

# **Framework para el desarrollo de aplicaciones IPTV en el ámbito académico (Universidad EAFIT)**

**Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz  
Camilo Vieira Mejía**

**Universidad EAFIT  
Pregrado de Ingeniería de Sistemas  
Medellín  
2008**

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

## TABLA DE CONTENIDO

1. Introducción .....	3
2. Definición del Problema .....	5
3. Objetivos.....	11
3.1 General.....	11
3.2 Específicos .....	11
4. Justificación .....	12
5. Marco de Referencia .....	13
5.1 Marco Teórico.....	13
▪ 5.1.1 IPTV.....	13
▪ 5.1.2 FRAMEWORK.....	23
6. Trabajos Relacionados .....	33
6.1 Frameworks e Investigaciones.....	33
6.2 Aplicaciones.....	33
7. Metodología .....	36
7.1 Modelado de negocio.....	38
7.2 Requisitos.....	38
7.3 Análisis y Diseño .....	39
7.4 Implementación .....	39
7.5 Pruebas.....	39
8. Framework para el desarrollo de aplicaciones IPTV en el entorno educativo (FIT)	41
8.1 Arquitectura. ....	41
8.2 Herramienta Multimedia.....	42
8.3 Protocolo de comunicación.....	45
8.4 Logger.....	47
8.5 Modelo de datos.....	47
8.6 Clases .....	49
▪ 8.6.1 Modelo .....	52
▪ 8.6.2 Control.....	55
▪ 8.6.3 Vista.....	57
▪ 8.6.4 Util.....	58
8.7 Código Fuente.....	61
9. Experimentos .....	64
9.1 Cliente y Servidor de Video bajo demanda .....	64
9.2 Cliente y Servidor de Guía electrónica de programación.....	66
9.3 Servidor de transmisión en vivo de canales.....	68
10. Resultados.....	70
11. Conclusiones.....	73
12. Trabajos futuros .....	74
13. Referencias.....	75
ANEXOS .....	78
Aleyda M. Hincapié Álvarez .....	78
René A. Rua.....	80
Raúl F. Ospina .....	81

## 1. Introducción

Actualmente estamos viviendo lo que muchos ya han llamado la era digital. Una continua evolución en todas las tecnologías y en especial en los medios utilizados para la comunicación se ve cada día, la aparición del Internet, quizá el mayor invento del siglo XX ha traído una gran cantidad de herramientas que han revolucionado entre otras cosas, las formas de educación para hacerla más accesible a todas las personas, y con recursos que antes ni siquiera se podrían imaginar.

La transmisión y reproducción de video a través de las redes ha sido una de las tecnologías que más lentamente ha evolucionado, debido a las grandes cantidades de ancho de banda requerido para un funcionamiento siquiera comparable con los actuales sistemas de televisión. Inicialmente en las redes de área local (LAN) y ahora que se está alcanzando buenos niveles de servicio incluso en Internet ha llegado la hora de explotar este servicio.

La televisión digital es un término que ha ido tomando fuerza en el medio tecnológico ya que, tratado de la manera correcta puede terminar desplazando por completo a los sistemas de televisión satelital y terrestre, que son actualmente los más utilizados en todo el mundo. IPTV es una tecnología de televisión montada sobre el protocolo de Internet, el cual es utilizado a diario para enviar y recibir millones de paquetes a través de muchas de las redes del mundo. Al estar montado sobre esto, se aprovechan todas las ventajas que esta globalizada red ofrece, como son: conexión a bajo costo entre diferentes lugares del mundo (teniendo en cuenta que un alto porcentaje de la población tiene acceso a este medio), alta calidad de la señal, comunicación bidireccional, etc.

La principal ventaja que ofrece IPTV sobre los servicios de televisión tradicional es la interactividad soportada producto de la comunicación

bidireccional, es decir, el usuario dejará de ser un simple espectador para convertirse en un actor activo sobre los programas de televisión, y esto puede ser explotado por el entorno educativo, el cual podría ver esto como una oportunidad para llegar incluso a la educación a distancia por medio de videoconferencias a las cuales los estudiantes tendrían acceso y la posibilidad de participar como si se estuviese de manera presencial.

Por otro lado, en el desarrollo de software ha ido creciendo una gran necesidad de la reutilización de código como premisa para realizar aplicaciones de mayor calidad en menos tiempo y con menor esfuerzo, motivo por el cual se desarrollan cada vez más “Marcos de Trabajo” (Frameworks). Un Framework se define como:

*“Estructura básica conceptual usada para resolver problemas complejos”[43]*

Ésta estructura de software será la que facilite el desarrollo de aplicaciones inmersas en el tópico de IPTV como lo son la transmisión de televisión, el video bajo demanda, las guías electrónicas de programación, etc. y que sean aplicables al entorno educativo, para así dejar una puerta abierta a la Universidad EAFIT a nuevas tendencias del lo que nosotros llamaremos “Educación Digital”.

Vale la pena aclarar que cuando nos referimos a que debe facilitar, esto debe ser medible en una reducción del tiempo de desarrollo necesario para realizar una aplicación sin variar el número de recursos utilizados en el proyecto ni el trabajo de las personas pertenecientes al mismo.

Así pues, en el presente documento se pretende realizar una descripción detallada del trabajo llevado a cabo para la realización de un Framework para aplicaciones para IPTV aplicables al entorno educativo, tanto la manera como fue desarrollado, como los experimentos realizados, los resultados obtenidos y posibles desarrollos futuros para el producto obtenido.

## 2. Definición del Problema

La televisión hoy en día es uno de los medios de difusión de información de mayor utilización alrededor del globo, debido a que permite involucrar tanto video como sonido, lo cual hace que la experiencia para el televidente sea mucho mas enriquecedora y de fácil comprensión. Gran parte de éste éxito que ha tenido la televisión también se debe a que se utiliza para muchos fines, que van desde el entretenimiento hasta la educación, pasando por el comercio, la información, etc.

Existen tres medios principales para la transmisión de la señal de televisión. El medio terrestre involucra señales de radio que se emiten desde un difusor principal, por medio de antenas y repetidores se esparce la señal a través de un área determinada. Cuando se habla de transmisión por cable, se refiere a que la señal se transmite por un medio físico (cable) entre los nodos, evitando así la necesidad de dispositivos encargados de captar la señal. Por último, en la televisión satelital, la señal es enviada a un satélite que se encarga de su distribución hacia antenas receptoras. Para que las señales viajen por estos medios de transmisión, deben ser codificados de alguna forma, y es por esto que existen dos tipos de difusión, análoga y digital. La análoga ha sido la forma de difusión usada por excelencia, pues la estructura de la transmisión de televisión fue concebida para hacerse a través de ondas de radio y microondas debido al bajo costo que esto implica, aunque esto puede traer graves deficiencias ya que estas ondas viajan por el aire y están expuestas a interferencia, lo cual deteriora su calidad. Además, por la manera misma en que fue estructurada la transmisión de señales análogas, no hay posibilidad de tener un canal de retorno y poder ofrecer servicios interactivos independientes de otros sistemas.

En los últimos años la televisión digital [3] ha ido surgiendo con mucha fuerza, pues a pesar de necesitar una infraestructura más compleja, brinda

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

los mismos servicios que la televisión análoga, introduce la posibilidad de interactividad utilizando el mismo canal de transmisión ya que permite flujo bilateral sobre el mismo medio, contrario a lo que vemos ahora (interactividad por un teléfono celular o por un computador, dispositivos totalmente independientes al aparato televisivo) y, además consigue una señal más tolerante a las interferencias, lo cual se traduce en transmisiones de mejor calidad. Si la televisión análoga ha sido un medio tan preferido en todo el mundo, ahora con la disponibilidad de la televisión digital habrá muchas más razones para que el televisor sea uno de los aparatos con mayor aceptación de la sociedad, pues las mismas aplicaciones que se tenían antes, como entretenimiento, educación, comercio, entre otras, serán mucho más atractivas debido a la interactividad que es permitida en la televisión digital. Por ejemplo, un programa televisivo de educación a través de televisión análoga sólo permite que la persona que dicte una conferencia hable, y su público se limite a escuchar.

Ahora imagínese esa misma conferencia pero a través de la televisión digital. Piense en que una de las personas que atiende decida preguntar algo, y el expositor pueda responder inmediatamente; esto plantearía la posibilidad de cursos a distancia sumamente interactivos. Los medios tradicionales para la transmisión de televisión digital han sido el satélite y el cable. A pesar de que la forma misma como está estructurada la transmisión de televisión digital abre la posibilidad de una televisión interactiva, conseguirlo con estos medios no es algo fácil. Por esta razón se ha venido desarrollando la difusión de televisión a través de Internet, pues el auge que posee esta tecnología en todo el mundo permitirá su fácil difusión entre las personas, y su modo de transmisión está orientado a la interactividad.

La infraestructura de las redes actuales se encuentra en una constante evolución, poco a poco en el tiempo se han ido desarrollando redes que no solo permiten la transmisión de datos y voz, sino de otros servicios tan importantes como la telefonía sobre IP o la transferencia de video. A éste innovador tipo de redes se les llama redes convergentes [6][7][9]. El manejar

todos estos servicios desde una misma red (y no diferentes para cada servicio) trae muchas ventajas, entre las que se puede encontrar la fácil administración de la red, el aprovechamiento al máximo del ancho de banda, entre otras.

Teniendo en cuenta todo lo anterior, podemos tomar televisión digital+redes convergentes+interactividad y nos encontraremos con una tecnología que poco a poco va tomando terreno en el ámbito televisivo, IPTV<sup>1</sup> (Internet Protocol Television)[1][2][4][5][8], la cual facilita la difusión del video basándose en tecnología video-streaming permitiendo que los usuarios reciban el contenido bajo demanda (VOD). Otros de los servicios que se pueden aprovechar de esta tecnología son acceso a Internet desde la televisión, servicios de comunicación electrónica (telefonía, mensajería instantánea, correo electrónico, etc.), personalización de la programación, juegos en línea multiusuario, participación activa sobre los programas (foros, votos, concursos, etc.) y servicios en línea (compras, apuestas, entre otras), etc. Todo esto se hace posible gracias a la infraestructura del protocolo de Internet (IP), el cual asigna una dirección a cada elemento de una red para su fácil identificación, y así poder transportar la información de un elemento a otro y viceversa sin tener que hacer un envío colectivo, es decir que cada mensaje no se enviará a todos los nodos de una red (broadcast), sino solo a los que lo deseen recibir (multicast).

Para la implementación de esta tecnología es necesario tener tanto una infraestructura lógica (Software) como física (Hardware) para el usuario y el proveedor del servicio. El primero tendría un dispositivo Set-Top Box<sup>2</sup> mientras el proveedor dispondría de un servidor encargado de la conversión de la señal análoga a digital para su distribución y atender todas las solicitudes de los clientes.

---

<sup>1</sup> “IPTV describe un sistema donde un servicio de televisión digital es entregado usando el Protocolo de Internet.” [4]

<sup>2</sup> Dispositivo encargado de convertir la señal digital a análoga para ser interpretada por el televisor y recibir todas las señales que el usuario desee enviar a través de un control remoto. [1] [2]

Adicional a los componentes de los equipos que transforman las señales entre digitales y análogas, los cuales son complejos y costosos, el hardware en general necesario para implementar esta tecnología está bastante desarrollado. Por el contrario, el desarrollo del software involucrado en la emisión, recepción e interactividad tiene un alto grado de complejidad debido a su reciente aparición y a la inmadurez existente en su proceso de desarrollo. Por esta razón se ha pensado en construir una herramienta que facilite la construcción de este tipo de software. Es por eso que se plantea desarrollar un marco de trabajo (framework [12]), el cual consta de un grupo de funciones y bibliotecas para la implementación de aplicaciones, enfocado en la construcción de aplicaciones para IPTV en el entorno educativo. El desarrollo de este framework brindaría la posibilidad de implementar aplicaciones para IPTV con menor esfuerzo al que se requiere actualmente, pues se liberaría al programador de ciertas tareas que se pueden resumir en funciones.

Todas las ventajas anteriormente mencionadas que ofrece IPTV, pueden ser aplicadas al ámbito educativo y más específicamente a la Universidad EAFIT. Cómo sería si todos los videos de la biblioteca Luís Echevarria Villegas, pudieran ser accedidos por los estudiantes sin tener que ir directamente a la Universidad, sino desde sus computadores e incluso desde sus televisores. De ésta forma no existirían inconvenientes tales como la cantidad insuficiente de videos y el robo o daño de los mismos.

Además, imaginemos un escenario en donde todos lo estudiantes pudieran asistir a sus clases remotamente, ya sea dentro del mismo campus universitario o desde su propia casa, sin tener que movilizarse, teniendo la misma interactividad con los docentes que si estuviesen presentes y, aún más, sin tener que cumplir un horario específico, sino con la posibilidad de grabar las clases y ver todas las inquietudes generadas durante el curso y todo esto, cuantas veces lo desee el estudiante. Todos estos beneficios se

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

podrían extender a las conferencias y actividades culturales que tienen lugar dentro de la Universidad.

Ahora, ante toda esta evidencia, se puede apreciar la necesidad de desarrollar un framework para comenzar a explorar esta área dentro de la Universidad, dejando así el camino abierto para que poco a poco se llegue a esta meta propuesta, “las clases virtuales”. El desarrollo de este framework proveería a la universidad de una base para futuros desarrollos en el área de modo que se vayan cubriendo todas las ventajas que ofrece esta innovadora tecnología.

Sería interesante comenzar por el desarrollo de dos servicios que entrarían a reforzar las estrategias que está implementando la Universidad de cara a consolidar una nueva cultura institucional que está encaminada hacia este tipo de educación virtual. La transmisión en vivo de conferencias, con las posibilidades de participación remota de estudiantes (pero dentro del campus), la visualización de más de una actividad en simultánea, y la grabación de las mismas para posterior visualización, junto con la digitalización de los videos existentes en la biblioteca para ofrecer el servicio de video bajo demanda son los dos servicios propuestas a desarrollar con este framework.

El primer servicio (transmisión de conferencias en vivo) permitiría además la masificación de las conferencias, al no limitarse por el espacio físico designado y, además, garantiza la interactividad de los usuarios con el orador, ya sea que estén físicamente presentes o que se encuentren en una Terminal remota dentro de la Intranet de EAFIT. También se hace factible que los usuarios que tengan otra actividad en ese horario, no se pierdan de la conferencia y puedan grabarla automáticamente por la aplicación. En este último caso no podría haber interactividad, pero por lo menos podría ver todo lo ocurrido durante la conferencia real. Esto les facilitaría a los estudiantes asistir a un mayor número de conferencias y actividades que les sean de

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

interés, mejorando a su vez la preparación que les pueda brindar la Universidad mediante estas actividades extracurriculares.

Ahora, ante la gran oportunidad que se abre para la universidad de ser pionera a nivel departamental y, por las anteriores razones nombradas, se ve la necesidad de utilizar la tecnología IPTV para el ámbito universitario y así los autores de este documento se hacen la pregunta:

¿Cómo elaborar un framework para el desarrollo de servicios para la tecnología IPTV?

Estas aplicaciones deberían permitir convertir la señal de televisión análoga a digital y difundirla vía unicast y multicast por la intranet universitaria, optimizando así el uso del ancho de banda; además, se deben construir las aplicaciones cliente que permita la interactividad de los usuarios finales.

## 3. Objetivos

### 3.1 General

Desarrollar un framework para el desarrollo de aplicaciones para el servicio IPTV en la Universidad EAFIT con el propósito de ofrecer el servicio de transmisión de video (video streaming) y video a demanda (video On-Demand).

### 3.2 Específicos

- Diseñar un framework para el desarrollo de aplicaciones de IPTV utilizando el lenguaje de modelado unificado (UML) y las técnicas de análisis y diseño especificadas en proceso unificado de racional (RUP).[10]
- Implementar un framework para el desarrollo de aplicaciones de IPTV en el lenguaje de programación Java [11], reutilizando las bibliotecas de funciones disponibles en el medio que puedan apoyar la implementación.
- Desarrollar un prototipo de aplicación, que preste los servicios de video streaming y video bajo demanda para su uso en el ámbito universitario, utilizando el framework desarrollado.
- Realizar pruebas del prototipo de aplicaciones en un ambiente controlado para poder comprobar el correcto funcionamiento de los servicios a ofrecer.

## 4. Justificación

El desarrollo de un framework que sirva como base para la implementación de servicios IPTV trae grandes beneficios para la Universidad EAFIT, ya que abre las puertas a futuras ideas e investigaciones en este novedoso campo. Como se propuso en la *Definición del Problema*, la tecnología IPTV puede ofrecer muchas oportunidades y ventajas al ser enfocada hacia un contexto educativo. Aplicaciones como la educación a distancia, la disponibilidad 7/24 de videos de interés educativo al público en general y sin límites de ejemplares ni usuarios, son sólo unos de los tantos servicios que se podrían ofrecer por medio de esta tecnología, mejorando la experiencia de los estudiantes universitarios en la institución y brindando un valor agregado a la misma.

La reutilización de código ha sido catalogada como una de las mejores prácticas en el desarrollo de software en la actualidad, representada principalmente por la construcción y utilización de funciones y librerías que eliminan la necesidad de implementar funciones básicas y genéricas, y así poder enfocarse en los desarrollos particulares. De allí la importancia de la construcción de un framework, pues este integra estas herramientas, en el caso específico, para el desarrollo de aplicaciones IPTV.

## 5. Marco de Referencia

### 5.1 Marco Teórico

#### ▪ 5.1.1 IPTV

Internet Protocol Television (IPTV) es una tecnología que transmite televisión programada y video bajo demanda (VOD) utilizando la técnica de Streaming sobre el protocolo de Internet (IP) lo que permite la comunicación bidireccional entre el proveedor de televisión y todos sus clientes, ofreciendo una nueva generación de televisión con interactividad sobre el mismo canal.

Esta es la definición de la tecnología sobre la cual se basan los conceptos de ésta investigación. Se trata de transmisión de señales de televisión por redes que utilizan IP como su protocolo de cabecera, es decir, las mismas redes por las que llega el Internet a las casas y empresas permitiendo entonces contar con todas las bondades de que disponen, así como también con todos sus problemas. Pero para entender mejor esto, se hace necesario saber que es IP.

IP (Internet Protocol) es la base fundamental del Internet. Se encarga de llevar datagramas<sup>3</sup> de la fuente al destino. Durante su transmisión se puede partir un datagrama en fragmentos que se montan de nuevo en el destino. Las principales características de este protocolo son:

- Es un protocolo no orientado a la conexión, es decir que no se preocupa en saber si el mensaje que transmitía llegó a su destino o no.
- Fragmenta paquetes si es necesario de acuerdo a las aplicaciones que este teniendo la red o el software en el momento.

---

<sup>3</sup> El datagrama IP es la unidad de transferencia en las redes IP. Básicamente consiste en una cabecera IP y un campo de datos para protocolos superiores.

- Direccionamiento mediante direcciones lógicas IP de 32 bits, aunque en la actualidad se está hablando ya de IP versión 6, en las cuales se cuentan con direcciones de 128 bits.
- Si un paquete no es recibido, este permanecerá en la red durante un tiempo finito para evitar la sobrecarga de la misma.
- Realiza el "mejor esfuerzo" para la distribución de paquetes utilizando algoritmos de optimización y enrutamiento.
- Sólo se realiza verificación de consistencia de datos al encabezado del paquete, no a los datos éste que contiene.

La principal ventaja con la que cuenta este protocolo para la transmisión de televisión, es la posibilidad de tener un canal que puede enviar información hacia ambos sentidos, permitiendo la transferencia de datos desde el cliente (la persona que se encuentra viendo un programa sentada en su sofá, desde el control remoto) hacia el servidor (El proveedor del servicio de TV), agregándole interactividad distinta a la que normalmente se obtiene con los sistemas de televisión, donde la comunicación se debe realizar por un canal alternativo al de la transmisión, como el teléfono o el Internet.

La tecnología IPTV, ha sido desarrollada basándose en el "streaming" de video. Este término se utiliza para nombrar la transmisión de un archivo de multimedia, a través de una red, sin necesidad de descargarlo completamente antes de reproducirlo. Para poder proporcionar un acceso claro, continuo y sin interrupciones, el streaming se apoya en las siguientes tecnologías:

### **Codecs**

La palabra codec se traduce de las palabras codificador y decodificador. Un codec no es ni más ni menos que una serie de funciones algorítmicas necesarias para comprimir un archivo, a este proceso de compresión se le denomina "codificación" y descomprimir o decodificar los datos de audio y vídeo. Los codecs pueden codificar el flujo o la señal (a menudo para la transmisión, el almacenaje o el cifrado) y recuperarlo o descifrarlo del mismo

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

modo para la reproducción o la manipulación en un formato más apropiado para estas operaciones. Muchos archivos multimedia contienen tanto datos de audio como de vídeo, y a menudo alguna referencia que permite la sincronización del audio y el vídeo.

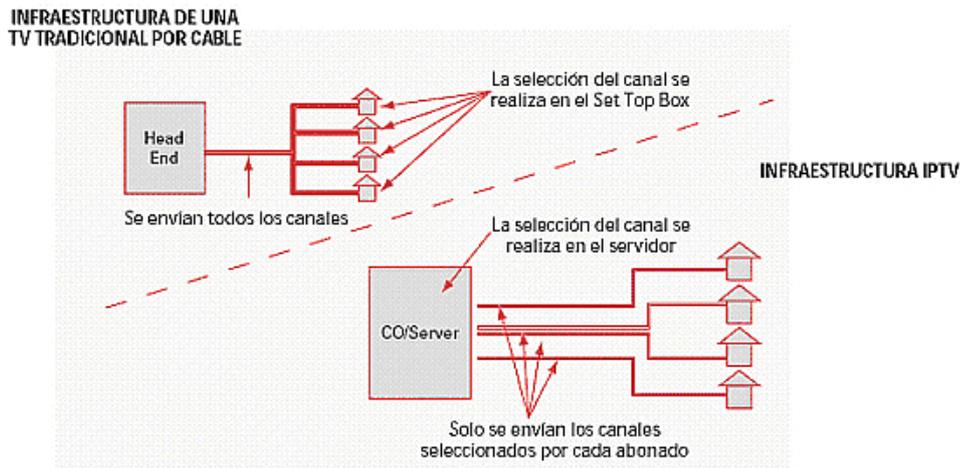
### **Protocolos Ligeros:**

UDP y RTSP (los protocolos empleados por algunas tecnologías de "streaming") hacen que las entregas de paquetes de datos desde el servidor a quien ve la página se hagan con una velocidad mucho mayor que la que se obtiene por TCP y HTTP. Esta eficiencia es alcanzada por una modalidad que favorece el flujo continuo de paquetes de datos. Cuando TCP y HTTP sufren un error de transmisión, siguen intentando transmitir los paquetes de datos perdidos hasta conseguir una confirmación de que la información llegó en su totalidad. Sin embargo, UDP continúa mandando los datos sin tomar en cuenta interrupciones, ya que en una aplicación multimedia estas pérdidas son casi imperceptibles. Se debe tener en cuenta que la utilización de estos protocolos no supone el reemplazo de IP, sino que son protocolos de otra capa en el modelo OSI y son encapsulados por IP para su transmisión en la red.

Esta tecnología transformará en un futuro próximo la televisión actual, aunque para ello son necesarias unas redes mucho más rápidas que las actuales, para poder garantizar la calidad en el servicio.

Entre otras ventajas, un sistema de IPTV prácticamente podrá individualizar a cada suscriptor, dar seguimiento a sus consumos y hacer mediciones de *rating* prácticamente perfectas. La personalización del contenido se convertirá en el próximo estándar siguiendo con el esquema de Web 2.0. A diferencia de la televisión por satélite, televisión por cable, o la televisión digital terrestre (TDT), el proveedor no emitirá sus contenidos continuamente para ser seleccionados por el usuario de forma local una vez se conecte, sino que los contenidos llegarán sólo cuando el cliente los solicite explícitamente, y aquí es donde entra el nuevo enfoque de infraestructura que diferencia a IPTV de la televisión tradicional (Figura 5.1). La clave está

en la personalización del contenido para cada cliente de manera que el usuario podrá seleccionar los contenidos que desea ver y de esta manera poder visualizarlos tantas veces como desee. Se trata en definitiva de un servicio que hace posible una televisión o un cine 'a la carta' en el que cada usuario puede ver el programa o película que desea y en el momento que desea.



**Figura 5.1 Infraestructura de una televisión tradicional vs. IPTV**

**Fuente:** <http://www.cinit.org.mx>

Entre las principales ventajas que tiene la televisión IP y la tradicional se destacan las siguientes:

- VIDEO ON DEMAND (Video bajo demanda): Se define como vídeo a la carta. Este sistema permite al usuario solicitar y disfrutar una película o un programa concreto en el momento exacto que el espectador lo desee. Por lo tanto se considera como la principal ventaja, ya que proporciona una nueva forma para disfrutar de los ratos de ocio donde el cliente conectado a un servidor de Video On Demand tiene un sistema alternativo a las tiendas tradicionales de alquiler de películas. Con este nuevo sistema los usuarios también pueden elegir sus productos desde un catálogo de títulos donde podrán seleccionar el archivo multimedia de su interés, descargarlo y visualizarlo, o bien en la pantalla del ordenador, o bien en una televisión digital o analógica (por medio de un Set Top Box).

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

- **MAYOR CONTENIDO:** IPTV puede ofrecer los mismos canales, eventos deportivos, etc. que las convencionales televisión por cable, satélite o TDT, pero además dentro de su oferta de contenidos puede contar con un almacén de películas, documentales, conferencias y programas de televisión que pueden ser vistas por los usuarios durante un tiempo mayor que las películas o eventos en emisión de una TV digital convencional, en resumen una gigantesca biblioteca de contenido de video puesta al servicio de los clientes.
- **PUBLICIDAD PERSONALIZADA:** Debido a que se trata de un canal bidireccional, los usuarios podrán determinar y seleccionar cuales son las áreas de interés sobre las que les gustaría recibir ofertas de publicidad, siendo así mucho más efectiva. Además, puesto que esta misma bidireccionalidad permite la interacción del receptor con el servidor, se podrían crear nuevos modelos de publicidad que hasta ahora estaban destinados únicamente para Internet, contando con poderosas herramientas como las encuestas, la publicidad orientada a los gustos, etc.
- **SERVICIOS DE VALOR AÑADIDO:** El usuario se encontraría sentado frente al televisor como si se tratase de la pantalla del ordenador, por lo que igualmente podría tener acceso a todo tipo de información, se podría tener acceso no solo a contenidos televisivos sino también por ejemplo a contenidos de e-learning, buscadores, e-mail, y todo lo que puede ofrecer el Internet.
- **VIDEOTELEFONO:** La integración de voz y televisión permitirá a las personas instalar una cámara sobre su televisor para comunicarse con otros usuarios mediante la red de banda ancha. Esto evolucionaría además a la videoconferencia entre dos o más usuarios. Esto permitiría desarrollar aplicaciones relacionadas con la videoconferencia, como la educación en línea, que haría que varios alumnos se comuniquen con su profesor a través del televisor.

En cuanto a tecnología, una característica común a todas las soluciones de

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

IPTV es que el transporte de la información (audio, video, datos asociados) se realiza mediante los protocolos IP sobre redes de banda ancha. El encapsulado de los contenidos de televisión se realiza sobre RTP y UDP, utilizando tecnologías tecnologías multicast para su distribución. Estas tecnologías tienen la ventaja de que cada canal de TV se transmite una sola vez, y se replica en todos los puntos de la red donde se bifurcan los caminos para alcanzar a usuarios diferentes que han sintonizado a ese canal (en terminología multicast, están 'suscritos' a ese canal).

En el caso de los contenidos en vídeo bajo demanda, y dado que el usuario tiene control completo sobre la reproducción del programa, no es posible compartir el mismo flujo entre múltiples usuarios, por lo que se utilizan las técnicas habituales unicast.

En la Figura 5.2 se puede observar la arquitectura básica necesaria para el servicio de IPTV:

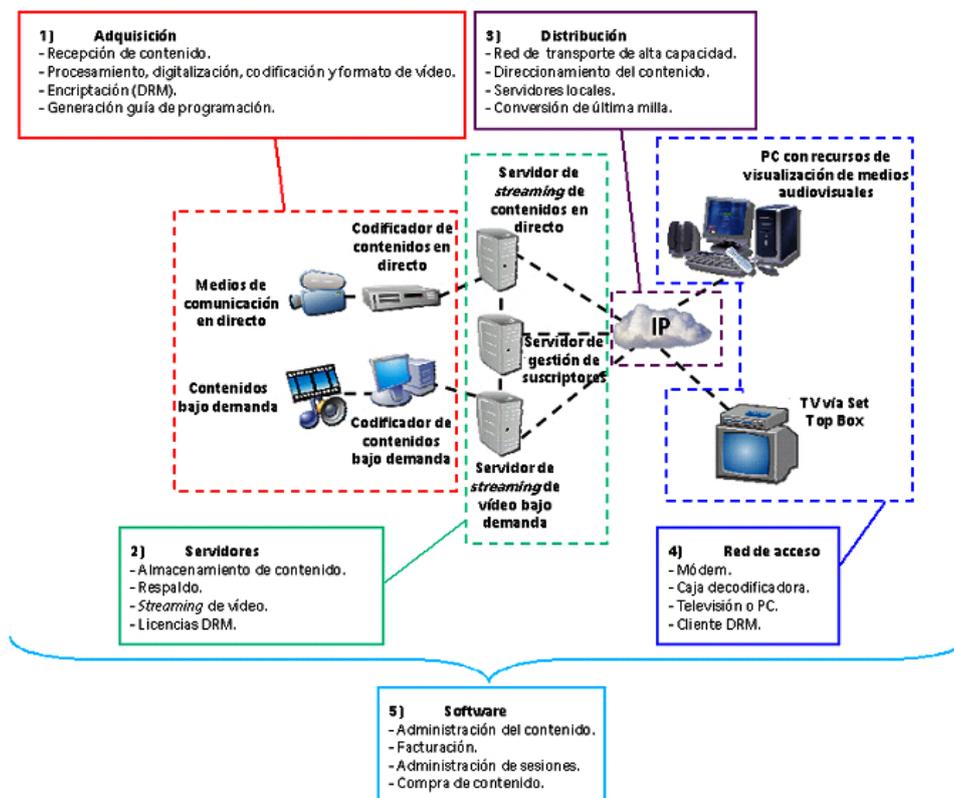


Figura 5.2 Arquitectura IPTV

Fuente: <http://www.cinit.org.mx>

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

Se divide en cinco componentes que son:

- Adquisición: Captura de la señal del video en vivo o almacenado como contenido para Video Bajo Demanda.
- Servidores: Encargados de obtener la señal desde los dispositivos de adquisición y recibir las solicitudes de los clientes para realizar esta conexión entre los dos actores,
- Distribución: Es la red que trabaja sobre protocolo de Internet como tal, es el medio sobre el cual se transportan los datos.
- Red de Acceso: Dispositivos que permiten al usuario interactuar con todo el contenido IPTV.
- Software: Componente encargado de la administración del sistema IPTV.

#### Servicios IPTV actuales

IPTV cada vez más se ha ido masificando, en España, a finales de 2005 Jazztel empezó a prestar este tipo de servicios con el nombre Jazztelia TV. Joost, creada por los fundadores de Kazaa y Skype, es una de las aplicaciones que más promete posesionarse en el mercado para video bajo demanda y transmisión de televisión en línea. Otras propuestas son la PPLive (en china), Orange (Orange TV) y Superbanda, France Telecom con su servicio MaLigne TV lanzado en 2003 y con casi 400.000 usuarios en la actualidad, y FastWeb en Italia con 350.000 usuarios, Deutsche Telecom, Telecom Italia, etc.

Para la transmisión de datos a través de una red, se utilizan varios protocolos. En especial mencionaremos dos protocolos que son utilizados especialmente para la transmisión de contenido multimedia, los cuales son RTP y RTSP.

#### Real-Time Transport Protocol (RTP)

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

Es el protocolo estándar de Internet para el transporte de datos en tiempo real, incluyendo audio y video. RTP puede ser usado para el acceso de datos multimedia bajo demanda, así como servicios interactivos como por ejemplo telefonía por Internet (VoIP). Este protocolo está compuesto por dos partes, una de datos y una de control. Esta última se conoce como Real-Time Control Protocol (RTCP).

La parte de datos del protocolo RTP es un pequeño protocolo que brinda soporte para aplicaciones con propiedades de tiempo real, como por ejemplo contenido multimedia continuo (audio o video), incluyendo reconstrucción sincronizada, detección de pérdida de información, seguridad e identificación de contenido.

RTCP, es decir, la parte de control del protocolo RTP, está basado en la transmisión periódica de paquetes de control a todos los participantes de una sesión usando el mismo mecanismo de distribución que los paquetes de datos.

Real Time Streaming Protocol (RTSP).

El protocolo de flujo de datos en tiempo real (del inglés Real Time Streaming Protocol) establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de video. Las funciones del RTSP son una especie de control remoto a través de servidores multimedia, pues tiene peticiones como reproducir (play), pausa (pause), adelantado rápido (fast forward), entre otras.

El protocolo RTSP es usado para controlar múltiples sesiones de entrega de datos, proveer un medio para seleccionar canales de entrega como UDP, TCP y UDP multicast, y brindar un medio para seleccionar mecanismos de entrega basados en RTP. El RTSP es un protocolo no orientado a conexión, en lugar de esto el servidor mantiene una sesión asociada a un identificador, en la mayoría de los casos RTSP usa TCP para datos de control del

reproductor y UDP para los datos de audio y vídeo aunque también puede usar TCP en caso de que sea necesario. En el transcurso de una sesión RTSP, un cliente puede abrir y cerrar varias conexiones de transporte hacia el servidor para satisfacer las necesidades del protocolo.

El protocolo RTSP está teniendo gran acogida en la distribución de contenido multimedia para la nueva generación de televisión digital, especialmente porque cuenta con las siguientes propiedades: es extensible, seguro, independiente del protocolo de transporte, tiene capacidad multi-servidor, controla dispositivos de grabación y es adecuado para aplicaciones profesionales. A continuación se explican brevemente estas propiedades:

*Extensible:* Se puede actualizar fácilmente, es decir, brinda facilidad para adicionar nuevos métodos y parámetros.

*Seguro:* Al estar tan relacionado al HTTP, RTSP reutiliza mecanismos de seguridad Web ya sea a los protocolos de transporte (TLS) o dentro del mismo protocolo. Se puede usar tanto la autenticación HTTP básica como la basada en resumen.

*Independiente del protocolo de transporte:* RTSP puede usar indistintamente protocolos de datagrama no fiables (UDP) o datagramas fiables (RDP, no muy extendido) o un protocolo fiable orientado a conexión como el TCP.

*Capacidad multi-servidor:* Debido a la necesidad de soportar aplicaciones distribuidas, cada flujo multimedia dentro de una presentación puede residir en servidores diferentes, y el cliente automáticamente establece varias sesiones concurrentes de control con los diferentes servidores. La sincronización en el flujo de los datos la realiza la capa de transporte.

*Control de dispositivos de grabación:* El protocolo puede controlar dispositivos de grabación y reproducción (como micrófonos, cámaras, etc).

## VLC

VIDEOLAN es un proyecto desarrollado inicialmente por estudiantes de

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

École Centrale Paris para la transmisión de video con formato MPEG en redes con alta capacidad de ancho de banda. Posteriormente se convirtió en un proyecto de código abierto con licencia GNU y ha sido mejorado por más de veinte personas de todo el mundo.

VIDEOLAN consta de dos partes:

- Servidor (VLS): Permite transmitir MPEG-1, MPEG-2 y MPEG-4, DVD, y video sobre la red Multicast o Unicast.
- Cliente (VLC – inicialmente solo cliente): Ahora puede hacer las mismas cosas del servidor y adicionalmente recibir el video, decodificarlo y mostrarlo.

VLC está desarrollado en el lenguaje de programación C, conocido por su alto rendimiento debido a la manera que fue diseñado y a que se ejecuta directamente contra el sistema operativo.

## **JMF**

Java Media Framework es un framework desarrollado por Sun Microsystems para la manipulación (transmisión y reproducción) de archivos multimedia desde aplicaciones Java.

Al estar desarrollado sobre una máquina virtual (JVM) posee gran portabilidad y estabilidad entre diferentes sistemas operativos, además está desarrollado en un esquema de fácil uso y posee una buena cantidad de documentación en la red.

Permite la reproducción de muchos de los formatos de video y audio más utilizados como WAV, MIDI, MP3 (en Audio) y MPEG, AVI y MOV (en Video).

### ▪ 5.1.2 FRAMEWORK

El concepto de *framework* es definido como una “estructura básica conceptual usada para resolver problemas complejos”[43]. Con la creciente necesidad de crear estándares para mejorar la calidad del software y mejorar la eficiencia de su desarrollo, el término framework ha tomado especial importancia en el ambiente de la computación y se ha definido como una “estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado”. Como el interés de ésta tesis es el software, a partir de aquí cuando se utilice la palabra *framework* se estará haciendo referencia a un *framework para el desarrollo de software*.

La finalidad principal con la que se crean los frameworks es para conseguir que la producción de software (tanto el diseño como el desarrollo) sea más fácil y confiable (libre de errores), pues estos liberan a los diseñadores y a los programadores de muchas tareas de bajo nivel que se vuelven tediosas y propensas al error debido a su complejidad, y permiten que estos dediquen sus esfuerzos en satisfacer los requisitos de software para producir un sistema funcional.

Las bibliotecas de funciones son un componente fundamental de los frameworks, y su robustez, facilidad de uso y flexibilidad marcan la diferencia entre un framework exitoso y uno que nunca será usado para desarrollar ninguna aplicación. Estas bibliotecas de funciones agrupan funcionalidades similares que requieren un alto esfuerzo de programar. Por tanto es evidente que usar funciones de alto nivel que hagan “transparente” las implementaciones de código de más bajo nivel es una ayuda en el desarrollo de software, tanto en el aumento de la eficiencia como en la disminución del riesgo de error.

Como se dijo previamente un framework debe ser una estructura para el desarrollo de un software determinado, por lo tanto, debe limitar las opciones y mostrar una guía clara de cómo resolver cierto problema. Esto puede convertirse en un arma de doble filo, pues evidentemente beneficia la

productividad, sobretodo en el desarrollo de sistemas complejos, pero a su vez puede hacer que las aplicaciones que se puedan desarrollar usando el framework sean muy limitadas. Por tanto, como también mencionamos antes, la flexibilidad del framework es fundamental, esto es, que aunque hay ciertas metodologías para usar el framework, los desarrolladores deben poder agregar nuevas funcionalidades que extiendan y complementan al framework y que le den algunas libertades al programador. Como dice Pree (1994), los frameworks están compuestos de puntos congelados y puntos calientes. Los puntos congelados son partes invariantes que no se pueden modificar, es decir los componentes principales que definen la arquitectura y la relación entre ellos. Y los puntos calientes son partes donde el programador puede agregar su propio código para implementar funcionalidades nuevas. La figura 3, muestra la ventaja que presenta un framework al permitir construir diferentes tipos de aplicaciones a partir de un código específico, reutilizando los componentes genéricos de un negocio inmersos en el centro del framework (Kernel), conectándose a través de los puntos calientes (Hot-Spot),

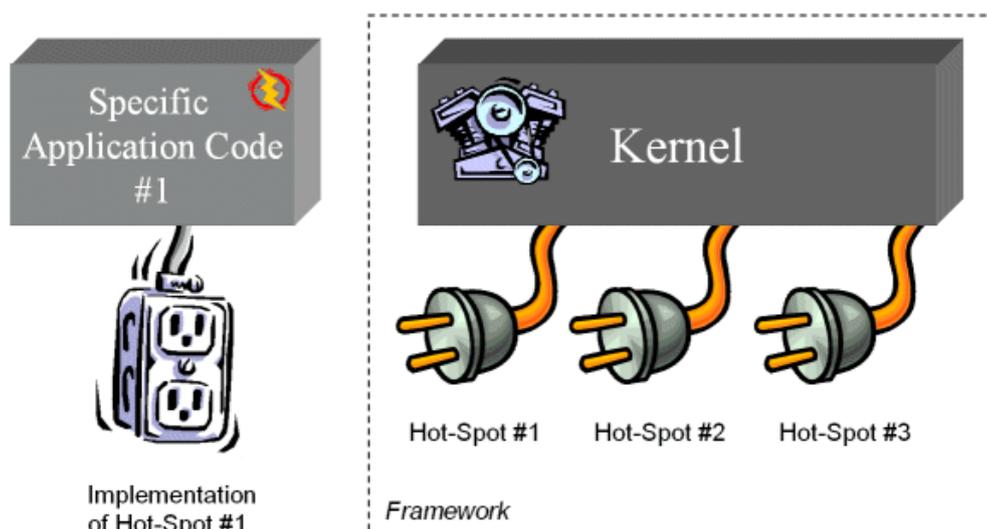


Figura 5.3 Framework

Fuente: <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>

Como aquí el objetivo es construir un framework que permita el desarrollo de aplicaciones para IPTV, especialmente para el ámbito académico, se deben

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

identificar las tecnologías que soportan IPTV y desarrollar herramientas que faciliten el uso de estas tecnologías para construir aplicaciones IPTV este entorno. De esta forma los estudiantes tendrán la posibilidad de desarrollar software para la Universidad usando el framework sin preocuparse de problemas tediosos de la tecnología, sino dejando volar su imaginación y produciendo soluciones más rápidas, útiles y libres de errores.

Uno de los framework relacionados con aplicaciones multimedia es Java Media Framework (JMF), construido por Sun Microsystems. JMF permite desarrollar aplicaciones de diversos tipos que tienen que ver con contenido multimedia, es decir, no se restringe a un ámbito determinado, sino que puede ser usado tanto en aplicaciones móviles, de escritorio, Web, sector educativo, empresarial, etc.

La construcción de un framework engloba muchas características a tener en cuenta. Un framework debe ser escalable, modificable, auditable, portable y mantenible, y debe tener alta cohesión<sup>4</sup> y bajo acoplamiento<sup>5</sup>. Algunos de estos aspectos deben ser desarrollados explícitamente por código y, para esto se propone la utilización de Patrones de Diseño<sup>6</sup>.

*¿Por qué basarse en patrones de diseño para este desarrollo?*

El diseño adecuado de software no es algo fácil de lograr y menos aún cuando se requiere diseñar algo que sea totalmente reutilizable, modificable, mantenible, etc. Se deben encontrar las relaciones específicas para cubrir el dominio que se desea (en este caso IPTV) y establecerlas estrechamente, pero dejando las puertas abiertas a cambios que no impacten directamente el negocio. El modo de almacenar datos (archivos planos, archivos xml,

---

4 "Cuando decimos que una clase tiene una alta cohesión queremos decir que el objeto tiene bien delimitadas sus responsabilidades" [27]

5 "El acoplamiento se puede definir como el número de relaciones que tiene una clase con las demás. Mientras más alto sea el acoplamiento, más difícil será independizarla para reutilizarla." [27]

6 "Design pattern is a widely accepted solution to a recurring design problem in OOP" [28]

bases de datos, etc.), la forma como se implementen ciertas funcionalidades, la configuración de una aplicación, todos estos son aspectos que se deben tener en cuenta, pueden cambiar y no debería impactar altamente toda la aplicación.

Para lograr este bajo acoplamiento en el código de una aplicación no es necesario desarrollar todo desde cero, sino por el contrario lo ideal sería poder utilizar elementos que ya hayan sido probados exitosamente en muchos proyectos y que faciliten el objetivo que acá nos ocupa.

El objetivo de un framework como el que se quiere desarrollar es el de no tener que “reinventar la rueda” cada vez que se quiera hacer un desarrollo para el ámbito universitario sobre el tema de la Televisión sobre Protocolo de Internet (IPTV), así que si se quiere facilitar la vida de los desarrolladores, ¿Por qué habría que complicarse empezando todo desde cero? Muchos patrones y frameworks ya han sido desarrollados y probados, lo cual nos da una base sobre la que nos podríamos apoyar para nuestro proyecto. Se propone la utilización de patrones como Modelo Vista Controlador (MVC), Objeto de Acceso a Datos (DAO) ó Fachada (Facade) que serán explicados más adelante en este documento y que proveen toda la flexibilidad y confiabilidad que se necesita a la hora de desarrollar un framework.

Para concluir, existen dos razones básicas que sustentan la utilización de patrones. La primera es que los patrones más conocidos ya han sido probados y han sido utilizados en numerosos desarrollos exitosos.

La segunda razón es que son reutilizables, basta solo con adaptarlos al dominio del desarrollo propio para poderlos aprovechar exitosamente.

Algunas de las ventajas que nos pueden ofrecer la implementación de patrones de diseño como los mencionados anteriormente son:

- Modificabilidad
- Mantenibilidad
- Escalabilidad

## MODELO VISTA CONTROLADOR (MVC)

El patrón Modelo Vista Controlador (MVC) consiste en separar la construcción de un artefacto de software en tres partes:

- El modelo:
  - Representa las reglas de negocio y el acceso a los datos.
- La vista:
  - Representa la interfaz al usuario final, ya sea una página web, un formulario, una interfaz de comunicación con un web service, o simplemente una clase por medio de la cual invoque al resto de la aplicación.
- El controlador:
  - Encargado de realizar las comunicaciones entre la vista y el modelo, controla el flujo de eventos, es decir, si un usuario realizó una solicitud por medio de la vista, el controlador la transmitirá al modelo y, si este envía una respuesta, el mismo controlador será el encargado de reflejarla por medio de la vista al usuario.



Diagrama MVC del Akelos Framework

Figura 5.4 Patrón MVC

Fuente: <http://code.google.com/p/akelosframework/wiki/TutorialSpanish>

El patrón surge bajo la necesidad de tener bajo acoplamiento de las partes en un desarrollo. Hasta hace algún tiempo toda la lógica del negocio venía “quemada” en la misma interfaz de usuario, entonces si por ejemplo se quería que diferentes usuarios accedieran por medios diferentes a una aplicación, uno por una aplicación winforms, otro por un Webservice otro por una aplicación Web, etc. se debería hacer tantos desarrollos iguales (en cuanto a lógica) como interfaces de usuario se quisieran tener. Con la utilización de este patrón, solo se tendría que desarrollar una nueva interfaz (vista) que se comunicara con el modelo y el controlador previamente creado.

### OBJETO DE ACCESO A DATOS (DAO)

La mayoría de las aplicaciones necesitan preservar los datos con los que se están tratando en el tiempo, y este framework no es la excepción. Si se desea un framework auditable, o se desean guardar datos (como videos, imágenes, etc.) se deberá pensar en un diseño que incluya el acceso a los datos. Además, si se piensa en las capacidades de ser modificable o

escalable, no se debe limitar a que siempre se deba tener la misma fuente de acceso a datos, es decir, tal vez en un principio se decida guardar el archivo físico de los videos, pero en otro momento se decida que es más eficiente almacenar todo el video en una base de datos determinada, o tal vez en otro tipo de base de datos. También se podría pensar en hacer un seguimiento a los movimientos del framework directamente en la base de datos, pero tal vez, después de decida hacerlo en archivos planos, o en archivos XML, etc.

El patrón DAO busca solucionar el inconveniente de cambiar entre una y otra fuente de datos. Éste patrón encapsula todos los modos de acceso a datos por medio de una interfaz que será consumida por las demás partes de la aplicación sin importar la manera en la que fue implementada.

Existe un objeto DAO por cada entidad de negocio, que implementará la conexión directa con su fuente de datos, ya sea una base de datos, un servicio de directorios, un archivo plano, etc. Así, este objeto es invocado por el objeto del negocio a través de la interfaz de una manera transparente, sin importar donde quedará almacenada su información.

La siguiente figura ilustra un diagrama de las relaciones entre las entidades asociadas en este patrón:

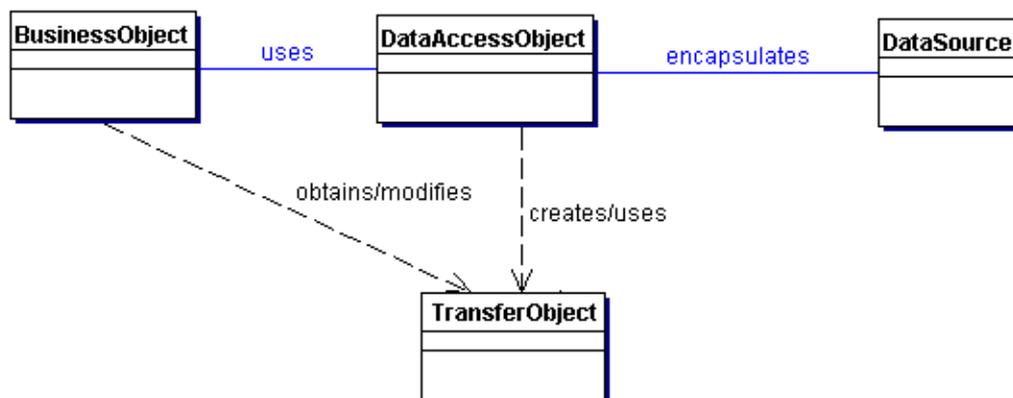


Figura 5.5

Fuente:

<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>

El objeto del negocio es quien requiere consultar, ingresar, modificar o eliminar algún dato desde su fuente, así que realiza la petición al Objeto de Acceso a Datos por medio del Objeto de transferencia, que puede llegar a ser una simple interfaz que simplemente hace el enlace. Pueden existir varias clases DAO, donde cada una guarde datos de una manera diferente.

Entre las ventajas que ofrece este patrón están:

- Permite Transparencia:

Los objetos de negocio pueden acceder a la capa de datos si importar de qué modo se accede.

- Permite Fácil Migración:

El hecho de manejar una capa solo para acceso a datos permite que cuando se quiera cambiar de fuente de datos, solo se requiera hacer el cambio en la misma capa, y no haya un mayor impacto en toda la aplicación.

- Reduce la complejidad del resto de la aplicación:

Al independizar el manejador de acceso a datos del resto de la aplicación, se disminuye la complejidad de todo el otro código que se tenga, ya que empieza a aparecer la alta cohesión, donde cada uno tiene bien definidas sus responsabilidades.

- Se centraliza toda la lógica de acceso a datos en una capa independiente.
- Esto permite tanto, reutilización como fácil mantenimiento y modificabilidad.

La única desventaja que se podría tener en cuenta acerca de la utilización de este patrón es la inversión de un mayor esfuerzo en el desarrollo inicial, pero ésta no será tomada en cuenta ya que la idea del presente proyecto es facilitar los desarrollos posteriores realizando las tareas genéricas complejas en el mismo.

## FACHADA (Facade)

El desarrollo de un framework como el planteado en este proyecto, lo que busca es facilitar el desarrollo de aplicaciones en el dominio de IPTV específicamente. Ahora, ¿qué tal fácil sería para un programador el tener que estar invocando un gran número de clases NO desarrolladas por él y con las cuales no está muy familiarizado?

El patrón Fachada propone centralizar todas las llamadas sobre un subsistema en una sola clase, de tal manera que se disminuya el número de conexiones entre subsistemas y sean mucho más fáciles tanto hacer invocaciones como modificaciones.

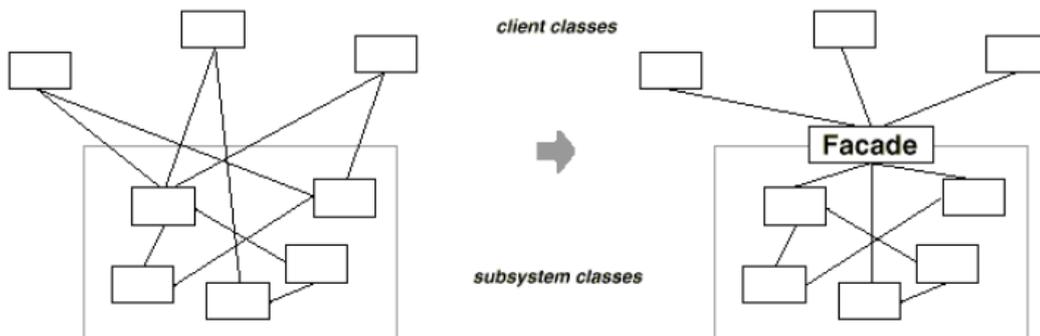


Figura 5.6

**Fuente: Design Patterns Elements of Reusable Object-Oriented Software Erich Gamma - Richard Helm - Ralph Johnson - John Vlissides**

En la imagen de la izquierda se observa un subsistema muy poco aislado de los demás, donde todo el que quiera puede invocar métodos de las clases internas del subsistema. En la imagen de la derecha se expone una clase fachada que se encarga de la invocación de las clases y métodos internos para que los objetos externos puedan accederlos.

Al buscar facilidad en los desarrollos, no solo se buscará que haya elementos reutilizables simplemente, sino también que éstos sean fáciles de usar, que no haya una alta complejidad para buscar de donde se debe invocar cierto método, sino que se centralice todo en un mismo punto para ser invocado.

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

## 6. Trabajos Relacionados

### 6.1 Frameworks e Investigaciones

- Delivery System for Networked Home Entertainment [1]

Esto es una tesis de maestría, la cual busca crear un prototipo de aplicación que permita realizar la transmisión de video, y la implementación del servicio IPTV para clientes livianos, como lo es un Set Top Box.

Restricciones: Este es un proyecto creado específicamente para clientes livianos, los cuales son costosos y no se encuentran disponibles para un entorno educativo.

- A Framework for Future Internet-Based TV Broadcasting [54]

Framework para aplicaciones IPTV, donde cada cliente pueda visualizar al mismo tiempo varios canales y enviar sus propios videos vía broadcast, y así tener un sin número de canales disponibles.

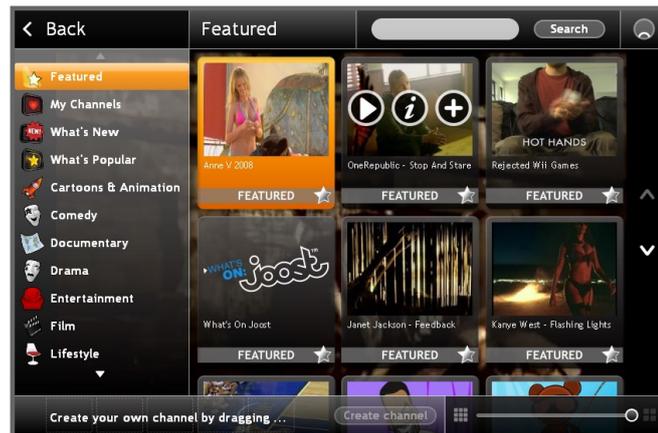
Restricciones: No está diseñado específicamente para el ámbito educativo, por lo que no se tendría acceso a funcionalidades colaborativas como el foro, etc. Adicional a esto, no se tiene acceso al código libre para realizar las modificaciones necesarias.

### 6.2 Aplicaciones

Las siguientes son aplicaciones comerciales que funcionan como clientes IPTV pero, el hecho de ser aplicaciones ya marcan una razón de la necesidad de desarrollar un framework, debido a que tienen una limitación en cuando a funcionalidades predefinidas a la hora de su desarrollo, mientras que el framework permite la personalización de las aplicaciones mediante un grupo de funcionalidades enmarcadas en IPTV.

- Joost: [55]

Creado por los fundadores de (Kazaa y Skype) se está convirtiendo en una de las aplicaciones líderes en el mercado de la transmisión de video por Internet, con una interfaz amigable y un gran número de canales para la visualización de televisión en vivo y el servicio de video bajo demanda.

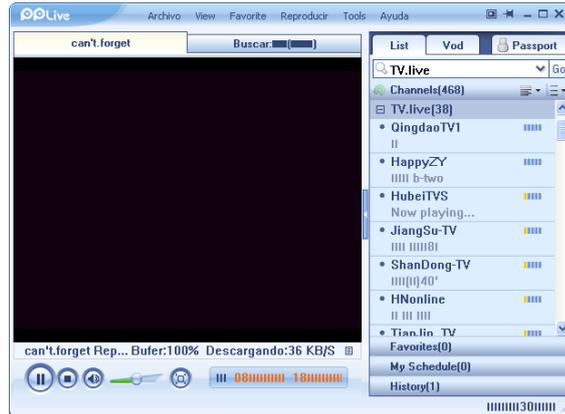


Restricciones:

Aunque es gratis, no se tiene acceso a los fuentes y está muy regido por la funcionalidad de transmisión y video bajo demanda, por lo cual se perdería la posibilidad de la participación activa de los alumnos en clases remotas.

- PPLive:

Es un aplicativo desarrollado en China y para chinos. Al igual que Joost, permite visualizar canales de televisión por Internet y acceder a una biblioteca de videos online.



Restricciones: Está desarrollada para tres idiomas (Mandarin, Cantonese y Koreano). Es software comercial al cual no se le pueden realizar modificaciones.

## 7. Metodología

La metodología a utilizar para este desarrollo es RUP (Rational Unified Process), ya que tiene gran prestigio para los grandes desarrollos de software en el medio. Es de aclarar que no se seguirá estrictamente el proceso de Rational, sino que se adaptará a las necesidades del proyecto, ya que el seguirlo al pie de la letra implica un gran número de artefactos y fases que serían útiles para un proyecto con muchos participantes y varias fases de definición, lo que para nuestro proyecto particular no aplica completamente. De cualquier manera se seguirá un esquema básico de esta metodología que será explicado más adelante y que está basado en el criterio que tenemos sobre la misma y su prestigio a nivel de desarrollo de software.

RUP implica que sea dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura. [10]

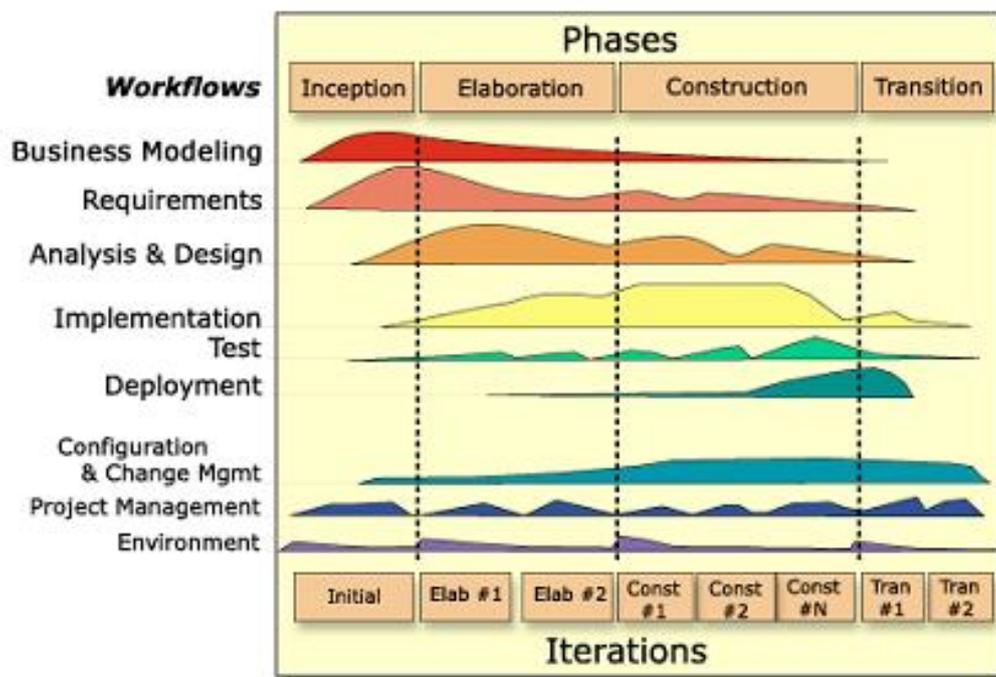


Figura 7.1 RUP

Fuente: [http://www.msc-inc.net/Documents/rup\\_fundamentals\\_presentation.htm](http://www.msc-inc.net/Documents/rup_fundamentals_presentation.htm)

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

La figura anterior muestra las diferentes fases de RUP y los diferentes flujos de trabajo exponiendo la relación entre ambos. En el lado izquierdo de la gráfica se listan los flujos de trabajo que normalmente tiene un proceso de desarrollo de software durante su tiempo de vida:

- Modelado del negocio
- Requerimientos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue
- Administración del cambio y la configuración
- Administración del Proyecto
- Ambientación del proyecto

En la parte superior de la imagen, se enuncian las cuatro fases involucradas en el desarrollo de software:

- Concepción
- Elaboración
- Construcción
- Transición

Dentro de cada una de estas fases pueden ocurrir una o más iteraciones que se representa con un producto resultante que se le entrega al cliente como avance del proyecto. Este número de iteraciones depende de la magnitud del proyecto.

Por último, en el centro de la imagen se puede observar la cantidad de esfuerzo requerido de cada flujo para una fase determinada del proyecto. En la fase de concepción por ejemplo, del proyecto de encuentra iniciando, así que el esfuerzo mayor se debe enmarcar en el modelado del negocio y la toma de requerimientos, mientras que en la fase de construcción los flujos de implementación y pruebas son los que requieren mayor dedicación. Los tres últimos flujos, Administración del cambio y configuración, administración del proyecto y Ambientación del proyecto, que son labores más

administrativas, mantienen un trabajo constante durante toda la duración del proyecto.

Para la ejecución de este proyecto, se tuvieron en cuenta los cinco primeros flujos de trabajo (Modelado del negocio, Requerimientos, Análisis y Diseño, Implementación y Pruebas) pues como se explicó anteriormente, algunas de las cosas no aplican para un proyecto en el cual se tienen solo tres desarrolladores que a su vez son revisores de calidad de la aplicación (tester), gerentes del proyecto, documentalistas, etc. Con los cinco flujos elegidos se considera que es suficiente para el presente desarrollo. En los siguientes puntos se describirá brevemente el resultado de cada uno de estos flujos.

### **7.1 Modelado de negocio**

En este flujo de trabajo se obtuvo una idea acerca de la evolución de la televisión hacia nuevas tecnologías, y si vislumbró la necesidad de incluir a la Universidad EAFIT en este concepto, lo cual le permitirá a la institución a ser una de las precursoras en el país en un nuevo concepto de la “educación digital”. Además, se planteó el desarrollo de un framework como una muy buena oportunidad para facilitar la labor de las personas que deseen crear nuevas aplicaciones y evitar la duplicidad innecesaria de código cuando se tiene la posibilidad de tener todo centralizado. Por último, y luego de revisar las herramientas que se encuentran en el medio, no se pudo obtener ninguna que tuviese el componente educativo como prioridad en su desarrollo, además del marco comercial que tienen muchas de estas.

### **7.2 Requisitos**

Luego de encontrarnos con estas necesidades en el mercado que evoluciona continuamente, se determinaron los requisitos funcionales y no funcionales del framework. Debido a la gran cantidad de funcionalidades que

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

ofrece la televisión digital, se definió un grupo de requisitos básicos para justificar el framework dentro de la tecnología IPTV. Dentro de los logros importantes de este flujo, se encuentra la definición de que el framework no solo debería permitir construir un tipo de aplicaciones, sino que debería tener funcionalidades disponibles para aplicaciones cliente (aplicaciones ejecutadas localmente en una máquina o un Set Top Box), como aplicaciones servidor (encargadas de atender solicitudes de uno o varios clientes y realizar las transmisiones de video).

Para mayor información acerca de los requisitos de este proyecto, refiérase al documento de diseño, el cual se adjunta con este documento.

### ***7.3 Análisis y Diseño***

En esta etapa se tuvo que realizar una exhaustiva revisión acerca de las características que debe tener un framework y los patrones de diseño sobre los cuales nos podríamos basar para alcanzar estas características. Se definieron las clases necesarias para el negocio, una base de datos acorde al proyecto, y un grupo de políticas que deberíamos seguir para obtener un framework flexible, escalable, utilizable y reutilizable. Para mayor información acerca del análisis y diseño de este framework, por favor referirse al documento de diseño, adjunto a este documento.

### ***7.4 Implementación***

La implementación del framework consistió, en la construcción y documentación completa del framework para permitir la construcción de aplicaciones inmersas en el entorno de la televisión digital, basados en el diseño previamente construido y los patrones elegidos.

### ***7.5 Pruebas***

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

En cuanto a la revisión de calidad de la herramienta desarrollada, se realizaron desde dos frentes muy bien delimitados:

- Pruebas unitarias: Se verificó la funcionalidad de cada componente del framework, como es transmisión de video, accesos a base de datos, servicios de mensajería, etc.
- Construcción de aplicaciones: Como lineamiento se definió que para sustentar nuestro framework como útil, describiendo útil como un framework que disminuya el esfuerzo de desarrollo de aplicaciones en un entorno específico, se deberían crear tres aplicaciones, teniendo un tiempo estimado del desarrollo de las mismas sin la utilización del framework, y tomando el tiempo real necesitado para desarrollarlas utilizándolo.

## 8. Framework para el desarrollo de aplicaciones IPTV en el entorno educativo (FIT)

En esta sección se describirá la manera como se construyó el framework, los componentes del mismo, los inconvenientes que se tuvieron en su desarrollo y las ventajas de haberlo construido de esta manera.

### 8.1 Arquitectura.

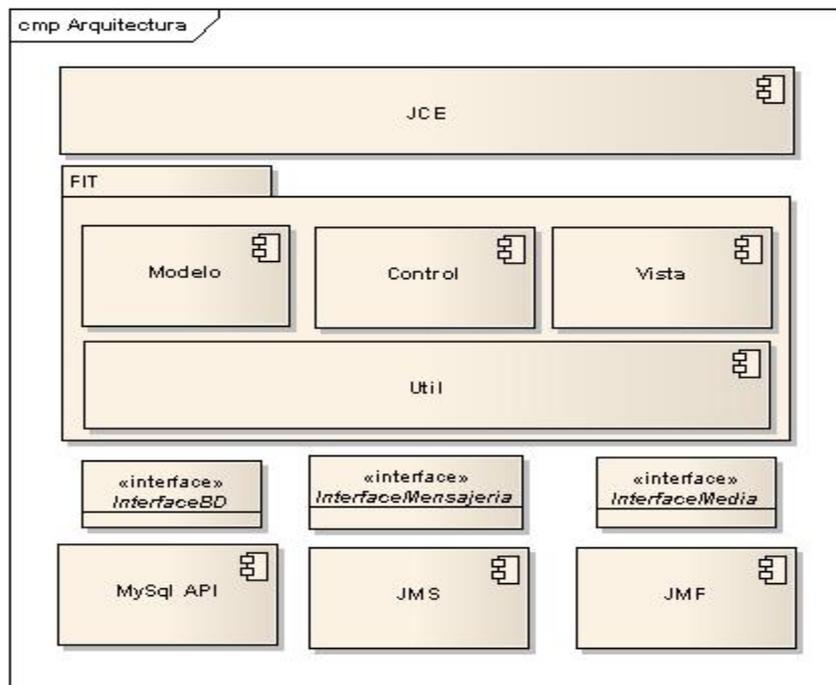


Figura 8.1 Arquitectura del FIT.

En la parte central de la imagen está ubicado el framework que será desarrollado, con los tres componentes del patrón MVC bien definidos. Es de aclarar que al no tratarse de una aplicación sobre la cual el usuario final realice acciones gráficas, en la vista solo se ubicarán las clases necesarias para invocar las funciones básicas del framework. Así pues, es en este punto

donde introduciremos otro de los patrones mencionados previamente, el patrón Fachada. En este componente (Vista), solo se incluyen dos clases encargadas de invocar las funcionalidades del framework. Por último, como es evidente, el patrón DAO será incluido en el componente modelo, donde se encuentran las clases del negocio, es decir clases como Video, Foro, Transmisión, etc. Por último, en el componente Util, transversal a los demás, se manejan las operaciones no funcionales del framework y que sirven como apoyo, como es el manejo de excepciones y logger.

Alrededor de la imagen se puede encontrar los componentes de software, planteados como útiles para este desarrollo. Tanto JMF como JMS tienen comunicación directa con el componente central del framework (Control), ya que éste será quien se encargue de coordinar la transmisión y recepción de mensajes y videos. JCE por su parte, aunque inicialmente solo será utilizado para la encriptación y desencriptación de la cadena de conexión a base de datos, podría ser utilizado con otros fines, motivo por el cual se muestra transversal a todas las capas del framework. Por último, el componente de conexión a base de datos se muestra paralelo al Modelo, que es quien tendrá la implementación del patrón DAO y por tanto las conexiones a base de datos.

## **8.2 Herramienta Multimedia**

Tal vez uno de los aspectos de mayor impacto en el desarrollo del proyecto elegir una herramienta que nos permita la captura, transmisión y reproducción de archivos multimedia, que sea confiable, tenga un buen soporte y esté disponible para diferentes plataformas. Las herramientas que se analizaron para el proyecto son QTJava, VLC, JVLIC y JMF.

¿Qué es QTJava?

Quick Time for Java es una API desarrollada por apple para la transmisión y reproducción de archivos multimedia, es de fácil uso y está soportado por

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

una de las más grandes compañías en la industria. Esta herramienta fue descartada para el presente desarrollo ya que tiene la gran restricción de que la parte de transmisión de video a través de la red está solo desarrollada para Macintosh, lo que implicaría desarrollar la parte del servidor del framework en este sistema operativo, no muy común en Colombia y mucho menos en la universidad.

¿Qué es VLC?

Video Lan Client es un reproductor de multimedia distribuido con licencia GPL y desarrollado en el lenguaje de programación C, que permite el llamado a funciones de reproducción de video y audio, como el transmisión de multimedia vía streaming para ser accedido remotamente.

¿Qué es JVLC?

JVLC es Java VLC. Hace unos años fue implementada una versión en Java que utiliza los mismos fuentes de VLC en su lenguaje original, pero se conecta por medio de JNI (Java Native Interface) para invocar estos métodos desde Java.

A pesar de la gran posibilidad que brinda este proyecto para nuestro Framework, aún no ha sido completamente desarrollado y, según los autores, solo ha sido desarrollado para Linux, lo cual traería la primera restricción para nuestro proyecto. Más adelante se explicará más claramente por qué no es viable su utilización.

¿Por qué no utilizar VLC o JVLC para el Framework?

VLC, un proyecto que lleva muchos años y que permite el acceso a sus fuentes para desarrollos adicionales, se planteó inicialmente como una buena alternativa, realizando una conexión JNI con estos fuentes directamente desde nuestro framework, pero poco tiempo después se

encontró poco viable debido al tiempo y la dedicación que esto implicaría, ya que esto alcanza las dimensiones de un proyecto completo.

Así pues, apareció JVLC como la alternativa desarrollada para lo que pensábamos hacer inicialmente (conexión JNI entre VLC y nuestro framework), pero aquí nos encontramos con varios inconvenientes:

- El ya mencionado, JVLC solo está completamente desarrollado para Linux. Se limita nuestro proyecto a únicamente ser desarrollado en ambientes Linux, y se elimina completamente la posibilidad de utilizarlo en el ambiente más común cuando se computadores personales de habla, Windows.
- JVLC es un proyecto relativamente nuevo y no tiene suficiente documentación. Para tomar este API y utilizarlo como base para un framework, lo mínimo que se requiere es una buena documentación para poder emplearlo acordemente, la cual no existe.
- JVLC no está disponible para todas las distribuciones de Linux, lo cual limita incluso más los ambientes de desarrollo y ejecución. Intentamos su utilización en la distribución Suse 9 y fue imposible, luego pasamos a utilizar el recomendado de la página (Ubuntu) y esta búsqueda fue un poco más fructífera, pero después de instalar una gran cantidad de prerequisites, funcionaba pero arrojando errores para los cuales no se tenía mucho soporte.
- JVLC se vuelve muy engorroso a la hora de ser utilizado, ya que tiene demasiados prerequisites, donde las versiones no están muy claras y esto complicaría mucho las cosas a la hora de desplegar nuestro framework, volviéndolo inutilizable.

Por todas estas razones, hemos tomado la decisión de NO utilizar JVLC ni VLC y simplemente utilizar Java Media Framework (JMF), desarrollado por Sun y del cual estamos seguros que puede suplir nuestras necesidades puntuales.

Dejamos claro que al estar desarrollando un framework lo haremos lo suficientemente modular para que en un proyecto futuro se realice la conexión con VLC u otra herramienta que brinden opciones adicionales a las que nos pueda ofrecer JMF.

### **8.3 Protocolo de comunicación.**

Para la comunicación de las aplicaciones cliente con el servidor (Solicitudes o cancelación de videos), se ha definido un pequeño protocolo:

#### Objetivo:

Definir un esquema de comunicación entre las aplicaciones construídas con el Framework de Desarrollo IPTV (FIT).

#### Descripción:

Se desea diseñar e implementar un protocolo de nivel de aplicación punto a punto (STV), sobre IP en una red de área local. El protocolo STV debe ofrecer el servicio de transmisión/recepción de información a dos protocolos de nivel de aplicación: Transferencia de archivos (FT) y Chat (CHAT).

Funcionalmente la siguiente gráfica muestra la estructura de los protocolos:

#### Descripción del protocolo:

##### STV: Sistema de Transmisión de Video.

- Implementado sobre el servicio de mensajería provisto por JMS (Servicio de mensajería de Java) con el servidor OpenJMS.
- No orientado a la conexión
- Comunicaciones asincrónicas
- Unidad de transmisión es ObjectMessage (Java)
- Primitivas de servicio:
  - STV.Request(idMensaje, datos)
  - STV.Send(datos)

Vocabulario de Mensajes

Los siguientes son los tipos de mensajes que soporta el protocolo STV:

Mensaje	Descripción
RVideo	Mensaje que solicita transmisión de un video
CVideo	Mensaje que solicita la cancelación de la transmisión de un video.

Codificación de tramas

[1]	[8]	[20]	[30]
IDMensaje	IDVideo	IDUsuario	DATOS

*IdMensaje (1 byte)*: Código del mensaje que se está enviando.

*IdProducto (8 byte)*: Código del video relacionado con la solicitud.

*IdUsuario (20 byte)*: Código del usuario que está realizando la solicitud.

*DATOS (30 bytes)*: Campo de datos que contiene la dirección IP del Cliente.

Procedimiento

Solicitud de Video.

La aplicación cliente solicita la transmisión de un video al servidor por medio de un mensaje de tipo Rvideo, con el identificador del video y del usuario, y con la IP de la máquina en el cuerpo del mensaje. El servidor envía por medio de JMF un tráfico Unicast con el video solicitado. Este procedimiento se ilustra en la siguiente imagen:

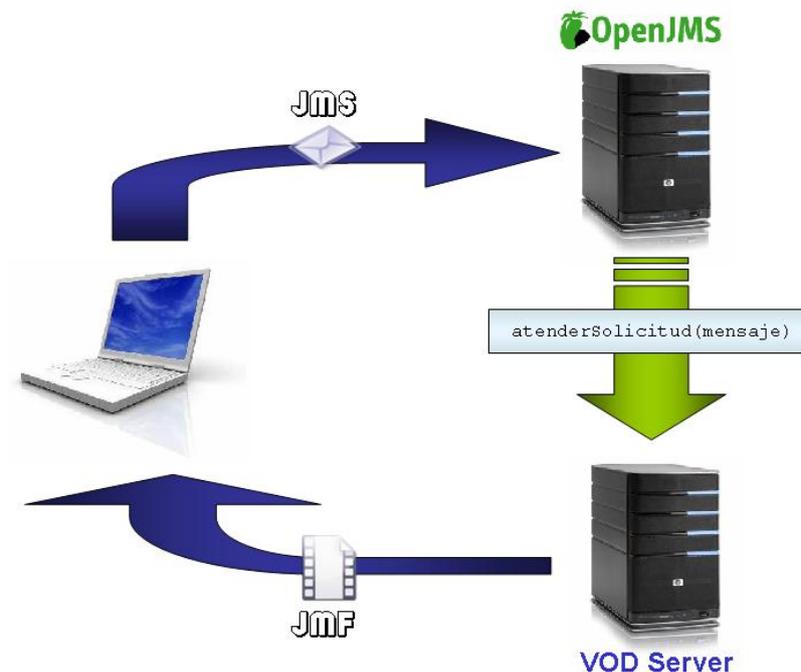


Figura 8.2 Procedimiento de una solicitud de video al servidor de VOD

Cancelación de Video.

La aplicación cliente solicita la cancelación de la transmisión de tipo CVideo, con el identificador del video y de la máquina en el cuerpo del mensaje. El servidor cancela este envío.

## 8.4 Logger

Para facilitar las funciones de seguimiento de errores en el framework se desarrollo una clase Logger, el cual registra cualquier tipo de excepción en un archivo de texto local. Para su utilización solo basta con invocar el método estático: *Logger.escribirLog*. Para mayor información, revisar el JavaDoc del proyecto.

## 8.5 Modelo de datos

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

El siguiente es el modelo de datos empleado en el framework. El número de tablas requeridas no es muy alto, ya que la información necesaria se basa únicamente en videos, foros, canales y transmisiones.

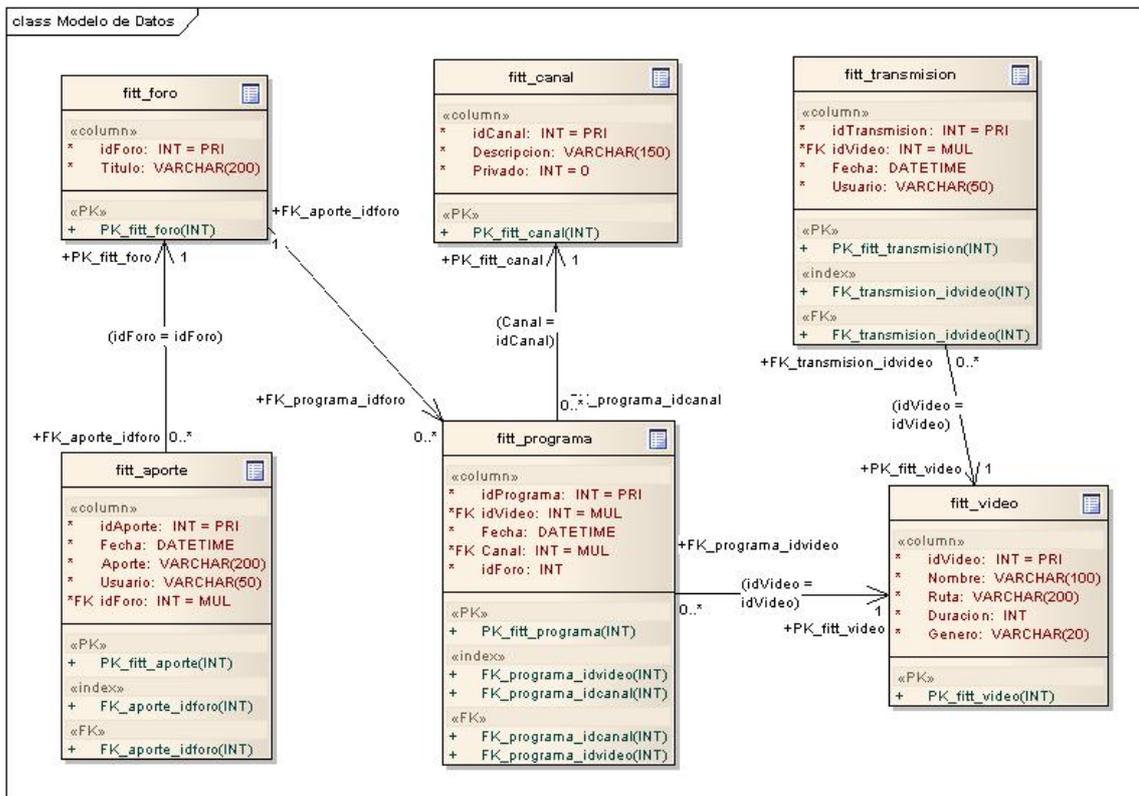


Figura 8.3 Modelo Entidad Relación - FIT

La descripción de las tablas es la siguiente:

**FITT\_VIDEO:** Tabla que almacena la lista de los videos disponibles. El atributo ruta se refiere a la ubicación física donde se encuentra ubicado en el servidor o, en el caso de las transmisiones en vivo la identificación del dispositivo del cual se capturará el video.

**FITT\_APORTE:** Tabla que almacena los aportes hechos por un usuario a un foro.

**FITT\_FORO:** Tabla que almacena un foro con su título.

**FITT\_CANAL:** Tabla que almacena un canal por el cual se puede transmitir

un programa.

FITT\_PROGRAMA: Tabla que almacena un programa, que consta de el video será transmitido por un canal una fecha determinada y que puede o no, tener un foro asociado.

FITT\_TRANSMISION: Tabla que almacena el histórico de las transmisiones asociadas, esto depende de las funcionalidades que el desarrollador que utilice el framework desee, está diseñada como un log de transmisiones para la generación de reportes.

## **8.6 Clases**

La abstracción a nivel de objetos y clases, y la descripción detallada de los componentes del framework se puede visualizar en el siguiente diagrama:

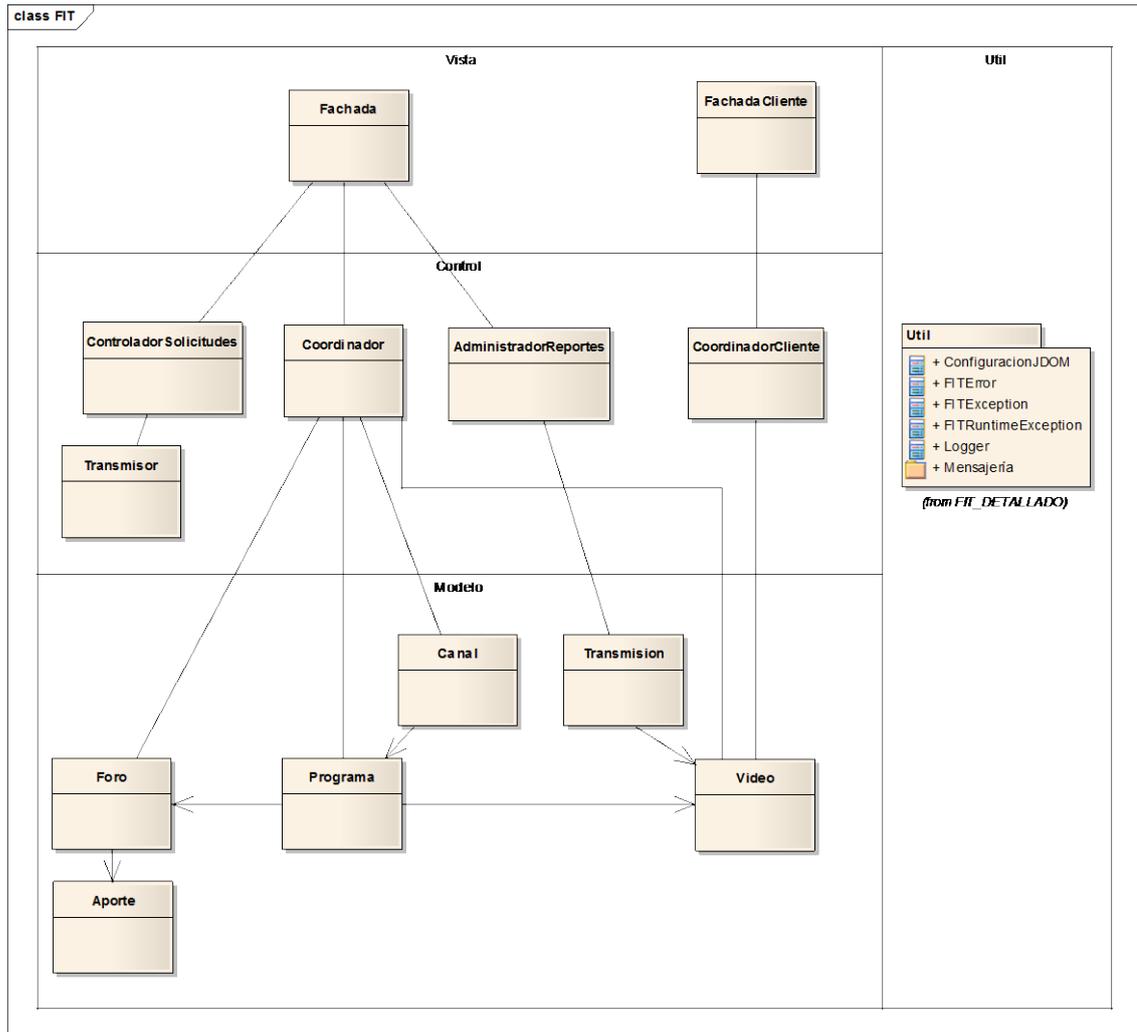


Figura 8.4 Diagrama básico de clases del FIT

Este diagrama es aún un poco menos detallado que el completo, ya que por cuestiones del tamaño de la imagen, el artefacto (en términos de UML) que tiene la descripción de las clases con atributos y métodos fue separado en varios diagramas, uno para cada uno de los recuadros (Modelo, Control, Vista, Util) que serán descritos posteriormente en este mismo documento. Aún así, se decidió también realizar este otro modelo, sin mucho nivel de detalle pero con la estructura completa del framework para tener una visión general de toda su composición.

Se aprecia claramente la separación entre los tres componentes sugeridos por el patrón MVC, y las ventajas que el hacerlo así traerá, no solo a la hora de modificarlo, sino incluso en el momento de su invocación.

En la sección del modelo se tienen las clases básicas del negocio: Video, Aporte, Canal, Programa, Transmisión y Foro.

### ▪ 8.6.1 Modelo

En este componente se puede apreciar claramente la distribución de las clases que constituyen el negocio:

- Video: Se podría describir como la clase básica para casi todas las funcionalidades inmersas en el framework, los atributos más importantes son su identificación, su nombre y la ruta donde el aplicativo debe ir a buscarlo (ya sea una ubicación física en el disco o un puerto donde leerlo). Adicional a esto se tiene alguna información como el género o la duración que se comporta como meta data para realizar más amigable las búsquedas a los usuarios.
- Aporte: Una de las clases que representa la parte del negocio que quizá podría funcionar sin tener en cuenta un Video (la otra sería Foro), aunque no fuese concebido así, representa el texto enviado por un usuario como comentario ó pregunta acerca de un programa específico.
- Foro: Clase que representa la herramienta colaborativa definida en el framework como apoyo esencial para las labores educativas permitiendo interactividad entre los diferentes participantes de una clase o conferencia. Se compone de una lista de aportes y un título para el foro.
- Programa: Componente vital para la televisión digital en el ámbito educativo, ya que asocia cada video con un foro y le asigna una fecha. Se podría definir como la unidad básica que se transmite por un canal.
- Canal: Representa lo que sería equivalente a un canal de la televisión tradicional, donde se están transmitiendo videos constantemente durante el día sin importar si están siendo visualizados o no. Dos de los atributos que marcan diferencia en esta clase sería la lista de programas que tiene un canal, los cuales estará transmitiendo según su fecha asociada, y la dirección IP, la cual hace referencia a una

dirección Multicast en la red sobre la cuál estará transmitiendo estos programas.

- Transmisión: Además de ser una clase clave para el manejo de históricos de transmisiones y administración de estadísticas se convierte en una oportunidad perfecta para aplicaciones como la programación por parte de los usuarios de transmisiones de videos en unos horarios determinados.

Se puede apreciar que ninguna de estas clases contiene métodos que por si solos servirían directamente para el propósito del framework, pero son la base del mismo. Los métodos encontrados acá son los que permiten acceder a los atributos de las clases (“accesores”) y los métodos CRUD (Crear, Consultar, Actualizar y Eliminar).

### Figura 8.5 Diagrama de Clases - Modelo

Por último, se muestra la implementación del patrón Objeto de Acceso a Datos (DAO) con la diagramación de las interfaces que heredan de una clase padre que contiene los métodos genéricos de acceso a los datos. Es importante anotar que se diagraman las interfaces pero cada una tiene una implementación asociada, construido así además, para permitir la fácil migración del modelo de datos a otra tecnología (en nuestro caso MySQL).

### ▪ 8.6.2 Control

Las clases encontradas en este componente serán quienes realicen el enlace entre el modelo y la vista, son las encargadas de atender solicitudes de los clientes y convertirlas en algo palpable.

- Transmisor: La única clase que no tiene comunicación con los demás componentes del framework, es la encargada de realizar una implementación de JMF de acuerdo a las necesidades del FIT, permite crear, detener, reanudar y cancelar una transmisión de video.
- Coordinador Solicitudes: Representa al objeto que será despertado una vez llegue un mensaje a la cola del servidor OpenJMS y para que se encargue de procesarlo y, si es necesario iniciar la transmisión de un video.
- Coordinador Cliente: Contiene las funcionalidades que únicamente serían necesarias para aplicaciones cliente, básicamente sería el envío de solicitudes convertidas en Mensajes a través de JMS para la recepción de un video.
- Coordinador: Representa el centro del framework, es quien mayor número de conexiones tiene al permitir el acceso y modificación a todos los objetos del Modelo.
- Administrador Reportes: La clase para la generación de estadísticas acerca de los videos solicitados por los usuarios. Permite la generación de estos reportes por uno de tres criterios:
  - Fecha de Transmisión.
  - Usuario
  - Video.

**Figura 8.6 Diagrama de Clases - Control**

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

### ▪ 8.6.3 Vista

Consta de dos únicas clases que implementan el patrón Fachada para centralizar las funcionalidades del framework y evitar que el desarrollador tenga problemas para encontrar cuál clase debe invocar para lograr una funcionalidad específica. Al igual que los coordinadores en el Control, se decidió separar en dos clases, una que tenga las funcionalidades comunes a los dos tipos de aplicaciones (Cliente y Servidor) y otra que tenga las funcionalidades específicas de las aplicaciones cliente.

**Figura 8.7 Diagrama de Clases - Vista**

#### ▪ 8.6.4 Util

El componente Util se subdividió en dos diagramas, ya que el modelo de mensajería ameritó un diagrama propio para evitar posibles confusiones acerca de las relaciones entre estos dos grupos de clases.

#### **General**

Contiene las clases que apoyan el desarrollo del framework y lógica No funcional del mismo.

- **Logger:** Clase encargada de realizar el registro de eventos inesperados del framework en un archivo de texto plano ubicado en la carpeta Log de la ruta donde se encuentre instalada la aplicación.
- **ConfiguraciónJDOM:** Clase que permite la lectura del archivo Xml de configuración, el cual tiene los tags:
  - Servidor en la base de datos.
  - Usuario de base de datos
  - Password en base de datos
  - Ubicación lógica en la red del servidor
- **Excepciones:** Las tres clases restantes corresponden a entidades que representa excepciones personalizadas para el framework.

**Figura 8.8 Diagrama de Clases - Util**

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

## Mensajería

Este subconjunto de clases está compuesto de las entidades necesarias para realizar una comunicación por medio del servicio de Mensajería de Java.

- Mensaje: Clase básica para el protocolo definido (STV). Los atributos son los siguientes:
  - o enTipoMensaje: Enumeración que contiene el tipo de mensaje, RVIDEO, CVIDEO.
  - o IdProducto: Identificación del video solicitado.
  - o objDatos: Dirección ip del cliente que solicita el video.
  - o sUsuario: Identificación del usuario que solicita el video.
- Transmisor: Se encarga de iniciar la conexión con el servidor OpenJMS y enviar los mensajes.
- ReceptorAsinc: Contiene el método disparado por JMS al recibir un mensaje en una cola determinada.
- Receptor: Inicializa el servidor que se duerme para esperar la llegada de mensajes de solicitudes.

Figura 8.9 Diagrama de Clases – Util – Mensajería

## 8.7 Código Fuente

A continuación se muestran algunos fragmentos del código del framework que pueden describir un poco más el funcionamiento del framework:

Interfáz de la clase padre de la implementación del patrón DAO.

```
public interface DAO {  
  
    <T> int crear(T objectToInsert);  
  
    <T> boolean borrar(T objectToInsert);  
  
    <T> T buscar(Class<T> clazz, int id);  
}
```

Métodos crear, borrar y buscar de la clase Video:

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

```
/**
 * Metodo por el cual un objeto video se crea a si mismo en la base de datos.
 * @return El identificador unico que se genero en la base de datos al crear el
 * video o -1 si no fue creado con exito.
 */
public int crear(){
    return VideoDAOImpl.crear(this);
}
/**
 * Metodo por el cual un video se borra a si mismo de la base de datos.
 * @return Un valor booleano para saber si el video fue borrado con exito.
 */
public boolean borrar(){
    return VideoDAOImpl.borrar(this);
}
/**
 * Este metodo permite buscar un video en la base de datos partiendo de su identificador unico.
 * @param El identificador del video que se quiere buscar.
 * @return El objeto video que se encontro, o un video nulo si no se encontro.
 */
public Video buscar(int video){
    return VideoDAOImpl.buscar(video);
}
```

Método encargado de invocar al servidor al recibir un mensaje en la cola especificada:

```
public void onMessage(Message message) {
    //Verifica que el mensaje si se de tipo Objeto
    if (message instanceof ObjectMessage) {
        ObjectMessage objObjectMensaje= (ObjectMessage) message;
        try {
            //Encapsula el mensaje como un objeto de la Clase Mensaje
            Mensaje objMensaje=((Mensaje) objObjectMensaje.getObject());
            //Invoca al servidor para atender la solicitud de dicho mensaje.
            ControladorSolicitudes.atenderSolicitud(objMensaje);
        } catch (JMSEException ex) {
            Logger.escribirLog("ERROR en ReceptorAsinc.onMessage()");
            Logger.escribirLogSinCabecera(ex.toString());
            ex.printStackTrace();
        }
    }
}
```

La siguiente es una sección del método encargado de atender las solicitudes enviadas en objetos de la clase Mensaje:

```
public static void atenderSolicitud(Mensaje objMensaje){
    try{
        //Verifica que efectivamente haya llegado un mensaje
        if(objMensaje!=null){
            //Verifica el tipo de mensaje que llegó
            if(objMensaje.getTipoMensaje()==TipoMensaje.RVVIDEO){
                //Consulta el video en la base de datos a partir del identificador enviado.
                Video objVideo = Coordinador.buscarVideo(objMensaje.getIdProducto());
                //Crea el objeto transmisor.
                Transmitter t = new TransmisorImpl();
                // crea la transmisión a partir del video consultado y los datos enviados en el mensaje
                t.createTransmission(TransmissionType.VOD,
                    "file:"+objVideo.getRuta()+objVideo.getNombre(),
                    (String) objMensaje.getDatos(), sPuerto);
            }
        }
    }
}
```

## 9. Experimentos

Para la verificación del framework, se plantea el desarrollo de tres aplicaciones utilizándolo para luego realizar una comparación de tiempos reales vs tiempos estimados sin el framework y así evaluar la relevancia y valides del framework:

**9.1 Cliente y Servidor de Video bajo demanda:** Aplicación que lista los videos almacenados disponibles para que un usuario los vea en el momento que así lo desee. Inicialmente muestra los videos nuevos, pero permite realizar búsquedas de videos por nombre, una vez el usuario lo seleccione, podrá visualizarlos. A continuación se muestra el proceso de interacción de la aplicación desarrollada con el framework.

### Figura 9.1 Diagrama de secuencia proceso de aplicación VOD

Cuando se ejecuta la aplicación, ésta invoca el framework para que le retorne los videos disponibles y los muestra para que el usuario seleccione. Cuando esta acción ocurre, la aplicación solicita al FIT el video, este a su vez envía un mensaje al servidor con la solicitud del video, el servidor

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

decodifica el mensaje y envía el video a la aplicación cliente, quien lo reproduce para que el usuario final lo visualice.

En la siguiente imagen se muestra el código requerido para crear el servidor de video bajo demanda, y la sección de la aplicación cliente donde se realiza la solicitud del video:

### Servidor

```
package test;

import co.com.fit.vista.Fachada;

public class TestServer {

    public static void main(String[] args) {
        //Se crea un objeto de tipo fachada.
        Fachada objFachada=new Fachada();
        //Se inicializa el servidor.
        objFachada.initServidor();
    }
}
```

### Cliente

```
public void mouseClicked(java.awt.event.MouseEvent evt) {

    String stCodigo = btnVideo.getName();

    try
    {
        //Se envia la solicitud del video por medio de un mensaje al servidor OpenJMS
        FachadaCliente.solicitarVideo(sUsuario, getLocalIpAddress(), Integer.parseInt(stCodigo));
    }
    catch (NumberFormatException e1)
    {
        Logger.escribirLog("VODFIT.PrincipalVOD"+e1.toString());
    }
    catch (FITException e1)
    {
        Logger.escribirLog("VODFIT.PrincipalVOD"+e1.toString());
    }
    try
    {
        //Se crea el servidor con el flujo de llegada
        createPlayer("rtp://" + servidor + "/video");
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```



Figura 9.2 Imagen de aplicación VOD

### 9.2 Cliente y Servidor de Guía electrónica de programación:

Aplicación que permite consultar la programación de los canales que normalmente se transmiten vía multicast desde un servidor IPTV. El usuario puede realizar búsquedas por nombre del programa, canal y fecha.

Figura 9.3 Diagrama de secuencia proceso de aplicación EPG

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

El usuario selecciona una fecha para la cuál desea ver la guía, la aplicación solicita al FIT la lista de canales y programas para esa fecha, el FIT realiza la consulta en base de datos y retorna la información. La aplicación despliega la información y permite realizar búsquedas por canal.

La siguiente imagen ilustra una parte del esfuerzo que se requirió para el desarrollo de esta aplicación.

```
/**
 * Método que llena la lista de canales a partir de una fecha
 * @return void
 */
public void traerCanales(){
    lCanales = Fachada.consultarEPG(getFecha());
}

/**
 * Método que llena la lista de canales a partir de una fecha
 * @return void
 */
public void buscarProgramas(String cadBusqueda)
{
    Canal objCanal = new Canal();
    for(Canal recCanal: lCanales){
        if(recCanal.getDescripcion().compareTo(canalActual) == 0){
            objCanal = recCanal;
            break;
        }
    }
}
```

Se aprecia el método que permite consultar una EPG a partir de la invocación de un único método en la clase Fachada, y una sección del método que utiliza los datos cargados en memoria para realizar la búsqueda de los programas de un Canal en esa fecha.

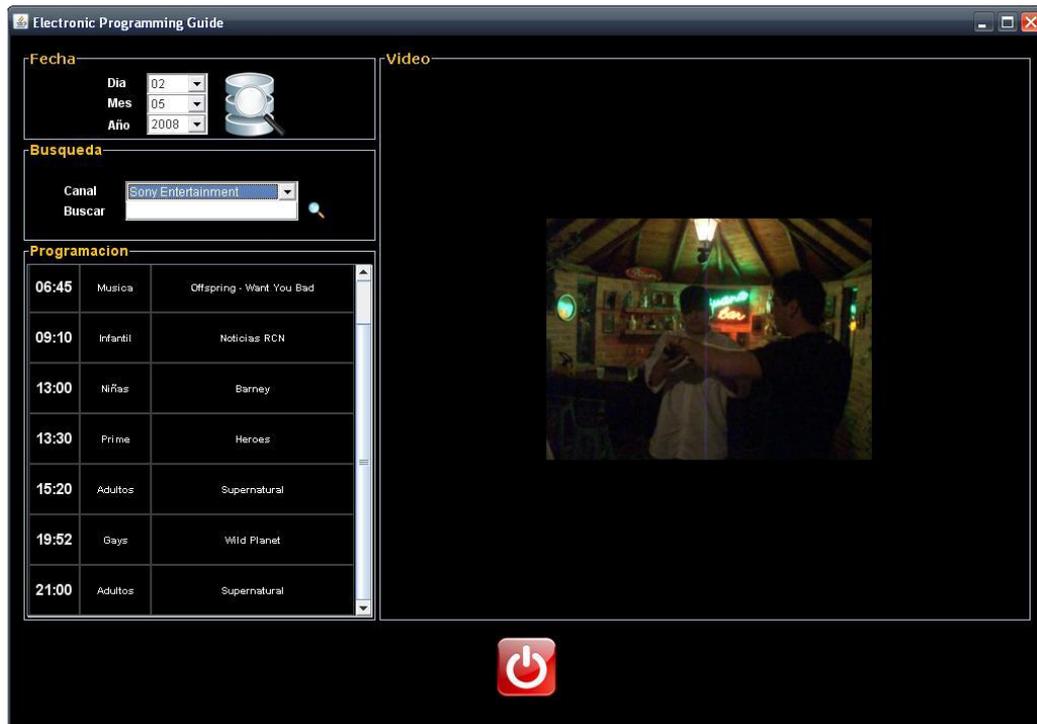


Figura 9.4 Imagen de aplicación VOD

**9.3 Servidor de transmisión programada de canales:** Aplicación que crea un servidor que se programa a partir de una base de datos para realizar la transmisión continua de todos los canales a una dirección IP multicast determinada para ser vistos por los usuarios que así lo deseen.

El flujo de la aplicación es el siguiente:

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

**Figura 9.5 Diagrama de secuencia proceso de aplicación Transmisión programada**

- Se inicializa el servidor, el cuál se encarga de leer la lista de canales y programas que debe transmitir en ese momento.
- Comienza la transmisión de todos los videos a través de la red hacia direcciones Multicast.
- El usuario solicita un canal para visualizarlo.
- La aplicación le solicita la reproducción de ese canal al fit y comienza a escuchar la dirección Multicast correspondiente para iniciar la reproducción del video actual.

## 10. Resultados

Se lograron desarrollar las aplicaciones en un corto tiempo con respecto a lo estimado, con alta reutilización de los componentes del framework y sin mayores dificultades.

Para realizar el análisis de los resultados se construyó una tabla del tiempo que se estima se hubiese requerido para desarrollar cada una de estas aplicaciones sin el framework, contra el tiempo utilizado realmente para desarrollarlas.

Aplicación	Tiempo Estimado (Horas)	Tiempo Real (Horas)
Video Bajo demanda	150	30
Guía Electrónica de Programación	25	9
Transmisión en vivo	180	40

Tabla 10.1

**Nota:** Este tiempo fue calculado basado en el método Delphi, el cual consiste en la consulta a expertos en el tema acerca de un tiempo estimado dada la definición de una aplicación. Para nuestro caso, tomamos tres fuentes para realizar esta consulta: dos desarrolladores de software de la compañía PSL (Productora de Software S.A.) y uno de Axede S.A (antes Trébol Software S.A.). A estas tres personas se les entregó el formato que vemos a continuación a partir del cual se tomaron tiempos promedio para cada aplicación como un tiempo estimado confiable:

Camilo Vieira Mejía  
Carlos Andrés Valencia Sánchez  
Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

<b>INTRODUCCION</b>			
<p>El siguiente formato fue definido para realizar una estimación de tiempo de implementación de tres aplicaciones multimedia.</p> <p>En la parte inferior usted encontrará una breve descripción de cada una de las aplicaciones, para un mayor detalle puede consultar a la persona que le entregó la encuesta.</p> <p>Adicionalmente encontrará una lista de tareas que se espera le sirvan de guía para asignar unos tiempos acordes a cada implementación que se planea realizar, si existe alguna actividad que usted piense que no se encuentra en esta lista, le puede asignar el tiempo en el último numeral (Otros).</p>			
<b>Datos Básicos</b>			
<b>Nombre</b>		<b>Edad</b>	
<b>Experiencia</b>		<b>Empresa</b>	
<b>Descripción de la aplicación</b>			
<b>Nombre y Descripción</b>	<p><b>Video Bajo Demanda</b></p> <p>Se requiere desarrollar una aplicación que permita transmitir videos almacenados en una máquina a través de una red de área local de acuerdo con solicitudes que realice una aplicación cliente en cualquier momento.</p> <p>Estos videos podrán ser consultados y visualizados por la misma aplicación cliente que realiza la solicitud.</p>		
<b>Tiempos</b>			
<b>Tarea</b>			<b>Tiempo en Horas</b>
1. Análisis y Diseño			
2. Creación de la base de datos			
3. Implementación de la aplicación servidor			
4. Implementación de la aplicación cliente			
5. Integración de las aplicaciones			
6. Pruebas			
7. Otros			
<b>TOTAL</b>			<b>0</b>

<b>Descripción de la aplicación</b>			
<b>Nombre y Descripción</b>	<p><b>Guía electrónica de programación</b></p> <p>Se requiere desarrollar una aplicación que permita consultar una Guía Electrónica de Programación (EPG), la cual consta de una lista de canales disponibles (como canales de televisión). Cada canal debe tener asociados a videos programado en un período de tiempo. El usuario debe poder consultar la guía para un día determinado y realizar consultas por canales y programas</p>		
<b>Tiempos</b>			
<b>Tarea</b>			<b>Tiempo en Horas</b>
1. Análisis y Diseño			
2. Creación de la base de datos			

3. Implementación de la aplicación servidor	
4. Pruebas	
5. Otros	
<b>TOTAL</b>	<b>0</b>

<b>Descripción de la aplicación</b>	
<b>Nombre y Descripción</b>	<p>Transmisión de Canales (Televisión a través de la red) Se requiere desarrollar una aplicación servidor que consulte una base de datos con una lista de canales y videos programados y realice la transmisión simultánea de estos videos a direcciones multicast.</p> <p>Además, se debe construir una aplicación cliente que sea capaz de escuchar cualquiera de estas direcciones multicast y reproducir el video que está siendo transmitido.</p>
<b>Tiempos</b>	
<b>Tarea</b>	<b>Tiempo en Horas</b>
1. Análisis y Diseño	
2. Creación de la base de datos	
3. Implementación de la aplicación servidor	
4. Implementación de la aplicación cliente	
5. Integración de las aplicaciones	
6. Pruebas	
7. Otros	
<b>TOTAL</b>	<b>0</b>

**Figura 10.1 Formato de encuesta de estimación de tiempos**

Las respuestas entregadas por los tres expertos se anexan al documento.

En la tabla 10.1 se puede apreciar un gran índice de disminución entre el tiempo de desarrollo estimado sin tener en cuenta el framework y el tiempo realmente utilizado, lo que permite darse una idea de la utilidad que puede llegar a tener el mismo.

## 11. Conclusiones

Luego de analizar los resultados en tiempos y esfuerzos utilizados para realizar las tres aplicaciones de prueba y los tiempos previamente estimados, podemos dar por sentado que el framework ofrece las funcionalidades básicas de IPTV y permite realizar aplicaciones de una manera mucho más ágil que si se realizaran cada una independientemente y desde cero.

La utilización de metodologías reconocidas como RUP o los patrones para realizar desarrollos de la magnitud que tiene un framework, facilitan en gran medida la planeación, análisis, diseño y administración de el proyecto.

IPTV puede convertirse en una muy pertinente herramienta para la educación a distancia, con interacción directa de los participantes en una clase a través de un único medio y un único sistema.

Además de facilitar el trabajo y disminuir los esfuerzos de desarrollo de software relacionado con IPTV, se puede valorar del framework, el ser una herramienta integradora, donde todas las aplicaciones que se hagan referente a este tema trabajen bajo el mismo esquema, los mismos objetos de acceso a datos y no se tenga cada aplicación como una “isla” independiente de todas las demás.

## 12. Trabajos futuros

Los siguientes son posibles futuros desarrollos para el framework:

- Integración de servicios como correo electrónico y telefonía IP con el framework.

Una de las principales ventajas que ofrece IPTV es la integración de servicios, como Internet, telefonía, y televisión. En este proyecto nos centramos en el desarrollo de la parte de video, permitiendo interactividad, una de las propuestas de desarrollo posteriores es realizar esta integración con los demás servicios.

- Implementación de cliente para Set Top Box (STB).

Un STB es un dispositivo que permite conectar un televisor tradicional con la red de Internet. Se encarga de la conversión de la televisión y señales digitales en análogas para ser vistas en un televisor, y recibe las señales desde un “control” que maneja el televidente para enviarlas por la red.

En el presente proyecto no se vio como una necesidad de primera mano la implementación de clientes para STB ya que en el ámbito educativo actualmente no se tiene mucho acceso a televisores, además del alto costo que tienen estos dispositivos. Aún así, se propone como un dispositivo que en el futuro podría estar en determinadas salas de la universidad para que los estudiantes accedan a conferencias o clases remotamente y participen activamente por medio de este dispositivo.

## 13. Referencias

- [1] Delivery System for Networked Home Entertainment - Master's Thesis  
Master of Science Engineering in Information Technology - Uppsala  
University – Sweden - Fredrik Andersson - Hans Flack
- [2] <http://www.iptvnews.net/>
- [3] <http://www.terra.es/tecnologia/articulo/html/tec10392.htm>
- [4] <http://www.uexternado.edu.co/noticias/pdf/telecomunicaciones/IPTVExpertGroup.pdf>
- [5] <http://www.tech-faq.com/iptv.shtml>
- [6] <http://www.cinit.org.mx/articulo.php?idArticulo=32>
- [7] <http://www.gestiopolis.com/canales8/ger/convergencia-de-medios-a-traves-de-la-red-triple-play.htm>
- [8] [http://www.tamblin.com/itv\\_iptv\\_portal.html](http://www.tamblin.com/itv_iptv_portal.html)
- [9] <http://www.sistemasdigitales.com.mx>
- [10] <http://www.rational.com>
- [11] <http://java.sun.com>
- [12] <http://www.iaa.upf.es/~xamat/Thesis/html/node2.html>
- [13] [www.coitt.es/antena/pdf/170/15\\_Internet.pdf](http://www.coitt.es/antena/pdf/170/15_Internet.pdf)
- [14] <http://www.cinit.org.mx/articulo.php?idArticulo=34>
- [15] <http://www.monografias.com/trabajos7/protoip/protoip.shtml#FDIP>
- [16] <http://es.wikipedia.org/wiki/Streaming>
- [17] [www.mastermagazine.info/termino/4316.php](http://www.mastermagazine.info/termino/4316.php)
- [18] <http://es.wikipedia.org/wiki/C%C3%B3dec>
- [19] [www.digitalfotored.com/videodigital/denicioncodec.htm](http://www.digitalfotored.com/videodigital/denicioncodec.htm)
- [20] <http://www.alegsa.com.ar/Dic/codec.php>
- [21] <http://www.codeproject.com/gen/design/WhatIsAFramework.asp>
- [22] <http://www.cs.columbia.edu/~hgs/rtsp/faq.html>

- [23] <http://es.wikipedia.org/wiki/Framework>
- [24] [http://geocities.com/intro\\_to\\_multimedia/RTP/index.html](http://geocities.com/intro_to_multimedia/RTP/index.html)
- [25] [http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework)
- [26] [http://geocities.com/intro\\_to\\_multimedia/SIP/index.html](http://geocities.com/intro_to_multimedia/SIP/index.html)
- [27] <http://www.askoxford.com>
- [28] <http://www.networksorcery.com/enp/protocol/rtsp.htm>
- [29] <http://java.sun.com/products/java-media/jmf/1.0/guide/JavaMediaFrame.fm1.html>
- [30] <http://java.sun.com/products/java-media/jmf/reference/faqs/index.html#jmf2-media-streaming>
- [31] <http://java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html>
- [32] <http://java.sun.com/developer/technicalArticles/Programming/singleton/>
- [33] <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>
- [34] <http://www.acm.org/crossroads/xrds5-1/patterns.html>
- [35] <http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/patterns/>
- [36] <http://www.code4net.com/archives/000030.html>
- [37] Implementación del patrón MVC en aplicaciones Web. Pello Xabier Altadill Izura 2003
- [38] Design Patterns Elements of Reusable Object-Oriented Software  
Erich Gamma - Richard Helm - Ralph Johnson - John Vlissides
- [39] <http://www.code4net.com/archives/000030.html>
- [40] <http://www.moock.org/>
- [41] Revista de UTBB - [http://www.utbb.edu.mx/gaceta/frame-tejiendo\\_una\\_red-art.html](http://www.utbb.edu.mx/gaceta/frame-tejiendo_una_red-art.html)
- [42] <http://www.freebsd.org/doc/es/books/handbook/network-dhcp.html>

- [43] [www.dev-mentor.com/Home/tabid/36/EntryID/12/language/es-CL/Default.aspx](http://www.dev-mentor.com/Home/tabid/36/EntryID/12/language/es-CL/Default.aspx)
- [44] [www.videolan.org](http://www.videolan.org)
- [45] <http://www.videolan.org/doc/streaming-howto/en/ch01.html>
- [46] <http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/intro.html#wp725>
- [47] [http://www.babylon.com/definition/Request\\_for\\_Comments/Spanish](http://www.babylon.com/definition/Request_for_Comments/Spanish)
- [48] <http://www.itvdictionary.com/iptv.html>
- [49] <http://www.techweb.com/encyclopedia/defineterm.jhtml?term=IPTV>
- [50] [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=IPTV&i=45410,0\\_0.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=IPTV&i=45410,0_0.asp)
- [51] [http://linux.pucp.edu.pe/downloads/linuxweek2006/lwp-programacion\\_java.pdf](http://linux.pucp.edu.pe/downloads/linuxweek2006/lwp-programacion_java.pdf)
- [52] <http://developer.apple.com/quicktime/qtjava/>
- [53] <http://www.iit.upcomillas.es/pfc/resumenes/467bd4241516a.pdf>
- [54] [http://www.zhenxiao.com/iptv2006/Camera\\_Ready/Jin\\_Liang\\_iptv\\_final.pdf](http://www.zhenxiao.com/iptv2006/Camera_Ready/Jin_Liang_iptv_final.pdf)
- [55] <http://www.joost.com/>

## ANEXOS

**Aleyda M. Hincapié Álvarez**

INTRODUCCION			
<p>El siguiente formato fue definido para realizar una estimación de tiempo de implementación de tres aplicaciones multimedia.</p> <p>En la parte inferior usted encontrará una breve descripción de cada una de las aplicaciones, para un mayor detalle puede consultar a la persona que le entregó la encuesta.</p> <p>Adicionalmente encontrará una lista de tareas que se espera le sirvan de guía para asignar unos tiempos acordes a cada implementación que se planea realizar, si existe alguna actividad que usted piense que no se encuentra en esta lista, le puede asignar el tiempo en el último numeral (Otros).</p>			
Datos Básicos			
<b>Nombre</b>	Aleyda María Hincapié A.	<b>Edad</b>	25
<b>Experiencia</b>	3.5 años	<b>Empresa</b>	Axede S.A (Trébol SW)
Descripción de la aplicación			
<b>Nombre y Descripción</b>	<p>Vídeo Bajo Demanda</p> <p>Se requiere desarrollar una aplicación que permita transmitir videos almacenados en una máquina a través de una red de área local de acuerdo con solicitudes que realice una aplicación cliente en cualquier momento.</p> <p>Estos videos podrán ser consultados y visualizados por la misma aplicación cliente que realiza la solicitud.</p>		
Tiempos			
Tarea			Tiempo en Horas
1. Análisis y Diseño			25
2. Creación de la base de datos			15
3. Implementación de la aplicación servidor			35
4. Implementación de la aplicación cliente			40
5. Integración de las aplicaciones			10
6. Pruebas			30
7. Otros			6
<b>TOTAL</b>			161

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

<b>Descripción de la aplicación</b>	
<b>Nombre y Descripción</b>	<p>Guía electrónica de programación</p> <p>Se requiere desarrollar una aplicación que permita consultar una Guía Electrónica de Programación (EPG), la cual consta de una lista de canales disponibles (como canales de televisión). Cada canal debe tener asociados a videos programado en un período de tiempo. El usuario debe poder consultar la guía para un día determinado y realizar consultas por canales y programas</p>
<b>Tiempos</b>	
<b>Tarea</b>	<b>Tiempo en Horas</b>
1. Análisis y Diseño	<b>4</b>
2. Creación de la base de datos	4
3. Implementación de la aplicación servidor	10
4. Pruebas	5
5. Otros	5
<b>TOTAL</b>	<b>28</b>

<b>Descripción de la aplicación</b>	
<b>Nombre y Descripción</b>	<p>Transmisión de Canales (Televisión a través de la red)</p> <p>Se requiere desarrollar una aplicación servidor que consulte una base de datos con una lista de canales y videos programados y realice la transmisión simultánea de estos videos a direcciones multicast.</p> <p>Además, se debe construir una aplicación cliente que sea capaz de escuchar cualquiera de estas direcciones multicast y reproducir el video que está siendo transmitido.</p>
<b>Tiempos</b>	
<b>Tarea</b>	<b>Tiempo en Horas</b>
1. Análisis y Diseño	<b>35</b>
2. Creación de la base de datos	18
3. Implementación de la aplicación servidor	45
4. Implementación de la aplicación cliente	30
5. Integración de las aplicaciones	25
6. Pruebas	20
7. Otros	15
<b>TOTAL</b>	<b>188</b>

**René A. Rúa**

<b>INTRODUCCION</b>			
<p>El siguiente formato fue definido para realizar una estimación de tiempo de implementación de tres aplicaciones multimedia.</p> <p>En la parte inferior usted encontrará una breve descripción de cada una de las aplicaciones, para un mayor detalle puede consultar a la persona que le entregó la encuesta.</p> <p>Adicionalmente encontrará una lista de tareas que se espera le sirvan de guía para asignar unos tiempos acordes a cada implementación que se planea realizar, si existe alguna actividad que usted piense que no se encuentra en esta lista, le puede asignar el tiempo en el último numeral (Otros).</p>			
<b>Datos Básicos</b>			
<b>Nombre</b>	René Alexander Rúa	<b>Edad</b>	26
<b>Experiencia</b>	3 años - 6 meses	<b>Empresa</b>	PSL
<b>Descripción de la aplicación</b>			
<b>Nombre y Descripción</b>	<p>Video Bajo Demanda</p> <p>Se requiere desarrollar una aplicación que permita transmitir videos almacenados en una máquina a través de una red de área local de acuerdo con solicitudes que realice una aplicación cliente en cualquier momento.</p> <p>Estos videos podrán ser consultados y visualizados por la misma aplicación cliente que realiza la solicitud.</p>		
<b>Tiempos</b>			
<b>Tarea</b>			<b>Tiempo en Horas</b>
1. Análisis y Diseño			<b>20</b>
2. Creación de la base de datos			18
3. Implementación de la aplicación servidor			40
4. Implementación de la aplicación cliente			35
5. Integración de las aplicaciones			8
6. Pruebas			22
7. Otros			8
<b>TOTAL</b>			<b>151</b>

<b>Descripción de la aplicación</b>	
<b>Nombre y Descripción</b>	<p>Guía electrónica de programación</p> <p>Se requiere desarrollar una aplicación que permita consultar una Guía Electrónica de Programación (EPG), la cual consta de una lista de canales disponibles (como canales de televisión). Cada canal debe tener asociados a videos programado en un período de tiempo. El usuario debe poder consultar la guía para un día determinado y realizar consultas por canales y programas</p>
<b>Tiempos</b>	
<b>Tarea</b>	<b>Tiempo en Horas</b>
1. Análisis y Diseño	<b>3</b>
2. Creación de la base de datos	5

Camilo Vieira Mejía  
 Carlos Andrés Valencia Sánchez  
 Juan Diego A. Restrepo Díaz

[cvieiram@eafit.edu.co](mailto:cvieiram@eafit.edu.co)  
[cvalen11@eafit.edu.co](mailto:cvalen11@eafit.edu.co)  
[jrestr52@eafit.edu.co](mailto:jrestr52@eafit.edu.co)

3. Implementación de la aplicación servidor	8
4. Pruebas	5
5. Otros	4
<b>TOTAL</b>	<b>25</b>

<b>Descripción de la aplicación</b>	
<b>Nombre y Descripción</b>	<p>Transmisión de Canales (Televisión a través de la red) Se requiere desarrollar una aplicación servidor que consulte una base de datos con una lista de canales y videos programados y realice la transmisión simultánea de estos videos a direcciones multicast.</p> <p>Además, se debe construir una aplicación cliente que sea capaz de escuchar cualquiera de estas direcciones multicast y reproducir el video que está siendo transmitido.</p>
<b>Tiempos</b>	
<b>Tarea</b>	<b>Tiempo en Horas</b>
1. Análisis y Diseño	<b>30</b>
2. Creación de la base de datos	16
3. Implementación de la aplicación servidor	50
4. Implementación de la aplicación cliente	25
5. Integración de las aplicaciones	25
6. Pruebas	16
7. Otros	16
<b>TOTAL</b>	<b>178</b>

## **Raúl F. Ospina**

<b>INTRODUCCION</b>			
<p>El siguiente formato fue definido para realizar una estimación de tiempo de implementación de tres aplicaciones multimedia.</p> <p>En la parte inferior usted encontrará una breve descripción de cada una de las aplicaciones, para un mayor detalle puede consultar a la persona que le entregó la encuesta.</p> <p>Adicionalmente encontrará una lista de tareas que se espera le sirvan de guía para asignar unos tiempos acordes a cada implementación que se planea realizar, si existe alguna actividad que usted piense que no se encuentra en esta lista, le puede asignar el tiempo en el último numeral (Otros).</p>			
<b>Datos Básicos</b>			
<b>Nombre</b>	Raúl Federico Ospina	<b>Edad</b>	29
<b>Experiencia</b>	6 años	<b>Empresa</b>	PSL
<b>Descripción de la aplicación</b>			

<b>Nombre y Descripción</b>	<p style="text-align: center;">Video Bajo Demanda</p> <p>Se requiere desarrollar una aplicación que permita transmitir videos almacenados en una máquina a través de una red de área local de acuerdo con solicitudes que realice una aplicación cliente en cualquier momento.</p> <p>Estos videos podrán ser consultados y visualizados por la misma aplicación cliente que realiza la solicitud.</p>	
<b>Tiempos</b>		
	<b>Tarea</b>	<b>Tiempo en Horas</b>
	1. Análisis y Diseño	16
	2. Creación de la base de datos	18
	3. Implementación de la aplicación servidor	35
	4. Implementación de la aplicación cliente	30
	5. Integración de las aplicaciones	10
	6. Pruebas	21
	7. Otros	8
	<b>TOTAL</b>	<b>138</b>

<b>Descripción de la aplicación</b>		
<b>Nombre y Descripción</b>	<p style="text-align: center;">Guía electrónica de programación</p> <p>Se requiere desarrollar una aplicación que permita consultar una Guía Electrónica de Programación (EPG), la cual consta de una lista de canales disponibles (como canales de televisión). Cada canal debe tener asociados a videos programado en un período de tiempo. El usuario debe poder consultar la guía para un día determinado y realizar consultas por canales y programas</p>	
<b>Tiempos</b>		
	<b>Tarea</b>	<b>Tiempo en Horas</b>
	1. Análisis y Diseño	4
	2. Creación de la base de datos	5
	3. Implementación de la aplicación servidor	8
	4. Pruebas	6
	5. Otros	0
	<b>TOTAL</b>	<b>23</b>

<b>Descripción de la aplicación</b>		
<b>Nombre y Descripción</b>	<p style="text-align: center;">Transmisión de Canales (Televisión a través de la red)</p> <p>Se requiere desarrollar una aplicación servidor que consulte una base de datos con una lista de canales y videos programados y realice la transmisión simultánea de estos videos a direcciones multicast.</p> <p>Además, se debe construir una aplicación cliente que sea capaz de escuchar cualquiera de estas direcciones multicast y reproducir el video que está siendo transmitido.</p>	
<b>Tiempos</b>		
	<b>Tarea</b>	<b>Tiempo en Horas</b>

1. Análisis y Diseño	<b>32</b>
2. Creación de la base de datos	20
3. Implementación de la aplicación servidor	44
4. Implementación de la aplicación cliente	27
5. Integración de las aplicaciones	23
6. Pruebas	20
7. Otros	8
<b>TOTAL</b>	<b>174</b>