanced Research Projects Agency Network. Agencia creada por el Departamento de Defensa (DoD) de los EE.UU como medio de comunicación para los distintos organismos de ese país.

información sobre los precios. Los sistemas de apoyo a las empresas o al negocio (BSS en sus siglas en inglés, correspondientes a Business Support Systems), son usados para ejecutar operaciones en el campo de los negocios que se desarrollan en la web, o sea el comercio electrónico. Entre las áreas principales de BSS están la gestión de productos, gestión de clientes, gestión de ingresos y gestión de pedidos.

La gestión de productos incluye regularmente ofertas de nuevos productos, descuentos, y programas de fidelización de clientes. La gestión de clientes, como solución óptima incluye una solución CRM (del inglés Customer Relationship Management), o sea gestión de las relaciones con el cliente, tema sobre el cual se hará referencia posteriormente en este trabajo. La gestión de ingresos está enfocada en la facturación, cobro y liquidación de cuentas, además incluye en ocasiones la realización de acuerdos de asociación de usuarios al sistema o aplicación que ofrece el servicio en cuestión. La gestión de pedidos, como lo dice el nombre, se encarga de lo referente a los pedidos, y al aprovisionamiento de productos o servicios.

Las actividades típicas de un sistema BSS pueden ser: tomar los pedidos de los clientes, gestionar información de los clientes, gestión de los datos del pedido, facturación, y ofrecer servicios B2B² y B2C³ (Business to Business y Business to Customer respectivamente). (2)

En la UCI no existe un sistema IPTV que brinde las comodidades y funcionalidades por las que estos sistemas se han extendido en el mundo, ya sea por la complejidad de los mismos o por el costo que el desarrollo y despliegue de estos sistemas implican. Al mismo tiempo, no se cuenta con una solución BSS que gestione todos los servicios que brinda un sistema IPTV, y que se encargaría de darle atención personalizada a los usuarios. Es por éstas razones que el Grupo de Desarrollo de Componentes Técnicos de la Facultad 9 de la UCI se ha trazado como solución a la gestión de usuarios, productos y pedidos en general, el análisis de una solución BSS/CRM para su posterior implementación y posible implantación en la Universidad como parte de un sistema IPTV.

Partiendo de la situación anteriormente expuesta se tiene como **problema científico** la inexistencia de una solución BSS/CRM que permita brindar un servicio personalizado, para un sistema IPTV en la UCI. Con el fin de darle solución a dicha problemática, se hace necesario elaborar la documentación

² Transacciones electrónicas hechas entre empresas que aseguran la rapidez y seguridad de las comunicaciones, entre otras ventajas.

³ Estrategia que siguen las empresas para llegar directamente al cliente o usuario final.

técnica del análisis de una solución BSS/CRM, utilizando las técnicas y herramientas de la Ingeniería de Software seleccionadas, lo cual constituye el **objetivo general** de esta investigación.

Para darle cumplimiento a tales objetivos trazados fue necesario enfocar la investigación en el proceso de funcionamiento de sistemas IPTV, lo cual representa el **objeto de estudio**. Fue necesario investigar sobre las soluciones BSS/CRM para sistemas IPTV, lo cual constituye a su vez el **campo de acción** de esta investigación.

Para lograr los objetivos trazados se acometen las siguientes **tareas de la investigación**:

- 1- Caracterizar las soluciones que implementan y soportan BSS/CRM en soluciones IPTV.
- 2- Valorar el estado del arte de las principales y más actuales tecnologías que utilizan BSS/CRM.
- 3- Realizar la documentación técnica correspondiente al Modelo de Dominio del módulo BSS/CRM.
- 4- Capturar los requisitos funcionales y no funcionales.
- 5- Realizar la documentación técnica correspondiente al Modelo de Casos de Uso del Sistema del módulo BSS/CRM.
- 6- Realizar la documentación técnica correspondiente al modelo de análisis del módulo BSS/CRM.

El desarrollo satisfactorio de las tareas expuestas anteriormente contribuirá al cumplimiento de la **idea a defender** de esta investigación, la cual plantea que si se elabora correctamente la documentación técnica correspondiente al análisis de una solución BSS/CRM para un sistema IPTV se logrará una adecuada implementación y desarrollo de dicha propuesta en la UCI, que permita un servicio cuyo enfoque sea el cliente como centro de atención.

Con el objetivo de llevar a cabo la investigación se utilizaron métodos tanto teóricos como empíricos. De los teóricos se utilizaron el histórico-lógico, análisis-síntesis, causal y el de modelación. De los empíricos, las entrevistas y encuestas.

El método **histórico-lógico** se aplica para investigar la existencia de sistemas IPTV que usen soluciones BSS/CRM. Por otra parte, el método de **análisis y síntesis**, se utiliza con el objetivo de analizar la bibliografía disponible relacionada con el proceso de funcionamiento de sistemas IPTV y las metodologías existentes para la construcción de los artefactos necesarios para elaborar la solución, así como para realizar una síntesis de los mismos. El método **causal** se emplea para determinar y analizar los factores que provocan la necesidad de desarrollo de una solución BSS/CRM dentro de un sistema IPTV en la Universidad. Además, el método de **modelación** se emplea para mostrar los diferentes diagramas y componentes que se construyen como resultado del proceso de Ingeniería de Software.

Las **entrevistas** se realizan a líderes del proyecto para recopilar información referente al proceso de funcionamiento de sistemas IPTV. Al mismo tiempo, se utilizan **encuestas** para identificar las herramientas óptimas destinadas a la construcción de soluciones BSS/CRM, además de conocer las principales funcionalidades que encierran los mismos.

Este trabajo de diploma cuyos **posibles resultados** están dados en la documentación general de los flujos de trabajo Negocio, Requerimientos y Análisis y Diseño de una solución BSS/CRM de un sistema IPTV para que de esta manera posteriormente se garantice una implementación robusta de la aplicación, se estructura en cuatro capítulos, que son descritos a continuación:

El **Capítulo 1** contempla la fundamentación teórica de esta investigación, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, y se especifican detalladamente todos los argumentos que esclarecen el objeto de estudio.

Por su parte el **Capítulo 2** trata acerca de las tendencias y tecnologías que se utilizan en la actualidad a nivel internacional para el desarrollo de soluciones BSS/CRM para sistemas IPTV. Paralelamente a esto se estudian un conjunto de metodologías (ágiles, pesadas) que se utilizan en el mundo para el desarrollo de software de forma general, con el objetivo de seleccionar de ellas, la más adecuada para guiar el proceso de desarrollo de este tipo de software.

En el **Capítulo 3** se presenta la solución propuesta a partir de la descripción del modelo de dominio y la identificación de requisitos funcionales y no funcionales del sistema a construir, que finalmente se agrupan en casos de uso del sistema para conformar el Modelo de Casos de Uso del Sistema.

Por último en el **Capítulo 4** se expone la construcción de la solución propuesta en términos de la realización de los artefactos que forman parte del flujo de trabajo de diseño. De manera general se expone la estructura de la solución propuesta. Para finalizar se presentan las conclusiones, las recomendaciones, y las referencias bibliográficas.

CAPÍTULO **1**

Fundamentación Teórica.

1.1 Introducción.

Este primer capítulo está dedicado a explicar los principales aspectos asociados con la tecnología IPTV, haciendo referencia a sus principales servicios, elementos, protocolos relacionados con el flujo de video transmitido, funcionalidades, arquitecturas de red, y diferencias respecto a otros tipos de tecnologías existentes. Todo esto con el objetivo de ofrecer una panorámica de la tecnología a la cual se le quiere elaborar una solución BSS/CRM factible dentro del propio sistema IPTV.

1.2 Breve historia de la televisión.

La palabra televisión resulta de la mezcla de la voz griega “tele” (distancia) y la latina “visio” (visión). El término como tal se refiere a todos los aspectos de transmisión y programación, muchas veces se le trata como simplemente TV, y se dice que fue usado por primera vez en 1900 en el Congreso Internacional de Electricidad de París.

A finales del s. XIX se quiso hacer lo que ya se había hecho con el sonido, se trataba ahora de captar imágenes utilizando una cámara, transmitir esas imágenes a través del aire, y recibirlas en algún dispositivo receptor a determinada distancia del lugar donde se generó la transmisión (telefotografía).

Estos primeros intentos se realizaron mediante electricidad y sistemas mecánicos. La electricidad servía de unión entre los puntos y para realizar la captación y recepción de la imagen, mientras, los medios mecánicos efectuaban las tareas de movimiento para realizar los barridos y descomposición secuencial de la imagen a transmitir.

La exploración de una imagen se realiza mediante su descomposición, primero en fotogramas a los que se llaman cuadros y luego en líneas, leyendo cada cuadro. Se realizaron muchos estudios empíricos para determinar el número de cuadros necesarios para que se pueda recomponer una imagen en movimiento, así como para obtener una óptima calidad de reproducción y de percepción por el ojo humano del color, resultando en que el número de cuadros debía ser al menos de 24 por segundo, y que el número de líneas debía ser superior a las 300.

Hacia la década del '40 existían multitud de sistemas que tenían resoluciones muy diferentes, que iban desde 400 hasta 1000 líneas. Poco a poco se fueron concentrando en dos sistemas, el de 512 estadounidense, y el de 625 adoptado por Europa. También se adoptó el formato 4:3 para la relación de aspecto de la imagen. El desarrollo de la televisión no se limitó a la transmisión de imagen y sonido, a finales de la década de los '80 se vio la ventaja de usar el canal para ofrecer otros servicios, como el llamado teletexto, que no era más que transmitir noticias e información en formato de texto utilizando los espacios libres de información de la señal de video.

Mucho tuvo que hacerse y experimentarse antes de que se lograra la TV a color a principios de la década de los '50, se introdujeron los conceptos de luminancia y crominancia: el primero para describir la información del brillo y luz de la imagen; el segundo, como portador de la información del color en los canales RGB (Red, Green, Blue). Se plantea que el primer sistema de televisión a color que respetaba la doble compatibilidad con la TV monocromática fue desarrollado en 1951 por un grupo de ingenieros norteamericanos, e inmediatamente aceptado por la Comisión Federal de Comunicaciones (FCC, Federal Communication Commission) de ese país. Este sistema fue conocido como NTSC (National Television System Commission) y tuvo un éxito tal que se extendió por toda América del Norte y Japón.

Este sistema fue la base para que investigadores europeos, en este caso alemanes, desarrollaran un sistema que subsanaba algunos errores del NTSC como los de fase, tal sistema se le conoce como PAL (Phase Alternating Line), y fue adoptado por Europa Occidental con la excepción de Francia, que tuvo un investigador que desarrolló otro sistema muy conocido, el SECAM (Sequentiel Couleur A Memoire), que basa su actuación en la transmisión secuencial de cada componente de color moduladas en FM.

Todos estos sistemas tenían ventajas e inconvenientes: mientras que el NTSC y el PAL dificultaban la edición de la señal de video por su secuencia de color en cuatro y ocho campos respectivamente, el sistema SECAM hacía imposible el trabajo de mezcla de la señal de video. (3) Muchos años de nuevos descubrimientos llevaron a encontrar soluciones robustas ante interferencias y a la vez que tuvieran buena recepción de la señal televisiva, estas soluciones son las de la televisión digital. Además de las ventajas anteriormente mencionadas, hay que hablar de los valores añadidos con que cuenta la señal, o sea la personalización de la oferta de contenidos, formas de pago y atención a los usuarios.

1.3 Sistemas de televisión digital.

La TV digital se ha estado desarrollando intensamente en los últimos años, resultando en la implementación de tres estándares de TV adoptados por distintos países del mundo. Las normas de

TV digital ATSC (Advanced Television System Committee), DVB (Digital Video Broadcasting) e ISDB (Integrated Services Digital Broadcasting) permiten el transporte de información de audio y video en tiempo real, reemplazando las normas analógicas PAL, NTSC y SECAM. (4)

El cambio de TV analógica a digital es más agresivo que la introducción de la TV a color en su momento, puesto que supone una nueva tecnología de transmisión totalmente incompatible con las predecesoras. Esto significa que el usuario tendrá que reemplazar su TV estándar por uno con receptor de TV digital o instalar un dispositivo convertidor de la señal digital a alguno de los formatos preexistentes.

A continuación se abordarán algunas definiciones que deben tenerse claras a la hora de tratar estos temas de televisión, los distintos tipos que de ella existen, así como una breve comparación entre la TV analógica y la digital.

1.3.1 Definición de televisión digital (DTV, Digital Television.).

La TV digital se refiere al conjunto de tecnologías de transmisión y recepción de imágenes y sonido a través de señales digitales en lugar de las tradicionales analógicas, de manera que la codificación de la señal se realiza de forma binaria.

1.3.2 Tipos de televisión digital (DTV).

Dentro de la DTV, se encuentran la TV por Cable (Cable TV), la TV por Satélite (en inglés, Satellite TV o STV), y la IPTV. Cable TV como su nombre lo indica, consiste en la transmisión de señales digitales a través de sistemas de TV por cable, ya sea coaxial o telefónico. Por su parte STV se trata de transmitir señales satelitales en formato digital, entre los líderes proveedores del mercado están las compañías Sky y DirecTV.

1.4 Ventajas de la DTV sobre la TV analógica.

La conversión de la señal analógica a digital permite que la información sea fácilmente manipulada. Esto permite el desarrollo de técnicas de compresión que minimizan tanto el ancho de banda de la transmisión como el almacenamiento de la información. (5)

La resolución de pantalla⁴ juega un papel fundamental en la claridad de la imagen. En los sistemas analógicos de TV se usan 480 líneas activas para dibujar una imagen en el receptor de TV, en comparación, la TV digital comprende 18 formatos, entre los cuales están por supuesto los de 480,

⁴ Resolución de pantalla: Se refiere al número de píxeles que puede ser mostrada en la pantalla por pulgada.

pero además están los de 720 y 1080 líneas, siendo éstos dos últimos los que se usan para TV de alta definición (HDTV⁵).

En cuanto a la información del color, en un sistema NTSC por ejemplo, la información es adicionada como un portador por separado que es multiplexado además en monocromático, esto permite que la señal sea recibida tanto por televisores a color como en blanco y negro. Por su parte, la TV digital transporta la información de color en cada pixel⁶. (5)

1.5 Objeto de estudio.

1.5.1 Definición de IPTV.

IPTV es una tecnología interactiva que emplea una red IP que garantiza la calidad del servicio para cada flujo de información de video, igualando o mejorando la calidad de la distribución de señales digitales de video que hoy implementan la DTV, sea tanto en su versión terrestre, CATV (Cable TV), o satelital. Se trata de de un sistema de TV que se desarrolla en un entorno controlado, en el cual el proveedor de servicios no solo podrá en cualquier momento gestionar la red de acceso y de transmisión, sino también la de los contenidos que se ofrecen.

1.5.2 Diferencias entre IPTV y DTV.

La principal diferencia entre IPTV y la versión terrestre y satelital de DTV, es la relacionada con el medio por el cual se transmiten, o sea, IPTV usa el cable y esas otras versiones usan el aire. En el caso de la versión Cable TV de DTV, evidentemente también usa cable igual que IPTV, pero la diferencia radica en que la segunda implementa en la red el protocolo IP, no siendo obligatoriamente así en el primero.

La DTV en todas sus variantes, y la IPTV en principio, tienen previsto incorporar interactividad con el usuario. Ahora bien, en la DTV, se requiere de otra red complementaria para implementar un canal de retorno (generalmente un “call center” o centro de llamadas), y así hacer posible la interactividad, puesto que las redes son básicamente unidireccionales.

