

Plataforma de Control en el Ámbito de la Televisión Digital Interactiva.



LENIN DAVID GÓMEZ MERA.
ANDRÉS GUERRERO ROSERO.

Trabajo de grado presentado como requisito para optar al título de
Ingeniero en Electrónica y en Telecomunicaciones.

Directora

MARY CRISTINA CARRASCAL REYES.
Ingeniera en Electrónica y en Telecomunicaciones.

Universidad del Cauca.
Facultad de Ingeniería Electrónica y Telecomunicaciones.
Departamento de Telemática.
Servicios Avanzados de Telecomunicaciones.
Popayán, Septiembre de 2010.



AGRADECIMIENTOS

A Dios y a la Virgen María, agradecido eternamente por estar a mi lado en todo momento, por iluminar mi mente y darme la fuerza necesaria para alcanzar este título.

A mi Mamita Mariela y a mi Papá Albertino, a quienes les debo todo lo que tengo en mi vida, faltarían palabras para hacerles saber lo mucho que los quiero.

A José Raúl, Everth y Edwin, que hasta el día de hoy están pendientes de mí. A pesar de la distancia, seguimos siendo los mejores hermanos.

A mis sobrinos, Juan Daniel y Sofía quienes me regalan su compañía en los momentos más inesperados.

A mis compañeros de Judo, en especial al profesor Alejandro Solano, quien nos formó desde muy pequeños y nos enseñó los mejores valores que un ser humano puede tener.

A la ingeniera Mary Cristina Carrasca, por su ayuda, apoyo y por aguantarme tanto.

A mis compañeros de lucha, por todo este tiempo, por los momentos que pasaron, por toda su ayuda, solo quiero expresarles mi gratitud por permitirme conocerlos y por toda la ayuda que me brindaron.

Por último, a las personas que no están aquí en este momento, que por alguna razón nos tomaron una vida de ventaja y que querían ser parte de este momento, quiero que sepan que aún no las he olvidado y quiero dedicarles este gran paso en mi vida, algún día nos volveremos a ver.

Lenin David Gómez Mera.



TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPITULO 1. GENERALIDADES PARA LA IMPLEMENTACIÓN DE SERVICIOS EN EL ÁMBITO DE LA TELEVISIÓN DIGITAL INTERACTIVA	5
1.1 TELEVISIÓN DIGITAL	5
1.1.1 <i>Ventajas de la Televisión Digital</i>	5
1.2 TELEVISIÓN DIGITAL INTERACTIVA	6
1.2.1 <i>Ventajas de la TDi</i>	8
1.3 ESTÁNDARES PARA TELEVISIÓN DIGITAL	8
1.3.1 <i>ATSC (Advance Television System Committee)</i>	8
1.3.2 <i>DVB (Digital Video Broadcasting)</i>	9
1.3.3 <i>ISDB (Integrated Service Digital Broadcasting)</i>	9
1.3.4 <i>DTMB (Digital Terrestrial Multimedia Broadcasting)</i>	10
1.4 MEDIADORES DE TELEVISIÓN DIGITAL.....	11
1.4.1 <i>Tipos de Mediadores</i>	11
1.4.2 <i>Sistemas Mediadores</i>	12
1.4.3 <i>Modelos de Aplicaciones</i>	16
1.5 REQUISITOS MÍNIMOS PARA EL DESPLIEGUE DE APLICACIONES EN EL SISTEMA DE TELEVISIÓN DIGITAL INTERACTIVA.....	18
1.6 RESUMEN DE LOS ELEMENTOS MÍNIMOS PARA EL DESPLIEGUE DE APLICACIONES EN EL SISTEMA DE TELEVISIÓN DIGITAL INTERACTIVA.....	27
CAPITULO 2. EXPLORACIÓN DE ARQUITECTURAS DE RED DE TELECOMUNICACIONES, PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS DE TELECOMUNICACIONES Y PARA EL ENTORNO DE LA TDI APLICABLES AL PROYECTO	28
2.1 ARQUITECTURAS DE RED DE TELECOMUNICACIONES Y PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS.....	29
2.1.1 <i>RED INTELIGENTE (Intelligent Network - IN)</i>	29
2.1.2 <i>CAMEL (CUSTOMIZED APPLICATIONS FOR MOBILE NETWORK ENHANCED LOGIC)</i>	34
2.1.3 <i>OSA - OPEN SERVICE ARCHITECTURE (ARQUITECTURA DE SERVICIOS ABIERTOS)</i>	38
2.1.4 <i>PARLAY Y PARLAY/OSA</i>	40
2.1.4 <i>JAIN – (Java APIs for Integrated Networks)</i>	44
2.1.5 <i>IMS - IP Multimedia Subsystem</i>	46
2.2 HERRAMIENTAS DE AUTORÍA Y PLATAFORMAS PARA DISTRIBUCIÓN DE CONTENIDOS MHP EN EL ENTORNO DE LA TDI.....	52
2.2.1 <i>HERRAMIENTAS DE AUTORÍA MHP PARA LA GENERACIÓN DE CONTENIDOS</i>	53
2.2.2 <i>PLATAFORMAS MHP PARA LA DISTRIBUCIÓN DE CONTENIDOS</i>	55
2.3 OBSERVACIONES Y CONCLUSIONES FINALES.....	57
2.3.1 <i>CONCLUSIONES Y OBSERVACIONES RESULTADO DE LA EXPLORACIÓN A LAS ARQUITECTURAS DE RED DE TELECOMUNICACIONES Y A LAS PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS DE TELECOMUNICACIONES</i>	57



2.3.2	CONCLUSIONES Y OBSERVACIONES RESULTADO DE LA EXPLORACIÓN DE LAS HERRAMIENTAS DE AUTORÍA Y PLATAFORMAS PARA LA DISTRIBUCIÓN DE CONTENIDOS EN MHP	58
2.3.3	COMPARACIÓN DE LAS TECNOLOGÍAS EXPLORADAS.....	59
CAPITULO 3. DISEÑO DE LA ARQUITECTURA PARA LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA TDI.....		60
3.1	PANORAMA GENERAL DEL PROBLEMA PROPUESTO	60
3.1.1	REQUISITOS PARA EL DISEÑO DE LA ARQUITECTURA.....	60
3.2	DESCRIPCIÓN DE LAS FUNCIONES QUE OFRECE LA ARQUITECTURA DE LA PLATAFORMA DE CONTROL PROPUESTA.....	61
3.2.1	Diagrama de casos de uso para el diseño de la arquitectura de la plataforma de control.	61
3.2.2	Descripción de escenarios de los casos de uso.....	63
3.3	PLANTEAMIENTO INICIAL DE LA SOLUCIÓN PARA EL DISEÑO DE LA ARQUITECTURA	69
3.4	ELECCIÓN DE LAS TECNOLOGÍAS A USAR EN EL DISEÑO DE LA PLATAFORMA DE CONTROL	70
3.4.1	ELECCIÓN DE LA ARQUITECTURA DE RED DE TELECOMUNICACIONES.....	71
3.4.2	ELECCIÓN DE LAS TECNOLOGÍAS PARA EL ENTORNO DE LA TDi	74
3.4.3	RESUMEN DE LAS TECNOLOGÍAS ELEGIDAS.....	75
3.5	ANÁLISIS DE LOS ENTORNOS Y ELEMENTOS QUE COMPONEN LA ARQUITECTURA.	75
3.6	ELECCIÓN DE LAS HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE LA ARQUITECTURA DISEÑADA.....	76
3.6.1	Herramientas para la construcción de la aplicación en el sistema de TDi.	76
3.6.2	Herramientas para el despliegue de la aplicación DVB-J.....	77
3.6.3	Herramienta que implementa la arquitectura de red de telecomunicaciones.....	78
3.7.	RESUMEN DE LAS HERRAMIENTAS A UTILIZAR EN LA IMPLEMENTACIÓN DE LA ARQUITECTURA	81
3.8	ARQUITECTURA PROPUESTA PARA LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA TDi.	82
3.8.1	PARA EL ENTORNO DE LA TELEVISIÓN DIGITAL INTERACTIVA	82
3.8.2	PARA EL ENTORNO DE LA ARQUITECTURA DE RED DE TELECOMUNICACIONES IMS.....	84
3.8.3	ARQUITECTURA DE LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA TELEVISIÓN DIGITAL INTERACTIVA.....	85
CAPITULO 4. IMPLEMENTACIÓN DE LA ARQUITECTURA PARA EL SOPORTE DE SERVICIOS EN EL SISTEMA DE TDI.....		87
4.1	DISEÑO E IMPLEMENTACIÓN DE LA PLATAFORMA DE CONTROL EN EL ENTORNO DE LA TDi.. ..	87
4.1.1	Descripción detallada de la arquitectura base.....	88
4.1.2	Diagrama de paquetes para la arquitectura propuesta	93
4.1.3	Diagrama de clases de la arquitectura propuesta	95
4.1.4	Diagrama de secuencia para la arquitectura propuesta.	96
4.2	PUESTA EN MARCHA Y VERIFICACIÓN DEL FUNCIONAMIENTO DE LA PLATAFORMA PROPUESTA	101
CAPITULO 5. VERIFICACIÓN DE LA PLATAFORMA DISEÑADA MEDIANTE EL DESPLIEGUE DEL PROTOTIPO IMPLEMENTADO.....		105
5.1	DESCRIPCIÓN DEL ENTORNO DE EJECUCIÓN DEL PROTOTIPO.....	105



5.2 DIAGRAMA DE PAQUETES DEL PROTOTIPO IMPLEMENTADO.....	108
5.2.1 Paquete Tesis.....	109
5.2.2 Paquete PhantomClient.....	110
5.2.3 Paquete SDS.....	110
5.3 DIAGRAMA DE CLASES DEL PROTOTIPO IMPLEMENTADO	110
5.3.1 Diagrama de clases para el paquete Tesis.....	111
5.3.2 Diagrama de clases para el paquete PhantomClient.....	112
5.3.3 Diagrama de clases para el paquete zPlacaServletH116.....	114
5.2 DIAGRAMA DE SECUENCIA DEL SERVICIO DE CHAT IMPLEMENTADO EN EL PROTOTIPO..	115
5.3 VERIFICACIÓN DEL PROTOTIPO DESPLEGADO SOBRE LA PLATAFORMA DE CONTROL PROPUESTA.	117
CAPITULO 6. APORTES, CONCLUSIONES Y TRABAJOS FUTUROS.....	118
6.1 APORTES	118
6.1.1 Aportes arquitectónicos.....	118
6.1.2 Aportes técnicos.....	118
6.1.3 Aporte social.....	119
6.2 CONCLUSIONES.....	119
6.3 TRABAJOS FUTUROS	121
BIBLIOGRAFÍA.....	122



LISTA DE FIGURAS

<i>Fig. 1 Arquitectura de la Televisión Digital Interactiva.....</i>	<i>6</i>
<i>Fig. 2 Estructura Simplificada de MHP.</i>	<i>15</i>
<i>Fig. 3 Relación entre Perfiles de DVB-MHP.....</i>	<i>16</i>
<i>Fig. 4 Arquitectura de la Plataforma MHP.....</i>	<i>16</i>
<i>Fig. 5. Lenguajes que intervienen en el modelo DVB-HTML.....</i>	<i>18</i>
<i>Fig. 6. Modelo de Referencia end to end para el estándar MHP.....</i>	<i>20</i>
<i>Fig. 7. Diagrama de Despliegue del sistema de TDi para el proyecto EDiTV.....</i>	<i>23</i>
<i>Fig. 8 Planos gráficos en MHP.....</i>	<i>26</i>
<i>Fig. 9. Red Inteligente como una plataforma universal de servicios.</i>	<i>29</i>
<i>Fig. 10. Funciones de la red inteligente y relaciones funcionales para el CS-1.....</i>	<i>31</i>
<i>Fig. 11. Creación de servicios para la red inteligente.</i>	<i>34</i>
<i>Fig. 12. Estructura general CAMEL fase 1.....</i>	<i>36</i>
<i>Fig. 13. Arquitectura OSA.</i>	<i>39</i>
<i>Fig. 14. Interfaz de Programación y Aplicación OSA/PARLAY.</i>	<i>41</i>
<i>Fig. 15. Arquitectura de OSA/PARLAY.</i>	<i>43</i>
<i>Fig. 16. Interfaz de Programación y Aplicación JAIN.</i>	<i>46</i>
<i>Fig. 17. Arquitectura IMS y capas de red según NGN.....</i>	<i>49</i>
<i>Fig. 18. Casos de uso para el diseño de la arquitectura de la plataforma de control.....</i>	<i>62</i>
<i>Fig. 19. Primera solución aproximada a nivel general.....</i>	<i>70</i>
<i>Fig. 20. Arquitectura definida por OpenIMSCore.</i>	<i>78</i>
<i>Fig. 21. Arquitectura definida por SDS de Ericsson.....</i>	<i>80</i>
<i>Fig. 22. Arquitectura propuesta por el laboratorio experimental de TDi del proyecto EDiTV.....</i>	<i>83</i>
<i>Fig. 23. Arquitectura propuesta para la arquitectura de red de telecomunicaciones IMS.....</i>	<i>84</i>
<i>Fig. 24. Arquitectura propuesta para la plataforma de control en el ámbito de la TDi.....</i>	<i>85</i>
<i>Fig. 25. Arquitectura general para el soporte de servicios en el entorno de la TDi.</i>	<i>87</i>
<i>Fig. 26. Diseño detallado de la arquitectura base para el soporte de servicios en el entorno de la TDi.</i>	<i>89</i>
<i>Fig. 27. Diagrama de paquetes para la arquitectura propuesta.....</i>	<i>93</i>
<i>Fig. 28. Diagrama de clases para la arquitectura propuesta.....</i>	<i>95</i>
<i>Fig. 29 Diagrama de secuencia del diseño de la arquitectura base para el soporte de servicios en el sistema TDi.</i>	<i>98</i>
<i>Fig. 30 Diagrama de secuencia del diseño de la arquitectura base para el soporte de servicios en el sistema TDi.....</i>	<i>100</i>
<i>Fig. 31 Ejecución del servidor CSCF.....</i>	<i>101</i>
<i>Fig. 32. Historial correspondiente al servidor CSCF.</i>	<i>101</i>
<i>Fig. 33. Inicio del servicio DNS.....</i>	<i>102</i>
<i>Fig. 34. Historial relacionado con el servicio DNS.....</i>	<i>102</i>
<i>Fig. 35. Ejecución del Servidor de Aplicaciones Sailfin.</i>	<i>103</i>
<i>Fig. 36. Inicio del servidor placa.</i>	<i>103</i>
<i>Fig. 37. Historial durante la ejecución del servidor placa.</i>	<i>104</i>
<i>Fig. 38 Visión general del servicio de presencia.</i>	<i>106</i>



<i>Fig. 39 Diagrama de paquetes para el prototipo implementado.</i>	<i>109</i>
<i>Fig.40 Diagrama de clases para el paquete Tesis.</i>	<i>111</i>
<i>Fig.41 Diagrama de clases para el paquete PhantomClient</i>	<i>112</i>
<i>Fig.42 Diagrama de clases para el paquete PhantomClient</i>	<i>113</i>
<i>Fig.43 Diagrama de clases para el paquete PhantomClient</i>	<i>114</i>
<i>Fig. 44. Diagrama de secuencia del servicio de Chat implementado para el prototipo.....</i>	<i>115</i>



LISTA DE TABLAS

<i>Tabla 1. Resumen Técnico Estándares para Televisión Digital.....</i>	<i>11</i>
<i>Tabla 2. Algunos middleware de tipo privativos como abiertos.</i>	<i>12</i>
<i>Tabla 3 Resumen de los elementos mínimos para el despliegue de aplicaciones en el sistema de TDi.</i>	<i>27</i>
<i>Tabla 4. Resume fases CAMEL.</i>	<i>38</i>
<i>Tabla 5. Comparación de las plataformas para distribución de contenidos exploradas.</i>	<i>56</i>
<i>Tabla 6. Similitudes entre las arquitecturas de red de telecomunicaciones exploradas.....</i>	<i>58</i>
<i>Tabla 7. Cuadro comparativo de las tecnologías exploradas.....</i>	<i>59</i>
<i>Tabla 8. Descripción del escenario para el caso de uso: Iniciar aplicación.</i>	<i>63</i>
<i>Tabla 9. Descripción del escenario para el caso de uso: Cargar contenido.....</i>	<i>64</i>
<i>Tabla 10. Descripción del escenario para el caso de uso: Cargar Servicio.</i>	<i>65</i>
<i>Tabla 11. Descripción del escenario para el caso de uso: Cargar información de usuario.....</i>	<i>65</i>
<i>Tabla 12. Descripción del escenario para el caso de uso: Cargar información de servicio.</i>	<i>66</i>
<i>Tabla 13. Descripción del escenario para el caso de uso: Controlar servicio.</i>	<i>66</i>
<i>Tabla 14 Descripción del escenario para el caso de uso: Controlar recurso TDi.....</i>	<i>67</i>
<i>Tabla 15. Descripción del escenario para el caso de uso: Controlar aplicación TDi.</i>	<i>68</i>
<i>Tabla 16. Descripción del escenario para el caso de uso: Desplegar aplicación.</i>	<i>68</i>
<i>Tabla 17. Descripción del escenario para el caso de uso: Controlar recurso.</i>	<i>69</i>
<i>Tabla 18. Descripción del escenario para el caso de uso: Controlar aplicación.</i>	<i>69</i>
<i>Tabla 19. Comparación de las arquitecturas de red de telecomunicaciones frente a la comunicación con el terminal de usuario.</i>	<i>72</i>
<i>Tabla 20. Comparación de las arquitecturas de red de telecomunicaciones frente al ambiente para el despliegue de servicios y aplicaciones.</i>	<i>72</i>
<i>Tabla 21. Comparación de las arquitecturas de red de telecomunicaciones frente mecanismo para el control de sesiones, servicios y aplicaciones.....</i>	<i>73</i>
<i>Tabla 22. Comparación de las arquitecturas de red de telecomunicaciones frente al disponibilidad de aplicaciones y servicios.....</i>	<i>73</i>
<i>Tabla 23. Comparación de las arquitecturas de red de telecomunicaciones frente a uso de protocolos flexibles.....</i>	<i>73</i>
<i>Tabla 24. Comparación de las arquitecturas de red de telecomunicaciones frente a reutilización de elementos.</i>	<i>73</i>
<i>Tabla 25. Valoración final de las arquitecturas frente a los aspectos técnicos.....</i>	<i>74</i>
<i>Tabla 26. Tecnologías elegidas para el diseño de la arquitectura.....</i>	<i>75</i>
<i>Tabla 27. Comparación entre DVB-J y DVB-HTML.....</i>	<i>77</i>
<i>Tabla 28. Herramientas a usar en la implementación de la arquitectura diseñada.....</i>	<i>82</i>



ACRÓNIMOS

AIT:	<i>(Application Information Table.)</i> Tabla de información de aplicaciones
API:	<i>(Application Programming Interface).</i> Interfaz de programación de aplicaciones.
ARIB:	<i>(Association of Radio Industries and Businesses).</i> Asociación de Industrial y Negocios de Radio.
ATSC:	<i>(Advanced Television Systems Committee).</i> Comité de Sistemas de Televisión Avanzada
AWT:	<i>(Abstract Window Toolkit).</i> Kit de Herramientas de Ventana Abstracta.
CAMEL:	<i>(Customized Applications for Mobile Network Enhanced Logic)</i>
CPU:	<i>(Central Processing Unit).</i> Unidad central de procesamiento.
CSS:	<i>(Cascading Style Sheets).</i> Hojas de Estilo en Cascada.
DASE:	<i>(DTV Applications Software Environment).</i> Ambiente de software de aplicación DTV
DAVIC:	<i>(Digital Audio Video Council).</i> Concilio Digital Audiovisual.
DIBEG:	<i>(The Digital Broadcasting Experts Group).</i> Grupo de Expertos de radiodifusión digital.
DMB-T:	<i>(Digital Multimedia Broadcast – Terrestrial).</i> Radiodifusión multimedia Digital – Terrestre.
DOM:	<i>(Document Object Model).</i> Modelo de Objetos de Documento.
DSM-CC:	<i>(Digital Storage Media - Command and Control).</i> Medio de almacenamiento digital – Comando y Control.
DTMB:	<i>(Digital Terrestrial Multimedia Broadcasting)</i>
DTV:	<i>(Digital Television)</i> televisión digital
DVB:	<i>(Digital Video Broadcasting).</i> Emisión de Video Digital.
DVB-C:	<i>(DVB- Cable).</i> Transmisión de vídeo digital para Cable.
DVB-H:	<i>(DVB-HandHeld).</i> Transmisión de vídeo digital para dispositivos móviles.
DVB-S:	<i>(DVB-Satelital).</i> Transmisión de vídeo digital Satelital.
DVB-T:	<i>(DVB-Terrestrial).</i> Transmisión de vídeo digital Terrestre.
ECMA:	<i>(European Computer Manufacturers Association).</i> Asociación Europea de Fabricantes de Computadores.
EDiTV:	<i>(Educación Virtual Basada en Televisión Interactiva para Apoyar Procesos Educativos a Distancia)</i>
ECMAScript:	<i>(lenguaje ECMA-262)</i> Especificación de lenguaje de programación publicado por ECMA
EPG:	<i>(Electronic Program Guide).</i> Guías Electrónicas de Programas.
ES:	<i>(Elementary Stream).</i> Flujo Elemental.
ETSI:	<i>(European Telecommunications Standards Institute).</i> Instituto Europeo de Estándares de Telecomunicaciones.
GEM:	<i>(Globally Executable MHP).</i> MHP Globalmente Ejecutable.
GSM:	<i>(Groupe Spécial Mobile).</i> Sistema Global para las Comunicaciones Móviles.
GUI:	<i>(Graphis User Interface).</i> Interfaz Gráfica de Usuario.
HAVI:	<i>(Home Audio/Video Interoperability).</i> Interoperabilidad de dispositivos de audio/video.



HD:	<i>(High Definition). Alta Definición.</i>
HDTV:	<i>(High Definition Television). Televisión de Alta Definición.</i>
ICP:	<i>(IMS Client Platform)</i>
IEEE:	<i>(Institute of Electrical and Electronics Engineers). Instituto de Ingenieros Eléctricos y Electrónicos.</i>
IN:	<i>(Intelligent Network). Red Inteligente.</i>
IPTV:	<i>(Internet Protocol Televisión). Televisión sobre IP.</i>
ISDB:	<i>(Integrated Services Digital Broadcasting). Emisión Digital de Servicios Integrados.</i>
ISDN:	<i>(Integrated Services Digital Network). Red digital de servicios integrados.</i>
iTV:	<i>(Interactive Television). Televisión interactiva.</i>
IMS:	<i>(IP Multimedia Subsystem). Sistema IP multimedia.</i>
JAIN:	<i>(Java APIs for Integrated Networks)</i>
JVM:	<i>(Java Virtual Machine). Máquina Virtual Java.</i>
MHEG:	<i>(Multimedia and Hypermedia Experts Group). Grupo de Expertos en Multimedia e Hipermedia.</i>
MHP:	<i>(Multimedia Home Platform). Plataforma de Hogar Multimedia.</i>
MPEG:	<i>(Moving Pictures Experts Group). Grupo de expertos en imágenes en movimiento.</i>
NGN:	<i>(Next Generation Network)</i>
NTSC:	<i>(National Television System Committee). Comisión Nacional de Sistemas de Televisión.</i>
OCAP:	<i>(OpenCable Applications Platform). Plataforma de aplicaciones OpenCable.</i>
OSA:	<i>(Open Services Access). Acceso a servicios abiertos.</i>
PAL:	<i>(Phase Alternating Line). Línea alternada en fase.</i>
PID:	<i>(Packet Identifier). Identificador único de paquete.</i>
PSI:	<i>(Program Specific Information). Información Específica de Programa.</i>
PSTN:	<i>(Public Switched Telephone Network). Red telefónica pública conmutada.</i>
QAM:	<i>(Quadrature Amplitude Modulation). Modulación de amplitud en cuadratura.</i>
QPSK:	<i>(Quadrature Phase Shift Keying). Modulación por desplazamiento de fase de 4 símbolos.</i>
RAM:	<i>(Random Access Memory). Memoria de acceso aleatorio.</i>
RGB :	<i>(Red Green Blue). Sistema de señal de vídeo que utiliza la señal de rojo verde y azul por separado.</i>
SDS:	<i>(Service Development Studio).</i>
SI:	<i>(Service Information). Información de servicio</i>
SPI:	<i>(Serial Peripheral Interface). Interfaz Serial de Periféricos.</i>
STB:	<i>(Set Top Box)</i>
TDi:	<i>(Televisión Digital Interactiva.)</i>
TS:	<i>(Transport Stream). Flujo de transporte.</i>
UMTS:	<i>(Universal Mobile Telecommunications System). Sistema Universal de Telecomunicaciones móviles.</i>
W3C:	<i>(World Wide Web Consortium) Consorcio WWW.</i>
XHTML:	<i>(eXtensible Hypertext Markup Language). Lenguaje de marcas hipertexto ampliable.</i>
XML:	<i>(Extensible Markup Language). Lenguaje de marcas ampliable.</i>





INTRODUCCIÓN

Aunque hoy en día existan nuevos y poderosos medios de comunicación, la televisión es un elemento que a través de los años ha permanecido, se ha adaptado y ha logrado posicionarse como una muy buena opción que brinda entretenimiento e información a gran cantidad de personas a nivel mundial.

Para lograr esto, el paradigma impuesto por el modelo de negocio que ofrece la televisión tuvo que sufrir una serie de cambios de acuerdo con los nuevos requisitos y exigencias por parte del cliente, de nuevas tecnologías de telecomunicaciones, a la convergencia de redes y servicios y además de otros factores de gran influencia como la globalización y la economía.

Uno de los cambios introducidos a la emisión de Televisión es la interactividad, complemento a los contenidos que se presentan donde se abre la posibilidad de ofrecer un amplio conjunto de servicios que hasta el momento solo eran disponibles a través de otros medios como el computador o el teléfono.

Una de las ventajas resultado de la inclusión de la interactividad, es la posibilidad de acceder a un amplio conjunto de servicios mediante un terminal y un mando a distancia, permitiendo al usuario decidir a cuál de los servicios y a qué tipo de contenidos quiere acceder. Para el futuro, los usuarios de la Televisión Digital Interactiva (*TDi*) podrán disponer de servicios que incluyen características nunca vistas en la televisión analógica: servicios personalizados de acuerdo a sus preferencias, servicios en formatos en alta definición y accesibles desde cualquier dispositivo como lo son televisores, teléfonos móviles, computadoras entre otros.

Por otra parte, la existencia de arquitecturas de red de telecomunicaciones dedicadas al control y despliegue de servicios y aplicaciones en áreas como la telefonía fija, telefonía celular, entornos combinados entre otros, donde estos servicios están disponibles para terminales con las capacidades y recursos necesarios para su ejecución además, de estar conectados a la red de acceso adecuada para acceder tanto a la arquitectura de red de telecomunicaciones como a los servicios que ofrece. A esto se suma, el aumento de la velocidad en la transmisión de datos, surgimiento de terminales inteligentes, Redes de Próxima Generación (*Next Generation Network – NGN*), entre otros aspectos, se necesita de un cambio radical en el sector de las telecomunicaciones donde se busca sobre un entorno complejo y competitivo, brindar convergencia de redes y servicios y ofrecer paquetes de servicios atractivos e innovadores a bajo costo.

Con todo lo anterior es necesario pensar en una arquitectura de red de telecomunicaciones que integre servicios de voz, video y datos además, de la integración de redes fijas y móviles con Internet mediante un conjunto de mecanismos que permitan el control, tarificación y demás procesos necesarios para estos escenarios.

Para el anteproyecto de grado, se mencionó el problema existente en la implementación de servicios sobre el sistema *TDi*, la cual se hace de manera vertical lo que conlleva a poca escalabilidad de las soluciones, problemas de compatibilidad y falta de flexibilidad,



impidiendo tener servicios eficientes que aprovechen de la mejor manera los recursos que ofrece la red.

Por esta razón, se presenta el diseño de una arquitectura horizontal con base en la arquitectura de red de telecomunicaciones *IMS (IP Multimedia Subsystem)*, común con el sistema de *TDi*, la cual brinda un punto de referencia en cuanto a la implementación y control de los recursos de red que el servicio necesita, partiendo de la concepción de que *IMS* presenta un número de funciones en común que pueden ser reutilizados por todos los nuevos servicios que están dentro o fuera de la red, sin importar el operador o la tecnología de acceso que el cliente usa.

Una vez diseñada e implementada la arquitectura para la plataforma de control, se procede con la verificación del funcionamiento de esta a través del despliegue de un prototipo. Este prototipo, implementa los servicios de *Chat* y de *Presencia*, además de algunas funciones que se ejecutan dentro del entorno de *IMS* las cuales se adaptaron de manera satisfactoria para estar a disposición de la aplicación que se ejecuta en el sistema de *TDi*.

El trabajo de grado tiene como alcance, mostrar la adaptación de una arquitectura de red de telecomunicaciones, en este caso *IMS*, que ofrece servicios como aplicaciones, además de otros mecanismos que pueden ser utilizados por la aplicación en el sistema de la *TDi* para su funcionamiento.

Al finalizar, se presenta la arquitectura detallada de la plataforma de control en el ámbito de la Televisión Digital Interactiva junto a la implementación de un prototipo desplegado el cual valida el correcto funcionamiento de la arquitectura.

Para la elaboración de este documento, se tuvieron en cuenta dos trabajos de grado enmarcados en el proyecto *EDiTV*¹ (Educación Virtual Basada en Televisión Interactiva para Apoyar Procesos Educativos a Distancia), los cuales fueron de gran ayuda y de apoyo en la construcción de la base teórica para el área relacionada con el entorno de Televisión Digital Interactiva.

- *“Recomendaciones para la generación y distribución de contenidos educativos orientados a Televisión Digital Interactiva” [15].*
- *“Arquitectura Básica de un Navegador DVB-HTML para Múltiples Terminales” [3].*

Para finalizar, este documento está organizado de acuerdo a los objetivos planteados en el anteproyecto de grado, es así que inicialmente se hace una exploración general en cuanto al sistema de *TDi* y algunas arquitecturas de red de telecomunicaciones, plataformas para la creación de servicios de telecomunicaciones y para el entorno de la *TDi*. Luego se hace la elección de las tecnologías que hacen parte en la arquitectura de la plataforma de control propuesta y finalmente se tiene el prototipo desplegado sobre la arquitectura que verifica el correcto funcionamiento de esta.

¹ Sitio Web del proyecto *EDiTV*: www.unicauca.edu.co/EDiTV



- **CAPITULO 1. GENERALIDADES PARA LA IMPLEMENTACIÓN DE SERVICIOS EN EL ÁMBITO DE LA TELEVISIÓN DIGITAL INTERACTIVA**

Este capítulo presenta algunos conceptos básicos relacionados con la Televisión Digital y la Televisión Digital Interactiva, sus estándares y los respectivos mediadores que define cada especificación. Al final se hace mención de los requisitos mínimos para el despliegue de servicios sobre este entorno.

- **CAPITULO 2. EXPLORACIÓN DE ARQUITECTURAS DE RED DE TELECOMUNICACIONES, PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS DE TELECOMUNICACIONES Y PARA EL ENTORNO DE LA TDi APLICABLES AL PROYECTO**

Este capítulo presenta la exploración realizada a algunas arquitecturas de red de telecomunicaciones, plataformas para la creación de servicios de telecomunicaciones y para el entorno de la *TDi*. Al final se presentan las conclusiones, observaciones y un cuadro comparativo resultado de esta exploración.

- **CAPITULO 3. DISEÑO DE LA ARQUITECTURA PARA LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA TDi**

Se presenta los requisitos del problema propuesto y posteriormente se presenta la solución general para este inconveniente. Además, se eligen las tecnologías que componen el diseño de la arquitectura para la plataforma de control y se finaliza con un diagrama que presenta con más detalle la solución a la cual se quiere llegar.

- **CAPITULO 4. IMPLEMENTACIÓN DE LA ARQUITECTURA PARA EL SOPORTE DE SERVICIOS EN EL SISTEMA DE TDi**

Una vez definidos los elementos que componen el diseño de la arquitectura, este capítulo presenta la implementación realizada de dicha arquitectura propuesta y además verifica el correcto funcionamiento de esta mediante la puesta en marcha de todos los elementos que la componen.

- **CAPITULO 5. VERIFICACIÓN DEL PROTOTIPO DESPLEGADO SOBRE LA ARQUITECTURA DE LA PLATAFORMA PROPUESTA**

Se describe y se presenta el prototipo implementado para verificar la adaptación realizada entre el entorno de *IMS* y el sistema de *TDi*, posteriormente se presentan los diferentes resultados obtenidos por el despliegue del prototipo sobre la plataforma propuesta.

- **CAPITULO 6. APORTES, CONCLUSIONES Y TRABAJOS FUTUROS**

Se presentan los aportes, conclusiones y trabajos futuros producto del trabajo de grado realizado.



- **ANEXO A. MODELO DE APLICACIÓN DVB-J**

Se presenta una breve descripción acerca del modelo de aplicación definido por el estándar Europeo *DVB*.

- **ANEXO B. DESCRIPCIÓN DEL PROCEDIMIENTO DE REGISTRO DE USUARIO EN *IMS*.**

Muestra el procedimiento de registro a realizar un usuario ante *IMS* para acceder a los servicios y recursos que esta le pueda ofrecer.

- **ANEXO C. CONFIGURACIÓN DE LA HERRAMIENTA *SDS* DE *ERICSSON*.**

Presenta la configuración realizada a la solución *SDS* de Ericsson para este proyecto.

- **ANEXO D. VERIFICACIÓN DEL PROTOTIPO DESPLEGADO SOBRE LA PLATAFORMA DE CONTROL PROPUESTA**

Muestra el despliegue del prototipo sobre la arquitectura planteada.



CAPITULO 1. GENERALIDADES PARA LA IMPLEMENTACIÓN DE SERVICIOS EN EL ÁMBITO DE LA TELEVISIÓN DIGITAL INTERACTIVA

Este capítulo presenta una breve descripción de los conceptos y definiciones básicas concernientes a la Televisión Digital y la Televisión Digital Interactiva con lo cual se busca ofrecer una visión muy general acerca de estos entornos además, expone los requisitos mínimos tanto a nivel *software* como a nivel *hardware* para la implementación y despliegue de servicios sobre este sistema.

1.1 TELEVISIÓN DIGITAL

La Televisión Digital es una tecnología que permite a los operadores la transmisión de los contenidos de Televisión con gran calidad tanto de audio como de video además, de información adicional que será recibida por el usuario [1].

En este momento, la emisión analógica de los contenidos de Televisión es la manera como se transporta la señal hasta todos los hogares en Colombia, como consecuencia se hace uso de demasiados recursos en especial del ancho de banda del espectro electromagnético [2].

Con la llegada de la digitalización de la Televisión, los parámetros analógicos en el sistema de televisión son representados por un sistema binario, es decir se cuantifica una señal analógica mediante el uso de los dígitos 0 y 1. Esta digitalización se realiza usando equipos para comprimir la señal, almacenarla y transportarla con un mínimo de recursos, proceso que no afectan ni la calidad del audio ni la del video [2].

De esta forma, el uso de las técnicas de compresión de las señales de imagen y de sonido permite que en el mismo ancho de banda que ocupa un sólo canal analógico se pueda transmitir a la vez varios programas con calidad de imagen similar a la actual (*SDTV*²) o transmitir un programa de Televisión de alta calidad en imagen y en sonido (*HDTV*³) [2].

1.1.1 Ventajas de la Televisión Digital

Algunas de las ventajas que presenta la Televisión Digital frente a la Televisión Analógica se mencionan a continuación [1] [3]:

- Aumento de la calidad de la señal de vídeo y del sonido.
- Incremento del número de canales que pueden emitirse sobre el ancho de banda del espectro electromagnético.
- Reducción de la interferencia de canal adyacente y eliminación de la señal fantasma.

² *Standard Definition Television (SDTV): Señales de televisión que no se consideran de alta definición.*

³ *High Definition Television (HDTV): Señales de televisión digital y de alta definición.*



- Optimización del espectro radioeléctrico.
- Menores costos de transmisión.
- Inclusión de datos adicionales en el flujo de transporte y suministro de mecanismos de canal de retorno.
- Prestación de servicios interactivos.
- Recepción portátil y móvil.

1.2 TELEVISIÓN DIGITAL INTERACTIVA

La Televisión Digital Interactiva (*TDi*) tiene como fin el uso de señales digitales y además, la posibilidad de brindar interactividad al usuario [3].

Técnicamente el termino interactividad sobre este sistema, implica tener un canal de retorno en el sistema de comunicación desde el usuario hasta la fuente de información. Este canal es el vehículo de los datos que representan la reacción del usuario, con esto se puede diferenciar entre sistemas catalogados como interactivos y los denominados sistemas de difusión [4].

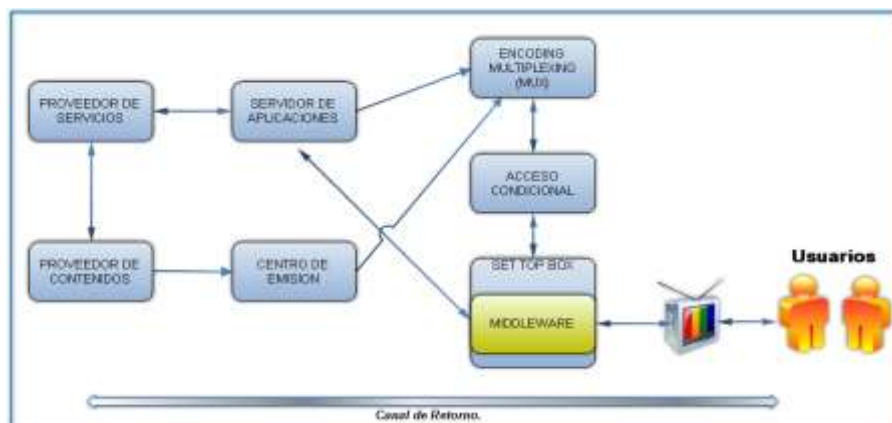


Fig. 1 Arquitectura de la Televisión Digital Interactiva.
Tomado de [6]

Un sistema de *TDi* básicamente se compone por los elementos mostrados en la figura 1, donde se busca inicialmente que las aplicaciones sean transmitidas mediante un previo empaquetamiento y luego se asocian a un canal en particular de Televisión [5].

La señal de salida del multiplexor esta modulada dependiendo de la forma a transmitirse, sea terrestre, satelital o mediante cable. Luego del proceso de modulación, la señal viaja hacia los receptores, exactamente hacia el *STB (Set-Top-Box)*⁴ donde primero se cargan y posteriormente se ejecutan las aplicaciones, así se establece un vínculo entre la fuente de información y el usuario.

⁴ Set Top Box (*STB*): dispositivo encargado de la recepción y opcionalmente decodificación de señal de televisión analógica o digital.



Adicionalmente a esto, otro enlace se crea a partir del canal de retorno donde la comunicación se hace desde el lado del usuario hacia la fuente de información, teniendo así en el sistema una comunicación bidireccional [5].

Se describe a continuación los bloques que componen la arquitectura mostrada en la figura 1:

Proveedor de Servicios: Este módulo proporciona los servicios o contenidos interactivos a través de un servidor de aplicaciones [6].

Proveedor de Contenidos: Suministra contenidos de video y de radio, y transmiten sus contenidos al centro de la emisión del operador [6] [2].

Servidor de Aplicaciones: Prepara a las aplicaciones para su respectiva codificación antes de hacer la emisión.

Centro de Emisión: Toma las señales provenientes del proveedor de contenidos y las prepara para su codificación y emisión [6] además, se ocupa de la sincronización con el canal de retorno y ofrece al proveedor del servicio la posibilidad de comunicarse directamente con el *STB* a través del mencionado canal [2].

Encoding Multiplexing (MUX): Sobre este módulo se hace la codificación de video, audio y datos convirtiéndolos en paquetes apropiados para la modulación. Además se hace la respectiva encriptación y posterior multiplexación para hacer la transmisión. El flujo de salida del multiplexor es un flujo de transporte (*Transport Stream - TS*⁵), el cual viene modulado para alguna de las tipologías posibles: terrestre, satelital, cable o en algunos casos para un ambiente móvil [2].

Set Top Box (STB): Este módulo se encarga de hacer la descryptación, comprobar los derechos que tiene el usuario y decodificar la señal para convertirla en analógica, esta señal posteriormente es enviada al televisor para su visualización [6].

Adicionalmente, sobre este dispositivo se ejecutan las aplicaciones descargadas. Este receptor ejecuta dos funciones principales:

- Recibir tanto las señales de video digital, como las aplicaciones emitidas con sus datos correspondientes, decodificando el video, audio, datos y de acuerdo a los comandos cargados ejecutar las aplicaciones [2].
- Enviar y recibir datos a través del canal de retorno, el cual facilita al usuario la interacción con las aplicaciones emitidas. El usuario también puede pedir información adicional a través del canal de retorno sobre un servidor del proveedor [2].

Middleware: Es una plataforma intermedia común para las aplicaciones interactivas de *TDi* independiente tanto de la empresa que provee los servicios interactivos como del receptor de televisión que se usa [7].

⁵ *Transport Stream (TS):* Es un protocolo de comunicación para audio, video y datos especificado en los estándares de MPEG-2 (*Moving Pictures Experts Group 2*).



Suministra una interfaz de programación (*API*) con el fin de facilitar tanto el desarrollo y ejecución de las aplicaciones en el entorno de la televisión como permitir la interacción con Internet.

Canal de Retorno: Es el medio por el cual se hace posible la interactividad entre el usuario y la fuente de información, este puede ser implementado por cualquier medio físico tal como cable telefónico, radiodifusión, enlace satelital o mediante el uso de *ADSL*⁶.

Finalmente, la calidad en la prestación del servicio depende del tipo de medio usado por el receptor y potencialmente también de la forma como se diseña la aplicación que soporta el servicio [2].

1.2.1 Ventajas de la *TDi*

Además de las ventajas ya mencionadas de la Televisión Digital, cabe resaltar la principal característica de un sistema *TDi*: el uso de la interactividad como herramienta fundamental que posibilita al receptor el dialogar a través del decodificador y del canal de retorno con el emisor. Esto le ofrece al espectador, una gama muy amplia de opciones, desde servicios puramente de entretenimiento, o publicidad interactiva, a servicios que faciliten algunas tareas de la vida diaria como transacciones bancarias, reservas de tiquetes, reserva de citas médicas, y entre otros procedimientos [5].

1.3 ESTÁNDARES PARA TELEVISIÓN DIGITAL

La codificación de la información en el sistema de Televisión debe realizarse mediante la implantación de cualquiera de los estándares disponibles o en lo posible, hacer la adaptación de algunos de ellos al entorno con algunas variantes. La elección depende de los requisitos o los servicios que se requieren implementar sobre el sistema, tales como la protección de datos, el tipo de señal a transmitir, calidad del video y de audio, distancia y la zona geográfica en donde se aplicará entre otras. Cada una de estas normas posee un formato de 1920 x 1080 *píxeles* e incorporan el estándar *HDTV* digitalizado, basado en el formato de comprensión *MPEG-2*⁷, pero aplicado a cada uno de estos con diferentes esquemas en cuanto a comprensión de sonido y de transmisión [8].

A continuación se presentan de manera muy general, algunos de los estándares existentes para Televisión Digital:

1.3.1 *ATSC (Advance Television System Committee)*

ATSC es un grupo encargado en desarrollar los estándares para el sistema de televisión digital en los Estados Unidos como reemplazo del sistema de televisión analógica *NTSC*⁸ [1].

Es una organización internacional abierta sin ánimo de lucro creada en 1982. Aproximadamente cuenta con 185 miembros en varios países, de la cual hacen parte

⁶ *Asymmetric Digital Subscriber Line (ADSL)*: Tecnología de acceso a Internet de banda ancha mediante línea telefónica.

⁷ *Moving Pictures Experts Group 2 (MPEG-2)*: Grupo de estándares de codificación de audio y vídeo para la transmisión de señales.

⁸ *NTSC (National Television System Committee)*: Sistema de codificación y transmisión de televisión en color analógico desarrollado en Estados Unidos e implementado en la mayor parte de América y Japón, entre otros países.



empresas de radiodifusión, cable, satélite, computación, cine y video, electrónica de consumo, fabricantes de equipos profesionales audiovisuales y de computación, institutos de investigación y organizaciones comerciales. En el año 2001 fue creada una institución afiliada a ATSC la cual tiene como fin promover el estándar de televisión digital ATSC principalmente en América Latina [1].

