



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, REDES Y
COMUNICACIÓN DE DATOS**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS**

**TEMA: ANÁLISIS Y DESARROLLO DE UNA APLICACIÓN
INTERACTIVA PARA TELEVISIÓN DIGITAL MEDIANTE EL
USO DEL API JAVA TV**

AUTOR: DAVID ALEJANDRO JARRÍN RODRÍGUEZ

DIRECTOR: ING. CARLOS GABRIEL ROMERO GALLARDO

**CODIRECTOR: DR. GONZALO FERNANDO OLMEDO
CIFUENTES**

SANGOLQUÍ, MAYO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS

CERTIFICADO

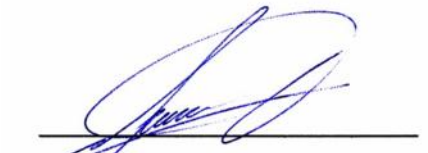
Ing. Carlos Gabriel Romero Gallardo y Dr. Gonzalo Fernando Olmedo Cifuentes

CERTIFICAN

Que el trabajo titulado “ **ANÁLISIS Y DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA TELEVISIÓN DIGITAL MEDIANTE EL USO DEL API JAVA TV** ” realizado por el señor David Alejandro Jarrín Rodríguez ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas – ESPE.


El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf).

Sangolquí, 28 de abril de 2015



Ing. Carlos Gabriel Romero Gallardo

DIRECTOR



Dr. Gonzalo Fernando Olmedo Cifuentes

CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS

DECLARACIÓN DE RESPONSABILIDAD

DAVID ALEJANDRO JARRÍN RODRÍGUEZ

DECLARO QUE:

El proyecto de grado denominado “ **ANÁLISIS Y DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA TELEVISIÓN DIGITAL MEDIANTE EL USO DEL API JAVA TV** ”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan en el documento, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 28 de abril de 2015



David Alejandro Jarrín Rodríguez

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS

AUTORIZACIÓN

DAVID ALEJANDRO JARRÍN RODRÍGUEZ

Autorizo a la Universidad de las Fuerzas Armadas – ESPE, la publicación en la biblioteca virtual de la Institución del trabajo “ **ANÁLISIS Y DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA TELEVISIÓN DIGITAL MEDIANTE EL USO DEL API JAVA TV** ”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y auditoria.

Sangolquí, 28 de abril de 2015



David Alejandro Jarrín Rodríguez

DEDICATORIA

Este proyecto de tesis va dedicado con mucho amor a toda mi familia, especialmente a mis padres porque con éxito han sabido guiarme y apoyarme en el transcurso de mi formación profesional y de mi vida en general; a mis amigos y profesores que directa o indirectamente me apoyaron para lograr este objetivo.

David Alejandro Jarrín Rodríguez

AGRADECIMIENTO

Agradezco en primer lugar a mis padres, por todo el amor, comprensión, motivación y ejemplo que me han brindado a lo largo de mi carrera, ya que si ellos no hubieran estado a mi lado talvez no hubiera podido llegar hasta donde hoy me encuentro.

A mi hermana por su completo apoyo y compañía.

A mi familia en general por todo el cariño y apoyo que me supieron demostrar a lo largo de mi carrera universitaria.

A todos mis amigos de "TM3" que compartieron conmigo este largo y difícil camino, y que de una u otra manera siempre estuvieron ahí para apoyarme.

A todos los mis profesores por las enseñanzas y la amistad brindada, y en especial al Ing. Carlos Romero y al Dr. Gonzalo Olmedo por la ayuda y orientación recibidos en la realización de este proyecto de tesis.

David Alejandro Jarrín Rodríguez

ÍNDICE DE CONTENIDO

DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN	xi
ABSTRACT	xii
CAPÍTULO I	1
PRESENTACIÓN DEL PROYECTO	1
1.1. ANTECEDENTES	1
1.2. JUSTIFICACIÓN E IMPORTANCIA	2
1.3. ALCANCE DEL PROYECTO	3
1.4. OBJETIVOS.....	3
1.4.1. General.....	3
1.4.2. Específicos	4
1.5. METODOLOGÍA	4
1.6. ORGANIZACIÓN DEL LIBRO.....	6
CAPÍTULO II	7
MARCO TEÓRICO	7
2.1. TELEVISIÓN DIGITAL	7
2.2. INTERACTIVIDAD EN LA TV DIGITAL.....	7
2.3. NIVELES DE INTERACTIVIDAD	8
2.4. ESTÁNDARES DE TRANSMISIÓN DE TELEVISIÓN DIGITAL.....	9
2.4.1. ATSC STANDARDS(ADVANCED TELEVISION SYSTEM COMMITTEE)	9
2.4.2. DVB - T (DIGITAL BROADCASTING TERRESTRIAL)	10
2.4.3. ISDB (IntegratedServices Digital Broadcasting).....	10
2.5. ESTÁNDARES MIDDLEWARE PARA TELEVISIÓN DIGITAL	
INTERACTIVA	11
2.5.1. DTV Applications Software Enviroment (DASE)	12

2.5.2.	ACAP (Advanced Common Application Platform for Interactive Television)	
	13	
2.5.3.	ARIB (Association of Radio Industries Business)	14
2.5.4.	GINGA.....	15
2.5.4.1.	GINGA J	17
2.5.5.	MHP (MULTIMEDIA HOME PLATAFORM)	19
2.5.5.1.	VERSIONES MHP	21
2.5.5.2.	DVB - GEM (Global Executable MHP)	23
2.5.5.3.	ARQUITECTURA MHP	24
2.5.5.4.	APLICACIONES MHP	26
2.6.	JAVA TV	27
2.6.1.	ENTORNO DEL API	28
2.6.2.	XLETS	29
CAPÍTULO III.....		33
DESARROLLO DEL PROTOTIPO DE APLICACIÓN		33
3.1.	ENTORNOS DE DESARROLLO INTEGRADO PARA DESARROLLO DE APLICACIONES CON JAVA TV	33
3.1.1.	DEFINICIÓN.....	33
3.1.2.	IDE's PARA EL DESARROLLO DE APLICACIONES PARA TV DIGITAL.....	33
3.1.3.	EVALUACIÓN DE UN IDE	35
3.2.	ANÁLISIS Y DISEÑO DEL PROTOTIPO	42
3.2.1.	ESPECIFICACIÓN DE REQUISITOS.....	42
3.2.2.	DESARROLLO DE PROTOTIPO.....	43
3.2.2.1.	SOFTWARE GENERICO	43
3.2.2.1.1.	NETBEANS	43
3.2.2.1.2.	XLETVIEW	44
3.2.3.	APIs NECESARIAS PARA EL DESARROLLO DEL PROTOTIPO	45
3.2.3.1.	APIs NUCLEO:	45

3.2.3.2.	APIs de JAVA MEDIA FRAMEWORK (JMF).....	47
3.2.3.3.	APIs JAVA TV	48
3.2.3.4.	APIs DE DAVIC (Digital Audio Video Council).....	49
3.2.3.5.	APIs DE HAVI (HOME AUDIO VIDEO INTEROPERABILITY).....	51
3.2.3.6.	APIs DE DVB	51
3.2.4.	HERRAMIENTAS DE SOFTWARE NECESARIAS.....	53
3.2.5.	EJEMPLO APLICACIÓN BASICO DE XLET CON NETBEANS Y XLETVIEW	54
3.2.6.	DISTRIBUCIÓN DEL PROYECTO	64
3.2.7.	PROTOTIPO DE APLICACIÓN TURISMO	67
3.2.8.	IMPLEMENTACIÓN JAVA TV	73
CAPÍTULO 4.....		75
CONCLUSIONES Y RECOMENDACIONES.....		75
4.1.	CONCLUSIONES.....	75
4.2.	RECOMENDACIONES	76

ÍNDICE DE FIGURAS

Figura 1. Estructura de un terminal de acceso	11
Figura 2. Arquitectura DASE	13
Figura 3. Arquitectura ACAP.....	14
Figura 4. Arquitectura ARIB.....	15
Figura 5. Arquitectura Ginga	17
Figura 6. Arquitectura General de GINGA J	18
Figura 7. APIs GINGA J.....	19
Figura 8. Arquitectura MHP	25
Figura 9. Estados de Ciclo de Vida Xlet.....	30
Figura 10. Interfaz XletView.....	45
Figura 11. Crear Nuevo Proyecto.....	54
Figura 12. Selección de Categoría y Tipo de Proyecto	55
Figura 13. Ingreso del nombre del proyecto.....	56
Figura 14. Creación Nueva Java Class.....	57
Figura 15. Nombre de nueva Java Class.....	58
Figura 16. Agregar Librerías.....	58
Figura 17. Selección de Librerías XletView y Java TV	59
Figura 18. Métodos del Xlet	59
Figura 19. Interfaz Xletview	62
Figura 20. Administrador de aplicaciones XletView	63
Figura 21. Selección del Xlet	63
Figura 22. Xlet Simulado.....	64
Figura 23. Clases del Proyecto.....	65
Figura 24. Carpetas Fotografías.....	66
Figura 25. Librerías	66
Figura 26. Pantalla Inicial de la Aplicación.....	67
Figura 27. Menú Principal de la Aplicación.....	68
Figura 28. Sub Menú Las 7 Maravillas de Quito	69
Figura 29. Información Sitio Turístico	69
Figura 30. Sub Menú Gastronomía Quiteña	70
Figura 31. Información Restaurante Café del Teatro	71
Figura 32. Sub Menú Juego de Trivia.....	72
Figura 33. Pantalla de Puntuación	73

ÍNDICE DE TABLAS

Tabla 1. Modelo de Calidad ISO 9126	37
Tabla 2. Análisis de Características del Modelo de Calidad ISO 9126	40
Tabla 3. Atributos adicionales	41
Tabla 4. APIs de Nucleo	46
Tabla 5. APIs de Java TV.....	48
Tabla 6. APIs de DAVIC	50
Tabla 7. APIs de DVB.....	51

RESUMEN

La televisión ha venido evolucionando a través de los años hasta llegar a lo que hoy en día se conoce como la Televisión Digital Interactiva. Ecuador en el 2010 decidió acoger el estándar brasileño – japonés para televisión digital terrestre ISDB-Tb, lo que trae consigo un cambio importante en la forma de ver televisión, haciéndola más atractiva y generando un sinnúmero de posibilidades de interacción; por esto es necesario estudiar esta nueva tecnología, en especial el desarrollo de servicios y aplicaciones interactivas para TDT. El presente proyecto propone la investigación y creación de una aplicación interactiva utilizando el API de JAVA TV. Para llevar a cabo el desarrollo de la aplicación se realizó un análisis de la arquitectura y funcionamiento del API, conociendo a fondo el modelo de aplicación llamada Xlet, su ciclo de vida, y la utilización de los métodos de la misma. Para llevar a cabo las pruebas de funcionamiento de la aplicación se utilizó el emulador XletView.

Palabras Claves:

- **TDT (TELEVISIÓN DIGITAL TERRESTRE)**
- **GINGA**
- **MHP (MULTIMEDIA HOME PLATAFORM)**
- **JAVA TV**
- **XLET**
- **XLETVIEW**

ABSTRACT

The television has evolved through the years to become what today is known as the Interactive Digital Television. Ecuador in 2010 decided to host the Brazilian Japanese standard for terrestrial digital TV ISDB-Tb which brings about a major change in the way people watch TV, making it more attractive and creating endless possibilities of interaction; therefore it is necessary to study this new technology, especially the development of services and interactive applications for DTT. This project proposes the research and creation of an interactive application using the API JAVA TV. To perform the application development an analysis of the architecture and function of API was made, knowing the call Xlet application model, life cycle, and the use of methods Xlet. And finally by the emulator XletView testing of application operation.

KEYWORDS:

- **DTT (DIGITAL TERRESTRIAL TELEVISION)**
- **GINGA**
- **MHP (MULTIMEDIA HOME PLATAFORM)**
- **JAVA TV**
- **XLET**
- **XLETVIEW**

CAPÍTULO I

PRESENTACIÓN DEL PROYECTO

1.1. ANTECEDENTES

La televisión en nuestros días se ha convertido en una fuente importante de información y entretenimiento de primera mano, que ha visto grandes cambios en los últimos años; el Ecuador es uno de los países que está atravesando este cambio tecnológico importante: la aplicación de la tecnología digital a la señal de televisión analógica convencional lo que se conoce como Televisión Digital Terrestre “TDT”; este proceso de cambio permitirá entre cosas la implementación de nuevos servicios audiovisuales e interactivos, la optimización del uso del espectro radioeléctrico, y aportando una mayor calidad de imagen y sonido.

Nuestro país incurrió en este importante avance en el 2010, cuando adoptó el estándar japonés-brasileño conocido como ISDB-Tb, que brinda a los usuarios calidad, movilidad, interactividad, portabilidad y multi-servicios, mediante la utilización del middleware Ginga como plataforma de desarrollo y presentación de los contenidos interactivos. [1]

En la actualidad se han creado distintos tipos de middleware para TDT, entre los que destacan el Estado Unidense (ACAP Advanced Common Application Plataform), el europeo (MHP Multimedia Home Plataform), el japonés (ARIB Association of Radio Industries Business) y el brasileño (ISDB-Tb); aunque son middlewares diferentes poseen características específicas guiadas por la norma GEM (UIT-T J.201) (UIT-T, 2001), logrando con esto que la implementaciones existentes se adapten y se garantice la ejecución global de las aplicaciones.

Java Tv aparece como una extensión de la plataforma Java que tiene como principal función apoyar la producción de contenidos interactivos de forma procedural para la Televisión Digital; proporcionando un conjunto de métodos, clases e interfaces para facilitar la creación de aplicaciones diseñadas para ejecutarse en distintas plataformas para la recepción de televisión digital independiente de las tecnologías utilizadas en la red de transmisión [2].

1.2. JUSTIFICACIÓN E IMPORTANCIA

La televisión desde hace muchos años atrás se ha convertido en uno de los elementos con mayor importación en nuestra sociedad, en nuestro país posee el mayor índice de penetración en la población ecuatoriana con un 90, convirtiéndolo en el medio de acceso a información predominante a la población en general; con la incursión de la Televisión Digital este medio de comunicación se ha visto fortalecido haciéndolo cada vez más atractivo y generando un sinnúmero de posibilidades de interacción [3].

La incorporación de la interactividad ha permitido llevar hasta el televisor una gran parte de los servicios que podemos encontrar en Internet, pero de forma más accesible y directa para el espectador. Logrando de esta manera constituir un canal de acceso alternativo a los servicios on-line a los colectivos que poseen una gran brecha digital [4].

Nuestro país al encontrarse en el proceso de cambio a TDT ha orientado la investigación al desarrollo de aplicaciones interactivas utilizando como principal herramienta Ginga NCL y hace poco la utilización de Ginga J, pero no ha enfocado el estudio en otras herramientas de desarrollo de aplicaciones, una de ellas Java TV que a diferencia de las plataformas mencionadas esta no está ligada a ningún conjunto de normas para televisión digital, lo que le permite funcionar sobre cualquier plataforma que la sustente, así como sobre cualquier sistema de difusión; Java TV proporciona un conjunto de elementos comunes que se requieren en cualquier plataforma de televisión

digital, tal como el modelo del núcleo de la aplicación, el ciclo de vida, el acceso a los servicios de difusión y al servicio de la información. Todas las características antes mencionadas generan una gran ventaja en la portabilidad de las aplicaciones desarrolladas con Java TV pues permiten a los proveedores de contenido y desarrolladores de aplicaciones oportunidades de venta o difusión de sus productos en diferentes mercados.