1.5.3 Diferencias entre IPTV y la TV sobre Internet.

Cuando se habla de TV sobre Internet e IPTV se tiende a tratar ambos términos como sinónimos, pero en realidad son dos tecnologías diferentes. La TV sobre Internet se refiere a la difusión de

⁵ High Definition Television.

⁶ Es la menor unidad homogénea en color que forma parte de una imagen digital.

noticias, estado del tiempo, y programas de televisión provenientes de estaciones de TV que agregan una interfaz de Internet a sus transmisiones “over-the-air” (Al Aire). Generalmente lo que hace esa interfaz es tomar las cintas de video y convertirlas a una secuencia de datagramas IP. Por su parte, IPTV se refiere a la transmisión de señales de video a una alta velocidad que permita a los usuarios, mediante su respectivo set-top box, la posibilidad de ver los distintos eventos televisivos a los que está suscrito sin necesidad de almacenar en buffer⁷. (5)

Por otra parte está el hecho de que IPTV se vale de redes IP sobre las cuales se logra implementar calidad de servicio para cada flujo de información de video, mientras que en la TV sobre Internet el flujo de información hace uso de un servicio basado en el “mejor esfuerzo” (servicio que ofrece el protocolo IP para la manipulación no fiable de los datagramas, o sea, lo hará lo mejor posible pero garantizando poco).

Por otra parte es preciso decir que la TV sobre Internet utiliza para su funcionamiento el protocolo http (del inglés Hyper Text Transfer Protocol), mientras que los sistemas IPTV para los servicios de Broadcast TV (Difusión de los contenidos de TV) utilizan el protocolo IGMP (del inglés Internet Group Multicast Protocol). (6)

1.6 Conceptos asociados al dominio del problema.

A continuación se hará alusión a una serie de conceptos que ayudarán a un mejor entendimiento del funcionamiento de los sistemas IPTV, así como de lo relativo a calidad de servicio y de experiencia, conceptos que hay que tener en cuenta a la hora de implementar soluciones BSS/CRM para sistemas IPTV.

1.6.1 Calidad de servicio (QoS).

Es una medida del funcionamiento del sistema a nivel del transporte de los paquetes y desde el punto de vista de la red. En sí QoS (del inglés Quality of Service), se refiere a las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado. También se conoce como QoS a los mecanismos que permiten al administrador de la red proveer servicios diferenciados a ciertos usuarios, otorgando prioridad a determinados flujos de tráfico de manera que se puedan mejorar por ejemplo los efectos de la congestión de las redes. (7) En el campo de las telecomunicaciones esta QoS es medible, entre estas medidas se pueden citar la disponibilidad de

⁷ Espacio de memoria en el que se almacenan datos para evitar que el programa o recurso que lo requiere, ya sea hardware o software, se quede sin información en algún momento.

las redes, el tiempo en que se tarda en realizar la comunicación y la tasa de errores en la descarga de algún archivo.

1.6.2 Calidad de experiencia (QoE).

Según el “Focus Group on IPTV” (Grupo Enfocado en IPTV) de la Unión Internacional de Telecomunicaciones (en inglés Internacional Telecommunication Union), se define como calidad de experiencia o QoE (Quality of Experience) a la aceptabilidad global de una aplicación o servicio, tal y como se percibe subjetivamente por el usuario final. Incluye la totalidad de efectos del sistema extremo a extremo (cliente, terminal, red, y servicios de infraestructura), y puede verse influenciada por las expectativas de los usuarios y por el contexto.

Entre los factores que influyen en la calidad de experiencia, se pueden citar el proceso de inicio de sesión, la forma en que opera el sistema una vez establecida dicha sesión, así como la finalización de la misma. Dentro de estos factores, se encuentran otros asociados a dichas etapas, por ejemplo, el esfuerzo requerido por el usuario para realizar la operación, el grado de respuesta de dicha operación, la fidelidad del contenido recibido, y la seguridad de la información personal del usuario.

1.6.3 Latencia.

Se le denomina latencia a la suma de retardos temporales dentro de una red de datos. Un retardo es producido por la demora en la propagación y transmisión dentro de la red. Otros factores que influyen en la latencia de una red son el tamaño de los paquetes transmitidos y el tamaño de los buffers dentro de los equipos de conectividad. (8)

El retardo tiene, a su vez, varios componentes, los cuales son retardo de codificación (depende del estándar utilizado), el retardo de señalización (depende de las interfaces de los equipos, siendo menor este retardo cuanto mayor sea la velocidad de la interfaz), el retardo de propagación (depende del medio físico que se utilice y la distancia recorrida), el retardo de encolamiento (tiempo en que un paquete está en una cola esperando a ser transmitido), y retardos de conmutación (tiempo que demora un switch (conmutador) o un router (enrutador) en poner los paquetes en una cola y decidir por cual interfaz los va a transmitir).

1.6.4 Multidifusión (Multicast).

Es similar a la difusión (broadcast⁸) en el sentido de que sus objetivos son un conjunto de máquinas dentro de una red de datos, pero no todas. O sea, en broadcast la señal le llega a todos los host dentro de la red, pero usando multicast, sólo le llega a un grupo de host específicos. Los hosts pueden escoger si desean o no participar en el grupo multicast (generalmente con el protocolo IGMP, que más adelante se abordará).

Se trata de usar la infraestructura de la red de manera muy eficiente requiriendo que la fuente envíe un paquete una sola vez, aun cuando éste sea enviado a múltiples destinos. Los nodos en la red se encargan de replicar el paquete para alcanzar varios destinos solamente cuando sea necesario. (9)

La multidifusión o multicasting, es uno de los conceptos claves de las redes IP. En realidad, existen dos significados muy diferentes del término que pueden ser aplicados al campo de los sistemas IPTV:

1. En transmisiones de TV digital over-the-air⁹, la multidifusión implica entregar múltiples señales de video simultáneamente a través de un único canal de transmisión digital.

2. En redes IP, la multidifusión implica entregar un único flujo de video a múltiples usuarios al mismo tiempo.

Para tener un mejor entendimiento de la multidifusión IP (IP multicasting), es aconsejable hacer una pequeña comparación con la unidifusión IP (IP unicasting). En unidifusión, cada flujo de video es enviado a exactamente un destinatario. Si múltiples destinatarios desean el mismo video, la fuente deberá crear por separado un flujo de video unicast para cada destinatario. Cada usuario que desee ver un video deberá hacer un pedido al proveedor o fuente de servicios, quien a su vez, deberá conocer la dirección IP de cada destinatario y crear un flujo de paquetes para cada usuario. A medida que el número de espectadores aumenta, la carga en la fuente aumenta evidentemente. (10)

Por su parte, la multidifusión IP, usa protocolos especiales para que en la propia red (ya no en la fuente de video) se hagan copias del flujo de video que llegará a los múltiples usuarios que lo solicitaron. Las copias son hechas en cada punto de la red solamente cuando son requeridas. (10)

En la siguiente figura se muestran las diferencias en la forma en que fluye la información bajo unidifusión y multidifusión:

⁸ Forma de emisión en la que el mismo contenido se difunde de forma homogénea a todos los usuarios a los que llega una determinada señal. Es la forma de emisión usada desde los orígenes de la televisión.

⁹ Forma de transmisión de señales digitales que usa antenas para la recepción.



Figura 1: Diferencias entre el flujo de unidifusión y de multidifusión.

1.6.5 Streaming.

Las técnicas para la transmisión de video y audio en las redes IP generalmente son conocidas como mecanismos de streaming. El término se refiere a la forma de transmitir información (generalmente multimedia) que permite al cliente ir consumiendo información al mismo tiempo que se va descargando. (11) Antes de la aparición de estas técnicas, las aplicaciones multimedia usaban Internet únicamente para realizar transferencias de archivos (una vez que los contenidos eran descargados completamente era que podían ser reproducidos).

1.6.6 Zapping.

El zapeo o zapping es el acto de saltar programación o canales de televisión. Aunque el término parece derivado del inglés, en realidad no lo es, se le llama en aquel idioma “flip channels”, o “channel surfing”. (12) En la actualidad, zapeo también se refiere al hecho de acelerar con el control remoto de video aquellas partes del programa que grabamos que no nos resulten interesantes, ya se trate de anuncios, o de algunas partes del programa en sí.

1.6.7 Códec.

Al par codificador-decodificador se le conoce como códec. Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos (stream) o una señal de video. El codificador convierte la señal previamente capturada a un formato comprimido, el cual será reconocido posteriormente por el decodificador, el cual regenera la señal de video intermedia para luego ser presentada en un monitor o televisor.

Los códecs son muy utilizados en Internet para reducir el tamaño de las películas de video o del audio y así facilitar su descarga. Los más conocidos son DivX, Windows Media Video (wmv), y Quicktime. (13)

1.6.8 Jitter.

Uno de los mayores problemas asociados a la entrega de video en tiempo real en redes IP es el asociado con la medida de tiempo entre el momento en que se espera que un paquete llegue y el momento en que efectivamente llega. Esta variación o diferencia de retardo entre paquetes es conocida como jitter, y es compensada normalmente a través del uso de un buffer de jitter en el receptor.

Este tipo de buffer puede ser pensado como un área de memoria usada para compensar los retardos en el flujo de paquetes a través de una red. Los paquetes tendrán que primero pasar al buffer para

luego ser extraídos en una secuencia de tiempo que minimice la diferencia de retardo entre un paquete y el otro. (5)

1.7 Descripción de algunos protocolos utilizados en IPTV.

Ningún estudio sobre IPTV puede aspirar a ser completo sin antes realizar al menos una pequeña descripción de los principales protocolos usados en estos sistemas. Esto incluye protocolos de transmisión IP como por ejemplo UDP y RTP, y además los llamados protocolos de señalización como RTSP e IGMP.

1.7.1 Protocolo de Internet (IP, Internet Protocol).

Es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados. Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas. IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente provee seguridad (mediante checksums o en español, sumas de comprobación) de sus cabeceras y no de los datos transmitidos; por tanto, el paquete podría llegar dañado, en otro orden respecto a otros paquetes, duplicado, o simplemente no llegar. (14)

A continuación en la Figura 2 se muestra el formato de un paquete IP:

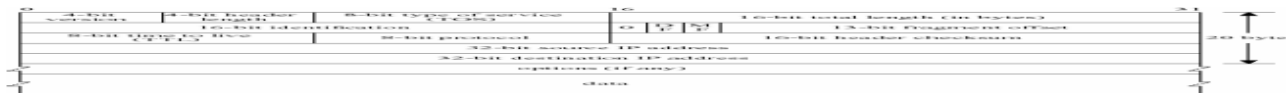


Figura 2: Formato de un paquete IP.

1.7.2 Protocolo de Datagramas de Usuario (UDP, User Datagram Protocol).

UDP es uno de los protocolos esenciales del paquete de protocolos de Internet. Como el protocolo IP, UDP tampoco tiene confirmación ni control de flujo. Proporciona una sencilla interfaz entre la capa de red y la capa de aplicación del modelo OSI¹⁰. Se utiliza por ejemplo cuando se necesita transmitir audio o video y resulta más importante transmitir con velocidad que garantizar el hecho de que lleguen absolutamente todos los bytes de información. UDP ha sido descrito por tanto, como el protocolo de “dispara y olvida”, por sus ya abordadas limitaciones. En un entorno IPTV, donde es

¹⁰ Open System Interconnection, es el modelo de red descriptivo lanzado en 1974 por la Organización Internacional para la Estandarización. Fue un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

esencial que la señal de video sea entregada confiablemente y en la secuencia correcta, el uso de UDP puede conducir a fallas en su implementación. (11)

A continuación en la Figura 3 se muestra el formato de una cabecera UDP:

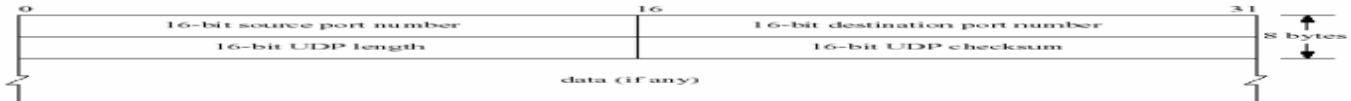


Figura 3: Cabecera UDP.

1.7.3 Protocolo de Transporte en tiempo Real (RTP, Real-time Transport Protocol).

RTP es conocido por ser un protocolo de nivel de sesión, y creado por la IETF (Internet Engineering Task Force, en español, Grupo de Trabajo en Ingeniería de Internet) como un formato basado en paquetes para la entrega de datos de audio y video. (15)

RTP en realidad consta de 2 partes estrechamente entrelazadas:

- RTP para proveer marcas de tiempo, numeración de secuencias, y otros mecanismos para atender problemas de “timing” o temporización. El uso de la numeración de paquetes permite que sean identificados los paquetes perdidos o fuera de la secuencia u orden establecido.
- Y el RTCP o protocolo de control de RTP, para monitorear la calidad del servicio (QoS) y para tener información acerca de los participantes en una sesión en curso.

A continuación se muestra en la Figura 4, el formato de la cabecera RTP:

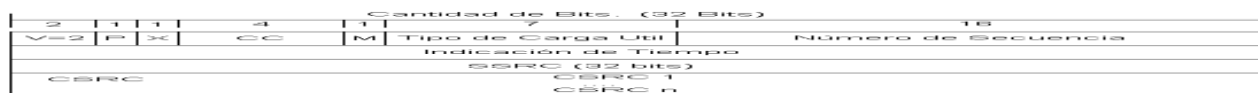


Figura 4: Cabecera RTP.

1.7.4 Protocolo de Control para el Transporte en tiempo Real (RTCP).

Es usado junto a RTP para la recepción de reportes estadísticos. Permite por ejemplo la detección de fallas en el árbol de distribución de multidifusión, número de paquetes perdidos, y estadísticas de jitter. Existen reportes emitidos por el receptor y por el emisor. Para amortizar el tamaño de la cabecera, este protocolo puede reunir varios mensajes RTCP y ser enviados en un mensaje RTCP compuesto. Los paquetes deberán estar compuestos al menos por un mensaje del receptor o emisor y el nombre canónico del participante y deben ser enviados periódicamente sin llegar a consumir el 5 % del ancho de banda de la sesión. (15)

Los paquetes RTCP se transportan sobre datagramas UDP, o sea que el paquete quedaría conformado como se muestra en la Figura 5:

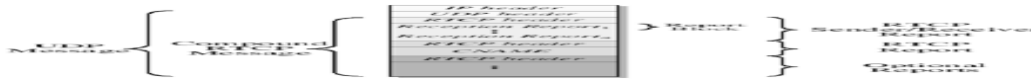


Figura 5: Capas en RTCP.

1.7.5 RTSP (Real Time Streaming Protocol).

Fue creado también por el IETF y describe una serie de controles semejantes a los que ofrece una videogradora (VCR) común. Típicamente, los mensajes RTSP son enviados del cliente al servidor, aunque existen excepciones donde ocurre inversamente, o sea del servidor al cliente. En sistemas IPTV, RTSP es usado en aplicaciones de video bajo demanda (VoD) para que el cliente acceda y controle el contenido almacenado en los servidores de VoD. VoD es en esencia una comunicación uno a uno, establecida usando unicast (unidifusión). Unicast es exactamente lo opuesto de broadcast o difusión, en la cual mandamos información a todos los usuarios de la red. Unicast permite que el servicio de VoD sea requerido y por tanto enviado a un único usuario. (11)

1.7.6 Protocolo de Gestión de Grupos de Internet (IGMP, Internet Group Management Protocol).

Definido por el IETF y creado para garantizar el multicast IP, el cual como ya se ha visto en este trabajo, se define como la transmisión de un datagrama IP a un grupo de hosts o PCs determinado. Este grupo de hosts está identificado por una única dirección IP de destino. En un sistema IPTV, el grupo de hosts sería un grupo de suscriptores que desean recibir un programa en particular.