El estándar ATSC fue creado principalmente para la televisión libre y gratuita usando los mismos 6 MHz de ancho de banda que ocupa un canal de Televisión analógica, en donde busca principalmente ofrecer al usuario: televisión en alta definición, multiprogramación, comunicación interactiva, ofrecer 6 veces mejor calidad de imagen que la televisión tradicional y mejor calidad de sonido tipo cine *audio dolby digital*⁹, además permite combinar varias señales con definición estándar o en Televisión en alta definición y entre otras características [1].

1.3.2 DVB (Digital Video Broadcasting)

Es una alianza con más de 280 compañías de difusión, fabricantes, operadores de red, desarrolladores de software, entidades reguladoras y otras instituciones en más de 35 países, comprometidos con el diseño de estándares globales para la Televisión Digital [1]. Los estándares DVB abarcan todos los aspectos de Televisión Digital, desde la transmisión, las interfaces, el acceso condicional y la interactividad del video, audio y datos digitales [1].

En el mes de septiembre de 1993, esta alianza de empresas crea el denominado grupo de lanzamiento Europeo con el objetivo de lograr la estandarización global y la interoperabilidad a largo plazo para desarrollar la televisión digital, firmando el memorando de entendimiento llamado *Muo* que establece el marco de trabajo en que se desarrollará la Televisión digital.

Se comenzó con la elaboración de informes que anticipaban la situación actual, vinculando la televisión Digital con nuevos conceptos HDTV, recepción en equipos móviles, compatibilidad con otros medios entre otros aspectos. [1].

El proyecto DVB tiene desarrollados más de 50 estándares, para televisión en sistemas por cable, televisión terrestre, sistemas digitales de satélite, redes de microondas, y otras aplicaciones [1].

Dentro de los estándares DVB existentes más utilizados para Televisión Digital se tienen [1]:

- **DVB-S** Estándar para Sistemas Digitales de Satélite.
- **DVB-C** Estándar para Sistemas Digitales de Cable.
- **DVB-T** Estándar para Televisión Digital Terrestre.

1.3.3 ISDB (Integrated Service Digital Broadcasting)

El estándar ISDB es un conjunto de normas diseñadas para la transmisión de radio digital y televisión digital desarrollado por Japón [1].

⁹ *Audio dolby digital* es el nombre comercial de una tecnología de compresión de audio desarrollada por Dolby



Promovido por el *DiGEB (Digital Broadcasting Experts Group)* de Japón, grupo de expertos conformado por las principales transmisoras y fabricantes que trabajan en el medio de la radiodifusión, fundado en septiembre de 1997 para promover el sistema de Televisión Digital *ISDB* por todo el mundo. El desarrollo de *ISDB* comenzó en 1980, pero el estándar fue creado en los años 90, comprende la transmisión de video digital por satélite (*ISDB-S*), por cable (*ISDB-C*) y terrestre (*ISDB-T*, incluye terminales móviles) [1].

1.3.4 DTMB (Digital Terrestrial Multimedia Broadcasting)

El estándar *DTMB* es el conjunto de normas para televisión digital terrestre para terminales fijos y móviles de China [1].

En el año de 1994 el gobierno Chino fundó el grupo de Expertos Ejecutivos Técnicos de Televisión de Alta Definición -*TEEG*, cuyos miembros llegaron de varias universidades e institutos de investigación a trabajar en el desarrollo de la televisión digital [1].

Después de tres años de esfuerzo, el grupo desarrolló la primera televisión en alta definición y fue aplicado satisfactoriamente para transmisión en vivo del 50 aniversario del Día Nacional en 1999 [1].

En 2001, China hizo un llamado para recibir propuestas para un estándar terrestre digital de la transmisión de televisión. *DMBT (Digital Multimedia Broadcasting – Terrestrial)*, *ADBT (Advanced Digital Television Broadcasting – Terrestrial)* y *TIMI (Terrestrial Interactive Multiservice Infrastructure)*. La norma China fue definida en 2006 y recibió la aprobación final de la República Popular China en Agosto 2007, comenzando transmisiones en Hong Kong el 31 de Diciembre 2007.

Su definición estuvo a cargo de la Universidad Jiaotong en Shanghai y la Universidad Tsinghua en Beijing.

DTMB es una fusión de varias tecnologías e incluye derivaciones de la norteamericana *ATSC* y la europea *DVB* [1].

El estándar de televisión chino *DTMB* posee un gran alcance de cobertura, mayor que los demás estándares existentes. Es capaz de transmitir *HDTV* de calidad aceptable a vehículos en movimiento a velocidades de hasta 350 Km/h, también permite la transmisión de varios canales por una misma frecuencia, está diseñado para redes de frecuencia única y redes multifrecuencia. Es un estándar que incluye desde sus inicios soporte para dispositivos móviles, como celulares y reproductores multimedia [1].

La tabla 1 resume las características técnicas que cada estándar adopta para la codificación de la información.



Estándar	Mediador	Sistema de Video	Sistema de Audio	Modulación	Frecuencia	Tipo de Sistema	Adoptado
ATSC	DASE	MPEG-2	Dolby AC3	8-VSB	6 MHz	ATSC-C	Estados Unidos, Canadá, Corea del Sur, México, Taiwán, Costa Rica, El Salvador, Alaska, Hawái, Islas Midway. Isla Wake, Honduras, México; Puerto Rico,
						ATSC-T	
DVB	MHP	MPEG-2	MPEG-2	COFDM	7 MHz y 8 MHz	DVB-C	Europa, Australia, Nueva Zelanda, Rusia, Uruguay, Colombia, Argelia, Benín, Egipto, Ghana, Libia Marruecos, Namibia, Sudáfrica; Túnez, Arabia Saudita, Camboya, India, Indonesia, Irán, Taiwán, Tailandia, Singapur, Groenlandia, Panamá.
						DVB-S	
						DVB-T	
ISDB	ARIB	MPEG-2	MPEG-2 AAC	BST-OFDM	6 MHz	ISDB-C	Japón, Filipinas, Belice, Argentina, Bolivia, Brasil, Chile, Ecuador, Paraguay, Ecuador, Perú, Venezuela-
						ISDB-S	
						ISDB-T	
DMBT	--	Cualquier con soporte HD	--	TDS-OFDM	8 MHz	DMB-T	China, Macao, Hong Kong,.

Tabla 1. Resumen Técnico Estándares para Televisión Digital. Tomado de [3] y de [15].

1.4 MEDIADORES DE TELEVISIÓN DIGITAL

1.4.1 Tipos de Mediadores

A medida que surge el interés por explotar al máximo las posibilidades que la *TDi* ofrece, se crean diversas plataformas, unas por entidades de carácter privativo diseñadas para dispositivos receptores específicos destinados a servicios que se prestan para televisión por pago, otras plataformas de carácter abierto que son estandarizadas por algún organismo o por la industria que puede ser implementado en cualquier caso y lo más importante, se usa para la televisión de libre distribución. Esta tendencia da como resultado la aparición de mercados verticales los cuales tienen su origen en las distintas opciones existentes, ya que cada una de estas plataformas son diseñadas e implementadas de manera independiente [9].



Con el desarrollo de la *TDi*, se hace necesario regular los estándares para lograr la migración a un mercado horizontal donde el servicio que se implemente sea independiente de la empresa que lo suministre y del receptor en donde se ejecute [9].

Pero en la realidad, las empresas, organizaciones y otros entes que definieron los mecanismos para la difusión de la Televisión Digital, se dieron a la tarea de definir sus propios conjuntos de estándares mediadores, cada uno de ellos diseñados para cumplir objetivos específicos en cuanto a servicios de información o a transmisión de datos se refiere, algunos de estos mediadores se presentan en la tabla 2 donde se nombran una serie de estándares tanto propietarios como abiertos [9].

Middleware Propietario	Middleware Abierto.
<i>OpenTV Core (OpenTV)</i>	<i>MHEG</i>
<i>MediaHighway (Canal+)</i>	<i>DAVIC</i>
<i>Microsoft TV (Microsoft)</i>	<i>MHP</i>
<i>Liberate</i>	<i>OCAP</i>
<i>PowerTV</i>	<i>ACAP</i>
<i>NDS Core (NDS)</i>	<i>ARIB B23</i>
	<i>JavaTV</i>

Tabla 2. Algunos middleware de tipo propietarios como abiertos.
Tomado de [3].

Es así, que para Europa está definido el estándar *DVB*, donde existen especificaciones para todas las redes de televisión digital: terrestre (*DVB-T*), cable (*DVB-C*), satélite (*DVB-S*). Estas especificaciones permiten disponer de un mismo estándar en cuanto a middleware de servicios interactivos se refiere: *MHP (Multimedia Home Platform)* [9].

Estados Unidos, Canadá y Corea adoptan el *middleware* definido por el estandar *ATSC* para la televisión digital terrestre: *DASE (Digital Television Application Software Environment)* [9].

Estados Unidos usa para el cable la definición de *CableLabs OCAP (OpenCable Applications Platform)*, definida a partir de *ATSC* y basada en *MHP*, la cual tiene un cierto número de características específicas para el mercado americano (al igual que *MHP* las tiene para el europeo) [9].

Japón usa una especificación denominada *BML (Broadcast Markup Language)*, que sólo se usa en Japón, aunque ya están trabajando en una especificación cercana a *MHP: ARIB (Association of Radio Industries and Businesses)* [9].

Brasil implementa para su sistema de televisión digital el *middleware Ginga*. [9].

En Corea se usa *OCAP* para cable, *DVB-S* con *MHP* para Satélite y *ACAP* para terrestre [9].

En el Reino Unido se usó *MHEG* definido por *ISO*, que posteriormente evolucionara hasta *MHEG-5*, pero también ha aceptado el estándar *DVB-T* [9].

1.4.2 Sistemas Mediadores

Como se mencionó anteriormente, cada ente estandarizador define sus mecanismos y normas para la difusión de la Televisión Digital, pero también define sus propios conjuntos



de estándares para la implementación del *middleware* a usar en los dispositivos receptores. Se hace a continuación una breve descripción con los sistemas mediadores abiertos de mayor interés:

MHEG-5 (Multimedia and Hypermedia Information Coding Expert Group): Desarrollado en la mitad de los 90, hace parte de *DAVIC (Digital Audio Video Council)*, el cual creó un gran número de estándares destinados a la navegación e interactividad de servicios multimedia en dispositivos de menor tamaño [11].

El software en el dispositivo del cliente consiste en un motor *MHEG-5* (maquina virtual) que interpreta las aplicaciones de este tipo y presenta la información que se ha entregado por el canal de difusión o por el canal de retorno [11].

MHEG-5 añade interacción entre el usuario y la aplicación a través del control remoto y algunas implementaciones que permite a la aplicación intercambiar información con un servidor mediante una conexión a Internet [11].

El lenguaje de programación para este *middleware* en general comprende objetos para presentación, enlaces que responden a eventos y a programas residentes [11].

Los objetos de presentación incluyen video, audio, listas, texto y gráficos. Los eventos responden a peticiones desde el control remoto, a un temporizador, a mensajes de sucesos en el flujo de la emisión o el resultado de una condición lógica en la aplicación. Los programas residentes son funciones nativas, definidas en el perfil¹⁰, que extienden la base de *MHEG-5* y proveen herramientas específicas para manipular los datos, y además pueden utilizarse para perfiles de mercado específico que permitan a las extensiones *MHEG-5* poder trabajar con sistemas de acceso condicional (*CA*¹¹), control *PVR*¹², *push VOD*¹³, y con el canal de retorno [11].

La emisión de perfiles tiene un ciclo de vida simple, permitiendo solamente una aplicación en ejecución a la vez. Una aplicación *MHEG-5* puede ser invocada por otras, pero como resultado, la que hizo el llamado tiene que terminar su ejecución. En un sistema de difusión, una aplicación de auto invocado puede haberse iniciado cuando un servicio lo seleccionó mediante una previa asociación. El auto lanzamiento puede iniciar otras aplicaciones y seleccionar a otros servicios para sincronizarse. La información se pasa entre aplicaciones haciendo uso de la persistencia que se almacena en el dispositivo receptor [11].

Una aplicación se carga normalmente desde el carrusel de objetos del *DSM-CC*¹⁴ u opcionalmente del canal de retorno desde un módulo *DVB-CI*¹⁵ [11].

¹⁰ Cuando se habla de perfil en MHEG-5, es el conjunto e interpretación de la especificación útil para un contexto en particular.

¹¹ *Conditional Access (CA): sistema que permite controlar el acceso a contenidos.*

¹² *Personal Video Recorder (PVR) Dispositivo interactivo de grabación de televisión en formato digital:*

¹³ *Push VOD (Push Video On Demand):* Es una técnica para simular video bajo demanda pero almacenando el contenido sobre un PVR.

¹⁴ *DSM-CC(Digital storage media command and control)* es un estándar de difusión de datos incluido en el estándar *MPEG-2*.

¹⁵ *DVB-CI (Common Interface):* define cómo tiene que ser la interconexión entre el receptor *DVB* o *host* y el módulo de acceso condicional.



Los datos cargados desde el carrusel pueden ser actualizados e inmediatamente pueden observarse sobre la pantalla. La información se presenta ya sea como contenido incluido, donde el texto y las gráficas están insertados en la aplicación, o como contenido referenciado, donde se toma la información desde el carrusel cuando sea necesario. La estrategia de construir aplicaciones donde predomine la referencia de contenido, le permite permanecer estable con cambios en el contenido y gráficos detallados durante largo periodo de tiempo [11].

DASE (DTV Application Software Environment): El *middleware* DASE de ATSC en Norte América, es la contraparte del estándar de televisión digital DVB-MHP y es la base para la nueva generación de estándares ACAP [3], [12], [13].

Define la capa *middleware* la cual permite al contenido programado y las aplicaciones se ejecuten en un receptor común [12] [13]. El *middleware* proporciona a las aplicaciones un entorno de ejecución de manera uniforme en cualquier dispositivo receptor el cual implemente a esta plataforma mediadora, ya que la implementación de los servicios y de las aplicaciones deben adaptarse a las características del ambiente en donde se va a desplegar el servicio [3].

OCAP (OpenCable Application Platform): OCAP define una especificación común para la capa mediadora en los sistemas de cable en los Estados Unidos. Surgió como un estándar abierto producto del trabajo conjunto entre *OpenLabs* con DVB [13].

Inicialmente tuvo como base la versión 1.0.0 de la especificación MHP, actualmente tiene como referencia a MHP Globalmente Ejecutable (*Globally Executable MHP - GEM*) el cual facilita el uso de elementos del estándar MHP en otras especificaciones [13].

MHP (Multimedia Home Platform): MHP y GEM, son dos especificaciones de *middleware* abiertos basados en *Java* y desarrollados por el proyecto DVB [14].

MHP fue diseñado de acuerdo a todas las especificaciones para las tecnologías de transmisión de DVB (DVB-T, DVB-C y DVB-S), mientras que GEM es una plataforma independiente la cual puede ser adaptada a un amplio rango de sistemas de transmisión incluyendo a los sistemas de Televisión Digital [14].

GEM es un núcleo común en donde las aplicaciones interactivas pueden ejecutarse, pero estas requieren saber cómo deben interactuar y como conectarse a este núcleo [14].

En el ámbito de MHP, se ha definido el concepto de “perfil” como un área de aplicación y, como consecuencia, con una serie de capacidades determinadas para cada uno de ellos.

La existencia de los perfiles marca la evolución de la plataforma, junto a una arquitectura y unos procesos flexibles pensados para facilitar la portabilidad e interoperabilidad de aplicaciones, que están sometidas a un ciclo de vida muy definido.



Fig. 2 Estructura Simplificada de MHP.
Tomado de [10].

Se definen tres tipos de perfiles, según las capacidades del receptor:

MHP Enhanced Broadcast (MHP 1.01): Perfil ideal para receptores de bajo costo y de capacidades limitadas, no incluye canal de retorno por lo que está pensado para la descarga a través del canal de *broadcast* de aplicaciones que puedan proporcionar únicamente interactividad en una sola dirección [8] [3].

MHP Interactive Broadcast (MHP 1.02): Este perfil incluye canal de retorno vía IP, permitiendo una comunicación bidireccional con el proveedor de servicios interactivos, incluye *APIs*¹⁶ para el control del canal. Este tipo de receptores permiten aplicaciones como vídeo bajo demanda, comercio electrónico, tele-voto, concursos interactivos y entre otros [8] [3].

MHP Internet Access (MHP 1.1): Además de incluir las capacidades de los dos perfiles anteriores, tiene características para prestar servicios más avanzados para el dispositivo receptor como acceso al correo electrónico, búsquedas *web* y otras actividades relacionadas.

Incluye soporte adicional para aplicaciones *DVB-HTML* [8] [3].

La arquitectura de *MHP* está definida en tres niveles que a su vez se subdividen en diversos campos: recursos, sistema de software y aplicaciones.

Recursos: Los recursos de *MHP* básicamente son: Procesador *MPEG*, dispositivos E/S, CPU, memoria, sistema de gráficos [3] [14].

¹⁶ *API* (Application Programming Interface): Conjunto de funciones y métodos disponibles que pueden ser incluidos en la implementación de software.

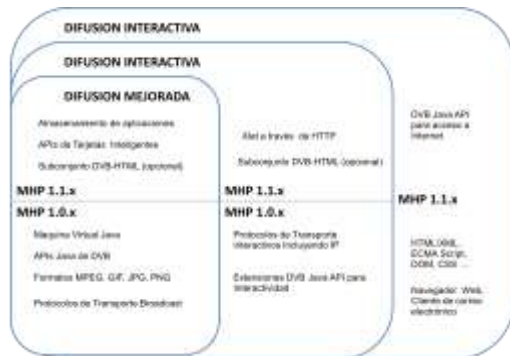


Fig. 3 Relación entre Perfiles de DVB-MHP.
 Tomado de [15].



Fig. 4 Arquitectura de la Plataforma MHP.
 Tomado de [15].

Sistema software: Este nivel aísla a las aplicaciones de los recursos existentes en el decodificador. Se implementa la *JVM*¹⁷, que ofrece las diversas bibliotecas *Java* que finalmente conformarán el *API* que los proveedores de aplicaciones interactivas podrán utilizar en sus desarrollos [3][14].

Define también el comportamiento que las aplicaciones deben presentar en su ejecución, es decir, el ciclo de vida de estas, permitiendo que incluso el operador pueda controlarlo desde la capa de mayor jerarquía [14].

Aplicaciones: Sobre esta capa se sitúan las diversas aplicaciones descargadas y que utilizan los servicios ofrecidos por las capas inferiores [14].

1.4.3 Modelos de Aplicaciones

MHP define dos modelos de aplicaciones. Por un lado, se tiene *DVB-J* basadas *Java*. Por otro lado se tiene *DVB-HTML* basado en *HTML* y *JavaScript*¹⁸ [14].

DVB-J: Una aplicación *DVB-J* (también conocida como *Xlet*¹⁹) es un programa escrito en *Java* que cumple dos condiciones principales [15]:

1. Hace uso únicamente de las librerías y *APIs* de clases *Java* (*JMF*, *JSEE*, *JavaTV*) definidas expresamente en la norma *MHP*.
2. Genera y atiende a una serie de señales que implementan un ciclo de ejecución perfectamente especificado en la norma *MHP*, y permite que una aplicación sea fácilmente controlada por el gestor de aplicaciones de la máquina *MHP*. El gestor de aplicaciones es la entidad encargada de iniciar y parar las distintas aplicaciones y de hacer el monitoreo de su ejecución.

¹⁷ *JVM* (*Java Virtual Machine*): es un programa que se ejecuta en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial, el cual es generado por el compilador del lenguaje *Java*.

¹⁸ *JavaScript*: Lenguaje de programación interpretado usado en páginas web.

¹⁹ *Xlet*: Tipo de aplicaciones en *Java* para entornos de televisión



DVB-J define un grupo común de APIs en Java para el desarrollo de aplicaciones de Televisión Digital:

- **Sun Java API:** El cual incluye:
 - **JMF (Java Media Framework):** API el cual define el control de la emisión de video y de audio [16].
 - **CORE JAVA:** Grupo de APIs básicos para implementar aplicaciones en Java. Adicionalmente provee un *framework* para gráficos con algunas partes de AWT²⁰ que han sido incorporados como *Component*, *Container*, *Graphics* entre otros [16].
 - **JAVA TV:** Maneja el ciclo de vida de las aplicaciones [16].
 - **JSEE: (Java Secure Sockets Extensión):** Conjunto de paquetes que proveen comunicaciones seguras a través de Internet [15].
- **Interoperabilidad de Audio/Video (HAVi):** Define las características más apropiadas que deben tener los elementos empleados para desarrollar la interfaz gráfica de las aplicaciones en un contexto como el de la televisión. Los paquetes HAVi incluidos en la especificación MHP sustituyen las funcionalidades del java.awt que han sido omitidas en MHP [15].
- **Consejo Digital de Audio Video (DAVIC):** Se ocupa de los temas relacionados con el flujo MPEG. Trata temas relacionados con funciones de bajo nivel del STB: sintonización del flujo de transporte, acceso condicional, filtrado de secciones privadas de MPEG-2, uso de recursos escasos, etc. [15].
- **APIs específicas definidas por DVB:** En ellas se definen todos los aspectos necesarios para gestionar las nuevas funcionalidades de los receptores (como el manejo de eventos, aplicaciones, carrusel de objeto etc.) que aún no han sido tenidas en cuenta por las API's ya mencionadas [15].

DVB-HTML: Es un modelo de aplicación especificado dentro de MHP 1.1, introducido como *plug-in* en el perfil *Enhanced Broadcast* y opcional en los perfiles *Interactive Broadcast* e *Internet Access* de MHP [16].

Usado por los operadores quienes pueden llevar contenidos basados en XML [16].

La Fig. 5 se presenta un esquema del modelo de aplicación DVB-HTML (*Digital Video Broadcast HyperText Markup Language*), incluyendo los estándares de la W3C (*World Wide Web Consortium*) tales como XHTML (*Extensible Hypertext Markup Language*), CSS 2.0 (*Cascading Style Sheets*), DOM (*Document Object Model*), and ECMAScript que lo componen.

²⁰ AWT (*Abstract Window Toolkit*): API estándar que suministra una interfaz grafica de usuario para un programa Java.

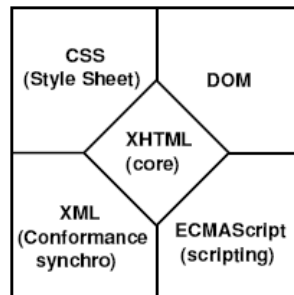


Fig. 5. Lenguajes que intervienen en el modelo DVB-HTML.
Tomado de [16].

Primero, el núcleo de *DVB-HTML* es una gran colección de elementos básicos de *XHTML* e incluye un nuevo módulo, eventos intrínsecos *DVB*, extensión dentro de un nuevo dominio *XML* (*Extensible Markup Language*), *dvbhtml* [16].

Segundo, *DVB-HTML* es totalmente compatible con *CSS 2.0*, añadiendo un nuevo tipo de medio (*dvb-tv*) e incluye algunas extensiones las cuales estarán presentes para diferentes plataformas (*PDAS*, teléfonos móviles, *PC*, *STB*) y lo hace capaz de diferenciar cuando se ejecuta en entornos de televisión [16].

Tercero, *DVB-HTML* incluye un número de módulos de *DOM 2.0* y nuevos módulos: *DOM DVB-HTML* para sustituir al módulo *DOM Nivel 2 HTML*, ya que *DOM nivel 2* soporta *DTD HTML 4.01*, el cual es innecesario; Eventos *DOM DVB* los cuales permite integrar *DVB-HTML* dentro de la plataforma *MHP*; *DOM key Events* para vincular los eventos al control remoto; *DOM DVB CSS* para tener acceso a las extensiones *CSS* en los entornos de televisión; Módulo de entorno *DVB*: para que una aplicación *DVB-HTML* pueda acceder a la programación de algunas variables de entorno, por ejemplo las que contienen la ventana o el marco o algunos parámetros del procesador de aplicaciones *HTML* [15].

1.5 REQUISITOS MÍNIMOS PARA EL DESPLIEGUE DE APLICACIONES EN EL SISTEMA DE TELEVISIÓN DIGITAL INTERACTIVA

Un sistema de *TDi* el cual permita realizar el despliegue de aplicaciones y la distribución de contenidos esta básicamente compuesta por los siguientes elementos:

- Un *Set Top Box*.
- Canal de retorno.
- Servidor de aplicaciones.
- Servidor de *Playout* con tarjeta moduladora.
- Televisor.

Además se tiene en cuenta:

- A nivel de diseño e implementación es necesario adaptarse al estándar que se ha seleccionado para el sistema de Televisión Digital.



- Si no se cuenta con la infraestructura básica de un sistema de *TDi*, perfectamente se hace el despliegue de la aplicación sobre emuladores de Televisión Digital Interactiva como lo es *XleTView* para verificar su funcionamiento, ya que estas herramientas ofrecen las funcionalidades básicas que presenta el sistema de *TDi* durante el proceso de despliegue de las aplicaciones.
- En cuanto el *middleware*, se elige según el estándar definido previamente para la difusión de la información en el sistema. Pero a nivel práctico, la elección de este no tiene mayor relevancia para la implementación de aplicaciones. La diferencia radica en el uso o el llamado de las librerías correspondientes a cada mediador que facilitan la interacción de la aplicación con los recursos del dispositivo receptor.

Es importante notar que debido a los diferentes estándares para la codificación de la información que existe, tanto abiertos como privativos, y sus respectivos mediadores, las descripciones siguientes se concentran sobre el estándar Europeo *DVB* con su mediador *DVB-MHP*.

Esta aclaración se hace debido que durante la realización de este proyecto, se toma como referencia los trabajos de grado ejecutados con base al laboratorio experimental de Televisión Digital Interactiva de la Universidad del Cauca y a la infraestructura propuesta por el proyecto *EDiTV*, producto de esto ya se tiene definido al estándar Europeo *DVB* con su mediador *DVB-MHP*, como los elementos para la codificación de la información y los elementos necesarios para la interacción con los recursos del dispositivo receptor dentro de esta infraestructura.

En la figura 6 se muestra la arquitectura de referencia *end-to-end* para el estándar *MHP*.

Emisión de los contenidos de programa (Sudio/video/subtítulos).: El *playout*²¹ es una función necesaria en la mayoría de las redes de difusión. El contenido del programa se envía desde el proveedor de contenidos para el sistema de emisión; aquí entra el *Playout* que alimenta a los equipos de red. La implementación actual de este componente depende del formato del contenido de programa cuando entre a un sistema en particular. Normalmente el *playout* esta implementado como servidores de video que almacenan y reproducen los flujos o como codificadores que convierten el material de la fuente sin comprimir a un flujo de transporte (*transport stream - TS*) compatible con el estándar para difusión de datos elegido [18].

Herramientas de autoria y de producción: Es un conjunto de herramientas destinada a la rápida creación y desarrollo de aplicaciones en el entorno de la Televisión Interactiva [19].

El proceso de creación y difusión pueden ser divididos en dos pasos [18]:

1. Diseño / Desarrollo de aplicaciones.
2. Creación de los datos para la emisión.

²¹ *Playout*: La transmisión de canales de radio o de televisión desde las estaciones de transmisión hasta las redes que distribuyen el servicio a los usuarios.

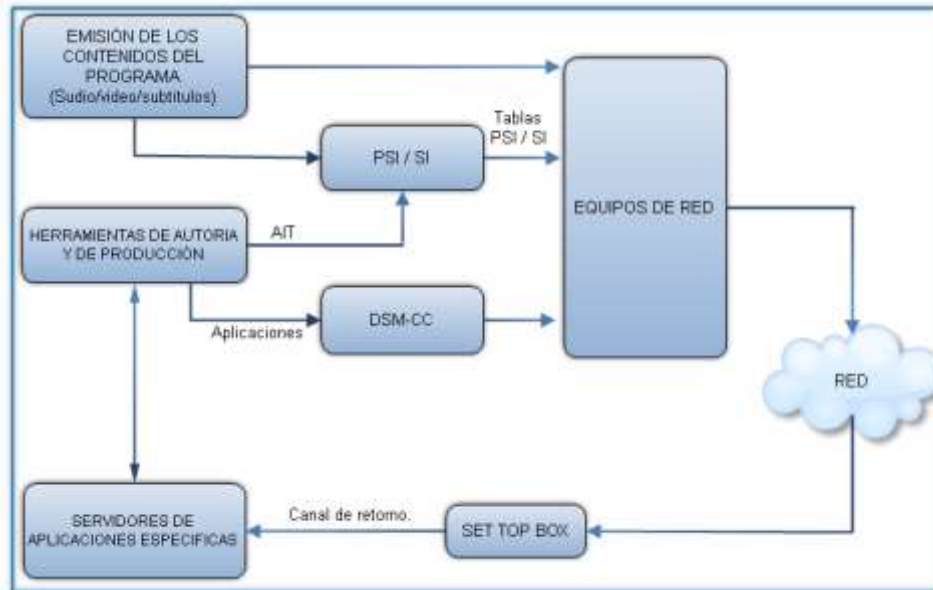


Fig. 6. Modelo de Referencia end to end para el estándar MHP.
Tomado de [18].

PSI / SI: Las tablas *PSI* son usados por el receptor para conocer la información que lleva el flujo de transporte (*TS*). El Audio, video y otros datos son multiplexados cuando se transmiten por un solo *TS*. Dentro del estándar *DVB* para la distribución, estas fuentes separadas de datos son representados con un único identificador (*PID*²²). Las tablas *PSI/SI* se utilizan como una tabla de contenidos, permitiendo navegar entre todos los *PID* recibidos [3] [15].

Program Specific Information (PSI) está definido como una parte del estándar *MPEG-2*, donde se define las siguientes tablas [3] [15].

- **Program Allocation Table (PAT):** Contiene los códigos de identificación e información del sistema asociado con programas que están contenidos sobre un flujo de transporte. El *PAT* normalmente es enviado cada 20 ms a 100 ms para permitir que el receptor adquiriera rápidamente una lista de programas disponibles [3] [15].
- **Program Map Table (PMT):** Contiene información que identifica y describe los componentes (como el video y audio) que forman parte de un programa. Usando los *PID*'s en el *PMT*, el receptor puede seleccionar y combinar los diferentes contenidos para representarlos en un programa de televisión [3] [15].
- **Conditional Access Table (CAT):** Contiene la información que es usada por un dispositivo de acceso (como un *STB* o una tarjeta inteligente) para decodificar programas que son parte de un sistema de acceso condicional (como por ejemplo programas bajo demanda). Sólo es obligatoria en caso de que algún programa del *TS* esté codificado [3] [15].

²² *PID (Process ID):* Un número entero usado por el kernel de algunos sistemas operativos para identificar un proceso de forma unívoca.



Service Information (SI): es una adición a la especificación *PSI*. Definida por el *DVB*, conocido también como *DVB-SI*. Define una serie de tablas de tipo obligatorias y opcionales, que proveen información extra acerca de los datos en un *TS* e información relacionada con datos externos y propiedades del *TS*. Las tablas obligatorias son [3] [15]:

- **Network Information Table (NIT)**: Contiene información sobre la red del actual o de otros *TS* que se está ejecutando.
- **Service Description Table (SDT)**: Describe todos los servicios disponibles en el actual y otros *TS*.
- **Event Information Table (EIT)**: Ofrece información sobre los eventos presentes o futuros de la trama de transporte. De esa forma el receptor conoce el momento de inicio del evento y su duración, entre otras características.
- **Time and Date Table (TDT)**: Esta tabla se utiliza para transmitir la hora y fecha actual, de tal forma que el receptor pueda sincronizarse a ella.

Las tablas opcionales son [3] [15]:

- **Bouquet Association Table (BAT)**: Provee agrupamiento lógico de servicios a través de *TS*.
- **Running Status Table (RST)**: Cada evento tiene un estado de emisión, es decir, en un preciso instante puede estar o no en emisión. La actualización sincronizada de la información de la *RST* permite cambiar automáticamente de evento tan pronto como este empiece.
- **Time Offset Table (TOT)**: Son tablas que permiten la corrección de horarios.

AIT (Application Information Table): Esta tabla contiene una descripción de todas las aplicaciones que pueden ejecutarse durante la emisión de un servicio [3] [15].

Se trata de una tabla adicional a la señalización de las tablas de información de servicio *PSI* y *SI* que definen *MPEG-2* y *DVB* respectivamente. Es una tabla obligatoria para cada servicio con aplicaciones asociadas, y debe retransmitirse con un intervalo máximo de repetición de 10 segundos [3] [15].

DSM-CC (Digital Storage Media Command and Control): Es una especificación que forma parte del grupo de estándares *MPEG-2* diseñada inicialmente para controlar contenido de tipo multimedia sobre redes, pero que finalmente se amplió para proporcionar servicios de aplicaciones sobre redes heterogéneas [3].

En los sistemas de emisión, la transmisión es en un solo sentido. Los datos se envían desde el sistema de emisión a un receptor que no puede solicitar datos del transmisor [3].

La solución para estos sistemas es que el organismo de radiodifusión transmita periódicamente todos los archivos en el sistema de archivos y el receptor espera el archivo que necesita. Este tipo de solución se conoce como un carrusel, cada uno de los archivos gira y el receptor debe esperar los archivos antes de utilizarlo [3].



Existen dos tipos de carrusel:

- **Carrusel de datos:** Proveen un método para que el *broadcaster* haga la transmisión periódica de módulos de datos a través de las redes de radiodifusión [20].
- **Carrusel de Objetos:** Consiste de un árbol de directorios, que está formado principalmente de imágenes, clases y archivos de texto, que se divide en un conjunto de módulos, los cuales contienen uno o más archivos o directorios. Los módulos son difundidos uno después de otro, hasta completar el envío de todos, luego el proceso se repite [3].

Equipos de red: Consiste en multiplexores y adaptadores de red necesarios para el tipo de red a utilizar (terrestre, cable o satélite) [3] [15].

SET TOP BOX: Este componente recibe la señal digital de televisión y la decodifica además de ofrecer el entorno adecuado donde se ejecutan las aplicaciones. [3] [15]. Este elemento proporciona al televisor, todos los datos que van a ser presentados al usuario como los son el contenido de televisión y las opciones con las cuales va a interactuar el usuario con el sistema [3] [15].

Canal de retorno: Es el medio de transmisión que comunica al usuario con la fuente de información [3][15].

Servidores de aplicaciones específicas: Los servidores de aplicaciones específicas son probablemente las últimas de las áreas en estandarizarse sobre la arquitectura *end to end* en el sistema de *TDi* [3] [15].

Algunas de las aplicaciones necesitan acceder o almacenar información en un repositorio central; este almacenamiento debe estar implementado en un servidor *back end* de aplicaciones específico. El *STB* se comunica con el servidor mediante el canal de retorno y el servidor debe responder a través del mismo canal o por el canal de *broadcast*.

Como el servidor tiene más recursos que el *STB*, es buena práctica hacer el diseño con la mayor parte de la lógica de la aplicación ejecutándose sobre el servidor, realizando así tareas que demanden el uso de demasiados recursos en nombre del dispositivo receptor [3] [15].

La figura 7 presenta el modelo propuesto por el proyecto *EDiTV*, el cual muestra el diagrama de despliegue de un sistema de televisión digital interactiva en cuanto a nivel hardware como software [3] [15].

Este modelo es el actual que se tiene implementado en la Facultad de Ingeniería Electrónica y Telecomunicaciones (*FIET*) de la Universidad del Cauca [3] [15].

Se hace la descripción de los elementos que aun no han sido mencionados en pasajes anteriores [3] [15]:

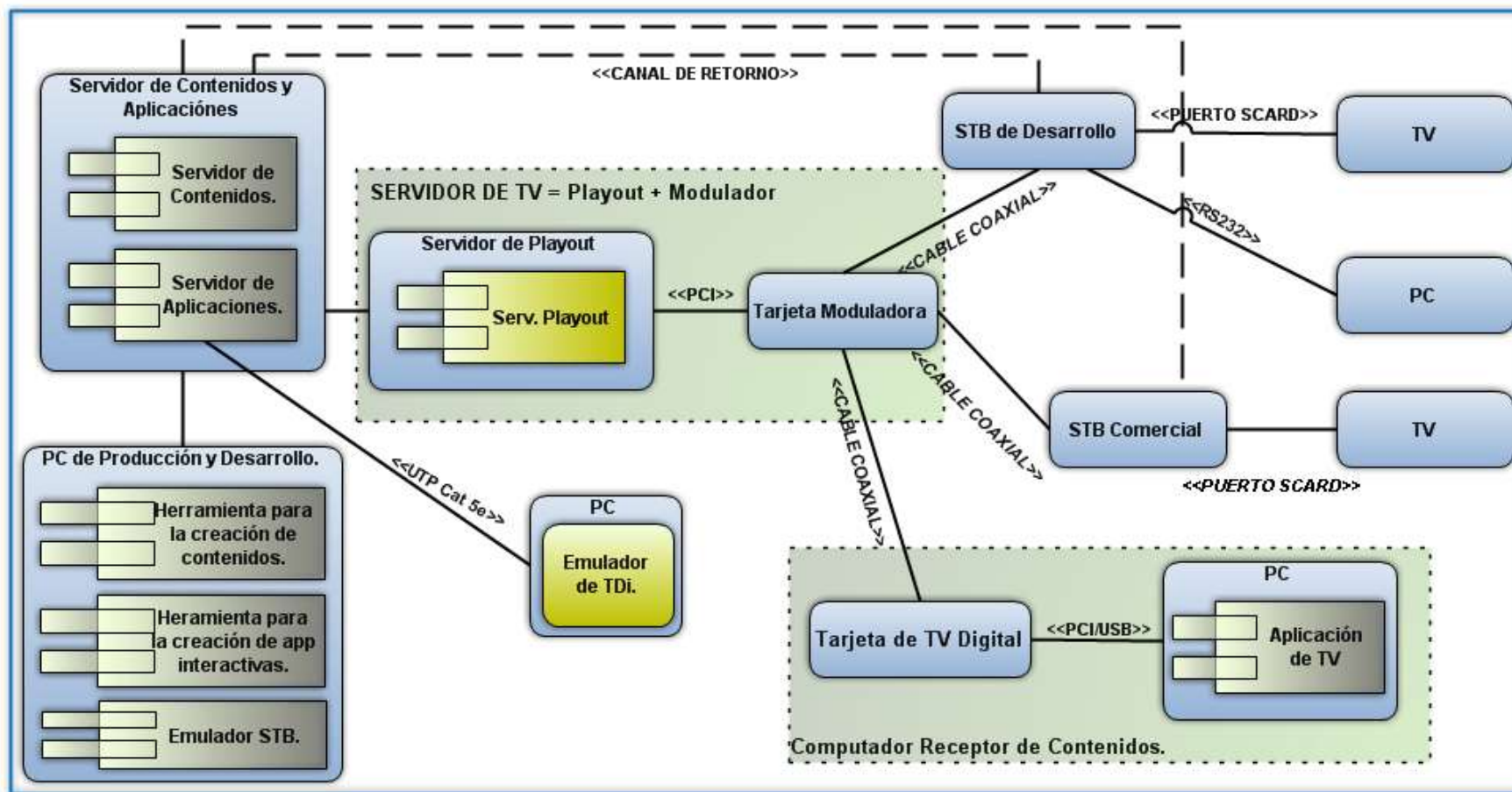


Fig. 7. Diagrama de Despliegue del sistema de TDi para el proyecto EDiTV.
 Tomado de [3] [15] [17].



Servidor de Televisión: Este componente es representado por un servidor con características especiales de procesamiento, memoria y almacenamiento para dar soporte a los servicios de *Playout* y de modulación [3] [15].

Dentro de este servidor se lleva a cabo la generación del flujo de transporte *MPEG-2* a través de la multiplexación de datos de diferentes fuentes como son: los flujos de audio/video, la información de programa/servicio y el sistema de archivos del carrusel de objetos, tareas que demandan gran cantidad de procesamiento y un funcionamiento continuo. El servidor presenta al menos una interfaz de red para la comunicación con el servidor de contenidos y aplicaciones; así como una interfaz de cable que lleva el flujo de transporte modulado hacia los equipos terminales [3][15].

El servidor de televisión está compuesto por dos elementos como se puede observar en la figura 7: el servidor de *playout* y la tarjeta moduladora.

Servidor de *Playout*: El servidor de *Playout* es un componente basado en software que integra la codificación del audio y video; la generación del carrusel de objetos, datos y eventos; la generación de la información de programas, servicios y aplicaciones; la multiplexación de todos estos contribuyentes para la generación de un único flujo de transporte *MPEG-2* que será modulado posteriormente para su transmisión [3] [15].

Dentro de este componente se realiza la programación de la emisión (programas, aplicaciones interactivas y eventos) y configuración de algunos parámetros de transmisión como la tasa de bits entre otros [3] [15].

Tarjeta Moduladora: El modulador realiza la adaptación de los flujos de televisión a una señal adecuada para la transmisión bajo un estándar, por ejemplo, *DVB-T* o *DVB-C* [3] [15].

Tarjeta de TV Digital: Este componente permite sintonizar los canales de Televisión Digital para ser visualizados en el computador a través del software que ofrece esta tarjeta [3] [15].

Aplicación de TV: Software comercial que permite el manejo de la señal de televisión y la ejecución en el computador de aplicaciones interactivas [3] [15].

En cuanto a software, es de vital importancia tener en el servidor de *playout* un conjunto de herramientas que permita gestionar los flujos de transporte (*TS*) *MPEG-2*, generar las tabla *PSI/SI* y, un generador de carrusel de objetos. Esto para la generación, procesamiento, difusión y la multiplicación de flujos de transporte [3] [15].

Emulador de *TDi*: El emulador de *TDi* es una aplicación la cual emula las funcionalidades y características de un sistema de Televisión Digital Interactiva. Esta opción es la más apropiado para un ambiente de diseño y de pruebas, ya que la ejecución y despliegue de las aplicaciones se hacen sin necesidad de tener a disposición un servidor de televisión y un dispositivo receptor [3] [15].

Cabe mencionar que las aplicaciones ejecutadas por este medio pueden posteriormente desplegarse de manera correcta sobre un servidor de televisión para su emisión.



Existen varias plataformas libres y de código abierto como también existen plataformas comerciales las cuales cumplen a cabalidad las funciones que se han mencionado. *Opencaster* y *Fraunhofer IMK MHP Payout System*, son dos plataformas de libre distribución y de código abierto las cuales brindan características y funcionalidades completas para un sistema de *layout*. Cualquiera de estos dos componentes podría tenerse en cuenta para la implementación de un laboratorio de pruebas para *TDi* de bajo costo y como herramienta para los desarrolladores de aplicaciones [3] [15].

En la planificación y el diseño de los diversos servicios que se pueden ofrecer en el entorno de la *TDi*, se debe examinar la viabilidad de estos al introducirlos en este ambiente, ya que se busca principalmente en mantener o mejorar el nivel de calidad y las prestaciones para beneficio tanto de los operadores de la red como para los usuarios finales.

Se presentan a continuación algunos de los servicios los cuales se estima serán de gran éxito y de gran acogida que pueda ofrecer el sistema de *TDi*:

EPG (Electronic Program Guide): Las listas con los canales de *TV* y programas disponibles [3] [15].

Digitext y Teletext: Brinda todo tipo de información adicional relacionada con el programa de televisión [3] [15].

E-mail y Chat: Estas aplicaciones requieren el uso del canal de retorno para conectarse a un servidor de correo en Internet y un teclado físico o virtual para la introducción de la información [3] [15].

Servicios de video bajo demanda: Los usuarios seleccionan las películas, estas se transmiten o se entregan en un horario fijo y luego el usuario se limita a permitir el acceso [3] [15].

T-Learning: La convergencia de la Televisión Interactiva y *e-learning*, entre otros [3] [15].

Una vez hecho el análisis de las ventajas que puede ofrecer un servicio desplegado sobre la plataforma de *TDi* y de tener seguro su viabilidad sobre este ambiente, se elige el modelo de aplicación para su desarrollo e implementación.

Se toma como referencia a los modelos de aplicación *DVB-J* y *DVB-HTML*, propuestos en el estándar *DVB*. Estos tienen características a favor y contra dependiendo del tipo de aplicación a implementar.