1.3. ALCANCE DEL PROYECTO

Para que este proyecto solviente la necesidad de conocimiento acerca del funcionamiento del API Java TV, se requiere incurrir en los siguientes puntos:

- Conocer acerca del API Java TV, sus principales funciones, componentes, que nos permitan el desarrollo de aplicaciones interactivas.
- Desarrollo de una aplicación interactiva utilizando Java TV.
- Realizar simulaciones de la aplicación de manera local mediante el emulador XetView.

1.4. OBJETIVOS

1.4.1. General

- Investigar y desarrollar una aplicación interactiva mediante la utilización del API Java TV utilizando la metodología XP.

1.4.2. Específicos

- Investigar la arquitectura y funcionamiento del API Java TV.
- Estudiar el funcionamiento de los componentes para el desarrollo de una aplicación tipo Xlet.
- Implementar el escenario de simulación y realizar las pruebas necesarias de la aplicación desarrollada.

1.5. METODOLOGÍA

Para el desarrollo del presente proyecto se escogió la utilización de la metodología XP (Programación Externa), desarrollada por Kent Beck en 1999, es una metodología liviana que nos brinda un conjunto de prácticas y reglas empleadas para el desarrollo de software, basada en ideas sobre cómo enfrentar ambientes muy cambiantes [5].

Este modelo se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación [5].

Esta metodología persigue básicamente dos objetivos, el primero es garantizar la calidad del software desarrollado y segundo realizarlo en el menor tiempo posible, consiguiendo con esto mejorar la productividad del proyecto y la satisfacción del cliente.

Para lograr estos objetivos la metodología utiliza los siguientes valores fundamentales [6]:

- Comunicación

Es muy importante que haya una continua comunicación con el cliente y dentro de todo el equipo de trabajo, con esto conseguiremos que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se cubran los requerimientos del cliente.

➤ Simplicidad

El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible.

➤ Retroalimentación

Es la comunicación constante entre el desarrollador y el usuario.

La metodología cruza por 4 fases:

- Planificación
- Diseño
- Codificación
- Pruebas

Entre las ventajas que podemos destacar de esta metodología [7]:

- El resultado de los productos se los puede obtener con mayor rapidez.
- Existe un continuo proceso de integración, por lo que se consigue una mayor facilidad al integrar el trabajo en su totalidad.
- Se cubren con mayor exactitud las necesidades del usuario, ofreciéndole continuas versiones del producto.
- Se obtienen productos fiables, robustos y con un código simple y fácil de entender, reduciendo con esto el número de errores y a la vez permitir que el código sea fácilmente modificado de acuerdo a los requerimientos del usuario.

Como desventajas se puede destacar:

Se debe utilizar para proyectos que van a ser realizados en corto plazo.

1.6. ORGANIZACIÓN DEL LIBRO

El libro está organizado de la siguiente manera:

En el capítulo 2 se describe los conceptos generales que se tuvieron en cuenta para el desarrollo de ese proyecto, se contempla en forma general los estándares de transmisión de Televisión Digital Terrestre y de middleware existentes en el mundo. Abordaremos el significado de Televisión Digital Interactiva, los beneficios que este trae consigo, sus niveles de interactividad y el tipo de aplicaciones que se pueden desarrollar. Se explicará que es MHP, en que consiste el API de Java TV y los Xlets.

El capítulo 3 contiene el análisis y Diseño del prototipo que contiene los requisitos de la aplicación, diagramas de secuencia y de clases; contiene además el software que se utilizó para el desarrollo del prototipo, como las diferentes APIs para la creación de la misma.

El capítulo 4 se muestra las pruebas de la aplicación que se realizaron en el emulador.

Finalmente se da a conocer las conclusiones y recomendaciones que se tuvieron al desarrollar este proyecto.

CAPÍTULO II

MARCO TEÓRICO

2.1. TELEVISIÓN DIGITAL

La televisión digital es una técnica de difusión de televisión que sustituye a la televisión analógica, y basa su funcionamiento en la transmisión de imágenes en movimiento y su sonido asociado mediante una señal digital a través de una red de repetidores; y que puede ser transmitida a través de cable, terrestre y satélite para posteriormente ser decodificado por el set-top-box, el cual es un dispositivo externo que permite a un televisor convencional reproducir las señales televisivas producidas por la señal digital, estos dispositivos poseen un middleware que puede soportar características de interactividad con los usuarios finales [8].

2.2. INTERACTIVIDAD EN LA TV DIGITAL

Se define a la interactividad en la TV Digital, como la capacidad de ofrecer servicios adicionales a la programación actual de la televisión; lo que permite al usuario final disponer de información asociada al contenido audiovisual, la programación de los canales, participación en concursos, votaciones, transacciones comerciales como la compra de productos y servicios, así como la participación en los propios programas.

A través de la interactividad los canales de televisión están en la capacidad de ofrecer un conjunto de servicios a la ciudadanía, incorporando nuevas funciones de comunicación y participación; y brindando apoyo en el desarrollo de la Sociedad de la Información mediante la transmisión de Servicios Sociales.

La interactividad en la televisión tiene como principal ventaja el complementar los contenidos emitidos en Televisión, mediante un amplio conjunto de servicios de públicos, comercial y de entretenimiento, que hasta el momento solo se podían acceder mediante computadores y teléfonos móviles; estos servicios pueden ser fácilmente adaptados a las necesidades de los colectivos en nuestra sociedad [9].

La televisión digital interactiva funciona a partir de la difusión de la televisión analógica, junto con los datos y servicios interactivos mediante el uso de canal de retorno dándole al usuario la posibilidad de enviar datos o hacer solicitudes al canal difusor, lo que permite al usuario interactuar de forma directa con los programas [7].

La televisión digital fue desarrollada en 1991 por el consorcio DVB-EBU (Digital Video Broadcasting European Broadcasting Union) por cadenas de televisión, programadores, fabricantes de aparatos electrónicos y diferentes entidades públicas. Entre sus objetivos principales está el desarrollo de la tecnología digital definiendo diferentes estándares que sean ampliamente aceptados como normas para la utilización de dicha tecnología [8].

2.3. NIVELES DE INTERACTIVIDAD

Los niveles de Interactividad pueden ser definidos de acuerdo a la finalidad que se tenga para el contenido desarrollado, manteniendo una estrecha relación con el tipo y calidad de los recursos tecnológicos a utilizar, así como de la programación informática usada en la implementación [10].

Considerando esto existen dos niveles de interactividad en las aplicaciones de Televisión Digital que se pueden definir como:

- Sin interacción (Solo TV digital).
- Con Interacción, esta posibilita que el usuario se convierta en un miembro activo capaz de interactuar con las aplicaciones y servicios personalizados de Televisión Digital, aquí se pueden ofrecer 2 tipos de interactividad:
- Interacción Local.- en este tipo de interactividad el espectador interactúa con la información almacenada en el receptor, es decir, puede acceder a contenidos interactivos pero no puede enviar datos de vuelta; la información almacenada en el receptor se renueva con cierta periodicidad.
- Interacción Con Upload.- esta permite el envío de datos vía canal de retorno
- Interacción avanzada.- envío y recepción de datos vía canal de retorno.

Estas dos últimas permiten que el espectador interactúe con el proveedor de servicios mediante el mencionado canal de retorno, esta interactividad permite al usuario acceder a contenido adicional a la programación.

2.4. ESTÁNDARES DE TRANSMISIÓN DE TELEVISIÓN DIGITAL

2.4.1. ATSC STANDARDS(ADVANCED TELEVISION SYSTEM COMMITTEE)

Conjunto de estándares desarrollados por ADVANCED TELEVISION SYSTEM COMMITTEE, para la transmisión de Televisión Digital sobre Terrestre, cable y satélite, principalmente en los Estados Unidos y en base a estos estándares se desarrolló en Canadá, México, Corea del Sur y recientemente en Guatemala, Honduras y Costa Rica; estos estándares están intentado reemplazar en los Estados Unidos a NTSC que es el sistema de codificación y transmisión de televisión analógica desarrollada en Estados Unidos y que se emplea en la actualidad en la mayor parte de América, Japón y otros países [11].

2.4.2. DVB - T (DIGITAL BROADCASTING TERRESTRIAL)

Es un estándar de transmisión de televisión terrestre creado por la organización Europea DVB; este estándar especifica la estructura de imagen, codificación de canal y modulación del sistema de radiodifusión de Televisión Digital; es un sistema flexible que permite entregar un amplio rango de servicios desde Televisión de alta definición (HDTV) hasta múltiples canales de definición estándar, recepción fija, móvil y portátil, a través de las diversas tecnologías de distribución de señal como Satélite (DVB-S), cable (DVB-C), dispositivos móviles (DVB-H) y nuevos estándares como IP TV (DVB-IPTV).

La gran diferencia que distingue a este estándar de los anteriores es que este estándar no requiere especificaciones propias de middleware independiente para cada medio de transmisión [7].

2.4.3. ISDB (Integrated Services Digital Broadcasting)

Es un estándar para transmisión de video digital usado en casi todos los países de Latinoamérica y en Japón. Este formato es usado para transmitir Televisión Digital en forma gratuita en dichos países.

ISDB está conformado por una familia de componentes como la televisión digital terrestre (ISDB-T e ISDB-Tb), la televisión satelital (ISDB-S), la televisión por cable (ISDB-C), servicios multimedia (ISDB-Tmm) y radio digital (ISDB-Tsb). Brasil desarrollo SBTVD (Sistema Brasileiro de Televisión Digital) basado en el estándar Japonés; estos dos estándares poseen su principal diferencia en la compresión de video que cada una utiliza.

Una vez definido el estándar brasileño en enero del 2000 casi todos los países de Sudamérica adoptaron el mismo; cabe destacar que el ISDB Japonés no es compatible con ISDB-T, es decir que un reproductor de Japón no funcionara en la zona ISDB-T.

2.5. ESTÁNDARES MIDDLEWARE PARA TELEVISIÓN DIGITAL INTERACTIVA

Como mencionamos en secciones anteriores para la ejecución de aplicaciones interactivas es necesario el uso de un Set-Top-Box que permita que los usuarios puedan controlar y manejar dichas aplicaciones gracias a la capa de software implementado en el terminal, donde se ejecutan las aplicaciones, se dibujan en la pantalla y se gestionan los eventos y se supervisan las etapas del ciclo de vida de la aplicación.

Por lo tanto el software mencionado es un middleware, una capa intermedia o API genérico, que admite el acceso a las aplicaciones y servicios interactivos independientemente de la plataforma de hardware o software donde se estén ejecutando [10].



Figura 1. Estructura de un terminal de acceso

2.5.1. DTV Applications Software Enviroment (DASE)

DASE es la norma de middleware americano desarrollada por la ATSC, fue desarrollada inicialmente como DASE nivel 1, cuya interactividad era solamente local ya que no se contempló nunca la necesidad de un canal de retorno; actualmente existe la versión DASE nivel 3 que no solamente contempla la interactividad, sino que tiene por objeto la integración de la Televisión con el Internet, brindando opciones de correo electrónico y navegadores web.

Las aplicaciones desarrolladas en este middleware, pueden ser de entrono declarativo y de procedimiento, las declarativas son representadas por documentos hipermedia y multimedia escritos en leguaje de marcado como XHTML, entre otros. Mientras que las aplicaciones de procedimiento son programas que utilizan netamente tecnología Java. La máquina virtual de Java recibió extensiones que permiten un desenvolvimiento más adecuado de estas aplicaciones.

Este sistema está formado por módulos DASE (Ambiente de Aplicación Declarativo) o PAE (Ambiente de Aplicación de Procedimiento). Un DAE es un subsistema lógico que procesa lenguaje de marcado, scripts y hojas de estilo, cuyo principal componente el DCDE realiza un análisis sintáctico del documento; mientras que un PAE procesa el contenido de los objetos activos, cuyo principal componente PCDE, contiene la máquina virtual de Java.

Cabe mencionar que el DASE no es exclusivamente declarativo o de Procedimiento, sino que una aplicación de procedimiento puede hacer referencia a contenido declarativo y viceversa [10].

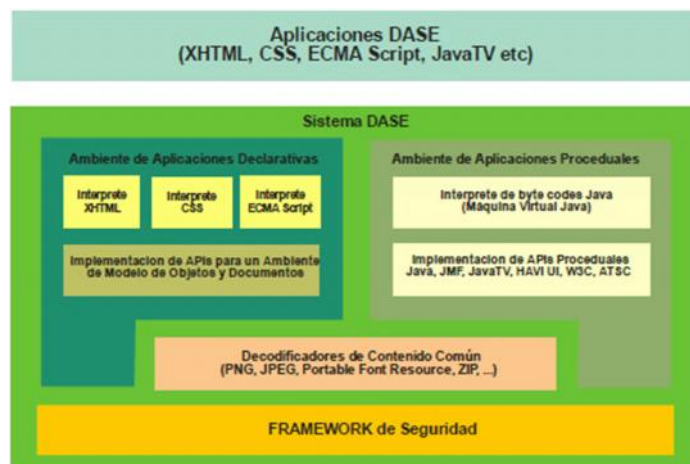


Figura 2. Arquitectura DASE

2.5.2. ACAP (Advanced Common Application Platform for Interactive Television)

Es un estándar creado por ATSC (Advanced Television Systems Committee) creado como base común para los sistemas de TV Digital Interactivo en Estados Unidos ya sean estos por cable, terrestres o por satélite; está basado en GEM (GloballyExecutable Multimedia) y añade elementos de OCAP, que es una derivación de MHP, adaptado a las características técnicas y de negocio de las empresas de difusión por cable de Estados Unidos.

Este estándar sigue las especificaciones del estándar GEM y utiliza el mismo conjunto de APIs de Java y el modelo de aplicación que utiliza MHP, la principal diferencia entre MHP y ACAP radica en que ACAP tiene obligado el canal de retorno, soporte a las aplicaciones almacenadas y utiliza la versión de carrusel DSM-CC.

ACAP divide las aplicaciones en declarativas y de procedimiento; las aplicaciones que tengan solo contenido de procedimiento son desarrolladas en JAVA y permiten combinar gráficos, video e imágenes, estas son denominadas ACAP - J; mientras que las

aplicaciones declarativas son representadas por documentos multimedia escritos en lenguaje de marcas XHTML, reglas de estilo y scripts, estas son llamadas ACAP - X. [8]

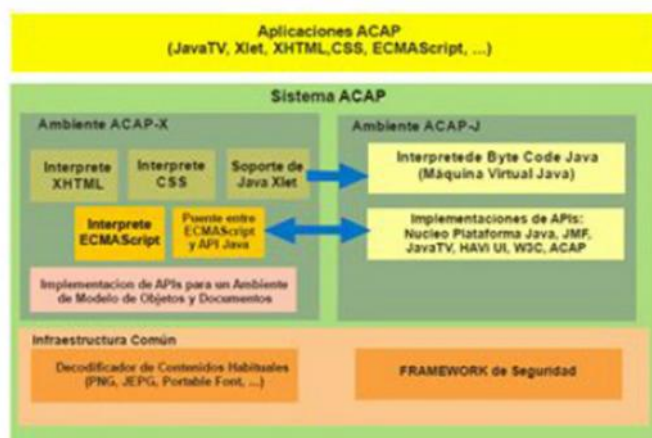


Figura 3. Arquitectura ACAP

2.5.3. ARIB (Association of Radio Industries Business)

Este middleware de ISDB – T está estandarizado por la organización japonesa ARIB, que ha definido dos estándares principales y cada una de ellas tienen las normativas necesarias para regular, manejar y controlar este proceso, estos 2 estándares son ARIB STD-B24 y ARIB STD-B23.

El estándar STD-B23 se ha desarrollado en base al estándar DVB-MHP, este define una plataforma como motor ejecutor de aplicaciones interactivas, así como establece una base común entre los estándares MHP, ACAP y GEM, para poder compartir y ejecutar aplicaciones entre estos estándares.