En la práctica, lo que esto significa es que los sistemas de transmisión que usan IGMP no mandan todo el contenido a todos los usuarios de la red. El multicasting, usando IGMP, permite el control de qué contenidos van a cuáles usuarios, y por tanto, controla la cantidad de datos que está siendo enviada a través de la red en cualquier momento.

IGMP es el protocolo usado para manejar los cambios de canales en sistema IPTV. En respuesta a los comandos que envía el control remoto, una serie de comandos IGMP son ejecutados para dejar el actual multicast y unirse a otro servicio. (11)

1.8 Principales servicios de IPTV.

A continuación se hará alusión a algunos de los principales servicios que ofrecen la mayoría de los sistemas IPTV, servicios por los cuales esta tecnología se ha convertido en una realidad y necesidad para los clientes que, en número creciente, se unen a su uso diario.

1.8.1 Servicio de Broadcast TV.

El usuario tiene acceso a un número determinado de señales de video, de manera similar a la CATV o televisión por cable, pero la velocidad de cambio de canales (zapping) es más lento, puesto que se superponen dos efectos:

1. La necesidad de que el STB (del inglés Set Top Box) seleccione un grupo de multidifusión (multicast) correspondiente al canal que se desea recibir, dejando el grupo anterior (proceso de "leave" y "join" de un grupo de multicast a otro). Esto podría tomar varias centenas de milisegundos.

2. La detección de las tramas de video comprimido que permiten reconstruir la imagen, lo que puede ocurrir cada varios segundos. Este aspecto es el principal contribuyente al proceso de zapping, puesto que el video comprimido es eficiente si la información permite reconstruir el cuadro de imagen en función de la información acumulada (incremental), enviando cuadros de actualización de una imagen completa cada una cierta cantidad no menor de cuadros diferenciales.

De acuerdo al tipo de equipo de acceso (STB) que se instale en el receptor (pudiera ser, la casa del cliente), al ancho de banda disponible en la interfaz ADSL¹¹, y al tipo de acceso de cobre, el cliente podrá acceder a una, dos o tres señales de video simultáneamente.

El cliente también podrá configurar su servicio, determinando qué paquetes de señales quiere recibir, así como si desea recibir temporalmente alguna señal que no está incluida en el paquete contratado.

(6)

1.8.2 Servicio de Video bajo Demanda (VoD, Video on Demand).

La televisión bajo demanda propiamente dicha necesita de un canal de banda ancha dedicado mientras dura la descarga. El VOD puede permitir al cliente unas pequeñas funciones de vídeo para la repetición o salto limitados de determinadas secuencias (siempre que se disponga de línea de retorno), pero básicamente se centra en la oferta de un único visionado del producto audiovisual.

Se necesitan servidores VOD. Pero estos servidores de vídeo especiales tienen limitado el número de demandas. Por ello es esencial el control sobre la demanda posible en cada momento. Esto podría tener también cierta importancia en el anterior nivel, pero mucho menor ya que la descarga no se está visionando en el momento y lo único a lo que afectaría sería al tiempo de descarga. Pero en este nivel sí se visiona a la vez y podría verse afectado este visionado.

¹¹ Asymmetric Digital Subscriber Line. Es una tecnología de compresión de las señales de datos creada por la compañía AT&T, que proporciona alta velocidad de transferencia de datos en el actual par de hilos de cobre.

El sistema consiste en un almacén de datos de CM (Continuous Media) del que toma los datos el Servidor de VOD que los retiene en un buffer propio y distribuye adecuadamente a los buffers de los distintos clientes. (16) La existencia de buffers en cada cliente es usada para proveer de alguna tolerancia en las variaciones de tráfico en la red y en los índices de consumo de datos.

A continuación se muestra la arquitectura básica de un sistema típico VoD:

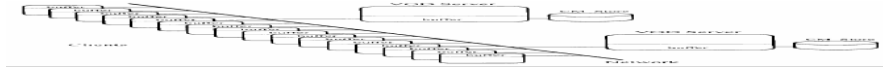


Figura 6: Arquitectura básica de un sistema típico VoD.

En sistemas IPTV, los clientes de servicios de VoD disponen entre otros, de una lista de películas, y eventos en vivo, de las cuales puede elegir las que desee (el costo de la misma se le cargará a la cuenta de usuario) para ver en el momento deseado.

1.9 Elementos de un sistema IPTV.

En un sistema IPTV común podremos identificar los siguientes bloques principales:

- ❖ Head End (HE) ó cabecera.
- ❖ Sistema de Video bajo Demanda (VoD).
- ❖ Middleware (MW) o Mediador.
- ❖ Sistema de control de derechos (DRM, Digital Rights Management).
- ❖ Set Top Box.
- ❖ Home Gateways (HG) o puertas de enlace.
- ❖ Servidores para la gestión de la red y de las direcciones IP de los STB, por ejemplo, servidores DHCP.

1.9.1 Head End.

Es el conjunto de elementos encargados de las siguientes funciones:

- ❖ Recepcionar las señales, ya sean locales o desde los satélites.
- ❖ Desencriptar las señales que se reciban cifradas.
- ❖ Codificar las señales de audio y video digitalmente incluyendo el DRM (Digital Right Management).
- ❖ Conformar los streams (flujos) digitales.
- ❖ Insertar propaganda y sobreimpresos.

- ❖ Multiplexar¹² las señales digitales en un único flujo que será distribuido en la red de acceso y transporte IP del operador. Cada canal de broadcast TV conformará un grupo IGMP propio al cual se unirán los STB que los deban recibir.

- ❖ Implementar la presentación al usuario final de un mosaico de canales.

- ❖ Posibilitar monitorear las señales en distintos sitios. (16)

Está constituido por antenas para recepción satelital, los receptores, los servidores de codificación / transcodificación, el sistema de ajuste de ancho de banda del flujo de información, y los equipos de encaminamiento que conforman los streams a insertar en la red de un operador¹³.

1.9.2 Servidores de Video.

Estas entidades son las que almacenan la información (archivos de video) que luego se distribuirá a los distintos clientes. Es un componente opcional del Head End. Son necesarios para dar soporte a los servicios de VoD, Near VoD, y PVR, no así para los servicios en los que se transmiten señales de broadcast como canales de TV, programación en vivo, entre otros, provenientes por ejemplo de un satélite. Deben poseer una gran capacidad de almacenamiento ya que deberán almacenar todas las películas y demás medios que serán puestos a disposición de los suscriptores. En general el almacenamiento se hace en arreglos de discos para brindar redundancia.

1.9.3 Middleware.

El Middleware es una plataforma de gestión de aplicaciones que interactúa con la red de acceso, el Head End y los STBs para permitir el aprovisionamiento y la distribución de servicios de televisión interactivos. Es una parte crítica de la solución IPTV extremo a extremo, o sea, constituye una suerte de “cerebro” de la red. Es una plataforma informática que provee las herramientas y tecnologías necesarias a los proveedores de servicios de banda ancha para ofrecer servicios de video haciendo uso de su infraestructura. El middleware interactúa con los STB por lo que ambos deberán contar con aplicaciones diseñadas en la modalidad cliente (en el STB) servidor (en el Middleware).

El Middleware es el responsable de asegurar la completa interoperabilidad del servicio de video. No está limitado a una única operación en el sistema, pero debe ser capaz de comunicarse directamente con cada componente para proveer soluciones de video. En el servicio de VoD, el usuario que desea ordenar una película para ser vista en determinado día y hora, realiza la solicitud al middleware,

¹² Se refiere a la acción que realizan los multiplexores de dividir el medio de transmisión en múltiples canales, para que varios nodos puedan comunicarse al mismo tiempo.

¹³ Operador se refiere en estos casos, a la compañía o institución que ofrece el servicio de IPTV.

estableciendo una sesión con éste. El middleware lo autorizará o no en función de la cuenta del usuario o su perfil. En caso afirmativo le ordenará al servidor de video que realice la transmisión de la película en el horario solicitado por el cliente.

Otras de sus funciones son, controlar la autenticación de los usuarios y manejar la información necesaria para la tarificación de los servicios, generando registros de consumo que serán procesados por otra entidad o sistema de facturación del operador. (16)

1.9.4 Sistema de Control de Derechos y de Acceso Condicional (DRM/CAS, Digital Rights Management/Conditional Access System).

En la actualidad es prácticamente imprescindible que una solución de IPTV cuente con un sistema de DRM (Digital Rights Management). Este sistema podría ser aplicado a los servicios VoD, AoD (Audio on Demand), broadcast TV y otros contenidos bajo demanda (ya sean almacenados o en vivo), de modo que se prevea la copia de la información digital y su posterior distribución.

El DRM implica un cierto cifrado del contenido multimedia, el cual luego puede ser reproducido si el receptor cuenta con la licencia correspondiente. La licencia es básicamente la clave para descifrar el contenido, esto evita la copia del contenido digital que se distribuye, puesto que el mismo nunca deja de estar cifrado por medio del DRM. Si un suscriptor copia el contenido digital que recibe a otra persona, si ésta no cuenta con la clave para descifrar el contenido no podrá reproducirlo en su sistema. (16)

1.9.5 Set Top Box (STB).

Estos son dispositivos de terminales de abonado, los cuales serán los que adapten las señales provenientes de la red de datos en señales visibles en un televisor convencional. Los STB presentan una interfaz Ethernet hacia la red de datos, que en general estará conectada al módem ADSL. En algunos casos los propios STB cuentan con el módem ADSL integrado. A su vez esos STB presentan una interfaz RCA o coaxial hacia el televisor, algunos además cuentan con interfaces de datos hacia el usuario.

Por lo general cuentan con un control remoto mediante el cual se seleccionan las distintas opciones, programación, por citar algunas, y en algunos casos incluyen o se les puede agregar un teclado inalámbrico el cual es útil en caso de realizar navegación en Internet desde la TV. Este STB, debe manejar el protocolo IGMP, ya que cuando se desea cambiar de canal se envía un mensaje IGMP denominado "Leave Group", en español dejar canal, hacia su servidor IGMP más próximo, y luego le enviará un mensaje llamado "Join Group" o en español unirse a un canal, con la dirección de

multicast del nuevo canal deseado, la cual obtiene de una lista que le envía el middleware al reiniciar el STB, y que es periódicamente actualizada.

Cuenta con un pequeño procesador el que corre una aplicación llamada EPG (Electronic Program Guide) generalmente sobre plataformas Linux o Windows. Ésta es la que da soporte para aplicaciones como un navegador de configuración del servicio, suscripción a canales, orden de películas en servicio de VoD, visualización del catálogo de películas, y control de acceso. (16)

1.9.6 Home Gateways.

El gateway es normalmente un dispositivo configurado para dotar a las máquinas de una red local (LAN) conectadas a él de un acceso hacia una red exterior, generalmente realizando para ello operaciones de traducción de direcciones IP (NAT: Network Address Translation). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada IP Masquerading (enmascaramiento de IP), usada muy a menudo para dar acceso a Internet a los equipos de una red de área local compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa. (16)

1.9.7 Servidores para la gestión de la red y de las direcciones IP de los STB.

Estos servidores son los encargados de gestionar cada uno de los elementos de la red y además son los encargados de asignar las direcciones IP dinámicas a los STB, para su correcto funcionamiento. (16)

1.10 Principales funcionalidades de un sistema IPTV.

A continuación se describen una serie de funcionalidades que son comunes en los sistemas IPTV:

- ❖ Guía electrónica de programación (EPG).
- ❖ Sistema de menús para la selección, búsqueda y compra de los videos bajo demanda.
- ❖ Sistema de grabación de video (PVR).
- ❖ Sistema de Time Shift TV (TSTV).
- ❖ Capacidad de presentar las señales (canales en vivo) en mosaico.
- ❖ Un sistema para control parental, y para compra de servicios como el de Video/Audio a demanda.
 - ❖ Contenidos con audio de alta calidad.
 - ❖ Servicios interactivos desarrollados por el operador o terceras partes que interactúan con el operador por medio de interfaces abiertas.
 - ❖ Otros servicios interactivos como: multilinguaje (subtítulos y audio), compras, integración con otras plataformas de mensajería, capacidad de navegación por Internet en el TV, y videoconferencia.

1.10.1 Guía Electrónica de Programación (EPG, Electronic Program Guide).

La guía electrónica de programación (EPG) es una agenda donde se presenta la programación prevista de los canales en vivo contratados. El alcance de la misma en general llega a ser hasta aproximadamente 14 días en adelante. Además de sustituir la revista mensual que se envía a los clientes con esta información, esta facilidad agiliza el cambio de canales y brinda una opción al zapping (cambio de canal).

Su presentación es configurada a medida por el operador, resultando en que a ésta se le puede ajustar a un diseño propio incluyendo publicidad o promociones. Navegando el EPG se puede acceder a detalles de la metadata¹⁴ de los programas actuales, pasados o futuros: duración, género, origen, director, actores entre otras cosas. (16)



Figura 7: EPG de Microsoft TV.

1.10.2 Grabación de Video personal (PVR, Personal Video Recorder).

El PVR, permite a los suscriptores del servicio de IPTV la grabación de programas en vivo para luego ser reproducidos posteriormente emulando las funcionalidades de un video grabador. Durante la reproducción se dispondrá las funciones de Play, Pause, Stop, REW (reversa, proveniente del inglés Rewind), FF (avance rápido, del inglés Fast Forward). Esta facilidad puede estar integrada al EPG, en ese caso se podrá programar la grabación marcando uno o más programas anunciados en la guía para ser almacenados en su momento de emisión. Por otra parte es útil que se disponga de un acceso rápido desde el control remoto para el comienzo de grabación en forma inmediata. (16)

1.10.3 Pagar por Ver (PPV, Pay Per View).

Mediante el PPV se brinda a lo suscriptores la facilidad de contratar la posibilidad de ver un programa en particular. Es una característica que generalmente es utilizada para eventos deportivos o conciertos que están siendo emitidos en vivo. La modalidad conjuga características de transmisión

¹⁴ Son datos que describen a otros datos.

de TV en vivo y exige una contratación del contenido como en los casos de VOD o NVOD (Near VoD). (16)

1.10.4 Time-Shift TV (TSTV).

El TSTV o pausa de TV en vivo es una novedosa funcionalidad que permite “congelar” una imagen de un programa en vivo o repetir (Replay) alguna escena que fue reproducida recientemente. Luego de la Pausa o Replay se puede continuar viendo el programa que se está transmitiendo en vivo pero desfasado en el tiempo de acuerdo a la duración de la pausa o replay que se hubiera hecho. Esta funcionalidad en general es accedida con un botón de acceso rápido del control remoto del STB.

Para implementar el TSTV se almacena en discos rígidos (en los STB o en una granja de servidores, dependiendo de la modalidad de trabajo que se hubiere elegido) un tiempo estipulado de la programación en vivo que se esté presenciando. Esa “porción de historia” de programa permitirá realizar un replay de lo ocurrido como máximo en ese tiempo almacenado o permitirá que durante la pausa o reproducción de replay se almacenen las escenas que se continúan emitiendo en el canal en vivo.