Las aplicaciones basadas en *DVB-HTML* toman mucho tiempo en ser cargadas y desplegadas en el sistema receptor de usuario, principalmente este modelo de aplicación es usado para aplicaciones que hagan uso del canal de retorno por lo que se implementa en servicios que necesitan acceder a recursos de Internet. Adopta estándares de la *W3C*, que permiten el fácil desarrollo mediante el uso de etiquetas. Las desventajas más sobresalientes para este modelo son: la mayoría de dispositivos receptores no implementan un soporte para este modelo de aplicación y las limitadas etiquetas que soporta este estándar las cuales no permiten la implementación de aplicaciones complejas. Como conclusión, este modelo se implementa hacia aplicaciones y servicios de fácil desarrollo y además con interacción hacia recursos disponibles en Internet [3] [15].



Las aplicaciones basadas en *DVB-J* invocan funcionalidades de las librerías Java, destinadas para trabajar con recursos multimedia, implementaciones para comunicaciones seguras y entre otros aspectos. La construcción de aplicaciones con este modelo de aplicación demanda de mucho tiempo, pero hay beneficios en cuanto a procesos de carga y ejecución relativamente más cortas y eficientes comparadas con las aplicaciones soportadas por *DVB-HTML*. Este modelo principalmente se implementa para aplicaciones y servicios con funcionalidades complejas, como los de votación y *rating*, video bajo demanda, entre otros [3] [15].

Es así que, para el funcionamiento eficaz y eficiente de la aplicación como del servicio a prestar, la elección del modelo de aplicación es primordial para cubrir los requisitos impuestos por el problema al cual se le quiere dar solución.

Teniendo definido el modelo de aplicación a usar, hay que comprender y aplicar el modelo de referencia gráfica para la implementación de aplicaciones *MHP*.

MHP proporciona las herramientas adecuadas para hacer el control de la posición del video, componentes de la interfaz como botones, etiquetas y entre otros, además de las gráficas básicas [3] [15].

La pantalla del sistema *MHP* está compuesta por tres planos: el plano de fondo, el de video y el de gráficos [3] [15].

Plano de fondo: Esta compuesta por imágenes estáticas o un fondo de color, que llena cualquier área no cubierta por la capa de video [3][15].

Plano de video: Sobre esta capa se añade el contenido del video que se desea visualizar. Puede ocupar la pantalla completa o alguna área de la pantalla y el espacio sobrante será cubierto con el contenido del plano de fondo [3][15].

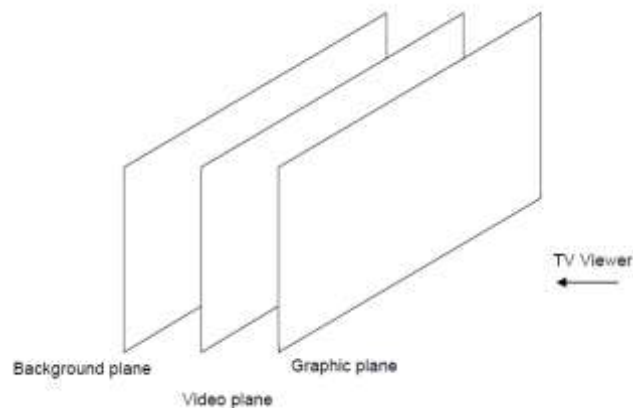


Fig. 8 Planos gráficos en MHP.
Tomado de [3].

Plano de gráficos: Sobre esta capa se añaden los componentes AWT de la aplicación [3] [15].

Los tres planos están superpuestos, así el contenido del plano de las gráficas está siempre visible (sobre la parte superior). El plano de video es visible sólo en localidades de píxel



donde el plano gráfico es transparente. El plano de fondo es visible sólo en localizaciones de píxel donde los planos de video y gráfico son transparentes [3][15].

1.6 RESUMEN DE LOS ELEMENTOS MÍNIMOS PARA EL DESPLIEGUE DE APLICACIONES EN EL SISTEMA DE TELEVISIÓN DIGITAL INTERACTIVA

Para finalizar este capítulo se presenta de acuerdo al análisis realizado, los elementos de la arquitectura necesarios para efectos de implementación y despliegue de aplicaciones las cuales se muestran en la tabla 3.

Elementos necesarios para el despliegue de aplicaciones de <i>TDi</i> .	Elementos necesarios para la construcción de aplicaciones de <i>TDi</i> .
<i>Set Top Box (STB)</i>	Determinar el tipo de estándar elegido e implementado en el Sistema de Televisión Digital
Canal de retorno.	Tener en cuenta el <i>middleware</i> que incorpora el receptor de Televisión y además de sus capacidades para el diseño de la aplicación.
Servidor de aplicaciones.	
Servidor de <i>Playout</i> .	
Televisores	

Tabla 3 Resumen de los elementos mínimos para el despliegue de aplicaciones en el sistema de *TDi*.



CAPITULO 2. EXPLORACIÓN DE ARQUITECTURAS DE RED DE TELECOMUNICACIONES, PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS DE TELECOMUNICACIONES Y PARA EL ENTORNO DE LA *TDi* APLICABLES AL PROYECTO

La evolución de las redes de telecomunicaciones ha sido marcada por tres grandes desarrollos sometidos a constantes procesos de optimización hasta el día de hoy: el servicio telefónico plano antiguo, el control de programa almacenado y la red de señalización por canal común [21].

- **Servicio Telefónico Plano Antiguo (*Plain Old Telephone Service - POTS*)**

La lógica del servicio de los sistemas de conmutación se basaba completamente sobre líneas cableadas. Cuando se requería implementar un sistema de conmutación, los operadores de red consultaban a los proveedores por los equipos de conmutación adecuados para los servicios que los usuarios demandaban.

- **Control por Programa Almacenado (*Stored Program Control Exchange - SPC*)**

Con la introducción del control por programa almacenado, se facilitó la implementación de servicios ya que la lógica de estos es programable, dando como resultado una mejora notoria comparada con los servicios que se ofrecían sobre los sistemas de conmutación sobre el servicio telefónico plano antiguo [21].

- **Red de Señalización por Canal Común (*Common Channel Signaling Network - CCSN*)**

El Sistema de Señalización Número 7 (*Signaling System Number 7 - SS7*) es el protocolo que se ejecuta sobre la red de señalización por canal común (*CCSN*). Con el uso del *SS7* lo que fundamentalmente se realiza es una separación de la señalización del canal de conversación [21].

Actualmente el rumbo está marcado por la implementación de redes y servicios de voz sobre *IP*, la cual ha tomado un lugar de privilegio en distintas áreas tales como empresas, operadores de telecomunicaciones y en usuarios finales. Como consecuencia, existe la necesidad en la creación de nuevos servicios sobre este tipo de redes en tiempos relativamente cortos y además con gran nivel de calidad, esto puede lograrse mediante el uso de plataformas orientadas a la creación de servicios [23].

En cuanto a la creación de aplicaciones para el sistema de *TDi*, una opción es hacer uso de herramientas de autoría para la implementación de manera fácil, rápida y sencilla sin necesidad de tener algún tipo de conocimiento en lenguajes de programación o conocer la arquitectura que se tiene implementada sobre el sistema.

Este capítulo explora algunas arquitecturas de red de telecomunicaciones, plataformas para la creación de servicios de telecomunicaciones, herramientas de autoría y plataformas para



distribución de contenidos que permitirá al final, tener una serie de conclusiones e indicadores que posteriormente serán tenidas en cuenta para la elección de las tecnologías a usarse en el desarrollo del proyecto. Cabe aclarar el alcance realizado en cuanto a la exploración de los elementos relacionado con el sistema de *TDi*, ya que es de gran interés los estándares y recursos propuestos y definidos en el laboratorio experimental de Televisión Digital Interactiva de la Universidad del Cauca y además de proyectos ejecutados en ese entorno como tales como *EDiTV*.

2.1 ARQUITECTURAS DE RED DE TELECOMUNICACIONES Y PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS

2.1.1 RED INTELIGENTE (*Intelligent Network - IN*)

La *Red Inteligente* es una arquitectura ampliamente conocida y madura, la cual tiene la capacidad de proveer servicios de valor añadido sobre la red telefónica básica mediante la inclusión de un entorno informático a la red de conmutación tradicional ofreciendo funciones diferentes a las propias que se tenían hasta ese momento[24] [26].

2.1.1.1 Característica de la *Red Inteligente*

La *Red Inteligente* tiene como principal característica presentar una arquitectura que separa las funciones de control de las funciones de conmutación, esto se traduce en la independencia entre los dos tipos de procedimientos mencionados anteriormente que están involucrados en el tratamiento de una llamada [21] [27].

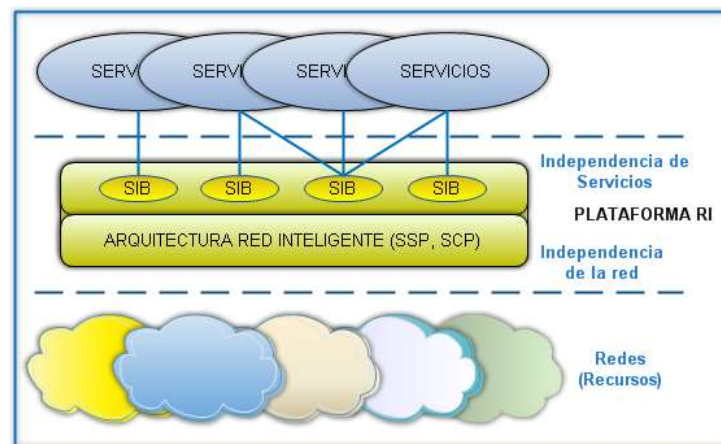


Fig. 9. Red Inteligente como una plataforma universal de servicios.
Tomado de [27].

Teniendo en cuenta lo anterior y con respecto a la figura 9, se muestra a la plataforma de la *Red Inteligente* como una capa adicional por encima de cualquier red de transporte, para permitir a los operadores desplegar y proveer nuevos servicios de manera más rápida respecto a las implementaciones de servicios hechas sobre los sistemas de conmutación tradicional, esto principalmente da como resultado la libertad en el mercado de las telecomunicaciones (los monopolios dejan paso a nuevos actores que empujan hacia un mercado abierto de servicios) lo cual brinda una competencia cada vez mayor y al mismo tiempo ofrecer servicios más eficientes y de gran calidad [27] [28].



Durante el desarrollo de los servicios para la *Red Inteligente*, quedó al descubierto que estos implementan muchas funcionalidades en común. Por lo tanto se comenzó por identificar un grupo genérico de componentes para posteriormente ser usados en la construcción de nuevos servicios, por ejemplo algunos componentes utilizados para la autenticación, otros para la interacción con el usuario, otros destinados al traslado de llamadas y así sucesivamente [27] [28].

Para estos casos, un diseñador del servicio puede hacer uso de estos componentes o hacer uso de una combinación de muchos de ellos para crear un nuevo servicio, estos componentes han sido llamados los Bloques Independientes del Servicio (*Service Independent Building Blocks - SIBs*²³) [27] [28].

La lógica de servicio resultante puede ser cargado en la red sobre uno de los Punto de Control de Servicios (*Service Control Points – SCP's*)²⁴ y el servicio inmediatamente quedará disponible para los usuarios de la red [27].

2.1.1.2 Arquitectura de la Red Inteligente

La *Red Inteligente* está definida por los estándares a través de un modelo conceptual²⁵ que incluye cuatro niveles llamados planos dentro del mencionado modelo: el de servicios, el funcional global, funcional distribuido y el físico [21].

En términos generales, dentro del modelo conceptual se definen dos tipos de esquemas que se especifican en forma de arquitectura: la arquitectura física y la arquitectura funcional [21][27].

La arquitectura física muestra como es la interacción de los componentes tanto físicos como lógicos de la red y la arquitectura funcional muestra la distribución de las funciones para la provisión del servicio [21] [27].

Para este proyecto, el gran interés sobre esta arquitectura se centra en las entidades ofrecidas para la creación de servicios, es por esta razón que solamente se tiene en cuenta la arquitectura funcional de la *Red inteligente*.

2.1.1.2.1 Arquitectura Funcional de la Red Inteligente

Se enumeran a continuación las nueve entidades funcionales, unas encargadas de la ejecución del servicio y otras orientadas a la creación y gestión de servicios.

Para la ejecución del servicio se tienen las siguientes entidades funcionales:

Función Agente de Control de Llamada (*Call Control Agent Function - CCAF*): La función agente de control de llamada puede denominarse como el terminal con el cual el usuario interactúa con la red, en donde se otorgan las capacidades de acceso para el usuario. Su implementación depende de la red de acceso en concreto a utilizar [21] [30].

²³ SIB es una especie de subrutina software que consiste de unas simples instrucciones y que constituye el bloque más pequeño dentro de un servicio [28].

²⁴ SCP es un sistema con bases de datos que son consultadas en tiempo real y que contiene aplicaciones específicas para los diferentes servicios demandados [29].

²⁵ Modelo visual de un sistema que ilustra las interconexiones de los componentes del modelo.



Función de Control de Llamada (Call Control Function -CCF): La función de control de llamada es la encargada de prestar las capacidades de conmutación básica de un sistema de conmutación (establecer, manipular, liberar llamadas y conexiones) y además de la activación del servicio. Esta función debe ir acompañada de otro objeto llamado la función de conmutación del servicio (*Service Switching Function- SSF*), encargada de dar soporte en la interacción del control del servicio y además de ayudar al CCF en procesos de reconocimiento para activación del servicio [21] [31].

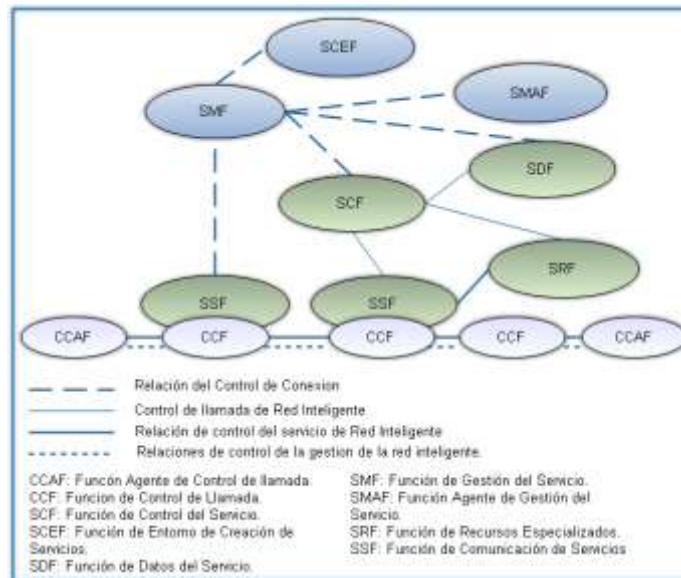


Fig. 10. Funciones de la red inteligente y relaciones funcionales para el CS-1.
 Tomado de [21].

Función de Conmutación de Servicio (Service Switching Function - SSF): La función de conmutación de servicio trabaja en conjunto con la función de control de llamada CCF como se comentó anteriormente, provee funciones de reconocer activadores²⁶ e interactuar con el control del servicio. En la Figura 10 se observa que estas dos entidades funcionales, por definición siempre deben estar juntas dentro de la arquitectura funcional de la *Red Inteligente* [21] [30].

Función de Control de Servicio (Service Control Function - SCF): La función de control de servicio es la encargada de procesar y ejecutar la lógica del servicio. Provee capacidades de influenciar en el proceso de procesamiento de llamadas a las SSF / CCF y otras de las entidades funcionales encargadas de la ejecución de servicios para realizar acciones específicas. Provee los medios necesarios para introducir nuevos servicios y características que no son propias de los sistemas de conmutación [21] [32].

Función de Recursos Especializados (Specialized Resource Function – SRF): La función de recursos especializados en términos generales son funciones que se encargan de facilitar la interacción entre el usuario final y el control del servicio. Estas funciones se ejecutan en tiempo real, tales como la reproducción de anuncios, recolección de la información desde el usuario (a través de tonos de voz o tonos *DTMF*), conversión de texto a voz y también es responsable de conversión de cierto tipo de protocolo [21][32].

²⁶ Por el término reconocer activadores, se entiende que es el responsable en determinar si un servicio de red debe ser invocado, y si es así, enviar los mensajes pertinentes para su activación [32].



Función de Datos del Servicio (*Service Data Function - SDF*): La función de datos del servicio es la encargada de extraer o actualizar la información necesaria relacionada con un servicio. Estas funciones se asemejan a las consultas que se hacen a una base de datos, todo esto orientado hacia la *SCF* o a otra *SDF* [32].

En cuanto a la gestión y creación de servicios se tienen las siguientes entidades funcionales:

Función de Entorno de Creación de Servicios (*Service Creation Environment Function - SCEF*): Esta función es la encargada de desarrollar y de probar la lógica del servicio. Consiste en una serie de funciones encargadas de la creación, verificación y prueba de nuevos servicios de *Red Inteligente* [21][31].

Función de Gestión del Servicio (*Service Management Function - SMF*): Luego que se tiene la lógica del servicio desarrollada sobre el *SCEF*, esta función se encarga de desplegar la lógica sobre las entidades funcionales de ejecución del servicio, por otro lado está encargada de administrar estas entidades funcionales para la personalización del servicio de acuerdo a información por parte del usuario y hacer la recolección de datos para procesos de facturación y creación de estadísticas de ejecución del servicio [21] [30].

Función de Agente de Gestión del Servicio (*Service Management Agent Function - SMAF*): Proporciona una interfaz hacia las anteriores funciones de gestión del servicio. Entre las operaciones contempladas en dicha interfaz se incluye la comprobación de la autorización pertinente, que es necesaria para manipularlas [21] [31].

Las anteriores entidades funcionales representan la primera etapa de estandarización para la *Red Inteligente* agrupados en el conjunto de capacidades 1 (*Capability Set -1 (CS-1)*), donde se da soporte inicialmente a un rango de servicios para algunos tipos de redes.

Para el conjunto de capacidades 2 (*Capability Set -2 (CS-2)*), se hace extensión de las recomendaciones existentes en el *CS-1*. Básicamente se han removido algunas limitaciones que se tenían en el conjunto de capacidades anterior, se adiciona la gestión de los servicios y el ambiente para la creación de servicios. Además se continúa adoptando el Modelo Conceptual de *Red Inteligente* encargado de hacer la descripción de la arquitectura de la red [27].

En la recomendación correspondiente al *CS-2*, se han introducido nuevas entidades funcionales:

Función de Acceso Inteligente (*Intelligent Access Function, IAF*): La función principal en gran parte es la conversión de protocolos, ya que se tienen funciones que ofrecen capacidades para proveer acceso a *Redes Inteligentes* desde redes que no sean de este tipo [21].

Función de Servicio no Relacionada con la Llamada (*Call Unrelated Service Function - CUSF*): La función de servicio no relacionada con la llamada, junto con la *SSF* y la *CCF*, provee una serie de elementos para lograr la interacción relacionando con la llamada de los usuarios y el procesamiento del servicio. Además de esto, ofrece funciones necesarias para la interacción entre la Función de Agente de Usuario de Control del Servicio (*Service Control User Agent Function - SCUAF*) y la función de Control de Servicio (*Service Control Function - SCF*) [21][33].



Función de Agente de Usuario de Control del Servicio (*Service Control User Agent Function – SCUAF*): La función de agente de usuario de control de servicio ofrece la interfaz entre el usuario y la Función de Servicio no Relacionada con la Llamada (*CUSF*) [21][30][33].

Con el conjunto de capacidades 3 (*CS-3*) se consigue ampliar el ámbito de aplicación permitiendo múltiples puntos de control y añadir servicios de red para los proveedores que tengan abonados con servicios netamente orientados hacia los conjuntos de capacidades *CS-1* y *CS-2*. Otras mejoras introducidas con el *CS-3* son el soporte de servicios suplementarios de la Red Digital de Servicios Integrados (*Integrated Services Digital Network - ISDN*) y la portabilidad de número²⁷ [34] [35] [36].

Los servicios del *CS-3* son un súper conjunto del *CS-2* y requiere las mismas entidades funcionales y los diferentes tipos de interfaces como las del conjunto de capacidades 1 y 2 [35]

El *CS-4* mejora las capacidades del *CS-3*, además ofrece soporte para la interoperabilidad con servicios en redes basadas en protocolo de Internet (*IP*) y servicios suplementarios [36].

2.1.1.3 Creación de servicios para la *Red Inteligente*

En el entorno de la *Red Inteligente*, se definen los servicios como una oferta comercial autónoma constituida por una o más Características de Servicios²⁸ (*Service Features – SF*) de manera modular [21].

En los inicios de la *Red Inteligente*, el *CS-1* admitía servicios diseñados acorde a las capacidades de esta arquitectura, pero se consideró que sus funcionalidades puedan dar un salto a versiones posteriores de Conjuntos de Capacidades. Esto beneficia a los proveedores de servicios, ya que se pueden hacer diseños pensando en el futuro con servicios con características diferentes a los que el *CS 1* tenía normalizados [21].

La independencia del servicio, característica que ofrece la arquitectura de Red Inteligente proporciona que cualquier servicio orientado a la red pueda ser desplegado sobre esta arquitectura. Esto permite tener una amplia visión de servicios de valor añadido ofrecidos sobre cualquier tipo de red de transporte [21].

Se describe a continuación los procesos que intervienen en la creación de un servicio para la Red Inteligente. Esto sólo es el producto resultante del Entorno de Creación de Servicios y muestra cómo este modulo hace la implementación de manera fácil, rápida y eficiente.

Inicialmente los bloques para construcción de servicios están expresados en un lenguaje formal y pueden ser invocados en cualquier momento. El usuario utiliza el editor gráfico para adicionar bloques de construcción a la imagen y relacionarlos entre sí. Cada bloque de construcción tiene una serie de atributos que pueden ser modificados usando la caja de propiedades asociada a cada uno de estos [21].

²⁷ La portabilidad de número se refiere a la capacidad de cambiar su proveedor de servicios de telecomunicación, manteniendo su número de teléfono actual.

²⁸ Características de Servicio (SF) es un aspecto específico de un servicio que puede utilizarse también junto con otros servicios/características de servicio como parte de la oferta comercial. La caracterización de los servicios y las características de los servicios sirven para identificar capacidades independientes del servicio que se requieren para construir y/o personalizar servicios por los usuarios o los operadores de red [21].

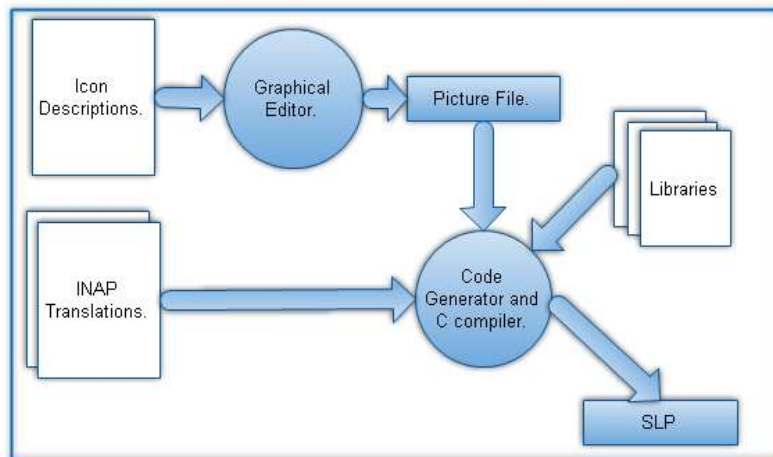


Fig. 11. Creación de servicios para la red inteligente.
Tomado de [27].

Una vez la imagen esta completa, esta se guarda en una imagen de archivo intermedio (*Picture File*) y luego es procesada por un generador de código. Este usa la información de traducción, que esta expresada en un lenguaje formal, y genera código en C, teniendo en cuenta la API soportada por la plataforma de la Red Inteligente. Finalmente, una serie de librerías de la plataforma y de la red específica son asociadas y permiten la creación definitiva de la Lógica del Programa del Servicio [21].

La lógica del programa del servicio final está lista para las pertinentes pruebas *off-line* y posterior despliegue sobre un nodo de la Red Inteligente. Las traducciones pueden ser cambiadas de manera fácil para soportar nuevas variantes de *INAP*²⁹ y agregar nuevas capacidades de red. La lógica del programa del servicio puede estar separada en diferentes procesos *UNIX*³⁰ y, por lo tanto, adicionar nuevas funcionalidades en caliente sobre un nodo el cual no tiene efecto sobre los demás servicios existentes.

2.1.2 CAMEL (CUSTOMIZED APPLICATIONS FOR MOBILE NETWORK ENHANCED LOGIC)

Cuando se dio inicio al desarrollo del sistema global para las comunicaciones móviles (*Global System for Mobile Communications- GSM*) a mediados de 1980, la *Red Inteligente* se usaba netamente en redes fijas, como la Red Telefónica Pública Conmutada y la Red Digital de Servicios Integrados [39].

La necesidad de tener servicios orientados a redes fijas y de proveer servicios más avanzados para la red *GSM*, dio como resultado la adaptación de estándares existentes de la Red Inteligente en la red *GSM* [32] [39].

De lo anterior, la solución a plantearse debe tener en cuenta dos aspectos [39]:

²⁹ *Intelligent Network Application Part (INAP)* es un protocolo para señalización usado en la arquitectura de red inteligente, parte de la familia de protocolos SS7.

³⁰ Procesos *UNIX* es un programa en ejecución y al objeto abstracto que crea el sistema para manejar el acceso de ese programa a los recursos del sistema.



- Los estándares existentes para la Red Inteligente fueron desarrollados para redes fijas.
- Los estándares existentes de la Red Inteligente no soporta la movilidad que ofrece la red *GSM*.

Para dar solución a este inconveniente, el Instituto Europeo de Estándares de Telecomunicaciones (*European Telecommunications Standards Institute- ETSI*) definió una serie de estándares denominado *CAMEL (Customized Applications for Mobile network Enhanced Logic - CAMEL)*, evolución natural de los estándares de la Red Inteligente donde fueron diseñados para trabajar sobre redes de tipo *GSM* o *UMTS* [40].

CAMEL describe una arquitectura similar a la de Red Inteligente pero aplicable al entorno de las Redes Móviles, la cual brinda a los usuarios la traslación de los servicios suscritos en su red de origen a la red visitada en la que se encuentra como itinerante (*roaming* entre diferentes redes) [32][39].

Muchos de los conceptos definidos para el *CS-1* y *CS-2* de la Red Inteligente fueron aplicados a los estándares *CAMEL*, es por eso que *CAMEL* por definición es un estándar de la Red Inteligente [39].

Los aspectos más importantes que distinguen la arquitectura de Red Inteligente *CAMEL* de los estándares tradicionales de Red Inteligente son [39]:

- *CAMEL* puede ser usado en redes *GSM* basados en equipos de múltiples proveedores.
- Los servicios *CAMEL* pueden ser usados para los usuarios en su propia red y para usuarios itinerantes.
- El estándar de Red Inteligente *CAMEL* se adapta a la red *GSM*.
- *CAMEL* ha tenido mayor desarrollo aplicado al entorno *GSM*.

2.1.2.1 Entidades *CAMEL*

En el ámbito de las comunicaciones móviles *GSM* y *UMTS*³¹ (*Universal Mobile Telecommunications System*), los servicios se encuentran definidos mediante la normalización de una serie de entidades funcionales y de relaciones entre ellas, concepto similar a los servicios de la Red Inteligente orientado a redes fijas. Esta idea facilita el diseño, despliegue y modificaciones de los servicios dependiendo de las necesidades del proveedor y la demanda de los usuarios [32].

El concepto de *CAMEL fase 1*, básicamente se sustenta en la separación de todos los elementos responsables del control de las llamadas y la lógica de procesado. La lógica reside en unas funciones centralizadas, que en este caso reciben el nombre de Funciones

³¹ *UMTS*: Es una de las tecnologías móviles de telecomunicaciones de tercera generación de banda ancha basada en la transmisión de paquetes de texto, voz digitalizada, video, y multimedia a velocidades mayores de 2Mbps.



de Control del Servicio *GSM* o *gsmSCF* (*GSM Service Control Functions*), equivalente a las Funciones de Control de Servicios de la Red Inteligente para las Redes Fijas [32].

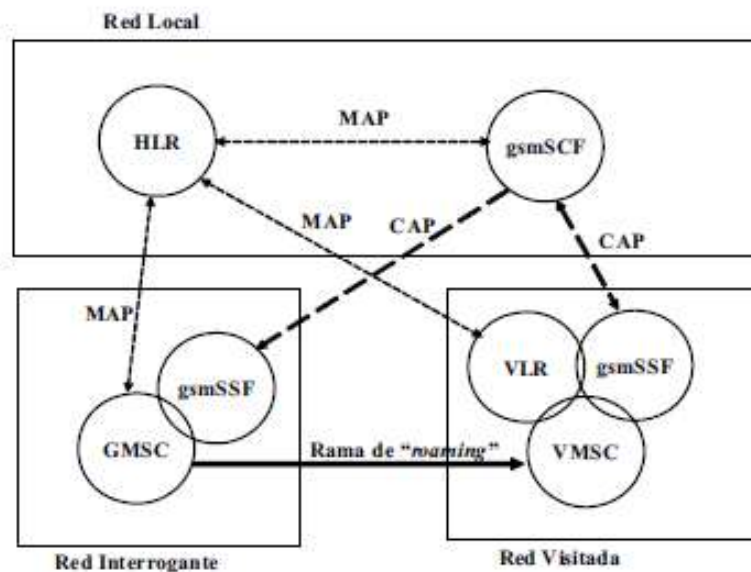


Fig. 12. Estructura general CAMEL fase 1.
Tomado de [33].

Por otra parte, las funcionalidades responsables de invocar la lógica y de actuar de acuerdo a sus instrucciones se denominan Funciones de Conmutación del Servicio *GSM* o *gsmSSF* (*GSM Service Switching Function*) o *gprsSSF* para el servicio *GPRS* (*General Packet Radio Service*)³², y desempeñan un papel homólogo a las funciones de conmutación del servicio de la Red Inteligente para las comunicaciones fijas [32].

Para la exploración de esta tecnología, se hace referencia solo para *CAMEL* fase 1. Para las fases posteriores solo es de interés comentar la adición de elementos que ofrecer mayores funcionalidades para los servicios a implementar. En la tabla 4 se presenta el resumen de las fases existentes y las características más relevantes para cada una de ellas.

2.1.2.2 Funcionamiento de los servicios *CAMEL*

Los servicios *CAMEL* son aplicables en las siguientes circunstancias [32]:

- Durante las actividades relacionadas con las llamadas originadas o destinadas a una estación móvil.
- Cuando se hace la invocación de los servicios suplementarios.
- Relacionado a las operaciones vinculadas a las sesiones *GPRS*.

³² *GPRS*: Es un servicios de comunicación inalámbrico basado en conmutación paquetes implementado como una extensión de *GSM* el cual brinda velocidades desde los 56 hasta los 114 *Kbps* y una conexión continua a Internet para teléfonos móviles y computadoras.



La ejecución de cada una de estas circunstancias se considera compuesto de una serie de eventos. Durante algunos de estos eventos se comprueba si se satisfacen determinados criterios: en caso afirmativo, se realiza una de las siguientes acciones [32]:

- Se envía una notificación a la lógica del servicio y se continua normalmente la ejecución:
- O bien, se suspende el servicio y se contacta con la lógica para solicitarle instrucciones sobre el modo de proseguir.

Las instrucciones proporcionadas pueden consistir en [32]:

- Continuar el proceso.
- Continuar el proceso, pero con información referente al mismo modificada,
- Conectar la parte llamante a una parte llamada específica,
- Liberar la llamada.

La lógica, por su parte, además de suministrar instrucciones relativas para el proceso de la llamada, realiza operaciones adicionales, como las siguientes [32]:

- Ordenar una interacción con el usuario mediante la generación de locuciones y la recopilación de tonos multifrecuencia.
- Actividades relacionadas con la tarificación.

2.1.2.3 Fases del estándar *CAMEL*

El estándar *CAMEL* está especificado en fases, cada una de estas han sido construidas en base a la anterior, existiendo hasta el momento 5 fases del estándar [39].

La fase 1 es un estándar simple, el cual fue adaptado a las redes basadas en *GSM*. Esta fase definía solamente control de servicios de llamada básica [39].

La fase 2 toma las capacidades de la fase 1, el principal enfoque fueron los servicios prepago.

La fase 3 incorpora gran variedad de servicios, incluido el servicio de mensajería corta (*Short Message Service - SMS*) así como con *GPRS* [39].

La fase 4 del estándar extiende las capacidades de la fase 3 agregando soporte de servicios multimedia pero no soporta tasas de contenido [39].

La fase 5 extiende las capacidades de la fase 4 de *CAMEL*, pero permite el soporte a servicios multimedia con calidad de servicios.



CAMEL Fase 1 GSM 2 - R1996	Las llamadas son encaminadas vía red local (home network), resultando la posibilidad de dos ramas de <i>roaming</i> : rama nacional y rama internacional.
CAMEL Fase 2 GSM 2+ -R1997	La red de <i>roaming</i> interroga a la red local por información y controla la llamada de prepago de acuerdo con esta información. De este modo solo se transfiere señalización a la red local, mientras la llamada en curso se maneja dentro de la red visitada. CAMEL 2 no ofrece soporte para servicios de datos o <i>roaming</i> GPRS.
CAMEL Fase 3 UMTS - R1999	Permite usar servicios de voz y de datos mientras se hace <i>roaming</i> . No soporta contenido ni algunos servicios SMS.
CAMEL Fase 4 UMTS R4	Extiende las capacidades de la Fase 3 agregando el soporte de servicios multimedia, pero no soporta tasas de contenido (velocidades o calidad de servicio).
CAMEL Fase 5 UMTS R5	CAMEL 5 supera el obstáculo anterior para dar un servicio de <i>roaming</i> sin fisuras (continuo).

Tabla 4. Resume fases CAMEL.
Tomado de [33].

2.1.3 OSA - OPEN SERVICE ARCHITECTURE (ARQUITECTURA DE SERVICIOS ABIERTOS)

La arquitectura de servicios abiertos (*OSA - Open Service Architecture*) es un estándar de *3GPP (3rd Generation Partnership Project)*, el cual define una arquitectura que permite a los operadores de la red y aplicaciones externas explotar la funcionalidad de la red móvil, a través de una interfaz abierta y estandarizada [40].

OSA provee una serie de *API's* para acceder a funciones de la red tales como la autenticación y autorización de usuarios. Las *API's* garantizan ser seguras, independencia con soluciones específicas elaboradas por cada proveedor y también independencia del lenguaje de programación orientado a objetos usado como por ejemplo *CORBA (Common Object Request Broker Architecture)*, *SOAP (Simple Object Access Protocol)*, entre otros [41].

Varios de los servicios como *VPN*, conferencia y muchos más pueden ser implementados con la ayuda de estas *API's* [41].

2.1.3.1 Arquitectura OSA

La arquitectura *OSA* define tres niveles [40] [41]:



Capa de Aplicación: Sobre esta capa se implementan todos los servicios de valor agregado. Estas aplicaciones serán usadas por la interfaz *OSA* para acceder a los recursos de la red móvil. Las aplicaciones pueden pertenecer a los operadores de la red o provenir de aplicaciones externas.

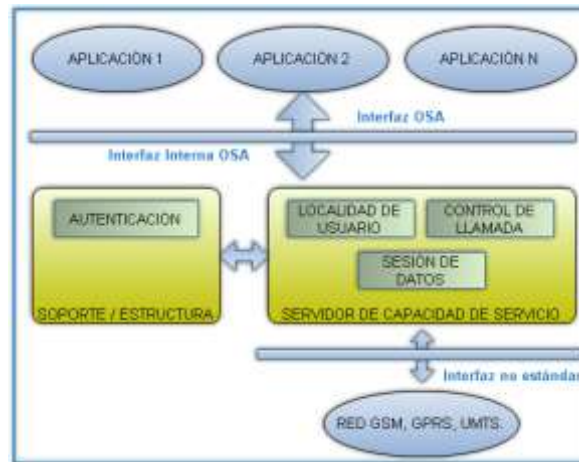


Fig. 13. Arquitectura OSA.
Tomado de [39].

Capa de Soporte: La capa de soporte o de estructura son el conjunto de aplicaciones básicas y necesarias usadas para el establecimiento de un ambiente apropiado con el objetivo de usar los servicios de la red. Como ejemplo la autenticación y, servicios de búsqueda entre otros.

Capa de servidores de capacidad de Servicio: En esta capa existe una interfaz *OSA* a disposición de las aplicaciones, la cual permite el acceso a las funciones del operador de la red móvil. Los servicios de la red que pueden ser accedidos a través de los servidores de capacidad de servicio son:

- **Control de Llamada.** Consiste en prestar las capacidades de conmutación básica de un sistema de conmutación (establecer, manipular, liberar llamadas y conexiones) y además de la activación de servicios.
- **Sesión de Datos.** Puede definir los eventos del modelo de conexión. Estos modelos son necesarios para conocer y modificar algunas características del procedimiento de reparación de la sesión de datos.
- **Localización de Usuario de Red.** Permite adquirir ubicación geográfica de la terminal móvil.
- **Estado del Usuario.** Permite adquirir información del estado del terminal, por ejemplo: actividad, encendido, apagado, etc.
- **Capacidad del Terminal.** Permite conocer las capacidades de terminales específicos.
- **Transferencia de Mensajes.** Permite la transferencia de mensajes a usuarios específicos, describiendo el procedimiento de eventos de transferencia.



2.1.4 PARLAY Y PARLAY/OSA

OSA es una arquitectura para redes de tercera generación, sin embargo la definición del 3GPP es muy abstracta y para efectos de implementación se basa en dos tecnologías adicionales: PARLAY y JAIN (*Java APIs for Integrated Networks*) [40].

Actualmente Internet ha creado oportunidades para implementar nuevas formas de comunicación y servicios, pero el modelo de Internet es totalmente opuesto a CAMEL y al concepto de Red Inteligente. Mientras que en la Red Inteligente los servicios son ofrecidos y gestionados por un operador de red, Internet abre la posibilidad para que cualquier entidad con infraestructura básica pueda ofrecer sus servicios. Parlay nace como la solución al problema de coexistencia entre estos modelos [40] [42] [43].

La solución de Parlay fue integrar las capacidades de la Red de Telecomunicaciones con las herramientas de desarrollo y aplicaciones del mundo de la industria de las telecomunicaciones mediante el uso de APIs basadas en tecnología abierta para permitir el acceso a la funcionalidad del núcleo de la red de una manera fácil y segura. Esto hace que las funcionalidades sean accesibles para una gran comunidad de desarrolladores, incluso ajenos al operador, cosa contraria a lo que se presentaba con la tecnología de Red Inteligente donde el grupo de desarrolladores de servicios estaba reducido al personal de cada empresa [40][43].

Las APIs Parlay pueden incluirse desde cualquier aplicación desarrollada con lenguajes de programación como C++, Java, C#, entre otros, las cuales cuentan con herramientas que ofrecen menores tiempos de desarrollo. Aparece así entonces, la facilidad y rapidez en la creación e implementación de los servicios deseados. Además de esto, la seguridad de las redes de los operadores no se ve amenazada debido a que los desarrolladores de servicios operan desde el exterior. Las APIs crean independencia entre las aplicaciones y los recursos de las redes de los operadores. Así, el núcleo de las redes puede ir evolucionando de manera independiente sin que las aplicaciones sean modificadas y estas a su vez pueden ser fácilmente migradas con portabilidad a otras tecnologías de red [43].

La primera edición de las especificaciones de Parlay se realizó en el transcurso de un año y su trabajo continuó bajo el nombre de Parlay 2 o Parlay fase 2. Alrededor del mismo tiempo en que el trabajo de Parlay 2 empezó (Junio de 1999), 3GPP y ETSI comenzaron a trabajar en APIs para el desarrollo de servicios en redes de tercera generación y redes fijas. Desde el principio 3GPP y ETSI han realizado su trabajo conjuntamente y concluyeron que las APIs Parlay podían ser usadas en el contexto en el que se encontraban trabajando. Debido a esto, las APIs Parlay más relevantes fueron incorporadas al trabajo 3GPP/ETSI para su estandarización en el contexto de lo que se llamaba OSA. Posteriormente el acrónimo OSA fue redefinido a Acceso de Servicios Abiertos OSA (*Open Service Access*). Además de las primeras APIs Parlay, se definieron en el trabajo 3GPP/ETSI unas nuevas y se produjeron nuevos protocolos de señalización de red [43].

Mucho del énfasis de ese trabajo fue empleado para juntar y alinear las APIs Parlay y las de 3GPP/ETSI. Debido a esto, estas organizaciones decidieron conformar una comunidad apuntando a su objetivo común: formaron el llamado *Joint API Group* y denominaron al resultado de su trabajo OSA/Parlay [43].



Conjuntamente al trabajo de *Parlay*, *3GPP* y *ETSI*, existe una colaboración informal a la denominada comunidad *JAIN* principalmente en el campo de las *API's* de control de llamadas.

En la actualidad estas tres organizaciones generan especificaciones totalmente alineadas entre sí y disponibles para quien las necesite [43].

Actualmente el grupo *Parlay* es conformado por compañías líderes de la industria de las telecomunicaciones, proveedores de servicios de Internet, desarrolladores de software, proveedores de dispositivos de red, departamentos de servicios, proveedores de aplicaciones de servicios además de grandes y pequeñas empresas [43].

2.1.4.1 Estructura del API

La *API OSA/PARLAY* consiste en dos grupos de interfaces [40] [44]:

- Interface de soporte (*Framework Interfaces*).
- Interfaces de servicios (*Services Interfaces*).

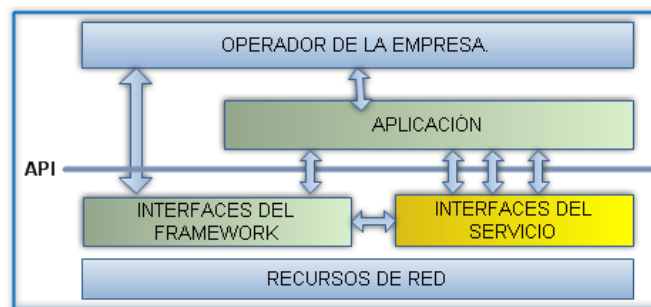


Fig. 14. Interfaz de Programación y Aplicación OSA/PARLAY.
Tomado de [44]

La interface de soporte brinda mecanismos básicos para el uso de funciones de red actuales. Incluyen, por ejemplo, autenticación y autorización para identificar la aplicación que quieren acceder a la *API*. Después de una autenticación exitosa, la función de descubrimiento puede ser usada para consultar información acerca de la disponibilidad de las funciones de la red.

Otras funciones realizan y evalúan la suscripción en línea del tipo de servicio o funciones de la red y otros acuerdos contractuales de suscripción de servicios.

El acceso al *framework* siempre es el primer paso para usar la *API OSA/PARLAY*. Después de esto, las interfaces de servicios (*Services Interfaces*) pueden ser usadas, tan rápido como la aplicación este autorizada [40] [44].

Las Interfaces de servicios permiten a las aplicaciones clientes acceder a las llamadas *SCF* (*Service Capability Features*). Ellos representan las funciones de la red disponibles que pueden ser usadas para implementar los servicios de telecomunicaciones para el usuario final.



La siguiente lista muestra una visión general de los componentes disponibles en la versión 3.0 de *Parlay* aprobado en diciembre de 2001 [40] [44]:

- Control de Llamada: Establecimiento y control de conexiones.
- Interacción con Usuario: Ejecución, Reporte, reconocimiento *DTMF*, envío de *SMS*, etc.
- Comportamiento de Usuario / Ubicación de Usuario: por ejemplo; Conmutación de llamadas, activación/desactivación, localización de la llamada.
- Control de Sesión de datos: por ejemplo el volumen de datos basado en la tarificación de *GPRS*.
- Capacidades de Terminal: para definir las propiedades del terminal.
- Mensajes Genéricos: Conversión de mensajes, conexión a correo electrónico, etc.
- Gestión de Conectividad: calidad de servicio real, etc.
- Cobro basado en el contenido: Tarificación basada en el contenido de la información.
- Gestión de cuenta: Gestión de tarjetas pre pagadas en redes móviles.

2.1.4.2 Arquitectura *OSA/Parlay*

La arquitectura de *OSA/Parlay* se presenta en la figura 15:

Aplicación Cliente: Es el software que necesita acceder a uno o más servicios a través de las *API's* residentes en el *Gateway* de *Parlay* para completar su función [43].

Gateway de *Parlay*: El *Gateway* de *Parlay* es el enlace que une las aplicaciones que utilizan el *API Parlay/OSA* con los elementos existentes en la red. Esta *Gateway* está bajo control del operador de la red o del proveedor del servicio, por ella pasan cada una de las interacciones que ocurren en cualquier aplicación que utilice el *API Parlay/OSA*. Esto significa que las aplicaciones están aisladas de los protocolos específicos usados dentro de la red por el operador, y la red puede evolucionar sin afectar aplicaciones y servicios existentes. Consta de los siguientes elementos:

- ***SCS y SCF:*** Los Servidores de Capacidades de Servicio o *SCS (Service Capability Server)* serán los que implementen las capacidades de servicio o *SCF (Service Capability Feature)* que mediante interfaces abiertos (*las APIs de OSA/Parlay*) permiten a las aplicaciones ejecutar la funcionalidad de los elementos de la Red de Telecomunicaciones subyacente. Un *SCS* puede implementar varios *SCF's*, aunque usualmente se habla de un *SCS* específico por *SCF* [43].
- ***Parlay/OSA Framework:*** Este *framework* permite que las aplicaciones cliente puedan descubrir las capacidades de servicio de la red de forma segura para el operador de telecomunicaciones. De esta manera, el operador permite abrir la red



de telecomunicaciones a aplicaciones externas proporcionando un entorno flexible y dinámico en el que se puedan incorporar nuevas capacidades de servicio. Los Servidores de Capacidades de Servicio registran en el *framework* características de las capacidades de servicio que implementan. Cuando el *framework* lo necesite podrá crear nuevas instancias de las capacidades de servicio usando las características registradas. Además, el *framework* permite la inclusión de capacidades de servicio no estandarizadas, lo que es crucial para la innovación y diferenciación de servicios. En realidad, podemos ver el *framework* en el entorno de *OSA/Parlay* como una capacidad de servicio especial con una interfaz definida que permite autenticar y autorizar a las aplicaciones al tiempo que facilita el registro dinámico de las diferentes capacidades de servicio [42] [43].

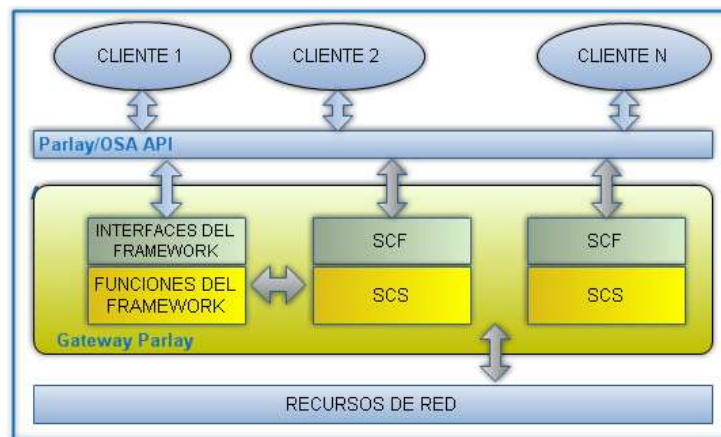


Fig. 15. Arquitectura de OSA/PARLAY.
Tomado de [44].

Recursos de la Red: Agrupan los nodos internos de redes fijas, móviles y de redes IP. *OSA/Parlay* no define protocolos nuevos dentro de estos elementos sino que utiliza los existentes en dichas redes. En estos recursos de la red se encuentran las Capacidades de la Red accedidas a través de las APIs *OSA/Parlay* [43].

2.1.4.3 Ventajas de OSA/PARLAY

A continuación se mencionan las ventajas más sobresalientes de *OSA/PARLAY*.

Creación rápida de servicios: Las aplicaciones desarrolladas con *OSA/Parlay* pueden ser operadas por terceras partes distintas al operador de la red de telecomunicaciones. Esto abre la red del operador para ser usada por desarrolladores externos, lo que beneficia la aparición de entornos abiertos de creación de servicios permitiendo menores tiempos de desarrollo y la posibilidad de un número mayor de desarrolladores potenciales [43].

Independencia en la red: El API *Parlay/OSA* se diseñó para que en todas las ocasiones fuera independiente de los niveles inferiores de red. Para la aplicación es independiente que tipo de protocolo o de estructura esté implantada, ya que tiene la posibilidad de adaptarse de manera transparente a cada una de ellas [42].

Independencia del vendedor: Como consecuencia de la independencia de red y tecnología de la API *Parlay/OSA*, los vendedores se tornan independientes también. Los proveedores



de servicios se benefician de tener un simple conjunto de *API's* que sean soportados en múltiples plataformas, esto provee flexibilidad adicional en el desarrollo [42].

Procedimiento de llamadas a capacidades de servicio independiente del lenguaje de programación: *OSA/Parlay* permite la invocación remota de las capacidades de servicio desde las aplicaciones externas y viceversa de una manera independiente de los sistemas operativos y de los lenguajes de programación. Esto lo hace a través de las tecnologías *middleware CORBA* y *Web Services*, ambas disponibles para diferentes sistemas operativos ofreciendo la posibilidad de utilizar múltiples lenguajes de programación [43].

Adaptación de aplicaciones al medio: *OSA/Parlay* permite la adaptación de las aplicaciones al contexto en el que se encuentra el usuario que solicita el servicio. Mediante el uso de las capacidades de servicio que permiten obtener la localización, disponibilidad e información del terminal que está usando un usuario determinado, se puede adaptar la funcionalidad ofrecida por el servicio dependiendo de las situaciones particulares en que este es invocado, así como tener en cuenta la funcionalidad de la red usada en el mismo [43].

2.1.4 JAIN – (Java APIs for Integrated Networks)

JAIN son un grupo de *API's Java* que facilitan el desarrollo de aplicaciones y servicios de telecomunicaciones de nueva generación sobre la plataforma *Java*. *JAIN* busca que el proceso de creación de servicios de redes de telecomunicaciones sea similar al proceso de desarrollo software, permitiendo que el desarrollo y creación de servicios de red esté al alcance de los proveedores de aplicaciones y empresas dedicadas al desarrollo de software [45] [46].

JAIN usa el término de Redes Integradas indicando que cuenta con una serie de interfaces en *Java* para creación de servicios en las redes *PSTN*, redes de paquetes (como redes *IP* o *ATM*) y redes inalámbricas. *JAIN* permite principalmente la integración entre protocolos *IP* y protocolos de

Redes Inteligentes de manera que los diseñadores de servicios no tengan que preocuparse por los aspectos de red en el momento de crear servicios [45].

La iniciativa *JAIN* brinda portabilidad de servicios, convergencia y acceso seguro a las redes de telefonía e Internet:

La portabilidad de servicios. Consiste en que los servicios se pueden cargar en un solo lugar centralizado de la red, pero pueden ser accedidos en forma múltiple, desde cualquier parte y por cualquier medio [45].

Convergencia de red. Las aplicaciones y los servicios pueden ser invocados y ejecutados en cualquier red ya sea *PSTN*, conmutación de paquetes (*IP*, *ATM*, etc.) ó redes inalámbricas [45].

El acceso seguro de red. Permite que aplicaciones que residen fuera de la red puedan tener acceso directo a los recursos y demás elementos del *backbone* para llevar a cabo acciones o funciones específicas. Las oportunidades que surgen son inmensas, ya que se habla de



permitir que usuarios o desarrolladores puedan tener acceso en forma controlada a las funcionalidades e inteligencia que se tenga en las redes de telecomunicaciones [45] [46].

Las iniciativas de la comunidad *JAIN* están orientadas al desarrollo de especificaciones para *API's* en dos áreas [45] [46]:

1. Especificaciones de *API's* para protocolos, que permiten hacer interfaz con protocolos de señalización de las redes integradas (comunicaciones fijas, *SS7*, inalámbricas e *IP (Internet Protocol)*).

Dentro de las *API's* para protocolos, actualmente sólo se están desarrollando para *IP* y *SS7*.

Entre las *API's* para *IP* figuran *JAIN H.323*, *SIP (Session Initiation Protocol)* o *JAIN MEGACO (JAIN - Media Gateway Control Protocol)*, para el soporte *IP* de conferencias multimedia y la interconexión con redes *SS7*.

Las *API's* para *SS7 (JAIN ISUP (JAIN ISDN User Part), TCAP (Transaction Capabilities Application Part), MAP (Mobile Application Part) y JAIN INAP (JAIN Intelligent Network Application Part))* incluyen tres interfaces, a través de las cuales se modela el acceso a la torre de protocolos con una interacción basada en eventos.

2. Especificaciones de *API's* para aplicaciones, las cuales permiten la creación de servicios dentro de una estructura *Java*.

En el subconjunto de las *APIs* para aplicaciones, los servicios se abstraen de las redes y protocolos sobre los que funcionan, apoyándose en:

JCC (JAIN Call Control): Es un modelo básico de control de llamadas basado en *JTAPI*³³ (*Java Telephony API*), aunque aquí las llamadas son sesiones multimedia multiprotocolo y multiusuario.

JCAT (JAIN Coordinations and Transactions): Complementa a *JCC*, proporcionando mayor control y un modelo de señalización más completo.

SPA (Service Provider APIs): Son las encargadas de la implementación de la especificación *Parlay* para facilitar el desarrollo de servicios por terceros. La principal diferencia de estos servicios con los que no se ejecutan sobre *Parlay*, es que los primeros tienen restringidas las capacidades de la red a las que tienen acceso por motivos de seguridad e integridad.

En la figura 16 se muestra que las aplicaciones *JAIN* se posicionan directamente sobre las *API's* de la capa de adaptación de protocolos *JAIN*, o sobre *JCC*, o bien sobre la *API JAIN Parlay*. Los servicios creados necesitan, a parte del modelo de control de llamadas, otras funcionalidades como puedan ser *JDBC (Java Database Connectivity)*, conversión texto-voz, *JNDI API (Java Naming and Directory Interface)*, etc [46].

³³ *JTAPI*: API basada en Java para desarrollo de aplicaciones de telefonía.

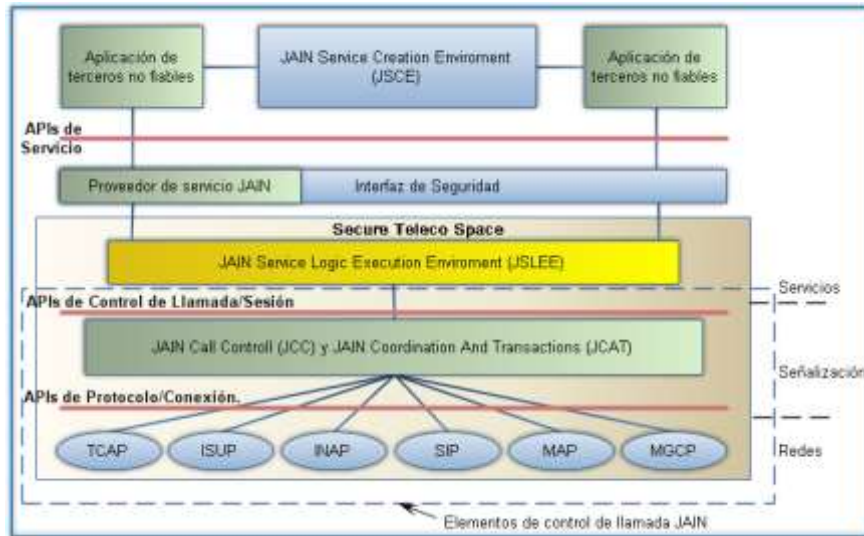


Fig. 16. Interfaz de Programación y Aplicación JAIN.
Tomado de [40].

Las aplicaciones se ejecutarán sobre un entorno denominado *SLEE (Service Logic Execution Environment)*, el cual se basa en un modelo de componentes que soporta una implementación distribuida. *SLEE* proporciona, además, portabilidad en las transacciones, persistencia, balance de carga, seguridad, pool de conexiones, entre otras [46].

El proveedor *JAIN* de acceso de servicio es una Interfaz de acceso para las funciones de aplicación a la red, que necesitan seguridad al acceder y de control específico [40].

2.1.5 IMS - IP Multimedia Subsystem

IMS es un conjunto de especificaciones que describen una arquitectura de Red de Nueva Generación para la implementación de telefonía y servicios multimedia basados en el protocolo *IP*. Define una completa arquitectura y una estructura soportada sobre una red *IP*, las sesiones aplicativos en tiempo real y las que se ejecutan en un entorno de tiempo no real. Además, la arquitectura ha sido definida independiente del tipo de red de acceso, lo cual facilita a cualquier tecnología de acceso ser utilizada como medio de transporte de los datos del usuario al entorno *IMS* sin afectar la funcionalidad de la red como tal [47] [48].

IMS inicialmente fue definido por el *3GPP (3rd Generation Partnership Project)*. La primera versión fue publicada en el *Release 5* del *3GPP*, donde se define al protocolo de inicio de sesiones *SIP (Session Initiated Protocol)* definido por *IETF (Internet Engineering Task Force)* como el protocolo principal. En las publicaciones *Release 6* y *7* del *3GPP* se han añadido características como la presencia y gestión de grupos, interoperabilidad con *WLAN*, sistemas basados en conmutación de circuitos y acceso de sistemas fijos de banda ancha [47].

Otro ente de estandarización, el *3GPP2 (3rd Generation Partnership Project 2)* definió su propio estándar para *IMS*. La publicación inicial de la especificación *IMS* del *3GPP2* adopta en gran medida la publicación *Release 5* del *3GPP* [47].

Las dos arquitecturas definidas son muy similares pero no son exactamente las mismas. *3GPP2* añadió ajustes de acuerdo a sus problemas específicos. Sin embargo, el propósito de



ambas organizaciones es asegurar que las aplicaciones *IMS* trabajen de manera consistente a través de diferentes infraestructuras de red [47].

Adicionalmente al *3GPP* y al *3GPP2*, *OMA (Open Mobile Alliance)* juega un papel importante en la especificación y desarrollo de estándares de servicios *IMS*. Los servicios definidos por *OMA* están construidos en la cima de la infraestructura *IMS*, como mensajería instantánea, servicio de localización, servicio de presencia, y servicio para la gestión de grupos entre otros [47].

Las principales características tecnológicas de *IMS* son [49]:

- La arquitectura *IMS* permite usar los servicios *IP* multimedia sobre cualquier red de acceso (fija, móvil o inalámbrica) que pueda proporcionar conectividad *IP*. Para los sistemas telefónicos antiguos, existen pasarelas que ofrecen el ingreso a esta infraestructura realizando la paquetización de las llamadas de voz y como consecuencia permitiendo el manejo de las mismas a través de interfaces estandarizadas.
- La arquitectura *IMS* tiene la capacidad de admitir y ofrecer servicios a abonados de otros operadores que emplean la misma tecnología y con quien se tiene acuerdos de negocio pertinente.
- La señalización de *IMS* se realiza mediante el protocolo *SIP* el cual aporta las funciones para el registro, establecimiento, modificación y finalización de las sesiones entre diversos dispositivos. Puesto que no todos los dispositivos son capaces de soportar los mismos servicios, al establecer la sesión se hace la negociación de las características de ésta mediante el protocolo *SDP (Session Description Protocol)*. Mediante *SDP*, los extremos de una sesión pueden indicar sus capacidades multimedia y definir el tipo de sesión que desean mantener. En este intercambio de señalización se hace la negociación de la calidad de servicio (*Quality of Service - QoS*), tanto durante el establecimiento como durante la sesión en curso.
- El transporte de red es realizado mediante *IPv6* en vez de *IPv4*, ya que la migración a *IPv6* está siendo desplegado en Internet y existen ya muchas empresas e instituciones que lo emplean internamente. De este modo, el *3GPP* prefirió dar compatibilidad hacia la situación más avanzada tecnológicamente.
- La provisión de servicios multimedia es realizada por protocolos del *IETF*³⁴:

SIP (Session Initiation Protocol): *SIP* permite establecer, controlar y terminar sesiones multimedia independientes del protocolo de transporte y del tipo de sesión que esté siendo establecida, por esta razón *SIP* ha sido elegido por el *3GPP* para el control de sesión y el control de servicio en *IMS*.

El protocolo *SIP* utiliza un modelo cliente/servidor por lo cual se tienen dos tipos de mensajes, solicitudes del cliente al servidor y respuestas en el sentido contrario. Los dos tipos de mensajes están conformados por una línea con el método o código de respuesta *SIP*, uno o más encabezados, una línea vacía indicando el fin de los

³⁴ *IETF*: Es una organización internacional abierta de normalización que regula las propuestas y los estándares de Internet



encabezados, y un cuerpo del mensaje que es opcional. En el RFC 3261 se definen 6 métodos SIP: *REGISTER* para registrar información de contacto, *INVITE* para establecer una sesión, *ACK* para dar respuesta a las solicitudes, *CANCEL* para abandonar sesiones en establecimiento, *BYE* para terminar sesiones establecidas y *OPTIONS* para interrogar las capacidades del servidor.

SDP (*Session Description Protocol*): Permite la descripción de sesiones multimedia, siendo útil para invitaciones, anuncios, y cualquier otra forma de inicio de sesión. Los mensajes *SDP* se pueden transportar mediante el protocolo *SIP* por lo cual es utilizado en la arquitectura de red *IMS*.

Diameter: Diseñado como un conjunto de mejoras a *RADIUS*. Diameter proporciona un marco de autenticación, autorización y contabilidad (*AAA, Authentication, Authorisation, y Accounting*), para aplicaciones de acceso a redes, movilidad en redes *IP* y *roaming*.

RTP (*Real Time Protocol*) y RTCP (*Real Time Control Protocol*): se utilizan para el transporte de flujos *IP* multimedia del plano de usuario.

RSVP (*Resource Reservation Protocol*) y DiffServ (*Differentiated Services*): Protocolos usados para asegurar la calidad de servicio extremo a extremo.

MEGACO (*Media Gateway Control Protocol*): Protocolo diseñado para el control de los *Media Gateway* y el control de las puertas de enlace para el soporte de llamadas de voz entre redes *RTPC-IP* o *IP-IP*.

- La arquitectura *IMS* emplea una arquitectura de seguridad independiente de la red de acceso, proporcionando mecanismos de autenticación, acuerdo de claves, confidencialidad, integridad y privacidad a través del establecimiento de asociaciones de seguridad *IPsec*³⁵ que protegen los mensajes de señalización *SIP* entre el *P-CSCF* y el terminal.

2.1.5.1 Arquitectura básica de *IMS*

En la figura 17 se presenta la arquitectura *IMS* definida en capas según *NGN*. Además se muestran las entidades que componen esta arquitectura [47].

Transport layer. La capa de transporte consiste en el conjunto de elementos necesarios que componen el *backbone IP* y las redes de acceso. Esta capa provee las interfaces necesarias para todo tipo de redes y dispositivos que requieran acceder a *IMS* [47].

Control layer: La capa de control, como su nombre lo indica, controla la autenticación, enrutamiento y distribución del tráfico *IMS* entre la capa de transporte y la capa de servicio [47].

³⁵ *IPSEC*: Es un conjunto de protocolos cuyo objetivo es el de hacer seguras las comunicaciones sobre el Protocolo de Internet (*IP*) autenticando y/o cifrando cada paquete *IP* dentro de un flujo de datos.



Sobre esta capa se encuentra el núcleo de *IMS* el cual contiene un componente vital: el *CSCF (Call Session Control Function)* que facilita la correcta interacción entre servidores de aplicaciones, servidores de medios y el *HSS (Home Subscriber Service HSS)* [47].

Service layer: En la cima de la arquitectura *IMS* se encuentra la capa de servicios, compuesta por servidores de aplicaciones *SIP* los cuales albergan la lógica de las aplicaciones y proveen los servicios a los usuarios [47].

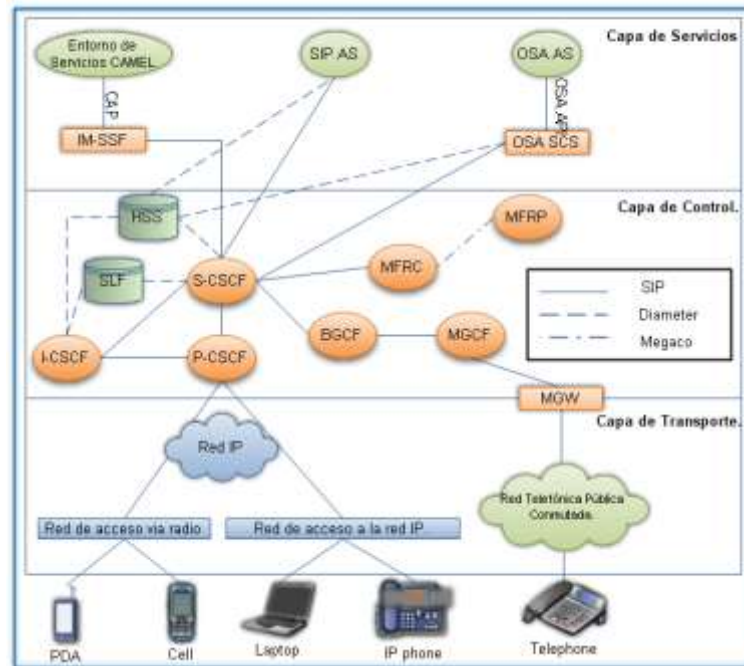


Fig. 17. Arquitectura IMS y capas de red según NGN.
Tomado de [47].

2.1.5.2 Entidades de IMS

A continuación se hace la descripción de las entidades que conforman la arquitectura *IMS* presentadas en la figura 17.

User Equipment: Se trata de una aplicación que se ejecuta sobre el equipo terminal y su función básica es la de realizar y recibir mensajes *SIP* [48].

HSS (Home Subscriber Server): La entidad *HSS* se trata de un repositorio de datos el cual almacena información relacionada con los usuarios y los servicios a los cuales está suscrito. Los principales campos que se encuentran almacenados en esta entidad son: identidades del usuario, informaciones de registro, parámetros de acceso, así como los datos que permiten la invocación de los servicios del usuario. El *HSS* interactúa con las entidades de la red a través del protocolo *Diameter* [48].

SLF (Subscriber Location Function): Si hay más de un *HSS* en la red, un *SLF* puede indicar la dirección del usuario a los correspondientes servidores *HSS*.

CSCF (Call Session Control Function): Su función es procesar los mensajes de señalización para controlar la sesión multimedia de los usuarios. Ejecuta tres roles diferentes: el de *Proxy-CSCF (P-CSCF)*, el *Interrogating-CSCF (I-CSCF)* y el *Serving-CSCF (S-CSCF)* [49].



- **P-CSCF (Proxy-CSCF):** Es el primer punto de contacto del terminal con *IMS* (en el contexto de la señalización). El *P-CSCF* actúa como un servidor *proxy*, en donde todos los mensajes de señalización *SIP* que se originan en el terminal o con destino a él atraviesan esta entidad. Incluye diversas funciones, algunas relacionadas con la seguridad. Establece el número de asociaciones seguras *IPSec* hacia el terminal. Una vez que esta entidad ha autenticado al usuario, lo comunica a las demás entidades de *IMS* para la ejecución de posteriores procesos, también verifica que las peticiones *SIP* sean conformes a la norma, genera información acerca de la tarificación entre otras funciones [50].
- **I-CSCF (Interrogating-CSCF):** Es el punto de contacto dentro de la red de un operador para todas las conexiones provenientes del exterior destinadas a sus suscriptores o un usuario *roaming* ubicado actualmente dentro su área de servicio [51].

Cuando el *I-CSCF* recibe una petición *SIP* desde otra red, obtiene del *HSS* la dirección del *S-CSCF* y envía la petición o respuesta *SIP* a esa entidad, es decir, ayuda a otros nodos a determinar el siguiente salto de los mensajes y a establecer un camino para la señalización, realizando funciones de enrutamiento [51].

Genera información para facturación y opcionalmente ejecuta funciones de ocultamiento de la topología de la red ante redes externas, de forma que los elementos ajenos a *IMS* no puedan conocer cómo se gestiona la señalización internamente [51].

- **S-CSCF (Serving CSCF):** Esta entidad es la encargada de coordinar con otros elementos de la red, el control de las sesiones. A cada usuario registrado en *IMS* se le asigna un *S-CSCF*, el cual se encarga de enrutar las sesiones destinadas o iniciadas por el abonado [48] [51].

Esta entidad procesa las solicitudes de registro, permite que la información esté disponible a través del *HSS*, notifica a los suscriptores de cualquier cambio en los procesos, establece, modifica y termina una sesión, supervisa la llamada y proporciona seguridad [51].

Interactúa con plataformas de servicios o servidores de aplicación para el soporte de servicios, provee a los terminales información de eventos de servicio, como por ejemplo mensajes de notificación de nuevas versiones de software, avisos con la localización de los recursos, notificación del saldo, etc. [51].

BGCF (Breakout Gateway Control Function): Esta entidad determina el camino a seguir de las llamadas con destino a redes de circuitos conmutados. Si el *BGCF* decide enrutar la llamada a una red *IMS* externa, enviara la señalización de la llamada a la entidad *I-CSCF* de esa red, si el enrutamiento debe hacerse en la misma red, el *BGCF* hace uso de la Función de Control de Pasarela de Medios (*MGCF*) para que haga la conexión del núcleo de red a la red telefónica pública conmutada y permita establecer el camino hacia el destino de la llamada [51].



MGCF (Media Gateway Control Function): Esta entidad tiene implementada las funciones de control para la interconexión de *IMS* con las redes de circuitos conmutados haciendo traducciones de señalización *SIP* a protocolo *SS7* y viceversa. Determina el paso siguiente dependiendo del número identificador para las llamadas provenientes de las redes de circuitos conmutados [51].

IMSMGW (IP Multimedia Subsystem - Media Gateway Function): Es el punto de interconexión del *Subsistema IP Multimedia* con las redes de circuitos. Realiza el control de portadora, procesamiento de carga útil, interactúa con el *MGCF* para establecer las capacidades que ofrecerá al terminal, posee y maneja recursos como canceladores de eco [48][51].

MRF (Multimedia Resource Function): Provee los recursos multimedia a la red local y se divide en dos partes:

- **MRFC (Multimedia Resource Function Controller):** Esta entidad interpreta la información proveniente de un servidor de aplicaciones o un *S-CSCF*, maneja los recursos del *MRCF*, genera datos de facturación [51].
- **MRFP (Multimedia Resource Function Processor):** Controla y monitorea las condiciones de las comunicaciones, origina y envía mensajes de conexión a todos los participantes de una sesión, procesa los mismos, realiza conversión de un códec a otro de audio y video, manejo de derechos de acceso a una conferencia [51].

Application Server (AS): Provee la lógica y las funciones relacionadas con cada servicio a los usuarios. Estos comprenden juegos multi-usuario, videoconferencia, mensajería y entre otros.

IMS define tres tipos de servidores de aplicaciones: Servidores de aplicaciones *SIP*, servidores de aplicaciones *OSA* y servidores de aplicaciones *CAMEL* [51].

- Los servidores *SIP* se comunican directamente al nodo *S-CSCF* a través del protocolo *SIP*.
- Los servidores *OSA* requieren de la pasarela *OSA-SCS (OSA - Service Capability Server)*, donde por un lado actúa como un servidor de aplicación (conectando al *S-CSCF* con *SIP*) y por el otro como una interface entre el servidor de aplicaciones *OSA* y el *OSA/API*. Este servidor hereda todas las capacidades que ofrece *OSA*, especialmente la capacidad de acceder de manera segura a *IMS* desde redes externas [51].
- El ambiente de servicios *CAMEL* son el conjunto de servicios desarrollados para *GSM* y prestan servicio sobre *IMS*. La pasarela *IM-SSF* actúa por un lado como un *SSF (Service Switching Function)* enlazando el *gsmSCF* con un protocolo basado en *CAP (CAMEL Application part)* y por otro lado como un servidor de aplicaciones (conectando al *S-CSCF* con *SIP*) [51].



Interconnection Border Control Function (IBCF): Permite realizar las conexiones entre dos dominios, permitiendo comunicaciones entre aplicaciones *IPv6* e *IPv4* seleccionando la señalización apropiada de interconexión [48] [51].

Transition Gateway (TrGW): Controlada por la *IBCF*, es la encargada de la traducción de puerto y traducción de protocolo *IPv4/IPv6* asignando las direcciones de red correspondientes [48] [51].

2.1.5.3 Ventajas de IMS

A continuación se mencionan algunas de las ventajas que presenta *IMS* [51]:

- Facilita la introducción rápida de servicios a través de interfaces estandarizadas, promoviendo la innovación y el desarrollo de nuevas aplicaciones por otros proveedores.
- Es independiente de la red de acceso. Esta ventaja permite la integración de redes móviles con fijas, donde el control dentro de *IMS* lo proporciona la Función de Control de Sesión de Llamada *CSCF*.
- Permite la coordinación y sincronización efectiva de varios medios en una sesión o en varias sesiones a través del *CSCF*, brindando una experiencia más enriquecida al usuario al permitir que el suscriptor inicie varias sesiones o una sesión entre diferentes contactos por diferentes medios (audio, video, texto, etc.).
- Hace posible la interacción e integración de servicios. Por ejemplo un usuario que tiene el servicio *push to talk over cellular* y el de mensajería instantánea puede utilizarlos individualmente, ejecutarlos desde una misma aplicación en su terminal o combinarlos como quiera en una sesión.
- Las operaciones de manejo, mantenimiento y entrenamiento se facilitan ya que todas las aplicaciones siguen una misma arquitectura, reduciendo los costos de operación.
- La arquitectura está diseñada para una fácil escalabilidad y redundancia.

2.2 HERRAMIENTAS DE AUTORÍA Y PLATAFORMAS PARA DISTRIBUCIÓN DE CONTENIDOS *MHP* EN EL ENTORNO DE LA *TDi*

En esta sección se hace la exploración de algunas herramientas de autoría y plataformas para la distribución de contenidos *MHP* esto, para mostrar la existencia de diversas herramientas diseñadas para la generación de contenidos interactivos en el ámbito de la *TDi* que permiten la creación de aplicaciones de manera fácil e intuitiva sin necesidad de tener conocimiento sobre la codificación o la estructura.

Se muestra además la existencia de plataformas para la distribución de contenidos, tecnologías usadas para la emisión de datos junto a los contenidos de televisión en el sistema de televisión digital.



La elección de las herramientas aquí presentadas obedece a la exploración realizada por proyectos anteriores las cuales nos presentan tecnologías con características que pueden llegar a ser de gran utilidad para el proyecto.

2.2.1 HERRAMIENTAS DE AUTORÍA *MHP* PARA LA GENERACIÓN DE CONTENIDOS

Las herramientas de autoría son programas *software* los cuales permiten integrar una amplia gama de medios para facilitar la creación y desarrollo de aplicaciones en el entorno de la Televisión Digital Interactiva [52].

A continuación se mencionan las herramientas de autoría para el sistema de Televisión Digital exploradas [15]:

2.2.1.1 *Alticomposer*

Alticomposer básicamente es una herramienta de autoría visual destinada a la creación de aplicaciones interactivas y dinámicas para el entorno de la televisión basada en tecnología *Java* que soporta la especificación *DVB-MHP* y es altamente adaptable a otras especificaciones basadas en la misma especificación como *ATSC-DASE*. La interfaz de usuario muy intuitiva y fácil de usar, la hace simple para la creación de aplicaciones sin tener conocimiento en programación con el lenguaje *Java*, la organización de las etiquetas *XML* o cambios en la estructura de codificación, ya que esto se presenta y se manipula de manera transparente al usuario [53].

Como ventajas esta herramienta de autoría presenta una interfaz de usuario intuitiva y fácil de usar, permite la ampliación de funcionalidades mediante la inclusión de *plugins* e integra el componente para el establecimiento de la comunicación mediante el canal de retorno. Como principal desventaja que presenta esta herramienta de autoría es la de no permitir la edición del código *Java* generado y por otro lado, el alto costo para adquirir esta herramienta no se justifica para fines educativos y ambientes de investigación y pruebas a baja escala [15] [53] [54].

2.2.1.2 *Icareus ITV Suite Author*

Esta herramienta de autoría, anteriormente llamada *Cardinal Studio* cuenta con características similares a su antecesor. Es una herramienta intuitiva y muy versátil para la creación de contenidos de manera visual, que habilita el rápido desarrollo de contenidos de Televisión Digital Interactiva por cualquier tipo de persona sobre estándares basados en tecnología *Java* como *MHP*, *Tru2Way*, *Ginga* entre otros [15][55].

Como principales características es el fácil e intuitivo uso de la interfaz, además de la característica *WYSIWYG*³⁶ que esta ofrece, compatibilidad con las últimas versiones del *middleware MHP*, permite la modificación del código *Java*, rápido desarrollo de aplicaciones mediante el uso de plantillas, y la amplia variedad de componentes existentes para esta herramienta. Como principal desventaja se puede mencionar que esta herramienta de autoría sólo es un componente del sistema de Televisión Digital que la empresa *Icareus Technologies* ofrece. Entonces para probar las aplicaciones en un emulador y para hacer el

³⁶ *WYSIWYG: What You See Is What You Get. A modo simple, se trata de un editor en el cual se visualiza el resultado de acuerdo a lo que se está creando.*



despliegue de esta en un sistema real, se tiene que adquirir los módulos necesarios para este hecho, lo cual resulta en gran inversión económica [15] [55].

2.2.1.3 *Ortikon ACE*

Es una completa solución para servicios de Televisión Digital Interactiva orientada hacia los operadores de televisión, proveedores de servicio de Internet y *broadcasters* [15] [56].

Estas soluciones se caracterizan por ser modulares, escalables y altamente ajustables a las necesidades de los operadores de servicios. En cuanto a los diferentes productos que maneja la empresa Finlandesa *Ortikon*, se destaca *ORTIKON ACE Publisher* y *ORTIKON ACE Browser Emulator* para el desarrollo de aplicaciones en el ámbito de la Televisión Digital Interactiva [15][56].

- ***ORTIKON ACE Publisher***: Es una herramienta para la producción de contenidos mediante una interfaz *web*. El texto y las imágenes se insertan a la interfaz y los diseños son seleccionados a partir de los estilos predefinidos. También pueden usarse editores de texto *HTML* ordinarios en la creación del contenido para el *browser* de *ORTIKON ACE* [15].
- ***ORTIKON Browser Emulator***: Una aplicación la cual se usa para visualizar el contenido simultáneamente con el diseño que se realiza en el editor *WYSIWYG* [15].

Como características favorables, esta herramienta de autoría presenta una interfaz intuitiva y fácil de usar, soporta *DVB-HTML*, uso de plantillas, el código es modular y optimizado a diferencia del código generado por otras herramientas de autoría, hace uso de cualquier editor de texto *HTML* en la generación de contenido para el *browser Ortikon*, optimiza el uso valioso del ancho de banda del canal de *broadcast* gracias al *browser DVB-HTML*. Como desventajas, este sistema necesita hacer el despliegue de las aplicaciones generadas por *Ortikon ACE Publisher* sobre el *STB* mediante el *browser* que *Ortikon* ofrece para este proceso y las funcionalidades de las aplicaciones se limitan únicamente a las etiquetas soportadas por este navegador [15].

2.2.1.4 *MHPGen*

Es una herramienta de autoría para la creación de aplicaciones según el estándar europeo *DVB-MHP* compatibles con Televisión Digital, de tipo *entry-level* o de iniciación, que permite a cualquier persona sin conocimientos técnicos de programación ni del estándar *MHP* poder generar una aplicación lista para su posterior despliegue [57].

La herramienta de autoría es multiplataforma (*Windows, Linux y Mac OS*), el uso de plantillas para minimizar el tiempo de generación, facilita la reusabilidad y mantiene un mismo estilo, permite un fácil diseño y navegación de las aplicaciones, válido para generar cualquier aplicación sin tener en cuenta el canal de retorno, importa contenidos de forma dinámica desde bases de datos. Como principales desventajas de la herramienta se puede mencionar la no implementación del mecanismo *WYSIWYG* y orientada para el desarrollo de aplicaciones de pequeña complejidad [15] [57].



2.2.1.5 iTV Creator

Es una herramienta de autoría producto del trabajo de grado “Recomendaciones para la generación y distribución de contenidos educativos orientados a Televisión Digital Interactiva” de la FIET³⁷ en la Universidad del Cauca como parte del proyecto EDiTV, donde se presenta una aplicación de escritorio capaz de generar contenido DVB-HTML para propósitos de *t-learning*³⁸ [15].

Como características esta herramienta de autoría presenta [15]:

- Permite la construcción de aplicaciones DVB-HTML.
- Proporciona una interfaz gráfica para la creación de aplicaciones.
- Interfaz simple y fácil de usar.
- Creación libre de aplicaciones.
- Verificación de los cambios de manera inmediata.
- Soporta el uso de plantillas.
- Las aplicaciones generadas se ejecutan sobre STB reales.
- Soporte para el uso de hojas de estilo.

2.2.2 PLATAFORMAS MHP PARA LA DISTRIBUCIÓN DE CONTENIDOS

A continuación, se mencionan y se presentan las características de algunas plataformas para distribución de contenidos.

2.2.2.1 Opencaster

Opencaster básicamente es un generador de TS que se conecta a una red DVB a través de una interfaz de salida DVB-ASI. Presenta un conjunto de herramientas gratuitas y de código abierto para generar, procesar, difundir y multiplexar un flujo de transporte MPEG-2 [15] [58].

2.2.2.2 Fraunhofer IMK MHP Playout System

Es una solución de bajo costo que cumple con gran parte de las características que debe tener un sistemas de *Playout MHP*, como lo son el carrusel de objetos, multiplexación en tiempo real, múltiple protocolo, encapsulación, secciones privadas y soporte de salida de flujos ASI³⁹ [15].

³⁷ FIET: Facultad de Ingeniería Electrónica y Telecomunicaciones.

³⁸ *T-learning* es un campo en el cual el aprendizaje interactivo se obtiene a través de un televisor

³⁹ ASI (*Asynchronous serial interface*): Formato de flujo de datos el cual contiene un MPEG-TS (*MPEG Transport Stream*).



2.2.2.3 Comparación entre *Opencaster* y *Fraunhofer IMK MHP Payout System*

Las plataformas para distribución de contenidos mencionadas ofrecen los requisitos básicos que debe tener un sistema de *Payout*, como es un generador de carruseles de objetos, un generador de tablas *PSI/SI*, un multiplexor en tiempo real de los *TS* y una salida de flujos *ASI*, además de soportar información basada en archivos [15].

En cuanto a la plataforma *Fraunhofer IMK MHP Payout System* es estable, fiable y de bajo costo, ofrece más funcionalidades que *Opencaster*, pero tiene el inconveniente de no soportar ningún tipo de evento. Cabe destacar que maneja un mayor número de tipos de datos, tanto para la salida como la entrada comparado con *Opencaster* además, la plataforma dispone de una interfaz gráfica de usuario lo que facilita su uso [15].

Por el lado de *Opencaster*, esta plataforma es gratuita y de código abierto, se puede utilizar en la implementación de un laboratorio de pruebas para *iTV* de bajo costo además, es una buena herramienta para los desarrolladores de aplicaciones. Es ideal para implementar servicios de televisión en zonas residenciales y hoteles que no requieren grandes prestaciones como los servicios que ofrecen los organismos de radiodifusión [15].

La tabla 5 presenta una comparación más detallada relacionada con las dos plataformas para distribución de contenidos *MHP* exploradas.

	<i>Opencaster</i>	<i>Fraunhofer IMK MHP payout System</i>
<i>Interfaz gráfica de usuario</i>	No	Si
<i>Configuración de archivos</i>	Si	Si
<i>Java API</i>	No	No
<i>Multiplexación en tiempo real</i>	Si	Si
<i>Múltiples carruseles por servicio</i>	Si	Si
<i>Tipo de dispositivo</i>	SW	SW
<i>Sistema Operativo</i>		
<i>Windows</i>	No	No
<i>Linux</i>	Si	Si
<i>Eventos</i>		
<i>Eventos Hágalo ahora</i>	Si	No
<i>Eventos programados</i>	Si	No
<i>Gestión</i>		
<i>Ancho de banda</i>	Si	Si
<i>Flujo de transporte</i>	Si	Si
<i>Tablas AIT</i>	Si	Si
<i>Entrada</i>		
<i>Información basada en archivos</i>	Si	Si
<i>Archivos de flujo de transporte</i>	Si	Si
<i>ASI-Stream</i>	No	Si
<i>Salida</i>		
<i>ASI-Stream</i>	Si	Si

Tabla 5. Comparación de las plataformas para distribución de contenidos exploradas.
 Tomado de [15].



2.3 OBSERVACIONES Y CONCLUSIONES FINALES

Para finalizar este capítulo, se presentan las observaciones y conclusiones resultado de la exploración de las tecnologías las cuales servirán para definir los elementos que conformarán el diseño de la arquitectura que ofrecerá las características para buscar cubrir los requisitos funcionales como los no funcionales del proyecto.

2.3.1 CONCLUSIONES Y OBSERVACIONES RESULTADO DE LA EXPLORACIÓN A LAS ARQUITECTURAS DE RED DE TELECOMUNICACIONES Y A LAS PLATAFORMAS PARA LA CREACIÓN DE SERVICIOS DE TELECOMUNICACIONES

- La arquitectura de Red Inteligente fue concebida para ofrecer servicios sobre redes fijas. Cuando surgió el sistema *GSM*, se buscó la forma de adaptar la Red Inteligente a este entorno, pero no fue posible ya que inicialmente no había soporte para el ambiente de movilidad que presenta *GSM*.
- La arquitectura *CAMEL* tiene la capacidad de proveer servicios de voz en un ambiente móvil como lo es el sistema *GSM*.
- *CAMEL* fue desarrollado basado de la arquitectura de Red Inteligente como una extensión de estas funcionalidades para su entorno.
- *CAMEL* no es la arquitectura apropiada para aplicarse en un ambiente con servicios integrados y servicios de Internet.
- El *API* de *OSA* es una interfaz abierta y estandarizada, la cual permite a las aplicaciones usar las capacidades de la red sin ser el propietario o conocer su tecnología.
- *JAIN* contiene un reducido número de versiones finales entre todo el conjunto de *API*'s que la componen.
- La arquitectura de red de telecomunicaciones *IMS* permite mantener los modelos de negocio actuales o evolucionar hacia nuevos modelos de negocio.
- *IMS* ofrece servicios multimedia sobre redes fijas y móviles usando estándares abiertos.
- Se evidenció durante la exploración a las arquitecturas de red de telecomunicaciones, la existencia de ciertas similitudes como lo muestra la tabla 6.

Funciones.	Red Inteligente.	CAMEL.	IMS.
Repositorio de Información.	<i>Home Location Register (HLR).</i>	<i>Home Location Register (HLR).</i>	<i>Home Subscriber Server (HSS).</i>
Interacción entre el usuario y la red.	<i>Specialised Resource Function (SRF).</i>	<i>GSM Specialised Resource Function (gsmSRF).</i>	<i>Call Session Control Function (CSCF).</i>



Lógica de las aplicaciones y servicios.	<i>Service Control Function (SCF).</i>	<i>GSM Service Control Function (gsmSCF).</i>	<i>Application Server (AS).</i>
Invocación y control de servicios.	<i>Service Switching Function (SSF)</i>	<i>GSM Service Switching Function (gsmSSF)</i>	<i>Call Session Control Function (CSCF).</i>

Tabla 6. Similitudes entre las arquitecturas de red de telecomunicaciones exploradas.

- Con la tercera generación de redes móviles, la gran variedad de nuevos servicios multimedia que se pueden ofrecer, el aumento del ancho de banda para la transmisión de datos, permite que la arquitectura *IMS* sea la más apropiada y pueda soportar todo tipo de servicios multimedia, esto es posible gracias a las nuevas plataformas de servicios (*CAMEL*, *OSA*, *servicios SIP*) y a la simplicidad y extensibilidad del protocolo *SIP* que podrá adaptarse para interactuar con estas plataformas.
- La facilidad de los proveedores para el desarrollo y despliegue de nuevos servicios y aplicaciones en el entorno *IMS* debido a las *API* de *OSA/PARLAY*, *JAIN* y al uso de protocolos estandarizados como el protocolo *IP* y *SIP*.