STD-B24 este estándar por su parte define un lenguaje declarativo llamado BML (Broadcast Markup Language) desarrollado por la propia ARIB, basado en XML y es utilizado para la especificación de los servicios multimedia interactivos; este lenguaje permite el control de la vista temporal y espacial de lo que se muestra en pantalla,

mantiene la relación entre la información y los medios de comunicación, la estructura del envío de información y la visualización de interfaces gráficas de usuario para la solicitud de información.

La arquitectura del middleware ARIB posee dos partes principales: presentación y ejecución. El modelo de ejecución utiliza la máquina virtual de Java para la interpretación de los procedimientos y funciones relacionadas con el contenido que se transmite; mientras que el modelo de presentación es el encargado de especificar la sintaxis del lenguaje de marcado BML, en el cual se incluyen etiquetas y atributos utilizados para la autoría del contenido declarativo. [10]

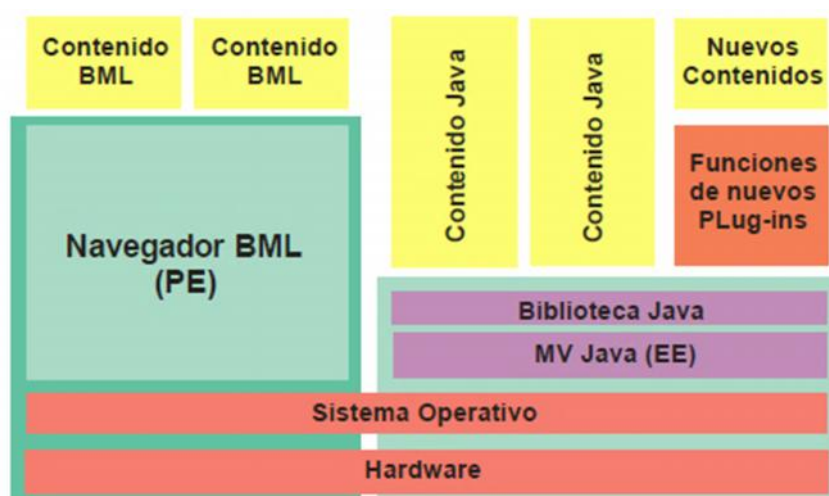


Figura 4. Arquitectura ARIB

2.5.4. GINGA

Es el nombre que recibe el middleware abierto del Sistema Brasileño de Televisión Digital (SBTVD). Es un software intermedio para el desarrollo de aplicaciones interactivas para Televisión Digital, desarrollado en Brasil, que se ha convertido en una de las especificaciones más avanzadas y la mejor solución para las necesidades de

Televisión Digital Digital tanto en Brasil como en otros países de Sudamérica que están incursionando en la Televisión Digital. Este middleware nos brinda la posibilidad de presentar contenidos en diversos receptores independientemente del hardware del fabricante y el tipo de receptor.

Las aplicaciones desarrolladas para este middleware pueden clasificadas de dos maneras dependiendo de la forma en que están escritas; las Aplicaciones de Procedimiento son escritas en lenguaje Java y las Aplicaciones Declarativas están escritas en lenguaje NCL. Existen dos subsistemas principales en este middleware, Ginga – J para las Aplicaciones de Procedimiento, y Ginga – NCL para las Aplicaciones Declarativas, ambos subsistema están entrelazados y permiten a las aplicaciones seguir dos paradigmas de programación diferentes.

Este middleware posee una gran diferencia respecto a los demás sistemas de hipertexto convencionales, y esta diferencia radica en que Ginga utiliza el modelo de “servicio tipo pull” en donde el programa solicita un nuevo documento y se procede con la búsqueda del contenido, como generalmente lo hace un navegador web; mientras que en la TV prevalece el modelo “servicio tipo push” en donde la emisora ofrece para la difusión flujos de audio, video y datos multiplexados.

Cambiar este paradigma de servicio le permite al usuario ver un programa ya iniciado, cambiar de canal y consecuentemente salir y entrar en otros programas en curso.

La arquitectura de Ginga puede ser dividida en tres módulos: Ginga – CC (Common Core), el entorno de presentación Ginga NCL (declarativo) y el entorno de ejecución Ginga – J (procedural), los dos últimos elementos son construidos sobre los servicios ofrecidos por el Common Core, esto facilita la interacción del Hardware con las necesidades de las aplicaciones tanto procedurales como declarativas.



Figura 5. Arquitectura Ginga

- Ginga – CC (Common Core), es el núcleo de presentación y ejecución. Este subsistema es un puente para la conexión con el hardware, es decir que aquí podemos acceder al sintonizador de canales, sistema de archivos, entre otros. Generalmente está compuesto por decodificadores de contenido común, los cuales sirven tanto para aplicaciones procedimentales como declarativas.
- Ginga – NCL provee un entorno de presentación multimedia para aplicaciones declarativas escritas en lenguaje NCL, este lenguaje es una aplicación XML con facilidades para aspectos interactivos.; NCL brinda soporte a dos lenguajes procedurales como LUA y Java.
- Ginga – J provee una infraestructura de ejecución de aplicaciones basadas en lenguaje Java conocidas como Xlet [8]

2.5.4.1. GINGA J

Ginga J es parte del middleware GINGA y fue desarrollado por la Universidad Federal de Paraiba UFPB, y está basado en el uso de la máquina virtual de Java y algunos APIs adicionales, lo que le proveen varias innovaciones sin perder su

compatibilidad con la Televisión Digital. Provee una infraestructura de ejecución basadas en Java de los llamados Xlets.

Como mencionamos el componente clave para la ejecución de este tipo de aplicaciones procedimentales es la Máquina Virtual de Java la cual provee el mecanismo de ejecución de contenido procedimental. Y adicional está basado en el uso de tres grupos de APIs conocidos como Verde, Amarillo y Azul.

Los Flujos de información ya sea esta audio, video, datos u otros elementos de los medios de comunicación, son recibidos y procesados por el software desarrollado en Ginga J, y estas aplicaciones son las encargadas de presentar el contenido interactivo a la audiencia [12].

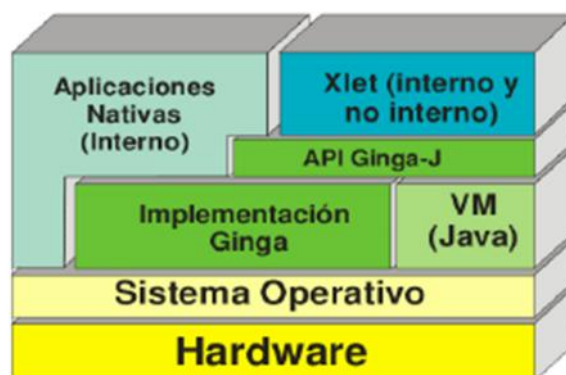


Figura 6. Arquitectura General de GINGA J

Con el fin de mantener la compatibilidad con el API de GEM, GINGA J se basa en tres grupos de APIs:

- API Verde.- estos son los APIs que son compatible con GEM, en este grupo se encuentran los APIs provenientes de los paquetes de Java TV, DAVIC y HAVI.
- API Amarillo.- este API está compuesto por el JMF 2.1 (Java Media Framework) que es una aplicación de Java de la librería javax.media.

- API Azul.- este API permite al receptor de Televisión Digital comunicarse con cualquier dispositivo que tenga una interfaz compatible, además aquí se encuentra el API que permite una conexión entre GINGA NCL y GINGA J.

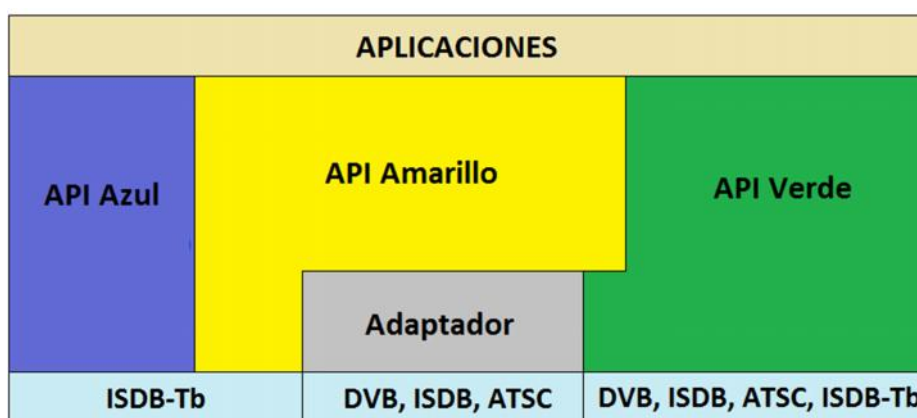


Figura 7. APIs GINGA J

2.5.5. MHP (MULTIMEDIA HOME PLATAFORM)

MHP es un sistema intermedio o middleware abierto, diseñado por la DVB y estandarizado por la ETSI, el cual define una plataforma común para las aplicaciones interactivas independientemente del proveedor de servicios o del receptor de televisión utilizado, logrando que cualquier proveedor de contenido cree y transmita aplicaciones interactivas que operen sobre cualquier tipo de terminal: Set Top Box de bajo o alto nivel, equipos de televisión digital integrados y PCs multimedia, teóricamente estas aplicaciones pueden ser transmitidas sobre cualquier tipo de red.

DVB poco después de los esfuerzos por estandarizar un API tomo la decisión de hacer uso de la tecnología Java como base de las nuevas APIs, logrando con esto un amplio mercado sin la necesidad de depender de un organismo de radiodifusión, con lo que los proveedores de servicios pueden desarrollar sus aplicaciones interactivas cumpliendo el único requisito que es ser compatible con MHP.

La primera versión del estándar (DVB – MHP 1.0) se dispuso en Julio del 2000, esta primera versión tuvo una serie de aspectos técnicos deficientes que no habían sido cubiertos en su totalidad; en consecuencia se crea la versión de la norma (MHP 1.0.1), que fue publicada por la ETSI en octubre del 2001, en junio del 2002 se introduce MHP 1.0.2.

MHP define cuatro perfiles para diferentes clases de funcionalidades:

- Enhanced Broadcast Profile (Perfil 1).- este perfil soporta la versión más simple de un ambiente MHP. Es encaminado por un STB sin un canal de retorno en un área de bajo costo, permitiendo el desarrollo de aplicaciones locales que carecen de un canal de retorno, estas aplicaciones pueden ser descargadas solo del flujo de contenido del radiodifusor y son aplicaciones como guías de programación electrónicos, tickers de noticias o aplicaciones de teletexto.
- Interactive Broascast Profile (Perfil 2).- este perfil incluye las funciones del perfil 1incluyendo soporte para un canal de retorno normalizado; basado en este canal de retorno, es posible una interacción entre los usuarios y los radiodifusores. Esto posibilita soporte a aplicaciones como Televoting, t-commerce, o pay per view (PPV). Otra ventaja de este perfil es que posibilita que las aplicaciones puedan ser descargadas vía Internet.
- Internet Access Profile (Perfil 3).- en este perfil, el perfil 3 es extendido con el soporte para aplicaciones de Internet. Aquí se especifican APIs para acceso a diferentes servicios punto a punto de Internet como email y WWW.
- IPTV Profile (Perfil 4).- este perfil integra DVB.IPTV en la plataforma MHP. DVB-IPTV está formado por una colección de varias especificaciones para el desarrollo de Televisión Digital usando IP.

2.5.5.1. VERSIONES MHP

Existen 3 versiones que cubren los perfiles estándar de MHP presentados anteriormente [13]:

➤ MHP 1.0 Enhanced&InteractiveProfiles

En esta versión se aplican los dos primeros perfiles, es decir, que no se puede tener internet en esta versión; la última versión lanzada fue la MHP 1.0.3 realizada en Marzo del 2006.

Las aplicaciones con esta versión de MHP pueden usarse para lo siguiente:

- Creación de Interfaces Gráficas e interacción con el control remoto
- Recepción de ficheros por medio del carrusel de datos, logrando con esto acceder a cualquier aplicación y brindar la información más actualizada a los usuarios.
- Recpta información de DVB-SI; este estándar permite la localización e identificación de los canales de TV, así como proporciona información de la programación actual y futura. Este estándar basa su funcionamiento en diversas tablas y permite que a través de una clave asignada guardar información en el STB con lo que se consigue que se realice una configuración automática y la sintonización de los programas del usuario.
- Recibir información de sesiones privadas de MPEG, esta información es utilizada para sincronizar las aplicaciones con la acción de la pantalla como por ejemplo los subtítulos.
- Canal de retorno TCP/UDP IP.
- Manipulación del audio y video.
- Sincronización con la emisión, es decir sincronizar la aplicación con ciertos puntos temporales del programa en emisión.

- Gestión y comunicación con otras aplicaciones interactivas que se estén ejecutando.
 - Uso de la memoria persistente del receptor, que permite guardar información aunque el terminal se encuentre apagado.
 - Gestión de subtítulos digitales.
- MHP 1.1 Enhanced Interactive & Internet Access

Esta versión aplica el perfil de acceso a Internet, Las aplicaciones de este perfil pueden usarse para:

- De manera opcional define un formato para aplicaciones basadas en XHTML (DVB-HTML) (Lenguaje extensible de marcado de hipertexto) y otras tecnologías 3WC, esto ha permitido a los desarrolladores de contenido dirigirse a diferentes dispositivos integrados permitiendo que el contenido se adapte al entorno, aprovechando de esta manera el contenido existente en la WEB.
- Permite guardar e instalara aplicación en el receptor MHP, esta funcionalidad está disponible para los PVR (Personal Video Recorders).
- Permite cargar aplicaciones mediante el uso del canal de retorno.
- Permite el uso de Plugins lo que añade funcionalidades a la plataforma, como por ejemplo ampliar el espectro de aplicaciones que han sido desarrolladas sobre otras plataformas y que puedan llegar a funcionar en la plataforma MHP.
- Permite el uso de APIs para acceso a Smart Card, esta es una tarjeta que permite almacenar datos para descifrar claves de encriptación que llegan al receptor. Además se incluye un API que permite comunicarse con un navegador WEB, cliente de e-mail, entre otros, siempre y cuando estos se incluyan en el software del receptor.

- Permite el uso de J2ME Personal Basis Profile la cual es una plataforma de Java J2ME, que proporciona un entorno robusto y flexible para aplicaciones que se ejecutan en dispositivos móviles, Teléfonos móviles, Set-Top-Boxes entre otros. Esta plataforma incluye interfaces de usuario flexibles y robustas, funcionalidades de seguridad, herramientas gráficas para aplicaciones específicas, función para protocolos de red y soporte para aplicaciones.
- Añade mejoras en la gestión de múltiples videos, aplicaciones de alta definición, entre otras.

➤ MHP 1.2

Se define la solución de DVB para plataformas multimedia en el hogar. En esta versión se añade soporte para la prestación de servicios sobre redes IPTV, además se añaden soporte para el operador, así como aplicaciones que se están ejecutando todo el tiempo mientras esté funcionando el ambiente MHP. En esta versión además se definió GEM (Global Executable MHP) que permite la interoperabilidad de MHP en otras plataformas de TDT.

2.5.5.2. DVB - GEM (Global Executable MHP)

Esta plataforma es considerada como un estándar de armonización entre plataformas de middlewares existentes con la finalidad de que todas las aplicaciones puedan ser utilizadas y presentadas en los terminales de acceso de estos estándares.

DVB decidió desarrollar esta especificación, aunque definió que no puede considerarse como una especificación completa para terminales de acceso, sino que es un Framework a partir del cual la implementación del middleware de un terminal de acceso puede ser instanciada garantizando la ejecución global de las aplicaciones.