Cuando se reanuda la reproducción del programa, se continuará viendo el mismo donde se dejó. De aquí en adelante la programación emitida se almacena y se reproduce desplazado en el tiempo lo almacenado un tiempo atrás. Se puede disponer de la facilidad de adelanto para regresar a la reproducción en tiempo real. (16).

Para brindar todas estas funcionalidades, se hace necesario contar con la infraestructura adecuada, dígase tanto de red como de recursos hardware en general, para poder hacer un uso eficaz del ancho de banda y que el usuario nunca perciba retardos en la reproducción de video ni ruido en la señal, que afecten la calidad de servicio y de experiencia que caracterizan estos sistemas IPTV.

1.11 Características de la red de acceso de los suscriptores.

1.11.1 Acceso ADSL.

Estos servicios de transmisión de video y multimedia, en general serán soportados sobre accesos de banda ancha ADSL o ADSL2+¹⁵, sobre el par de cobre actual que llega al sitio del cliente. Se pueden alcanzar velocidades máximas de bajada teóricamente de 8 Mbps en el caso de ADSL y de 24 Mbps en el caso de ADSL2+, lo que en la realidad se ve disminuido por efectos tales como

¹⁵ ADSL2+ es una evolución del sistema ADSL y ADSL2 que se basa en un aumento del espectro frecuencial.

interferencias de otros servicios de datos o atenuación debida a la longitud de los cables o el estado de los mismos.

Para brindar este acceso se necesita contar con un módem ADSL o ADSL2+ en el domicilio del cliente, el cual se conecta mediante el par de cobre a un multiplexor¹⁶, en este caso uno denominado DSLAM¹⁷ (del inglés Digital Subscriber Line Access Multiplexer). Desde el DSLAM se llega mediante el núcleo (core) de datos, ya sea ATM o Ethernet, a los dispositivos centrales de la solución, como pueden ser servidores de video, middleware y codificadores.

Los módems ADSL mencionados tienen más de un puerto Ethernet de manera de poder servir al STB y a la PC o a la red interna del domicilio del abonado, de lo contrario se necesitará un dispositivo adicional como switch a la salida del módem al que se conectarán el STB y la PC. (16)

1.11.2 Normas de acceso ADSL.

La siguiente tabla ejemplifica las distintas normas de acceso xDSL disponibles comercialmente:

Tabla 1: Normas de acceso xDSL.

En la tabla 2 se presentan, a modo de resumen, situaciones factibles de despliegue de servicios. Se puede observar que utilizando la codificación de línea ADSL existen casos en los que no se podría dar servicio de video, independientemente de la codificación de video que se utilice. Por el contrario, utilizando la codificación de línea ADSL2+ existe gran probabilidad de poder brindar el servicio, aún cuando la codificación de video sea MPEG2¹⁸, contándose además con ancho de banda remanente suficiente para soluciones de acceso a Internet. (16)

Tabla 2: Situaciones factibles de despliegue de servicios.

¹⁶ Dispositivo que puede recibir varias entradas y transmitir las por un medio de transmisión compartido. Para ello lo que hace es dividir el medio de transmisión en múltiples canales, para que varios nodos puedan comunicarse al mismo tiempo.

¹⁷ Es un multiplexor localizado en la central telefónica que proporciona a los abonados acceso a los servicios DSL sobre cable de par trenzado de cobre.

¹⁸ Estándar de codificación de audio y video usado en HDTV.

1.12 Arquitectura de referencia de una red IPTV según ATIS¹⁹.

A continuación se muestra la arquitectura de referencia de una red IPTV según ATIS, de donde se ha tomado la siguiente figura:

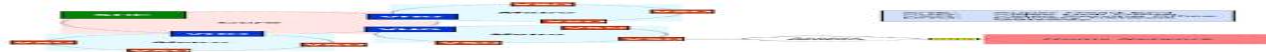


Figura 8: Arquitectura de red IPTV según ATIS.

En esta arquitectura se definen los siguientes puntos de referencia:

- ❖ SHE - Super Head End: como se vio en los módulos anteriores es el sitio donde se adquieren (reciben) las señales de broadcast desde diferentes medios, se agregan, se procesan, codifican y desde allí se distribuye al resto de la red. También puede ser un punto donde se realice un almacenamiento primario de contenidos a demanda.
- ❖ VHO – Video Hub Office: es el sitio desde donde se distribuyen los contenidos a grupos definidos de usuarios finales a través de la red de agregación (también denominada Metro). En este punto se ubican típicamente los servidores de video y se agrega publicidad.
- ❖ VSO – Video Serving Office: es el sitio desde donde se accede a los usuarios a través de las redes de acceso. En este punto se ubican típicamente los nodos de agregación y equipos de acceso (DSLAMs).
- ❖ DNG – Delivery Network Gateway: es el equipo de red que se instala en la casa del cliente. Tiene una interface WAN hacia la red del Proveedor de Servicio (típicamente Ethernet 10/100 Mbps que se conecta al equipo terminal del acceso,) y una interface LAN hacia casa del cliente. (16)

1.13 Situación problemática: las soluciones BSS/CRM en sistemas IPTV.

La necesidad de brindar servicios de valor agregado está transformando a las empresas proveedoras de telecomunicaciones, y cada vez más se preocupan y ocupan en tratar de atraer y retener clientes, y a la vez, entienden que la introducción de nuevos servicios y que puedan ser accedidos por una amplia gama de dispositivos es una necesidad creciente. Además, estos servicios

¹⁹ ATIS: Alliance for Telecommunications Industry Solutions, en español: Alianza para Soluciones de la Industria de Telecomunicaciones.

deberán ser entregados con la calidad y precio requeridos, para que de esta forma el cliente perciba que vale la pena pagar por el servicio que se le está brindando.

La llave del éxito en la industria de las telecomunicaciones hoy día reside en la agilidad y calidad con que se presten los servicios, para lograrlo, las empresas tendrán que lidiar no solo con la introducción de nuevos servicios, sino con las tareas de gestión y administración que permitan que el cliente en verdad reciba beneficios de esos nuevos servicios. (17)

Comprendiendo todos estos retos, los proveedores a nivel mundial trabajan para construir Plataformas de Entrega de Servicios (Service Delivery Platforms, SDP por sus siglas en inglés). Las SDP constituyen una arquitectura cada vez más promocionada por los desarrolladores de aplicaciones de valor agregado para servicios de telecomunicaciones. No existe un claro consenso sobre el alcance de esta nueva arquitectura, y la frontera entre las plataformas que incluye y otras que hacen al soporte del negocio aún no han quedado bien definidas. Frecuentemente se habla de “orquestación de servicios” como principal cometido de estas plataformas. (16)

El término BSS (Business Support Systems) es aplicado al grupo de herramientas de software que dan soporte a proveedores de servicios de telecomunicaciones, de información, y de servicios de redes en general. Estos sistemas brindan un servicio dentro de la SDP –uno que es crítico para el éxito de la introducción de nuevos servicios.

Se trata de fabricar aplicaciones con objetivos bien especificados, y apoyados en una filosofía CRM. CRM es un conjunto de estrategias de negocio, marketing, comunicación e infraestructuras tecnológicas, diseñadas con el objetivo de construir una relación duradera con los clientes, identificando, comprendiendo y satisfaciendo sus necesidades. CRM va más allá del marketing relacional, es un concepto más amplio, es una actitud ante los clientes y ante la propia organización, que se apoya en procesos multicanal (teléfono, Internet, correo, y fuerza de ventas) para crear y añadir valor a la empresa y a sus clientes.

Por otro lado, la consultora internacional Forrester Research, compañía independiente que se dedica a realizar investigaciones relacionadas con tecnologías y mercados, define CRM como "una aproximación, basada en la red, a la sincronización de las relaciones con el consumidor a través de canales de comunicación, funciones de negocios y audiencias". En otras palabras: sumar a Internet como canal a través del cual obtener y compartir información sobre el cliente. Este pretende extender el concepto de las relaciones personalizadas con el cliente a sistemas que operen en Internet.

El CRM facilita la importante tarea de mantener relaciones a largo plazo con los clientes, permitiendo efectuar marketing uno a uno, automatizar las fuerzas de ventas, brindar el mejor soporte a los clientes, optimizar campañas de marketing, telemarketing, soporte y servicio. Esta filosofía les

permite a los miembros de una empresa revisar la base de datos de un cliente y saber quiénes son sus contactos o cuál ha sido su historia. A través de catálogos interactivos en línea, es posible saber cuáles son las preferencias de los usuarios, realizar un seguimiento y constituir un historial de las consultas y requerimientos de los clientes. Así puede brindar una atención personalizada, reduciendo el tiempo de su ciclo de ventas y fomentando fidelidad en sus clientes. Las aplicaciones de e-CRM están pensadas para convertirse en un apoyo indispensable para los miembros de la empresa que están en contacto directo con clientes. La ventaja de este sistema es que no sólo permite administrar las bases de datos de los clientes, sino que se extiende hacia Internet, teléfono, correo, y fuerza de ventas. (18)

Considerando el modelo de base del TMForum²⁰ como base para las funciones de los sistemas BSS de un operador, los principales servicios a implementar son los que se detallan a continuación:

- ❖ Administración de órdenes: Es el servicio responsable de la manipulación de órdenes recibidas de los clientes y dispara la secuencia de operaciones requeridas para satisfacer dichas órdenes.

- ❖ Administración de inventario: Es el servicio responsable de proveer acceso a los recursos requeridos por el servicio. Tal es el caso de inventario de home gateways, y los STB, los cuales deberán estar debidamente catalogados e identificados. Además, en caso de que el servicio sea brindado en un área geográfica extensa, es requerido también una adecuada gestión de logística para asegurar la disponibilidad de los elementos fundamentales para el servicio lo más próximo que se pueda a los clientes.

- ❖ Facturación a los clientes: Este servicio es el que gestiona la emisión de facturas detalladas por utilización de servicios de IPTV, y se vincula directamente con los sistemas de CRM (gestión de la relación con los clientes por medio de un call center, página WEB, y sistema de conocimiento de las preferencias de los clientes) y Billing (Sistema de facturación del operador). En el caso de un operador de "Triple Play" el sistema de facturación de IPTV estará, generalmente, integrado con el sistema de facturación de los demás servicios, y cabe la posibilidad de que el sistema sea único para todos los servicios en la medida que los clientes se consideren de forma unificada. Para potenciar una oferta de Triple Play el sistema de facturación deberá poder implementar promociones cruzadas entre distintos servicios, resultando, por ejemplo en que el uso de un servicio brinde crédito sobre otro distinto por ejemplo.

²⁰ Telemanagement Forum. Organización fundada en 1988 por un grupo de empresas proveedoras de servicios que se dedica a desarrollar estándares para la implementación de sistemas OSS/BSS.

❖ Administración de los accesos a los contenidos: es la función de gestión del acceso a los contenidos multimedia. Estos contenidos pueden estar hospedados en las instalaciones del proveedor de servicio de IPTV, tal sería el caso de los contenidos a demanda, o pueden ser brindados a demanda por plataformas de terceras partes. En la IPTV sobre acceso de banda ancha los requerimientos de tiempo de respuesta y ancho de banda hacen poco viable que los contenidos se hospeden fuera de las premisas del operador, salvo aquellos que son recibidos y transcodificados a través del Head End. Para las aplicaciones que transcurren a través del middleware hacia los Set Top Boxes el soporte de aplicaciones de terceros se realizará, generalmente, a través de un portal (B2B). (16)

El otro aspecto relevante relacionado con la estrategia CRM es la personalización, la cual puede entenderse en tres sentidos diferentes:

❖ Todo contenido interesa a alguna persona que puede convertirse en un cliente potencial. Es necesario hacer accesible el contenido a la persona que pueda estar interesada por el mismo y ofrecerlo a un precio que el interesado esté dispuesto a pagar. La distribución tradicional se basa por el contrario en buscar el éxito masivo en ciertos contenidos y considerar el resto de contenidos como marginales.

❖ Cada cliente tiene un perfil personal. Las nuevas herramientas permiten mantener un registro histórico de las compras de un cliente y, a partir del mismo, poder ofrecerle un servicio de asesoramiento y recomendaciones personalizado. En la distribución tradicional, por el contrario, el asesoramiento está basado en “lo más leído”, “lo más visto”, “lo más jugado”.

❖ Las personas deciden el contenido. Mientras que en la producción de contenidos tradicional es la propia productora la que decide qué contenidos va a comercializar en función de lo que ofrecen los creadores, la distribución digital permite que el cliente final tenga una mayor capacidad de prescripción sobre los contenidos que se crean y distribuyen: el cliente no compra lo que le ofrecen, compra lo que quiere.

A continuación se presentan algunas de las soluciones IPTV de proveedores líderes a nivel mundial, para tener una idea de la magnitud y complejidad que conllevan ofrecer estas funcionalidades que para muchos países incluyendo nuestra bella isla todavía no son más que un sueño.

1.14 Análisis de soluciones existentes.

Huawei Technologies Co. Ltd.²¹

La solución IPTV de Huawei consiste en 6 módulos funcionales:

1. Gestión de suscriptores: El sistema permite la entrega de mensajes, que son usados para habilitar o inhabilitar un suscriptor, detener la suscripción a aquellos que deben los pagos acordados, habilitar nuevamente la suscripción una vez saldada esa deuda, y sincronizar la información referente al usuario con los otros sistemas relacionados.
2. Gestión de CP/SP: El sistema permite sincronizar la información CP o SP a otros sistemas.
3. Gestión de producto: Un producto se refiere a la unidad más pequeña que el suscriptor puede comprar, puede ser un programa, una columna, o una combinación de ambos. Estos productos son empaquetados en una amplia variedad de formas, precios, y estrategias de promoción.
4. Gestión de las cuentas: Se refiere al cargo de la experiencia del servicio, de acuerdo a los modos de cobro, estrategias de promoción, y a los precios de los productos. El sistema mantiene el historial de las cuentas por más de 6 meses y puede borrar la información repetida. El sistema permite además, el acceso a servicios prepagados (tarjetas de crédito) y deduce el monto en tiempo real. Los cuatro pasos para el cobro son: recoger la información de la cuenta, procesar la información, confirmar los precios, y entregar la factura.
5. Análisis estadístico: El sistema permite realizar consultas sobre SPs, CPs, productos, suscriptores, historial de autenticación de suscriptores, así como su comportamiento.
6. Administración del sistema: Incluye gestión de roles, administración de software y hardware (verificar el estado de esos componentes), backup de la base de datos (almacena la información en el medio designado, y la restaura de ser necesario dado un mal funcionamiento), administración de logs (almacenan la información de las operaciones que el sistema ejecuta). (19)

ZTE

ZhongXing Telecommunication Equipment Company Limited, es un proveedor global de equipamiento de telecomunicaciones y soluciones de redes.

Desde 1996 la compañía ha proporcionado productos y servicios en 135 países y regiones del Sureste asiático, Asia meridional, Norteamérica, Europa, Latinoamérica, África y la Comunidad de Estados Independientes. A través de una combinación de marketing estratégico, diferenciación de

²¹ Es el mayor fabricante de equipamiento de redes y telecomunicaciones en China y uno de los líderes mundiales en esta industria, dedica 10 % de sus ganancias a investigación y desarrollo. Cuenta con operadores en todo el mundo.

productos, mejoramiento de las ventajas de costo, optimización de los recursos humanos y manejo de la propiedad intelectual, ZTE ha establecido lazos comerciales con cerca de 500 operadores alrededor del mundo.

Ofrece un servicio de facturación flexible para todo el contenido de voz, datos, servicio de convergencia de prepago y postpago que soporta los servicios de las redes de nueva generación. Su arquitectura basada en eventos y su motor de facturación en tiempo real permiten poner al cliente en el centro del negocio y tener un paso adelante en lo que se refiere a competitividad. La gestión exacta de los clientes es la base del servicio de soporte. Completo entendimiento de las demandas y comportamiento de los clientes, especificando los valores de los clientes para la empresa, puede ser llevado a cabo recogiendo y analizando la información obtenida. La gestión se realiza de una forma unificada, provee múltiples canales para obtener y segmentar información relativa al cliente. Gestiona perfiles e historial de usuario, órdenes de productos, comportamientos de consumo, deudas de pago, entre otras. (20)

1.15 Situación de Cuba ante los retos de la TV Digital e IPTV.

Las dificultades a las que Cuba se tiene que enfrentar para el cambio a la TV Digital Terrestre son grandes y numerosas. El cambio es inevitable, así como lo ha sido en todo lo demás (teléfonos, prensa, cine). Se puede demorar, pero no se puede evitar. Mientras más demore, más difícil será después, porque habrá que hacerlo todo en plazos más apretados. El documento citado plantea que el país que comience la transición a lo digital ahora o en el futuro, se enfrenta a un problema adicional, que es el cambio total de tecnología que se está produciendo a nivel mundial de los televisores, que se dejan de fabricar en la forma que se conocen hoy, con Tubos de Pantalla (Tubo de Rayos Catódicos).

Además de los gastos adicionales individuales de todo el que posea un televisor, el país tendrá que destinar importantes recursos para instalar nuevos transmisores digitales, que tienen que estar en el aire al mismo tiempo que los actuales analógicos mientras se produce la transición. También tiene que invertir recursos en digitalizar al ICRT (Instituto Cubano de Radio y Televisión), que se encuentra en un estado de gran deterioro tecnológico, para que todo el proceso se pueda llevar a cabo.

Se plantea que sea cual sea la solución que se adopte, debe diseñarse para que en un futuro pueda convertirse en fuente de contenidos para posibles servicios de video bajo demanda de un sistema de IPTV. (21)

1.16 Conclusiones Parciales.

Con el estudio de la situación actual del dominio del problema, recogida en este capítulo, se profundizó en los conceptos relacionados con el proceso de funcionamiento de los sistemas IPTV,

explicando las condiciones actuales del desarrollo de éstas soluciones en Cuba y el resto del mundo, además se profundizó en el tema que comprende el campo de acción, o sea las soluciones BSS/CRM. Se hizo una breve historia de la televisión para aterrizar en ese mundo, en lo que se ha logrado, en las nuevas tecnologías y retos del presente y del futuro. También se abordó el tema de los protocolos usados para implementar sistemas IPTV, para tener conocimiento de su funcionamiento a nivel de red. Por otra parte, se hizo un estudio detallado de los diferentes elementos que componen estos sistemas, para poder ver la creciente necesidad de ofrecerle estos servicios al usuario final de la manera más sencilla y eficiente posible. Se ha hecho evidente la necesidad del diseño de un módulo BSS/CRM dentro del sistema IPTV que arroje la documentación necesaria para en el futuro desarrollar una correcta implementación de un sistema IPTV en la UCI, teniendo en cuenta las actuales circunstancias de competitividad del mercado y exigencias por parte del usuario final.

Tendencias y tecnologías actuales.

Introducción.

No es un secreto para nadie que la producción de software a nivel mundial crece aceleradamente dada la demanda de clientes que desean automatizar en algunos casos sus actividades diarias, o en otros simplemente reducir la cantidad de trabajo dedicada a esas actividades.

Tampoco es menos cierto que los múltiples procesos que conforman el desarrollo de un buen software son en su generalidad procedimientos difíciles de tratar, por lo cual se recomienda el empleo de metodologías especializadas en estas actividades, las cuales proporcionan una guía completa de trabajo donde no solo se implementa un producto determinado sino que se puede observar la evolución que va teniendo desde sus inicios hasta que se culmina el trabajo.

Aunque el uso de metodologías facilita el trabajo de los desarrolladores, también ha de tenerse en cuenta qué herramientas y de qué forma se puede obtener un buen modelado del problema. En el actual capítulo se reseña el estudio de diversas metodologías de desarrollo de software facultadas para llevar a cabo casi cualquier proceso de desarrollo de software. De igual forma se puede obtener información en cuanto a Herramientas CASE, unido al ya universal Lenguaje Unificado de Modelado (UML) que conforma un complemento ideal para el producto en cuestión.

2.1 Metodologías de desarrollo de software.

Las tendencias actuales exigen aplicaciones elaboradas bajo un proceso bien definido, minuciosamente gestionado y profundamente documentado, para cumplir con los parámetros requeridos en productos informáticos.

Un proceso de desarrollo de software puede hacerse tan complejo como el producto lo requiera, y muchas veces el control de dicho proceso se va de las manos de los desarrolladores debido a la no utilización de alguna metodología que guíe de manera coherente y eficiente todas sus actividades. Este proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos

requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo. Un proceso de software detallado y completo suele denominarse "Metodología". (22)

Por todo lo anterior y para evitar insatisfacciones de los clientes, aumento de presupuesto del producto, y mayor carga de trabajo por la parte de los desarrolladores, es que se aconseja basar todo el trabajo en una metodología de desarrollo de software, principalmente una que se adecúe a las características del proyecto, del ambiente de los desarrolladores, y del producto final como tal.

"Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, y guías para uso de herramientas de apoyo. Habitualmente se utiliza el término "método" para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño". (23)

2.1.1 Metodologías tradicionales de desarrollo.

Las metodologías tradicionales o pesadas de desarrollo de software se guían expresamente a través de un plan durante el proceso de desarrollo, en el cual se generan gran cantidad de artefactos y roles mediante un modelado y documentación detallados, apoyándose en poderosas herramientas de trabajo y notaciones definidas en la arquitectura del software.

"Este esquema tradicional para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del "buen hacer" de la ingeniería del software, asumiendo el riesgo que ello conlleva." (24)

Por otra parte, en estas metodologías los roles son más importantes que las personas, o sea, son partes reemplazables, los individuos no son importantes, solo los roles lo son.

MSF (Microsoft Solution Framework).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se

centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características: adaptable, es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar; escalable, puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más; flexible, es utilizada en el ambiente de desarrollo de cualquier cliente; tecnología agnóstica, porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. (25)

RUP (Rational Unified Process).

RUP (Rational Unified Process) es una metodología cuyo fin es entregar un producto de software. Se estructuran todos los procesos y se mide la eficiencia de la organización. Es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Describe cómo aplicar enfoques para el desarrollo del software, llevando a cabo pasos para su realización. Se centra en la producción y mantenimiento de modelos del sistema además de servir como guía de cómo usar UML.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada uno de ellos, estos a la vez se dividen en fases y donde se debe tomar una decisión importante. El ciclo de vida organiza las tareas en fases e iteraciones. A su vez, divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 9 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



Figura 9: Fases y Flujos de Trabajo según RUP.

Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se instala el producto en el cliente y se entrena a los usuarios. Surgen nuevos requisitos a ser analizados.

En cada una de sus fases pertenecientes a la estructura estática realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema. RUP brinda una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Pretende implementar las mejores prácticas en Ingeniería de Software, posee un desarrollo iterativo, administración de requisitos. Usa una arquitectura basada en componentes que posibilita el control de cambios. También ofrece un modelado visual del software así como la verificación de la calidad del mismo. El RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios. (22)

2.1.2 Metodologías ágiles de desarrollo.

Las metodologías ágiles se enfocan generalmente en intentar evadir las burocráticas guías de las metodologías tradicionales. Centran sus procesos de desarrollo a los individuos que interactúan con el proyecto así como con el equipo de trabajo, obteniendo buenos resultados, es decir, se desarrolla un buen software en cuanto a su funcionamiento pero no genera una documentación consistente.

Las metodologías ágiles están basadas en heurísticas provenientes de prácticas de producción de código, preparadas para cambios durante el proyecto y el cliente es parte del equipo de desarrollo. Los grupos de trabajo están diseñados con poco personal trabajando todos en el mismo sitio, generan pocos artefactos, poseen pocos roles, además de no hacer énfasis en la arquitectura del software. (24)

XP (Xtreme Programming).

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

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencia se han encargado

de dicha tarea. A continuación se presentan las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas. (24)

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

- Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- Estándares de programación. XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. Esto se ilustra en la Figura 10 donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo. (24)

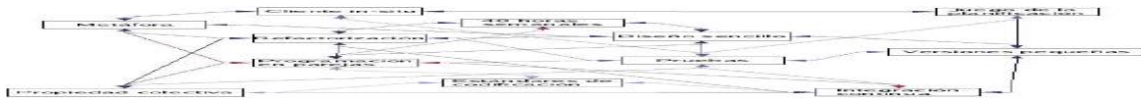


Figura 10: Las prácticas se refuerzan entre sí.

Concluyendo, en XP, el cliente tiene derechos como por ejemplo decidir qué se implementa, saber el estado real y el progreso del proyecto, añadir, cambiar o quitar requerimientos en cualquier momento, obtener lo máximo de cada semana de trabajo y obtener un sistema funcionando cada 3 o 4 meses. Además el desarrollador decide cómo se implementan los procesos, crea el sistema con la mejor calidad posible, pide al cliente en cualquier momento aclaraciones de los requerimientos, estima el esfuerzo para implementar el sistema y cambia los requerimientos en base a nuevos descubrimientos.

XP construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, lo que es más, combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada de entre todas las metodologías adaptables. (26)

SCRUM.

Scrum, más que una metodología de desarrollo software, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. Scrum ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro. Permite además seguir de forma clara el avance de las tareas a realizar, de forma que los "jefes" puedan ver día a día cómo progresa el trabajo. Sin embargo, Scrum no indica qué se debe hacer para hacer el código. Debería, por tanto, complementarse con alguna otra metodología de desarrollo. Se lleva bien con las metodologías ágiles y en concreto, con la programación extrema. (27)

Scrum no dice nada de si hacer o no hacer diseño, si hacer o no hacer pruebas unitarias, si hacer o no hacer documentación, o si trabajar en parejas o no. Scrum únicamente nos indica cómo conseguir que todos trabajen con el mismo objetivo, a corto plazo y deja bastante visible como avanza el proyecto día a día. (27)

Crystal Clear.

Crystal se define con mucho énfasis en la comunicación y de forma muy liviana en relación a los entregables. Crystal maneja iteraciones cortas con feedback frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios. Otra de las

cuestiones planteadas es la necesidad de disponer de un usuario real aunque sea de forma “part time” para realizar validaciones sobre la Interface del Usuario y para participar en la definición de los requerimientos funcionales y no funcionales del software.

La familia Crystal dispone un código de color para marcar la complejidad de una metodología: cuanto más oscuro un color, más “pesado” es el método. Cuanto más crítico es un sistema, más rigor se requiere. El código cromático se aplica a una forma tabular elaborada por Cockburn que se usa en muchas metodologías ágiles para situar el rango de complejidad al cual se aplica una metodología. En la Figura 11 se muestra una evaluación de las pérdidas que puede ocasionar la falla de un sistema y el método requerido según este criterio. Los parámetros son Comodidad (C), Dinero Discrecional (D), Dinero Esencial (E) y Vidas (L). En otras palabras, la caída de un sistema que ocasione incomodidades indica que su nivel de criticidad es C, mientras que si causa pérdidas de vidas su nivel es L. Los números del cuadro indican el número de personas afectadas a un proyecto.

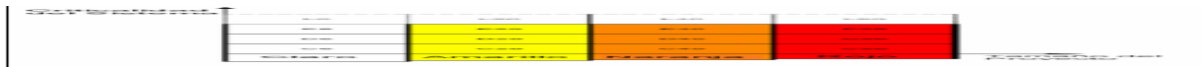


Figura 11: Familia de Crystal Methods.

Los métodos se llaman Crystal evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías: Crystal Clear (“Claro como el cristal”) para equipos de 8 o menos integrantes; Amarillo, para 8 a 20; Naranja, para 20 a 50; Rojo, para 50 a 100. Como casi todos los otros métodos, Crystal Clear consiste en valores, técnicas y procesos. Microsoft estableció la idea de los builds cotidianos, y no es una mala práctica. Muchos equipos ágiles compilan e integran varias veces al día. (29)

ASD (Adaptive Software Development).

James Highsmith, desarrolló ASD hacia el año 2000 con la intención primaria de ofrecer una alternativa a la idea de que la optimización es la única solución para problemas de complejidad creciente. Este método ágil pretende abrir una tercera vía entre el “desarrollo monumental de software” y el “desarrollo accidental”, o entre la burocracia y la adhocracia²².

Deberíamos buscar más bien, afirma Highsmith, “el rigor estrictamente necesario”; para ello hay que situarse en coordenadas apenas un poco fuera del caos y ejercer menos control que el que se cree necesario.

²² Es la ausencia de jerarquía, por tanto es lo contrario de burocracia.

ASD presupone que las necesidades del cliente son siempre cambiantes. La iniciación de un proyecto involucra definir una misión para él, determinar las características y las fechas y descomponer el proyecto en una serie de pasos individuales, cada uno de los cuales puede abarcar entre cuatro y ocho semanas. Los pasos iniciales deben verificar el alcance del proyecto; los tardíos tienen que ver con el diseño de una arquitectura, la construcción del código, la ejecución de las pruebas finales y el despliegue.

2.2 Lenguajes de modelado.

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos.

Muchas organizaciones los usan en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. (22)

2.2.1 Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (Unified Modeling Language, o UML por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. También documenta todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones). Este lenguaje de modelado es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (22)

2.3 Herramientas CASE.

Para superar las dificultades existentes del uso de las diversas tecnologías, la industria de computadoras ha desarrollado un soporte automatizado para el desarrollo y mantenimiento de software. Este es llamado Computer Aided Software Engineering (CASE). Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

Las Herramientas CASE fueron desarrolladas para automatizar procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software. La mejor razón para la creación de estas herramientas fue el incremento en la velocidad de desarrollo de los sistemas. Por esto, las compañías pudieron desarrollar sistemas sin encarar el problema de tener cambios en las necesidades del negocio, antes de finalizar el proceso de desarrollo. (30)

Enterprise Architect.

Es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea. El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible. (31)

Rational.

Rational es un agrupamiento de metodologías y herramientas que abarca todos los aspectos del desarrollo de software, desde su concepción hasta la elaboración del producto. La metodología de RATIONAL propone el desarrollo de software basado en las mejores prácticas recopiladas en innumerables proyectos exitosos. Esta metodología esta compilada en proceso de desarrollo Rational Unified Process.

Para ayudar en la implementación del proceso, Rational ha desarrollado un conjunto de herramientas integradas que permiten desarrollar las actividades del proceso y obtener una sinergia al integrarse entre las distintas herramientas permitiendo a todo el equipo de desarrollo compartir y utilizar la información necesaria en el momento adecuado.

Entre sus principales características se encuentran:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++.
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Capacidad de análisis de calidad de código.
- 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.
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación
- Integración con otras herramientas de desarrollo de Rational.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

Rational Rose Enterprise Edition es una herramienta que se puede encuadrar dentro del grupo de herramientas más técnicas debido a que se encarga de llevar a cabo tanto la automatización de los sistemas para la posterior generación de código (esto es, realización de los distintos diagramas y generación del código posterior), como para labores de ingeniería inversa, es decir, realización de los diagramas una vez conocido el código.

La clave está en la creación de componentes, los cuales van a contener una serie de archivos dentro de los cuales se encuentran las distintas clases pertenecientes a dicho componente. Mediante la especificación de la sintaxis que presentan dichos ficheros, se realiza de forma automática la ingeniería inversa. Se plantea que Rational Rose presenta una pequeña desventaja, y es que necesita de mucha memoria para poder de alguna forma ser manejado de forma rápida y eficiente.

(32)

Visual Paradigm.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

(33)

Otras de las características que presenta el VP son:

- Soporte ORM - Generación de objetos Java desde la base de datos

- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación
- Generador de informes para generación de documentación
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML
- Importación y exportación de ficheros XML.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

2.4 Observaciones del estudio de las tendencias y tecnologías.

Luego de realizar un estudio detenido y profundo de varias metodologías tanto ágiles como tradicionales y herramientas CASE que están a la vanguardia en el desarrollo y mantenimiento de software a nivel mundial, se hace necesario resaltar las características de cada una de ellas las cuáles las hacen más o menos recomendables para su uso en la elaboración de la documentación de este trabajo.

Por una parte, MSF está dedicada principalmente a proyectos de varias dimensiones, pero su debilidad reside en que se basa en el control de entregables por parte de los desarrolladores en vez de estar orientada a procesar una documentación. Por otro lado, XP se plantea el trabajo directo con el cliente, teniendo en cuenta sus necesidades y deseos, pero incluso usando otra metodología de apoyo como SCRUM, no se llega a llevar una traza de la documentación necesaria para desarrollar un sistema bien pensado, es más bien una combinación para un proyecto sin requisitos bien definidos y sin alcance claro. Lo mismo sucede con Crystal Clear y ASD, su perspectiva de trabajo es estar siempre en disposición de cambiar, y enfocados en resolver los riesgos más críticos que vayan surgiendo sobre la marcha, entregando funcionalidades parciales que en el mejor de los casos conformarían un producto completo e integrado.

Sin embargo, la metodología que más se acerca a lograr un desarrollo de software robusto y consistente es la RUP, por ser abarcadora en la guía de procesos de desarrollo de software y además por ser completa en la determinación de la documentación ingenieril y de su eficaz integración con el UML, siendo esta una poderosa alternativa. Aunque precisamente por el gran volumen de documentación que dicha metodología posee resulta un peligro para la motivación del

equipo de trabajo y la agilidad en la construcción del sistema. Pero RUP también presenta la cualidad de ser altamente configurable y adaptable a las exigencias del software en construcción, permitiendo reducir dicha documentación a los artefactos que solo sean del interés y uso del equipo desarrollador.

Por otro lado, se decidió que Visual Paradigm será la herramienta CASE con la que se le realizará el modelado de los procesos y artefactos arrojados por la metodología escogida, ya que es una herramienta potente que genera documentación de muy buena calidad, así como la posibilidad de trabajo con tablas de bases de datos y diagramas entidad relación. También porque soporta ingeniería inversa y generación de código. Por otra parte está el hecho de que en nuestra Universidad se cuenta con una licencia para su uso.

2.5 Conclusiones Parciales.

Tras analizar las distintas tendencias y tecnologías propuestas en el presente capítulo, con el objetivo de escoger cuál de las disímiles metodologías de desarrollo de software así como de la variedad de herramientas CASE existentes, han de utilizarse en la elaboración de la documentación ingenieril correspondiente al análisis de una solución BSS/CRM. A través de la observación de las particularidades de cada una de ellas, se ha emitido un firme criterio en cuanto a las que resultaron más cómodas y se ajustan de una forma u otra a las exigencias de dichas soluciones.

Luego de culminar con el enfoque general de lo que ha tratado el capítulo, se llega a la conclusión de que la metodología a utilizar por los desarrolladores de una solución BSS/CRM para un sistema IPTV en la UCI en las actividades de construcción del software será el RUP, así como el Visual Paradigm será la herramienta CASE con la que se le realizará el modelado de los procesos y artefactos arrojados por la metodología escogida.

CAPÍTULO 3

Análisis del sistema.

Introducción.

El modelado de análisis es independiente del paradigma de ingeniería a usar. Es realmente un conjunto de modelos que muestran las primeras presentaciones técnicas del sistema. Este modelo debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se puedan validar una vez que se construye el software.

Son conocidos los estudios realizados por investigadores donde alegan haber identificado los problemas del análisis y sus causas, además de las heurísticas para dar solución a los mismos. Cada método de análisis tiene su punto de vista y cada uno de ellos tiene relación con un conjunto de principios operativos:

- Debe representarse y entenderse el dominio de información de un problema.
- Deben definirse las funciones que debe realizar el software.
- Debe representarse el comportamiento del software (como consecuencia de acontecimientos externos).
- Deben dividirse los modelos que representan información, función y comportamiento de manera que se descubran los detalles por capas o jerárquicamente.

En el actual capítulo se hace referencia al análisis del sistema siguiendo la guía metodológica de RUP, de esta manera se podrá consultar la representación del Modelo de Dominio, el levantamiento de los requisitos funcionales y no funcionales de software, así como la representación del Diagrama de Casos de Uso del Sistema (DCUS) y la descripción detallada de los mismos.

3.1 El modelo de dominio en el desarrollo de software.

En la actualidad la intención de los desarrolladores de software es que los mismos cumplan normas y estándares que lo conviertan en escalables, sencillos de usar, y que permitan cambios sin que provoque efectos negativos en el producto. Un desarrollador o un equipo de desarrolladores tiene

que manejar una gran cantidad de conceptos, estilos de arquitectura, patrones de diseño e implementación, lenguajes, tecnologías, mejores prácticas y comunicación con los demás miembros del equipo de trabajo, además de la relación con los clientes, lo que puede ser una tarea engorrosa, debido al cúmulo de la información que se puede estar manejando en el mismo.

Lo que se denomina como Modelo de Dominio es empleado fundamentalmente cuando los flujos de información son difusos, es decir, que tengan múltiples orígenes y cuando son solo eventos o sucesos. También la imposibilidad de realizar subsistemas en el proceso de desarrollo del software dado por las múltiples interconexiones, es uno de los factores que influyen en la decisión de realizar un modelo de este tipo. Además de la existencia del solapamiento o de múltiples responsabilidades y la dificultad en el establecimiento de las reglas del funcionamiento del producto a implementar.

Un Modelo del Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, además de no incluir las responsabilidades de las personas que ejecutan las actividades. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo cual no significa que siempre que haya un Modelo de Dominio tenga que existir obligatoriamente un Modelo de Negocio. Por lo que el Modelo de Dominio no es más que la representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. (35)

3.1.1 Modelo de dominio BSS/CRM.

En el actual epígrafe se expone un análisis detallado de los objetos y eventos más significativos en el entorno del sistema. Teniendo en cuenta que no se tienen bien definidos los procesos del negocio se realizará una modelación del dominio y se procederá a explicar cada uno de los conceptos que forman parte del mismo. Todo ello para tener una mejor comprensión de la estructura y dinámica de la organización, los problemas actuales dentro de esta e identificar las mejoras potenciales.

Se muestra el modelo de dominio correspondiente al entorno donde estará el sistema teniendo en cuenta el proceso de funcionamiento de un módulo BSS/CRM, los conceptos, actividades, y personas involucradas en ello.

3.1.2 Diagrama de clases del modelo de dominio.



Figura 12: Diagrama de clases del modelo de dominio.

3.1.3 Definición de las clases del modelo de dominio.

Usuario.

Descripción de la clase: Persona que interactúa con el sistema IPTV para recibir o proveer contenidos audiovisuales.

Administrador

Descripción de la clase: Usuario que tiene la responsabilidad de velar por el buen funcionamiento del sistema, su mantenimiento, atención y apoyo a las necesidades de los usuarios. Además se encarga de gestionar cuentas y de realizar ofertas de programación.

Login.

Descripción de la clase: Mediante ella los usuarios van a autenticarse en el sistema para poder hacer uso de los servicios que en él se brindan.

Conexión.

Descripción de la clase: Permite establecer la conexión de las clases implicadas en lograr la conectividad de los usuarios con la aplicación y sus servicios.

Base de Datos.

Descripción de la clase: Es el conjunto de clases encargada de almacenar toda la información.

Suscripción.

Descripción de la clase: Contiene la información referente a una suscripción, dígame por ejemplo el identificador de usuario, la fecha en que se realizó, además, contiene la información del contenido que se le ofrece a determinado usuario.

Contenido.

Descripción de la clase: Es la clase que se encargará de facilitar la gestión de los contenidos a ofrecer al usuario, o sea, los archivos que estarán disponibles para ser consumidos, dígame series, películas y música por ejemplo.

Oferta_Contenido.

Descripción de la clase: Es la clase controladora. Contiene la lista de suscripciones. Por otra parte, guardará información relativa al historial de programas dada un usuario, para de esta manera ofrecer programación relacionada con lo anteriormente visto por el usuario.

3.1.4 Breve descripción del diagrama.

Con el sistema interactúan tanto invitados como usuarios registrados, dentro de este último grupo está el administrador, encargado de un número de funcionalidades que los usuarios invitados no tendrán acceso. Estos usuarios deberán autenticarse correctamente para poder hacer uso de sus privilegios, a su vez, se contará con una base de datos capaz de sustentar todo este proceso. Cuando un usuario selecciona o solicita algún servicio, es adicionado a una lista de suscripciones, la cual almacena también información sobre el contenido que se le ofrece a ese usuario. Se contará con una clase controladora que recoja una lista de suscripciones, y ofrezca información detallada

sobre el historial de programas de un usuario dado, además de ofrecer opciones de programación a partir de anteriores selecciones realizadas por los usuarios.

3.2 Requerimientos.

El proceso de construcción de un software está dado por disímiles actividades que han de ser llevadas a cabo por alguna guía metodológica, garantizando así la calidad del producto y la satisfacción de los clientes y desarrolladores: al mismo tiempo está dado también por técnicas y procedimientos que se llevan a cabo con el objetivo de realizar la documentación relacionada con cada uno de los flujos de actividades que tienen lugar durante la construcción del sistema.

Una etapa inicial y muy importante dentro del proceso de la Ingeniería de Software, es la Ingeniería de Requerimientos, donde se lleva a cabo el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones, del sistema de software que se desea producir; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requerimientos del software. Los requerimientos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar.

Para la obtención de los requerimientos de la solución BSS/CRM en cuestión, se hizo necesario acudir a varias técnicas relacionadas con dicho proceso como es la entrevista a varios líderes del proyecto. También se utilizó la técnica llamada lluvia de ideas así como la del enfoque grupal, donde cada uno de los integrantes del equipo de desarrollo expresa su punto de vista e ideas relacionadas con las funcionalidades que requiera el producto. Contando además con el previo estudio e investigación de software en los que de una forma u otra se desarrollan estos sistemas de apoyo personalizados.

Al lograr identificar los requerimientos iniciales del software y realizar el análisis de los mismos mediante la técnica de verificación, donde se limaron las ambigüedades y la falta de consistencia, permitieron llevar a cabo la selección de los requisitos que debe cumplir una solución BSS/CRM.

3.2.1 Requerimientos funcionales.

Presentan las funcionalidades que el sistema deberá ser capaz de realizar, incluyendo las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el mismo.

RF 1 Insertar Usuario.

El sistema debe permitir insertar nuevos usuarios.

RF 2 Eliminar Usuario.

El sistema debe permitir eliminar usuarios previamente almacenados en la Base de Datos.

RF 3 Modificar Usuario.

El sistema debe permitir modificar en cualquier momento la información que se tiene almacenada de un determinado usuario.

RF 4 Autenticar Usuario.

El sistema debe realizar un proceso de autenticación que permita distinguir a los usuarios, y de esta manera darle acceso y privilegios a servicios según corresponda.

RF 5 Suspende Usuario.

El sistema debe permitir suspender la cuenta a un usuario que previamente notificó que deseaba suspender su cuenta temporalmente.

RF 6 Reactivar Usuario.

El sistema debe permitir reactivar la cuenta de un usuario que anteriormente había sido suspendida.

RF 7 Cambiar Contraseña Usuario.

El sistema debe permitir cambiar la contraseña de un usuario, en caso de pérdida u olvido por parte del usuario.

RF 8 Ordenar servicio.

El sistema debe permitir que el usuario en cualquier momento seleccione los servicios que desea recibir de entre una lista que el sistema le ofrezca.

RF 9 Cancelar Servicio.

El sistema debe permitir al usuario cancelar algún servicio que haya estado recibiendo hasta el momento y ya no desee más.

RF 10 Ver Historial de Usuario.

El sistema debe permitir al usuario visualizar la información que el sistema almacena sobre su historial, últimas visitas, programas vistos, entre otros.

RF 11 Ofertar programación.

El sistema debe permitir que teniendo en cuenta el historial de programas de un usuario, el administrador pueda conformar una propuesta de programación de la cual el usuario seleccionará los que desee recibir.

3.2.2 Requerimientos no funcionales.

Los requerimientos 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. A continuación se definen los siguientes:

RNF 1 Usabilidad.

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de las funcionalidades que el sistema ofrece.

RNF 2 Requerimientos del Software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor.

Requerimientos mínimos para las PCs clientes:

- Un Navegador como Mozilla Firefox, Safari u otro que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows.

Para los servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior.
- PostgreSQL como Sistema Gestor de Base de Datos.

RNF 3 Requerimientos del Hardware

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 64 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

RNF 4 Confiabilidad

La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores. La información manejada por el sistema está protegida de acceso no autorizado y de divulgación a terceras personas.

RNF 5 Rendimiento

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, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

RNF 6 Restricciones de diseño

La aplicación deberá ser sencilla, intuitiva, amigable, debido a que los usuarios no son expertos en el uso de la misma. El producto de software final debe diseñarse sobre una arquitectura cliente-servidor. Deben emplearse los estándares establecidos (diseño de interfaces, base de datos y codificación), y lograrse un producto altamente configurable y extensible.

RNF 7 Portabilidad

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

RNF 8 Legales

El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

Como producto, los componentes BSS/CRM para sistemas 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.

RNF 9 Seguridad

- Los usuarios deben autenticarse antes de entrar al sistema.
- Garantizar el acceso controlado a la información, se mostrarán las interfaces a los usuarios, en dependencia de su nivel de acceso.

3.3 Descripción del sistema. Modelo de Casos de Uso del Sistema.

Luego de haber realizado anteriormente la descripción de los requisitos, quienes constituyen las funcionalidades esenciales del sistema, se le da paso a la siguiente etapa de construcción de un software que es la descripción del Modelo de Casos de Uso del Sistema (MCUS). El MCUS describe un sistema en cuanto a su utilización, es un modelo que contiene actores, casos de uso y las relaciones que se establecen entre ellos.

Los requerimientos del software se agrupan según sus características, utilidad y usabilidad conformando así los llamados casos de uso, los cuales no son más que fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Estos casos de uso describen cómo se comporta y funciona un sistema específico, mediante un documento que narra de manera secuencial las acciones relevantes que realiza un actor en interacción con el sistema hasta completar un proceso, sin darle importancia a los detalles de la implementación.

Los actores se definen como los roles que puede tener un usuario, pueden ser humanos, otros sistemas, máquinas, o hardware que interactúan con un sistema para de esta forma intercambiar datos, aunque en algunos casos pueden constituir un recipiente pasivo de información. Un actor no es parte del sistema en desarrollo, es un agente externo que interactúa con el mismo en pos de obtener un resultado esperado. El componente BSS/CRM para sistemas IPTV cuenta con dos actores que se especifican en la siguiente tabla:

Actor	Descripción
Usuario	Es la persona cuyos permisos se restringen a gestionar los servicios que desea recibir, ver su historial de programación.
Administrador	Es la persona o sistema autenticado cuyos permisos abarcan la gestión de usuarios, de cuentas, y ofertar programación.

Tabla 3: Descripción de los actores del sistema.

3.3.1 Modelo de Casos de Uso del Sistema.

A partir de la identificación de los actores que interactúan con la aplicación, así como la recopilación del conjunto de funcionalidades escritas en forma de requerimientos que a su vez han sido agrupados en casos de uso según sus peculiaridades, se conforma el Modelo de Casos de Uso que se representa a continuación:

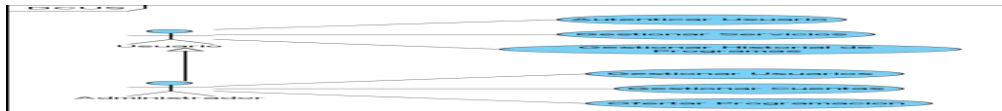


Figura 13: Diagrama de Casos de Uso del Sistema.

3.4 Descripción textual de Casos de Uso del sistema.

Cada uno de los casos de uso identificados presenta características particulares que para su mayor entendimiento se hace necesario describirlas textualmente. Además que para lograr un mejor rendimiento del tiempo en la construcción del software es recomendable identificar los casos de uso arquitectónicamente significativos, que en la descripción textual se diferencian porque tienen la prioridad de crítico.

3.4.1 Gestionar Usuarios.

Caso de Uso:	Gestionar Usuarios
Actores:	(Administrador)
Resumen:	El caso de uso permite insertar, eliminar, modificar o cambiar la contraseña de un usuario, registrándose los cambios en la Base de Datos.
Precondiciones:	Que el administrador se haya autenticado como tal.
Referencias	RF 1, RF 2, RF 3, RF 7.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema solicita	1.1. El sistema muestra la interfaz para

visualizar interfaz para insertar, eliminar, modificar o cambiar contraseña de un usuario.

insertar, modificar, eliminar o cambiar contraseña de un usuario:

- a) Si se desea insertar un usuario, ir a la sección "Insertar Usuario".
- b) Si se desea modificar un usuario, ir a la sección "Eliminar Usuario".
- c) Si se desea eliminar un usuario, ir a la sección "Modificar Usuario".
- d) Si se desea cambiar la contraseña de un usuario, ir a la sección "Cambiar Contraseña"

Sección "Insertar Usuario"

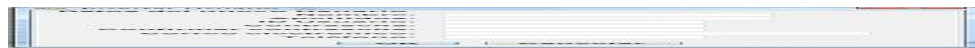
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema entra los datos del usuario para luego solicitar su inserción en el sistema.	1.1. Comprueba la validez de la información. 1.2. Busca el usuario en la Base de Datos para verificar que no exista. 1.3. Almacena los datos del nuevo usuario en la Base de Datos. 1.4. Muestra mensaje indicando que se ha creado y guardado correctamente el nuevo usuario.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	1.2 Muestra mensaje advirtiendo el error en la información.
	1.3 Muestra un mensaje advirtiendo que el usuario existe.

Prototipo de Interfaz

Interfaz 1



Sección "Modificar Usuario"

Acción del Actor	Respuesta del Sistema
2. El administrador del sistema introduce el usuario que desea modificar.	2.1. Verifica existencia del usuario especificado. 2.2. Muestra información del usuario.
3. Modifica los parámetros deseados y solicita modificar al usuario.	3.1. Verifica que esté correcta la información. 3.2. Verifica que el usuario a modificar no se encuentre autenticado. 3.3. Se actualiza el usuario en la Base de Datos y se finaliza el caso de uso.

Flujos alternos

Acción del Actor	Respuesta del Sistema
	2.2. Se emite mensaje de error porque no existe el usuario.
	3.2. Se emite mensaje de error en la información que se desea modificar. 3.3. Muestra mensaje informando que el usuario a modificar se encuentra actualmente autenticado en el sistema y termina el CU redirigiendo a la página principal.

Prototipo de Interfaz

Interfaz 2



Sección “Eliminar Usuario”

Acción del Actor	Respuesta del Sistema
2. El administrador del sistema introduce el usuario a eliminar y selecciona validar la información.	2.1. El sistema verifica existencia del usuario especificado. 2.2. Muestra la información del usuario.
3. Selecciona Eliminar el usuario.	3.1. El sistema pide la confirmación para eliminar. 3.2. Verifica que el usuario no se encuentre autenticado. 3.3. Se elimina el usuario, se actualiza la Base de Datos y finaliza el CU.

Flujos alternos

Acción del Actor	Respuesta del Sistema
	2.2. Se emite mensaje de error porque no existe el usuario.
	3.2. El usuario permanece en el sistema y se finaliza el caso de uso.
	3.3. Muestra mensaje informando que el usuario a modificar se encuentra actualmente autenticado en el sistema y termina el CU redirigiendo a la página principal.

Prototipo de Interfaz

Interfaz 3



Sección “Cambiar Contraseña”

Acción del Actor	Respuesta del Sistema
1. El administrador del sistema introduce el usuario que desea cambiar la contraseña.	1.1 Verifica existencia del usuario especificado. 1.2 Muestra interfaz para introducir la nueva contraseña.
2. Entra el dato y hace clic en Establecer nueva contraseña.	2.1 Verifica que esté correcta la información. 2.2 Verifica que el usuario a modificar la contraseña no se encuentra autenticado. 2.3 Se actualiza la nueva contraseña en la Base de Datos y finaliza el CU. 2.4 Se envía un correo al usuario informando la nueva contraseña solicitada.

Flujos alternos

Acción del Actor	Respuesta del Sistema
	1.2 Se emite mensaje de error porque no existe el usuario.
	2.2 Se emite mensaje de error en la información que se desea modificar.
	2.3 Muestra mensaje informando que el usuario a modificar se encuentra actualmente autenticado en el sistema y termina el CU redirigiendo hacia la página principal.

Prototipo de Interfaz

Interfaz 4



Poscondiciones

Es cambiado el estado del usuario en el sistema.

Tabla 4: Descripción del CU Gestionar Usuario.

3.4.2 Autenticar Usuario.


Caso de Uso:	Autenticar Usuario.	
Actores:	(Usuario).	
Propósito	El caso de uso se lleva a cabo con el objetivo de que el usuario entre al sistema y pueda hacer uso de los privilegios que se le ofrecen según su nivel de acceso.	
Precondiciones:	El usuario debe existir en la Base de Datos.	
Referencias	RF 4	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita entrar al sistema.	1.1 Muestra la interfaz de Autenticar Usuario.	
2. Llena los campos de usuario y contraseña y solicita entrar al sistema.	2.1 Verifica existencia del usuario en la Base de Datos. 2.2 Verifica que los datos proporcionados están correctos. 2.3 Muestra interfaz correspondiente al usuario en función del nivel de acceso y permisos.	
Prototipo de Interfaz		
Interfaz 5		
		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	2.2 Se emite mensaje de error porque no existe el usuario.	
	2.3 Se emite mensaje de error en la información que se desea modificar.	

Tabla 5: Descripción del CU Autenticar Usuario.

3.4.3 Gestionar Cuentas.

Caso de Uso:	Gestionar Cuentas.
Actores:	(Administrador)

Resumen:	El caso de uso se inicia cuando el actor suspende temporalmente la cuenta de un usuario que así lo solicitó o reactiva una cuenta ya suspendida, y finaliza con la suspensión o reactivación de los servicios que se le brindan a ese usuario, dependiendo del caso.
Precondiciones:	Que el administrador se haya autenticado como tal.
Referencias	RF 5, RF 6.
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El administrador del sistema solicita visualizar interfaz para suspender o reactivar a un usuario.	1.1. Muestra la interfaz para suspender o reactivar a un usuario: a) Si se desea suspender un usuario, ir a la sección "Suspender Usuario". b) Si se desea reactivar un usuario, ir a la sección "Reactivar Usuario".

Sección "Suspender Usuario"

Acción del Actor	Respuesta del Sistema
2. El administrador del sistema entra los datos del usuario para luego solicitar su suspensión.	2.1. Busca el usuario en la Base de Datos para verificar que exista. 2.2. Agrega al usuario a la lista de usuarios suspendidos. 2.3. Muestra mensaje indicando que se ha suspendido y guardado correctamente el estado del usuario.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.2 Muestra un mensaje advirtiendo que el usuario no existe.

Prototipo de Interfaz

Interfaz 6



Sección "Reactivar Usuario"

Acción del Actor	Respuesta del Sistema
2. El administrador del sistema introduce el usuario que desea reactivar.	2.1 Busca el usuario en la Base de Datos para verificar que exista. 2.2 Verifica que el usuario se encuentra en estado suspendido. 2.3 Elimina al usuario a la lista de usuarios


	suspendidos. 2.3 Muestra mensaje indicando que se ha reactivado y guardado correctamente el estado del usuario.
Flujos alternos	
Acción del Actor	Respuesta del Sistema
	2.2 Muestra un mensaje advirtiendo que el usuario no existe.
	2.3 Muestra mensaje indicando que el usuario no se encuentra en estado suspendido.
Prototipo de Interfaz	
Interfaz 7	
	

Tabla 6: Descripción del CU Gestionar Cuentas.

3.4.4 Gestionar Servicios.

Caso de Uso:	Gestionar Servicios.
Actores:	(Usuario)
Resumen:	El CU tiene por objetivo darle al usuario la posibilidad de en cualquier momento ordenar la programación de su preferencia, así como cancelar la que no sea más de su agrado.
Precondiciones:	Que el usuario se haya autenticado correctamente.
Referencias	RF 8, RF 9.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario solicita visualizar interfaz para ordenar o cancelar servicios.	1.1. Muestra la interfaz para ordenar o cancelar servicios: c) Si se desea ordenar servicios, ir a la sección "Ordenar Servicio". d) Si se desea cancelar servicios, ir a la sección "Cancelar Servicio".
Sección "Ordenar Servicio"	
Acción del Actor	Respuesta del Sistema
2. Solicita la opción de ordenar servicios.	2.1. Muestra la lista de servicios disponible.

3. Selecciona los servicios y solicita que le sean asignados.	3.1. Verifica que se ha seleccionado algún servicio 3.2. Guarda la información de los servicios solicitados. 3.3. Muestra mensaje confirmando que los servicios han sido seleccionados y asignados a su cuenta. 3.4. Se redirige al actor a la página principal.
---	---

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. Selecciona cancelar la orden de servicios.	3.1 Se redirige al actor a la página principal.
	3.2 Muestra un mensaje advirtiendo que no se ha seleccionado ningún servicio.

Prototipo de Interfaz

Interfaz 8



Sección "Cancelar Servicio"

Acción del Actor	Respuesta del Sistema
2. Selecciona los servicios y solicita que le sean cancelados.	2.1. Verifica que se ha seleccionado algún servicio. 2.2. Cancela los servicios solicitados, o sea, elimina esos servicios de la lista de servicios que tiene dicho usuario. 2.3. Muestra mensaje confirmando que los servicios han sido seleccionados y cancelados correctamente. 2.4. Se redirige al actor a la página principal.

Flujos alternos

Acción del Actor	Respuesta del Sistema
2. Selecciona que ya no desea cancelar ningún servicio.	2.1 Se redirige al actor a la página principal.
	2.2 Muestra un mensaje advirtiendo que no se ha seleccionado ningún servicio.

Prototipo de Interfaz

Interfaz 9



Tabla 7: Descripción del CU Gestionar Servicios.

3.4.5 Gestionar Historial de Programas.

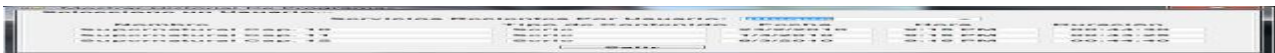
Caso de Uso:	Gestionar Historial de Programas.	
Actores:	(Usuario).	
Propósito	El caso de uso se lleva a cabo con el objetivo de que el actor pueda ver en cualquier momento los programas recientes a los que ha tenido acceso.	
Precondiciones:	El actor está correctamente autenticado.	
Referencias	RF 10.	
Prioridad	Crítico.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. Selecciona la opción de Ver Historial de Programas.	1.1 Muestra una lista de los programas vistos recientemente por el usuario.
	2. Hace clic en OK.	2.1. Redirige hacia la página principal.
Prototipo de Interfaz		
Interfaz 10		
		
Flujos Alternos		
	Acción del Actor	Respuesta del Sistema

Tabla 8: Descripción del CU Gestionar Historial de Programas.

3.4.6 Ofertar Programación.

Caso de Uso:	Ofertar Programación.
Actores:	(Administrador).
Propósito	El caso de uso se lleva a cabo con el objetivo de brindar al usuario una serie de ofertas de programación basadas en los anteriores programas seleccionados por el mismo.
Precondiciones:	El administrador está correctamente autenticado, y el usuario al que


	se le ofertará el servicio está autenticado actualmente en el sistema.
Referencias	RF 8, RF 11
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador solicita la interfaz “Formar Programación Para Usuario ”	1.1 Muestra la interfaz para conformar una programación para un usuario, basado en el historial de programas de dicho usuario.
3. El administrador introduce los datos del usuario.	2.1 Verifica que el usuario existe. 2.2 Muestra una interfaz que contiene una serie de programas relacionados con los que el usuario tiene en su lista de programas.
3. Selecciona uno o más programas de la lista y da clic en OK.	3.1 Muestra los nuevos servicios seleccionados, disponibles para el usuario en una ventana emergente equivalente al de la interfaz “Ordenar Servicio”.
Prototipo de Interfaz	
Interfaz 11	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.2 Muestra un mensaje advirtiéndole que el usuario no existe en la Base de Datos.

Tabla 9: Descripción del CU Ofertar Programación.

3.5 Diagramas de clases del análisis.

Los diagramas de clases del análisis representan la relación entre las clases que intervienen en los casos de uso. Proporcionan una vista interna del sistema. Es utilizado para comprender de forma general la estructura del sistema y sirve como entrada para la etapa de diseño.

3.5.1 DCA CU Autenticar Usuario.



Figura 14: DCA CU Autenticar Usuario.

3.5.2 DCA CU Gestionar Usuario.



Figura 15: DCA CU Gestionar Usuario.

3.5.3 DCA CU Gestionar Cuentas.



Figura 16: DCA CU Gestionar Cuentas.

3.5.4 DCA CU Gestionar Servicios.



Figura 17: DCA CU Gestionar Servicios.

3.5.5 DCA CU Gestionar Historial de Programas.



Figura 18: DCA CU Gestionar Historial de Programas.

3.5.6 DCA CU Ofertar Programación.



Figura 19: DCA CU Ofertar Programación.

3.6 Diagramas de colaboración del análisis.

3.6.1 Diagrama de Colaboración CU Autenticar Usuario.



Figura 20: Diagrama de Colaboración CU Autenticar Usuario.

3.6.2 Diagrama de Colaboración CU Gestionar Usuario.



Figura 21: Escenario Insertar Usuario.



Figura 22: Escenario Eliminar Usuario.

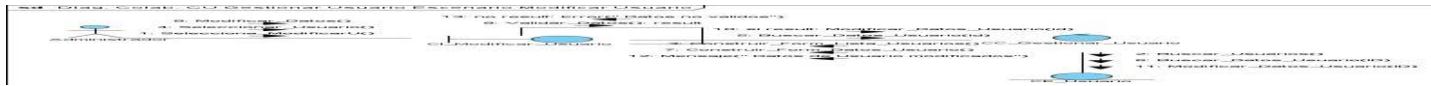


Figura 23: Escenario Modificar usuario.



Figura 24: Escenario Cambiar Contraseña.

3.6.3 Diagrama de Colaboración CU Gestionar Cuentas.



Figura 25: Escenario Reactivar Usuario.



Figura 26: Escenario Suspender Usuario.

3.6.4 Diagrama de Colaboración CU Gestionar Servicios.



Figura 27: Escenario Ordenar Servicio.



Figura 28: Escenario Cancelar Servicio.

3.6.5 Diagrama de Colaboración CU Gestionar Historial de Programas.



Figura 29: Diag. Colab. CU Gestionar Historial de Programas.

3.6.6 Diagrama de Colaboración CU Ofertar Programación.



Figura 30: Diag. Colab. CU Ofertar Programación.

Conclusiones Parciales.

En el presente capítulo se ha tratado de manera específica los artefactos generados en el análisis correspondiente según lo planteado por RUP, lo cual representa la metodología escogida para llevar a cabo los procesos ingenieriles referentes al desarrollo del componente BSS/CRM para un sistema IPTV.

Además de abundar en el proceso de construcción del Modelo de Dominio, así como su diagrama y descripción de las clases, se expuso la importancia del levantamiento de los requerimientos empleados en la construcción de un software, y se muestran los requisitos funcionales y no funcionales que presenta el producto. También se muestra, a través del análisis, el Modelo de Casos de Uso del Sistema, la descripción detallada de los mismos, y los diagramas de clases correspondientes al análisis, así como los de colaboración, con el objetivo de que el lector se lleve una idea general de las funcionalidades que puede presentar una solución BSS/CRM robusta, eficiente y que tiene al cliente como centro del negocio.

CAPÍTULO 4

Diseño del sistema.

Introducción.

En el siguiente capítulo se realizará el diseño del sistema, de esta forma se traducirán los requisitos a una especificación que describe cómo implementar el sistema, es decir, se transforman los requisitos en un diseño para el futuro sistema utilizando una arquitectura correcta, logrando adaptar el diseño para que sea consistente con el entorno de implementación.

4.1 El diseño de software.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. En el diseño se modela el sistema y determina su forma para que soporte todos los requerimientos, incluyendo los no funcionales. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requerimientos. Los propósitos del diseño se definen en transformar los requerimientos en un diseño de cómo el sistema debe ser, desarrollar una robusta arquitectura del sistema y adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

4.2 Descripción de la arquitectura del sistema.

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema. Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La arquitectura de software establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. La arquitectura define un conjunto de elementos, conectores, restricciones y un sistema de control que caracterizan a un sistema o a una familia de sistemas. Es por estas razones que se hace imprescindible seguir algún patrón de arquitectura a la hora de diseñar un sistema de software, en el

caso que nos ocupa, se trata del patrón Modelo-Vista-Controlador (MVC, proveniente del inglés Model-View-Controller).

4.2.1 Patrón de arquitectura MVC.

MVC tiene la intención de desacoplar el acceso a datos y la lógica de negocio (Modelo) de la presentación de los datos y la interacción con el usuario (Vista). Para ello, MVC introduce un componente intermedio, el Controlador. Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

También permite que no haya alguna dependencia directa entre la Vista y el Modelo, esto es de gran beneficio ya que brinda un mayor soporte a los cambios realizados en las diferentes interfaces existentes; que tienden a cambiar con más frecuencia que la lógica del negocio. Es importante mencionar que con el uso de este patrón se puede añadir nuevas vistas sin afectar el Modelo, es decir, múltiples páginas en la aplicación web pueden emplear los mismos elementos del Modelo.

Definición de las partes.

Modelo: Encapsula los datos y las funcionalidades. Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

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

Controlador: Recibe las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

En la siguiente figura se pueden apreciar claramente la estructura del patrón MVC:



Figura 31: Estructura del patrón MVC.

Este modelo de arquitectura presenta las siguientes ventajas:

- Hay una clara separación entre los componentes de un programa, lo cual permite implementarlos por separado.
- La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- Es posible tener diferentes vistas para un mismo modelo.

- Es posible construir nuevas vistas sin necesidad de modificar el modelo.

En el estilo arquitectónico MVC se fomenta la escalabilidad del sistema, la seguridad y la reutilización de código.

Elementos del patrón MVC utilizados en este trabajo:

- Modelo: datos y funcionalidades del sistema.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las funcionalidades del sistema.
- Lleva un registro de las vistas y controladores del sistema.
- Si se está ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El controlador es responsable de:

- Recibe los eventos de entrada.
- Contiene reglas de gestión de eventos.

Las vistas son responsables de:

- Reciben datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de “Actualización ()”, para que sea invocado por el controlador o por el modelo.

4.3 Patrones de diseño.

Un patrón de diseño es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, es un proyecto o estructura de implementación que logra una finalidad determinada, un lenguaje de programación de alto nivel, una manera más práctica de describir ciertos aspectos de la organización de un programa.

Para el diseño del sistema que se propone se utilizaron los patrones de asignación de responsabilidades (del inglés GRASP, General Responsibilities Assignment Software Patterns):

Alta Cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Es un principio que se debe tener

presente en todas las decisiones de diseño: es la meta principal que ha de buscarse en todo momento. Brinda como solución el diseño de una clase con responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.

Esto hace bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. Soporta un aumento de la capacidad de reutilización. El patrón alta cohesión presenta semejanzas con el mundo real. Se sabe que si alguien asume demasiadas responsabilidades, sobre todo las que debería delegar, no sería eficiente.

Bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. El bajo acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad.

Experto: Brinda como solución asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa la idea de que los objetos hacen cosas relacionadas con la información que poseen.

Creador: Brinda como solución asignarle a una clase B la responsabilidad de crear una instancia de clase A. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

4.4 Diagramas de clases del diseño.

Los diagramas de clases del diseño son una representación más concreta y detallada que los diagramas de clases del análisis, aunque también representan la parte estática del sistema conteniendo las clases y sus relaciones.

En el diseño se modela el sistema y se encuentra su forma, incluyendo la arquitectura, para que soporte todos los requisitos y otras restricciones relacionadas con el entorno de la implementación, tiene impacto en el sistema a desarrollar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es utilizado como entrada fundamental de las actividades de implementación.

A continuación se presentan los diagramas de clases del diseño elaborados para el sistema que se propone en este trabajo:

4.4.1 Realización del CU Autenticar Usuario.



Figura 32: DCD CU Autenticar Usuario.

4.4.2 Realización del CU Gestionar Usuario.



Figura 33: DCD CU Gestionar Usuarios.

4.4.3 Realización del CU Gestionar Cuentas.



Figura 34: DCD CU Gestionar Cuentas.

4.4.4 Realización del CU Gestionar Servicios.



Figura 35: DCD CU Gestionar Servicios.

4.4.5 Realización del CU Gestionar Historial de Programas.



Figura 36: DCD CU Gestionar Historial de Programación.

4.4.6 Realización del CU Ofertar Programación.



Figura 37: DCD CU Ofertar Programación.

Conclusiones Parciales.

En el presente capítulo se ha tratado el tema correspondiente al diseño del sistema, se ha establecido un patrón de arquitectura como modelo a seguir a la hora de la implementación del mismo, así como se ha propuesto el uso de una serie de patrones de diseño que ayudan entre otras cosas a un correcto desempeño de la futura aplicación, así como a la reutilización del código que se produzca, de manera que la calidad del sistema sea la adecuada para este tipo de tecnologías. Se

presentaron además los diagramas correspondientes a las clases del diseño, que ofrecen una generalidad de la estructura interna de la futura aplicación, cada una de las clases que intervienen en la realización de cada uno de los casos de uso, así como de la relación existente entre ellas.

Conclusiones.

En el presente Trabajo de Diploma se presenta el análisis de una solución BSS/CRM para un sistema IPTV, la cual constituye una propuesta ingenieril que arroja las siguientes conclusiones:

- El estudio de sistemas IPTV que poseen gran prestigio a nivel mundial, arrojó que no existe una solución BSS/CRM que esté desarrollada bajo el paradigma de software libre, que integre el amplio conjunto de funcionalidades de alto procesamiento que son necesarias para que un sistema IPTV funcione correctamente, que cuente con cada una de las personalizaciones que se requieren como servicios en estos sistemas, y que además sea construido sobre un entorno Web.
- La herramienta, metodología y lenguaje de modelado identificados a partir de la investigación de las tendencias y tecnologías actuales para el desarrollo de software garantizará una modelación estable y organizada de los procesos que formarán parte de la solución.
- La modelación del sistema propuesto así como el diseño correspondiente al mismo contribuirá a una correcta implementación de la solución BSS/CRM.
- La documentación ingenieril correspondiente a la solución propuesta contribuirá a la posterior personalización de cada uno de los servicios que sean requeridos por los clientes.
- El diseño propuesto proporciona una estructura altamente escalable, de manera tal que se le pueden seguir añadiendo, quitando o modificando funcionalidades, en pos de realizar cambios para un negocio específico que lo requiera.

Recomendaciones.

Tras realizar un análisis exhaustivo de un significativo número de elementos que contribuyeron a la elaboración de la documentación ingenieril correspondiente a la solución BSS/CRM, se hace necesario que se listen un grupo de recomendaciones que permitan la mejora continua de la propuesta en cuestión:

- Analizar y aplicar consecuentemente el resultado de esta investigación en pos de lograr una correcta implementación de la solución.
- Realizar un seguimiento de la aplicación de la modelación realizada e identificar mejoras potenciales en la concepción del diseño propuesto.
- Continuar añadiendo funcionalidades a la solución en función de nuevos requerimientos que puedan ser identificados.
- Concebir la documentación ingenieril correspondiente a la fase de construcción y transición de la solución.
- Realizar la validación de los requerimientos del sistema.
- Realizar los diagramas de interacción correspondientes al diseño en pos de ilustrar mejor el comportamiento de las clases definidas así como sus relaciones.

Referencias Bibliográficas.

1. **Ramírez, David.** *IPTV Security*. Chichester : John Wiley & Sons, 2008.
2. **Telecommunications., Outstanding Performance for.** [En línea] [Citado el: 20 de Noviembre de 2009.] http://www.amd.com/us/Documents/43283-A_AMD64_and_Comarch_for_Telecom_White_Paper_PDF.pdf.
3. **Masivos, Medios de Comunicación.** [En línea] [Citado el: 20 de Noviembre de 2009.] www.emagister.com/uploads.../Comunidad_Emagister_64646_64646.pdf.
4. **Fortan, Fernando.** [En línea] [Citado el: 22 de Noviembre de 2009.] http://www.citel.oas.org/newsletter/2006/septiembre/iptv_i.asp.
5. **Held, Gilbert.** *Understanding IPTV*. New York : Taylor & Francis Group, 2007.
6. **IP, Convergencia hacia televisión sobre.** [En línea] [Citado el: 20 de Noviembre de 2009.] <http://biiec.epn.edu.ec:8180/dspace/bitstream/123456789/526/3/T10457CAP2.pdf>.
7. **QoS.** [En línea] [Citado el: 21 de noviembre de 2009.] <https://www.u-cursos.cl/ingenieria/2005/1/EL55A/1/.../69375>.
8. **Redes, Libro Blanco de Buenas Practicas Para el Despliegue de.** [En línea] [Citado el: 21 de Noviembre de 2009.] www.sandetel.es/Libro_Blanco_Redес_Inalambricas_Municipales.pdf.
9. **Multicast., Introduction to.** [En línea] [Citado el: 21 de Noviembre de 2009.] <http://www.firewall.cx/multicast-intro.php>.
10. **Howard Greenfield, Wes Simpson &.** *IPTV and Internet Video*. Burlington, USA : Elsevier Inc., 2007.
11. **A guide to IPTV: the technologies, the challenges and how to test IPTV.** [En línea] [Citado el: 21 de Noviembre de 2009.] <http://www.tektronix-resources.com/0707MTM400/IPTVPrimer.pdf>.
12. **Furlock, Miguel.** [En línea] [Citado el: 21 de Noviembre de 2009.] <http://almas.tv/edicion3/todoenlacabeza/zapping.html>.
13. **Rives., Jose Francisco.** [En línea] [Citado el: 22 de Noviembre de 2009.] <http://www.um.es/atika/gat/gat2/soluciones/que-son-los-codecs/>.
14. **IP, El protocolo.** [En línea] [Citado el: 22 de Noviembre de 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ip.html>.
15. **applications., RTP: a transport protocol for real-time.** [En línea] Julio de 2003. [Citado el: 22 de Noviembre de 2009.] <ftp://ftp.isi.edu/in-notes/rfc3550.pdf>.
16. **Telecomunicaciones, UIT. Union Internacional de.** *Curso a distancia de la UIT*. 2004-2005.
17. **Sun Microsystems, Inc.** *CHANGING TRENDS IN OPERATION AND BUSINESS SUPPORT*. 2007.
18. **www.gerenciaynegocios.com.** [En línea] [Citado el: 11 de Diciembre de 2009.] <http://www.gerenciaynegocios.com/teorias/crm>.
19. **Huawei.** *Huawei IPTV solution*.
20. **ZTE.** www.zte.com.cn. [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.zte.com.cn/en/>.
21. **DIGITAL, SUB-COMISION TECNICA DE LA TV.** *Informe Final v2*. La Habana : s.n., 2009.
22. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educacion, 2000. 84-7829-036-2.
23. **Valencia, Universidad Politécnica de.** [En línea] [Citado el: 15 de Diciembre de 2009.] www.dsic.upv.es/asignaturas/.../lsi/.../IntroduccionProcesoSW.doc.
24. **José H. Canós, Patricio Letelier y^M Carmen Penadés.** [En línea] [Citado el: 12 de Enero de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
25. **MSF.** [En línea] [Citado el: 13 de Enero de 2010.] <http://www.willydev.net/descargas/articulos/general/MSF.aspx>.
26. **Sanchez., Ing. Maria A. Mendoza.** [En línea] [Citado el: 12 de Enero de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

27. **Scrum, Met.** [En línea] [Citado el: 14 de Enero de 2010.] <http://www.chuidiang.com/ood/metodologia/scrum.php>.
28. **Scrum, Como funciona.** [En línea] [Citado el: 14 de Enero de 2010.] <http://www.proyectosagiles.org/como-funciona-scrum>.
29. **Amaro Calderon, Sarah Damaris, Valverde Rebaza, Jorge Carlos.** [En línea] [Citado el: 18 de 1 de 2010.] <http://www.seccperu.org/files/Metodologias%20Agiles.pdf>.
30. **CASE, Herramientas.** [En línea] [Citado el: 18 de 1 de 2010.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capitulo2.pdf.
31. **www.sparxsystems.com.ar.** Enterprise Architect. [En línea] [Citado el: 27 de Enero de 2010.] <http://www.sparxsystems.com.ar/products/ea.html>.
32. **Rational.** [En línea] [Citado el: 27 de Enero de 2010.] <http://www.insight.com.uy/consultoria-rational-ques-rational.html>.
33. **Paradigm, Visual.** [En línea] [Citado el: 27 de Enero de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
34. **Pressman, R.** *Ingeniería de Software. Un enfoque práctico.* La Habana : Félix Varela, 2005.
35. **Loja, Universidad Católica de.** [En línea] [Citado el: 5 de Marzo de 2010.] <http://www.utpl.edu.ec/eva/descargas/material/175/G18401.8.pdf>.