2.3.2 CONCLUSIONES Y OBSERVACIONES RESULTADO DE LA EXPLORACIÓN DE LAS HERRAMIENTAS DE AUTORÍA Y PLATAFORMAS PARA LA DISTRIBUCIÓN DE CONTENIDOS EN *MHP*

- *Icareus ITV Suite Author* es una de las pocas herramientas que permite editar código *Java*, de tal manera que permite implementar aplicaciones de mayor complejidad [15].
- *Ortikon* ofrece una herramienta de autoría basada en *Web*, pero a diferencia de las demás genera aplicaciones *DVB-HTML*. Esta herramienta tiene el inconveniente que para hacer las pruebas del contenido *DVB-HTML* en un computador, se necesita de una aplicación que hace las veces de *browser*, igualmente sucede en el *STB*; si se desea ejecutar la aplicación se necesita el *browser* propietario suministrado por la empresa.[15].
- *MHPGen* tiene el inconveniente de tener disponible sólo los diseños suministrados por plantillas. Para generar nuevos diseños se tiene que crear nuevas plantillas para lo cual se requiere tener conocimiento en el lenguaje de programación *Java*. Además, esta herramienta ha sido concebida multi propósito, lo cual resulta en una herramienta muy pobre en cuanto a funcionalidades específicas [15].
- La plataforma *Fraunhofer IMK MHP* es estable, fiable y de bajo costo, ofrece más funcionalidades que *Opencaster*, pero tiene el inconveniente que no soporta ningún tipo de evento. Cabe destacar que maneja un mayor número de tipos de datos, tanto para la salida como la entrada, además la dispone de una interfaz gráfica de usuario lo que facilita su uso [15].
- *Opencaster* es un grupo de herramientas gratuitas, de código abierto y altamente configurables, de ahí que puede ser utilizado en la implementación de un laboratorio de pruebas para Televisión Interactiva de bajo costo. Es ideal para



implementar servicios de televisión en zonas residenciales y hoteles que no requieren grandes prestaciones como los servicios que ofrecen los organismos de radiodifusión [15].

2.3.3 COMPARACIÓN DE LAS TECNOLOGÍAS EXPLORADAS

La tabla 7 muestra la comparación realizada de las diferentes tecnologías presentadas, teniendo en cuenta una serie de indicadores producto de la exploración realizada a lo largo de este capítulo.

Tecnologías exploradas	1	2	3	4	5	6	7	8
Red Inteligente	X				X			
CAMEL	X	X			X			
OSA		X						
OSA / PARLAY		X						
JAIN		X						
IMS	X				X	X	X	X
Herramientas para generación de contenidos de TDi.			X					
Plataforma distribución de contenidos.				X				

Tabla 7. Cuadro comparativo de las tecnologías exploradas.

1. Arquitectura de red de telecomunicaciones.
2. Plataforma para la creación de servicios de telecomunicaciones.
3. Herramienta para la generación de contenidos en el entorno de la Televisión Digital Interactiva.
4. Plataforma para el despliegue y ejecución de aplicaciones en el sistema de Televisión Digital Interactiva.
5. Uso de protocolos estandarizados en la arquitectura de red de telecomunicaciones que permiten la adaptación con plataformas de creación de servicios.
6. Soporte para servicios multimedia y servicios de Internet.
7. La arquitectura de red de telecomunicaciones permite mediante la implementación de mecanismos, poder hacer uso de funciones como la autenticación, autorización y acceso que pueden ser implementados en aplicaciones que requieran tener este tipo de capacidades.
8. La arquitectura de red de telecomunicaciones ofrece el control sobre procesos que ejecuta tales como el establecimiento, mantenimiento y liberación de las sesiones, servicios ofrecidos, datos como los perfiles de usuario, perfiles del servicio entre otros aspectos.



CAPITULO 3. DISEÑO DE LA ARQUITECTURA PARA LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA *TDi*

Antes de definir la arquitectura, se debe conocer los requisitos tanto funcionales como no funcionales que el proyecto demanda, a partir de estos y con las conclusiones obtenidas en la exploración realizada en capítulos anteriores, se hace el respectivo análisis y se presenta la propuesta de la arquitectura.

3.1 PANORAMA GENERAL DEL PROBLEMA PROPUESTO

En términos generales, la solución que busca dar este proyecto de grado surge de la experiencia adquirida en proyectos de investigación previos como *EDiTV*, en donde se detectó inconvenientes en la implementación de las aplicaciones sobre el entorno de la *TDi*, ya que esta se realiza de manera vertical sin tener en cuenta algún tipo de referencia que le permita aprovechar de mecanismos de control y de optimización de recursos para lograr tener soluciones escalables, compatibles y flexibles.

Lo que se busca es entonces, poder contar con una plataforma que permita mejorar el control del sistema de *TDi* con el objetivo de facilitar y agilizar la construcción y despliegue de servicios y aplicaciones innovadores que complementen el contenido audiovisual emitido por el sistema de Televisión Digital.

3.1.2 REQUISITOS PARA EL DISEÑO DE LA ARQUITECTURA

3.1.2.1 Requisitos funcionales

- Permitir el despliegue y ejecución de las aplicaciones sobre la arquitectura diseñada.
- Cargar los elementos necesarios de la aplicación relacionados con el contenido a los cuales el usuario tiene privilegio.
- Ofrecer opciones de personalización de servicios al usuario.
- Gestionar el acceso a los servicios ofrecidos por la arquitectura, por parte del usuario.
- Controlar y gestionar aplicaciones, servicios y recursos desplegados sobre la arquitectura diseñada por parte del operador de la red.

3.1.2.2 Requisitos no funcionales

- El diseño de la arquitectura debe aportar los elementos necesarios para ser usados durante las etapas de diseño, construcción y despliegue de servicios y aplicaciones en el entorno de la Televisión Digital Interactiva.



- La arquitectura debe ofrecer mecanismos que permitan la comunicación entre diferentes tipos de terminales.
- Los servicios y aplicaciones desplegadas sobre esta arquitectura estarán disponibles a los terminales de usuario siempre y cuando cumplan con las capacidades y recursos mínimos que el servicio o la aplicación demanden.
- La arquitectura debe ejercer control sobre el funcionamiento de la aplicación y de los servicios desplegados.
- La arquitectura considera los mecanismos para interpretar y direccionar los mensajes entre los entornos y elementos que componen su entorno.
- La arquitectura diseñada debe soportar los servicios y aplicaciones construidos mediante el uso de plataformas para creación de servicios para redes de telecomunicaciones.
- El diseño de la arquitectura define para su estructura, el uso de protocolos estandarizados los cuales pueden adaptarse para interactuar con plataformas para la creación de servicios.

3.2 DESCRIPCIÓN DE LAS FUNCIONES QUE OFRECE LA ARQUITECTURA DE LA PLATAFORMA DE CONTROL PROPUESTA

Para describir las funciones que la arquitectura ofrece, se utiliza el diagrama de casos de uso con las respectivas descripciones de los escenarios como se muestra en la figura 18, donde se describe a través de casos de uso y se definen los actores, las funciones y las relaciones existentes dentro del sistema.

3.2.1 Diagrama de casos de uso para el diseño de la arquitectura de la plataforma de control

La figura 18 muestra el diagrama de casos de uso con las funciones que debe ofrecer la arquitectura a diseñar con los diversos actores que interactúan con el sistema.

Se observa que existen dos ambientes: el entorno de la TDi y la arquitectura de red, cada uno ofrece funciones, servicios y aplicaciones que se ejecutan en su propio entorno y están a disposición para ser usados en la aplicación que se despliega sobre la arquitectura diseñada.

En cuanto a los actores, existe el usuario que mediante la validación de sus datos puede convertirse en un usuario autenticado y acceder a la aplicación. El Operador del sistema de TDi y el Operador de la red de telecomunicaciones son los encargados de gestionar y controlar las aplicaciones y los recursos en cada uno de sus entornos.

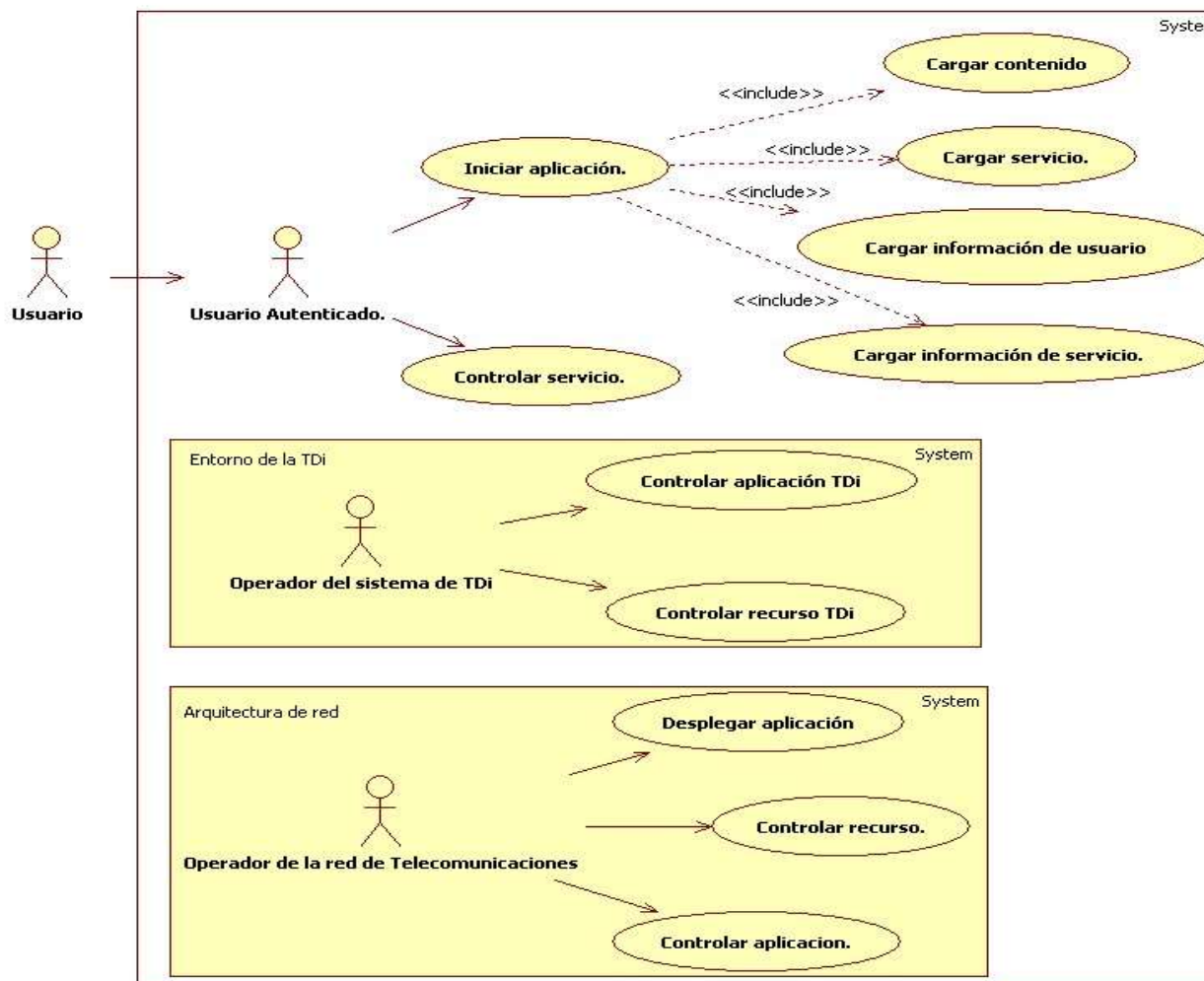


Fig. 18. Casos de uso para el diseño de la arquitectura de la plataforma de control.



3.2.2 Descripción de escenarios de los casos de uso

Nombre	Iniciar Aplicación.
Iniciador	Usuario autenticado quien inicia la aplicación desde un terminal con capacidades suficientes para su ejecución.
Prioridad	Alta
Propósito	Iniciar la aplicación usando funciones del entorno de la arquitectura de red para procesos de validación, autorización, control y asignación de recursos.
Descripción	Una vez iniciada la aplicación, esta espera el ingreso de un <i>login</i> y <i>password</i> , válida para el entorno de la arquitectura de red y que ha sido extendida y adaptada como funcionalidad usada por la aplicación en el entorno de la <i>TDi</i> .
Precondiciones	Tener la aplicación desplegada sobre la arquitectura base.
Flujo principal	Se hace la carga de la aplicación desde el sistema de <i>TDi</i> , ya sea por parte del usuario o por configuración automática de arranque sobre el dispositivo receptor. La aplicación espera por un usuario que ingrese su <i>login</i> y <i>password</i> y los envía a la arquitectura de red quien se encarga de hacer la validación. Si esta validación es exitosa, la arquitectura autoriza a la aplicación para invocar a las funciones encargadas de cargar los recursos para la presentación de la aplicación, cargar los datos de usuario y los datos del servicio, además de hacer el disparo de estos en el dominio de la arquitectura de red y se reflejen sobre el sistema de <i>TDi</i> .

Tabla 8. Descripción del escenario para el caso de uso: Iniciar aplicación.

Nombre	Cargar contenido.
Iniciador	Se inicia inmediatamente cuando el caso de uso <i>Iniciar aplicación</i> se ejecuta, esto debido a la relación <<include>> existente entre estos dos casos de uso.
Prioridad	Alta
Propósito	Recibir y mantener a disposición los elementos necesarios que hacen parte del contenido de la aplicación en el sistema de <i>TDi</i> .
Descripción	Una vez el usuario se encuentre autorizado para acceder a la aplicación, la arquitectura de red pone a disposición los elementos



	necesarios para que la aplicación obtenga estos recursos y construya todo el contenido visual para posteriormente ser presentado al usuario.
Precondiciones	El usuario debe estar autenticado y autorizado en el entorno de la aplicación.
Flujo principal	<p>Una vez ejecutada la aplicación en el entorno de la <i>TDi</i>, esta hace el llamado de ciertos recursos y datos provenientes de este entorno y de la arquitectura de red para elaborar la presentación para cada usuario.</p> <p>Para este caso de uso, los recursos, provenientes del canal de <i>broadcast</i>, han sido descargados y almacenados. Estos recursos son imágenes, videos, sonidos además de otros recursos destinados para la construcción del contenido visual de la presentación. Estos elementos se organizan formando una estructura definida por la codificación realizada sobre los <i>Xlets</i> y como producto final se muestra el contenido visual al usuario.</p>

Tabla 9. Descripción del escenario para el caso de uso: Cargar contenido.

Nombre	Cargar Servicio.
Iniciador	Se inicia inmediatamente cuando el caso de uso <i>Iniciar aplicación</i> se ejecuta, esto debido a la relación <<include>> existente entre estos dos casos de uso.
Prioridad	Alta
Propósito	Hacer el disparo de servicios disponibles en la arquitectura de red desde la aplicación desplegada en el sistema de <i>TDi</i>
Descripción	Una vez el usuario accede a la aplicación, esta le presenta los servicios provenientes de la arquitectura de red.
Precondiciones	El usuario debe estar autenticado y autorizado en del entorno de la aplicación.
Flujo principal	<p>Una vez ejecutada la aplicación en el entorno de la <i>TDi</i>, esta hace el llamado de ciertos recursos y datos provenientes de este entorno y de la arquitectura de red para elaborar la presentación para cada usuario.</p> <p>Para este caso de uso, la disponibilidad del servicio está</p>



	determinada si se encuentra desplegada sobre la arquitectura de red. Existiendo este servicio sobre la arquitectura de red, se hará la traslación del servicio hacia el sistema de TDi valiéndose de procesos como la traducción de protocolos de un entorno a otro mediante la implementación y uso de pasarelas software.
--	---

Tabla 10. Descripción del escenario para el caso de uso: Cargar Servicio.

Nombre	Cargar información de usuario.
Iniciador	Se inicia inmediatamente cuando el caso de uso <i>Iniciar aplicación</i> se ejecuta, esto debido a la relación <<include>> existente entre estos dos casos de uso.
Prioridad	Alta
Propósito	Consultar la información de usuario en la base de datos de la arquitectura de red.
Descripción	Se hace la consulta a la base de datos donde se abstraen los datos relacionados con cada usuario que están registrados dentro de la arquitectura de red.
Precondiciones	El usuario debe estar autenticado y autorizado en el entorno de la aplicación.
Flujo principal	Una vez ejecutada la aplicación, esta hace el llamado de ciertos recursos y datos provenientes del entorno de la TDi y de la arquitectura de red para elaborar la presentación para cada usuario. Para este caso de uso, algunos datos de usuario provistos por la arquitectura de red son consultados y usados por la aplicación para procesos de enrutamiento, localización, entre otros aspectos.

Tabla 11. Descripción del escenario para el caso de uso: Cargar información de usuario.

Nombre	Cargar información de servicio.
Iniciador	Se inicia inmediatamente cuando el caso de uso <i>Iniciar aplicación</i> se ejecuta, esto debido a la relación <<include>> existente entre estos dos casos de uso.
Prioridad	Alta
Propósito	Consultar la información del servicio en el repositorio de información de la arquitectura de red.



Descripción	Se hace la consulta a la base de datos y se abstrae los datos correspondientes al estado en que se encuentran los servicios definidos por cada usuario.
Precondiciones	El usuario debe estar autenticado y autorizado en el entorno de la aplicación.
Flujo principal	<p>Una vez ejecutada la aplicación en el entorno de la <i>TDi</i>, esta hace el llamado de ciertos recursos y datos provenientes de este entorno y de la arquitectura de red para elaborar la presentación para cada usuario.</p> <p>Para este caso de uso, los datos provistos por la arquitectura de red corresponden a los servicios. Estos son de vital importancia ya que se hace la consulta y de acuerdo a esta información, se procede a conceder el permiso para el acceso a los servicios disponibles en la arquitectura de red además, se aplican las características para cada servicio definidas según la personalización realizada por cada usuario.</p>

Tabla 12. Descripción del escenario para el caso de uso: Cargar información de servicio.

Nombre	Controlar servicio.
Iniciador	Usuario autenticado.
Prioridad	Media
Propósito	Habilitar el acceso y personalización del servicio.
Descripción	El usuario controla y define a que servicios puede tener acceso, quedando así almacenada esta elección en el perfil de cada usuario.
Precondiciones	El usuario debe estar autenticado y autorizado en el entorno de la aplicación.
Flujo principal	Una vez iniciada la aplicación con su respectivo contenido visual, los respectivos servicios y demás características, el usuario determina los servicios de los cuales quiere hacer uso. Esto se hace mediante la modificación de ciertos atributos en la base de datos de IMS que inmediatamente se reflejara en el perfil de servicio del usuario.

Tabla 13. Descripción del escenario para el caso de uso: Controlar servicio.



Nombre	Controlar recurso TDi.
Iniciador	Operador del sistema de <i>TDi</i> .
Prioridad	Alta
Propósito	Adicionar y quitar recursos necesarios para la aplicación de Televisión Digital Interactiva.
Descripción	El operador del sistema de TDi puede adicionar, modificar y eliminar recursos que hacen parte o sean invocados desde las aplicaciones desplegadas en el sistema de TDi.
Precondiciones	Operador autenticado y autorizado para realizar cambios sobre los recursos almacenados en el sistema de TDi.
Flujo principal	Una vez el operador del sistema de TDi haga el despliegue de la aplicación, este debe adicionar los recursos que necesita la aplicación para su ejecución. Si ya los recursos no prestan funcionalidad, el operador procede al retiro de estos sobre el sistema de TDi.

Tabla 14 Descripción del escenario para el caso de uso: Controlar recurso TDi.

Nombre	Controlar aplicación TDi.
Iniciador	Operador del sistema de <i>TDi</i>
Prioridad	Alta
Propósito	Controlar y gestionar la disponibilidad de las aplicaciones en el sistema de TDi.
Descripción	El operador del sistema de TDi se encarga de gestionar, adicionar o eliminar aplicaciones dentro del entorno del TDi.
Precondiciones	Operador autenticado y autorizado para realizar cambios a las aplicaciones sobre el entorno de la TDi.
Flujo principal	Sobre el entorno de la TDi se tienen desplegadas y ejecutándose diversas aplicaciones, el operador del sistema de TDi puede manipular estas y cambiar su estado: puede dar de baja una aplicación y esta quedara como no disponible para cualquier usuario de la arquitectura, hacer el despliegue de la aplicación para que inmediatamente quede a disposición a todos los usuarios o hacer un



	redespliegue de esta cuando se haga la carga de nuevas versiones de la aplicación.
--	--

Tabla 15. Descripción del escenario para el caso de uso: Controlar aplicación TDi.

Nombre	Desplegar aplicación.
Iniciador	Operador de la red de telecomunicaciones.
Prioridad	Alta
Propósito	Desplegar aplicaciones sobre la arquitectura de red de telecomunicaciones.
Descripción	El operador de la red de telecomunicaciones hace el despliegue de aplicaciones sobre la arquitectura de red las cuales estarán disponibles para los usuarios.
Precondiciones	Operador de la red de telecomunicaciones autenticado y autorizado para realizar cambios sobre la arquitectura de red.
Flujo principal	Teniendo la aplicación desarrollada y además funcionando eficiente y eficazmente, el paso a seguir es desplegarla en la arquitectura de red por medio del operador de la red de telecomunicaciones, quien sube esta aplicación al servidor de aplicaciones y hace el respectivo despliegue de esta quedando disponible para los usuarios.

Tabla 16. Descripción del escenario para el caso de uso: Desplegar aplicación.

Nombre	Controlar recurso.
Iniciador	Operador de la red de telecomunicaciones.
Prioridad	Alta
Propósito	Gestionar recursos necesarios para la aplicación desplegada en la arquitectura de red.
Descripción	El operador de la red de telecomunicaciones puede adicionar, modificar y eliminar recursos que hacen parte o que se invocan desde las aplicaciones desplegadas en la arquitectura de red.
Precondiciones	Operador de la red de telecomunicaciones autenticado y autorizado para realizar cambios sobre diversos recursos existentes en la arquitectura de red.



Flujo principal	Una vez el operador de la red de telecomunicaciones haga el despliegue de la aplicación, este debe adicionar los recursos que necesita la aplicación para su ejecución. Si los recursos existentes no prestan funcionalidad alguna, el operador de la red de telecomunicaciones procede al retiro de estos.
------------------------	---

Tabla 17. Descripción del escenario para el caso de uso: Controlar recurso.

Nombre	Controlar aplicación.
Iniciador	Operador de la red de telecomunicaciones.
Prioridad	Alta
Propósito	Controlar y gestionar la disponibilidad de las aplicaciones en la arquitectura de red.
Descripción	El operador de la red de telecomunicaciones es el encargado de gestionar, adicionar o eliminar aplicaciones dentro de este entorno.
Precondiciones	Operador de red autenticado y autorizado para realizar cambios sobre la arquitectura diseñada.
Flujo principal	Sobre la arquitectura se tienen desplegadas y ejecutándose diversas aplicaciones, el operador de red de telecomunicaciones puede manipular estas y cambiar su estado: puede dar de baja una aplicación y esta quedará como no disponible para cualquier usuario de la arquitectura de red, hacer el despliegue de la aplicación para que este disponibles por todos los usuarios registrados o hacer un redespiegue de esta cuando se haga la carga de nuevas versiones de la aplicación.

Tabla 18. Descripción del escenario para el caso de uso: Controlar aplicación.

3.3 PLANTEAMIENTO INICIAL DE LA SOLUCIÓN PARA EL DISEÑO DE LA ARQUITECTURA

Una vez conocidos el problema y los requisitos, el paso a seguir fue plantear la solución.

Teniendo en cuenta el éxito y la experiencia que ofrece una arquitectura de red de telecomunicaciones respecto al despliegue de servicios y aplicaciones sobre este ambiente, además de los procesos de control y de asignación de recursos que esta ejecuta, se piensa en ofrecer una solución basada en mayor medida en las características y funcionalidades que ofrecen las redes de telecomunicaciones con el fin de posibilitar el control unificado para el soporte de servicios y aplicaciones que se ofrecen en el entorno de la *TDi* además, a través de este tipo de arquitectura considerar también funciones como las de autenticación



y de autorización que se ejecutan en la arquitectura de red de telecomunicaciones y que pueden ser altamente adaptables a los servicios o aplicaciones en el sistema de *TDi*.

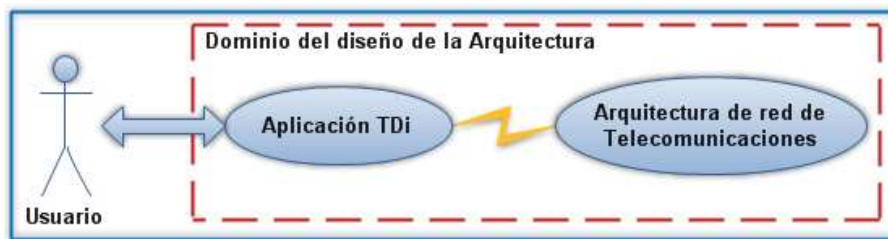


Fig. 19. Primera solución aproximada a nivel general.

La figura 19, muestra de manera muy general, la solución que se propone.

Básicamente la solución propuesta es el diseño de una arquitectura para *TDi* basada en propuestas provenientes de las arquitecturas de servicios de las redes de telecomunicaciones que permita habilitar la implementación de aplicaciones y servicios aprovechando las características que estas tienen de independizar los servicios que se pueden desplegar de la tecnología de la red de transporte sobre la cual se implementa y además, tener en cuenta que la arquitectura de red de telecomunicaciones también ofrece mecanismos y funcionalidades de control, establecimiento de sesiones y otras características que pueden ser aprovechadas por las aplicaciones del sistema *TDi*.

Como resultado se obtiene una plataforma horizontal, común para el sistema de *TDi* que ofrece un punto de referencia en cuanto a la implementación y control de los recursos de red que el servicio o aplicación necesita, partiendo de la concepción de que existen arquitecturas estandarizadas en el entorno de las redes de telecomunicaciones, las cuales presentan un número de funciones en común para ser reutilizados por todos los nuevos servicios o aplicaciones que están dentro o fuera de la red, sin importar el operador o la tecnología de acceso que el cliente usa.

3.4 ELECCIÓN DE LAS TECNOLOGÍAS A USAR EN EL DISEÑO DE LA PLATAFORMA DE CONTROL

Para elegir las tecnologías indicadas, básicamente se tuvieron en cuenta los dos entornos involucrados como se observa en la figura 19, la arquitectura de red de telecomunicaciones y el sistema de *TDi*.

Para la arquitectura de red de telecomunicaciones se debieron analizar los requisitos que el problema presenta y buscar que estos fueran cubiertos en mayor medida por las características que la arquitectura ofrece. Es por eso que se utiliza como punto de partida, la exploración realizada en el capítulo anterior y así se eligió a la arquitectura de red de telecomunicaciones que de mejor manera se adapta a las necesidades del sistema *TDi*.

En cuanto al entorno de la *TDi*, se parte de la existencia del proyecto *EDiTV*, el cual tiene definido su arquitectura y los estándares de Televisión Digital para la implementación y despliegue de aplicaciones sobre este ambiente. Se busca entonces, aprovechar los recursos y mecanismos que se ofrecen para así plantear una solución aceptable respecto a los requisitos que el problema presenta.



3.4.1 ELECCIÓN DE LA ARQUITECTURA DE RED DE TELECOMUNICACIONES

La elección de la arquitectura de red de telecomunicaciones es una de las partes más importantes para el diseño de la arquitectura ya que sobre esta se realizan diversas funciones que permitirán la ejecución tanto de las aplicaciones en el entorno de la TDi como de los servicios para los usuarios.

Al final del capítulo anterior, se observó de acuerdo a las conclusiones y valoraciones realizadas que la arquitectura de red *IMS* se mostraba como la mejor opción la cual podríamos adaptar en el diseño de la arquitectura, no obstante la Red Inteligente y *CAMEL* presentan características favorables respecto a los requisitos planteados por el problema y de alguna manera podrían convertirse en la base del diseño de la arquitectura.

Entonces para elegir a la arquitectura de red de telecomunicaciones indicada, se plantea a continuación una serie de aspectos técnicos los cuales necesita el diseño de la arquitectura presentar para ofrecer el soporte a las aplicaciones en el entorno de la *TDi*.

1. **Comunicación con el terminal de usuario:** La aplicación de TDi se ejecuta sobre su entorno y mediante la implementación de las pasarelas necesarias, se comunica con la arquitectura de red de telecomunicaciones para solicitar funciones, recursos y servicios para su ejecución.
2. **Ambiente para el despliegue de servicios y aplicaciones:** El despliegue de servicios y de aplicaciones se realiza sobre una arquitectura de red de telecomunicaciones independiente de los servicios a desplegar como del tipo de red de acceso implementado.
3. **Mecanismo para el control de sesiones, servicios y aplicaciones:** La arquitectura de red de telecomunicaciones debe ofrecer un mecanismo que permita el registro, establecimiento, modificación y finalización de las sesiones entre diversos dispositivos, además de tener a disposición los habilitadores y disparadores necesarios para los servicios implementados, como de las herramientas para la construcción de aplicaciones que se ejecuten en este entorno.
4. **Disponibilidad de aplicaciones y servicios:** Las aplicaciones y servicios desplegados sobre la arquitectura de red están disponibles para cualquier tipo de dispositivo.
5. **Uso de protocolos flexibles:** La arquitectura de red de telecomunicaciones debe incorporar protocolos simples y flexibles que permitan la creación y despliegue de todo tipo de servicios y aplicaciones con la ayuda de plataformas para la creación de servicios como *OSA/PARLAY*, *JAIN* y entre otras.
6. **Reutilización de elementos:** La arquitectura de red de telecomunicaciones debe poner a disposición funciones como la autenticación y autorización que pueden ser adaptadas en las aplicaciones o servicios que se implementan en el entorno de la Televisión Digital Interactiva.



A continuación se compara cada uno de estos aspectos técnicos con las características que presenta cada arquitectura de red de telecomunicaciones exploradas y se define la mejor opción para el diseño de la arquitectura.

Comunicación con el terminal de usuario.	
<i>Red Inteligente</i>	Esta arquitectura de red de telecomunicaciones se diseñó para ofrecer servicios de voz sobre redes fijas, en consecuencia, la comunicación del terminal de usuario con la arquitectura de Red Inteligente se establecería dependiendo de las interfaces y pasarelas hardware a construir tanto en la arquitectura como en el equipo terminal, necesarias para establecer un canal de comunicación entre estas entidades.
<i>CAMEL</i>	Esta especificación se diseñó para ofrecer servicios de voz sobre redes móviles como una extensión de la Red Inteligente a este entorno, al igual que la Red Inteligente, la comunicación con el terminal de usuario está limitada por las interfaces y pasarelas hardware a construirse tanto en la arquitectura como en el terminal de usuario necesarias para brindar un canal de comunicación.
<i>IMS</i>	Básicamente <i>IMS</i> es independiente de la red de acceso, entonces se puede llegar a pensar en hacer uso de las pasarelas definidas dentro de la arquitectura o simplemente implementar las pasarelas necesarias basadas en el protocolo <i>TCP/IP</i> para establecer un canal de comunicación entre el terminal de usuario e <i>IMS</i> .

Tabla 19. Comparación de las arquitecturas de red de telecomunicaciones frente a la comunicación con el terminal de usuario.

Ambiente para el despliegue de servicios y aplicaciones.	
<i>Red Inteligente</i>	Los servicios desplegados sobre esta arquitectura de red, exclusivamente están disponibles para redes fijas y ofrecen solo servicios de voz.
<i>CAMEL</i>	Los servicios y aplicaciones basados en <i>CAMEL</i> básicamente están disponibles para redes móviles y prestan servicios de voz.
<i>IMS</i>	<i>IMS</i> se caracteriza principalmente por ofrecer una arquitectura de red de telecomunicaciones independiente tanto de los servicios a implementar como del tipo de red de acceso donde se encuentre conectado el usuario.

Tabla 20. Comparación de las arquitecturas de red de telecomunicaciones frente al ambiente para el despliegue de servicios y aplicaciones.

Mecanismo para el control de sesiones, servicios y aplicaciones.	
<i>Red Inteligente</i>	En cuanto a la Red Inteligente y <i>CAMEL</i> , existen mecanismos los cuales controlan y ejecutan procesos de inicio, establecimiento y liberación de una llamada. Por otro lado el entorno de creación de servicios ofrece los disparadores necesarios para la ejecución de servicios relacionado con servicios de voz.
<i>CAMEL</i>	
<i>IMS</i>	<i>IMS</i> ofrece un control sobre las sesiones entre dispositivos como sesiones entre servicios, desde la negociación, el establecimiento, modificación y la finalización de esta. La existencia tanto de disparadores como habilitadores de servicios, hacen que la invocación de estos se haga de manera fácil y rápida. Además de estos aspectos, se



	complementan con mecanismos de seguridad que implementa IMS, el cual permite a un usuario autenticado y autorizado hacer peticiones para establecer sesiones además de disponer de los recursos y de los servicios que la arquitectura ofrece.
--	--

Tabla 21. Comparación de las arquitecturas de red de telecomunicaciones frente mecanismo para el control de sesiones, servicios y aplicaciones.

Disponibilidad de aplicaciones y servicios.	
Red Inteligente	Tanto la Red Inteligente como CAMEL, están limitados a ejercer su función en redes fijas y redes móviles respectivamente.
CAMEL	
IMS	
	IMS se caracteriza por ofrecer servicios independientemente de la red de acceso donde el terminal de usuario se encuentre conectado. En cuanto al terminal de usuario, el protocolo SDP se encarga de la negociación de la sesión según las capacidades que el dispositivo presente.

Tabla 22. Comparación de las arquitecturas de red de telecomunicaciones frente al disponibilidad de aplicaciones y servicios.

Uso de protocolos flexibles.	
Red Inteligente	La red Inteligente y CAMEL básicamente usan un conjunto de protocolos destinados solamente a la señalización para la telefonía, usados por las llamadas telefónicas sobre redes de telefonía pública conmutada.
CAMEL	
IMS	
	IMS define a un conjunto de protocolos estandarizados los cuales debido a su simplicidad y extensibilidad pueden adaptarse para interactuar con plataformas de servicios como CAMEL, OSA y Servicios SIP, lo cual resulta en el diseño e implementación de todo tipo de servicios multimedia como transmisión de audio, vídeo, cualquier tipo de información de usuario, servicios de presencia, mensajería, nuevos servicios de emergencia, multiconferencias y demás.

Tabla 23. Comparación de las arquitecturas de red de telecomunicaciones frente a uso de protocolos flexibles.

Reutilización de elementos.	
Red Inteligente	CAMEL y la Red Inteligente ofrecen funcionalidades que pueden ser adaptadas a aplicaciones que se ejecuten en sus respectivos entornos.
CAMEL	
IMS	
	IMS puede ofrecer variedad de procesos y funcionalidades que se implementan dentro de la arquitectura, como los de validación y de autorización que puede aprovechar la aplicación de TDi para adaptarlos a su entorno. Además pone a disposición la información relacionada con la personalización tanto de los usuarios como de los servicios para asignar los respectivos privilegios a cada uno de los usuarios que accedan a la aplicación.

Tabla 24. Comparación de las arquitecturas de red de telecomunicaciones frente a reutilización de elementos.

Al final, la tabla 25 muestra la valoración para cada arquitectura de red de telecomunicaciones respecto a los aspectos técnicos anteriormente mencionados.

Aspecto técnico	Red Inteligente	CAMEL	IMS
Comunicación con el terminal de usuario.	No recomendable	No recomendable	Recomendable
Ambiente para el	Redes Fijas	Redes Móviles	Independiente



despliegue de servicios y aplicaciones.			de la red de acceso
Mecanismo para el control de sesiones, servicios y aplicaciones.	No recomendable	No recomendable	Recomendable
Disponibilidad de aplicaciones y servicios.	Solo a redes fijas	Solo a Redes Móviles	Depende de capacidades del dispositivo
Uso de protocolos flexibles.	Protocolos no flexibles	Protocolos no flexibles	Protocolos flexibles y adaptables.
Reutilización de elementos.	No recomendable	No recomendable	Recomendable

Tabla 25. Valoración final de las arquitecturas frente a los aspectos técnicos.

Como conclusión, se define a *IMS* como la arquitectura de red de telecomunicaciones más indicada para ser implementada como la base del diseño de la arquitectura, ya que cumple en mayor medida con todos los aspectos planteados y con los requisitos que el problema plantea.

3.4.2 ELECCIÓN DE LAS TECNOLOGÍAS PARA EL ENTORNO DE LA *TDi*

3.4.2.1 Elección de la herramienta para la implementación de las aplicaciones en el entorno de la *TDi*

En el numeral 2.2.1, se mencionaron algunas herramientas de autoría *MHP* para la generación de contenidos las cuales permiten desarrollar aplicaciones en el entorno de la *TDi* de manera rápida y eficiente. El inconveniente de usar estas herramientas de autoría radica cuando los requisitos de las aplicaciones son complejos, ya que limitan el desarrollo de la aplicación en cuanto a diseño y funcionalidades se refiere.

En cuanto al diseño, estas herramientas presentan plantillas donde la variedad de los elementos a disposición que pueden ser incluidos en la presentación son muy escasos. En cuanto a las funcionalidades que la aplicación pueda ofrecer, la herramienta de autoría limita su implementación ya que existen solamente componentes definidos y soportados únicamente por cada una de estas herramientas y ejecutados sobre su respectiva plataforma [3].

Adicionalmente, algunas de estas herramientas de autoría generan su propia estructura de codificación y hacen en algunos casos, al código resultante inmodificable resultando con una solución que no satisface con los requisitos planteados.

Por estas razones y para la fase de desarrollo e implementación, no se usa ningún tipo de herramienta de autoría ya que limitaría muchas de las funciones que se quiere implementar en el prototipo. Para este fin, se decide hacer uso del modelo de aplicación *DVB-J* y construir la aplicación empleando los conocimientos en el lenguaje de programación *Java* para tener como resultado un producto que ofrezca las funcionalidades requeridas.



Como se mencionó en el primer capítulo, las aplicaciones elaboradas mediante el modelo de aplicación *DVB-J* necesitan de mayor tiempo para su construcción, pero se caracterizan por tomar funcionalidades de las librerías *Java* las cuales ofrece un gran número de funciones que pueden ser usadas por las aplicaciones. En cuanto al rendimiento de la aplicación (despliegue, carga y ejecución), este modelo es más eficiente que las aplicaciones implementadas con *DVB-HTML*, ya que *DVB-HTML* al momento de hacer la carga y la ejecución de la aplicación, toma todo el contenido asociado a esta (*XML, CSS*, imágenes, etc.) y construye una representación visual en una página resultando en una aplicación que demanda de mucho tiempo para realizar su ejecución [3][15].

En el anexo A se encuentra mayor información relacionada con el modelo de aplicación *DVB-J*.

3.4.2.2 Elección de la plataforma *MHP* para distribución de contenidos

En apartados anteriores se mencionó que este trabajo de grado tiene como referencia a la arquitectura definida por el laboratorio experimental de Televisión Digital Interactiva de la Universidad del Cauca, por esta razón se hace uso y se adapta a los elementos que componen a la infraestructura propuesta por el laboratorio y sobre todo a la plataforma *MHP* para la distribución de contenidos: *Opencaster*.

3.4.3 RESUMEN DE LAS TECNOLOGÍAS ELEGIDAS

Se presentan a continuación las tecnologías elegidas para el diseño de la arquitectura.

Implementación de la aplicación de <i>TDi</i>	Arquitectura de Red de Telecomunicaciones
<ul style="list-style-type: none">Modelo de aplicación <i>DVB-J</i>.	<ul style="list-style-type: none">Basada en <i>IP Multimedia Subsystem (IMS)</i>.

Tabla 26. Tecnologías elegidas para el diseño de la arquitectura.

3.5 ANÁLISIS DE LOS ENTORNOS Y ELEMENTOS QUE COMPONEN LA ARQUITECTURA.

Como se muestra en la fig 18, el diseño implementa en su primera aproximación, las funciones representadas por el diagrama de casos de uso para la plataforma. El diseño de la arquitectura cubre cada uno de estos aspectos mediante la abstracción de funcionalidades propias de los elementos que componen los dos entornos y los aplica para cubrir los requisitos funcionales impuestos por el problema planteado.

En cuanto a los requisitos no funcionales del problema, la solución que se ofrece es transparente tanto para los usuarios *STB* como para los usuarios registrados en el entorno de *IMS*, como los demás actores que intervienen con la arquitectura diseñada. Para cubrir estos requisitos no funcionales, es necesario tener en cuenta aspectos tales como el diseño de la aplicación, el mecanismo para envío y recepción de los datos entre los distintos componentes de la arquitectura y además, la implementación de las pasarelas necesarias para la comunicación con todas las entidades que intervienen en todo el diseño de la arquitectura.



El resultado final, muestra el nivel de adaptación que puede llegar a tener dos entornos de diferente naturaleza para realizar el control unificado de una aplicación y servicios ofrecidos al sistema de *TDi*, valiéndose de la implementación de diferentes interfaces y pasarelas que permiten el paso de información entre estos entornos y la aplicación. La aplicación se despliega sobre la arquitectura diseñada y puede accederse mediante un televisor conectado a un *STB*, un computador con una tarjeta para recepción de televisión digital que lo convierte en el dispositivo receptor o simplemente la mejor opción en cuanto a nivel de diseño, implementación y escenario de pruebas sobre un emulador de *TDi* el cual ofrece las características básicas y necesarias para hacer la ejecución y posterior visualización de la aplicación de Televisión Digital Interactiva.

3.6 ELECCIÓN DE LAS HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE LA ARQUITECTURA DISEÑADA

Esta sección define las herramientas a utilizar para la implementación de la arquitectura.

3.6.1 Herramientas para la construcción de la aplicación en el sistema de *TDi*.

Para el desarrollo de las aplicaciones *MHP* se usó el modelo de aplicación *DVB-J* como se había mencionado anteriormente.

Se inclinó directamente por el modelo de aplicación basado en *Java* debido a que este presenta prácticamente características ilimitadas respecto al tipo de interactividad y funcionalidades a ofrecer, debido a las diversas *API's Java* disponibles y definidas dentro del estándar *MHP*, concepto análogo al modelo de aplicación *DVB-HTML* que presenta soporte solo con las etiquetas definidas dentro del modelo y de las secuencias de comandos (*Scripting*) [15]

Para este proyecto, en cuanto al desarrollo e implantación del prototipo a desplegarse sobre la arquitectura propuesta, se necesita del modelo de aplicación *DVB-J* donde se hace la invocación de las clases y métodos definidos dentro del lenguaje de programación *Java* que ofrecen las herramientas y medios necesarios para implementar las funcionalidades que el prototipo requiera además de la implementación de las pasarelas necesarias para el intercambio de información entre los entornos que componen la arquitectura, conceptos que no podrían ser desarrollados e implementados haciendo uso del modelo de aplicación *DVB-HTML*.

En la tabla 27 se presenta la comparación entre los modelos de aplicación definidos por el estándar *MHP*.

<i>DVB-J</i>	<i>DVB-HTML</i>
Usado para realizar cálculos complejos.	Rápida actualización de las aplicaciones.
Usado para procesamiento pesado del lado del cliente.	Fácil desarrollo de contenido.
	Presentaciones estáticas sobre la pantalla.
Soporta Animaciones.	Funcionamiento es análogo a un sitio web.
Generalmente es distribuido a través de	Pre-procesamiento, procesamiento del



uno o varios carruseles el cual tiene un ancho de banda limitado.	lado del servidor.
La distribución de aplicaciones <i>DVB-J</i> está más sujeta a restricciones.	Fácil de distribuir, por canal de retorno o mediante el carrusel.

Tabla 27. Comparación entre *DVB-J* y *DVB-HTML*.
Tomado de [15].

Las aplicaciones desarrolladas con el modelo de aplicación *DVB-J* reciben el nombre de *Xlet*. A pesar de que un *Xlet* se programa con el lenguaje *Java*, este no permite contar con todos los métodos y clases básicas propias de este lenguaje [60].

En las primeras implementaciones de la norma *MHP* (*MHP 1.0.x*), se apoyan en la versión 1.1 de *Personal Java*⁴⁰, para la implementación 1.1.2 se basa en *J2ME*⁴¹ permitiendo mayor número de funcionalidades que las primeras versiones, pero con la necesidad de tener sistemas decodificadores con mayores recursos.

Teniendo en cuenta que la versión de la maquina virtual que emplea el equipo decodificador para ejecutar las aplicaciones es aproximadamente la versión 1.1.8 de *Java*, hay que garantizar que los métodos y clases a invocar en la aplicación puedan ser interpretadas de manera correcta en el equipo receptor para evitar algún tipo de inconveniente durante la ejecución de la aplicación, es por eso que fundamentalmente nos basaremos en el Kit de Herramientas de Ventana Abstracta *AWT* el cual está definido en la versión 1.1.8 de *Java* que contiene los elementos necesarios para el manejo de gráficos, interfaz de usuario, y sistema de ventanas.

Adicionalmente se tiene a disposición parte de las especificaciones *JavaTV*, *HAVi*, *DAVIC*, *JMF*, *DVB* las cuales nos aportan funcionalidades para el manejo y control multimedia, control de la tarjeta de red, descarga de las aplicaciones, control del canal de retorno y entre otras opciones.

Ademas se hace uso de *Eclipse IDE*, un entorno integrado de desarrollo de código abierto y multiplataforma el cual permitirá mediante la inclusión de las *API's* para televisión digital, la construcción y ejecución de la aplicación *DVB-J* de manera fácil, rápida y sencilla desde la misma plataforma.

3.6.2 Herramientas para el despliegue de la aplicación *DVB-J*

Como se mencionó en pasajes anteriores, la definición de *Opencaster*, el sistema de modulación, el estándar para la trasmisión de datos de Televisión digital, el *middleware*, el *Set Top Box* entre otros elementos, ya estaban definidos, elegidos e implementados para el laboratorio experimental de Televisión Digital Interactiva. Por esta razón se buscó las maneras para adaptarse al entorno presentado para el desarrollo e implementación del prototipo.

El inconveniente surge cuando las funcionalidades a incorporarse en el prototipo necesitan de recursos que están disponibles en Internet, entonces examinando las capacidades de los *STB* existentes en el laboratorio se encontró que estos implementan la versión de *MHP*

⁴⁰ *Personal Java*: Es una edición gratis de *Java* para sistemas móviles y embebidos basados en *Java 1.1.8*.

⁴¹ *J2ME* es una plataforma java usado en dispositivos móviles y sistemas embebidos.



Es una implementación de las funciones de control de llamada/sesión (*CSCF, Call Session Control Functions*) y del servidor local del suscriptor (*HSS, Home Subscriber Server*), que juntos forman el núcleo de la arquitectura *IMS*.

En la fig 20 se muestra la arquitectura definida por *OpenIMSCore*, su ubicación dentro de la arquitectura *IMS* y los protocolos por medio de los que se comunican. Entre los componentes bases de este software de código abierto están *SIP Express Router (SER)* y *MySQL* [60].

Es importante resaltar que *OpenIMSCore* no tiene fines comerciales, su objetivo es proveer una implementación de referencia para probar las tecnologías de *IMS* y facilitar la construcción de prototipos.

3.6.3.2 Ericsson's Service Development Studio (SDS)

Básicamente el *SDS* de *Ericsson* es una herramienta para el desarrollo y pruebas extremo a extremo tanto del lado del servidor como del lado del cliente de nuevas aplicaciones convergentes todo *IP-SIP*. Incluyen el núcleo de *IMS* y un gran rango de habilitadores (*enablers*) para servicios de comunicación como telefonía multimedia *IP, IPTV, Push to talk*, presencia y gestión de grupos, mensajería *IMS*, entre otros [61].

Ofrece soluciones globales y estandarizadas para el desarrollo de nuevas aplicaciones *IMS* y servicios. *SDS* provee variedad de plantillas y asistentes que proporciona facilidad en la elaboración de proyectos en el menor tiempo posible, el uso de *APIs* de alto nivel para tener a disposición y controlar capacidades avanzadas como la presencia y gestión de grupos (*PGM*), voz sobre *IP (VoIP)*, *Push to talk (PTT)*, mensajería *IMS (IMS-M)*, protocolo de internet para televisión (*IPTV*) y los próximos desarrollos añadirán capacidades como la telefonía multimedia *IP* [61].

SDS ofrece las siguientes características:

En cuanto a desarrollo y depuración de aplicaciones *IMS*:

- Basado en *Eclipse IDE* para desarrollos en *Java EE (Eclipse IDE 3.4 y WTP 3.0)*.
- *EclipseME 1.7.9* integrado para el desarrollo de *Midlet*⁴² en *JavaME* con kits de desarrollo inalámbrico (*Wireless Toolkit - WTK*) proporcionados por *SUN, SEMC* y *NOKIA*.
- Desarrollo de aplicaciones en cuanto al lado del servidor como del cliente con ayuda de asistentes y de plantillas.

⁴² *Midlet* es un programa en lenguaje de programación *Java* para dispositivos embebidos



Visión General de SDS

Construir, Probar, Entorno extremo a extremo. Sobre un PC con todos los emuladores.

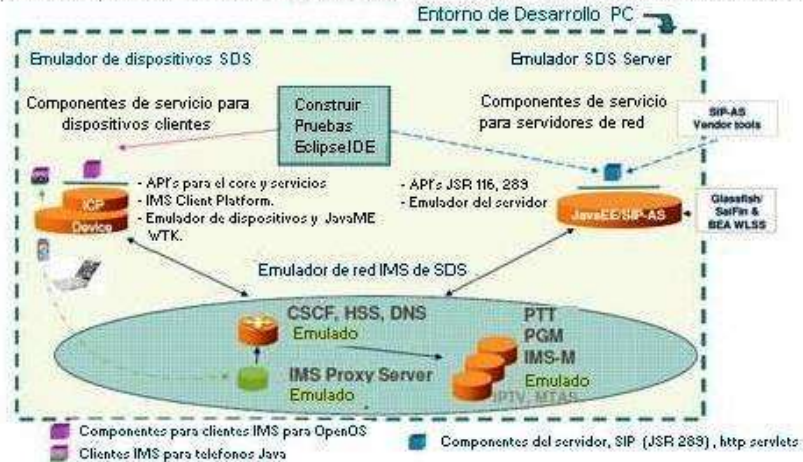


Fig. 21. Arquitectura definida por SDS de Ericsson.
 Tomado de [61].

En cuanto al emulador del núcleo IMS, SDS provee un entorno donde los nodos dentro del núcleo son simulados, donde se tiene:

- **S-CSCF:** Realiza el control de las sesiones para los terminales registrados.
- **I-CSCF:** El punto de contacto dentro de la red del operador para todas las conexiones destinadas a un usuarios de esa red del operador.
- **P-CSCF:** El primer punto de contacto dentro del núcleo de IMS.
- **BGCF:** Servidor SIP que enruta peticiones SIP URI (SIP Uniform Resource Identifier⁴³) y TEL URI hacia redes externas.⁴⁴
- **HSS:** El principal almacenamiento de datos relacionado con toda la información de los usuarios y de los servicios.
- **DNS:** Servidor de nombres de dominio que permite traducir de nombre de dominio a dirección IP y vice-versa
- **ENUM:** es un protocolo que utiliza el servidor DNS de Internet para traducir números de teléfono E.164⁴⁵ a esquemas de dirección de IP.

En cuanto a tecnologías de acceso soporta:

- Acceso móvil.
- Acceso fijo de Banda ancha

⁴³ URI es una cadena corta que identifica un recurso.

⁴⁴ SIP URI es un esquema usado para direccionar llamadas con SIP y TEL URI usado para direccionar llamadas SIP a la red telefónica pública conmutada.

⁴⁵ E.164 es el plan de numeración y describe cómo y por quien son asignados los números telefónicos.



- Acceso inalámbrico.

3.6.3.3 Elección de la herramienta que implementa la arquitectura de red

Hay un aspecto en especial necesario para comparar, y muy importante para el proyecto propuesto el que permitió decidir la herramienta a usar.

Lo que se buscó fue construir tanto los servicios y las aplicaciones en el lado del servidor como en el lado del cliente de la manera más sencilla y rápida. Por un lado *OpenIMSCore* ofrece sólo el núcleo de *IMS* y a partir de esto se comienza con la implementación de los servicios y aplicaciones sin tener referencia alguna, por otro lado *SDS* define en el lado del cliente una arquitectura y diversas *API's* para la implementación de la lógica sobre el cliente. En el lado del servidor se implementan servicios y aplicaciones a partir de plantillas, ayudantes y además de habilitadores que permiten una implementación eficiente y rápida de estas.

Por lo anterior, el uso de *SDS* de *Ericsson* es el más indicado para usarse en el diseño de la arquitectura.

3.7. RESUMEN DE LAS HERRAMIENTAS A UTILIZAR EN LA IMPLEMENTACIÓN DE LA ARQUITECTURA

A continuación se hace una síntesis de las herramientas que se usaran en la implantación arquitectura de la plataforma de control propuesta.

- La herramienta *SDS* de *Ericsson*⁴⁶, en su versión demo destinada a desarrollos a baja escala y para propósitos educativos. Brinda los elementos básicos y necesarios a nivel software que componen el núcleo de la arquitectura *IMS*, además de un entorno de creación de servicios y un ambiente de pruebas tanto del lado del cliente como del lado del servidor.
- Para el despliegue del prototipo en este proyecto se hace mediante el emulador de Televisión Digital Interactiva *XleTView* que soporta estándar *DVB* con el *middleware MHP*.

Este emulador implementa funcionalidades y características similares a las que un dispositivo receptor presenta además, hace la ejecución de la aplicación de manera aproximada a un *STB*.

A nivel de diseño de la arquitectura, adicional al uso de emuladores se puede desplegar la aplicación a través de:

- Hacer la descarga de la aplicación y los recursos necesarios directamente desde el servidor de contenidos al *STB* mediante el canal de *broadcast*, además de esto, se dispone del canal de retorno como medio de comunicación entre el

⁴⁶ *SDS* de *Ericsson* disponible en: http://www.ericsson.com/developer/sub/open/technologies/ims_poc/tools/sds_40



usuario y la fuente de información o con las demás entidades que componen la arquitectura diseñada.

- Hacer la descarga de la aplicación y los recursos a usar directamente del computador hacia un *STB* de desarrollo sin necesidad de tener un canal de *broadcast*.
- En cuanto a la implementación de los mediadores y artefactos software que ofrecen los canales de comunicación necesarios para el flujo de mensajes y de datos entre las entidades de los entornos mencionados y la aplicación además, de la implementación de las aplicaciones, se hace uso del lenguaje de programación *Java* con la ayuda del entorno integrado de desarrollo *Eclipse IDE*, con la adición de las funcionalidades para trabajar en el entorno de la Televisión Digital mediante la inclusión de las respectivas *API's como lo es HAVI* entre otras.

Implementación aplicación DVB-J	Ejecución de la aplicación DVB-J	Arquitectura de Red de Telecomunicaciones.
<ul style="list-style-type: none">• <i>Eclipse IDE.</i>• <i>JDK 1.1.8.</i>• <i>JDK 1.6</i>• <i>MHP STUBS 1.1.1.</i>	<ul style="list-style-type: none">• <i>Opencaster.</i>• <i>XleTXiew</i>• <i>MimundoTV</i>	<ul style="list-style-type: none">• <i>SDS de Ericsson</i>

Tabla 28. Herramientas a usar en la implementación de la arquitectura diseñada.

3.8 ARQUITECTURA PROPUESTA PARA LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA *TDi*

En las figuras 22, 23, 24 se presenta la arquitectura para la Plataforma de Control en el Ámbito de la Televisión Digital Interactiva, donde se observa a los dos entornos involucrados que interactúan mediante un elemento en común: el *servidor placa*.

3.8.1 PARA EL ENTORNO DE LA TELEVISIÓN DIGITAL INTERACTIVA

La figura 22 muestra la arquitectura propuesta en el laboratorio experimental de *TDi* del proyecto *EDiTV*, donde se hace la emisión de los contenidos de televisión junto a los datos y aplicaciones y posteriormente, la carga y ejecución sobre el dispositivo receptor.

La arquitectura del laboratorio experimental de *TDi* está compuesto por:

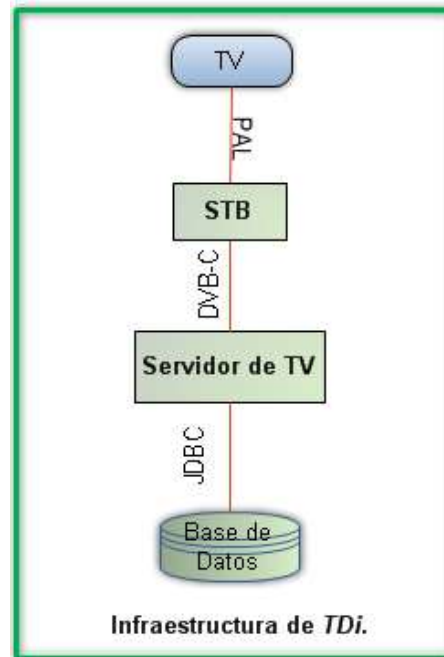


Fig. 22. Arquitectura propuesta por el laboratorio experimental de TDi del proyecto EDiTV

- **Televisor:** Sobre este elemento se visualizan los contenidos de televisión emitidos por las estaciones de *broadcasting*. Desde el punto de vista de la recepción de la señal de televisión, el televisor puede ser analógico o digital a los cuales se les conecta el dispositivo receptor *STB*. Cabe mencionar que algunos televisores ya tienen incorporados el dispositivo receptor *STB* que permiten la recepción de señales tanto analógicas como digitales [15].
- **Set Top Box:** El *STB* es el elemento encargado de recibir señales de televisión digital, hacer la descodificación, comprobación de los derechos de usuario, decodificación y posterior visualización de los contenidos sobre el televisor [15].

Básicamente este elemento ejecuta dos funciones principales:

- Recibir tanto las señales de video digital, como las aplicaciones emitidas con sus datos correspondientes, decodificando el video, audio, datos y de acuerdo a los comandos cargados ejecutar las aplicaciones [2].
- Enviar y recibir datos a través del canal de retorno, el cual facilita al usuario la interacción con las aplicaciones emitidas. El usuario también puede pedir información adicional a través del canal de retorno sobre un servidor del proveedor [2].
- **Servidor de Televisión:** Dentro de este servidor se ejecutan tareas como la generación del flujo de transporte *MPEG-2* a través de la multiplexación de datos como los flujos de audio/video, la información de programa/servicio y el sistema de archivos del carrusel de objetos [3] [14].

El servidor de televisión como se observa en la figura 22, está compuesto por dos elementos: el servidor de *playout* y la tarjeta moduladora.



El *servidor de playout* es el encargado de implementar las funciones en cuanto a la codificación del audio y video, generación del carrusel de objetos, datos y eventos; la generación de la información de programas, servicios y aplicaciones; la multiplexación de todos estos contribuyentes para la generación de un único flujo de transporte *MPEG-2* que será modulado posteriormente para su transmisión [15].

La *tarjeta* moduladora es el elemento el cual realiza la adaptación de los flujos de televisión a una señal adecuada para la trasmisión [15].

3.8.2 PARA EL ENTORNO DE LA ARQUITECTURA DE RED DE TELECOMUNICACIONES IMS

La figura 23 muestra el *núcleo* de la arquitectura de red de telecomunicaciones *IMS*. Estos elementos son los básicos y necesarios para su operación y ofrecen las características necesarias para este proyecto. Consta de los siguientes elementos:

- **Call Session Control Function (CSCF):** La función de control de llamada es la entidad más importante sobre el plano de control de la arquitectura de telecomunicaciones *IMS* donde se procesan los paquetes de señalización *SIP* mediante *servidores SIP* o *proxies*.

El *CSCF* está compuesto por el *P-CSCF*, *I-CSCF*- *S-CSCF*. El *P-CSCF* es el primer punto de contacto para un terminal con respecto a la arquitectura de red *IMS*, el *S-CSCF* es el nodo central para la trasmisión de señales y finalmente el *I-CSCF* localizado en el extremo de un dominio administrativo, donde su dirección *IP* se en el *DNS* del dominio para que servidores remotos tengan acceso a él y lo puedan usar como salto intermedio en la comunicación.

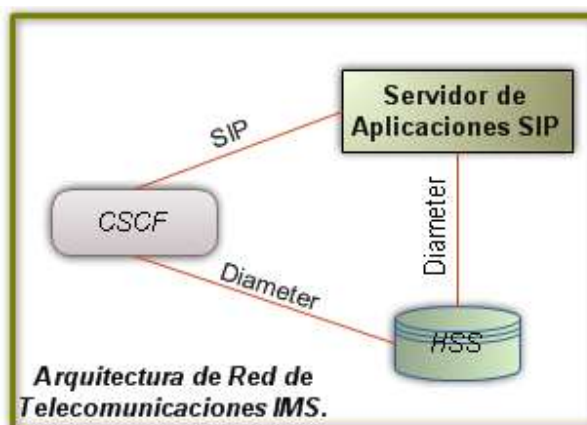


Fig. 23. Arquitectura propuesta para la arquitectura de red de telecomunicaciones *IMS*.

- **Servidor de Aplicaciones SIP (AS):** El servidor de aplicaciones es una entidad donde se alojan y se ejecutan los servicios ofrecidos a los usuarios finales. Existen tres tipos de servidor de aplicaciones: *SIP AS (SIP Application Server)*, *OSA-SCS (Open Service Access - Service Capability Server)* e *IM-SSF (IP Multimedia Service Switching Function)*.



El servidor *SIP AS* almacena y ejecuta servicios multimedia *IP* basados en *SIP*, el servidor *OSA-SCS* proporciona una interfaz al servidor de aplicaciones del *framework* de *OSA* y finalmente, el servidor *IM-SSF* permite reutilizar servicios que fueron desarrollados para *GSM* usando *CAMEL* en *IMS*.

Para la arquitectura presentada, solo se tiene en cuenta el servidor de aplicaciones *SIP*, ya que solo se implementaran servicios multimedia basados en el protocolo *SIP*.

- **Home Subscriber Server (HSS):** El *HSS* es el repositorio de información principal de almacenamiento de los datos de usuarios y de servicios existentes en la arquitectura de red *IMS*. Contiene toda la información de suscripción del usuario necesaria para gestionar sesiones multimedia. Esta información incluye, entre otras, información de localización, de seguridad (autenticación y autorización), perfil de usuario (indica los servicios a los que el usuario tiene acceso) y el *S-CSCF* donde se encuentra el usuario.

3.8.3 ARQUITECTURA DE LA PLATAFORMA DE CONTROL EN EL ÁMBITO DE LA TELEVISIÓN DIGITAL INTERACTIVA

Finalmente, la figura 24 muestra la arquitectura propuesta para la Plataforma de Control en el Ámbito de la Televisión Digital Interactiva, donde se puede observar los dos entornos anteriormente mencionados los cuales se comunican con un elemento en común: el servidor placa.

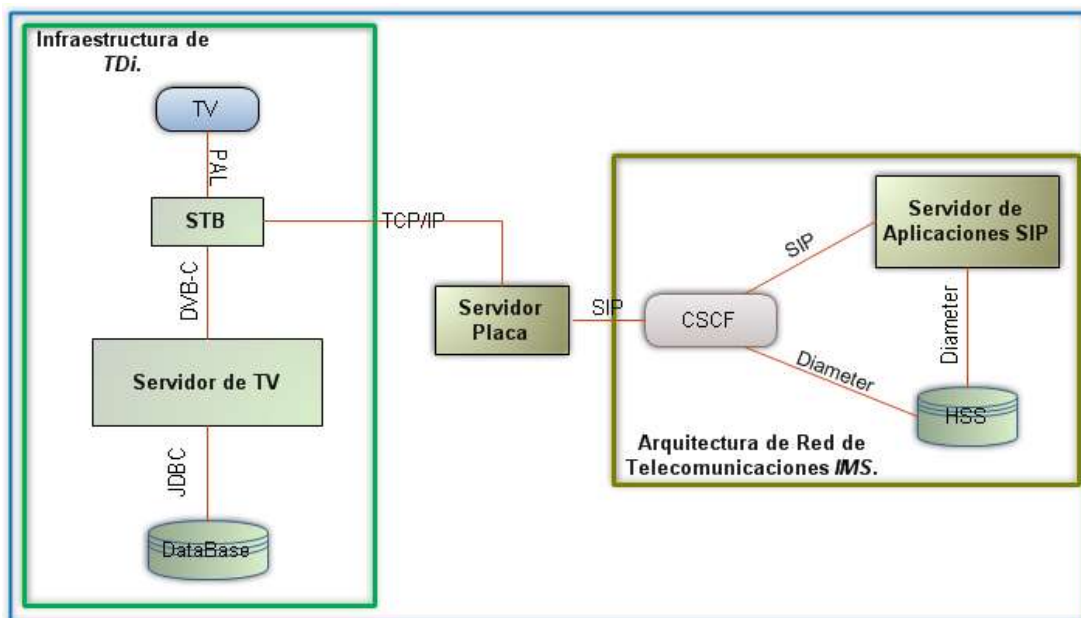


Fig. 24. Arquitectura propuesta para la plataforma de control en el ámbito de la TDi

Por parte de *IMS* se tienen los diversos servicios desplegados en la arquitectura, el repositorio de información relacionado con los datos de usuario y los datos de los servicios, los disparadores de los servicios y entre otros elementos.



Sobre la arquitectura propuesta por el laboratorio experimental de *TDi* se hace el despliegue y ejecución de las aplicaciones de televisión. Esto comprende la emisión de los contenidos de audio y de video, aplicaciones y datos que se presentan a los usuarios con los privilegios necesarios para recibir este tipo de contenidos.

Es entonces, que el *servidor placa* está concebido como el mediador entre estos dos entornos donde las aplicaciones ejecutadas en el entorno de la *TDi* puedan tener acceso a las funcionalidades, recursos, información de usuario y de servicios y hacer uso de los servicios que están en la arquitectura de red *IMS* dando como resultado, una experiencia televisiva con mayores características y con servicios propios del mundo de las telecomunicaciones adaptados a la *TDi*.



CAPITULO 4. IMPLEMENTACIÓN DE LA ARQUITECTURA PARA EL SOPORTE DE SERVICIOS EN EL SISTEMA DE *TDi*.

Este capítulo presenta el diseño detallado de la arquitectura para la Plataforma de Control en el Ámbito de la Televisión Digital Interactiva de acuerdo con las tecnologías y elementos anteriormente analizados las cuales de acuerdo a sus características y ventajas, ofrecen soluciones a los requisitos que el problema impone.

Al final se mostrara la adaptación y la interacción existente entre los elementos elegidos, dar a conocer la definición de la arquitectura, las funciones a cargo de cada entorno, la conversión de los mensajes y demás aspectos transparentes desde el punto de vista del usuario, pero de gran importancia para la ejecución de la aplicación y de los servicios.

4.1 DISEÑO E IMPLEMENTACIÓN DE LA PLATAFORMA DE CONTROL EN EL ENTORNO DE LA *TDi*.

La figura 25 muestra el diseño de la arquitectura compuesta por el sistema de *TDi* y la arquitectura de red de telecomunicaciones *IMS*, comunicadas mediante la implementación de pasarelas y artefactos software necesarios que permiten el paso de información entre los componentes involucrados.

Por una parte el sistema de Televisión Digital Interactiva alberga a la aplicación implementada mediante el modelo de aplicación *DVB-J*. La característica más importante es el uso que hace la aplicación del canal de retorno como medio de transporte y de comunicación con los demás elementos que componen el diseño de la arquitectura, representada en la figura con los dos óvalos en la nube *TDi*, donde el ovalo en la *Aplicación DVB-J* es la parte *software* encargada de la implementación de los métodos necesarios para el manejo del canal de retorno y el ovalo ubicado en el límite de la nube *TDi* es el canal de retorno físico que comunica este entorno con *IMS*.

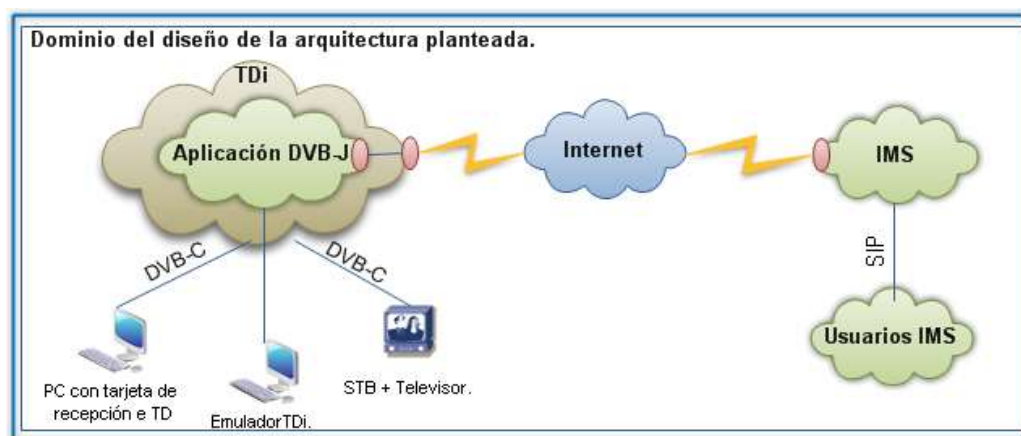


Fig. 25. Arquitectura general para el soporte de servicios en el entorno de la *TDi*.

La aplicación envía y recibe datos de *IMS* usando el canal de retorno, pero estos no son comprensibles para este entorno, es así que se hace necesaria la implementación y uso de



mediadores además de otros artefactos *software* los cuales buscan hacer la traducción de protocolos para interpretar solicitudes y la posterior ejecución de estas.

La implementación de estos mediadores y artefactos están representados sobre el ovalo en el límite de la nube *IMS*. Este ovalo es la implementación a nivel *software* de las clases y métodos necesarios encargados de recibir las peticiones de la aplicación *DVB-J*, interpretar y traducir estas para que sean comprensibles en el entorno *IMS*, luego direccionarlas a las diferentes entidades encargadas de la ejecución de la solicitud y, cuando se genera una respuesta por parte de la arquitectura de red *IMS*, los mediadores y artefactos tienen la función de llevar a cabo el proceso contrario al de la solicitud, convertir la respuesta a un mensaje comprensible por la aplicación *DVB-J* para su procesamiento y ejecución.

Como se aprecia en la figura 25, el acceso a la aplicación se hace a través de cualquier terminal con los recursos necesarios para ejecutar las instrucciones que la aplicación solicite, un televisor y un equipo decodificador son los más indicados para este hecho.

Para efectos de ejecución del prototipo sobre la arquitectura, la solución más indicada es el uso de emuladores de Televisión Digital Interactiva, ya que estas implementan en gran parte, las características más sobresalientes de un sistema de Televisión Digital y garantizan que el funcionamiento en este entorno sea muy similar al funcionamiento sobre los sistemas de televisión digital.

En cuanto a *los usuarios IMS*, se refiere a los usuarios que tienen acceso a la arquitectura *IMS* y pueden de alguna forma, interactuar con los usuarios que acceden a la aplicación *TDi*.

A continuación la figura 26 muestra el diseño detallado de la arquitectura con todos los elementos involucrados y sus respectivas interacciones, además de los protocolos usados para realizar la comunicación, señalización, control y entre otras funciones entre las entidades que se presentan.

4.1.1 Descripción detallada de la arquitectura base.

En la sección 1.5 de este documento se hizo la descripción de los elementos mínimos necesarios para el despliegue de aplicaciones en un sistema *TDi*, en donde se hace mención de un elemento fundamental encargado de entregar los contenidos al dispositivo receptor para su posterior visualización: el servidor de televisión compuesto a su vez por un *servidor de layout* quien se encarga de la codificación del audio y del video; la generación del carrusel de objetos, datos y eventos; la generación de la información de programas, servicios y aplicaciones; la multiplexación de todos los elementos para la generación de un único flujo de transporte *MPEG-2* y una tarjeta moduladora encargada de realizar la adaptación de los flujos de televisión a una señal adecuada para su transmisión [3][15].

Como se observa en la figura 26, la señal de salida del servidor de televisión puede estar directamente conectada hacia un *STB*, este recibe la señal en el formato estándar *DVB-C* y lo convierte a un formato de codificación (*PAL*) compatible para la recepción en el televisor.

Otra opción como se mencionó anteriormente es el uso de tarjetas para la recepción de televisión digital conectadas al puerto *USB* o al puerto *PCI* de un computador. Esta tarjeta lo

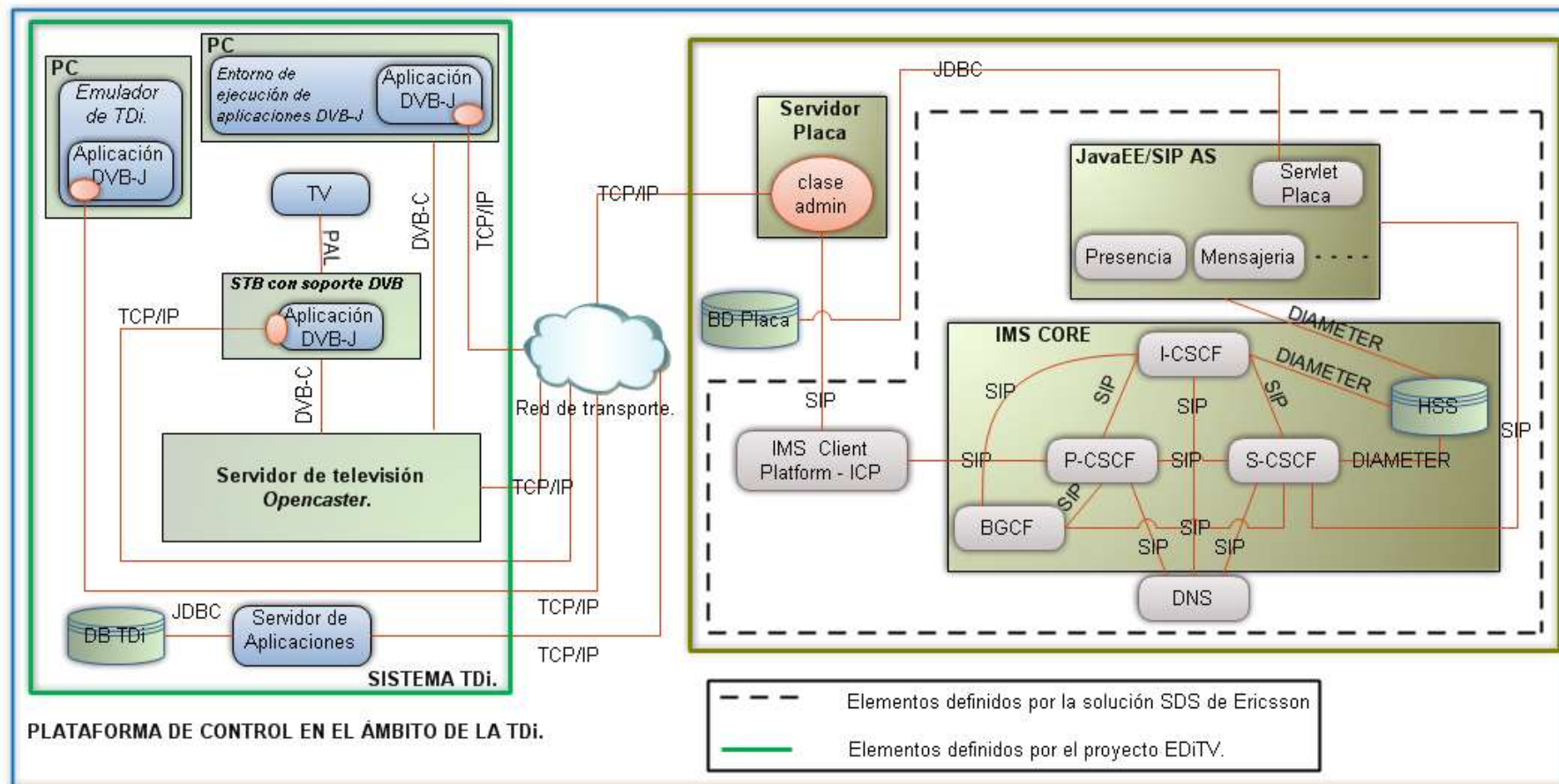


Fig. 26. Diseño detallado de la arquitectura base para el soporte de servicios en el entorno de la TDi.



convierte en un equipo receptor de la señal *DVB-C* y al mismo tiempo en un televisor para la posterior visualización por medio del software con el cual viene junto a la tarjeta que simula el ambiente de ejecución de la aplicación dentro del dispositivo receptor del sistema de *TDi*.

Además de estas opciones, y quizá la más importante para la implementación y pruebas por ser la más sencilla, viable es hacer uso de emuladores para Televisión Digital Interactiva. Estos emuladores implementan las características necesarias para la ejecución de las aplicaciones con el modelo de aplicación *DVB-J*.

La aplicación se despliega sobre el emulador y hace el llamado de todos los componentes y recursos necesarios para que se haga sin ningún contratiempo su ejecución. El emulador implementa la interfaz para el funcionamiento del canal de retorno, útil para la comunicación con los demás elementos que componen la arquitectura diseñada.

En cuanto al servidor de aplicaciones y a la base de datos del sistema *TDi*, estas son usadas netamente para la aplicación *DVB-J* cuando se necesite de un repositorio de información o hacer llamados a funciones complejas que no puede realizar el *STB* y se encuentran implementadas sobre este servidor, o a recursos destinados a la construcción del contenido que se le presenta al usuario como es el caso del contenido visual que debe construirse con base en las instrucciones presentadas por la aplicación y a los llamados a los recursos multimedia y datos disponibles tanto en el servidor de aplicaciones como en la base de datos.

Se observa que la aplicación desplegada sobre el *STB*, *PC* o emulador hace uso del canal de retorno para comunicarse con los demás elementos que componen el diseño de la arquitectura. El primer punto de contacto en el diseño de la arquitectura luego de hacer el paso por el canal de retorno es un servidor de aplicaciones llamado *servidor placa*, de vital importancia ya que sobre él se tiene implementadas funciones como el de mediador (*middleware*) encargado de recibir, procesar y enviar los mensajes entre los entornos involucrados, crear instancias de la *clase admin* para procesos de registro y autenticación en *IMS*, validación de usuarios para acceder a los servicios suscritos en *IMS* y la conexión y comunicación entre los usuarios.

Inicialmente, para que el *servidor placa* tenga acceso al perfil de usuario y poder invocar a los servicios *IMS*, este debe encontrarse registrado. Por esta razón, este debe ser el primer procedimiento que debe ejecutar el servidor a su inicio: registrarse en el dominio *IMS* representando al *cliente STB* e inmediatamente pasar a un estado de espera ante alguna petición de algún *cliente STB* proveniente del canal de retorno.

Para hacer el registro, la *clase admin* del *servidor placa* se comunica con el *IMS Client Platform (ICP)* del *SDS de Ericsson*. El *ICP* es una implementación parte de *SDS* que permite el desarrollo de aplicaciones clientes teniendo en cuenta servicios y funcionalidades existentes o desarrollos hechos por terceros utilizando la interfaz de programación de aplicaciones *IMS* de alto nivel que *SDS* ofrece.

Cuando se inicia el *servidor placa*, se crea una instancia de la *clase admin* llamada *cliente admin*, este le indica al *ICP* que quiere validarse en *IMS* enviándole un perfil válido en *IMS*, el nombre de servicio o una lista de servicios válidos para este perfil y un nombre de registro que es el identificador del cliente a validarse. *ICP* hace la comprobación de los



datos recibidos con sus propios parámetros y al final si se realiza una confirmación correcta a la petición de registro se procede a asignar un *p-asserted identity*, el cual es un encabezado obligatorio que indica que el usuario esta registrado y autenticado en *IMS* y puede interactuar con los elementos que la arquitectura ofrece además, de poder invocar a los servicios disponibles. Si este encabezado no está presente o es nulo, el simulador retornara un error *403 forbidden*, y no permitirá el acceso a la red *IMS*. Luego que el *cliente admin* hace el registro en nombre del *cliente STB*, el *servidor placa* abre un *server socket* que comunica el entorno del sistema *TDi* con los demás elementos que componen el diseño de la arquitectura y pasa a un estado de espera por *clientes STB* para que lo contacten por medio del canal de retorno.

Una vez el *cliente admin* se registre en *IMS*, este puede enviar y recibir datos de los demás componentes de la arquitectura como lo muestra la figura 24. Estos son: el núcleo de *IMS* definido por *SDS* compuesto por el *P/I/S-CSCF*, *BGCF*, y el *HSS*, un servicio que emula el funcionamiento de un servidor *DNS*, el servidor de aplicaciones *SIP* y una base de datos que contiene información relevante en cuanto a los datos de la aplicación *DVB-J* y a los *clientes STB*.

Básicamente el *CSCF* es un servidor *SIP* que para el diseño de esta arquitectura, cumple con el control de la sesión de los terminales registrados y se comporta de acuerdo a los estímulos provenientes de los mensajes *SIP*. Además de esto, se encarga junto con los datos consultados por el *HSS* y el *DNS*, de enrutar los mensajes/peticiones *SIP* a los dispositivos o servidores encargados de recibir y procesar estas solicitudes. Básicamente el *CSCF* está compuesto por tres entidades: el *P-CSCF*, *I-CSCF* y el *S-CSCF*.

El *P-CSCF* es el primer punto de contacto en *IMS*, este actúa como un servidor *proxy SIP* el cual recibe y envía mensajes tanto para fuera como dentro del entorno *IMS*. Las funciones que presta en el diseño de la arquitectura son las siguientes:

- Almacena la identidad publica del usuario por defecto para usarlo en el encabezado *p-Asserted-Identity*.
- Envía las peticiones de registro (*SIP REGISTER*) recibidas por el terminal a un determinado *I-CSCF* usando el nombre de dominio provisto por el terminal. Si el usuario corresponde al mismo dominio de red del *P-CSCF* con el cual interactúa, la petición se envía directamente al *S-CSCF*.
- Envía peticiones y respuestas *SIP* recibidas por el usuario a un *S-CSCF* con dirección conocida gracias al proceso de registro.
- Enviar peticiones y respuestas *SIP* al usuario.
- Afirma la identidad de los usuarios con el resto de los nodos de la red.
- Se aplica los mecanismos de privacidad requeridos para ocultar la identidad del usuario ante el exterior de la red.

El *I-CSCF* actúa como un *proxy SIP* localizado en el límite del dominio de la red local. Es un punto intermedio que brinda soporte a la operativa de *IMS* ya que determina el siguiente



salto de los mensajes *SIP*. Las diferentes direcciones de los *I-CSCF* están registradas en el *DNS*, esto es necesario cuando el *P-CSCF* necesita conocer el *I-CSCF* para determinar cual *S-CSCF* ha de servir a cada usuario.

El *S-CSCF* es el nodo central del plano de señalización y tiene la responsabilidad de controlar y enrutar las sesiones destinadas o iniciadas por el usuario en *IMS*. Se comunica con el *HSS* para obtener el perfil de servicio de cada usuario. El perfil de servicio contiene uno o muchos *iFC*⁴⁷ (*Initial Filter Criteria*). El *iFC* define la prioridad del disparo⁴⁸ y al servidor de aplicaciones al cual va dirigida la petición. Cada *iFC* contiene uno o más *Service Point Triggers*⁴⁹ (*SPTs*) que identifican el tipo de petición que hará el disparo del *iFC*.

El *BGCF* es un nodo el cual no se usará en este proyecto, pero está incluido por el *SDS* dentro del núcleo *IMS*. Este tiene la misión de enrutar las peticiones *Tel URI* a redes externas. Este elemento sólo es usado en sesiones que son iniciadas por un terminal y dirigidas a los usuarios en una red de conmutación de circuitos, como por ejemplo la red telefónica pública conmutada.

La *HSS* es la principal entidad de almacenamiento de datos de todos de los suscriptores y datos de los servicios relacionados con el entorno *IMS*. Los datos almacenados incluyen identidades de usuario, parámetros de acceso e información sobre el disparo de servicios.

Los parámetros de acceso son usados para configurar las sesiones e incluyen la identidad privada del usuario y el *password* usado para autenticar al usuario. La información para el disparo de los servicios permite la ejecución de servicios *SIP* e incluye información que define las condiciones que determinan cuando una petición concuerda y además provee datos para enrutamiento usados si este criterio se cumple.

El *DNS* resuelve las sesiones del flujo general de llamadas entre usuarios comparando direcciones de dominio, números de teléfono globalizado, direcciones *IP* y protocolos de transporte.

En el lado del servidor, los servicios en *SDS* están construidos sobre una arquitectura abierta basada en *Java* con soporte para *SIP/HTTP Servlets* y *EJB* (*Enterprise Java Beans*) usando el servidor de aplicaciones *SIP Sailfin* de *Sun Microsystems*.

El servidor de aplicaciones *Sailfin* es producto de la colaboración entre *Sun* y *Ericsson* en donde *Ericsson* contribuye con los estándares basados en el servidor de aplicaciones *SIP Servlet 1.0* al proyecto de código abierto *GlassFish/Sailfin* bajo la licencia *CDDL*⁵⁰ (*Common Development and Distribution License*). En un futuro *Ericsson* tomara propiedad de este proyecto con ánimos de desarrollar un contenedor *SIP* con la especificación *JSR-289* (*Sip Servlet Api 1.1 - SSA 1.1*).

SDS soporta el desarrollo de aplicaciones de acuerdo con la especificación *JSR-116* y parte de la especificación *JSR-289*. *JSR-116* define al *SIP Servlet API* mientras que el *JSR-289*

⁴⁷ *iFC* (*initial Filter Criteria*): Parámetro que define la prioridad de disparo y el servidor de aplicación al cual la petición va a ser enviada.

⁴⁸ Si hay más de una *iFC* asignada para un suscriptor, la prioridad describe el orden en el que el *S-CSCF* las evalúa.

⁴⁹ *SPT* son las reglas que determinan en qué circunstancias el servidor de aplicaciones es contactado.

⁵⁰ *CDDL* es una licencia de software de código abierto la cual permite el libre acceso del usuario al mismo siempre y cuando se respeten las condiciones que imponga esta licencia.



actualiza la especificación *JSR-116* y define un modelo estándar de programación de aplicaciones para mezclar *SIP Servlets* y componentes *Java EE*.

La *base de datos placa* que esta fuera del entorno *SDS* y del sistema de *TDi* como lo muestra la figura 23, es consultada por la lógica del servicio implementado sobre el *servlet placa* en el servidor de aplicaciones *SIP*. Esta almacena los datos de los *clientes STB* para efectos de validación y autorización, además es un repositorio de almacenamiento auxiliar relacionado con los servicios que se tienen desplegados sobre *IMS* y entre otras opciones que pueden ser añadidas a futuro como apoyo a nuevos servicios.

4.1.2 Diagrama de paquetes para la arquitectura propuesta

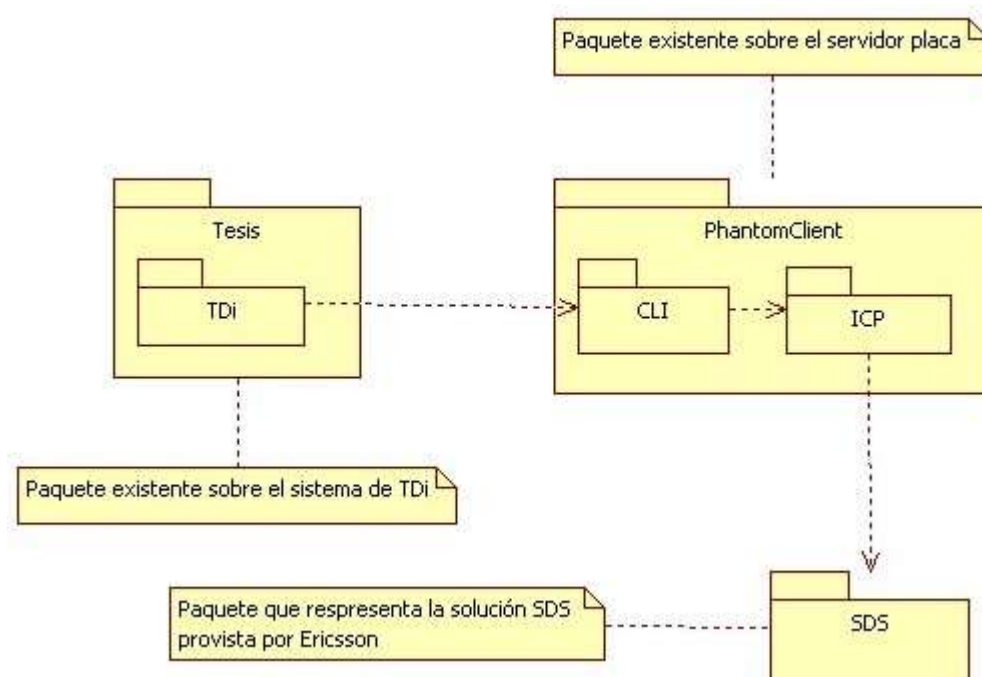


Fig. 27. Diagrama de paquetes para la arquitectura propuesta

La figura 27 muestra el diagrama de paquetes de la arquitectura planteada donde se tiene en cuenta los dos entornos involucrados y la dependencia existente entre los paquetes definidos para la arquitectura propuesta.

Por una parte el paquete *Tesis*, donde se encuentran diversos paquetes relacionados con el sistema de televisión digital interactiva, pero a nivel de arquitectura se presenta el paquete *TDi*. Principalmente este paquete contiene las clases y métodos necesarios para la mediación entre la aplicación de *TDi* y la *clase admin*.

Una clase muy importante dentro de este paquete es *HiloEscuchar*, quien se encarga de recibir y enviar los mensajes en el *STB*.

Por otro lado se tiene el paquete *PhantomClient*, donde a nivel de arquitectura se definen dos paquetes: *CLI* e *ICP*.



Sobre el paquete *CLI* existen las clases *hiloservidor*, *hiloescuchar* y *servidor*, quienes se encargan de hacer la mediación entre la arquitectura de red de telecomunicaciones *IMS* y el sistema de televisión digital interactiva además, de atender peticiones de acuerdo a eventos producidos por la aplicación *TDi* o cualquier evento producido en la arquitectura *IMS*.

El paquete *ICP* contiene las clases *cliente admin*, *ProfileAdapter*, *PlatformAdapter* y *ServiceAdapter*, y además de las clases necesarias para la interacción con el *core* de la arquitectura de red de telecomunicaciones *IMS* dentro el paquete *SDS* a través de las *API's* que ofrece la arquitectura *ICP*.

Por último se tiene el paquete *SDS* provisto por *Ericsson*, donde se tiene el *core de IMS* y sus interfaces definidas mediante las cuales el *ICP* puede enviar y recibir peticiones *SIP*.



4.1.3 Diagrama de clases de la arquitectura propuesta

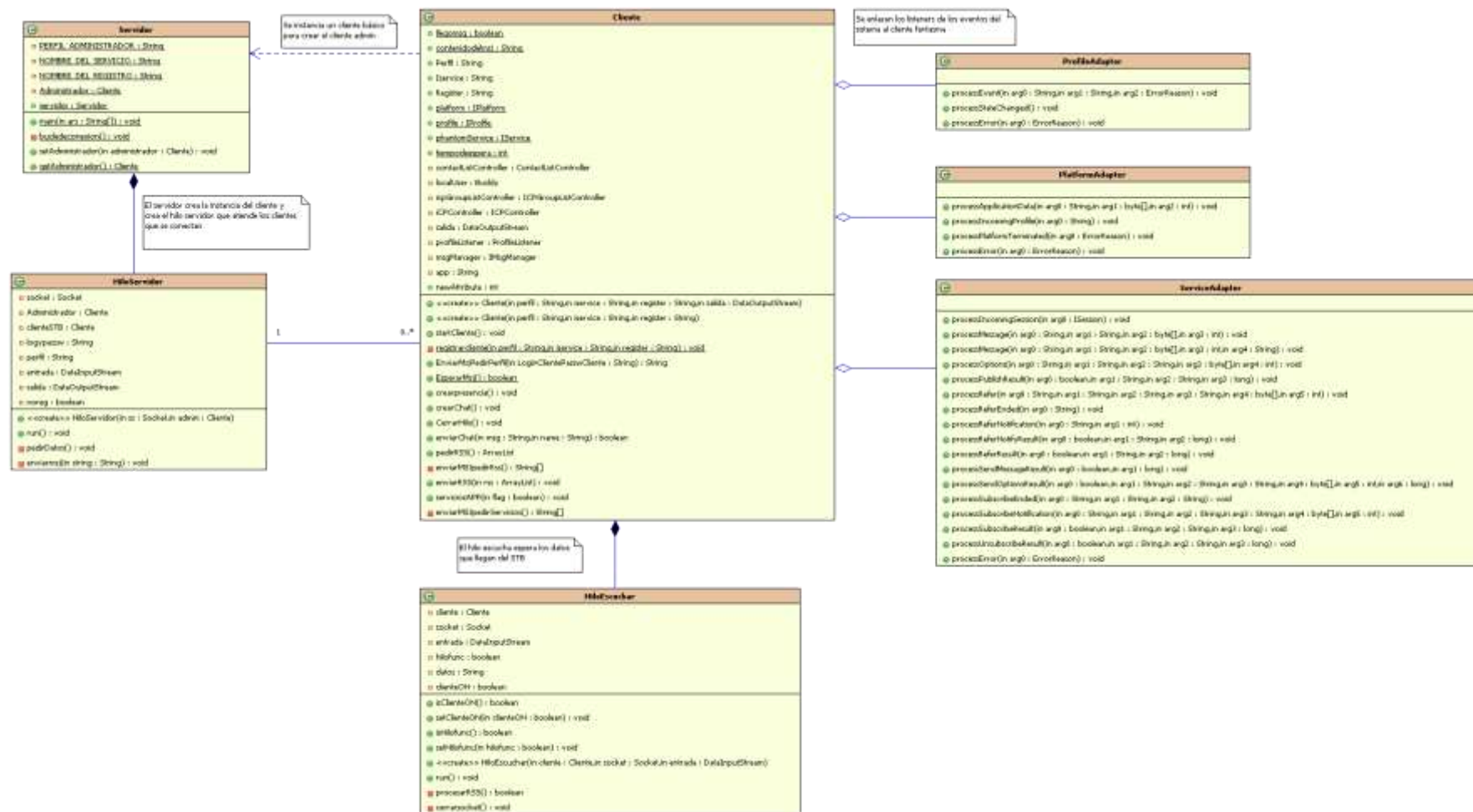


Fig. 28. Diagrama de clases para la arquitectura propuesta



4.1.4 Diagrama de secuencia para la arquitectura propuesta.

Los diagramas de secuencia representados en las figuras 29 y 30 muestran el paso de los mensajes y las relaciones existentes entre las distintas entidades que componen la arquitectura propuesta.

Inicialmente se asume que la aplicación esta ejecutándose en el sistema de *TDi*, sea mediante un emulador de televisión digital, de un *STB* o en un computador con su respectiva tarjeta de recepción de señal de televisión digital.

Por otra parte el *servidor placa* como se comentó en el apartado anterior, debe estar registrado en *IMS* para permitir el envío de mensajes del *cliente STB* a las respectivas entidades además, este servidor debe permanecer en un estado de escucha ante cualquier mensaje de un *cliente STB* proveniente del canal de retorno del sistema *TDi*. Se describió además que el proceso de registro en *IMS* consistía en la creación del *cliente admin* a partir de instanciar de la *clase admin*.

El *cliente admin* se encarga de realizar el registro y autorización en *IMS* y de representar a los *clientes STB* que se conecten al *servidor placa*. Este hace la petición de registro al *ICP* enviándole una serie de parámetros el cual los recibe y hace la comprobación de estos con los propios que tiene almacenados y como resultado, si se realiza una confirmación a esta petición de registro se procede a asignar un *p-asserted identity*, el cual es un encabezado que indica que este usuario esta registrado y autenticado en *IMS* permitiendo interactuar con los diversos elementos que la arquitectura ofrece además, de poder hacer el disparo de los servicios disponibles. Este proceso se muestra en la figura 29 donde interactúan las cuatro entidades que intervienen hasta este momento: el *Servidor placa*, *cliente admin*, *clase admin*, el *CSCF*, el servicio *DNS* y el repositorio de datos *HSS*.

Sobre el proceso de registro, autenticación y autorización que efectúa el *ICP* con el núcleo *IMS*, transparente a los usuarios, se hace el siguiente análisis: la arquitectura *IMS* desde la perspectiva netamente de estandarización, implementa una solución de autenticación y acceso especificada en la *TS 33.203* del *3GPP* llamada *Access Security for IP-Based Services* comúnmente llamada *IMS AKA* (*IMS Authentication and Key Agreement* descrita en el Anexo B).

En el entorno del *SDS*, el *ICP* es la arquitectura encargada para el desarrollo de aplicaciones clientes como se mencionó anteriormente. Respecto a los componentes los cuales integran al *ICP*, se desataca al *ICP runtime core* el cual es un elemento clave para este proyecto.

El *ICP runtime core* se ejecuta como un servicio en *background*, el cual ofrece una plataforma común para todas las aplicaciones *IMS* instaladas y desarrolladas usando el *API ICP*. Este maneja todas las interacciones con la infraestructura de red del núcleo *IMS* y las interacciones con los servidores de aplicaciones.

Los parámetros necesarios para que el servicio *ICP* ejecute los procesos de autenticación y acceso usando el *IMS AKA* están ubicados sobre el *User Profile* el cual contiene la información usada para la identificación y registro en la red *IMS*. Los siguientes son los datos necesarios por el *User profile*:



- **Public User ID:** Es el nombre de usuario visible para los demás clientes. Esta información es usada por los otros usuarios para establecer sesiones. Una *Public User ID* puede consistir de un *TEL URI* o de una dirección *SIP URI*.
- **Private User ID:** Es un nombre de usuario único para registrarse en el núcleo *IMS*.
- **Password:** Es la palabra secreta usada para el registro a la red *IMS*.

Una vez configurados estos parámetros hay que tener en cuenta los estados en que puede funcionar el *ICP runtime core*:

- **Activo:** *ICP* está registrado al núcleo *IMS* cuando un cliente está instalado. El *ICP* está ejecutándose y listo para servir al cliente.
- **Inactivo:** No hay clientes instalados y el *ICP* no está registrado al núcleo *IMS*. El *ICP* no está en ejecutando y se encuentra listo para realizar cualquier cambio en el cliente como en sus parámetros de ejecución.

Cabe mencionar el proceso que se ejecuta cuando el *ICP runtime core* esta en la transición entre un estado inactivo a un estado activo, automáticamente este toma los parámetros del *User Profile* y realiza el proceso de registro y acceso a la red *IMS*, es decir, todo este proceso es manipulado y ejecutado mediante el *ICP*.

En conclusión, *SDS* ofrece una solución extremo a extremo con elementos y funciones ya definidas. En el lado del cliente, obligatoriamente se hace la implementación de estos mediante el uso del *API ICP* el cual me brinda los métodos necesarios para hacer la invocación tanto de los servicios como funcionalidades que la arquitectura *IMS* ofrece, además de hacerse el despliegue de los clientes sobre la plataforma *ICP* para su ejecución. Para la propuesta de este proyecto hay dos aspectos fundamentales dentro de esta arquitectura a tener en cuenta:

- La aplicación *DVB-J* no puede ser implementada usando la arquitectura *ICP*, ya que esta aplicación debe ejecutarse sobre el sistema de *TDi*.
- El dispositivo de acceso a la red *IMS* (en este caso el *STB*), no es un dispositivo *IMS* y por lo tanto no soporta el protocolo *SIP*.

Por estas razones, se han definido la *clase admin*, al *cliente admin* y al *cliente fantasma*, quienes se implementan a partir de las funcionalidades que ofrece el *API ICP* para realizar tareas que están descritas a los largo de este capítulo como los son el registro, la autenticación y acceso a la red *IMS*, el envío de mensajes y la implementación del middleware para los dos entornos involucrados.

Ya aclarado este proceso, retomamos la descripción del diagrama de secuencia.

Una vez confirmado el registro y la autenticación en *IMS*, tarea ejecutada por *admin*, el *servidor placa* pasa a un estado de escucha el cual atenderá a los *clientes STB* que se comuniquen con él a través el canal de retorno, asignándoles a cada uno un *Server Socket* el cual es el canal de comunicación entre los dos entornos.

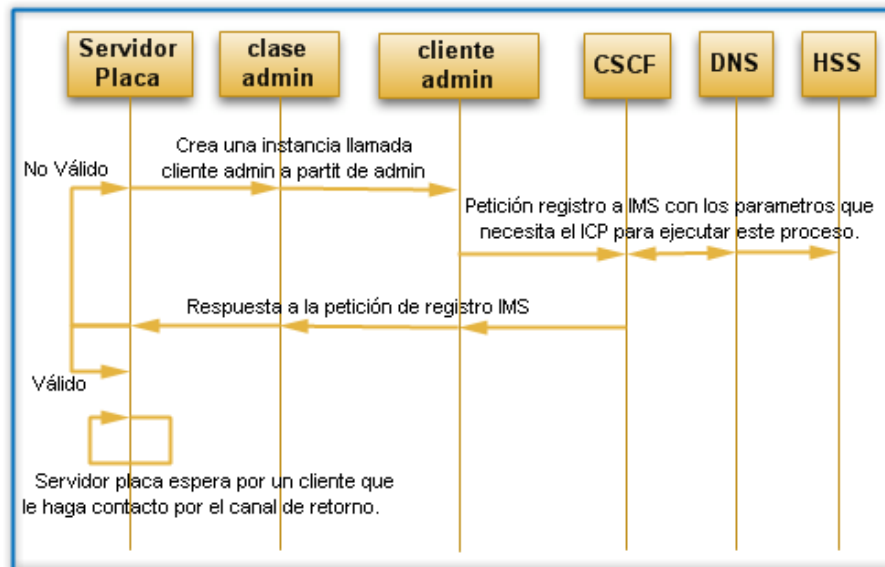


Fig. 29 Diagrama de secuencia del diseño de la arquitectura base para el soporte de servicios en el sistema TDi.

El diagrama de secuencia de la figura 30, muestra el flujo de los mensajes y la relación existente entre las entidades para el caso en que el *servidor placa* se encuentra registrado y autorizado para acceder a los recursos *IMS* y posteriormente un cliente *STB* se conecta a él.

Una vez desplegada la aplicación *DVB-J* en el entorno de la *TDi*, esta le pide al usuario hacer su respectiva validación mediante el ingreso de su *login* y *password*. Cuando estos datos sean ingresados se envían a través del canal de retorno del *STB* hacia el *servidor placa*, este servidor está permanentemente en estado de escucha hasta que un cliente se conecte y posteriormente se proceda a la recepción de los datos.

Cuando los datos de validación llegan al *servidor placa*, el *cliente admin* (quien es un usuario registrado en *IMS* y representa a los *clientes STB* que se conectan al *servidor placa*) los toma y los procesa convirtiéndolos en un mensaje *SIP*. Este mensaje para ser atendido por el *servlet placa* (quien tiene la función de hacer la validación de los usuarios) tiene que viajar a través del núcleo *IMS* y llegar hasta el servidor de aplicaciones donde se encuentra el *servlet placa* ejecutándose quien atenderá esta petición. La petición comienza su viaje dirigiéndose al primer punto de contacto con la red *IMS*: el *P-CSCF*.

El *cliente admin* se comunica con el *P-CSCF* en representación del *cliente STB* que se encuentra en ese momento conectado con el *servidor placa*, le entrega el mensaje *SIP* y a continuación lo remite al *S-CSCF* evitando el paso por el *I-CSCF*, ya que para el entorno del proyecto, tanto el usuario como el *P/I/S-CSCF* se encuentra en el mismo dominio de red.

Una vez el mensaje se encuentre en el *S-CSCF* este consulta al *DNS* por la dirección del *HSS*, luego se comunica con el *HSS* y consulta por el perfil del servicio del usuario registrado en *IMS*, este perfil del servicio contiene uno o mas *iFC* (*Initial Filter Criteria*) donde se define el servidor de aplicaciones al cual se le envía el mensaje además, este *iFC* contiene uno o más *Service Point Triggers*(*SPT*) que identifican el tipo de petición que dispara el *iFC*. En el momento en que el *S-CSCF* haga la comparación y encuentre una coincidencia entre el mensaje y el *STP*, se hace la consulta al *DNS* por el servidor de aplicaciones asociado con el *iFC* y a continuación envía el mensaje a este para que el *servlet placa* haga el respectivo proceso de validación de usuario.



Cuando el mensaje llega al servidor de aplicaciones, el *servlet placa* lo toma y compara los datos de usuario contenidos en el mensaje con los datos consultados de la *base de datos placa*, una vez realizado el proceso de validación de usuario por parte del *servlet placa*, este genera la respuesta a esta petición la cual viaja de regreso hacia la aplicación *DVB-J* por el camino inverso tomado por el mensaje de validación es decir, la respuesta parte desde el servidor de aplicaciones donde está el *servlet placa* y la pasa al *S-CSCF*, este sin necesidad de consultar al *DNS* lo remite al *P-CSCF* y posteriormente lo envía al *servidor placa* para que el *cliente admin* haga la conversión de esa respuesta encapsulada en protocolo *SIP* a un formato comprensible por la aplicación *DVB-J* la cual la procesa y presenta el resultado de la validación al *cliente STB*.

La aplicación *DVB-J* le presenta al *cliente STB* la pantalla de validación siempre y cuando esta no se realice de manera exitosa. Si la validación es correcta la aplicación procederá a presentar al *cliente STB* el contenido de *TDi* sumado a los servicios los cuales está suscrito y de los cuales puede hacer uso.

Por un lado, el contenido de Televisión Digital Interactiva es llamado desde su propio entorno, se hace la invocación de los recursos multimedia necesarios y entre otros aspectos para la construcción visual y de navegación por medio de las instrucciones implementadas en la aplicación *DVB-J*. Aparte de encargarse del servicio de *TDi*, la aplicación simultáneamente debe recibir la información concerniente al perfil de usuario respecto a los servicios que tiene acceso, esta información proviene de la consulta realizada por el *servlet placa* al *HSS* en el momento de confirmarse la validación del *cliente STB*, esta consulta viaja de la misma manera como lo hizo la respuesta a la petición de validación de un *cliente STB*, recordando además el paso que debe hacer estos datos de estos datos por el proceso de traducción de protocolo *SIP* a un mensaje entendible⁵¹ por la aplicación *DVB-J*.

Una vez cargada la presentación del contenido tanto visual, como del esquema de navegación y otros elementos, debe implementarse una interfaz entre la aplicación y los servicios *IMS* habilitados para y por el usuario, es por eso que debe existir un canal de comunicación para los *clientes STB* que sirva por una parte, enviar y recibir mensajes de los servicios proveídos por *IMS* al entorno de *TDi* y por otro lado, conectar y comunicar este cliente con usuarios registrados en la red *IMS*. Para lograr este objetivo se implementa el *cliente fantasma* que se ejecuta en el *servidor placa*, el cual recibe los mensajes provenientes de la aplicación *DVB-J* los traduce a un mensaje *SIP* y posteriormente hace el disparo de los *servlets* que implementan los diferentes servicios disponibles en *IMS*. El *cliente fantasma* es muy similar al *cliente admin*, ya que estos dos hacen su instanciación a partir de la *clase admin*, la diferencia radica en que el *cliente fantasma* aparte de hacer la traducción de mensajes, tiene la capacidad de conducir los mensajes a través del núcleo *IMS* destinados a hacer el disparo de los *servlets* contenidos en el servidor de aplicaciones *SIP* que implementan los servicios disponibles de la red *IMS*, otra función que desempeña es la de conectar y comunicar a la aplicación *DVB-J* con los usuarios registrados de *IMS*.

⁵¹ El proceso que ejecuta el *servidor placa* relacionado con la traducción de protocolo *SIP* a un mensaje entendible para la aplicación *DVB-J* y viceversa, es simplemente la construcción y extracción de información que el servidor debe hacer sobre un mensaje *SIP* tipo *MESSAGE*, donde principalmente existen los campos para el destino, el destinatario, la ruta la cual tiene que seguir este mensaje y el contenido del mensaje.

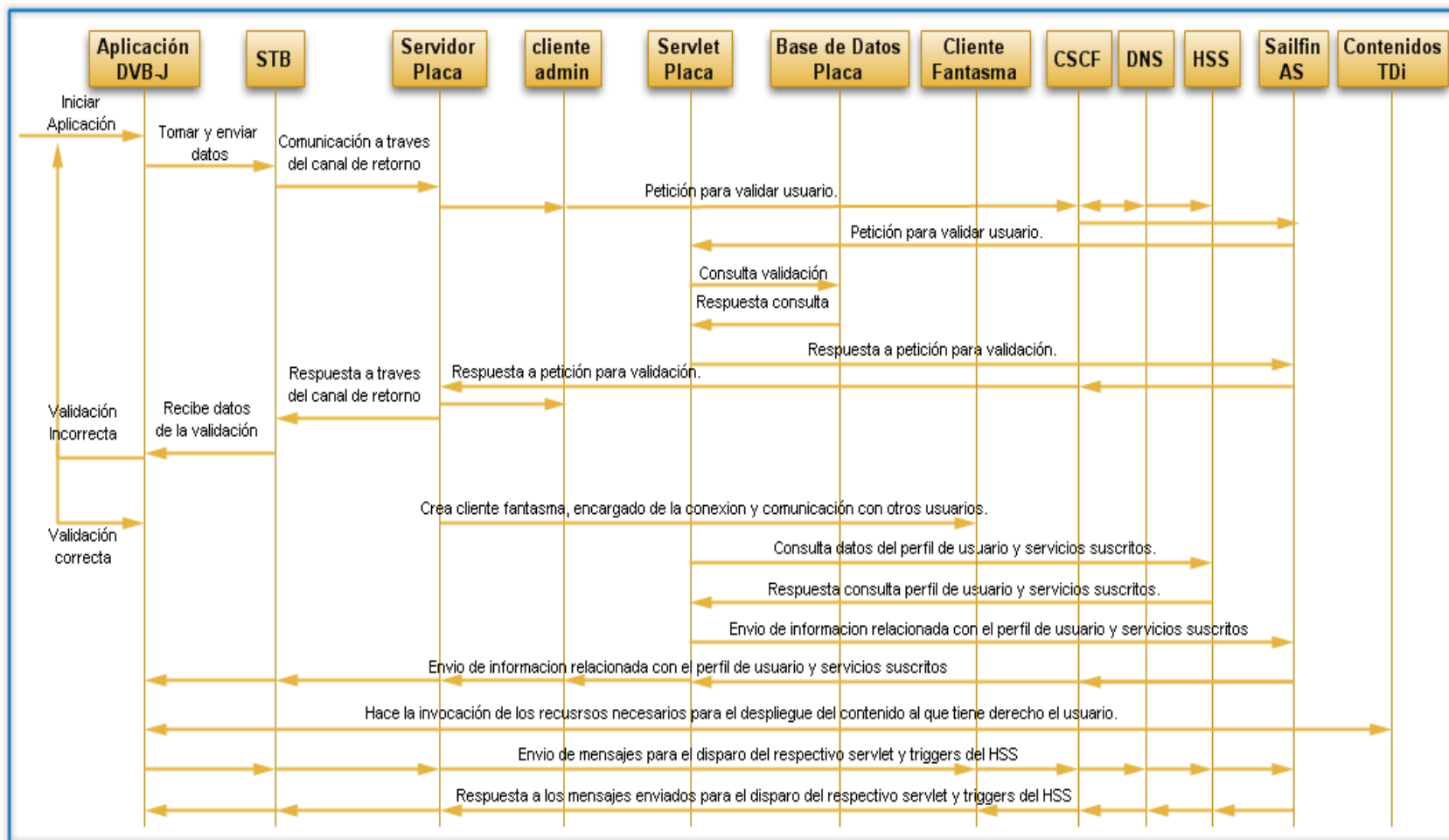


Fig. 30 Diagrama de secuencia del diseño de la arquitectura base para el soporte de servicios en el sistema TDi



4.2 PUESTA EN MARCHA Y VERIFICACIÓN DEL FUNCIONAMIENTO DE LA PLATAFORMA PROPUESTA

Se presenta la puesta en marcha de la plataforma de control en el entorno de la *TDi* y se verifica el funcionamiento de esta mediante la inspección de los historiales (*logs*) que ofrecen cada una de las herramientas usadas para la implementación de la plataforma. Para la herramienta *SDS* de *Ericsson* usada en la implementación de la plataforma propuesta, el Anexo C ofrece la información relacionada con la configuración realizada a esta herramienta.

Básicamente la puesta en marcha consiste en ejecutar todos los nodos involucrados en la arquitectura de la plataforma propuesta, comenzando por la arquitectura de red de telecomunicaciones *IMS* y posteriormente con el servicio *DNS*, ya que estos son el soporte de la plataforma tanto para los servicios, aplicaciones y demás recursos ofrecidos a la aplicación *DVB-J*, como del mecanismo de traducción de direcciones para identificar, localizar y enrutar a las entidades que componen *IMS*.

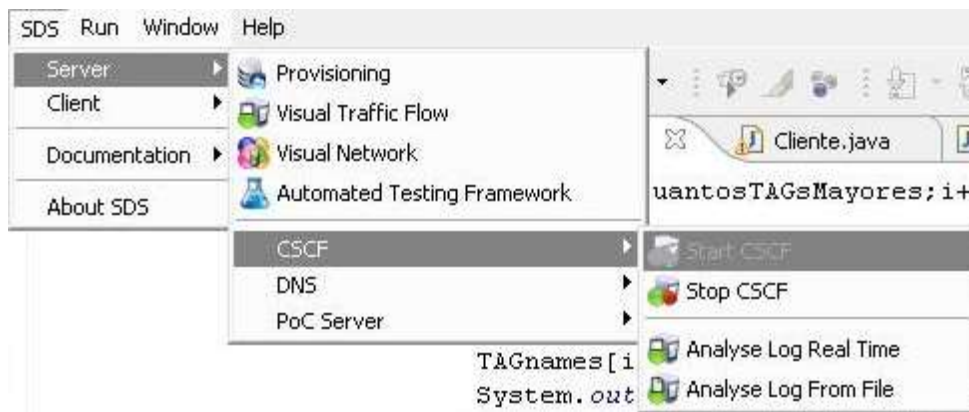


Fig. 31 Ejecución del servidor *CSCF*.

Para dar inicio al servidor *CSCF*, se usan las opciones de la pestaña *SDS* del entorno de desarrollo *Eclipse IDE* como muestra la figura 31, donde se encuentran los botones correspondientes para controlar al servidor *CSCF*.

Una vez seleccionada la opción *Start CSCF*, se inicia al servidor y posteriormente se hace la verificación del estado actual en que se encuentra mediante la inspección en el historial *CSCF Console Log* como lo muestra la figura 32.

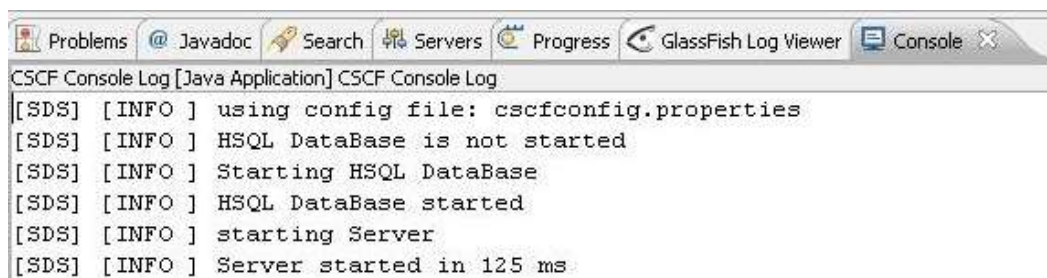


Fig. 32. Historial correspondiente al servidor *CSCF*.



Iniciado el servidor *CSCF* se procede con ejecutar la orden de inicio del servicio *DNS*, proceso similar al realizado con el servidor *CSCF* usando las opciones en la pestaña *SDS*, pero relacionado con el servicio *DNS* y al final, se verifica su estado en el historial de sucesos *DNS Console Log* como lo muestra las figuras 33 y 34.

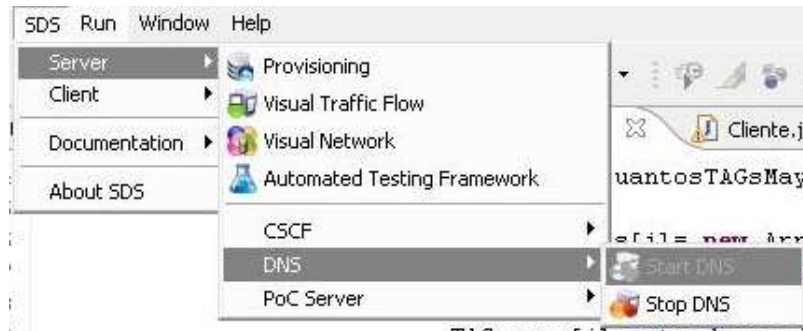


Fig. 33. Inicio del servicio DNS.

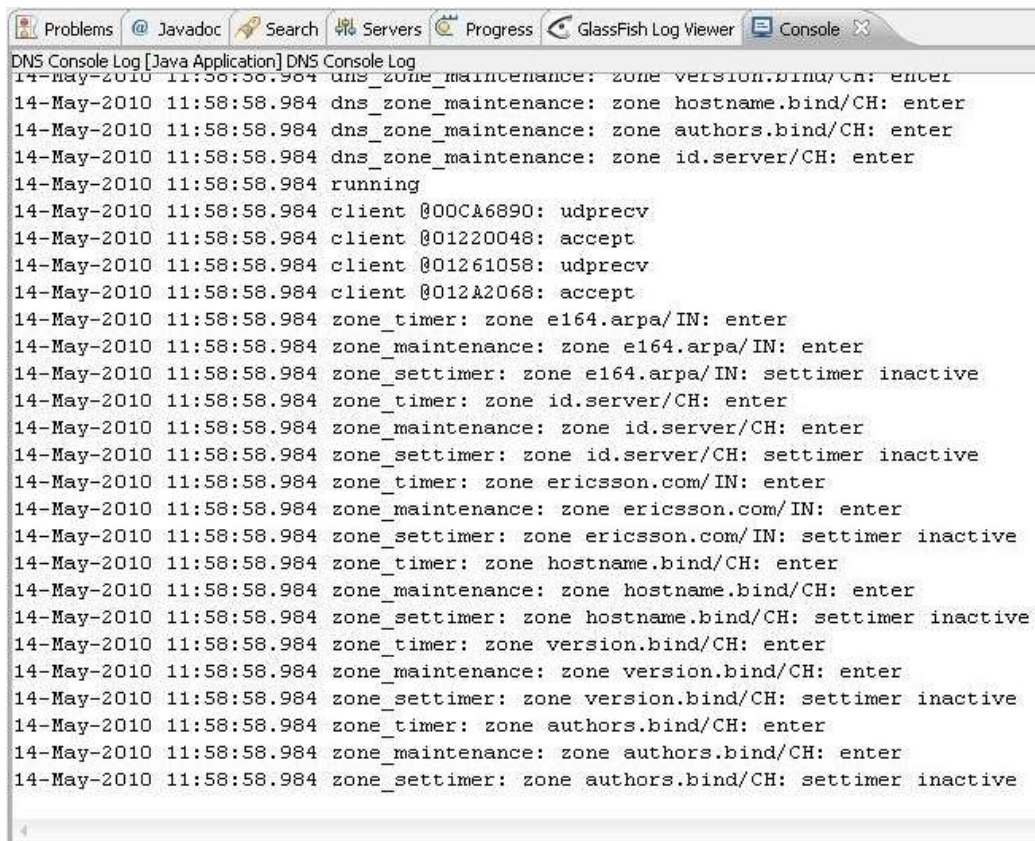


Fig. 34. Historial relacionado con el servicio DNS.

Funcionando de manera correcta la base de la plataforma de control, se procede a la ejecución del *servidor placa* donde se encuentran las clases *cliente admin*, *clase admin* y *cliente fantasma* encargadas de la mediación entre el entorno de la *TDI* e *IMS*.

Como se observa en la figura 36, se tiene el proyecto donde se implementa al servidor *placa* con nombre *zPlacaServletH116*, el cual se despliega sobre el servidor de aplicaciones *Sailfin*, por lo que este servidor previamente debe estar ejecutándose como lo muestra el historial del servidor de aplicaciones en la figura 35.

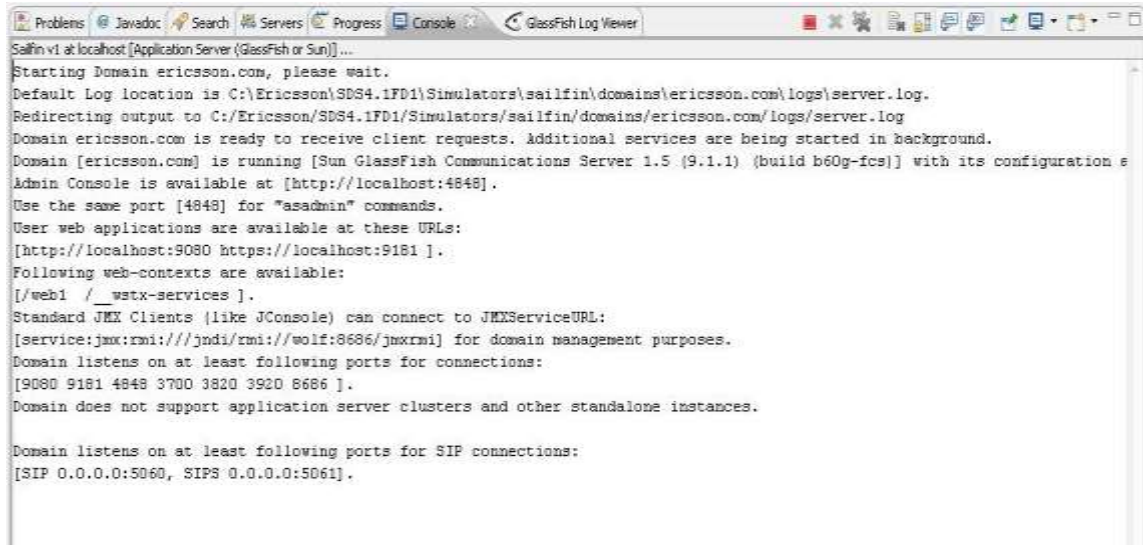


Fig. 35. Ejecución del Servidor de Aplicaciones Sailfin.

Una vez ejecutándose el servidor de aplicaciones *Sailfin*, se procede a la ejecución del *servidor placa* de la forma como se muestra en la figura 35.

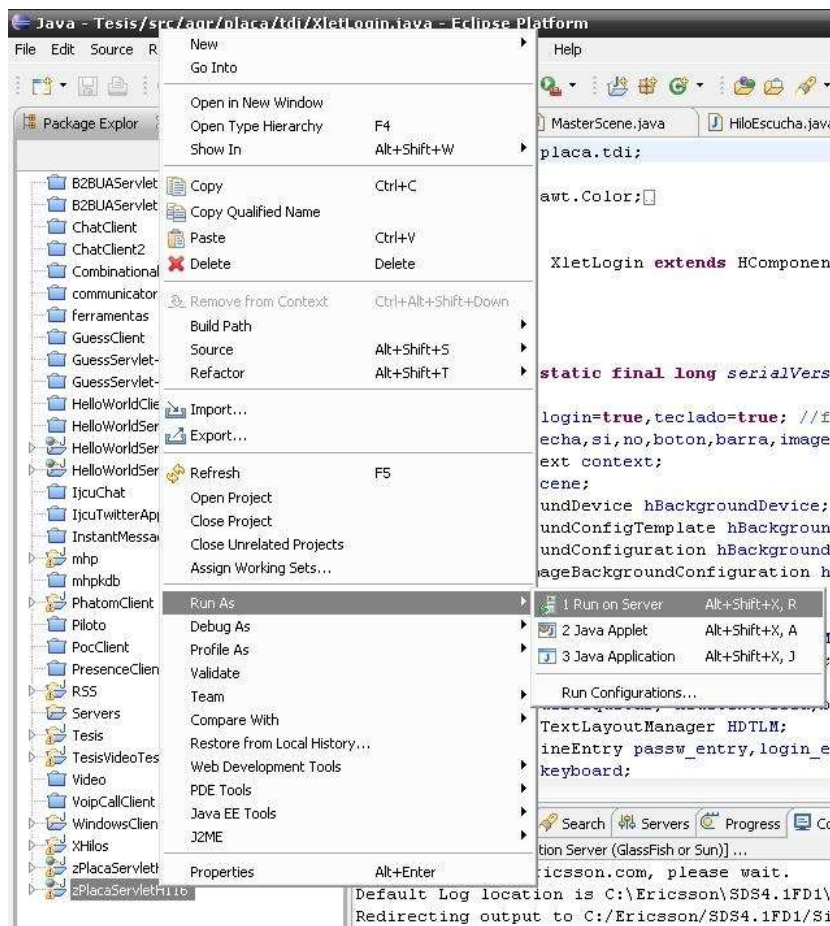
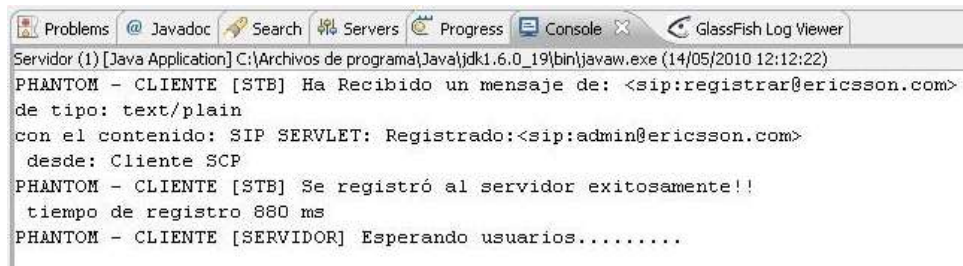


Fig. 36. Inicio del servidor placa.

Y finalmente, se observa en la figura 36 durante la ejecución del *servidor placa*, se realiza el proceso descrito durante este capítulo el cual el *servidor placa* se registra en *IMS*



representando al *cliente STB* y posteriormente pasa a un estado de espera por cualquier solicitud proveniente de la aplicación *DVB-J*.



```
Problems @ Javadoc Search Servers Progress Console GlassFish Log Viewer
Servidor (1) [Java Application] C:\Archivos de programa\Java\jdk1.6.0_19\bin\javaw.exe (14/05/2010 12:12:22)
PHANTOM - CLIENTE [STB] Ha Recibido un mensaje de: <sip:registrar@ericsson.com>
de tipo: text/plain
con el contenido: SIP Servlet: Registrado:<sip:admin@ericsson.com>
desde: Cliente SCP
PHANTOM - CLIENTE [STB] Se registró al servidor exitosamente!!
tiempo de registro 880 ms
PHANTOM - CLIENTE [SERVIDOR] Esperando usuarios.....
```

Fig. 37. Historial durante la ejecución del servidor placa.



CAPITULO 5. VERIFICACIÓN DE LA PLATAFORMA DISEÑADA MEDIANTE EL DESPLIEGUE DEL PROTOTIPO IMPLEMENTADO.

Este capítulo describe el prototipo implementado para verificar el funcionamiento de la arquitectura propuesta. El prototipo comprende la aplicación *DVB-J* en el lado del cliente, que contiene los elementos necesarios para la interacción con el usuario y por otro lado se tiene la arquitectura *IMS*, elemento de vital importancia para el diseño de la arquitectura ya que ofrece las características y funcionalidades necesarias para ser usadas en la implementación de la aplicación.

5.1 DESCRIPCIÓN DEL ENTORNO DE EJECUCIÓN DEL PROTOTIPO.

Una vez propuesta la arquitectura para la Plataforma de Control en el Ámbito de la Televisión Digital Interactiva, el paso siguiente es la implementación del prototipo que verifique el funcionamiento de la arquitectura definida. La implementación del prototipo demuestra el nivel de adaptación que puede llegar a tener funciones que se ejecutan en la arquitectura *IMS* en las aplicaciones en el sistema de *TDi*. Es por eso que la implementación de este prototipo aprovecha las funcionalidades necesarias que le ofrece el diseño de la arquitectura propuesta para ejecutar acciones como autenticación, autorización, control de sesiones y entre otras características que se irán describiendo a lo largo de este capítulo.

Básicamente en la implementación del prototipo tiene en cuenta dos aspectos.

- Las funcionalidades, características y servicios que aporta la arquitectura *IMS* y,
- cómo la aplicación *DVB-J* hace uso de estos recursos para adaptarlos a sus necesidades.

En cuanto a servicios, para el proyecto se define la implementación del servicio de *Presencia* y el servicio de *Chat*, ya que estos dos son ampliamente conocidos y además se pueden crear diversas aplicaciones y servicios a partir de estos.

Respecto al servicio de presencia, básicamente se define como un perfil dinámico del usuario, el cual es visible por otros y usado para representarse a sí mismo, compartir información y controlar servicios. La presencia puede ser vista como el estado de un usuario percibida por otros usuarios, este estado puede incluir cierta información como disponibilidad de la persona y del terminal, preferencias de comunicación, capacidades del terminal, actividad actual, localización, servicios disponibles actualmente como se muestra la figura 38 [62].



Fig. 38 Visión general del servicio de presencia.
Tomado de [62].

Por otro lado, en el capítulo 1 se mencionó al servicio de *Chat* como uno de los servicios más exitosos que se puede ofrecer sobre el sistema de *TDi*, siendo viable su inclusión por la acogida y el éxito que tiene sobre otras plataformas por ser una herramienta de uso cotidiano en distintas áreas como el hogar, el trabajo, estudio entre otras más.

En su forma más básica, es un servicio de mensajería en tiempo real donde los mensajes de cualquier tamaño pueden enviarse y recibirse entre dos usuarios *IMS* mediante el previo establecimiento de una sesión entre los usuarios implicados.

Para efectos de la verificación del proyecto, el servicio de *Chat* se limita a realizar solo la consulta de la información relacionada con el estado de cada usuario ofrecido por el servicio de presencia de *IMS*, las demás opciones que este servicio ofrece no serán tomadas en cuenta para la implementación del prototipo.

Respecto a *IMS*, cualquier servicio puede estar disponible desde cualquier red de acceso y por cualquier dispositivo, esto permite incluir a estos servicios en el prototipo y observar su desempeño sobre la arquitectura diseñada, pero antes de eso, se dio solución a un inconveniente presentado.

Ya que teóricamente los servicios pueden estar disponibles para cualquier red de acceso y para cualquier dispositivo, esto no se cumple siempre y cuando el dispositivo de usuario no reúne con las capacidades y recursos necesarios para soportar al servicio y en el caso concreto para este proyecto, se habla de un televisor conectado a un *STB* los cuales no implementan soporte para el protocolo *SIP*, necesario para interactuar directamente con *IMS*.

La solución está en el *servidor placa* dentro de la arquitectura diseñada, quien se encarga de ofrecer al servicio de *Chat*, la conversión de los mensajes provenientes de un cliente no *IMS* a mensajes *SIP* comprensibles por la arquitectura de red y viceversa, lo cual resulta en que un usuario que acceda al servicio de *Chat* desde un *STB* conectado a un televisor pueda entablar y mantener una conversación con un usuario *IMS* que acceda a este servicio por otro medio.



En general para cualquier servicio que se implemente y se despliegue sobre la arquitectura diseñada, se debe tener en cuenta la implementación de las pasarelas y artefactos *software* necesarios sobre el *servidor placa*, que será en definitiva los directamente responsables de mediar entre el usuario no *IMS* y los recursos, funcionalidades y servicios que la arquitectura pueda ofrecer.

En cuanto a las funcionalidades que ofrece la arquitectura *IMS* a la aplicación *DVB-J* se tiene:

- **Registro en *IMS*:** Este procedimiento transparente para la aplicación *DVB-J* y para los usuarios, es necesario y de vital importancia para el funcionamiento global del sistema, pero en general para todas las solicitudes que genera la aplicación diseñada con destino a la arquitectura *IMS*, ya que para poder acceder tanto a las funciones como a los servicios debe existir un previo registro y autorización para disponer de los recursos requeridos.
- **Autenticación:** Básicamente la autenticación es el método mediante el cual se confirma y se valida las identidades de las personas o dispositivos participantes en la comunicación. Para el prototipo y la arquitectura diseñada existen dos momentos en que se ejecuta este proceso.

Primero cuando el *servidor placa* tiene que registrarse para poder solicitar los servicios de red y luego pasar a un estado de escucha en donde los *clientes STB* se conectan a él y le hacen llegar las solicitudes que luego son procesadas en la arquitectura *IMS*. Por otro lado, la aplicación *DVB-J* solicita la autenticación de los *clientes STB* al *servlet placa* donde se tiene implementada esta funcionalidad, si esta solicitud de autenticación resulta exitosa, se procede a garantizar el acceso a los contenidos y servicios a los cuales el usuario tiene privilegio.

- **Autorización:** Una vez autenticado el *cliente STB*, la arquitectura *IMS* concede la autorización para permitir que la aplicación disponga de los recursos e información que necesita durante su ejecución como por ejemplo los datos relacionados con el perfil de usuario, el perfil de servicio y en general todo lo relacionado según los privilegios asignados al cliente.

La consulta a los datos de los usuarios y a los datos de los servicios, además del disparo de estos es controlado y supervisado por la arquitectura *IMS* mediante el establecimiento de asociaciones de seguridad que protegen a la red de amenazas tales como *Interrupciones* debido a la inutilización de algún elemento del sistema debido a cualquier ataque, *Intercepción* por parte de una entidad no autorizada quien consigue tener acceso a la información, *Modificación* de algún parámetro del sistema ya que puede tener acceso a la información y *Creación* por parte de la entidad no autorizada quien puede insertar cierta información en la comunicación, estas pueden llegar a ser originadas dentro o fuera de la misma

- **Control de acceso.** El control de acceso protege a la red asegurando que sólo los usuarios o dispositivos permitidos puedan usar los recursos que ofrece *IMS*. Una vez validado y autorizado el *cliente STB*, *IMS* le proporciona a la aplicación *DVB-J*



solamente los recursos necesarios y los elementos a los cuales tiene privilegio el usuario.

- **Control de sesiones:** El control de sesiones es la característica que ofrece *IMS* mas importante que puede tomar la aplicación *DVB-J*, ya que mediante el control que hace la arquitectura sobre las sesiones (registro, establecimiento, liberación y mantenimiento) se puede iniciar las comunicaciones con otros usuarios y con los servicios multimedia, como por ejemplo cuando se da inicio y se mantiene una sesión de *Chat* entre el *cliente STB* con un *usuario IMS*.

En cuanto a la aplicación *DVB-J*, esta se implementa y se ejecuta en el sistema de *TDi* fuera del dominio de la arquitectura *IMS*. Esta aplicación inicialmente le solicita al usuario no *IMS* (o *cliente STB*) el ingreso tanto del *login* como de su *password* para hacer la respectiva validación, luego de la validación exitosa, el *cliente STB* accede a las características que le ofrece la aplicación y a los servicios a los cuales tiene privilegio, esto se explica posteriormente en las pruebas realizadas al prototipo.

Desde el punto de vista del usuario, la implementación de la aplicación y sus funcionalidades no revisten de mayor relevancia, el verdadero logro de este proyecto es transparente al usuario ya que todas estas funcionalidades y servicios están provistas por la arquitectura de red *IMS* y la mediación esta soportada por el *servidor placa*.

5.2 DIAGRAMA DE PAQUETES DEL PROTOTIPO IMPLEMENTADO.

A continuación, la figura 39 muestra el diagrama de paquetes del prototipo implementado y se hace la descripción para cada uno de ellos.

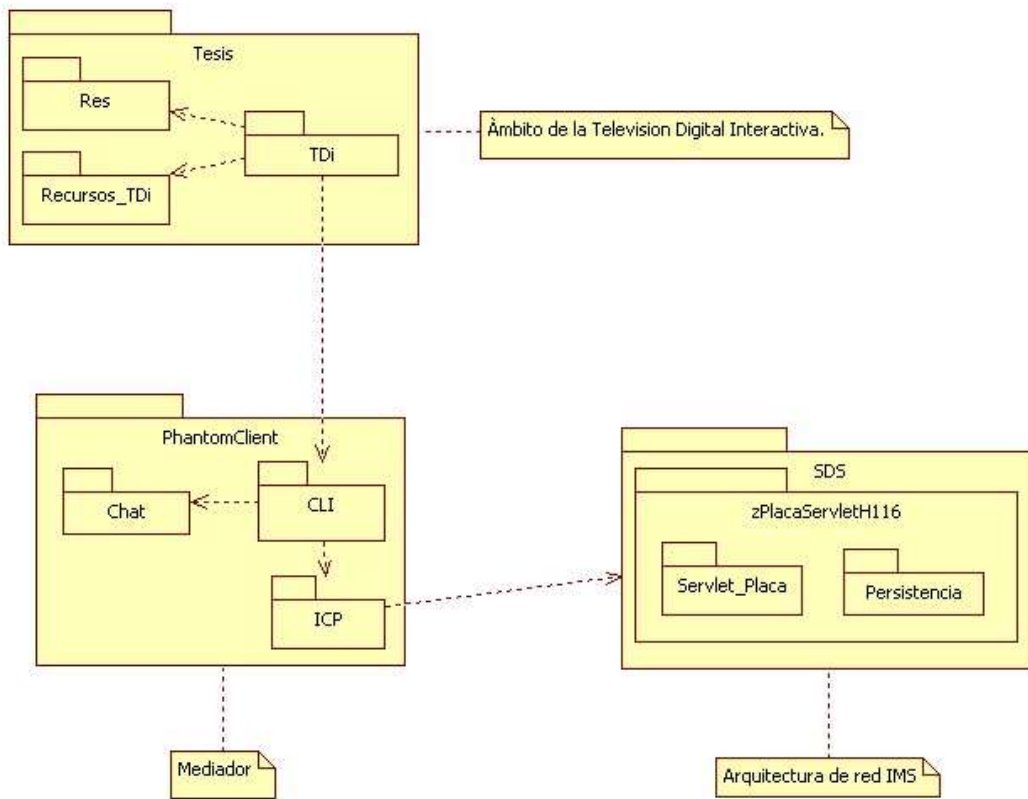


Fig. 39 Diagrama de paquetes para el prototipo implementado.

5.2.1 Paquete Tesis

El paquete Tesis está compuesto principalmente por:

- **TDi:** Contiene las funciones relacionadas con la validación de los *clientes STB* y las interacciones entre la aplicación con los servicios proveídos por *IMS*. Las clases principales son:

XletLogin: Esta clase es la encargada de recibir y enviar los datos del *cliente STB* para su validación.

MasterScene: Esta clase implementa el menú de la aplicación y gestiona las interacciones salientes (las peticiones, flujo de salida del *Chat*, etc).

HiloEscucha: Esta clase hace la mediación entre la aplicación y el *cliente fantasma*, además maneja la lista de amigos y controla los mensajes entrantes y salientes del servicio de *Chat*.

- **Res:** En este paquete se encuentran los fondos, imágenes, sonidos y botones que necesita la aplicación *DVB-J* para construir el contenido visual que se le presenta a los *clientes STB* además, de se encuentra la implementación del *teclado virtual*.



- **Recursos_TDi:** En este paquete se encuentra el contenido multimedia que la aplicación *DVB-J* ofrece a los *clientes STB*. Comprende las fotos y el contenido de video disponible para el sistema de *TDi*.

5.2.2 Paquete *PhantomClient*

Se encuentran implementadas las clases más importantes para el funcionamiento del prototipo. Principalmente contiene la siguiente estructura:

- **Chat:** Contiene las clases relacionadas con el control de la sesión de *Chat* con los usuarios.
- **CLI:** Sobre este paquete se hace la implementación de la clase genérica *clase admin*, donde se derivan el *cliente fantasma* y el *cliente admin*, además se implementan las clases *hiloServidor* e *hiloEscucha*.

La clase *hiloServidor* es la encargada de escuchar si un *cliente STB* se conecta y la clase *hiloEscucha* es la encargada de atender peticiones mínimas, por ejemplo notificar cuando un *cliente STB* se desconecta o para reciclar los recursos que no están en uso.

- **ICP:** Básicamente las clases que se implementan en este paquete están orientadas a la interacción con el *IMS Client Platform (ICP)*. Estas clases importan las librerías que ofrece el *ICP* para realizar el registro en *IMS*, la gestión de los contactos, el estado de estos y entre otras funciones relacionadas con los servicios que *IMS* provee.

5.2.3 Paquete *SDS*

En este paquete se encuentra todo lo que provee la solución *SDS* de *Ericsson*. No se especifica su constitución, ya que la solución solo presenta las *API's* e interfaces necesarias para interactuar con este paquete mediante el envío y recepción de mensajes. Solamente se describe el paquete implementado para la validación de los *usuarios STB*.

- **zPlacaServletH116:** Sobre este paquete se encuentra implementada la lógica del *servlet Servlet_placa*, encarga de la validación de los *usuarios STB* y de la consulta a los datos de los usuarios y del servicio en *IMS* además, el paquete *Persistencia* donde se encuentran todos los métodos relacionados con la capa de acceso a datos mediante *Hibernate*.

5.3 DIAGRAMA DE CLASES DEL PROTOTIPO IMPLEMENTADO

Se presentan los diagramas de clases de los paquetes mencionados anteriormente:



5.3.2 Diagrama de clases para el paquete *PhantomClient*



Fig.41 Diagrama de clases para el paquete *PhantomClient*

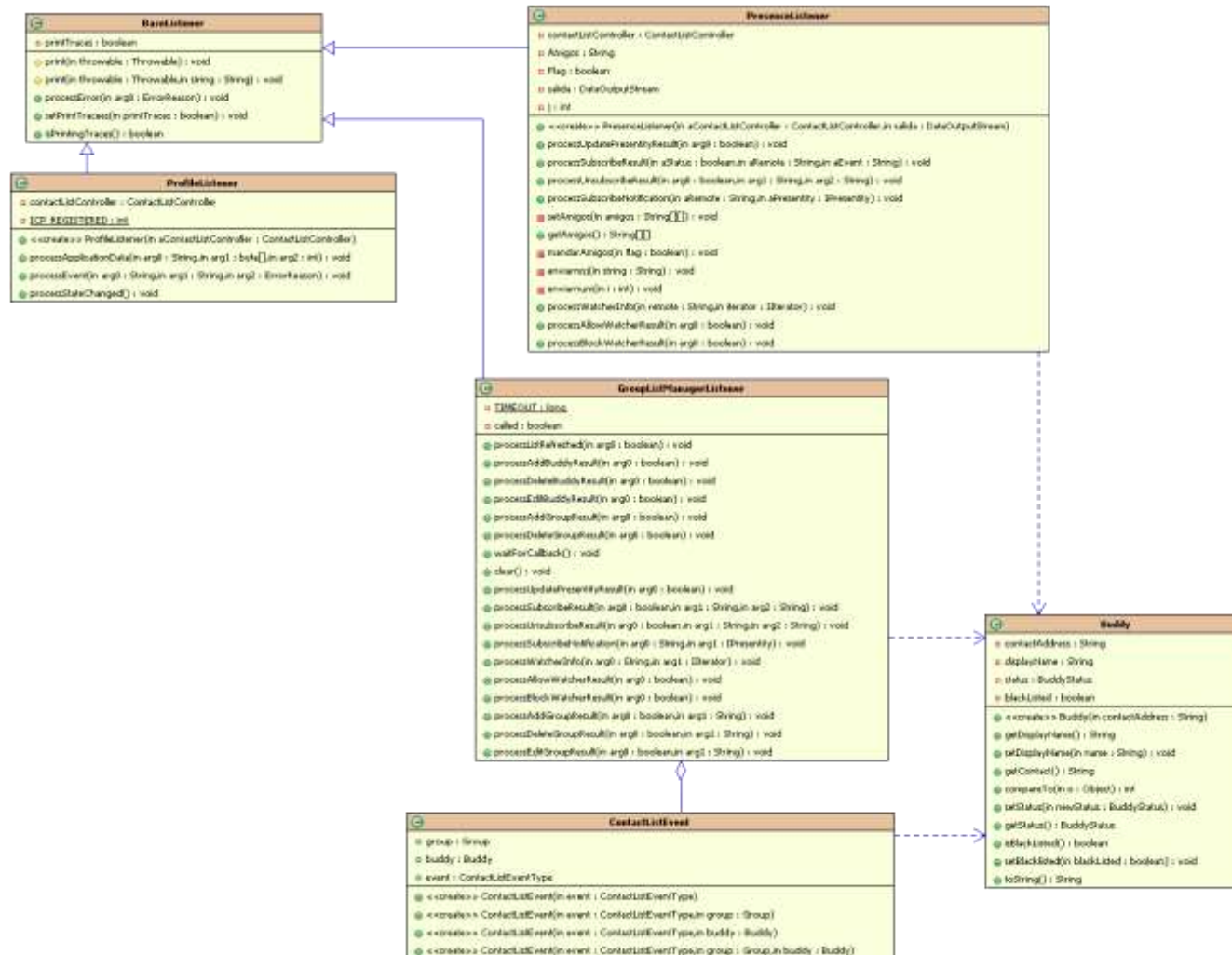


Fig.42 Diagrama de clases para el paquete PhantomClient



5.2 DIAGRAMA DE SECUENCIA DEL SERVICIO DE CHAT IMPLEMENTADO EN EL PROTOTIPO

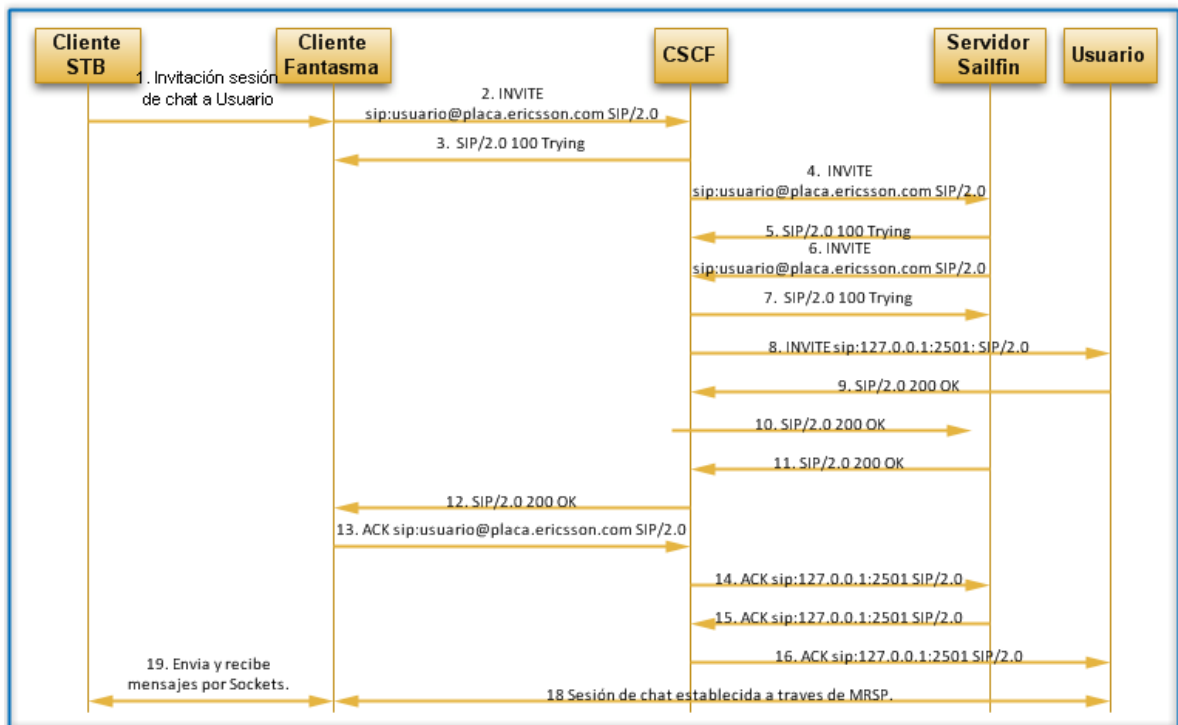


Fig. 44. Diagrama de secuencia del servicio de Chat implementado para el prototipo.

La figura 44 muestra el diagrama de secuencia para el servicio de *Chat* implementado en el prototipo.

Se asume que previamente se ha realizado con éxito el proceso de registro y autorización en *IMS* para acceder tanto a los servicios y a los recursos que esta arquitectura ofrece, también se asume que el *cliente STB* se encuentra autenticado y autorizado para usar los servicios y recursos disponibles.

No se ha definido un diagrama de secuencia para el servicio de *presencia*, ya que el servicio de *Chat simplemente* le solicita el estado de los *buddies* los cuales tiene inscritos para así poder establecer una sesión (si el usuario está *online* o *offline*), no se hace uso de las demás funcionalidades que este servicio ofrece ya que se definió hasta este punto, el alcance de la implementación del prototipo.

Básicamente cuando un *cliente STB* quiere establecer una sesión de *Chat* con *usuario IMS*, se ejecuta el procedimiento mostrado en la figura 44 el cual se describe a continuación:

- 1- Inicialmente se envía la petición para iniciar la sesión de *Chat* desde el *cliente STB* al *cliente fantasma*,
- 2- este recibe el mensaje y genera una petición *INVITE* con destino al usuario con quien se quiere establecer la sesión, para que esta petición alcance al usuario destino, esta tiene que hacer transito por *IMS* iniciando por el *CSCF*.



- 3- Cuando el mensaje llega al *CSCF*, el *cliente fantasma* recibe una respuesta a este esta petición *100 Trying*, la cual indica que el mensaje aun no ha llegado a su destino y aún se encuentra en tránsito.
- 4- El mensaje *INVITE* pasa al servidor de aplicaciones *Sailfin* donde se tiene implementada la lógica del servicio de *Chat*, este recibe el mensaje y determina que hay una solicitud para establecer una sesión de *Chat*, verifica que el usuario esté *online* consultando el estado proveído por el servicio de presencia e inmediatamente enruta la solicitud al usuario destino.
- 5- El *CSCF* recibe la confirmación de la llegada la petición al servidor de aplicaciones mediante la recepción de un mensaje *100 Trying*.
- 6- La petición vuelve al *CSCF*, quien es el que determina el siguiente salto, en este caso será el usuario al cual va dirigida la petición.
- 7- EL servidor de aplicaciones recibe el mensaje *100 Trying* por parte del *CSCF*, indicado que el mensaje llegó a esta entidad y se pasa a enrutar al usuario destino.
- 8- La petición *INVITE* llega a su destino esperando ser aceptada por el *usuario IMS*, para establecer un canal de comunicación entre los usuarios involucrados.
- 9,10,11,12 - Una vez aceptada la invitación, se envía un mensaje *200 OK* para indicarle al *cliente STB* que su petición para establecer una sesión de *Chat* ha sido aceptada, este mensaje viaja de vuelta desde el *usuario IMS*, pasando por el *servidor Sailfin*, el *CSCF* y finalmente al *cliente fantasma*.
- 13, 14, 15,16- Finalmente el *cliente fantasma* envía un mensaje *ACK* al *usuario IMS* Indicando que la negociación para el establecimiento de la sesión de *Chat* ha sido exitosa y que inmediatamente se puede realizar el intercambio de información a través de la sesión.
- 18-Se establece una sesión de *Chat* entre el *cliente fantasma* y el *usuario IMS* mediante el protocolo *MSRP*⁵²(*Message Session Relay Protocol*).
- 19-El *cliente STB* para poder recibir y enviar mensajes a través de la sesión de *Chat* establecida, debe conectarse mediante sockets⁵³ con el *cliente fantasma*, y este a su vez, se encarga de enviar y recibir por la sesión los mensajes y hacer la respectiva mediación. Si el flujo de los mensajes viaja del *cliente STB* hacia *IMS*, el *cliente fantasma* debe empaquetar estas peticiones dentro de mensajes *SIP*. Si el flujo de los datos va desde *IMS* hacia el *cliente STB*, el *cliente fantasma* recibe los mensajes *SIP* y posteriormente los desempaqueta y los pasa a un mensaje comprensible para la aplicación en el *cliente STB*.

⁵² *MSRP* es un protocolo para transmitir una serie de mensajes instantáneos en el contexto de una sesión de comunicaciones.

⁵³ Sockets: Definido por una dirección IP, un protocolo de transporte y un número de puerto en donde se crea un canal de comunicación para el intercambio de cualquier flujo de datos, generalmente de manera fiable y ordenada.



5.3 VERIFICACIÓN DEL PROTOTIPO DESPLEGADO SOBRE LA PLATAFORMA DE CONTROL PROPUESTA.

El prototipo implementado se despliega sobre la arquitectura propuesta, donde se observa su comportamiento, funcionalidades y características que ofrece.

Una vez cargada la aplicación de *TDi*, esta presenta la interfaz de validación a los *clientes STB*.

Luego del ingreso de datos correctos por parte del usuario y de la validación exitosa, la aplicación presenta todo el contenido de televisión, aplicaciones y datos a los cuales tiene privilegio.

Para este prototipo se presenta el menú de opciones donde se tiene para el caso mas general las siguientes opciones:

- **Amigos:** Esta opción permite establecer sesiones de *Chat* con los *buddies* a los cuales el *cliente STB* tiene inscritos. Puede establecer sesión de *chat* con otros *clientes STB* y clientes que acceden a la arquitectura de red *IMS* y hacen uso de este servicio.
- **TV:** Esta opción presenta al *cliente STB* los canales de televisión a los cuales tiene privilegio.
- **APP:** Esta opción presenta las aplicaciones a las cuales el *cliente STB* tiene acceso.

En el anexo D, se muestra de manera extensa, la verificación del prototipo desplegado sobre la plataforma de control propuesta.



CAPITULO 6. APORTES, CONCLUSIONES Y TRABAJOS FUTUROS

6.1 APORTES

6.1.1 Aportes arquitectónicos

- La adaptación que se realizó de funciones como: registro en la arquitectura de red *IMS*, autenticación de clientes *STB*, autorización, control de acceso, control de sesiones, información de los perfiles de servicios y usuarios, recursos y servicios que se ejecutan en el entorno de la arquitectura de red de *telecomunicaciones IMS* en las aplicaciones en el sistema de la *TDi* (*capítulo 4 de la monografía*).
- La arquitectura de la plataforma propuesta por este proyecto muestra la interoperabilidad entre dos entornos de diferente naturaleza, logrado mediante la implementación de las pasarelas y mecanismos necesarios que permiten el paso de la información sin tener en cuenta el entorno en donde se encuentren (*Secciones 4.1 y 4.2*).
- La arquitectura de la plataforma propuesta, incluye al considerar a *IMS* como base para el control de las sesiones, el control de acceso, la lógica de los servicios, los distintos *triggers* para el disparo de estos, deja a disposición la información relacionada con los perfiles de los usuarios y de los servicios, el registro y la autenticación tanto para la red *IMS* como para la aplicación *DVB-J*, todo esto mediante el eficiente control y supervisión por parte de la arquitectura de red de telecomunicaciones ayudado por el establecimiento de asociaciones de seguridad que la protegen de amenazas que se originen dentro o fuera de la misma, todo esto para que la aplicación *DVB-J* en el entorno de la *TDi* tome los recursos necesarios para su ejecución. (*Secciones 4.1, 2.1.5*)
- La arquitectura de la plataforma planteada deja a disposición los servicios desplegados no sólo para el sistema de *TDi*, estos también pueden ser aprovechados por otros sistemas siempre y cuando se hagan las modificaciones necesarias para acceder a los servicios. Esto gracias a que *IMS* despliega los servicios independientes de la red de acceso y del dispositivo terminal de usuario (*2.1.5, Capítulo 4*).

6.1.2 Aportes técnicos.

- Durante las fases de implementación tanto de la arquitectura de la plataforma de control como del prototipo para verificar el funcionamiento de la arquitectura, se realizaron aplicaciones y configuraciones a las herramientas usadas para el proyecto las cuales son un aporte a la comunidad académica de la *FIET* de la Universidad del Cauca como punto de referencia relacionado con los entornos involucrados, esto se puede ver en los recursos que se entregan junto a este documento (*Anexo C, Anexo D*).



- Definición e implementación de la *clase admin*, una clase genérica implementada mediante el lenguaje de programación *Java* de la cual se instancia el *cliente admin* y el *cliente fantasma* para la mediación entre la aplicación construida con el modelo de aplicación *DVB-J* y la arquitectura de red de telecomunicaciones *IMS*, las cuales son el punto de referencia al momento de adicionar nuevos servicios y funcionalidades sobre la plataforma (*Sección 4.1, código fuente*).
- Implementación de la aplicación en el entorno de la *TDi* mediante el uso del modelo de aplicación *DVB-J* definido en el middleware *mhp* además del uso de las *API mhp* necesarias que ofrecen funcionalidades sobre este entorno (*Capítulo 5, código fuente*).
- Implementación de los servicios de *presencia* y de *mensajería* empaquetados sobre *services.sar*, la implementación del *servlet placa zPlacaServletH116.sar* donde se tiene la lógica para la aplicación *DVB-J*, todo esto desplegado sobre el servidor de aplicaciones *Sailfin*, además de la definición de los puntos de disparo *IFC's* para los servicios, y la definición tanto de los perfiles de usuario como los perfiles de los servicios existentes en *IMS* (*Capítulo 5, Código fuente*).
- La implementación dentro del prototipo del teclado virtual y del carrusel del menú de las opciones de usuario, quedan definidas como elementos reutilizables para la implementación de aplicaciones *DVB-J* para versiones *mhp* posteriores a la 1.0.2 (*Capítulo 5, Código fuente*).
- La implementación dentro del prototipo de un agregador *Web feed* para el entorno de la Televisión Digital Interactiva (*Capítulo 5, Código fuente*).

6.1.3 Aporte social

A nivel social este trabajo de grado propone una plataforma de control en el ámbito de la Televisión Digital Interactiva tanto para los usuarios, para los operadores de televisión y de la red de telecomunicaciones, la que permite ofrecer una extenso conjunto de servicios y funcionalidades que pueden ser adaptadas en la emisión de la televisión mejorando así notablemente la experiencia televisiva comparada con la que se obtenía resultado de la emisión analógica de televisión.

6.2 CONCLUSIONES

- La elección de la arquitectura de red de telecomunicaciones *IMS* ofreció para los requisitos que imponía el proyecto (principalmente en el despliegue y ejecución de las aplicaciones sobre la arquitectura diseñada, la carga de los elementos necesarios para la aplicación relacionados con el contenido a los cuales el usuario tiene privilegio, la disponibilidad de los elementos necesarios para usarse durante etapas de diseño, construcción y despliegue de aplicaciones en el entorno de la *TDi*, el control sobre el funcionamiento de la aplicación y de los servicios desplegados, la consideración de los mecanismos para interpretar y direccionar los mensajes entre los entornos involucrados que componen su entorno, entre otros) los mecanismos necesarios (entre las cuales están: funciones para el registro, establecimiento, modificación y finalización de las sesiones entre dispositivos, negociación y



definición de las sesiones mediante el protocolo *SDP* a partir de las capacidades de los terminales, arquitectura independiente de los servicios como de la red de acceso, entre otras) para brindar el soporte suficiente a las aplicaciones ejecutadas en el entorno de la *TDi* descrito.

- Durante la implementación de la arquitectura diseñada, la construcción del prototipo y su posterior despliegue sobre la arquitectura, se evidenció la dimensión e importancia que tiene *IMS* en el proyecto, ya que se convirtió en el corazón de la plataforma de control en el ámbito de la *TDi* ofreciendo principalmente el control y la gestión para las sesiones, para los mecanismos de registro y autenticación, para los datos de los perfiles de los usuarios y los datos relacionados con la aplicación de *TDi*, para los servicios desplegados y entre otros recursos puestos a disposición para que la aplicación que se implementó en el entorno de la *TDi*, incluya los mecanismos necesarios para cumplir con los requisitos planteados.
- El uso de la solución extremo a extremo que *Ericsson* ofrece, facilitó la implementación de los servicios de presencia y de mensajería, parte de la lógica que apoya a la aplicación *DVB-J* en el *servlet placa*, construcción de los clientes y además de otras implementaciones al servicio del proyecto. Por el lado negativo, ya que *SDS* está concebido con fines comerciales, limita en su versión *demo* a ciertas funcionalidades de la extensa variedad que esta solución ofrece, pero que en definitiva son las necesarias para este proyecto. Por otro parte, la curva de aprendizaje para adaptarse a esta herramienta es muy alta y además debe ceñirse respecto a los parámetros y patrones que esta define relacionado con la configuración de las entidades y la construcción de los servicios y aplicaciones como se observa en el anexo C.
- Debido a que *IMS* adopta en su arquitectura el uso de estándares abiertos y por otro lado, los servicios están definidos por entes que publican estándares como lo es *OMA*, hizo posible la implementación de las pasarelas y artefactos software necesarios para lograr la mediación entre los recursos que ofrece la arquitectura de red de telecomunicaciones *IMS* y la aplicación construida en el entorno de la *TDi*.
- La implementación de la aplicación en el entorno de la *TDi* se hizo mediante el modelo de aplicación basado en Java *DVB-J*, el cual ofreció el soporte necesario para las características y funciones implementadas en el prototipo debido a las clases y métodos del lenguaje Java definidas por el estándar *MHP*.
- La implementación de la plataforma de control en el ámbito de la Televisión Digital Interactiva muestra que esta arquitectura es el primer paso para permitir integrar las mejores características que puede ofrecer el entorno de la televisión con las mejores características que pueden ofrecer la arquitectura de red de telecomunicaciones *IMS*.
- Con respecto a los equipos de recepción de *TDi*, se tiene que consultar la versión del *middleware*, las capacidades y los recursos con los que disponen estos dispositivos para hacer el diseño tanto los servicios como de las aplicaciones, con el fin de aprovechar al máximo las características que los dispositivos ofrecen.



6.3 TRABAJOS FUTUROS.

- Dado el uso de *SDS* de *Ericsson* como la solución que implementa la arquitectura de red de telecomunicaciones para este proyecto, y teniendo en cuenta el valor de la licencia para obtener la versión completa de esta, se plantea una alternativa mediante la implementación de la plataforma de control en ámbito de la *TDi* usando una solución gratuita y de libre distribución de la arquitectura de red de telecomunicaciones *IMS* como lo puede ser *OpenIMS*.
- Implementar un mecanismo de seguridad para la plataforma propuesta haciendo énfasis principalmente en la adaptación realizada entre el *cliente STB* y la arquitectura de red de telecomunicaciones *IMS*.
- Proponer un conjunto de servicios y aplicaciones teniendo en cuenta el entorno social Colombiano y posteriormente hacer su despliegue sobre la plataforma diseñada para hacer un análisis y evaluación del impacto y la viabilidad que generan su inclusión en este ámbito.



BIBLIOGRAFÍA.

- [1]. **COMISIÓN NACIONAL DE TELEVISIÓN.** “República de Colombia: Televisión Digital Terrestre”. Internet: http://www.cntv.org.co/cntv_bop/tdt/documentos/tdt_colombia.pdf, Agosto 2008[Consulta: Junio 2009].
- [2]. **C. ARBOLEDA.** “Televisión de Calidad 2008,” presentado en la VII Muestra y Seminario Internacional: Televisión Digital Terrestre en Colombia, Bogotá, Colombia, 2008.
- [3]. **F. MARTÍNEZ, W. CASTILLO.** “Arquitectura Básica de un Navegador DVB-HTML para Múltiples Terminales”. Proyecto de fin de carrera, Universidad del Cauca, Colombia, 2009.
- [4]. **R. ERGUL. (Enero, 2007).** “DIGITAL BROADCASTING and INTERACTIVE TELEVISION IN DISTANCE EDUCATION: Digital And Interactive Television Infrastructure Proposal for Anadolu University Open Education Faculty”. [En línea]. Vol. 8, número 1, paginas 72. Disponible: http://tojde.anadolu.edu.tr/tojde25/pdf/article_5.pdf [Consulta: Junio 2009].
- [5]. **S. CHAVERT.** “Desarrollo de una plataforma de servicios administrativos del Comune de Grosseto para la televisión digital sobre una plataforma DVB-MHP”. Proyecto de fin de carrera, Universidad de Vigo, España, 2006.
- [6]. **A. GONZALES, K. JIMÉNEZ.** (2006). “La televisión digital interactiva y sus aplicaciones educativas”. [En línea]. Vol 1, paginas 93-101. Disponible: http://dialnet.unirioja.es/servlet/fichero_articulo?codigo=1985787&orden=0 [Consulta: Junio 2009].
- [7]. **M. SAGUES.** “Televisión y Multimedia. La Televisión Digital Interactiva”. Internet: <http://mhproject.org/media/blogs/mhpenlaces/Interno/Presentaciones/Television%20y%20Multimedia/Television%20y%20Multimedia%20TVDi.ppt>, 2008 [Consulta: Junio 2009].
- [8]. **M. SAMUDIO.** “Actualización de los estándares y mercado de Televisión Digital Terrestre TDT,” presentado en el Congreso Nacional y Andino de Telecomunicaciones ANDICOM, Cartagena de Indias, Colombia, Octubre 25 de 2007.
- [9]. **E. PÉREZ.** “Curso Multimedia Home Platform 1.1.2”. Internet: <http://www.code4tv.com/res/mhp112course/29-CODE4TV-MHP-NON-CA-SMARTCARD.pdf>, 2008 [Consulta: Julio 2009].
- [10]. **A. DANIELS.** “The Multimedia Home Platform: Creating a Multimedia World in the STB” Internet: <http://www.eetimes.com/electronics-news/4142985/The-Multimedia-Home-Platform-Creating-a-Multimedia-World-in-the-STB>, 2002. [Consulta: Julio 2010].
- [11]. **IMPALA MHEG EVERYWHERE.** “How MHEG-5 Works and Applications”. Internet: <http://www.impala.org/sites/default/files/What%20is%20MHEG.pdf>, 2008. [Consulta: Julio 2009].
- [12]. **E. SCHWALB.** *iTV Handbook: Technologies and Standards*. Editorial Pearson/Prentice Hall, 2004, 752 páginas.
- [13]. **S. MORRIS, A. SMITH.** “Interactive TV Standards”, Editorial Focal Press, 2005, páginas 585.
- [14]. **A. DA SILVA.** “Hacia la TV Digital Interactiva. Fundamentos técnicos. Internet: http://www.networkworld.es/Hacia-la-TV-Digital-Interactiva-3_Fundamentos-te/seccion-Telecomunicaciones/articulo-134360, 2002. [Consulta: Julio 2009].



- [15]. **E. TULANDE, D. ROJAS.** “Recomendaciones para la generación y distribución de contenidos educativos orientados a Televisión Digital Interactiva.”. Proyecto de fin de carrera, Universidad del Cauca, Colombia, 2009.
- [16]. **P. CESAR.** “A Graphics Software Architecture for High-End Interactive TV Terminals.”. Tesis doctoral, Universidad de Tecnología de Helsinki, Finlandia, 2005.
- [17]. **J. AMAYA, F. URBANO, W. CAMPO, J. ARCINIEGAS.** “Infraestructura Tecnológica para un laboratorio experimental de Televisión Digital Interactiva,” presentado en el Congreso Colombiano de Comunicaciones – COLCOM, Popayán, Colombia, Septiembre 2008.
- [18]. **THE MHP KNOWLEDGE PROJECT.** “The MHP-Guide”. Internet: <http://www.mhpkdb.org>, 2006. [Consulta: Julio 2009].
- [19]. **J. AMAYA.** “Definición del Laboratorio de iTV.”. Colombia. Número de documento EDiTV-RT-6-V1.0. Abril 16, 2008.
- [20]. **C. PENG.** “Digital Television Applications.” Tesis doctoral, Universidad de Tecnología de Helsinki, Finlandia, 2002.
- [21]. **S. OVIEDO, G. JURADO.** “Redes Inteligentes”. Internet: <http://afrodita.unicauca.edu.co/~solivero/redesint/intro.htm>, 2003 [Consulta: Septiembre 2009].
- [22]. **UNIVERSIDAD AUTÓNOMA TAMAULIPAS.** “SS7”. Internet: colaboracion.uat.edu.mx/dt/cportes/telefonía/Lists/Attachments/73/SS7.doc, 2005. [Consulta: Septiembre 2009].
- [23]. **J. ANDRINAL, J. MARTÍNEZ, A. GARCÍA.** “Arquitecturas Orientadas a Servicios en Redes de Nueva Generación”. Internet: www.collector.org.ar/papers/19.pdf, 2000. [Consulta: Septiembre 2009].
- [24]. **H. FROLOSHKI, E. PENCHEVA.** (2007). “Enabling Architecture for Third Party Applications in Intelligent Network”. [En línea]. Vol. 7, número 1, páginas 42. Disponible en: http://www.cit.iit.bas.bg/CIT_07/v7-1/33-44.pdf [Consulta: Octubre 2009].
- [25]. **R. CAÑIZARES.** (2007). “Arquitectura Orientada a Servicios (SOA)”. Revista Auditoría y Seguridad. Volumen 56, 2007.
- [26]. **INTERNATIONAL ENGINEERING CONSORTIUM.** “Intelligent Network (IN)”. Internet: <http://www.iec.org/online/tutorials/in/topic03.asp>, 2007. [Consulta: Septiembre 2009].
- [27]. **T. MAGEDANZ.** “Intelligent Network Evolution - Impact of Internet, CORBA, TINA and Mobile Agent technologies”. Internet: http://www.tinac.com/conference/tina99/tutorial_p.pdf, 1999. [Consulta: Octubre 2009].
- [28]. **J. HUIDROBO.** “La Red Inteligente”. Internet: <http://www.coit.es/publicac/publbit/bit111/quees.htm>, 2000. [Consulta: Septiembre 2009].
- [29]. **L. CASTILLO BARROS.** “Red Inteligente Avanzada”. Internet: [https://www.u-cursos.cl/ingeniería/2008/1/EL629/1/material_docente/objeto/177297](http://www.u-cursos.cl/ingeniería/2008/1/EL629/1/material_docente/objeto/177297), 2008. [Consulta: Octubre 2009].
- [30]. **J. ACEVEDO, O RUIZ MATEUS.** “Análisis de los Servicios de Telecomunicaciones basados en interfaces PARLAY OSA”. Proyecto para optar al título de especialista, Universidad Industrial de Santander, Colombia, 2005.



- [31]. **J. RÁBANOS**. “Telecomunicaciones móviles”. Editorial Marcombo. Segunda Edición, 1998, 276 páginas.
- [32]. **M. ESPAÑA**. “Servicios Avanzados de Telecomunicación”. Editorial Díaz de Santos S.A. Segunda Edición, 2003. 816 páginas.
- [33]. **M. LAVADO, A. MARTÍ**. “Evolución de la plataforma de servicios CAMEL mas allá de los sistemas 3G.”. *IEEE Latin America Transactions*. Vol. 4, numero. 5, septiembre 2006
- [34]. **J. ZUIDWEG**. “Next Generation Intelligent Networks.”. Editorial Artech House INC. Segunda Edición, 2002. 342 páginas.
- [35]. **J. ANDERSON**. “Intelligent Networks. Principles and Applications”. Editorial MPG Books Ltd. Segunda Edición. 2002. 241 páginas.
- [36]. **J. VAN BOSSE, F DEVETAK**. “Signaling in Telecommunication Networks.”. Editorial Jhon Wiley & Sons. Inc., Segunda Edición. 2007. 829 paginas.
- [37]. **A. GÓMEZ, C. RAMOS**. “Redes y Servicios I: Red Inteligente”. Internet: http://asignaturas.diatel.upm.es/rrss1/documentacion_archivos/TEORIA%20ACTUAL/Tema2RedInteligenteCurso09-10.pdf, 2002. [Consulta: Octubre 2009].
- [38]. **L. CASTILLO**. “Red Inteligente Avanzada”. Internet: https://www.u-cursos.cl/ingenieria/2009/1/EL629/1/material_docente/objeto/234048, 2008. [Consulta: Octubre 2009].
- [39]. **R. NOLDUS**. “CAMEL INTELLIGENT NETWORKS FOR THE GSM, GPRS AND UMTS NETWORK”. Editorial John Wiley & Sons Ltd, 2006. 428 páginas.
- [40]. **J. SERRANO**. “OSA/ PARLAY”. Internet: <http://nmg.upc.es/~mserrano/pdf%20files/%5BCNMT%5DOSA-Parlay%20Spanish.pdf>, 2006. [Consulta: Octubre 2009].
- [41]. **S. H. RAI**. “GPRS and 3G Wireless Networks”. Internet: <http://www.it.iitb.ac.in/~satyajit/seminar/node31.html>, 2002. [Consulta: Octubre 2009].
- [42]. **O. CAICEDO, J. HURTADO, A. SERNA, S. SOLARTE**. “Servicios basados en la localización bajo la plataforma OSA / PARLAY.”, presentado en el III Seminario Internacional Tecnologías en Internet. SITI 2004, 14 páginas.
- [43]. **R. HERNÁNDEZ, E. QUINTERO**. “Estudio Para el Desarrollo de Servicios de Telecomunicaciones de Nueva Generación en Colombia”. Proyecto de fin de carrera, Universidad Industrial de Santander, Colombia, 2009.
- [44]. **U. HERZOG**. “OSA & Parlay. Enabling an open services market.” Internet: <http://www.eurescom.eu/message/messageJun2002/tutorial.asp>, 2002. [Consulta: Octubre 2009].
- [45]. **H. ORTEGA, H. ESPÍNDOLA, W. CASTELLANOS**. “Aplicaciones y Servicios para las Redes de Próxima Generación en un Ambiente de Procesamiento Distribuido”, presnetado en el Congreso Universidad Nacional de aplicaciones NGN, Bogotá, Colombia.



- [46]. **M. BARTOLOMÉ, D. GALLEGO, J. LÓPEZ, J. GÓMEZ.** “UMTS: hacia una red todo IP”. Internet: http://195.235.92.45/documentos/revista_comunicaciones_i%2Bd/numero24.pdf, 2002. [Consulta: Octubre 2009].
- [47]. **R. CHEN, E. SU, V. SHEN, Y. WANG.** “Introduction to IP Multimedia Subsystem (IMS), Part 1: SOA Parlay X Web services”. Internet: <http://www.ibm.com/developerworks/webservices/library/ws-soa-ipmultisub1/>, 2006. [Consulta: Octubre 2009].
- [48]. **S. ZNATY, J. DAUPHIN, R. GELDWERTH.** “IP Multimedia Subsystem: Principios y Arquitectura”. Internet: http://www.efort.com/media_pdf/IMS_ESP.pdf, 2005. [Consulta: Octubre 2009].
- [49]. **R. MILLÁN.** “IP Multimedia Subsystem. Convergencia total en IMS”. Internet: <http://www.ramonmillan.com/tutorialeshtml/ims.htm>, 2006. [Consulta: Octubre 2009].
- [50]. **R. SEGARRA.** “Estudio y Diseño de un Backbone IP Multimedia Subsystem (IMS) para Proporcionar Servicios de Datos Multimedia en la Zona Norte de Quito.” Proyecto de fin de carrera, Escuela Politecnica de Ecuador, Ecuador, 2009.
- [51]. **A. ESPINOZA.** (2008). HTML: “Soporte de Comunicaciones Para Grupos de Usuarios Basado en IMS. Cliente y Pruebas del Servidor”. Proyecto de fin de carrera, Universidad de Sevilla, España, 2008.
- [52]. **J. HARRIS.** “Una Introducción a las herramientas de autoría.” [En línea]. Internet: <http://autoriadecontenidosudl.googlepages.com/herramientasautoria.pdf>, 2008. [Consulta: Octubre 2009].
- [53]. **ALTICAST.** “Alticast Unveils the Developer Edition of AltiComposer”. Internet: http://www.alticast.tv/pressroom/pressroom_release_read.html?years=2000&num=25, 2000. [Consulta: Octubre 2009].
- [54]. **H. PANTLE.** “Kulturtechnische Aspekte interaktiven Fernsehens und MHP-Authoring Tools. Retrieved”. Tesis de Maestria, Humboldt Universitat zu Berlin, Alemania, 2005.
- [55]. **ICAREUS TECHNOLOGIES.** “ICAREUS ITV SUITE AUTHOR DATA SHEET”. Internet: http://extranet.icareus.com/datasheets/iTV_Suite_Author_Datasheet.pdf, 2007. [Consulta: Octubre 2009].
- [56]. **ORTIKON INTERACTIVE.** “Ortikon Interactive”. Internet: <http://www.digitaltvindustry.com/ortikon-interactive-716.html>, 2007. [Consulta: Octubre 2009].
- [57]. **ARANNOVA.** “MHPGen. Plataforma de generación de contenidos MHP”. Internet: <http://www.mhpgen.com/>, 2007. [Consulta: Octubre 2009].
- [58]. **AVALPA.** “OpenCaster 2.2: the free digital tv software”. Internet: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>, 2009. [Consulta: Octubre 2009].
- [59]. **L. PALLARA, A. VENTURI.** “Opencaster User Manual”. Internet: <http://www.avalpa.com>, 2009. [Consulta: Enero 2010].
- [60]. **FRAUNHOFER FOKUS NGN.** “Open IMS Core”. Internet: <http://www.openimscore.org/>, 2009. [Consulta: Abril 2010].



[61]. **ERICSSON**. “Ericsson Service Development Studio (SDS) 4.1 Technical.”. Internet:
http://www.ericsson.com/developer/sub/open/technologies/ims_poc/docs/tech_desc_sds_40,
2009 [Consulta: Enero 2010].

[62]. **M. POIKSELKA, G. MAYER**. “IMS. IP Multimedia Concepts and Service”. Editorial Artech John
Wiley and Sons, Ltd. Tercera Edición, 2009. 520 páginas.