Consiste en un subconjunto de MHP diseñado tienen en cuenta todas las posibles implementaciones del mismo sobre diferentes estándares de middleware. Este subconjunto incluye especificaciones técnicas de interoperabilidad derivados de estándares anteriores, así como especificaciones relacionados con los mecanismos de transmisión.

Este sistema busca permitir a las distintas organizaciones trabajar en armonía en especificaciones técnicas logrando con esto que tanto las aplicaciones como el contenido funcionen en plataformas basadas en GEM.

GEM ha sido adaptado por distintas organizaciones entre las que se incluye:

- Multimedia Home Plataform (MHP), desarrollado por DVB
- Open Cable ApplicationPlataform (OCAP), desarrollado como la capa de middleware para los sistemas de cable americano.
- ARIB B.23, especificación de radiodifusión en Japón.
- AdvancedCommonApplicationPlataform (ACAP), estándar de middleware de ATSC
- BD - J, plataforma JAVA para Blu-ray
- Ginga – J, framework del middleware brasileño.

2.5.5.3. ARQUITECTURA MHP

MHP define una arquitectura de alto nivel, en donde podemos encontrar 3 capas: Recursos, Software del Sistema y aplicaciones, como podemos observar en la figura a continuación [14].



Figura 8. Arquitectura MHP

Capa de Recursos: esta capa representa las diferentes plataformas de hardware existentes en el Set-Top-Box, así como los componentes de la CPU, memoria, interfaz de red y los componentes propios del televisor como interfaz DVB y el decodificador MPEG-2.

Capa del Sistema de Software: esta capa incluye el Real Time OperatingSystem (RTOS), la Java Virtual Machine (JVM) y los controladores para la comunicación con los dispositivos de hardware, brinda una vista abstracta de los recursos existentes, es una capa intermedia que aísla la aplicación del Hardware, permitiendo portabilidad de la aplicación.

Manejador de Aplicaciones: esta capa incluye una aplicación que es la responsable de la gestión del ciclo de vida de las aplicaciones en función de la señalización insertada en el Stream emitido por un proveedor DVB.

Capa API: es la interfaz entre el Software del Sistema y las Aplicaciones. Esta capa habilita la interoperabilidad de las aplicaciones sobre diferentes desarrollos MHP, proveedores de servicios y autores de aplicaciones consecuentes con el estándar DVB-MHP.

Capa de Aplicación: MHP es capaz de manejar múltiples aplicaciones sobre una misma JVM (Java Virtual Machine).

Además de las aplicaciones habituales en MHP pueden ser implementados Plugins que no son más que aplicaciones que se relacionan con otras aplicaciones para aportarles una función nueva y generalmente muy específica, estos Plugins son ejecutados por las aplicaciones principales e interactúan por medio del API.

Hay dos categorías de Plugins y aplicaciones en MHP: las interoperables y las no interoperables; las interoperables pueden ser usados por todos los tipo de receptores MHP sobre el MHP API, estas aplicaciones DVB-J han sido especificados en el API de JAVA TV. Las no interoperables utilizan código nativo o APIs especiales de JAVA que no están disponible en MHP, por esta razón no se las puede correr en cualquier tipo de receptor MHP.

2.5.5.4. APLICACIONES MHP

Las aplicaciones pueden ser interpretadas como un conjunto de funciones organizadas que solicitan la activación de diferentes recursos del software y/o software del terminal.

Las aplicaciones MHP pueden estar escritas tanto en Java (DVB-J) o en HTML (DVB-HTML), y como se vio en la sección anterior MHP soporta descarga de aplicaciones de flujos MPEG así como a partir de MHP 1.1 existe soporte para cargar aplicaciones en conexión IP utilizando protocolos con HTTP entre otros.

Existen diferencias importantes entre los tipos de aplicaciones que pueden ser ejecutadas en un STB [15]:

- **Aplicaciones Unbound.-** estas aplicaciones son descargadas pero no se asocian a ningún servicio y generalmente son usadas por los operadores de red para ofrecer servicios añadidos como por ejemplo una guía de programación electrónica. Estas

aplicaciones pueden ser cargadas mediante un canal IP de dos vías, además de que no pueden ser usadas en sistemas MHP.

- Aplicaciones Services Bound.- estas aplicaciones son descargas que sea necesario ejecutarlas, y pueden ser iniciadas por el usuario o por el receptor y son asociadas a un evento en específico.
- Aplicaciones Almacenadas.- estas aplicaciones son transmitidas y almacenadas en el terminal para ser usadas posteriormente, esto se para evitar la descarga continua o para cuando la aplicación carga más rápido localmente o también pueden ser aplicaciones independientes como un juego.
- Aplicaciones Nativas.- son aplicaciones escritas en un lenguaje diferente a Java o HTML y estas vienen incorporadas en el receptor.
- Aplicaciones Locales.- estas aplicaciones son guardadas en una memoria no-volátil del terminal MHP por el fabricante y ejemplos de estas aplicaciones son las herramientas de configuración del terminal, el navegador, etc.

2.6. JAVA TV

Sun Microsystem en Marzo de 1998 anuncia la API de JAVA TV que define una plataforma para aplicaciones de Televisión Digital, desarrollada totalmente en JAVA. Este API mejora la difusión y la experiencia visual al proporcionar información de la programación y anuncios.

Java TV proporciona un conjunto de métodos, clases e interfaces para facilitar la creación de aplicaciones interactivas de forma procedural, que se ejecutaran en distintas plataformas para la recepción de Televisión Digital independientemente de las tecnologías utilizadas en la red de transmisión.

El API brinda características como el control sobre la ejecución de contenido multimedia, acceso a la información transmitida por la señal de televisión, selección de

contenido entre otros. También proporciona acceso a las funciones de los receptores de Televisión Digital como: Streaming de Audio/Video, además de ofrecer la posibilidad de generar otras aplicaciones con otros flujos. Acceso condicional a la información de servicio, sintonizador para cambiar de canal, control gráfico de pantalla y finalmente permite la sincronización de los medios de comunicación y el control de ciclo de vida de la aplicación.

A partir de 1999 cuando DVB decidió utilizar JAVA en MHP, el API de JAVA TV fue asegurado a partir de MHP 1.0. El API de JAVA TV es un paquete opcional de JME (Java Micro Edition) [7].

2.6.1. ENTORNO DEL API

El entorno de software está formada por la plataforma JAVA y el API JAVA TV que se ejecuta sobre un Sistema Operativo. El nivel más alto del entorno de software son las aplicaciones desarrolladas para un receptor de TDT y puede usar el API y las librerías desarrolladas de clase en la plataforma JAVA. En el nivel más bajo, el sistema operativo y las librerías de los dispositivos controlan el hardware mediante una colección de controladores; el Sistema Operativo provee al sistema el soporte necesario para para implementar la máquina virtual y las librerías de la plataforma JAVA.

El entorno de hardware del API opera dentro del contexto de hardware del receptor de TDT. Este API proporciona una capa de abstracción que permite que la aplicación pueda acceder al hardware de televisión ignorando los detalles del dispositivo.

2.6.2. XLETS

El modelo de aplicaciones normales de JAVA no es compatible como producto de consumo como son los STB. En particular se asume que solo una aplicación está ejecutándose en la máquina virtual de JAVA (JVM), y cuando la aplicación termina, la máquina virtual también lo hace. Esto puede resultar complicado cuando se trabaja en un sistema donde no se pueden hacer estas suposiciones. El modelo común del ciclo de vida de una aplicación también asume que una aplicación se carga se ejecuta de inmediato y termina, lo que no es cierto en un sistema de consumo [15].

El modelo de ciclo de vida de los applets en la web es más apropiado, ya que el buscador carga una applet en la JVM, lo inicializa y lo ejecuta. Si una página contiene dos applets, estos pueden correr sin ningún problema sobre la misma JVM sin crear ninguna interferencia entre ellos; cuando una applet termina es removida de la JVM sin afectar a las demás que estén corriendo en ese momento.

Dada que el modelo de applet antes mencionado aún posee muchas funcionalidades atadas a la Web, se vio la necesidad de generalizar un modelo de ciclo de vida que sea más apropiado para los sistemas de consumo, dando como resultado los “Xlets”.

Los Xlets son la base para todos los sistemas basados en JavaTV, incluyendo MHP y OCAP. Como los applets los Xlets permiten una interfaz externa para controlar el ciclo de vida de una aplicación y brindarle comunicación con su entorno.

Existen varias similitudes en applets y Xlets, pero también existen diferencias entre ellas, y la más importante es que la ejecución de un Xlet puede ser pausada y reanudada, ya que en un ambiente como los de un receptor de TDT varias aplicaciones pueden estar corriendo al mismo tiempo, y debido a las restricciones de hardware una sola aplicación puede ser vista por el usuario, las demás aplicaciones que no están siendo usadas deben ser pausadas para mantener los recursos de hardware libres.

Un Xlet tiene 5 estados principales: Not Loaded, Loaded, Paused, Started y Destroyed. Una aplicación también puede estar en un estado Invalid, cuando no puede correr el servicio actual.

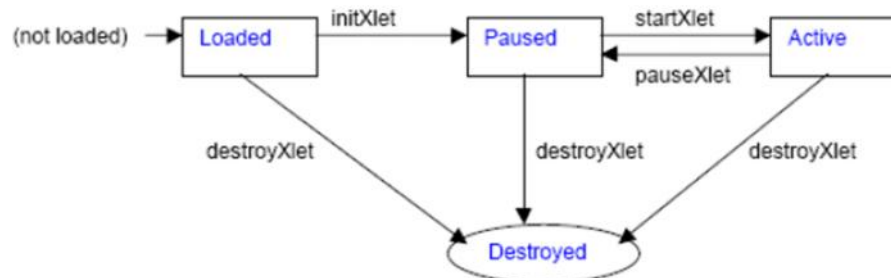


Figura 9. Estados de Ciclo de Vida Xlet

El procesamiento de los estados del ciclo de vida de una Xlet se puede describir [15]:

- **Loading:** cuando el receptor recibe la información acerca de la aplicación, la aplicación se encuentra en estado de Not Loaded. Después cuando el Xlet es iniciado, el application manager en el receptor carga los Xlets al archivo clase principal y crea la instancia del Xlet llamando al constructor por defecto. Una vez que esto ha pasado, el Xlet se encuentra en el estado Loaded.
- **Initialization:** cuando el usuario elige iniciar el Xlet o el operador de red le dice al receptor que el Xlet debe ser iniciado automáticamente, el application manager en el receptor llama al método *initXlet()*, pasando a crear un objeto tipo *XletContext*. El Xlet puede usar este objeto para inicializarse o para pre cargar grandes recursos como por ejemplo imágenes que pueden requerir de algún tiempo cargarlo desde el carrusel de objetos. Cuando la inicialización esta completa, el Xlet está en estado Paused y listo para comenzar.
- **Execution:** una vez el método *initXlet()* retorna, el application manager puede llamar al método *startXlet()*, este es capaz de mover al Xlet desde el estado de Paused al estado de Started, y el Xlet es capaz de interactuar con el usuario.

- **Pausing and Resuming:** durante la ejecución del Xlet el application manager puede llamar al método *pauseXlet()*. Este causa que la aplicación pasa nuevamente al estado Paused. La aplicación puede más tarde ser movida nuevamente al estado Started, llamado al método *startXlet()*. Esto puede ocurrir varias veces durante la ejecución del Xlet.
- **Termination:** cuando el usuario elige matar el Xlet, o cuando el operador de red le dice al receptor que el Xlet debería terminar, el application manager llama al método *destroyXlet()*. Esto causa que el Xlet sea movido a l estado Destroyed y libere todos los recursos. Después de esta instancia el Xlet no puede ser iniciado nuevamente, en las versiones MHP 1.0.3 y 1.1 .1 cambio esto, haciendo que la aplicación este temporalmente en el estado Destroyed después de esto pasaba al estado Not Loaded.

La razón de tener diferentes estados de aplicación se debe a dos restricciones MHP:

- Las aplicaciones se ejecutan en entornos de bajo rendimiento; en un terminal MHP por lo general los recursos del sistema pueden ser utilizadas por una aplicación hasta un nivel donde ninguna otra aplicación pueda empezar. En lugar de una destrucción continua de la primera aplicación, es posible liberar los recursos y ejecutarse en modo pausado. Por lo tanto es posible utilizar los recursos del terminal de manera eficiente.
- Las aplicaciones no deben colisionar entre sí: para mantener el terminal “limpio”, es necesario que las aplicaciones se liberen después de ser terminadas. Esto se logra cuando la aplicación ingresa al estado Destroyed.

El Xlet Manager es una aplicación residente, que provee un mecanismo de comunicación interno que permite a las Xlets la comunicación y la cooperación entre sí. Un solo Xlet Manager es permitido por JVM.

La clase inicial de las aplicaciones MHP debe implementar la interface `javax.xlet.Xlet` y todos sus métodos definidos, que el Xlet Manager usará para indicar

los cambios de estado de las aplicaciones. Como ya vimos anteriormente cuando el application manager es invitado a iniciar una aplicación, llamara en primer lugar a `InitXlet(XletContext ctx)` y después llamara la `startXlet()`. El método `pauseXlet()` puede enviar al Xlet al fondo y el `startXlet()` puede reanudar el Xlet. Mientras que para finalizar la aplicación el Xlet Manager llama al `destroyXlet(boolean unconditional)` donde el booleano se usa para decirle Xlet si es obligado a terminar o se tiene la opción de negar la solicitud.

Si un Xlet quiere cambiar el estado actual del ciclo de vida de la aplicación por sí mismo, debe informarle al Xlet Manager acerca del cambio. Hay tres métodos para informarle al Xlet Manager acerca del cambio:

- `resumeRequest()`: si el Xlet desea pasar de su estado Paused a Started, tiene que invocar a este método, hay que tener en cuenta que el Xlet Manager no necesita configurar el Xlet como estado activo.
- `notifyPaused()`: tan pronto como el Xlet entre al estado Paused por sí mismo sin la iniciativa del Xlet Manager, es necesario invocar este método para indicar el estado Paused al Xlet Manager, esto no es necesario en cambio cuando el Xlet Manager invoca el método `pauseXlet()`.
- `notifyDestroyed()`: cuando el Xlet entre al estado Destroyed por sí mismo sin la iniciativa del Xlet Manager mediante la invocación del `destroyXlet(boolean unconditional)`, entonces el Xlet utiliza este método para informar al Xlet Manager acerca de la autodestrucción de la aplicación.

CAPÍTULO III

DESARROLLO DEL PROTOTIPO DE APLICACIÓN

3.1. ENTORNOS DE DESARROLLO INTEGRADO PARA DESARROLLO DE APLICACIONES CON JAVA TV

3.1.1. DEFINICIÓN

Un Entorno de Desarrollo Integrado, es un entorno de programación que ha sido empaquetado en un programa, el cual posee un conjunto de herramientas que ayuda a los programadores en el desarrollo de aplicaciones. Puede dedicarse a un solo lenguaje de programación o nos permite utilizar varios [16].

Generalmente estos programas poseen las siguientes herramientas:

- Editor de texto
- Compilador
- Depurador
- Interprete
- Herramientas de Automatización
- Constructor de interfaz gráfica (GUI).

3.1.2. IDE's PARA EL DESARROLLO DE APLICACIONES PARA TV DIGITAL

Existen dos posibilidades para el desarrollo de aplicaciones de TDT en MHP: las herramientas de Desarrollo Privado y las de Desarrollo Libre. Los IDE's propietarios

que son desarrolladas por empresas privadas, y los IDE's Libres que son desarrollados por comunidades de software libre. Cada una de estas tiene sus ventajas y desventajas que las veremos a continuación [17].

Herramientas de Desarrollo Privado

Estas son herramientas de autor especializadas en el desarrollo de aplicaciones en MHP

Ventajas:

- Facilitan el desarrollo de aplicaciones, ya que poseen una interfaz gráfica muy visual lo que permite una construcción rápida de las pantallas de la aplicación.
- Mejora el acabado final de la aplicación y reduce el tiempo de desarrollo de la misma.
- No se precisan conocimientos avanzados en Java
- Permite una clara diferenciación entre las tres capas que componen las aplicaciones: presentación, propiedades y contenidos, realizando un manejo independiente de cada uno.
- Cuentan con un Emulador para PC, permitiendo hacer pruebas de diseño y funcionamiento en cualquier etapa del desarrollo.

Desventajas:

- Son herramientas que involucran un alto coste
- No se permite control sobre el código generado, por lo que no se permite la optimización del mismo.
- Lo que finalmente cree una dependencia del Entorno de Desarrollo.

Herramientas de Desarrollo Libre

En esta opción podemos optar por la utilización de herramientas de código abierto, como los entorno de desarrollo Netbeans o Eclipse y el Kit de Desarrollo de Java (JDK).

Estas herramientas al ser genéricas permiten el desarrollo de cualquier tipo de aplicación, y es por esto que estas no presentan funcionalidades ni ayudas para el desarrollo de aplicaciones para Televisión Digital.

En cuanto a las posibilidades de emulación, tenemos de forma gratuita el emulador gratuito XletView, aunque esta es una implementación incompleta hasta la actualidad:

Ventajas:

- Requiere solamente de un editor de texto con un compilador de Java.
- Se precisan conocimientos de Java.
- Se profundiza en el desarrollo de aplicaciones.
- Se tiene un control total sobre el código de la aplicación y por tanto la optimización del mismo.

Desventajas:

- Se requiere incluir todo el conjunto de clases o librerías, para poder crear aplicaciones.
- No se cuenta con un entorno amigable.
- El desarrollo se realiza de forma más lenta.
- Y la presentación final puede no ser satisfactoria si no se le dedica tiempo.

3.1.3. EVALUACIÓN DE UN IDE

Para la evaluación de un producto de software existen una diversidad de estudios y técnicas, pero en este proyecto se escogió el modelo ISO 9126 [18].

Este modelo es un estándar internacional que proporciona un esquema para la evaluación de la calidad de Software.

Esta normativa define 6 características de la aplicación, y estas a su vez poseen sub-características, las cuales representan un modelo más detallado para la evaluación.

La norma recomienda medir las características de forma directa, aunque no especifica cómo se debe realizar esta medida. Por lo que se propone la utilización de la Técnica de Evaluación llamada Análisis de Características, el cual nos brinda un tipo de valoración bastante sencillo que lo podemos utilizar para tazar y categorizar atributos.

Para deberemos seguir lo siguientes pasos:

- Primero debemos escoger los atributos claves que deberá tener la herramienta.
- Identificamos las herramientas posibles.
- Evaluamos los criterios antes mencionados en un rango de 1 (No Satisface) hasta 5 (Satisface Completamente).
- Y a cada criterio se le da un nivel de importancia desde 1 (Poco Importante) a 5 (Muy Importante).
- Multiplicamos el valor de la importancia por el puntaje del criterio y se realiza una suma.
- Y sobre la base de estas puntuaciones, se puede seleccionar la herramienta.

A continuación se describen las características y atributos propios de este estándar que se encuentran dentro de las vistas interna y externa, las cuales usaremos para evaluar el software

Tabla 1. Modelo de Calidad ISO 9126

Características	Atributos
Funcionalidad	Adecuación
	Exactitud
	Interoperación
	Seguridad
Confiabilidad	Madurez
	Tolerante a defectos
	Facilidad de Recuperación
Facilidad de Uso	Facilidad de Comprensión
	Facilidad de Aprender
	Facilidad de Operar
Eficiencia	Comportamiento en el Tiempo
	Comportamiento en los Recursos
Portabilidad	Adaptabilidad
	Facilidad de Instalación
	Conformidad
	Facilidad de Reemplazo

A continuación se definen las características del Modelo

- **Funcionalidad.-** Conjunto de atributos que permiten calificar si el software maneja las funciones adecuadas que satisfagan las necesidades para las cuales fue diseñado.
- **Confiabilidad.-** Conjunto de atributos que indican sobre la capacidad del software para mantener su nivel de desempeño durante un tiempo establecido y bajo un conjunto de condiciones definidas.
- **Facilidad de Uso.-** Conjunto de atributos que determinan el esfuerzo necesario para utilizar el producto satisfactoriamente por parte de un conjunto de usuarios.
- **Eficiencia.-** Conjunto de atributos que afectan la relación entre el desempeño del software y la cantidad de recursos utilizados.

- Portabilidad.- Conjunto de atributos que afectan sobre la capacidad del software para ser transferido de una ambiente a otro.

A continuación se definen los atributos del Modelo:

- Adecuación.- Evalúa si el software cuenta con las funciones apropiadas para realizar las tareas para las cuales fue desarrollado.
- Exactitud.- Evalúa si el software presenta los resultados acorde a las necesidades para los cuales fue desarrollado.
- Interoperación.- Evalúa la habilidad el software para interactuar en otros sistemas especificados.
- Seguridad.- Evalúa la habilidad del software para prevenir acceso no autorizado a los programas y datos.
- Madurez.- Evalúa la capacidad del software para evitar fallas como resultado de errores del software.
- Tolerancia a defectos.- Evalúa la habilidad del software para mantener un nivel específico de funcionamiento en caso de fallas.
- Facilidad de Recuperación.- Hace referencia a las capacidad de reestablecer el nivel de operación y recuperar los datos que fueron afectados por una falla, así como el tiempo y esfuerzo requeridos para esta recuperación.
- Facilidad de Comprensión.- Hace referencia al esfuerzo requerido por parte de los usuarios para reconocer las estructura lógica así como los conceptos relativos al software.
- Facilidad de Aprender.- Hace referencia al esfuerzo que deben realizar los usuarios para aprender a usar la aplicación del software.
- Facilidad de Operar.- Agrupa los conceptos que avalúan las operación y control del sistema.
- Comportamiento en el Tiempo.- Atributos de software relativo a tiempos de respuesta y de procesamiento de datos.

- Comportamiento de los recursos.- Atributos del software referentes a la cantidad de recursos utilizados y la duración de su uso en la realización de sus funciones.
- Adaptabilidad.- Evalúa la capacidad de adaptación del software a diferentes ambientes sin la necesidad de aplicarle modificaciones.
- Facilidad de Instalación.- Esfuerzo necesario para instalar el software en un ambiente determinado.
- Conformidad.- Capacidad del software para adaptarse a estándares relacionados con portabilidad.
- Facilidad de Reemplazo.- Hace referencia al esfuerzo requerido para sustituir el software por otro producto con funciones similares.

El proyecto actual se va a enfocar en el desarrollo de una aplicación interactiva mediante el uso netamente del API de Java TV por lo se va a optar por el uso de Herramientas de Desarrollo Libre y el IDE seleccionado fue Netbeans con el Emulador XletView.

Aunque ya hayamos seleccionado la herramienta de desarrollo, vamos a realizar la evaluación del mismo con el esquema mencionado anteriormente.

Tabla 2. Análisis de Características del Modelo de Calidad ISO 9126

Características	Atributos	Netbeans	Importancia
Funcionalidad	Adecuación	3	3
	Exactitud	3	3
	Interoperación	5	5
	Seguridad	-	-
Confiabilidad	Madurez	3	5
	Tolerante a defectos	4	5
	Facilidad de Recuperación	4	5
Facilidad de Uso	Facilidad de Comprensión	2	5
	Facilidad de Aprender	2	5
	Facilidad de Operar	2	5
Eficiencia	Comportamiento en el Tiempo	3	5
	Comportamiento en los Recursos	4	5
Portabilidad	Adaptabilidad	5	5
	Facilidad de Instalación	4	3
	Conformidad	3	5
	Facilidad de Reemplazo	5	5
Total		240	

Algunos atributos adicionales que podemos tomar en cuenta en la elección de Netbeans son los siguientes:

Tabla 3. Atributos adicionales

Atributos	Netbeans
Licencia	Propietaria
Precio	Gratis
Licencia del Emulador	Libre
Software Adicional	JRE, JMF, J2ME, Java TV, HAVI, DVB
S.O. que lo soporta	MAC OS, Linux, Windows 2000 en adelante
Niveles de Interactividad de las aplicaciones	Unidireccion al Solo emisión
Tipo de Software que puede desarrollarse	Todo tipo de aplicaciones

Según la Tabla del Análisis de Características se puede observar que Netbeans con el emulador XletView permite la realización de aplicaciones interactivas para Televisión Digital pero la codificación se vuelve extensa y bastante monótona, a pesar de esta la selección de esta herramienta se da ya que el enfoque de este proyecto es analizar la utilización del API Java TV, y para esto es necesario que podamos tener control sobre el código de la aplicación.

3.2. ANÁLISIS Y DISEÑO DEL PROTOTIPO

3.2.1. ESPECIFICACIÓN DE REQUISITOS

El prototipo de la aplicación será de carácter informativo, y permitirá a los usuarios conocer acerca de los sitios turísticos más representativos de la Capital, mostrará información importante sobre los lugares, así como una galería de fotografías de cada sitio.

Adicional se incluye un pequeño juego de trivias sobre la información que se presentó sobre cada uno de los lugares.

La aplicación estará alojada de manera local en el decodificador, el cual será el encargado de presentarlo a los usuarios.

REQUERIMIENTOS DEL USUARIO

- La aplicación debe ser fácil de usar.
- Debe ser fácilmente navegable mediante los botones del control remoto.
- No debe impedir que el usuario pueda mirar la programación en Televisión, mientras esté utilizando la aplicación.
- La información presentada por la aplicación debe ser legible desde una distancia de 2 m.
- Su uso debe ser intuitivo.

REQUERIMIENTOS DEL SISTEMA

- La interfaz de usuario debe basarse en el uso del API Java TV para el desarrollo de la aplicación interactiva.
- La aplicación debe ser netamente desarrollada con Java.
-

3.2.2. DESARROLLO DE PROTOTIPO

En esta sección del capítulo se dará a conocer el proceso de desarrollo del prototipo, se enlistara las herramientas de software utilizadas, y se presentara un breve tutorial de cómo se puede desarrollar una aplicación con el API de Java TV; y finalmente presentaremos el resultado del desarrollo del Prototipo de la aplicación de Turismo ejecutándose en el emulador seleccionado.

3.2.2.1. SOFTWARE GENERICO

3.2.2.1.1. NETBEANS

NetBeans es un entorno de desarrollo integrado muy completo. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. También incorpora un editor propio.

Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con los APIs de NetBeans y un archivo especial llamado manifest file que lo identifica como módulo. Las aplicaciones creadas a partir de módulos pueden ser fácilmente extendidas agregándole nuevos módulos. Debido a la independencia en el desarrollo de los módulos, las aplicaciones desarrolladas en esta plataforma pueden ser fácilmente extendidas por otros desarrolladores de software [7].

NetBeans es un proyecto de código abierto que ha tenido un gran éxito con una gran base de usuarios y una comunidad en constante crecimiento. La plataforma ofrece servicios comunes para aplicaciones de escritorio, permitiendo al desarrollador

enfocarse en la lógica específica de su aplicación. Entre las características que nos ofrece esta plataforma encontramos [7]:

- Gestión de las interfaces de usuario.
- Gestión de las configuraciones de usuario.
- Gestión del almacenamiento.
- Gestión de ventanas.
- Framework basado en asistentes (Diálogos Paso a Paso).
- Librería visual de NetBeans.
- Herramientas de desarrollo integrado.

3.2.2.1.2. XLETVIEW

XletView, es un emulador usado para ejecutar Xlets en un computador de escritorio, es de código abierto, está bajo una licencia GPL (General Public Licence), además de tener una implementación de referencia a la API de Java TV, trae además implementaciones de otras APIs especificadas en el estándar MHP, como HAVI (Home Audio Video Interoperability), DAVIC (Digital Audio Video Council) e implementaciones especificadas por la propia DVB (Digital Video Broadcasting), además de las bibliotecas de Personal Java [2].

XletView está desarrollado en Java y para su ejecución independiente del sistema operativo, es necesaria la utilización de Java 2 Standar Development Kit para compilar los Xlets y ejecutar XletView. Este emulador utiliza JMF (Java Media Framework) 2.1.1 pero con alguna deficiencias como por ejemplo la incapacidad de exhibir videos MPEG relacionados o controlados por un Xlet.

Se puede acceder libremente a las versiones más actualizadas de este emulador en la dirección electrónica <http://www.xletview.org/>.

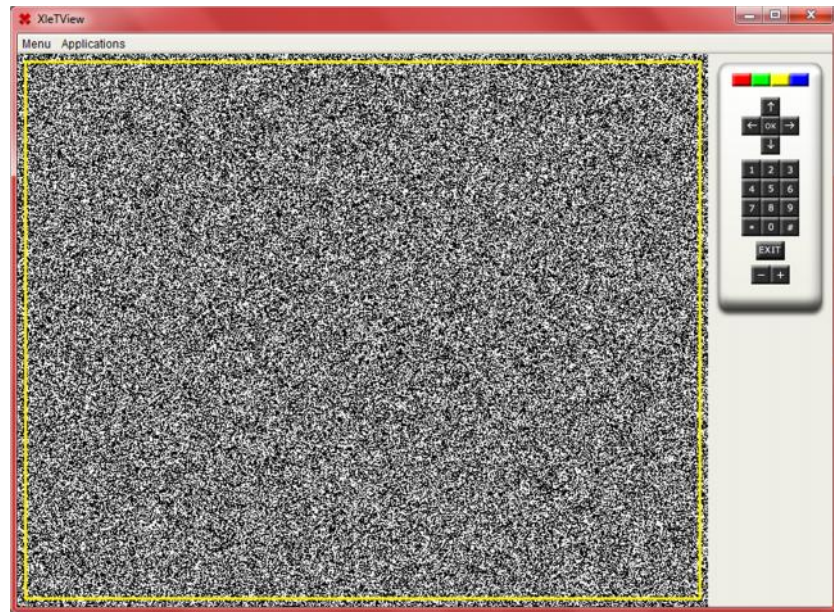


Figura 10. Interfaz XletView

3.2.3. APIs NECESARIAS PARA EL DESARROLLO DEL PROTOTIPO

A continuación se describen las principales APIs Java utilizadas para el desarrollo de aplicaciones interactivas, y que fueron tomadas en cuenta para el desarrollo del prototipo en este proyecto [7]:

3.2.3.1. APIs NUCLEO:

Tabla 4. APIs de Nucleo

API	Descripción
java.awt	Es la encargada de construir aplicaciones con interfaces de usuarios gráficas. Se la puede clasificar en dos formas: Creación de la interfaz Definición del comportamiento.
java.awt.event	Proporciona interfaces y clases para tratar con diferentes eventos lanzados por otros componentes AWT.
java.awt.image	Contiene clases que nos permiten la creación de nuevas imágenes y modificar las imágenes ya existentes.
Java.beans	Contiene las clases relacionadas al desarrollo con Java Beans, el cual nos permite encapsular varios objetos en un único objeto llamado Bean, y hacer uso de este único objeto en vez de varios más simples.
Java.io	Proporciona el sistema de entrada y salida a través del flujo de datos, la serialización y los archivos del sistema.
Java.lang	Proporciona las clases fundamentales para el diseño en lenguaje de programación Java.
Java.lang.reflect	Proporciona clases e interfaces que nos permiten obtener información reflexiva sobre clases y objetos.
Java.net	Proporciona clases que nos permiten implementar aplicaciones de Red.
Java.security	Proporciona las clases e interfaces para el marco de seguridad.

CONTINUA



Java.security.cert	Proporciona clases e interfaces para el análisis y el manejo de certificados
Java.util	Contiene los marcos de colecciones, clases de colección de legados, modelo de eventos, facilidades para fecha y tiempo, internacionalización, y diversas clases de utilidades como por ejemplo, generador de números randómicos, arreglo de bits, entre otros.
Java.util.zip	Proporciona clases para la lectura y escritura de archivos en formato ZIP y GZIP

3.2.3.2. APIs de JAVA MEDIA FRAMEWORK (JMF)

Este API es utilizado para controlar como son decodificados y mostrados los contenidos de audio y video. Si bien su uso principal es para el control de los datos que se transmiten en forma de video, las aplicaciones que utilizan JMF pueden también reproducir archivos de audio o mostrar un tipo especial de formato de imagen.

JMF 1.0 fue diseñado originalmente para controlar el flujo de audio y video desde un archivo local o desde un servidor de archivos de video en una red; para los flujos de emisión JMF fue extendido por DAVIC con su paquete org.davic.media. El JMF 1.0 utilizado para televisión interactiva contiene los siguientes paquetes:

- Java.media
- Java.media.protocol

3.2.3.3. APIs JAVA TV

Tabla 5. APIs de Java TV

API	Descripción
Javax.tv.graphics	Añade soportes a AWT en cuestiones específicas de televisores como la mezcla de video y gráficos y descubrir el contenedor principal de la aplicación
Javax.tv.locator	Proporciona mecanismos de referencia como URL de transmisión de servicios y clips de video
Javax.tv.media	Añade soportes para funcionalidades específicas de TV a JMF
Javax.tv.media.protocol	Agrega soporte para protocolos de emisión de flujos a JMF
Javax.tv.net	Proporciona mecanismos para el acceso a datagramas IP contenidos en una emisión de flujo
Javax.tv.service	Proporciona conceptos de alto nivel para describir servicios de TV digital. Este paquete también proporciona el mecanismo básico para consultar la información del servicio de radiodifusión.
Javax.tv.service.guide	Soporte para aplicaciones de tipo EPG, incluyendo conceptos para describir eventos individuales, cronogramas de programas e información de rating.

CONTINUA



Javax.tv.service.navigation	Soporte para servicios navegables, donde se puede definir a un canal de tv como un servicio. Esto incluye soporte para lista de servicios favoritos, componentes dentro de los servicios y tipos de servicios.
Javax.tv.service.transport	Conceptos que describen los mecanismos de transporte usados por un servicio de TV Digital, como el flujo de transporte, redes de difusión y ramas.
Java.tv.util	Clases de utilidades para aplicaciones JavaTV, incluyendo la administración de timers y eventos timer.
Java.tv.xlet	Modelo de ciclo de vida de las aplicaciones y clases de soporte.

3.2.3.4. APIs DE DAVIC (Digital Audio Video Council)

DAVIC fue una organización creada para promover las aplicaciones audiovisuales mediante estándares abiertos. Se basó en MPEG y agregó un API de Java ofreciendo un mejor control de un receptor para una aplicación.

Tabla 6. APIs de DAVIC

API	Descripción
Org.davic.media	Incluye extensiones de JMF relevantes para los contenidos de TV. Los más importantes es el control de selección de idioma para el audio y los subtítulos.
Org.davic.mpeg	Incluye conceptos del sistema MPEG a través del flujo de transporte, los servicios y las clases de Flujo Elemental
Org.davic.mpeg.dvb	Contiene versiones extendidas de clases con información adicional específica de DVB
Org.davic.mpeg.sections	Proporciona la habilidad de filtrar secciones específicas de MPEG en un flujo de transporte
Org.davic.net	Trata referencias de contenido por medio de la clase Locator.
Org.davic.net.dvb	Esta clase es más específica para DVB
Org.davic.net.tuning	Maneja la sintonización explícitamente entre los flujos de transporte.
Org.davic.resources	Contiene el marco de notificación de recursos adaptado en MHP.

3.2.3.5. APIs DE HAVI (HOME AUDIO VIDEO INTEROPERABILITY)


HAVI definió un extenso conjunto de paquetes basados en Java para dispositivos electrónicos de consumo. Una parte de HAVI que ha sido adoptado por los estándares que soportan Java, incluyen paquetes de Interfaz de Usuario.. Este conjunto de paquetes requieren de un subconjunto de paquetes del estándar AWT para una definición más completa. La interfaz de usuario de HAVI contiene los paquetes:

- Org.havi.ui
- Org.havi.ui.event

3.2.3.6. APIs DE DVB

Tabla 7. APIs de DVB

API	Descripción
Org.dvb.application	Proporciona acceso a las listas de aplicaciones, las cuales están disponibles en su contexto y habilidad para ser ejecutadas

CONTINUA 

Org.dvb.dsmcc	Proporciona acceso a los archivos transportados en la emisión de flujo. Permite acceso a objetos como flujos, eventos de flujo y descriptores NPT. Una de las clases más importantes es la DSMCCObject, en donde instancias de este objeto representa un objeto del carrusel DSM-CC, DSM-CC es un formato para la transmisión de datos e información de control en una sección privada me MPEG-2
Org.dvb.event	Proporciona acceso a eventos de entrada de usuario antes de que sean procesados a través del mecanismo de eventos del paquete AWT.
Org.dvb.io	Proporciona soporte para la comunicación entre aplicaciones y una extensión del paquete java.io para acceder a archivos que se mantienen en almacenamiento persistente.
Org.dvb.io.ixc.lxcRegistry	Clase que permite obtener una referencia a un objeto remoto que reside en un Xlet diferente pero se encuentra en el mismo terminal.
Org.dvb.io.persistent	Proporciona acceso a archivos en almacenamiento persistente.

CONTINUA



Org.dvb.lang	Proporciona características relacionadas con la plataforma central que no se encuentra en el paquete java.lang.
Org.dvb.media	Proporciona extensiones específicas DVB para JMF.
Org.dvb.net	Proporciona características generales de las redes que no han sido consideradas en otros paquetes, posee extensiones para el API de acceso condicional de DAVIC, administración de sesión de conexiones IP bidireccionales.
Org.dvb.si	Proporciona acceso a los servicios de información DVB.
Org.dvb.ui	Proporciona funcionalidad grafica ampliada.
Org.dvb.user	Proporciona acceso a configuraciones y preferencias realizadas por el usuario final.

3.2.4. HERRAMIENTAS DE SOFTWARE NECESARIAS

Para poder desarrollar una aplicación interactiva con el API de JAVA TV es necesaria incluir lo siguiente:

- Descargar el paquete de Java TV disponible en <http://download.oracle.com/otndocs/jcp/javatv-1.1-mr-oth-JPR/>
- Descargar el paquete de XletView disponible en Sourceforge (<http://sourceforge.net/projects/xletview/files/XleTView/xletview-0.3.6/>).

- Herramientas de Java para la ejecución de NetBeans (<https://netbeans.org/downloads/>).

3.2.5. EJEMPLO APLICACIÓN BASICO DE XLET CON NETBEANS Y XLETVIEW

A continuación, se dará una introducción, por medio de un ejemplo básico, do como se puede desarrollar una aplicación mediante el uso del API de Java TV con las herramientas que mencionamos anteriormente.

Iniciamos NetBeans y creamos un Nuevo Proyecto

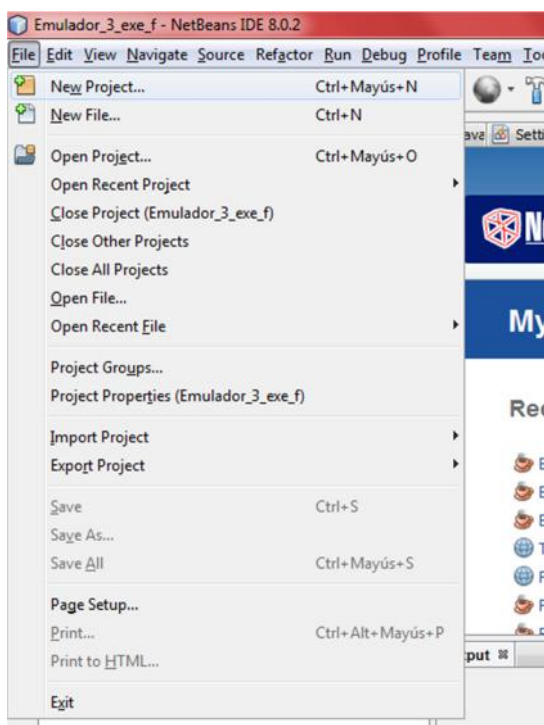


Figura 11. Crear Nuevo Proyecto

Seleccionamos la Categoría Java y seleccionamos el tipo de proyecto Java Application

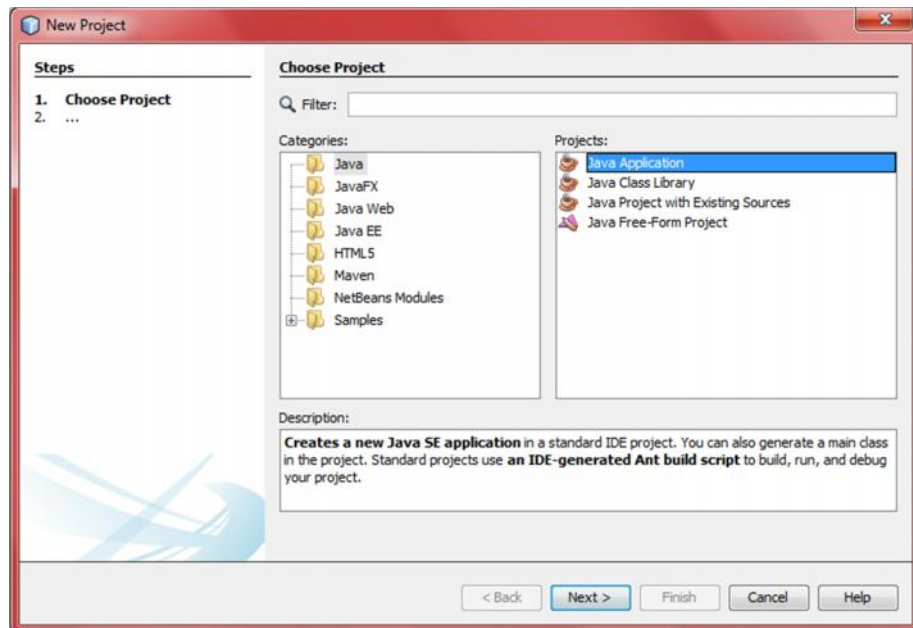


Figura 12. Selección de Categoría y Tipo de Proyecto

Escribimos el nombre del Proyecto en nuestro caso será Hola_Mundo, y quitamos la selección para crear una clase Main

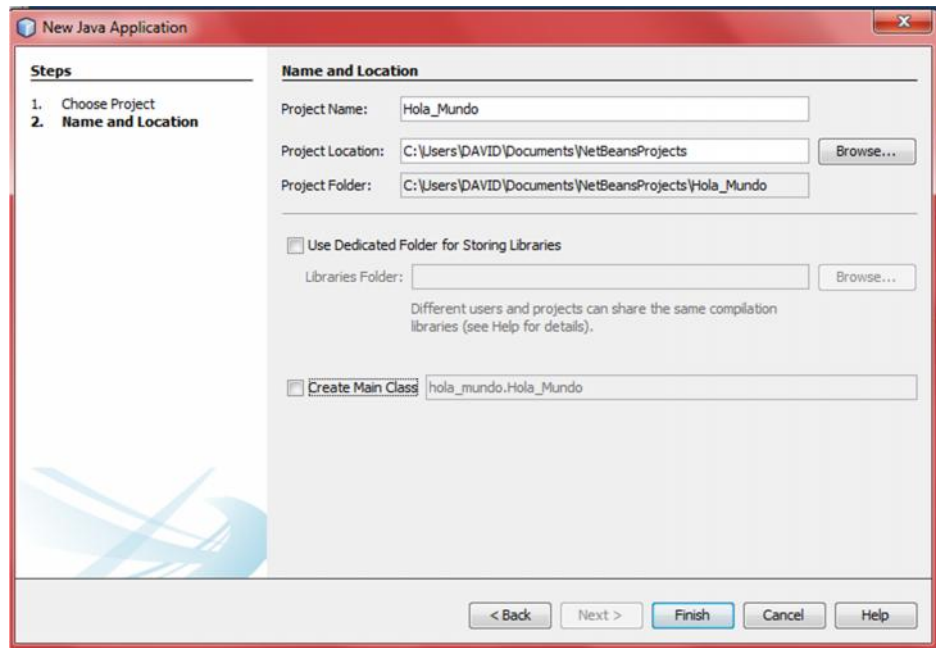


Figura 13. Ingreso del nombre del proyecto

Presionamos Finalizar

En la barra de Navegación de NetBeans veremos ya nuestro proyecto creado, ahora lo que haremos es crear una nueva clase Java, para esto damos clic derecho sobre nuestro proyecto, seleccionamos New y hacemos clic sobre Java Class.

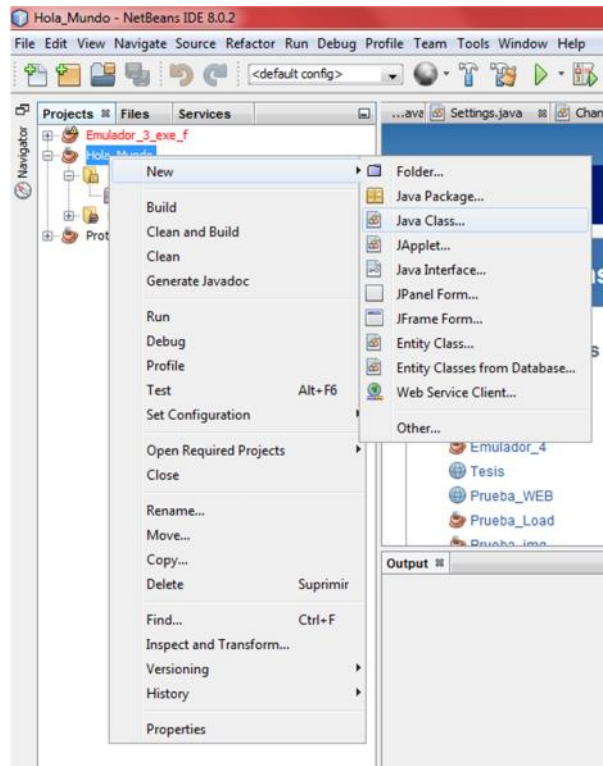


Figura 14. Creación Nueva Java Class

Escribimos un nombre para nuestra nueva clase y damos clic en Finalizar.

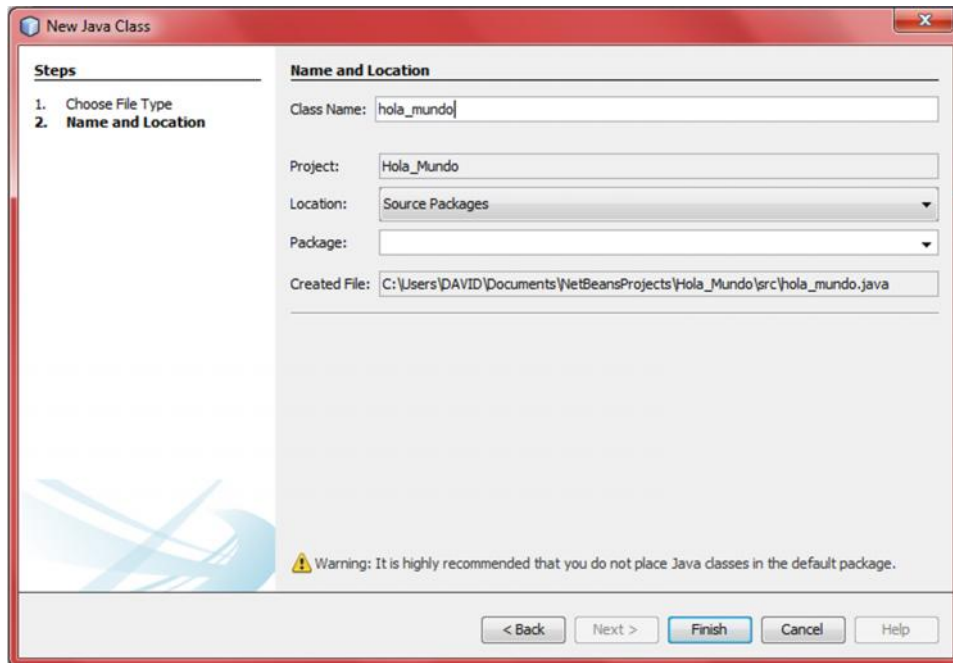


Figura 15. Nombre de nueva Java Class

Ahora debemos incluir las librerías necesarias que fueron descritas en las secciones anteriores, para esto damos clic derecho la carpeta Libraries de nuestro proyecto, y seleccionamos Add Jar/Folder

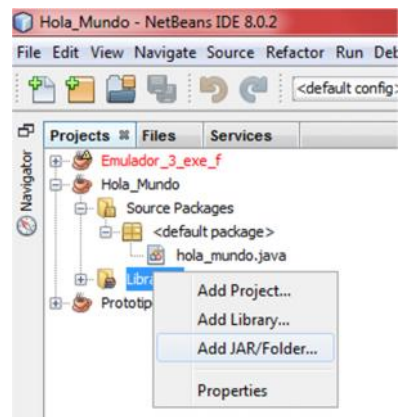


Figura 16. Agregar Librerías

Navegamos hasta donde tengamos las librerías necesarias, las seleccionamos y presionamos Abrir

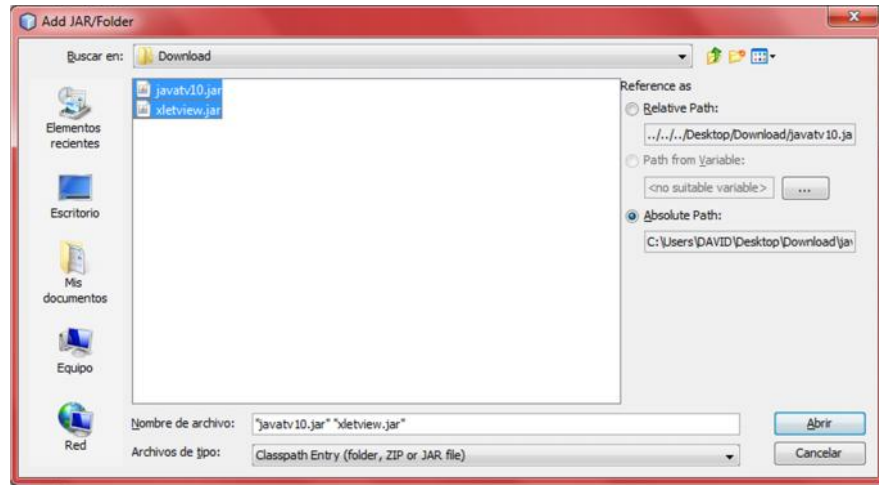


Figura 17. Selección de Librerías XletView y Java TV

Ahora dentro de la clase que creamos anteriormente debe crear la estructura básica de un Xlet, la cual se muestra a continuación

```

public class hola_mundo implements Xlet {
    public hola_mundo() { ...3 lines }
    public void initXlet(XletContext xletContext) throws XletStateChangeException { ...5 lines }
    public void startXlet() throws XletStateChangeException { ...29 lines }
    public void pauseXlet() { ...3 lines }
    public void destroyXlet(boolean unconditional) throws XletStateChangeException { ...7 lines }
}

```

Figura 18. Métodos del Xlet

Y escribimos el código de la aplicación

```
import java.awt.*;
import java.awt.event.*;
import javax.tv.xlet.*;
import org.havi.ui.*;
import org.havi.ui.event.*;

public class hola_mundo implements Xlet{
    private XletContext context; //Variable que nos permite guardar el contexto
    private HScene scene; //Variable que hace referencia a la escena
    private boolean i; //Variable que nos indica si se ha ingresado al metodo Start

    public hola_mundo() {
    }

    public void initXlet(XletContext xletContext) throws XletStateChangeException {
        /* guardando el contexto... */
        this.context = xletContext; //Almacenamos el Contexto
        i = true; //Indicamos que el Xlet se ha Inicializado
    }

    public void startXlet() throws XletStateChangeException {
        if (i = false){
            return; //El Xlet no se ha inicializado aun, se debe invocar a Init
        }

        i = false; //Con esto conseguimos que se pueda volver a entrar al init cuando
        //se reanude el Xlet
    }
}
```

```

//Conseguimos la referencia a la capa grafica a traves de la pantalla
HScreen pantalla = HScreen.getDefaultHScreen(); //referenciamos a la pantalla
por defecto
HGraphicsDevice capagraf = pantalla.getDefaultHGraphicsDevice();//ref a la
capa grafica
//a
traves de la ventana

//Ahora conseguimos una HScene a pantalla completa
HSceneFactory factory = HSceneFactory.getInstance();
scene = factory.getFullScreenScene(capagraf); //Pasamos la Escena a la Capa
Grafica
scene.setBackgroundMode(HScene.BACKGROUND_FILL);

//Ahora añadimos los elementos que forman la interfaz de usuario

HStaticText label = new HStaticText ("HOLA MUNDO! \nEjemplo Basico
Xlet",50,100,600,10,
new java.awt.Font("Verdana",java.awt.Font.BOLD,28),
java.awt.Color.BLACK,java.awt.Color.WHITE,
new HDefaultTextLayoutManager());

scene.add(label); //Agregamos el Texto a la Escena

scene.show(); //Hacemos visible la Escena
}

public void pauseXlet() {
/* Método vacío */

```

```
    }  
  
    public void destroyXlet(boolean unconditional) throws  
XletStateChangeException {  
        if (scene != null){           //Se destruye la escena en caso de estar creada  
            scene.setVisible(false); //Dejamos de mostrar la escena  
            scene.dispose();         //Liberamos los recursos  
            scene = null;           //Con esto evitamos que se vuelva a ingresar al Destroy  
        }  
    }  
}
```

Compilamos el proyecto y ahora nos dirigimos hacia donde tengamos el emulador XletView descargado, y descomprimos y ejecutamos el archivo .jar, el cual nos mostrara la interfaz del emulador.

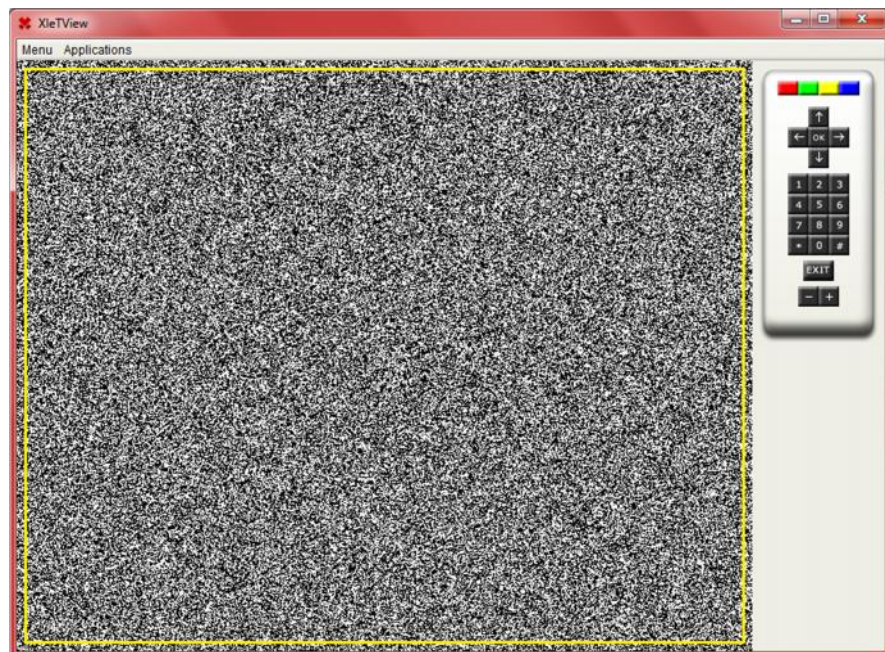


Figura 19. Interfaz Xletview

En el emulador nos dirigimos a Applications y presionamos en Manage Applications

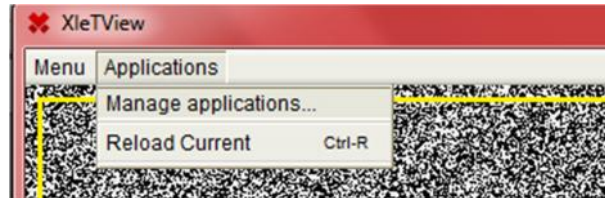


Figura 20. Administrador de aplicaciones XletView

En esta ventana presionamos sobre el botón New Application, en Name colocamos el Nombre que deseamos darle a la aplicación, seleccionamos la ruta en donde compilamos nuestro Xlet, y en Xlet colocamos el archivo .class en donde esta compilado nuestro proyecto.

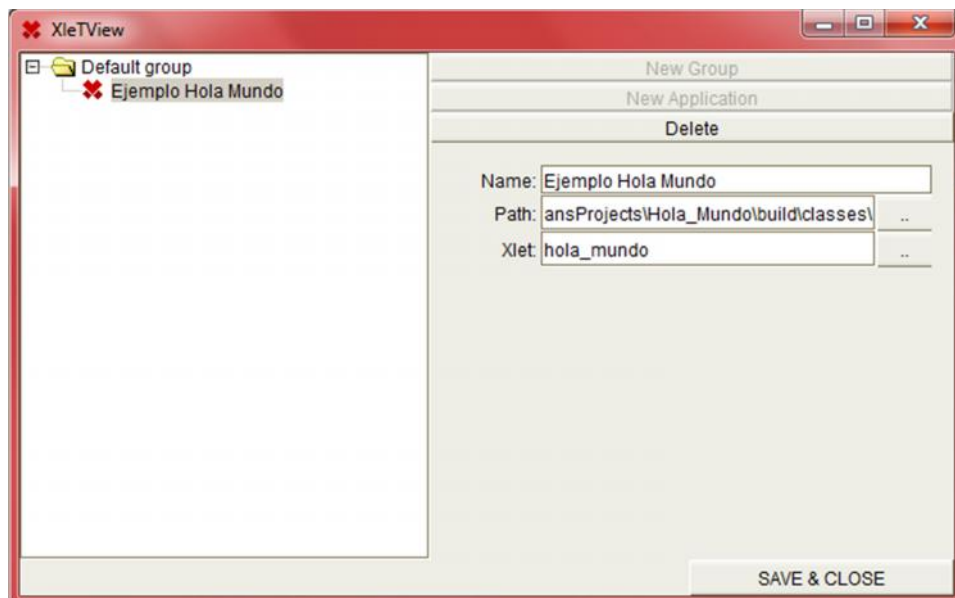


Figura 21. Selección del Xlet

Finalmente iremos a Applications, escogemos el nombre de la aplicación ingresada en el paso anterior y podemos ver el despliegue del Xlet en la pantalla.

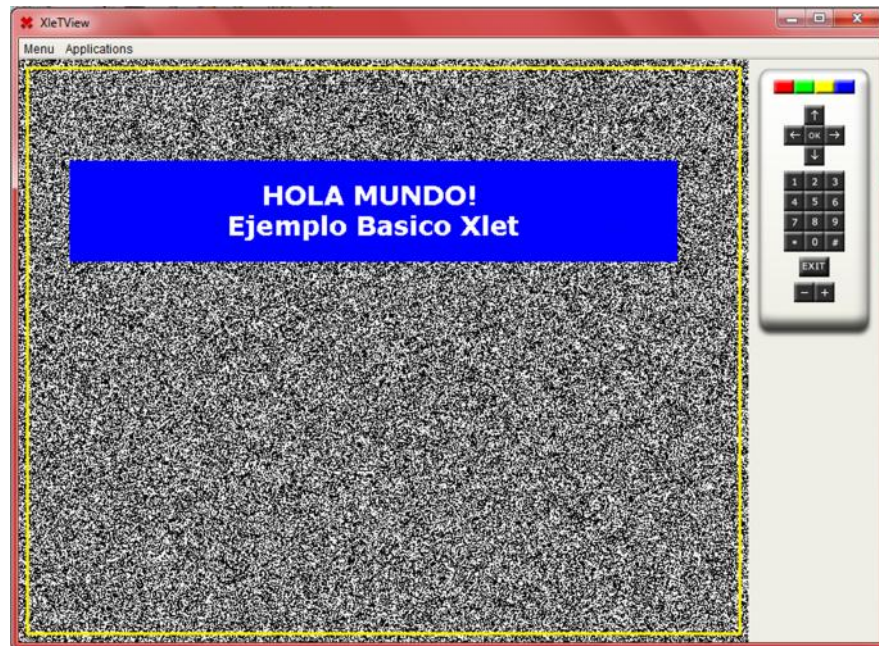


Figura 22. Xlet Simulado

3.2.6. DISTRIBUCIÓN DEL PROYECTO

Como vimos anteriormente el prototipo se realizó utilizando el IDE Neatbeans, el cual nos permitió codificar la aplicación interactiva para luego probar su uso mediante el uso del emulador XletView.

A continuación se muestra la distribución del proyecto en Neatbeans:

Las clases que se observan en la imagen anterior conforman la interfaz de nuestro prototipo, cada una está asociada a una pantalla de presentación dentro de la aplicación.

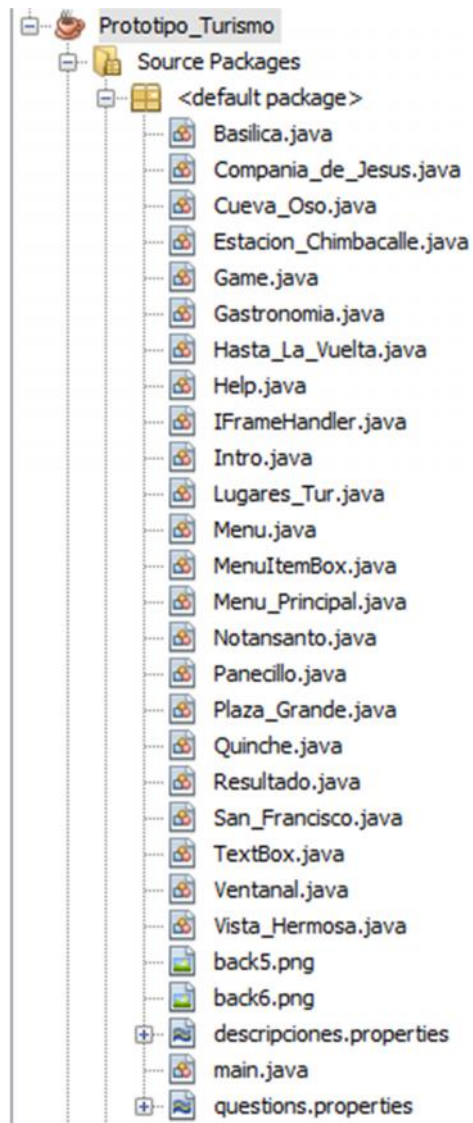


Figura 23. Clases del Proyecto

Las carpetas que se denotan dentro del proyecto contienen las fotografías de cada uno de los sitios turísticos que se van a presentar en la aplicación.

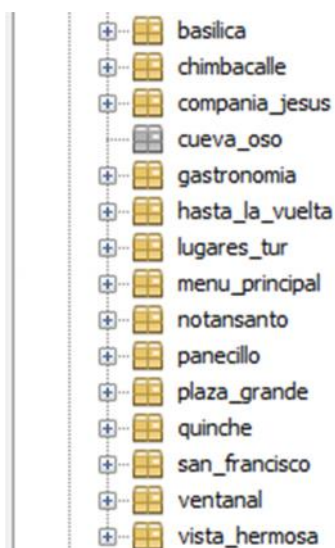


Figura 24. Carpetas Fotografías

Para poder utilizar el API de Java TV para el desarrollo del prototipo se necesitan de las librerías que se muestran en la imagen:



Figura 25. Librerías

La librería *xletview* contiene las librerías adicionales que nos permiten la creación de las interfaces de la aplicación. Y la librería *javatv* contiene las librerías que nos permiten el desarrollo de las aplicaciones interactivas para Televisión Digital; cada una de estas librerías será expuesta más adelante en este capítulo.

3.2.7. PROTOTIPO DE APLICACIÓN TURISMO

Esta sección se enfoca en mostrar el resultado desarrollado del prototipo de aplicación desarrollado con el API Java TV utilizando el IDE NetBeans.

El prototipo se ejecuta en el emulador XletView, la aplicación puede ser habilitada presionando el botón OK cuando la leyenda en la pantalla se lo indique, y puede ser deshabilitada cuando nos encontremos en el Menú Principal de la aplicación presionando el botón verde.

A continuación se presenta la pantalla de introducción, donde la aplicación se encuentra a la espera de que el usuario presione el botón OK para ingresar a la aplicación.



Figura 26. Pantalla Inicial de la Aplicación

Al presionar el botón OK, se muestra el menú principal de navegación.

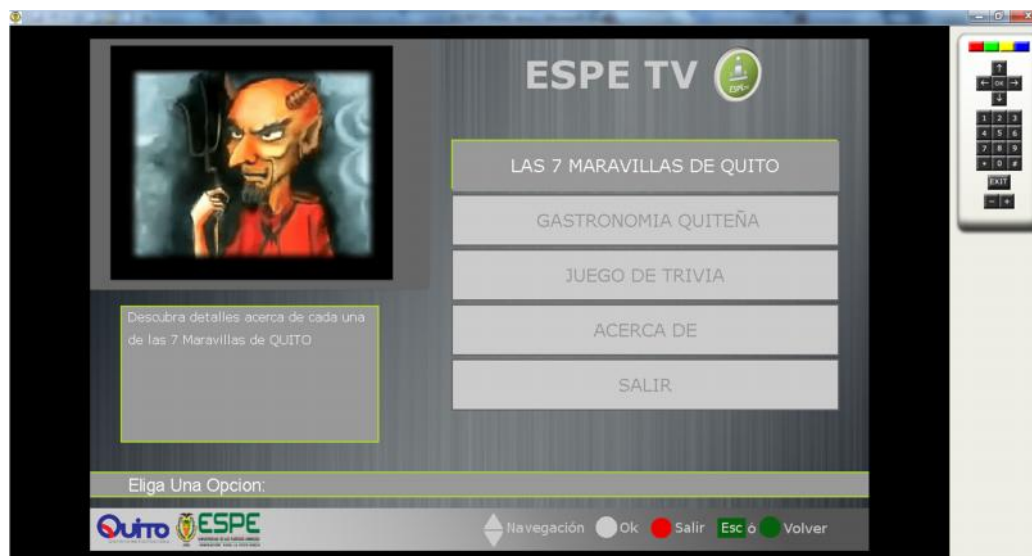


Figura 27. Menú Principal de la Aplicación

Con las flechas de navegación podemos hasta las opciones del menú, se presentan 5 opciones, la opción “Las 7 Maravillas de Quito” nos llevará a un submenú donde nos presentará cada una de los lugares turísticos; la opción “Gastronomía Quiteña” nos llevara de igual manera a un submenú donde se presentaran 5 restaurantes representativos de la capital; la opción “Juego de Trivia” nos llevara a un juego en donde se presentaran preguntas de las opciones acerca de las 2 opciones anteriores; “Acerca De” nos da información sobre el creador de esta aplicación y finalmente “Salir” nos permite volver a la Televisión.

Presionando el botón OK accederemos la opción “Las 7 Maravillas de Quito” y se presenta el siguiente submenú.



Figura 28. Sub Menú Las 7 Maravillas de Quito

En este submenú podremos elegir uno los 7 lugares turísticos propuestos para acceder a la información y a las fotografías de este sitio.



Figura 29. Información Sitio Turístico

Como se observa en la figura anterior dentro de cada sitio turístico podremos encontrar información relevante del sitio turístico, así como fotografías del mismo.

Ahora presionando el botón ESC o el Botón Verde del control remoto podremos retroceder a pantallas anteriores.



Figura 30. Sub Menú Gastronomía Quiteña

Ahora ingresamos al Sub Menú Gastronomía Quiteña, en donde se presentaran 4 restaurantes representativos de la capital, y podremos acceder a cada uno de ellos.

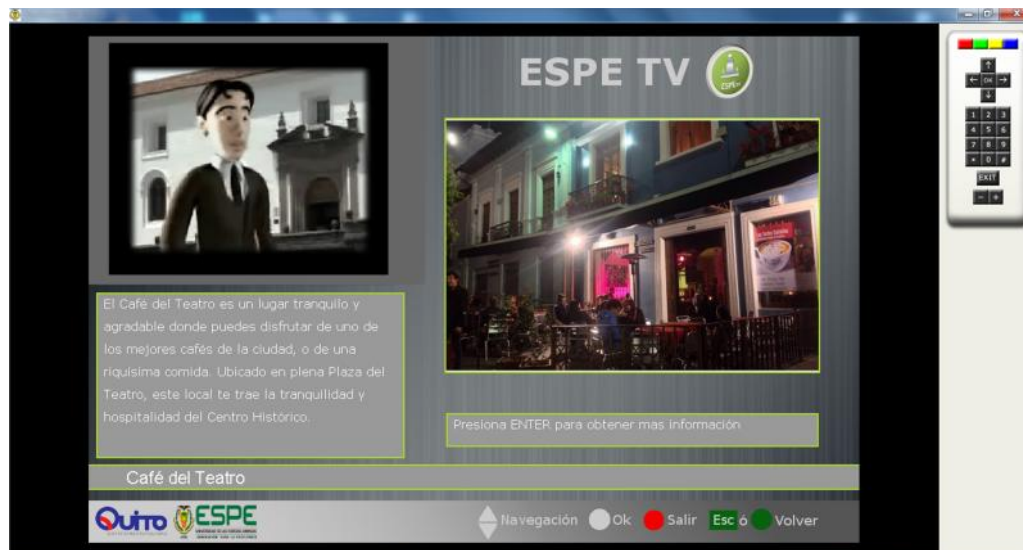


Figura 31. Información Restaurante Café del Teatro

Como podemos observar en la figura anterior en cada opción de cada restaurante podremos encontrar información relevante del restaurante como horarios de apertura, localización, etc.

Ahora accederemos al sub menú de Juego de Trivia, como se muestra a continuación



Figura 32. Sub Menú Juego de Trivia

En este juego se presentaran 5 preguntas de cultura general y se podrá elegir la respuesta correcta de entre 4 opciones presentadas en pantalla, una vez seleccionada la opción con el botón OK, se desplegara si la respuesta fue correcta o incorrecta y se sumara o no un punto; al terminar las 5 preguntas se despliega una pantalla que nos indica la puntuación final.



Figura 33. Pantalla de Puntuación

En esta pantalla se presenta la opción de volver a jugar o regresar al Menú Principal.

Y finalmente para poder salir de la aplicación interactiva tenemos la posibilidad de hacerlo en el Menú Principal con la opción Salir, o también con el botón Rojo del control remoto.

3.2.8. IMPLEMENTACIÓN JAVA TV

Como pudimos conocer en el capítulo 2, tanto el estándar europeo DVB - T como el estándar brasileño ISDB-Tb, utilizan el Api de Java TV para el desarrollo de aplicaciones interactivas.

DVB junto a su middleware MHP definió el uso de la tecnología Java como base de los APIs para la creación de aplicaciones interactivas; en este estándar teóricamente se buscó que las aplicaciones puedan ser transmitidas sobre cualquier tipo de red,

independientemente del proveedor de servicios o del receptor de televisión utilizado. MHP soporta la ejecución de aplicaciones locales, así como aplicaciones que soporten la utilización de un canal de retorno, y que posibiliten el acceso a diferentes servicios punto a punto de Internet.

Como ya fue tratado en apartados anteriores, nuestro país va a sufrir el cambio de televisión analógica a televisión digital, acogiendo el estándar brasileño ISDB-Tb y su middleware Ginga, el cual nos provee la posibilidad de ejecutar aplicaciones basadas en lenguaje Java mediante el uso de Ginga J. Este middleware de desarrollo procedimental para aplicaciones de Televisión Digital, posee tres características muy importantes en su implementación [12]:

- Multi – Red.- Ginga J es compatible con todos los medios que trabajan con Televisión Digital (cable, terrestre, satélite entre otros).
- Multi – Sistema.- Ginga J incluye una tabla de procesamiento del servicio de información de DVB, ATSC y ARIB, utilizando este API de servicio de información como salida de datos.
- Compatibilidad.- debido a la implementación del API Verde en Ginga J, este middleware posee la capacidad de ejecutar todas las aplicaciones desarrolladas en otros middleware de GEM.

Con esto podemos observar que las aplicaciones desarrolladas con el API de Java TV van a poder ser fácilmente implementadas con el estándar aceptado en nuestro país para televisión digital.

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

La Televisión Digital Interactiva abre un nuevo canal de información a la sociedad, disminuyendo la brecha digital existente ya que posibilita un acercamiento a las nuevas tecnologías a un sector de la población que no se encuentra familiarizado con los ordenadores y el acceso a información a través de Internet, la facilidad del uso de esta nueva herramienta conlleva que tenga una gran acogida.

Este proyecto presento un prototipo de aplicación interactiva para Televisión Digital, y se logró analizar una nueva alternativa para la programación de las mismas, en una plataforma tan ampliamente difundida y usada como es Java a través de su API Java TV.

Java TV se puede adaptar a nuestro entorno de Televisión Digital ya que como se observó en el desarrollo de este proyecto, nos provee de un conjunto de características para todos los receptores de Televisión Digital que puedan contar con la máquina virtual de Java. Además en el transcurso de la investigación se observó que Java TV se encuentra incluido como uno de los API principales dentro de la arquitectura de GINGA J que funcionarían próximamente en el Estándar adoptado por nuestro país para la transmisión de Televisión Digital (ISDb-T).

NetBeans es un IDE poderoso y que por medio de la inclusión de las librerías de JavaTV permitió el desarrollo del prototipo de aplicación. Existe cierta dificultad en la

simulación de las aplicaciones en el emulador XletView ya que aun en su versión más actualizada no posee aun implementadas todas las librerías del API Java TV.

El desarrollo de una aplicación utilizando el API de Java TV conlleva conocer la arquitectura de un Xlet, el cual nos va a permitir el almacenamiento y despliegue de los componentes gráficos de las aplicaciones.

La información presentada en este proyecto facilita el estudio y comprensión de diversos conceptos que a futuro podrán ser utilizados para el desarrollo de futuras aplicaciones interactivas mediante el uso de Java TV.

4.2. RECOMENDACIONES

Es necesario profundizar los conocimientos en el middleware de GINGA J para el desarrollo de aplicaciones interactivas mediante la utilización del lenguaje JAVA, ya que nuestro país está próximo a adoptar el nuevo estándar de televisión y el desarrollo de este tipo de aplicaciones interactivas será beneficioso para la población como ya se mencionó anteriormente.

Debido a las dificultades existentes en el uso de XletView como emulador para pruebas de las aplicaciones, se ve como alternativa el uso del emulador GINGA J para poder realizar las pruebas pertinentes.

Referencias Bibliográficas

- [1] «Televisión Digital Terrestre en el Ecuador,» 22 Marzo 2012. [En línea]. Available: <http://www.telecomunicaciones.gob.ec/television-digital-terrestre-en-el-ecuador/>. [Último acceso: 5 enero 2015].
- [2] J. M. Villabuena y C. Diaz Velazquez, *Informe Preliminar: Estado del Arte de Receptores Set-Top-Box - Aplicaciones*, 2010.
- [3] Supertel, «Estadísticas de Radiodifusión y Televisión abierta y pagada,» [En línea]. Available: <http://www.supertel.gob.ec/index.php/Estadisticas>. [Último acceso: 05 enero 2015].
- [4] E. Molina Muñoz, *PLATAFORMA TDT INTERACTIVA. SERVICIOS MUNICIPALES PARA LA PROVINCIA DE SEVILLA*, Sevilla, 2010.
- [5] *Programación Extrema*, Cali, 2011.
- [6] «Programación extrema: “Metodología para desarrollo ágil de aplicaciones”,» 11 Junio 2012. [En línea]. Available: http://www.uv.mx/universo/486/infgral/infgral_15.html. [Último acceso: 23 marzo 2015].
- [7] L. M. Garcés Rodríguez y D. A. Rovira Sánchez, *Prototipo de Aplicación de Telemedicina Basado en el Estandar DVB-MHP para la Televisión Digital Terrestre Interactiva*, Bucaramanga, 2009.
- [8] P. T. Galabay Toalongo y F. R. Vivar Espinoza, *Manejo del software Ginga para el desarrollo de aplicaciones interactivas para televisión digital, basdo en el estándar Brasileño ISDB-Tb*, Cuenca, 2012.
- [9] «Interactividad,» Diciembre 2014. [En línea]. Available: <http://www.televisiondigital.gob.es/tecnologias/Interactividad/Paginas/interactividad.aspx>. [Último acceso: 17 febrero 2015].
- [10] J. E. Torres Altamirano, *Diseño y Desarrollo de una aplicación de contenidos interactivos par TV Digital basada en el Middleware Ginga del Sistema Brasileño*,

Sangolquí, 2010.

- [11 «Televisores otros : Definicion de ATSC,» Marzo 2012. [En línea]. Available:
] <http://skp.samsungcsportal.com/integrated/popup/FaqDetailPopupMobile.jsp?cdsite=latin&seq=143913>. [Último acceso: 21 enero 2015].
- [12 Á. L. Quingaluisa Quispe y J. A. Torres Beltrán, *Estudio e Investigación del Middleware Ginga J del estándar Brasileño de Televisión Digital. Caso Práctivo: Desarrollo de una aplicación interactiva aplicando metodología OpenUp/Basic como parte del proyecto ESPE-GINGA*, Sangolquí, 2011.
- [13 M. F. Guerrero Hernandez y K. I. Ramirez Ariza, *MHP, Multimedia Home Platform, el camino hacia una alternativa estándar de Middleware en Redes IPTV*, Bucaramanga, 2010.
- [14 A. Martinez P, «MHP: UNA OPORTUNIDAD PARA LA CONVERGENCIA TV-INTERNET,» Junio 2005. [En línea]. Available:
] http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1316-48212005000200007&lang=es. [Último acceso: 7 enero 2015].
- [15 S. Morris y A. Smith Chaigneau, *Interactive TV Standars, A guide to MHP, OCAP and Java TV*, Burlington: Elsevier, 2005.
- [16 «Entorno de Desarrollo Integrado (IDE),» 25 Enero 2013. [En línea]. Available:
] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>. [Último acceso: 13 Marzo 2015].
- [17 R. Sanz de Acedo, «Herramientas Desarrollo MHP para TDT,» