

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Plataforma web para la televisión digital
interactiva**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Asiel Hernández Valdes

Tutor: Ing.Yoandrys S. Pacheco Jeréz

Co-tutor: Ing.Dayami Chavez Ayala

La Habana, mayo de 2013
“Año 55 de la Revolución”

Cualquier tonto puede hacer cosas muy grandes, muy complejas, y muy violentas. Se necesita un toque de genialidad y de mucho valor para ir en la dirección opuesta.

Albert Einstein.

- *A mis padres por tanta ayuda, apoyo, y sacrificios en todos estos años de estudios.*
- *A mi hermano por estar ahí en los momentos difíciles.*
- *A Joandry's Pacheco y Dayami Chávez mis tutores por su imprescindible ayuda en esta tesis.*
- *A mis amigos de la universidad, por su siempre fiel compañía y ayudarme en estos maravillosos 5 años de estudios, en especial a Karel R. Arman, Yadir Hernández, Eduardo Medina, Aida Socarras, Yanet Diaz y, Lisandra Matos.*

A mi mamá, mi papá y mi hermano.

Declaración de autoría

Declaro por este medio que yo, Asiel Hernández Valdes con carné de identidad 89040315149, soy el autor de este trabajo y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firma la presente declaración jurada de autoría en La Habana a los 19 días del mes 6 del año 2013.

Asiel Hernández Valdes

Yoandrys S. Pacheco Jérez

Nombre
Autor

Nombre
Tutor

Resumen

La televisión y el Internet son dos tecnologías que han revolucionado la cultura mundial en los últimos años. A principios de los 90 estas convergen dando lugar a la televisión digital sobre las redes de Internet, marcando esto un prometedor futuro para la televisión interactiva. La televisión interactiva es un nuevo paradigma en el cual el usuario final es protagonista de todos los contenidos audiovisuales transmitidos por algún proveedor. Revisar las carteleras de los canales en tiempo real, consumir videos bajo demanda, hacer compras online y grabar programas televisivo, son algunas de las funcionalidades más comunes de la televisión interactiva. El presente trabajo es el desarrollo de una plataforma interactiva para la televisión digital en el proyecto Sistema de Transmisión de Canales Virtuales de la Universidad de las Ciencias Informáticas. Se presentan los fundamentos teóricos de la investigación, las herramientas y tecnologías utilizadas, así como los artefactos generados durante el ciclo de desarrollo de la plataforma interactiva.

Palabras clave: IPTV, Set Top Box, Televisión Interactiva.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN:..... 1

CAPÍTULO 1: Fundamentos Teóricos de la Investigación.....6

 1.1 Introducción..... 6

 1.2 Televisión digital..... 6

 Tipos de televisión digital (DTV). 6

 Ventajas de la TV digital sobre la TV analógica. 7

 Diferencias entre IPTV y DTV. 8

 1.3 Conceptos asociados al dominio del problema..... 8

 Definición de televisión digital (DTV, Digital Televisión). 8

 Televisión Interactividad. 8

 Definición de IPTV. 9

 Streaming. 9

 1.4 Elementos de un sistema IPTV. 9

 1.4.1 Head End..... 10

 1.4.2 Servidores de Video..... 10

 1.4.3 Set Top Box (STB)..... 11

 1.4.4 Middleware. 12

 1.4.5 Sistema de Control de Derechos y de Acceso Condicional (DRM/CAS, Digital Rights Management/Conditional Access System)..... 12

 1.4.6 Home Gateways. 13

 1.5 Televisión Interactiva..... 13

 Dirección de comunicación. 13

 Niveles de Interactividad..... 14

 1.6 Principales funcionalidades de un sistema interactivo según niveles de interactividad. 15

 1.6.1 Nivel 1 de interactividad. 15

 1.6.2 Nivel 2 de interactividad. 17

 1.6.3 Nivel 3 de interactividad. 18

 1.7 Análisis de soluciones existentes. 18

 1.7.1 MediaHighway. 19

 1.7.2 OpenTV. 19

 1.7.3 Betanova. 20

1.7.4 MHEG (Multimedia e Hipermedia Expert Group).....	20
1.7.5 MHP (Multimedia Home Platforms).....	21
1.7.6 GEM.	22
1.8 Estado actual en Cuba.	23
1.9 Conclusiones Parciales.	24
CAPÍTULO 2: Herramientas para la solución.....	25
2.1 Introducción.....	25
2.2 Metodología de Desarrollo de Software.....	25
RUP.....	26
Lenguaje Unificado de Modelado (UML v2.0)	27
2.3 Framework de desarrollo.....	28
Symfony2	28
jQuery.....	29
2.4 Lenguaje de desarrollo.....	29
PHP v5.3	29
JavaScript:.....	30
CSS.....	31
HTML.....	31
2.5 Servidor Web	31
Apache 2.2	32
2.6 Entorno de Desarrollo Integrado (IDE)	32
Netbeans v7.0.1	32
2.7 Herramienta CASE.....	33
Visual Paradigm for UML v8.0 Enterprise Edition v5.0	33
2.8 Sistema gestor de base de datos	34
PostgreSQL.....	35
2.9 Conclusiones parciales.....	35
CAPÍTULO 3: Descripción de la solución propuesta.	36
3.1 Introducción.....	36
3.2 Modelo de Dominio	37
3.2.1 Glosario de Términos.....	37
3.3 Requisitos	38
3.3.1 Requisitos Funcionales.....	38
3.3.2 Requisitos No Funcionales	39

3.4 Modelo del Sistema	40
3.4.1 Diagrama de Casos de Uso del Sistema (CUS)	40
3.4.2 Descripción de los actores del sistema	42
3.4.3 Descripción de los casos de usos del sistema.	42
3.5 Patrones.....	45
3.5.1 Patrones de arquitectura.....	45
3.5.2 Patrones de diseño	47
3.6 Diagramas de clases del diseño.....	48
3.7 Diagrama de despliegue.....	49
3.8 Conclusiones parciales.....	50
CAPÍTULO 4: Implementación y prueba	51
4.1 Introducción.....	51
4.2 Modelo de Implementación	51
4.3 Estándares de codificación.....	52
4.4 Pruebas.....	53
4.4.1 Tipos de pruebas	54
4.4.2 Pruebas de caja negra	55
4.5 Conclusiones Parciales	57
CONCLUSIONES GENERALES	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS.....	60
BIBLIOGRÁFIAS CONSULTADAS	63
ANEXOS.....	63

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Dirección de Comunicación (7) 14
Ilustración 2 EPG de canal español BTV (6). 16
Ilustración 3 Diagrama del esfuerzo de actividades según la etapa del proyecto (20). 26
Ilustración 4 Modelo de Dominio 37
Ilustración 5 Diagrama de casos de uso del sistema..... 41
Ilustración 9 Diagrama de clases del diseño 49
Ilustración 10 Diagrama de despliegue. 49
Ilustración 11 Diagrama de componentes 51
Ilustración 12 Ejemplo de notación camello 52
Ilustración 13 Ejemplo de notación pascal 53
Ilustración 14 Interfaz Mostrar información de canales 64
Ilustración 15 Interfaz Mostrar cartelera de un canal..... 64

ÍNDICE DE TABLAS

Tabla 1 Actores del sistema..... 42
Tabla 2 Descripción del caso de uso Mostrar información de canales en vivo 43
Tabla 3 Descripción del caso de uso Mostrar Cartelera 44

INTRODUCCIÓN:

El siglo XX pasó a la historia como el siglo de la comunicación de masas. La televisión (TV), uno de los inventos más revolucionarios logró conquistar a millones de espectadores y transmitir mucho de los grandes hitos de la historia de ese siglo. Un proceso, por otro lado, basado en un principio unidireccional¹ del mensaje, en la que el emisor formula mensajes que le llegan a un receptor quien recibe pero no responde, influyendo sólo como número en una audiencia y no como parte activa del proceso de comunicación. La creación de contenidos siempre se ha enfocado a las grandes audiencias y han pasado por alto los principios particulares y las diferentes idiosincrasias del público. La televisión por cable y satelital vienen a solventar en parte este problema ya que el espectador puede escoger entre una serie de ofertas, siempre finitas y caracterizadoras de un canal, y crear una programación fiel a sus intereses particulares, proceso en el cual el usuario final sigue siendo un simple receptor del mensaje.

En 1969 ARPANET² fue creada, y con ella comenzó una nueva fase en las comunicaciones. Más adelante, en 1983, el protocolo núcleo de ARPANET migró de NCP (*Network Control Protocol*), a TCP/IP (*Transfer Control Protocol/Internet Protocol*) y de esta forma nació Internet. Aparece Internet para introducir una variable fundamental en el control sobre el mensaje: la interactividad. Es decir, ya no solo se reciben los mensajes enviados por algún proveedor, sino que se cuenta con un canal de retorno para poder tener una retroalimentación con esos contenidos, dejando de ser un simple espectador. La televisión utilizando el protocolo IP de Internet comúnmente llamada IPTV (Ramírez, 2008) estimula al individuo a participar del contenido, usándolo, modificándolo e incluso, creándolo con el único fin de llegar a los demás. Una de las claves fundamentales de la interactividad es que permite que se puedan intercambiar los roles de transmisor y receptor, además, se incita a controlar el contenido.

Un buen ejemplo lo constituye la web 2.0 donde los usuarios finales son verdaderos protagonistas de toda la gestión de contenidos. Los contenidos son construidos a partir de

¹ En una sola dirección.

² Advanced Research Projects Agency Network. Agencia creada por el Dpto. de Defensa de los EE.UU como medio de comunicación para los distintos organismos de ese país.

colaboraciones entre usuarios y no solo y exclusivamente gracias a las grandes corporaciones. Los usuarios son entes activos que crean espacios virtuales para el intercambio de información, manejan perfectamente la búsqueda de datos y sienten la web como algo que le pertenece y no como un factor externo del que recibe mensajes como ocurre con los medios de comunicación tradicionales.

Actualmente los proveedores de servicios de televisión digital no solo se limitan a la transmisión de canales, sino que invierten grandes cantidades de dinero en lograr que sus usuarios disfruten de la interactividad. Cuba, actualmente cuenta con una infraestructura televisiva basada totalmente en tecnologías analógicas, y comenzó a dar sus primeros pasos hacia las tecnologías digitales. Con la colaboración de entidades -ya sean nacionales o extranjeras- el Instituto Cubano de Radio y Televisión (ICRT) ha posibilitado la incorporación de sistemas, tecnologías, profesionales y consultores que ayudan al desarrollo, implementación y capacitación en los temas de la televisión digital en la isla.

Una de estas entidades es la Universidad de las Ciencias informáticas (UCI), que cuenta con un centro de desarrollo que trabaja las líneas de procesamiento de imagen y video, incluyendo los elementos relacionados con la televisión. Dicho centro cuenta con un departamento que a su vez tiene varios proyectos de desarrollo trabajando en sistemas para automatizar los procesos que se realizan en una televisora. Un ejemplo de ello lo constituye los productos que se implementan en el proyecto Sistema de Transmisión de Canales Virtuales (STCV); dichos productos brindan soluciones para la monitorización de señales digitales, transmisión de video a través de la red de datos haciendo uso de la tecnología *streaming*, la planificación y programación de carteleros y recursos que serán necesarios para la creación, transmisión y visualización de las señales televisivas a través de la red de datos.

Todos los productos del citado proyecto están encaminados a transmitir canales de televisión de forma tradicional, es decir donde los receptores sólo pueden visualizar los contenidos. Esto trae consigo que los usuarios finales no puedan disfrutar a plenitud todas las bondades de la televisión digital, restándole también posibles clientes al producto en general ya que no poseen valor agregado. Uno de los inconvenientes que presenta el proyecto a la hora de desplegarlo consiste en la descentralización de los servicios que ofrecen, tales como la información de los canales, sus posibles carteleros y las medias disponibles para consumir video bajo demanda. La televisión digital y la interactividad a través del televisor es un elemento importante debido a que los usuarios suelen

identificarse más con este medio de comunicación que a la hora de enfrentarse ante una computadora e Internet. Gestionar contenidos actualmente a través del televisor resulta un proceso muy engorroso para los usuarios finales, ya que normalmente cuentan simplemente con un control remoto.

Partiendo de la situación anteriormente expuesta se ha definido como **problema a resolver** ¿Cómo lograr la interactividad para los usuarios de televisión digital? Con el fin de darle solución a dicha problemática, se hace necesario desarrollar una plataforma que les permita a los usuarios de televisión digital lograr la interactividad con los contenidos audiovisuales que se transmiten y centralizar el acceso a los diferentes servicios que se brindan por parte del proyecto STCV, 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**.

El **campo de acción** de la presente investigación lo constituye las plataformas interactivas para sistemas IPTV.

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

1. Caracterización de la televisión digital interactiva.
2. Revisión bibliográfica de las soluciones existentes en Cuba y el mundo.
3. Selección de las herramientas y tecnologías a utilizar en la construcción de la propuesta de solución.
4. Identificación de los requisitos funcionales y no funcionales.
5. Identificar los casos de usos del sistema.
6. Diseño arquitectónico de la aplicación.
7. Implementación y validación de la propuesta de solución.

El desarrollo satisfactorio de las tareas expuestas anteriormente contribuirá al cumplimiento de la **idea a defender** de esta investigación, la cual plantea que la implementación de una plataforma interactiva permitirá al proyecto STCV proveer contenidos interactivos para los usuarios de televisión digital.

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

Métodos Teóricos

“Modelación”: En el desarrollo de la investigación es necesaria la modelación de la arquitectura que se propone, es por eso que se utiliza este método para crear modelos con vistas que favorezcan la comprensión de la misma.

“Histórico – Lógico”: Se aplica para investigar la evolución y desarrollo de la televisión en el mundo, además de la diferentes etapas del surgimiento de los sistemas interactivos.

“Analítico - Sintético”: Se utilizó 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 esta bibliografía.

Métodos Empíricos

“Observación”: Este método facilita conocer el panorama real de una situación mediante la percepción directa. Con este método se validan los objetivos propuestos para el buen desempeño de la plataforma interactiva, revisando como se comporta el sistema ante las diferentes entradas de datos.

El documento está constituido por cuatro capítulos.

Capítulo 1. Fundamentos Teóricos de la Investigación.

Contempla la fundamentación teórica de la 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. Además se realiza un estudio de cómo se encuentra el estado del arte de las plataformas interactivas en Cuba y el mundo.

Capítulo 2. Herramientas para la solución.

Se dispone de un análisis sobre las principales herramientas, metodologías de desarrollo, lenguajes de programación, ambientes de desarrollo escogidos para darle una solución apropiada a la problemática planteada.

Capítulo 3. Descripción de la solución propuesta.

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.

Capítulo 4. Implementación y Prueba

Se aborda lo relacionado con la implementación del sistema, se seleccionan los estándares de codificación, se define el diagrama de componentes y el diagrama de despliegue. Se realizan las pruebas de caja negra para validar el correcto funcionamiento del sistema.

CAPÍTULO 1: Fundamentos Teóricos de la Investigación.

Introducción

Este primer capítulo está dedicado a explicar los principales aspectos relacionados con la tecnología IPTV, haciendo referencia a sus principales servicios y elementos. Se describen las principales características de los sistemas interactivos así como los principales servicios que brinda esta tecnología digital. Además abordará como se encuentra el estado del arte de las plataformas interactivas en Cuba y el mundo.

1.1 Televisión digital (DTV por sus siglas en inglés).

La televisión digital se ha estado desarrollando intensamente en los últimos 10 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 PAL³ y NTSC⁴. (Fortan, 2012)

El cambio de TV analógica a digital es más brusco que la introducción de la TV a color en su momento, ya 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.

Tipos de televisión digital.

Dentro de la DTV, se encuentran la TV por Cable (TDC), la TV por Satélite (TDS), y la IPTV. La TDC como su nombre lo indica, consiste en la transmisión de señales digitales a

³ -Es la sigla de Phase Alternating Line (en español línea de fase alternada). Es el nombre con el que se designa al sistema de codificación utilizado en la transmisión de señales de televisión analógica en color en la mayor parte del mundo

⁴ - National Television System Committee, en español Comisión Nacional de Sistema de Televisión es un sistema de codificación y transmisión de Televisión en color analógico desarrollado en Estados Unidos en torno a 1940, y que se emplea en la actualidad en la mayor parte de América y Japón, entre otros países.

través de sistemas de TV por cable, ya sea coaxial, fibra óptica o telefónico. Por su parte TDS se trata de transmitir señales satelitales en formato digital.

Ventajas de la TV digital sobre la TV analógica.

La conversión de la señal analógica a digital permite que la información sea fácilmente manipulada por los sistemas de transmisión y recepción. Esto permite el desarrollo de técnicas de compresión que minimizan tanto el ancho de banda de la transmisión como el almacenamiento de la información. (Held, 2007)

La resolución de pantalla ⁵ 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 los de 480, pero además están los de 720 y 1080 líneas, siendo éstos dos últimos los que se usan para TV de alta definición (HDTV⁶).

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 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 píxel⁷. (Held, 2007)

Resumiendo, se pueden citar las siguientes ventajas que la TV digital permite:

- ❖ Incrementar notablemente el número de programas y servicios al realizar un mejor uso del ancho de banda del medio de transmisión.
- ❖ Mejorar la calidad de las imágenes y el sonido de las transmisiones y recepciones televisivas.
- ❖ Establecer servicios personalizados e interactivos de radiodifusión y telecomunicaciones.
- ❖ Facilitar la convergencia entre el sector audiovisual, las telecomunicaciones y la informática.

⁵ Resolución de pantalla: Se refiere al número de pixels que puede ser mostrada en la pantalla por pulgada.

⁶ High Definition Television.

⁷ Es la menor unidad homogénea en color que forma parte de una imagen digital.

- ❖ Difusión de múltiples canales con distintas modalidades de cifrado.
- ❖ Aplicar técnicas de compresión de video y audio.

Diferencias entre IPTV y otros tipos televisión digital.

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 la red de datos y la televisión terrestre y satelital usan el aire. En el caso de la versión CATV (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.

1.2 Conceptos asociados al dominio del problema.

Definición de televisión digital.

La TV digital, es la transmisión y recepción de video y sonido a través de un canal determinado, codificando la información en forma binaria. Al contrario de la televisión tradicional que lo hace de forma analógica.

Televisión Interactividad.

La interactividad es la forma en la que los mensajes de una secuencia se relacionan entre ellos, y especialmente la forma en la que mensajes posteriores influyen en mensajes anteriores. (García, 2010).

La televisión interactiva se podría definir de múltiples modos. Por ejemplo, podría entenderse como la radiodifusión de un medio de transmisión de información de imagen y sonido que es capaz de recibir información de cada usuario y tenerla en cuenta para modificar su propio contenido en tiempo real, es decir, mientras se realiza la emisión. Evitando este tipo de expresiones formales, que muchas veces aún confunden más a lectores poco expertos, se podría definir también como una televisión en la que el

telespectador puede hacer algo más que simplemente verla o/y oírla, pasando a ser una especie de “teleparticipante”, ya que tiene la posibilidad de realizar acciones que pueden cambiar el contenido mostrado por su televisor. (García, 2010)

Definición de IPTV.

Definición dada por la Unión Internacional de Telecomunicaciones:

Servicios multimedia de datos tales como televisión/video/audio/texto/gráficos/ entregados sobre redes basadas en el protocolo IP, gestionados para proporcionar el nivel requerido de calidad de, servicio y experiencia en seguridad, interactividad y fiabilidad. (Held, 2007)

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 televisión digital, sea tanto en su versión terrestre, CATV (Cable TV), o satelital. (Greenfield Howard, 2007)

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. (Tektronix, 2010)

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).

1.3 Elementos de un sistema IPTV.

Para comprender con mayor facilidad el funcionamiento de un sistema interactivo bajo la red de datos es importante conocer los diferentes elementos que conforman un sistema IPTV. En un sistema IPTV común se puede identificar los siguientes bloques principales:

- ❖ *Head End* ó cabecera.
- ❖ Sistema de Video bajo Demanda.
- ❖ *Middleware* ó Mediador.
- ❖ Sistema de control de derechos (*Digital Rights Management*).
- ❖ Set-top box.

- ❖ *Home Gateways* o puertas de enlace.

1.3.1 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.
- ❖ Conformar los *streams*⁸ digitales.
- ❖ Insertar propaganda y sobreimpresos.
- ❖ Multiplexar⁹ las señales digitales en un único flujo que será distribuido en la red de acceso y transporte IP del operador. Cada canal de broadcast¹⁰ TV conformará un grupo IGMP¹¹ propio al cual se unirán los dispositivos clientes que los deban recibir.
- ❖ Implementar la presentación al usuario final de un mosaico de canales.
- ❖ Posibilitar monitorear las señales en distintos sitios. (Tektronix, 2010)

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 operador¹².

1.3.2 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 video bajo demanda, grabado de canales,

⁸ Flujos de las señales digitales.

⁹ 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.

¹⁰ Trasmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultanea.

¹¹ Protocolo de red para la trasmisión de sonido y video en tiempo real.

¹² Operador se refiere en estos casos, a la compañía o institución que ofrece el servicio de IPTV.

no así para los servicios en los que se transmiten señales de *broadcast* como canales de TV, programación en vivo, 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.

1.3.3 Set Top Box (STB).

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 ADSL¹³. En algunos casos los propios STB cuentan con el módem ADSL integrado. A su vez esos dispositivos presentan una interfaz RCA¹⁴ 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, 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 IGMP¹⁵, ya que cuando se desea cambiar de canal se envía un mensaje IGMP denominado “*LeaveGroup*”, en español dejar canal, hacia su servidor IGMP más próximo, y luego le enviará un mensaje llamado “*JoinGroup*” o en español unirse a un canal, con la dirección de *multicast* del nuevo canal deseado.

Cuenta con un pequeño procesador en el que corren aplicaciones tales como un browser de configuración del servicio, suscripción a canales, orden de películas en servicio de video bajo demanda, visualización del catálogo de películas y control de acceso. (Telecomunicaciones, 2005)

¹³ Codificador –decodificador de señales digital-analógicas utilizado para conectarse a internet y que tiene la propiedad de poder usar el teléfono y la computadora al mismo tiempo para conectarse por el mismo canal.

¹⁴ Conector utilizado para los televisores analógicos conformados principalmente para dos señales de audio y uno de video.

¹⁵ Protocolo utilizado para la transmisión de canales por la red de datos.

1.3.4 Middleware.

El Middleware es una plataforma de gestión de aplicaciones que interactúa con la red de acceso, el *Head End* y los dispositivos utilizados por los usuarios finales para permitir el aprovisionamiento y la distribución de servicios de televisión interactivos.

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 video bajo demanda, el usuario que desea ordenar una película para ser vista en determinado día y hora, realiza la solicitud al *middleware*, estableciendo una sesión con éste. El middleware lo autorizará o no en función de la cuenta del usuario y 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. (Telecomunicaciones, 2005)

1.3.5 Sistema de Control de Derechos.

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 video bajo demanda, audio bajo demanda, *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. (Telecomunicaciones, 2005)

1.3.6 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.

Los “*home gateways*” deberán disponer de al menos dos puertas *Ethernet* para mapear diferentes calidades de servicio. Una puerta es para conectar la red residencial de Internet (basada en servicios de mejor esfuerzo) y la otra para conectar a los STB de IPTV, que requieren de calidad de servicio (*QoS*). Si el operador implementa cada servicio en una LAN virtual diferente (*VLAN*), cada *VLAN* se mapea en una puerta *Ethernet* distinta. (Telecomunicaciones, 2005)

1.4 Televisión Interactiva.

Alguna de las características más importantes de la televisión interactiva son:

Dirección de comunicación.

La dirección de la comunicación es otro factor clave en todo espacio de interacción. (McMillan 2008) propone cuatro posibles direcciones de comunicación en entornos interactivos atendiendo a los principales tipos posibles de relación entre emisores y receptores.

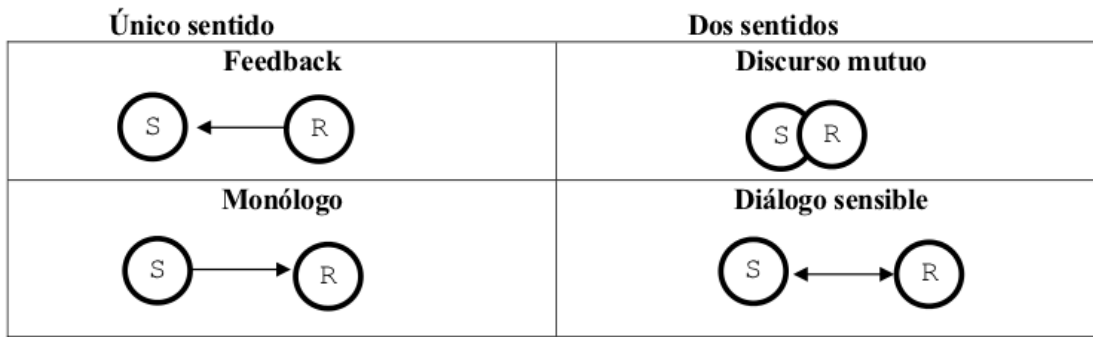


Ilustración 1 Dirección de Comunicación (Jiménez, 2009)

En la ilustración 1 las flechas indican la dirección de comunicación y se puede observar que el nivel máximo de interactividad se produce cuando existe un **diálogo sensible** entre emisor y receptor, aunque es el emisor el que retiene el control primario sobre la comunicación, hecho que se produce en situaciones como el e-comercio en el que el emisor ofrece posibilidades que el receptor selecciona. En el **monólogo** la dirección de comunicación indica el grado de interacción que se pueden encontrar en la mayoría de páginas web, donde un emisor manda mensajes cerrados. En el modo de **feedback** se puede encontrar el sentido de dirección que se produce cuando es el receptor el que domina la situación, por ejemplo, cuando se envía un correo electrónico y no se sabe si va a contestar o no. En lo que se refiere al **discurso mutuo**, esta dirección de comunicación se produce cuando es fácil intercambiar las posiciones de emisor y receptor, por ejemplo, en los chats. (Jiménez, 2009)

Niveles de Interactividad.

Desde un punto de vista de servicio y aplicación, actualmente se han definido tres perfiles, permitiendo a los fabricantes desarrollar una serie de productos que proporcionen diferentes funcionalidades. Cada perfil puede coexistir con los demás y se corresponde con un nivel de interactividad:

- *Enhanced broadcast profile*: Nivel 1 de interactividad. Interactividad local. Este tipo de aplicaciones, como el teletexto digital y PVR (*Personal Video Recording*), obtienen los datos requeridos del *stream* emitido. Únicamente soportan interactividad local donde el cliente puede interactuar sólo con los datos almacenados en el STB.

- *Interactive broadcast profile*: Nivel 2 de interactividad. Interactividad sin respuesta por el canal de retorno. Aplicaciones como el voto “*on-line*” permiten al usuario proporcionar información de respuesta en una sola dirección, sin una contestación directa sobre el mismo enlace, ya que la información vuelve por el canal de radiodifusión.

- *Internet acces profile*: Nivel 3 de interactividad. Acceso total a Internet. Aplicaciones como el *T-Mail*, navegación Web o juegos on-line, permiten al usuario adquirir información de fuentes externas al *stream* emitido por el canal de retorno. Suele necesitar conexión de banda ancha. (Jiménez, 2009)

1.5 Principales funcionalidades de un sistema interactivo según niveles de interactividad.

En este apartado se describe las principales aplicaciones que soportan los sistemas interactivos según los diferentes niveles de interactividad. Normalmente los más utilizados son los del nivel de interactividad uno ya que son los que principalmente prioriza el proveedor de servicios televisivos.

1.5.1 Nivel 1 de interactividad.

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. 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 metadata¹⁶ de los programas actuales, pasados o futuros: Duración, Género, Origen; Director, Actores entre otras cosas. (Jiménez, 2009)

¹⁶ Son datos que describen a otros datos.



Ilustración 2 EPG de canal español BTV.

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 (comenzar), Pause (pausar), Stop (parar), REW (reversa), FF (avance rápido).

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. (Jiménez, 2009)

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.

Capítulo 1: Fundamentos Teóricos de la Investigación

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.

En el caso que se almacene en discos rígidos de la red el exceso de tráfico puede llegar a ser muy grande si no se controla esta funcionalidad. En caso de que se esté transmitiendo un evento deportivo, es razonable que se requiera por muchos usuarios al mismo la repetición de una misma escena provocando esto un pico de tráfico en la red (a la cual posiblemente no esté diseñada). Por otro lado, si existen muchos usuarios que estén viendo un programa desplazado en el tiempo también penalizaría la red salvo que se disponga de discos rígidos en los STB. (Jiménez, 2009)

1.5.2 Nivel 2 de interactividad.

Pagar por Ver (PPV, Pay Per View).

Mediante el PPV se brinda a los 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 de TV en vivo y exige una contratación del contenido como en los casos de video bajo demanda. (Jiménez, 2009)

T-Encuestas.

El servicio de tele-encuesta, con el cual, eligiendo una de las opciones disponibles se puede, entre otros, dar las siguientes posibilidades:

-Elegir por mayoría qué programa será retransmitido o incluso que quiere que suceda en una serie de televisión.

-Responder simplemente a una encuesta hecha por el proveedor de servicios.

-Votar para alguna clase de evento local o privado.

1.5.3 Nivel 3 de interactividad.

T-Comunicación.

En el ámbito de la comunicación encontramos tanto los clientes de correo por TV como los espacios de mensajería instantánea. También son interesantes las aplicaciones de mensajería instantánea similar a las utilizadas para PC, que permitiera, mediante el uso de una cámara, poder hacer video-llamadas.

T-Juegos.

Aquí cabe destacar que los juegos pueden pertenecer a cualquiera de los niveles de interactividad, todo está en la forma en que este funcione. Para los juegos del nivel 3, deben ser juegos participativos en la que 2 o más usuarios intervengan.

T-Salud.

En el ámbito de la salud se ofrecen varias nuevas posibilidades. Una de ellas es la posibilidad de reservar consulta con su médico de cabecera, pudiendo introducir algunos de los síntomas para avanzar el trabajo del doctor. La teleconsulta es otra posibilidad que consiste en estar en contacto con su doctor sin necesidad de desplazarse al centro de salud. Este sistema de teleconsulta permite la realización de estadísticas que permitirían una mejor actuación de cara, por ejemplo, a la detección de un brote de gripe.

1.6 Análisis de soluciones existentes.

Algunos de los middleware más utilizados para la televisión digital son:

Privativos:

- MediaHighway
- OpenTV
- Betanova

Abiertos:

- MHP (Multimedia Home Platforms)
- MHEG (Multimedia e Hipermedia Expert Group)

1.6.1 MediaHighway.

MediaHighway fue desarrollado en 1993 por el departamento de I + D de Canal + (que posteriormente se convirtió en Canal + Technologies) inicialmente para el lanzamiento del primer servicio de televisión digital francés en abril de 1996: Canal Satélite Numérique (CSN). MediaHighway es utilizada por todas las variaciones nacionales de Canal Satélite Digital, que se iniciaron poco después (España, Italia, Polonia, etc.) Más recientemente se propuso a los prestadores de servicios que no pertenecen al grupo Canal +, como ONdigital (Reino Unido) en 1998, que luego se convirtió en ITV Digital, y algunos otros operadores de cable y satélite en Europa, Asia y los Estados Unidos. Canal + Technologies exige un duro proceso de certificación, que incluye un importante conjunto de pruebas unitarias antes de permitir la comercialización de productos con la etiqueta MediaHighway. (Keith, 2009)

1.6.2 OpenTV.

OpenTV es un middleware propuesto por una empresa (ahora independiente) que lleva el mismo nombre. En su origen fue una unión entre Thomson Multimedia y Sun Microsystems (Sun Interactive Alianza formada en 1994). El primer proveedor de servicios de televisión digital para uso OpenTV fue el satélite francés ramo TPSin 1996. OpenTV ahora es utilizado por más de 30 proveedores de servicios de televisión digital en todo el mundo y está instalado en más de 10 millones de set top box con 30 proveedores. OpenTV apoya aplicaciones de video en demanda, impulsa los de pago por visión y permite descargar datos y aplicaciones. OpenTV existen muchas versiones.

- OpenTV 1.0x todavía es la versión utilizada por los pioneros, sobre todo el cable y el satélite francés de servicios. Que no se ha propuesto a los nuevos clientes desde principios de 2000. No obstante, ha sido mejorado por muchas descargas en el tiempo.
- OpenTV 1.2 es un derivado de la británica BskyB, el proveedor de servicios por satélite.
- ES OpenTV es la segunda generación de OpenTV.
- OpenTV EN2 es la tercera generación, que hereda de ambos (Keith, 2009).

1.6.3 Betanova.

Desarrollado por Betaresearch, la rama de tecnología de los grandes medios de comunicación del grupo alemán Kirch, Betanova, en contraste con la mayoría de los demás middlewares, no pretende ser independiente de la plataforma de hardware o RTOS a la que se ejecuta. Está íntimamente relacionado con el "d-box", el set-top box fue desarrollado para el satélite de pago Premiere Workd. Una variante del cable de esta plataforma ha sido adoptada por DeutscheTelekom por sus redes de cable. (Keith, 2009)

1.6.4 MHEG (Multimedia e Hipermedia Expert Group).

El MHEG (Multimedia e Hipermedia Expert Group) es un lenguaje descriptivo de presentaciones multimedia, comparables a HTML (HyperText Markup Language) para las páginas de hipertexto. La versión MHEG-5, en virtud de la normalización de ISO / IEC 13522-5 de referencia, es un subconjunto de MHEG-1 especialmente dedicada a los receptores de televisión digital (set-top box o un receptor de TV digital integrado, IDTV), con limitado poder de procesamiento y tamaño de la memoria. Es prácticamente la única versión MHEG en volumen real de uso.

En el vocabulario MHEG, una presentación multimedia es una aplicación MHEG. Una aplicación MHEG es un conjunto de escenas MHEG. Una escena MHEG se compone de tres grandes grupos de objetos o ingredientes:

- Los objetos interactivos, como botones del cursor, o de otros objetos similares.
- La relación de objetos (enlaces) que definen los procesos desencadenados por las acciones del usuario sobre los objetos interactivos.
- La presentación de objetos (los objetos), controlada por el mencionado enlace objetos.

Cuando una aplicación MHEG se inicia, en general, espera una acción del usuario. Si el usuario pulsa un botón, la aplicación genera el evento "seleccione el botón". Un enlace puede ser definido en este (interactivo) de objetos, por ejemplo, con el fin de iniciar una secuencia de vídeo (presentación de objetos). Los objetos pueden a su vez generan eventos que pueden activar los enlaces. (Hervé, 2008)

1.6.5 MHP (Multimedia Home Platforms).

El sistema MHP define una plataforma común para las aplicaciones interactivas de la televisión digital, independiente tanto del proveedor de servicios interactivos como del receptor de televisión utilizado. El estándar MHP lo que ofrece básicamente es interoperabilidad entre diferentes aplicaciones y terminales y entre los propios terminales por medio de una interfaz genérico entre las aplicaciones digitales interactivas proporcionadas por la fundación Digital Video Broadcasting y los terminales en los cuales se van a ejecutar, que no tienen por qué ser receptores tradicionales, sino STB o incluso ordenadores personales. (Hervé, 2008)

El estándar MHP soporta distintos tipos de aplicaciones interactivas:

- Guía Electrónica de Programas (EPG).
- Servicios de información como noticias, deportes, supe teletexto.
- Aplicaciones sincronizadas con el contenido de los programas.
- E-mail e Internet.
- Otros servicios: comercio electrónico, servicios de educación y salud.

DVB-MHP utiliza el lenguaje de programación Java para sus aplicaciones y define la plataforma conocida como DVB-J, basada en la Máquina Virtual de Java (JVM) especificada por Sun Microsystems. DVB-J define un conjunto de APIs (Application Program Interface en inglés) genéricas, situadas entre las aplicaciones y el sistema de software, para proporcionar a las distintas aplicaciones acceso a los recursos disponibles en el receptor. (Hervé, 2008)

DVB Project desarrolló MHP (Multimedia Home Platform) como el primer estándar abierto para la televisión interactiva a nivel mundial. Aunque MHP se diseñó para ejecutarse en plataformas DVB, se mostró interés en extender su interoperabilidad a otras plataformas de televisión digital. Este interés dio lugar a GEM (Globally Executable MHP), un marco de trabajo que permite a otras organizaciones definir especificaciones basadas en MHP. Una de estas especificaciones es OCAP™ (Open Cable Application Platform), que ha sido adoptada por el sector del cable en Estados Unidos. En OCAP se eliminan las diversas tecnologías y especificaciones de DVB que no se utilizan en el sector del cable en Estados Unidos y se sustituyen por aplicaciones con funciones equivalentes, tal y como se especifica en GEM. (Hervé, 2008)

1.6.6 GEM.

MHP permite extender el estándar definido a otras redes de transmisión. Globally Executable Multimedia Home Platform (GEM) fue creado con ese propósito: permitir que otros cuerpos de estandarización u organizaciones pudieran definir unas especificaciones basadas en el estándar MHP. GEM, basada en MHP versión 1.0.2, elimina los elementos específicos orientados a DVB, permitiendo así su sustitución por otros más adecuados según la aplicación. GEM constituye actualmente la base de estándares como ACAP (ATSC), ARIB B23 (ARIB) u OCAP (U.S CableLabs).

DVB-GEM Aparece como consecuencia de una petición de CableLabs para unificar MHP con el estándar DASE definido por ASTC.

La especificación GEM consiste en un subconjunto de MHP que ha sido diseñado teniendo en cuenta las diferentes posibles implementaciones del mismo por diferentes estándares de middleware. Este subconjunto incluye lo siguiente:

- Aspectos técnicos de la interoperabilidad derivados de los estándares anteriores como OCAP o DASE.
- Aspectos relacionados con los mecanismos de transmisión: modulación, entrega y certificados.
- Aspectos de mercado de los operadores de red (Hervé, 2008)

GEM es un framework orientado a permitir a las distintas organizaciones a trabajar en armonía en cuanto a especificaciones técnicas, como por ejemplo la elección de un único runtime¹⁷ de ejecución y un único conjunto de APIs. El objetivo es que tanto aplicaciones como contenido funcionen en todas las .plataformas basadas en GEM.

La especificación GEM lista aquellas partes de la especificación MHP que se entiende son específicas de la tecnología o el mercado de DVB. GEM permite la sustitución de estas allí donde sea necesario siempre que la nueva tecnología sea funcionalmente equivalente a la original. (Hervé, 2008)

A través del estándar abierto para el desarrollo de servicios interactivos (MHP), DVB permite enviar a los decodificadores información adicional a la programación (relacionada con el contenido o independiente al mismo). Provee la base para el Global Executable MHP (GEM), estándar global de middleware, basado en el lenguaje Java.

Tres perfiles:

¹⁷ Proviene del ingles y significa tiempo de ejecución

Capítulo 1: Fundamentos Teóricos de la Investigación

- Enhanced Broadcasting: receptores y aplicaciones con interactividad local SIN canal de retorno.
- Interactive broadcasting: Idem CON canal de retorno.
- Internet access: Receptores y aplicaciones que, adicionalmente, tienen acceso a Internet. (Hervé, 2008)

Esta plataforma es la base para la creación de diferentes middleware como es el caso del middleware GINGA de Brasil. Ginga es el nombre del Middleware Abierto del Sistema Brasileño de TV Digital (SBTVD). Ginga es un conjunto normalizado de tecnologías e innovaciones, divide en dos grandes subsistemas interconectados, permitiendo el desarrollo de aplicaciones que utilizan dos diferentes paradigmas de programación. Dependiendo de las características requeridas en el diseño de cada aplicación, un paradigma es más adecuado que el otro. Estos dos subsistemas son llamados Ginga-J (de procedimiento para las aplicaciones Java) y Ginga-NCL (las solicitudes de declaración de NCL).

Uno de los principales inconvenientes de los middleware antes mencionados es que los set top box utilizados para recibir la señal debe ser compatibles con este. Los proveedores de servicios interactivos tienen que definir el middleware a utilizar para que los receptores se adapten a él. La solución que se propondrá debe ser adaptable a cualquier tipo de set top box además de los diferentes middleware que implemente el proveedor de servicios interactivos. El set top box solo debe cumplir la característica de tener un canal de retorno mediante el protocolo http, la cual cumplen la inmensa mayoría de los set top box IPTV existente en el mercado.

1.7 Estado actual en Cuba.

Como se mencionaba en la introducción del capítulo, en Cuba actualmente la infraestructura televisiva está completamente basada en tecnologías analógicas. Cuba está dando sus primeros pasos en la televisión digital terrestre y está trabajando en buscar el estándar más adecuado para comenzar el apagón analógico. Los estudios que se hacen actualmente son más bien enfocados a las formas de transmitir las señales por lo que no se tienen soluciones encaminadas a la explotación de la televisión interactiva sobre las redes de datos de Internet. Según Manuel Barreras, director de la Oficina cubana para la Digitalización de la Radio y la Televisión el apagón analógico pudiera llegar dentro de una década. En el transcurso de todo este tiempo estarán en paralelo

Capítulo 1: Fundamentos Teóricos de la Investigación

ambas tecnologías, digital y analógica. El apagón analógico irá a ser potenciado mayormente por la Televisión Digital Terrestre (TDT) y no con la IPTV. Esto trae como inconveniente que con la TDT se alcanza nada más los 2 primeros niveles de interactividad, no así con la IPTV que alcanza los 3 niveles.

1.8 Conclusiones Parciales.

Una de las conclusiones más importantes que se debe resaltar es la importancia que tiene los sistemas interactivos en la televisión actual. El futuro de la televisión depende de cuan interactivas sean las transmisiones y la cantidad de servicios que se brindan. Los set-top-box jugarán un papel fundamental para lograr dichos objetivos ya que estos brindan todas las funcionalidades para que la interactividad a través del televisor sea una realidad. Resaltar que el sistema a implementar es el primero de su tipo en el país y será más adaptable a los diferentes tipos de set-top-box y middleware que sean definidos tanto por el cliente como el proveedor de servicios.

CAPÍTULO 2: Herramientas para la solución

Introducción

Este capítulo es el resultado del estudio realizado sobre las tecnologías adecuadas para el desarrollo del sistema. En el se analizan la metodología de desarrollo, herramienta CASE¹⁸, entorno de desarrollo (IDE), lenguajes de programación, gestor de bases de datos y otras herramientas necesarias para lograr la evolución del sistema.

2.1 Metodología de Desarrollo de Software

Una metodología de desarrollo de software es aquella que hace posible la planificación, organización y desarrollo de un sistema o proyecto, independientemente de su temática o complejidad. Actualmente estas metodologías son una guía en el proceso de desarrollo de las aplicaciones informáticas, permitiendo que se obtengan resultados con la mayor calidad, rapidez y eficiencia posible, para evitar cometer errores futuros.

En el momento de comenzar una aplicación, normalmente se debe seleccionar la metodología de desarrollo a emplear, para lo cual existen distintos tipos. Las metodologías se clasifican en ágiles y pesadas o tradicionales. Estas últimas son más prácticas cuando el proyecto o aplicación a desarrollar es compleja y se tienen claros los requisitos de la misma. Además comprenden una definición detallada de los procesos y tareas a realizar, que sirven de apoyo, ya que generan documentación suficiente para una mejor comprensión, en este caso se encuentra el *Rational Unified Process* (RUP).

Por otra parte están las metodologías ágiles, las cuales son más propensas al cambio, a la falta de información, generan poca documentación y se conciben para proyectos más sencillos, ajustándose a equipos de desarrollo pequeños con poco tiempo de entrega del producto. Un ejemplo clásico de estas es *Extreme Programming* (XP), conocido en español como Programación Extrema. (Pressman, 2002)

De acuerdo a lo antes descrito y a las necesidades para la solución del problema investigativo, se define como metodología a aplicar: RUP. La decisión anterior se debe a que desde un inicio están bien definidos los requisitos de software, sobre los que no ocurrirán cambios constantes y además se necesita de una documentación completa y

¹⁸ Computer Aided Software Engineering o Ingeniería de Software Asistida por Ordenador

detallada. No se puede dejar de mencionar que la misma sigue la línea de desarrollo del proyecto al cual pertenece, además de ser actualmente una de las más utilizadas en la Universidad. A continuación se describe más detalladamente la metodología en cuestión.

RUP

Proceso Unificado de Desarrollo (*Rational Unified Process*, RUP) es una de las metodologías pesadas más conocidas y utilizadas. Esta es bastante robusta y precisa, permite controlar y documentar muy bien el desarrollo del software, eliminando los riesgos que puedan existir en el proceso, propiciando en todo momento un enfoque de trabajo bien estructurado, así como la asignación correcta y óptima de tareas y responsabilidades. (Jacobson, 2000)

Para guiar el proceso de desarrollo del software emplea nueve flujos de trabajo organizados en cuatro fases de desarrollo como muestra la Ilustración 3. Las cuatro fases de desarrollo que definen el ciclo de vida son:(Jacobson, 2000)

- Inicio: tiene como objetivo establecer la visión del proyecto o aplicación y lo que se quiere realizar.
- Elaboración: es donde se define la arquitectura y los elementos base para la implementación.
- Construcción: es donde se implementa y se va obteniendo una solución inicial parcial.
- Transición: es donde se adquiere el producto acabado y definido, incluyendo su mantenimiento.

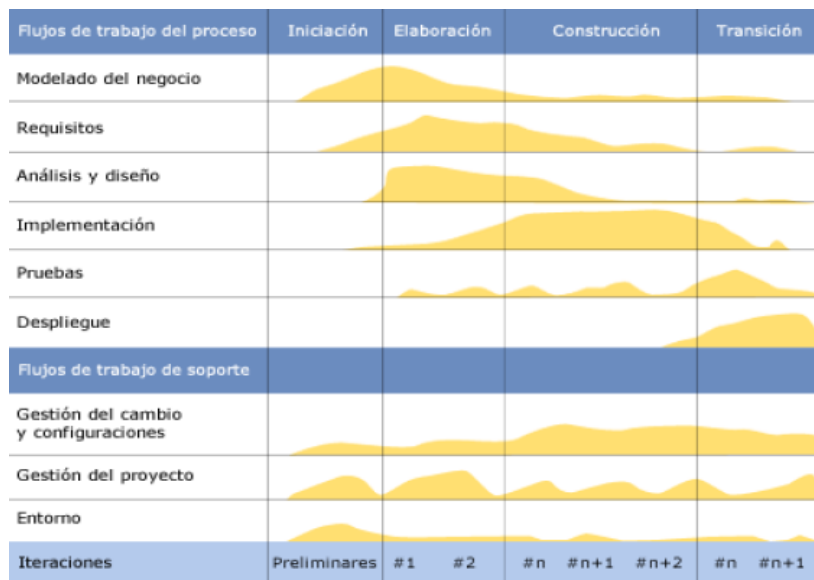


Ilustración 3 Diagrama del esfuerzo de actividades según la etapa del proyecto. (Jacobson, 2000)

Las características que definen a RUP son:

- **Guiado por Casos de Uso:** los casos de uso (CU) son una técnica de captura de requisitos que representan funcionalidades del sistema, las cuales definen lo que el usuario desea obtener y permiten guiar todo el ciclo de vida de la aplicación o proyecto, para crear un resultado que satisfaga las necesidades esperadas. Además integran a todos los flujos de trabajo de RUP, sirviendo de punto de partida y de hilo conductor. En otras palabras, esta característica es la que permite establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.
- **Centrado en la arquitectura:** la arquitectura es la organización o estructura de las partes más relevantes de un sistema, ya que brinda una perspectiva clara y una visión común del mismo entre todos sus implicados. En esta se describen los procesos del negocio que son más importantes, teniendo en cuenta los elementos de calidad, rendimiento, reutilización y flexibilidad. En otras palabras, es la que permite entender bien el sistema y la toma de decisiones que indican cómo tiene que ser desarrollada la aplicación y en qué orden.
- **Iterativo e incremental:** permite dividir el trabajo en partes más pequeñas conocidas como fases de desarrollo, donde cada una se puede ver como una iteración, de la cual se obtiene un incremento que produce un crecimiento en el producto. Estas iteraciones son planificadas, a medida que se obtienen sus resultados, se van integrando. Esta característica permite ir mejorando los resultados, para lograr una optimización y corrección de errores en el momento adecuado, perfeccionando así la aplicación final. En otras palabras, va iterando a medida que se van ejecutando las diversas fases y se vuelven a retocar para ser mejoradas o corregidas, mediante lo cual va incrementando la obtención de resultados y la optimización del sistema en general. Esto acarrea que no se cometan fallos en el desarrollo de la aplicación y que se adquiera calidad y experiencia. (Kruchten, 2000)

Lenguaje Unificado de Modelado (UML v8.0)

Es actualmente uno de los lenguajes de modelado más usado a nivel mundial por las grandes empresas productoras de software, ya que permite visualizar, especificar, desarrollar y documentar los artefactos de la aplicación de forma eficiente y entendible (Inc, 2003). Básicamente UML es un método gráfico para permitir a los desarrolladores de

una aplicación tener una visión de lo que desarrollarán y entender completamente la misma.

Para lograr lo antes descrito se emplean una serie de elementos llamados diagramas, los que representan las diferentes proyecciones del sistema según sea la fase de desarrollo en la que se encuentre, donde cada diagrama tiene fines distintos dentro del proceso de desarrollo. UML es un lenguaje fácil de aprender, bien descriptivo y permite documentar todo el proceso de creación del software, empleándose en todas sus etapas. (Giraldo, 2005)

2.2 Framework de desarrollo

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (Alfonso, 2010)

Las principales ventajas de la utilización de framework son:

1. Los frameworks son los paradigmas de la reutilización.
2. El desarrollo rápido de aplicaciones. Los componentes incluidos en un framework constituyen una capa que libera al programador de la escritura de código de bajo nivel.
3. El uso y la programación de componentes que siguen una política de diseño uniforme. Un framework orientado a objetos logra que los componentes sean clases que pertenezcan a una gran jerarquía de clases, lo que resulta en bibliotecas más fáciles de aprender a usar.

Symfony2 (2.0.1): es un conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones Web, es decir, Symfony2 es un framework para construir aplicaciones Web con el lenguaje de programación PHP. Además, este framework ha integrado en uno las ventajas de este y otros ya existentes, el resultado del mismo es un

framework estable, productivo y muy bien documentado. El mismo se diseñó para que se ajustara a una serie de requisitos entre ellos: que fuese fácil de instalar y configurar en la mayoría de plataformas, independiente del sistema gestor de base de datos, preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa que decidiera utilizarlo.

El framework symfony2 está basado en la arquitectura Modelo Vista Controlador. Posee un alto nivel de organización, es de desarrollo ágil y tiene buena seguridad, ya que utiliza la ORM doctrine, esto impide la inserción de inyecciones SQL a las bases de datos. También proporciona herramientas para agilizar aplicaciones complejas y guiando al desarrollador a trabajar de forma organizada.

JQuery 1.5.1: JQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Permite la manipulación, control de eventos, animaciones y peticiones Ajax mucho más simple que con otras librerías. Con una API¹⁹ fácil de usar que funciona en un gran número de navegadores web. Permite una amplia combinación de versatilidad y capacidad de ampliación. JQuery ha cambiado la forma en que millones de personas escriben JavaScript.

2.3 Lenguaje de desarrollo

Un lenguaje de desarrollo o programación es aquel elemento dentro de la Informática que nos permite crear programas usando instrucciones y operadores que se rigen por reglas. En otras palabras, es el lenguaje que emplean los desarrolladores para que la computadora realice las acciones que desean. Actualmente existen diversos tipos de lenguajes de programación que pueden clasificarse tanto en Alto o Bajo nivel, como en Estructurado u Orientado a Objetos, o con fines Web o Desktop. Algunos de estos lenguajes son PHP, C, Java y ASP.NET. (Marin Arias, 2008)

PHP v5.3

Es un lenguaje de alto nivel con técnicas de Programación Orientada a Objetos, multiplataforma, robusto, sencillo de usar, rápido, integrable, excelente para crear aplicaciones web dinámicas y robustas, además de ser de software libre. PHP es uno de

¹⁹ API acrónimo inglés de application programming interface que es una capa de abstracción para utilizar mejor una código determinado.

los lenguajes de programación que nos permiten programar scripts del lado del servidor, insertados dentro del código HTML, con una gran cantidad de librerías de funciones y mucha documentación. PHP5 ofrece mejoras significativas con respecto a versiones anteriores, con una orientación a objetos similar a la de Java y un rendimiento del nivel de la plataforma .NET, ya que es la versión libre del sistema equivalente de Microsoft ASP para la creación de aplicaciones complejas. (Herrera, 2006)

En resumen, con PHP se pueden hacer grandes cosas con pocas líneas de código. La práctica ha demostrado que PHP5 es mejor que Java para la creación de aplicaciones web dinámicas específicamente, ya que el desarrollo en PHP es mucho más rápido, y el rendimiento y la escalabilidad es más fácil de lograr que en Java. Además, Java depende de su lenta máquina virtual y la separación entre presentación y controlador o lógica de negocio que en JSP es bastante mala. Tanto Java como PHP5 son ampliamente aceptados, sin embargo muchas organizaciones utilizan PHP para sus aplicaciones web porque en Java es mucho más difícil hacer cosas simples. (Hardz Website 2007)

JavaScript 2.0:

Provee el comportamiento, tercer pilar en el actual paradigma de las aplicaciones web, el cual define a las páginas web como entes consistentes de tres partes claramente distinguibles: el contenido (HTML), la presentación (CSS) y el comportamiento (JavaScript). El código JavaScript se ejecuta sobre un ambiente anfitrión, habitualmente los navegadores web, quienes son los más comunes pero no los únicos. Esto quiere decir que posee carácter multiplataforma y además puede ser empleado en la programación de todo tipo de aplicaciones y entornos. (Stefanov, 2008)

JavaScript es un poderoso lenguaje prototipado orientado a objetos que actualmente se puede ejecutar en el lado servidor, en aplicaciones de escritorio y en medios enriquecidos (*rich media*), además es funcional en otra docena de aplicaciones como son:

- ✓ Crear poderosas aplicaciones web (ejecutables sobre navegadores web).
- ✓ Codificar aplicaciones del lado servidor, tales como ASP, o por ejemplo, código ejecutable usando Rhino²⁰.
- ✓ Programar *scripts* para automatizar tareas en los escritorios de Windows, a través del *Windows Scripting Host*.

²⁰ **Rhino** – Motor de JavaScript escrito en Java.

- ✓ Crear extensiones/*plugin* para aplicaciones de escritorio como son: Firefox, Chrome, Dreamweaver.
- ✓ Crear aplicaciones sobre Adobe Air, las cuales son ejecutables en ambientes de escritorio. (Stefanov, 2008)

CSS 3.0

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Este lenguaje se seleccionó porque tiene varios beneficios entre los que se pueden mencionar: que puede controlar la presentación de muchos documentos desde una única hoja de estilo, además de que permite emplear numerosas técnicas avanzadas y sofisticadas.

HTML 5

Siglas de HyperText Markup Language (lenguaje de marcado de hipertexto), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML no es más que un documento xml con etiquetas propias que lo identifican. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

2.4 Servidor Web

Un servidor web es un programa que, empleando el modelo cliente/servidor y el protocolo de transferencia de hipertexto (HTTP), brinda una salida al contenido desde las páginas web a sus usuarios, los que deben poseer equipos de cómputo con aplicaciones clientes de HTTP que manejen sus peticiones e interpreten las respuestas. Todo equipo de cómputo que reside en Internet, para brindar servicios web, debe tener un software servidor. Los dos más populares son *Microsoft's Internet Information Server (IIS)* y el más usado Apache, del cual a continuación se brindarán algunas características. (Rouse, 2008)

Apache 2.2

Apache es el servidor web de uso por excelencia, su fácil configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal, es una tecnología gratuita, de código abierto, es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache, cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada. Tiene una alta configurabilidad en la creación y gestión de *logs*. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor. (Foundation, 2011)

2.5 Entorno de Desarrollo Integrado (IDE)

En aras de agilizar el proceso de desarrollo de la solución propuesta y garantizando que se efectúe de una forma lo más entendiblemente posible, o sea, con formateo de código fuente, nomenclatura bien definida de los componentes obtenidos. Posibilitando además que se generen la menor cantidad de errores posibles durante la implementación de la plataforma se decide el empleo de uno de los IDE con mejores prestaciones actualmente y del que se agregan algunos elementos a continuación.

Netbeans v7.0.1

Se define a Netbeans como IDE debido a que responde a software libre, soporta lenguajes de alto nivel orientado a objetos como Java y PHP, así como *frameworks* de desarrollo tales como ExtJS y Symfony. Es bastante robusto, hace el trabajo más cómodo y fácil, siendo agradable para el usuario y sencillo de usar. Netbeans es un producto gratuito y no tiene restricciones de uso, ya que es de código abierto, tiene una comunidad que le ofrece soporte a nivel mundial. (Afiliates, 2011)

Este IDE permite que las aplicaciones sean desarrolladas a partir de módulos, lo cual posibilita que se pueda extender la aplicación, agregándole módulos nuevos. Netbeans incluye herramientas de esquemas XML, orientación a servicios web y modelado UML, así como soporte para PHP5. También permite a los desarrolladores crear web dinámica y posee bastantes fuentes de documentación. Unido a esto, soporta formato JSON y

HTML5, genera documentación PHP y puede ser integrado con varios sistemas. (Afiliates, 2011)

2.6 Herramienta CASE

CASE, *Computer Aided Software Engineering*, es un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Este tipo de herramientas son altamente utilizadas en la actualidad por muchas empresas de producción de software para la automatización y representación de los elementos fundamentales que componen el proceso de desarrollo de las aplicaciones o sistemas. Por tanto, aportan un alto valor económico y buenos resultados del producto a obtener. (Giraldo, 2005)

Estas herramientas se relacionan estrechamente con la metodología de software a emplear y con el lenguaje UML, puesto que sirven de apoyo para la puesta en práctica de los mismos. Existen tres tipos de herramientas CASE: las de Alto Nivel, que son las que apoyan las fases iniciales del ciclo de vida del desarrollo del software, las de Bajo Nivel, que apoyan solamente las fases finales, y finalmente las de Cruzado de Ciclo de vida, que tienen lugar a lo largo de todo el proceso de desarrollo. Ésta última clasificación es la más recomendada, ya que asegura guiar y entender el proceso en todo momento, en este caso entran Visual Paradigm y Rational Rose. (Giraldo, 2005)

Visual Paradigm for UML v8.0 Enterprise Edition v5.0

Visual Paradigm For UML es una herramienta CASE cruzado de ciclo de vida multiplataforma, la cual pertenece a la categoría de software libre, brindando muchas facilidades con esto, tales como buena documentación y soporte online. Además es una herramienta fácil de utilizar, con una interfaz agradable, presenta soporte para la notación y modelado de procesos de negocios, y emplea una rápida respuesta con poca memoria utilizando moderadamente los tiempos del procesador, lo que le permite manejar grandes y complicadas estructuras de un proyecto en una forma muy eficiente. (International, 2011)

También cuenta para la parte de base de datos, con un generador de mapeo de objetos-relacionales para los lenguajes de programación Java, .NET y PHP. Unido a ella, permite su integración con otras herramientas como son los IDE Netbeans y Eclipse. Esta

herramienta es reconocida porque realiza de forma clara y organizada la diagramación visual y el diseño de proyectos, permitiendo así integrar y desplegar las aplicaciones (Giraldo, 2005). Visual Paradigm es apoyado por un conjunto de idiomas tanto en la generación del código, como en la ingeniería inversa, ya que proporciona compatibilidad con 10 lenguajes, entre ellos C#, Delphi, C++, Java y PHP (Aurora Vizcaíno, 2010). Agregando a todo lo anterior que es una herramienta con alta interoperabilidad y que tiene también conexión con Rational Rose en sus archivos de proyecto, así como importación y exportación a formato XML. (International, 2011)

Resumiendo, se selecciona esta herramienta para el diseño y modelado de la solución del problema de la investigación, ya que presenta grandes ventajas por encima de otras similares, esto se demuestra con sus características, como son la navegación entre código y modelo, su sincronización de código fuente, su poderoso generador de documentación y reportes UML PDF/HTML/MS Word, su alto entorno de modelado visual, su soporte completo de notaciones UML y aplicaciones Web, sus diagramas de diseño automáticos sofisticados y su fácil instalación y actualización.

2.7 Sistema gestor de base de datos

Una base de datos(BD) es una aplicación que gestiona, organiza y permite un rápido almacenamiento y recuperación de los datos. La mayor parte de las BD contienen múltiples tablas, cada tabla pueden incluir una cantidad diferentes de campos que son relevantes a la información almacenada en ellas.

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (González, 2000)

PostgreSQL 9.1: El SGBD relacional orientado a objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley, distribuida bajo licencia BSD²¹. Con más de una década de desarrollo tras él, PostgreSQL ha demostrado ser un gestor de bases de datos de código abierto muy avanzado, ofreciendo control de concurrencia multiversión (MVCC), soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación como pueden ser C, C++, Java, Perl, TCL y Python. (PostgreSQL, 2010)

La lista de lenguajes escritos anteriormente puede actualizarse añadiendo al Object Pascal, PHP, Ruby, Pike y C#.

La utilización de este SGBD en la plataforma interactiva obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad. Trabajar con este gestor es factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. Consecuentemente tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Además sigue la línea del proyecto STCV ya que todos los datos están resguardados bajo este SGBD.

2.8 Conclusiones parciales

La idea fundamental para la selección de las herramientas y tecnologías es que fueran libres, siguiendo con las políticas del país para el uso de tecnologías de código abierto. Se consideraron además que fueran multiplataforma para su despliegue en diferentes entornos y su fácil migración. También es importante destacar que la aplicación que se desarrolló al estar enmarcada como parte importante de los productos que implementa el proyecto Sistema de Transmisión de Canales Virtuales hace uso de muchas tecnologías del citado proyecto como son la metodología de desarrollo (RUP) y el SGBD PostgreSQL.

²¹ Licencia de software otorgada principalmente para los sistemas (Berkeley Software Distribution) BSD. Es una licencia de software libre, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público, permite el uso del código fuente en software no libre.

CAPÍTULO 3: Descripción de la solución propuesta.

Introducción

Para lograr la transición al Diseño normalmente se utiliza el flujo de trabajo Análisis debido a que según lo propuesto por RUP este le antecede durante el proceso de desarrollo de software. El flujo de trabajo análisis es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software. Se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y el modelo completo del análisis es creado. (Pressman 2005)

Una variante válida es no utilizar en absoluto el modelo del análisis para describir los resultados del Análisis, producto a que los requisitos se analizan como parte integrada de la captura de requisitos, lo que trae como exigencia durante el proceso un mayor formalismo en la realización del modelo de casos de uso (Jacobson, y otros, 2000).

RUP es una metodología de desarrollo muy flexible y adaptable a los diferentes entornos en que se utiliza. Es posible con esta metodología tomar en cuenta simplemente aquellas partes que se consideren necesarias, por lo que se decidió prescindir de la realización del modelo de análisis en el desarrollo de esta investigación debido a que:

- ✓ Desde un inicio están bien definidos los requisitos de software, sobre los que no ocurrirán cambios constantes, y con una comprensión detallada de los mismos.
- ✓ Se logra evitar el costo en tiempo y recursos de mantener este flujo.

Por las razones antes mencionadas se decide no realizar el flujo de trabajo del análisis pasando directamente al diseño del sistema.

En el presente capítulo se realiza la descripción detallada del sistema a desarrollado, se identificarán los actores, y los casos de usos que permitirán una mejor comprensión y abstracción sobre las características del sistema. Además se especifican los requerimientos funcionales y no funcionales que debe tener el sistema, junto con su descripción textual; así como los distintos diagramas necesarios para entender los procesos que se efectúan en la plataforma interactiva.

3.1 Modelo de Dominio

El modelo de dominio representa la realidad de las interacciones entre los objetos del proyecto. Este es un diagrama que está compuesto por los objetos que se identifican como reales que se relacionan con el proyecto que se desea desarrollar y las relaciones existentes entre los mismos. La importancia del modelo de dominio es que ayuda a comprender mediante un diagrama de clases los conceptos más significativos sobre cómo es en realidad el sistema, sus relaciones y sus procesos. (Laman, 2004)

En la siguiente figura se muestra el modelo de dominio identificado para la aplicación propuesta:

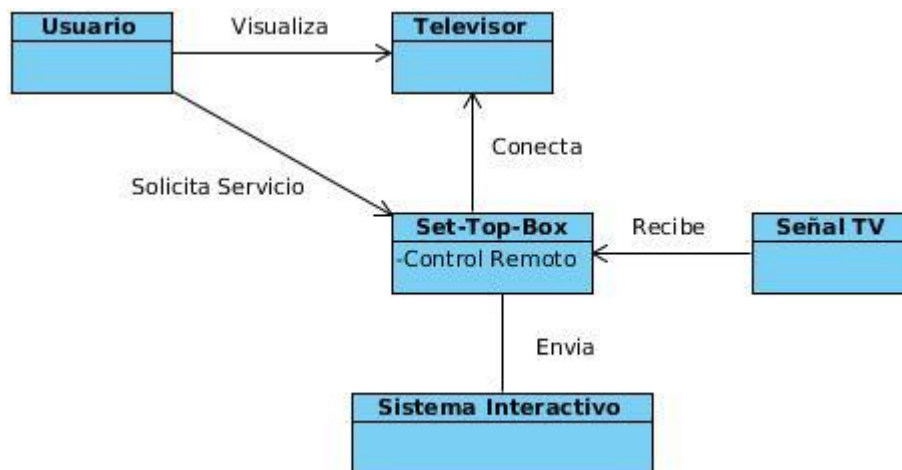


Ilustración 4 Modelo de Dominio

3.1.1 Glosario de Términos del Dominio

Clase Usuario: Persona suscrita al servicio de televisión digital.

Clase Set-Top-Box: Dispositivo encargado de decodificar las señales digitales a analógicas para que los televisores tradicionales puedan disfrutar de la TV digital. Es el dispositivo fundamental para lograr la interactividad.

Clase Televisor: Dispositivo electrónico para ver canales de televisión.

Clase Sistema Interactivo: Es el sistema encargado de brindar una serie de servicios a los usuarios de televisión digital. Es la plataforma que se desarrolla.

3.2 Requisitos

En el desarrollo de software se tiene presente la arquitectura, flujo imprescindible para llevar a feliz término el producto final. Esta se basa en los análisis detallados de los requerimientos funcionales que constituyen capacidades o condiciones que el sistema debe cumplir para lograr un objetivo determinado (Jacobson, 2000) y los requisitos no funcionales que representan las propiedades o cualidades que el sistema debe tener. (Jacobson, 2004)

La clave del éxito del desarrollo de software es la correcta especificación de los requisitos, tanto funcionales como no funcionales.

3.2.1 Requisitos Funcionales

- **RF 1: Autenticar Usuario:** Controla el acceso a las funcionalidades según los permisos de los usuarios.
- **RF 2: Mostrar canales en transmisión:** Mostrar los canales que se encuentran en transmisión.
- **RF 3: Consultar cartelera de programación:** Debe permitir por parte del usuario consultar la cartelera para cada uno de los canales en transmisión.
- **RF 4: Enviar Solicitud de transmisión:** Debe permitir enviarle una solicitud de transmisión al transmisor. El transmisor manda a reproducir lo deseado por el cliente.
- **RF 5: Mostrar información de canales:** Mostrar información de los canales en transmisión, tales como número, el nombre del canal y descripción del mismo.
- **RF 6: Mostrar lista Medias Disponibles:** Permitir mostrar las medias disponibles para consumir video bajo demanda.
- **RF 7: Jugar:** Permitir que los usuarios puedan jugar mientras esperan un programa x.
- **RF 8: Mostrar Noticias:** Permitir a los usuarios mostrar noticias de un servidor RSS.

Capítulo 3: Descripción de la solución propuesta

- **RF 9: Internacionalizar el sistema:** La aplicación debe poder mostrarse en varios idiomas.
- **RF 10: Adicionar Infocinta:** Permitir adicionar infocinta.
- **RF 11: Eliminar Infocinta:** Poder eliminar las infocintas que desee el administrador.
- **RF 12: Modificar Infocinta:** Permitir modificar las infocintas existentes.
- **RF 13: Listar Infocintas:** Debe poder listar todas las infocintas existentes.
- **RF 14: Mostrar Log:** Debe mostrar los log generados por el sistema en aras de agilizar el proceso de mantenimiento.
- **RF 15: Cerrar sesión:** Debe poder cerrar la sesión del usuario autenticado.
- **RF 16: Configurar RSS:** Permitir al usuario poder cambiar el RSS²² de noticias.
- **RF 17: Consumir video bajo demanda:** Poder ver un video o una lista de videos bajo demanda.

3.2.2 Requisitos No Funcionales

- **RNF 1: Software:** Debe estar basado en tecnologías libres.
 - Sistema Operativo: Multiplataforma
 - Servidor Web: Apache 2.2.
 - Servidor de Bases de Datos: PostgreSQL.
 - Paquetes: PHP5-cli, PHP5-pgsql.
- **RNF 2: Hardware:**
 - Cliente:
 - Televisor
 - Set Top Box
 - Servidor:

²² Es un formato para la suscripción a servicios de noticias.

-Procesador: Intel Core 2 Duo a 2.20 Ghz

- RAM: 4GB

- Disco Duro: 80 GB

- Red: Tarjeta de red Ethernet 100 Mbps.

- **RNF 3: Usabilidad:** La aplicación debe ser manejada tanto por usuarios domésticos como por empleados de instituciones con conocimientos básicos en informática.
- **RNF 4: Requisitos de apariencia o interfaz externa:** La interfaz externa de la aplicación debe ser lo más amigable posible además de fácil navegación por toda la aplicación con la utilización únicamente de un control remoto.
- **RNF 5: Restricciones en el diseño y la implementación:** La aplicación debe poder ejecutar todas las funcionalidades para los diferentes tipos de set-top-box que tengan conexión a Internet y una API java script que permita la interactividad. La aplicación debe estar orientada a plugins²³, es decir para los diferentes tipos de set top box solo hay que hacer el plugins correspondiente para que el sistema funcione correctamente.

3.3 Modelo del Sistema

3.3.1 Diagrama de Casos de Uso del Sistema (DCUS)

Los diagramas de casos de uso se utilizan para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y entre otros sistemas. Además estos diagramas de casos de uso ilustran los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo y se especifican las relaciones entre los actores y los casos de uso. (Jacobson, 2004)

A continuación se muestra el diagrama de casos de uso del sistema que se implementa.

²³ Fragmento de código que cumple una funcionalidad específica dentro de una aplicación.

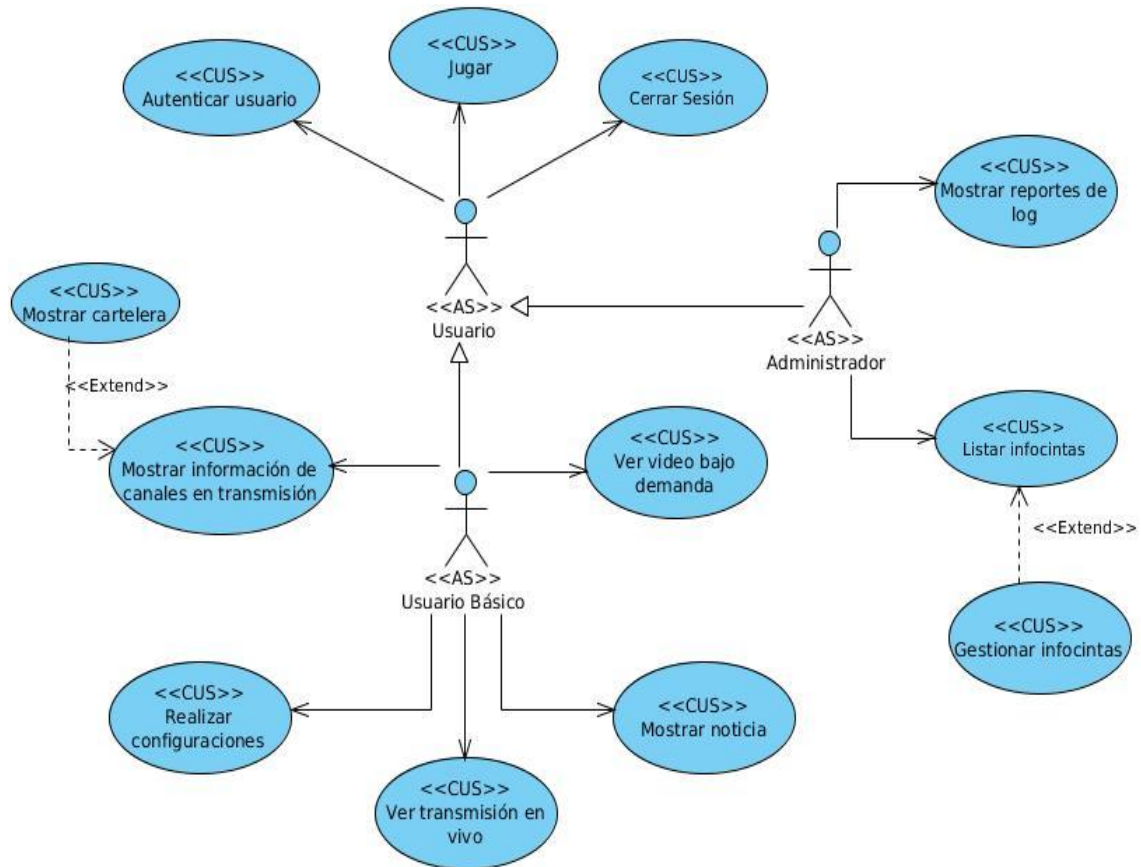


Ilustración 5 Diagrama de casos de uso del sistema

- **Autenticar usuario:** Permite tener privilegios según tipo de usuario a las diferentes funcionalidades del sistema. Para autenticarse, el usuario necesita introducir el nombre de usuario y la contraseña de acceso al sistema.
- **Jugar:** Da la posibilidad de entretener al usuario mientras espera un programa determinado. El sistema cuenta con juegos tradicionales como tetris, buscaminas y ahorcado.
- **Cerrar sesión:** Quita los privilegios que el usuario tenía en el sistema y pasa a ser un usuario común.
- **Mostrar información de canales en transmisión:** Muestra información de los canales en transmisión tales como nombre, dirección, logo y descripción de los mismos.
- **Mostrar cartelera:** Permite visualizar la cartelera para el canal seleccionado, mostrando información de la hora de inicio de un programa determinado para ese canal.

Capítulo 3: Descripción de la solución propuesta

- **Realizar configuraciones:** Permite cambiar las configuraciones de idioma del sistema, teniendo como lenguajes por defecto ingles-español.
- **Ver transmisión en vivo:** Configura el set top box para que pueda ver los canales en transmisión en el televisor, con solo presionar el número del canal en el control remoto del set-top-box.
- **Mostrar Noticias:** Permite poder visualizar las noticias que existan en una canal RSS y poder configurar ese canal RSS.
- **Ver Video bajo demanda:** Lista las medias publicadas en el proyecto STCV y permite consumir video bajo demanda de las medias seleccionadas en la lista.
- **Mostrar repostes de log:** Permite visualizar los log de reporte que el sistema halla guardado.
- **Listar Infocintas:** Muestra un listado de todas las infocintas que existan en el sistema.
- **Gestionar Infocintas:** Permite crear, eliminar y modificar las infocintas.

3.3.2 Descripción de los actores del sistema

Los actores son terceros fuera del sistema que interactuarán con la aplicación solicitando y recibiendo información en aras de obtener un resultado.

Tabla 1 Actores del sistema

Usuario	Actor humano. Este usuario tiene privilegios mínimos en el sistema y puede leer las infocintas y jugar.
Usuario Básico	Actor humano. Encargado de utilizar todas las funcionalidades del sistema.
Administrador	Actor humano. Encargado de gestionar las infocintas y ver los reportes log.

3.3.3 Descripción de los casos de usos del sistema.

Caso de uso *Mostrar información de canales en transmisión.*

Capítulo 3: Descripción de la solución propuesta

Tabla 2 Descripción del caso de uso Mostrar información de canales en vivo

Objetivo	Mostrar la información de los canales en transmisión.	
Actores	Usuario Básico: (Inicia) Mostrar información de canales en transmisión.	
Resumen	El CU se inicia cuando el Usuario Básico desea consultar la información de un canal en transmisión, dicha información puede ser dirección IP, Puerto, Descripción, Número de canal, Logo y Nombre. Termina este CU cuando se haya mostrado dicha información.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	Existe al menos un canal en transmisión. Existe conexión la BD.	
Postcondiciones	Se mostró la información de los canales en transmisión.	
Flujo de eventos		
Flujo básico Consultar cartelera de programación		
	Actor	Sistema
1.	Selecciona la aplicación EPG.	
2.		Muestra en el contenido de la página todos los canales en transmisión así como las diferentes propiedades de estos (Ip, Puerto, Descripción, Número de canal, Logo y Nombre). Termina el caso de uso.
Relaciones	CU Incluidos	No existen.
	CU	Ver CU Mostrar cartelera.

Capítulo 3: Descripción de la solución propuesta

	Extendidos	
Requisitos no funcionales	RNF 2 Para el funcionamiento del módulo Programación Interactiva el sistema requiere la instalación de Set-top-box, y un televisor.	
Asuntos pendientes	No existen.	

Caso de uso *Mostrar Cartelera*.

Tabla 3 Descripción del caso de uso *Mostrar Cartelera*

Objetivo	Mostrar cartelera.	
Actores	Usuario Básico: (Inicia) Mostrar cartelera	
Resumen	El CU se inicia cuando el Usuario Básico desea consultar la cartelera para un determinado canal en transmisión. Termina este CU cuando se haya realizado la consulta de la cartelera.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	Existe al menos un canal en transmisión. Existe conexión con la BD. Estar listado los canales en transmisión.	
Postcondiciones	Se consultó la cartelera de programación.	
Flujo de eventos		
Flujo básico Consultar cartelera de programación		
	Actor	Sistema
1	Selecciona un canal de los canales que se muestran en transmisión.	
2		Muestra la cartelera del canal para ese día.

Capítulo 3: Descripción de la solución propuesta

		Termina el caso de uso.
Relaciones	CU Incluidos	No existen.
	CU Extendidos	No existen.
Requisitos no funcionales	RNF 2 Para el funcionamiento del módulo Programación Interactiva el sistema requiere la instalación de Set-top-box, y un televisor.	
Asuntos pendientes	No existen.	

3.4 Patrones

Un patrón detalla un problema que ocurre una y otra vez en el ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que se puede usar esa solución un millón de veces más, sin repetir lo mismo dos veces (González Jiménez, 2011).

3.4.1 Patrones de arquitectura

Los patrones de arquitectura de software son patrones de diseño de software que constituyen una vía en la solución de problemas de arquitectura de software. Los mismos poseen un nivel de abstracción mucho mayor que los patrones de diseño. Además brindan una descripción de los elementos y el tipo de relación que tienen, así como las restricciones a para su uso. Los patrones se especifican describiendo los componentes, con sus responsabilidades, relaciones, y las formas en que colaboran. (Buschmann F, y otros, 1996)

Los patrones de arquitectura escogidos fueron modelo-vista-controlador, cliente-servidor y basado en componentes. El primero se utilizó ya que el framework symfony2 lo usa en su funcionamiento interno. Este patrón de arquitectura es una de los más usados en los framework web y presenta las siguientes ventajas:

- Posee soporte para múltiples vistas, debido a que la Vista se separa del Modelo y no hay ninguna dependencia directa entre ambos, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos al mismo tiempo.

Capítulo 3: Descripción de la solución propuesta

- Proporciona un mecanismo de configuración a componentes complejos mucho más tratable que los patrones puramente basados en eventos.
- Permite un mayor soporte a los cambios, debido a que los requisitos de interfaz tienden a cambiar más rápidamente que las reglas de negocio. Puesto que el modelo no depende de las vistas, la adición de nuevos tipos de vista al sistema generalmente no afectan al modelo. Por tanto, el ámbito del cambio se limita a la vista. (Burbeck, 2002)

El patrón cliente/servidor define una relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor fuente de datos) (César de la Torre Llorente, 2010). La elección del patrón cliente/servidor se debe a que es una aplicación web la cual se distribuye las tareas entre cliente y servidor, siendo el cliente el que realiza peticiones al servidor. Entre sus principales ventajas se encuentran:

- Posee un fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, un servidor, mientras que los clientes no se verán afectados por ese cambio o la afectación será mínimamente.
- Facilita la integración entre sistemas diferentes, compartiendo tanto información como recursos.
- Contribuye a proporcionar a los diferentes departamentos una organización al brindar soluciones locales y permitir la integración de la información. (Gamma, 2005)

Para lograr llegar a una gran diversidad de dispositivos set-top-box el sistema debe estar orientado a *plugins*, esto se logra con una arquitectura basada en componentes, de ahí el uso del patrón arquitectónico basado en componentes. Entre los principales beneficios de este patrón se encuentran:

- Fácil despliegue ya que se puede sustituir un componente por su nueva versión sin afectar a otros componentes o al sistema.
- Reducción de costes ya que se pueden usar componentes de terceros para abaratar los costes de desarrollo y mantenimiento.

- Reusables ya que son independientes del contexto se pueden emplear en otras aplicaciones y sistemas.
- Reducción de la complejidad gracias al uso de contenedores de componentes que realizan la activación y gestión del ciclo de vida. (César de la Torre Llorente, 2010)

3.4.2 Patrones de diseño

Un patrón de diseño constituye una solución estándar para un problema común de programación en el desarrollo del software. Además es una técnica muy eficaz para flexibilizar el código haciéndolo satisfacer ciertos criterios, así como permite una manera más práctica de describir ciertos aspectos de la organización de un programa. (Gamma, 2005)

Patrones GRASP

Experto: Es la clase encargada de manipular cierta información ya que ella es la experta en dicha información. El framework symfony lo utiliza en las clases entidades ya que estas son las encargadas de acceder exclusivamente a los datos del sistema.

Creador: El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Una de las características más poderosas del framework symfony2 es la utilización del contenedor de inyecciones de dependencias, que es un objeto que sabe cómo crear los objetos de la aplicación. Para ello, conoce todas las relaciones entre tus clases y la configuración necesaria para instanciar correctamente cada clase.

Controlador: Este patrón define quién debe encargarse de atender un evento del sistema. El controlador es una parte vital de todo proyecto web ya que es el encargado de procesar las diferentes peticiones hechas por el cliente y se utiliza en el sistema con esos fines. Cada aplicación (EPG, VoD, Noticias, Juegos) con que cuenta el sistema que se implementa tiene asociado un Controlador para el manejo de los eventos.

Bajo Acoplamiento: Este patrón da la medida de la fuerza con que una clase está conectada a otras clases, a las que conoce y recurre a ellas. Un buen ejemplo de este patrón es la clase ExceptionListener que se puede reutilizar en otros proyectos y no tiene dependencia de clases.

Patrones GOF

Observador: Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. El framework symfony lo utiliza para la captura de errores ya que el núcleo del sistema captura los errores y notifica a todas las clases que estén “escuchando” dicho error.

3.5 Diagramas de clases del diseño

Representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** herencia, composición, agregación, asociación y uso.

Los diagramas de clases del diseño constituyen un elemento fundamental en la concepción de la aplicación que se propone, ya que servirán de guía a los desarrolladores al constituir una aproximación del sistema que se desea implementar, contribuyendo de esta forma a la calidad del producto final. (Pressman, 2002)

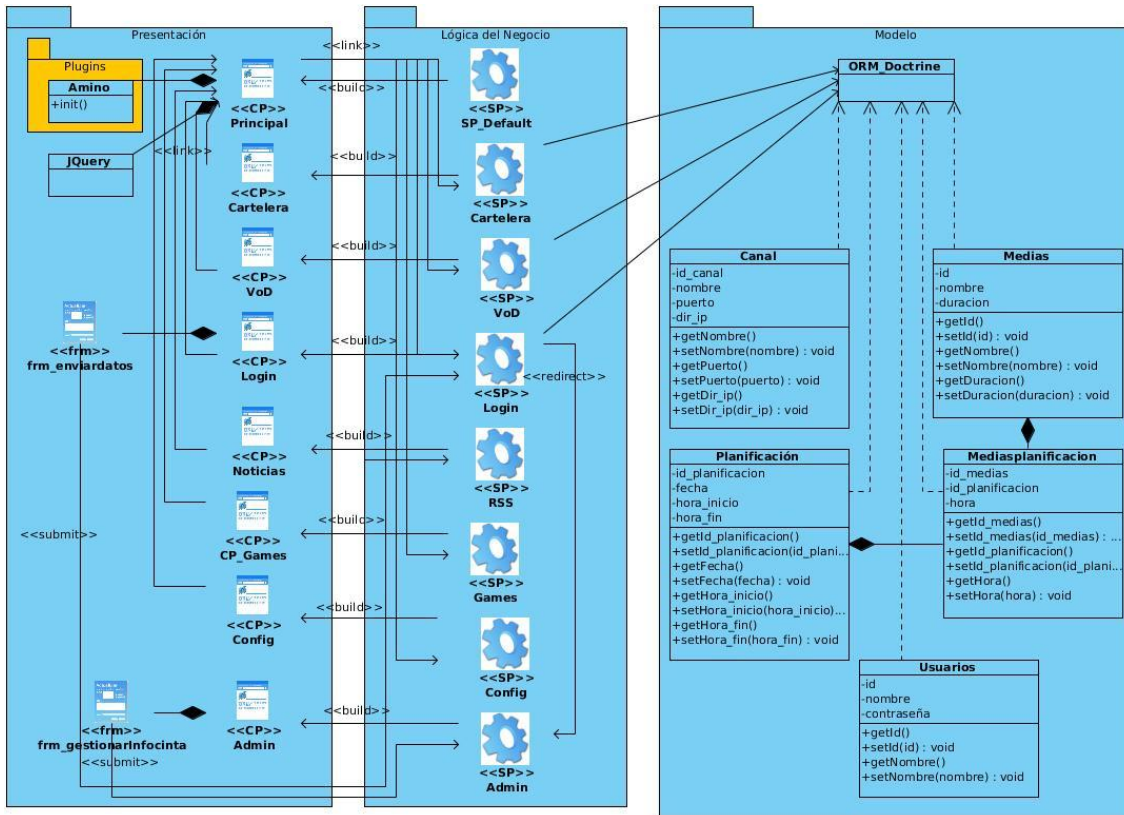


Ilustración 6 Diagrama de clases del diseño

3.6 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. En la siguiente ilustración se muestra como quedarán distribuidos los componentes del sistema.

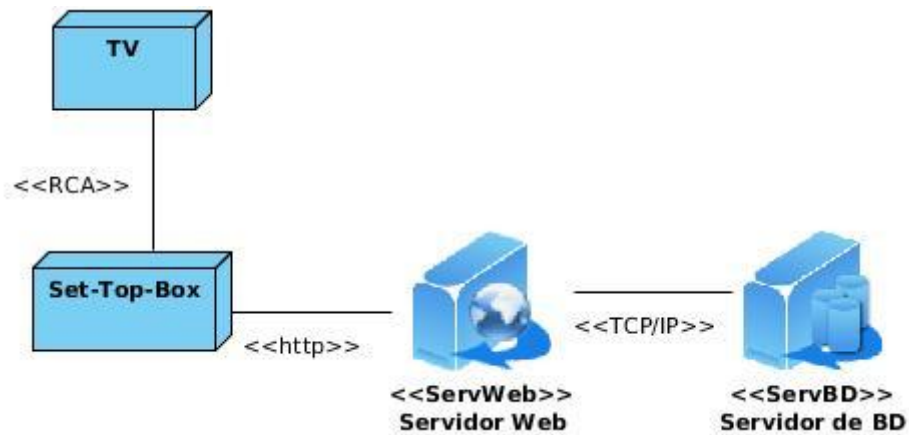


Ilustración 7 Diagrama de despliegue.

Capítulo 3: Descripción de la solución propuesta

- **TV:** Televisor que se encarga de visualizar los contenidos interactivos.
- **Set-Top-Box:** Dispositivo que permite mostrar los contenidos digitales en los televisores analógicos.
- **Servidor Web:** Servidor de aplicaciones web donde se encuentra alojada la plataforma interactiva.
- **Servidor BD:** Servidor de base de datos donde esta corriendo el postgresql para almacenar toda la información relacionada con la planificación y programación de canales.

3.7 Conclusiones parciales

Para la propuesta de solución se tuvieron en cuenta muchas de las características con que cuentan los sistemas interactivos actuales. Se definieron 17 requisitos funcionales siendo los del 2 al 8 los de mayor impacto en el sistema, ya que estas son las aplicaciones más comunes en estos entornos. Es imprescindible tener en cuenta el requisito no funcional 5, que implica que la plataforma interactiva sea orientada a plugins ya que es la base para que pueda ser utilizada en una gran diversidad de dispositivos. Una de las características más importantes que debe tener un desarrollador de software es el conocimiento de los diferentes patrones que existen. Los patrones dan una poderosa intuición para resolver problemas del desarrollo, ya que son soluciones ya probadas y documentadas que agilizan el diseño, implementación, las pruebas y el mantenimiento de la aplicación desarrollada.

CAPÍTULO 4: Implementación y prueba

Introducción

El presente capítulo está destinado a mostrar como se transforma el análisis y diseño vistos en el capítulo anterior en un software listo para su utilización. Como parte de la elaboración de un software se mostrará el diagrama de componentes, la codificación utilizada y las pruebas de caja negra hechas para validar el correcto desempeño del sistema.

4.1 Modelo de Implementación

El modelo de implementación describe cómo las clases se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modelación disponible en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes uno de otros. (MeRinde, 2010)

La siguiente ilustración muestra como colaboran los diferentes componentes en el diagrama de implementación de la plataforma interactiva.

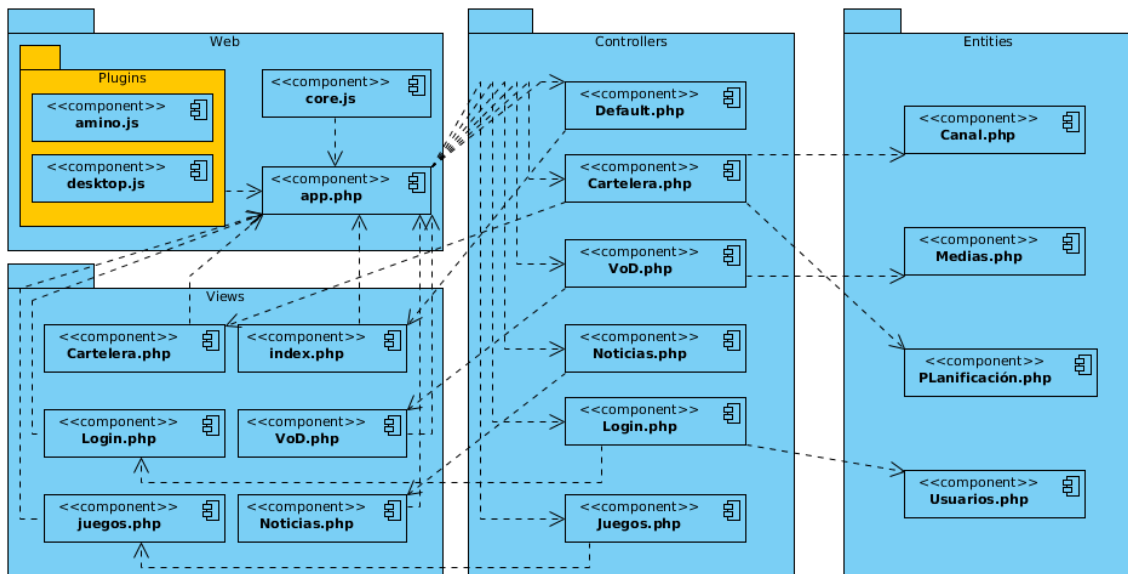


Ilustración 8 Diagrama de componentes

4.2 Estándares de codificación

El estándar en el código de programación es muy importante en cualquier proyecto, principalmente si este involucra muchos desarrolladores. Esto asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener.

Es importante que durante la codificación se consideren permanentemente los siguientes criterios de calidad:

- **Facilidad de Comunicación:** Proporcionar al usuario entradas y salidas fácilmente asimilables.
- **Autodescripción:** Proporcionar en el código, explicaciones sobre la implantación realizada.
- **Simplicidad:** La implantación realizada debe hacerse de la forma más comprensible posible.

Los estándares utilizados para el desarrollo del sistema fueron los siguientes:

Notación Camello

Consiste en escribir los identificadores de variables o funciones con la primera letra en mayúscula y el resto en minúscula o viceversa en dependencia de la variante que escojas. El nombre se debe porque los identificadores recuerdan las jorobas de un camello. Un ejemplo de su utilización en el código es:

```
public function indexAction()
{
    $em = $this->getDoctrine()->getEntityManager();
    $canales = $em->getRepository('InteractiveBundle:Canal')->findAll();
    if(!$canales)$this->container->get('msgExc')->sendWarningMessage('No existen canales');
    return $this->render('InteractiveBundle:appEpg:index.html.twig', array(
        'name'=>'epg',
        'canales'=>$canales
    ));
}
```

Ilustración 9 Ejemplo de notación camello

Notación Pascal

Es una forma de definir las variables de nuestro entorno comenzando siempre en mayúscula la letra inicial de la variable así como las siguientes palabras contenidas dentro de esta. Ejemplo de ello en el código de la aplicación es:

```
function SendAjax(url,data,func){
    var xhr=null;
    if (window.XMLHttpRequest) {
        xhr=new XMLHttpRequest();
    }
    else if (window.ActiveXObject) {
        xhr= new ActiveXObject("Microsoft.XMLHTTP");
    }
    return xhr;
}
```

Ilustración 10 Ejemplo de notación pascal

Es importante aclarar que estos estándares también son parte de los frameworks que se utilizaron y todas las clases y métodos de estos frameworks hacen un uso extensivo de ellos.

4.3 Pruebas

Las pruebas son una actividad en la cual el sistema o los componentes son ejecutados bajo ciertas condiciones o requerimientos especificados donde los resultados son observados y registrados para realizar una evaluación de algún aspecto del sistema o del componente. Entre las buenas prácticas para realizar pruebas al sistema es que dichas pruebas se realicen por un actor externo que no sea el desarrollador de la aplicación. (Quesada, 2007)

Objetivos de las pruebas

Las pruebas buscan encontrar los posibles fallos en la implementación, en la usabilidad y en la calidad de un programa determinada para validar su correcto funcionamiento. Entre los objetivos de la realización de las pruebas se tiene:

- Detectar defectos en la aplicación.
- Verificar que todos los requisitos se han implementado satisfactoriamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el producto al usuario o cliente final.

- Diseñar los casos de prueba que permitan identificar diferentes clases de errores utilizando la menor cantidad de tiempo y esfuerzo. (Quesada, 2009)

4.3.1 Tipos de pruebas

Las pruebas poseen una estructura en la que se define el objetivo de la prueba a realizar, la descripción detallada de la prueba y la técnica a utilizar. RUP define cuatro niveles de prueba:

- **Pruebas de unidad:** se realizan durante la fase de construcción, específicamente en el flujo de trabajo de implementación; las cuales se basan en probar los componentes implementados como unidades individuales. Las pruebas de unidad están divididas en dos grupos, pruebas de caja blanca y pruebas de caja negra. (Quesada, 2009)
- **Pruebas de integración:** se llevan a cabo durante la fase de construcción, las mismas involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional. Verifican que los componentes interactúan entre sí de un modo apropiado después de haber sido integrados en el sistema. Se toman como casos de prueba los casos de uso del diseño. Para ello se utiliza el diagrama de secuencia correspondiente y se diseñan combinaciones de entrada y salida del sistema que lleven a distintas utilidades de las clases y en consecuencia de los componentes, que participan en el diagrama. (Quesada, 2009)
- **Pruebas de sistema:** prueban que el sistema funciona globalmente de forma correcta. Cada prueba del sistema prueba combinaciones de casos de uso bajo condiciones diferentes. Se prueba el sistema como un todo probando casos de uso unos detrás de otros y si es posible, en paralelo. En este nivel existen una gran variedad de pruebas que se utilizan según el interés que se tenga respecto al funcionamiento del software. (Quesada, 2009)
- **Pruebas de aceptación:** se realizan para permitir que el cliente valide y verifique todos los requisitos pactados. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo del sistema. El cliente es quien impone los requisitos pues quien mejor que él para dar fe de su satisfacción. (Quesada, 2009)

Para validar el correcto desempeño del sistema implementado se escogieron las pruebas unitarias, específicamente la de caja negra para validar que el sistema responde adecuadamente a los diferentes requisitos del diseño.

4.3.2 Pruebas de caja negra

Las pruebas de Caja negra se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. (Jacobson, Booch, Rumbaugh, 2000)

La prueba de caja negra para Pressman (Pressman, 2005), intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Casos de Prueba

Los casos de prueba son el diseño de un conjunto de variables o condición de entrada para demostrar que los requisitos de una aplicación se cumplen parcial o completamente. Estos casos de prueba normalmente se describen por cada caso de uso del sistema y contiene un identificador, la descripción del caso de uso y variables asociada a dicho caso de uso.

Caso de Prueba del Caso de Uso Autenticar.

Descripción general: El caso de uso se inicia cuando el usuario selecciona la aplicación autenticar y así acceder a todas la funcionalidades del sistema. El caso de uso termina cuando el usuario esta autenticado.

Condiciones de Ejecución: No tiene.

Escenario: Autenticar.

Capítulo 4: Implementación y prueba

Escenario	Descripción	usuario	contraseña	Respuesta de sistema	Flujo central
EC 1.1 Autenticarse correctamente	Se introduce el usuario y contraseña válida para la autenticación.	V asielhv	V Password,segur0	Refresca la página, con el usuario ya autenticado.	URL de acceso/link autenticar
EC 1.2 Autenticarse incorrectamente	Se introduce el usuario y contraseña inválido.	V asielhv I asiel	I Password V Password,segur0	Se muestra el mensaje: "El usuario o la contraseña es incorrecta"	URL de acceso/link autenticar
EC 1.3 Autenticarse con campos vacíos.	Se introduce el usuario o la contraseña con campos vacíos.	I Campo Vacío V asielhv	V Password,segur0 I Campo Vacío	Se muestra el mensaje: "Los campos no pueden estar vacíos"	URL de acceso/link autenticar

Caso de Prueba del Caso de Uso Mostrar noticias.

Descripción general: El caso de uso se inicia cuando el usuario desea cambiar el rss de donde obtiene las noticias. El caso de uso termina cuando el usuario cambia las configuraciones de rss.

Condiciones de Ejecución: Debe estar autenticado.

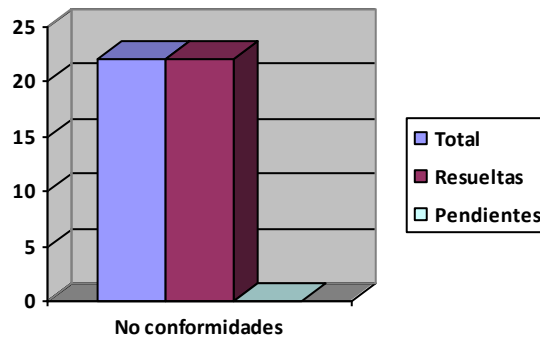
Escenario: Cambiar rss.

Escenario	Descripción	rss	Respuesta de sistema	Flujo central
EC 1.1 Cambiar rss correctamente	Se introduce la dirección url del rss válido.	V http://humanos.uci.cu/feed/	Muestra el mensaje "Se cambió correctamente el RSS".	URL de acceso/link Configurar RSS

Capítulo 4: Implementación y prueba

EC 1.2 Cambiar rss incorrectamente	Se introduce la dirección url del rss inválido.	I http://novalido.uci.cu	Se muestra el mensaje: "Error cargando sus RSS de Noticias.Revice que este bien configurado"	URL de acceso/link Configurar RSS
EC 1.3 Cambiar rss con campos vacíos.	Se introduce la dirección con el campo vacío.	I Campo Vacío	Se muestra el mensaje: "El campos no puede estar vacío"	URL de acceso/link Configurar RSS

El resultado de las pruebas se resume en la siguiente gráfica de no conformidades.



4.4 Conclusiones Parciales

Es bueno mencionar la importancia que se le debe conceder a los estándares de codificación a utilizar, ya que facilita la comunicación con otros desarrolladores y aumenta la calidad de tu código. Muchos desarrolladores no le dan la relevancia requerida a las pruebas de software, dejándolo a un segundo plano, lo que provoca en varias ocasiones un producto de software con mala calidad. Es por tal motivo que se deben desarrollar buenas técnicas de casos de pruebas, mientras más complejas y abarcadoras sean, mayor validez tendrá el software implementado.

CONCLUSIONES GENERALES

Al culminar el proceso de investigación se concluye que:

- El futuro de la televisión depende de cuan interactivas sean las transmisiones y la cantidad de servicios que se brindan, de ahí la gran importancia que tienen los sistemas interactivos en la televisión actual. Los set-top-box jugarán un papel fundamental para lograr dichos objetivos ya que estos brindan todas las funcionalidades para que la interactividad a través del televisor sea una realidad.
- La solución propuesta es la primera de su tipo en el país y está orientada a funcionar en todos los set-top-box que tengan nivel 3 de interactividad gracias a su arquitectura orientada a *plugins*.
- Se desarrollo con tecnologías libres siguiendo las políticas de soberanía tecnológica que impulsa el país.
- Las herramientas utilizadas son multiplataforma lo que facilita cualquier tipo de migración.
- Las pruebas realizadas al software validan el correcto funcionamiento de la plataforma interactiva para su uso en el proyecto STCV. Por lo que se cumplen los objetivos trazados al comienzo de la investigación y se le da solución al problema propuesto.

RECOMENDACIONES

Luego de haber concluido el sistema propuesto para la investigación y con los objetivos de la misma vencidos se exponen las siguientes recomendaciones:

- Crear más aplicaciones interactivas para la plataforma, como pueden ser encuestas, grabar los canales en transmisión y juegos entre usuarios.
- Incrementar la cantidad de plugins para tener más alcance a mayor cantidad de set-top-box.

REFERENCIAS BIBLIOGRÁFICAS

afiliates, Oracle Corporation &. 2011. NetBeans. [En línea] 2011. [Citado el: 6 de 2 de 2013.]
[http://netbeans.org/community/releases/70/..](http://netbeans.org/community/releases/70/)

Alfonso, Alex. 2010. www.diigo.net. [En línea] 5 de 10 de 2010. [Citado el: 20 de 1 de 2013.]
<http://www.diigo.net/profile/sesser>.

Aurora Vizcaíno, Félix Oscar y Caballero García, Ismael. 2010. *Una Herramienta CASE para ADOO: Visual Paradigm*. 2010.

Burbeck, Steve. 2002. *Application programming in Smalltalk-80 : How to use Model-View-Controller (MVC)*. 2002.

Buschmann F, Meunier y R, Rohnert. 1996. *Patterns., Pattern-Oriented Software Architecture*. 1996.

César de la Torre Llorente, Unai Zorrilla Castro, Miguel Angel Ramos Barros, Javier Calvarro Nelson. 2010. *Guía de Arquitectura N-Capas orientada al Dominio*. 2010.

Fortan, Fernando. 2012. www.citel.oas.org. [En línea] 10 de 11 de 2012.
http://www.citel.oas.org/newsletter/2006/septiembre/iptv_i.asp.

Foundation, Apache Software. 2011. Visión general de las nuevas funcionalidades de Apache 2.0. [En línea] 2011. [Citado el: 5 de 2 de 2013.]
http://httpd.apache.org/docs/2.0/new_features_2_0.html..

Gamma, Erich. 2005. *Design Patterns: Elements of Reusable Object-Oriented Software*. 2005.

Gamma, Erich. Design Patterns. 2005. *Elements of Reusable Object-Oriented Software*. 2005.

García, Alberto García. 2010. *Principios de interactividad: televisión interactiva y realidad virtual*. 2010.

Giraldo, Luis y Zapata, Yuliana. 2005. *Herramientas de desarrollo de ingeniería de software para Linux*. 2005.

González, Francisco Ruiz. 2000. *Sistema Gestor de Bases de Datos*. 2000.

Greenfield Howard, Simpson Wes. 2007. *IPTV and Internet Video*. Burlington : Elsevier Inc., 2007.

Held, Gilbert. 2007. *Understanding IPTV*. New York : Taylor & Francis Group, 2007.

Herrera, Gerardo. 2006. *Festival Latinoamericano de instalación de software Libre: PHP*. 2006.

Hervé, Benoit. 2008. *Digital Television. Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework*. 2008.

- Inc, Object Management Group. 2003.** *Unified Modeling Language Specification*. Masachussets : s.n., 2003.
- International, Visual Paradigm. 2011.** Visual Paradigm. What VP-UML Provides. [En línea] 2011. [Citado el: 6 de 2 de 2013.] [http://www.visual-paradigm.com/product/vpuml/provides/..](http://www.visual-paradigm.com/product/vpuml/provides/)
- Jacobson, Booch. 2004.** *El Lenguaje Unificado de Modelado*. 2004.
- Jacobson, I., Booch, G. y Rumbaugh, J. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
- Jiménez, Alberto Cobo. 2009.** *Nuevas aplicaciones y servicios interactivos de TDT*. 2009.
- Keith, Jack. 2009.** *Video demystified a Handbook for the Digital Engineer*. 2009.
- Kruchten. 2000.** *The Rational Unified Process: An Introduction*. 2000.
- Laman, Craig. 2004.** es.scribd.com. [En línea] 2004. [Citado el: 6 de 2 de 2013.] <http://es.scribd.com/doc/36837598/Applying-UML-and-Patterns-3rd-Ed-Craig-Larman-2004..>
- Marin Arias, David Marvin. 2008.** Definición de lenguaje de programación. Tipos. Ejemplos. [En línea] 16 de 10 de 2008. [Citado el: 20 de 1 de 2013.] <http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos..>
- MeRinde. 2010.** merinde.net. [En línea] 2010. [Citado el: 7 de 2 de 2013.] merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
- Meyer, Jhonson. 2012.** Página oficial de MHP. [En línea] 20 de 1 de 2012. [Citado el: 11 de 11 de 2012.] <http://www.mhp.org>.
- PostgreSQL, Equipo de desarrollo de. 2010.** *Manual de Usuario de PostgreSQL*. 2010.
- Pressman, Roger. 2002.** *Ingeniería del Software: Un enfoque práctico*. Madrid : McGraw-Hill Companies, 2002.
- . **2005.** *Ingeniería del Software: Un enfoque práctico*. 2005.
- Quesada, Juan Antonio López. 2009.** *Pruebas del software*. 2009.
- . **2007.** *Pruebas del software*. 2007.
- Ramírez, David. 2008.** *IPTV Security*. Chichester. 2008.
- Rouse, Margaret. 2008.** SearchSoftwareQuality. [En línea] 20 de 5 de 2008. [Citado el: 25 de 10 de 2012.] <http://searchsoftwarequality.techtarget.com/definition/>.
- Rumbaugh, J, Jacobson, I y Booch, G. 2000.** *El Lenguaje Unificado de Modelado*. 2000.

Referencias bibliográficas

Stefanov, Stoyan. 2008. *Object-Oriented JavaScript. Create scalable, reusable high-quality JavaScript applications, and libraries.* 2008.

Tektronix. 2010. www.tektronix-resources.com. [En línea] 16 de 4 de 2010. [Citado el: 11 de 11 de 2012.] <http://www.tektronix-resources.com/0707MTM400/IPTVPrimer.pdf>..

Telecomunicaciones, Union Internacional de. 2005. *Curso a distancia de la UIT.* 2005.

Website, Hardz. 2007. Hardz Website. *PHP HiPeRTeXTo PRe-PRoCeSaDo. Blog de WordPress.com.* [En línea] 7 de 2 de 2007. [Citado el: 22 de 1 de 2013.] <http://hardz.wordpress.com/2008/02/07/php-hipertexto-pre-procesado/>.

BIBLIOGRÁFIAS CONSULTADAS

1. **applications. RTP: a transport protocol for real-time.** *www.isi.edu*. [En línea] Julio de 2003. [Citado el: 22 de Noviembre de 2012.] <ftp://ftp.isi.edu/in-notes/rfc3550.pdf>.
2. **digital, sub-comision tecnica de la tv.** *Informe Final v2*. La Habana : s.n., 2009.
3. **Furlock, Miguel.** *www.almas.tv*. [En línea] [Citado el: 21 de Noviembre de 2012.] <http://almas.tv/edicion3/todoenlacabeza/zapping.html>.
4. **IP, Convergencia hacia televisión sobre.** *www.bieec.epn.edu.ec*. [En línea] [Citado el: 20 de Noviembre de 2012.] <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/526/3/T10457CAP2.pdf>.
5. **IP, El protocolo.** *www.neo.lcc.uma.es*. [En línea] [Citado el: 22 de Noviembre de 2012.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ip.html>.
6. **Keith Jack .Video demystified a Handbook for the Digital Engineer.** Third Edition. Capítulo 14, Epígrafe 14.1.4.3, Página 650.
7. **Multicast., Introduction to.** *www.firewall.com*. [En línea] [Citado el: 21 de Noviembre de 2009.] <http://www.firewall.cx/multicast-intro.php>.
8. **Rives., Jose Francisco.** *www.um.es*. [En línea] [Citado el: 22 de Noviembre de 2012.] <http://www.um.es/atiga/gat/gat2/soluciones/que-son-los-codecs/>.
9. **Telecommunications., Outstanding Performance for.** *www.amd.com*. [En línea] [Citado el: 20 de Noviembre de 2012.] http://www.amd.com/us/Documents/43283-A_AMD64_and_Comarch_for_Telecom_White_Paper_PDF.pdf.
10. **Redes, Libro Blanco de Buenas Practicas Para el Despliegue de.** *www.sandetel.es*. [En línea] [Citado el: 21 de Noviembre de 2012.] www.sandetel.es/Libro_Blanco_Redess_Inalambricas_Municipales.pdf. ANEXOS

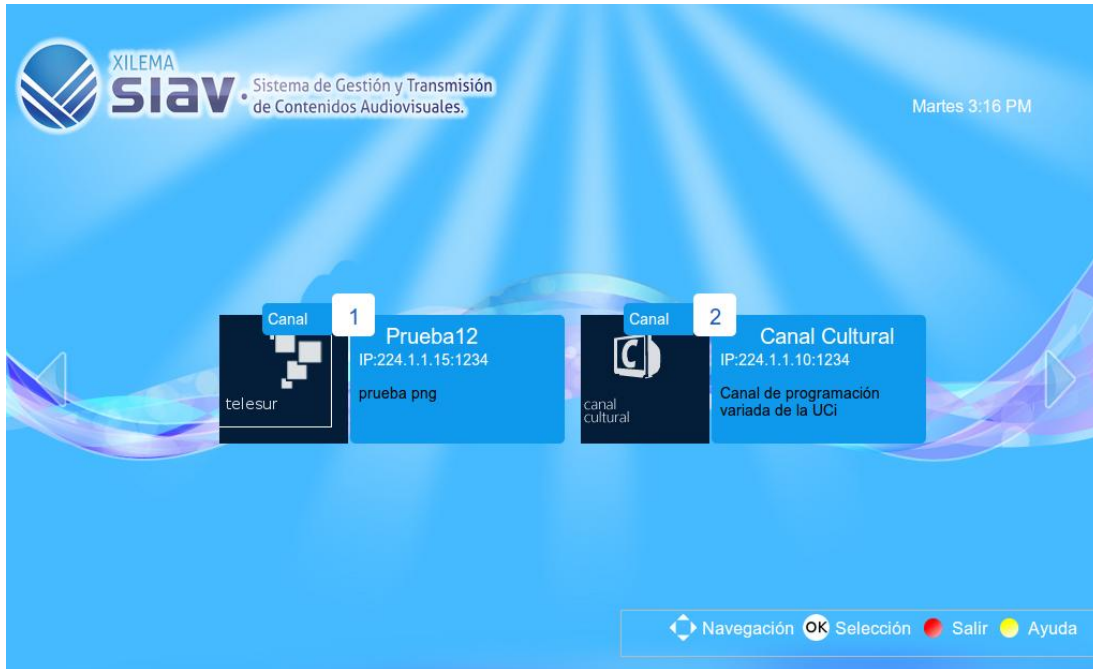


Ilustración 11 Interfaz Mostrar información de canales

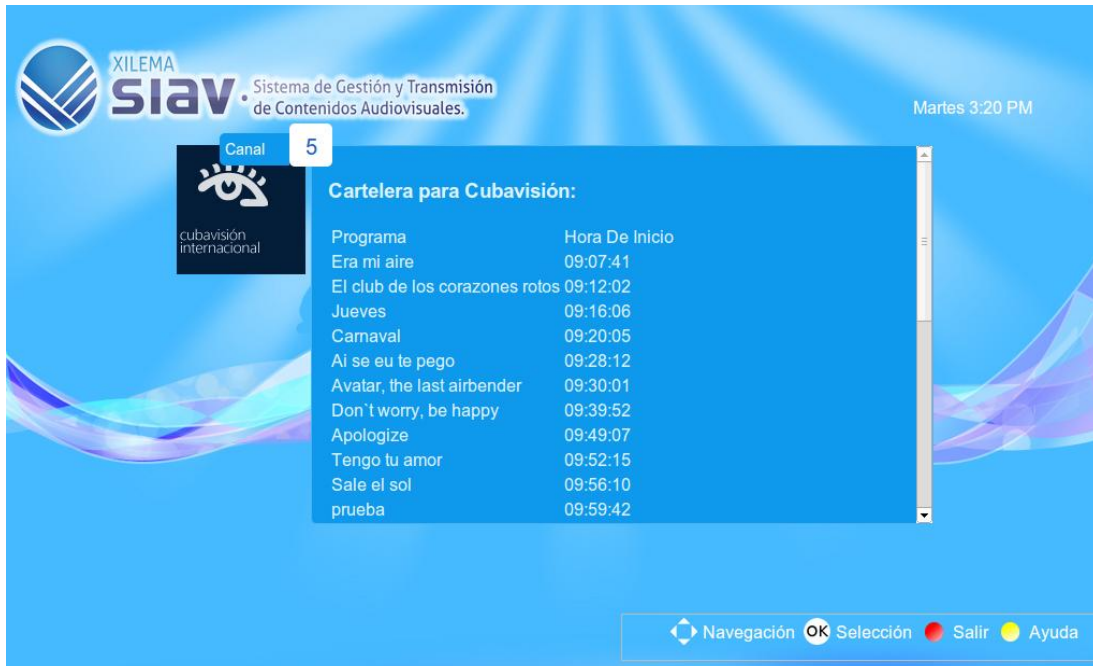


Ilustración 12 Interfaz Mostrar cartelera de un canal

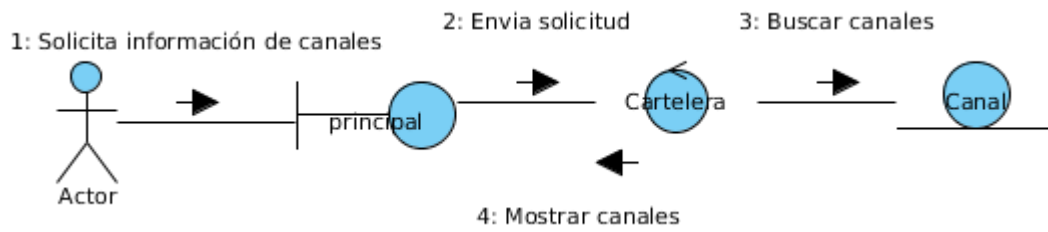


Ilustración 13 Diagrama de colaboración para el caso de uso Mostrar información de canales

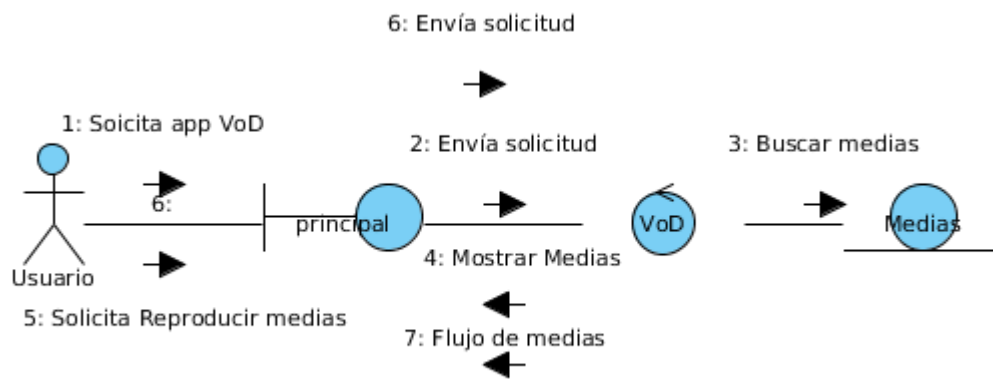


Ilustración 14 Diagrama de colaboración para el caso de uso Consumir video bajo demanda

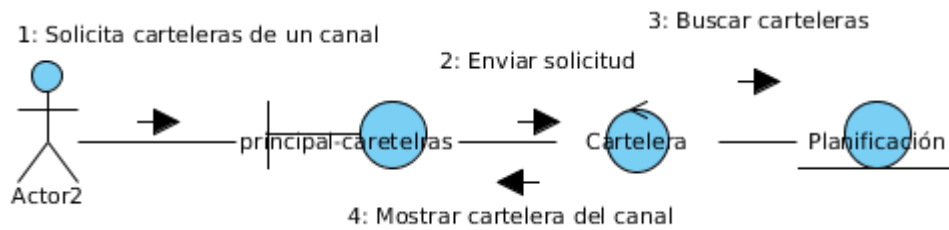


Ilustración 15 Diagrama de colaboración para el caso de uso Mostrar cartelera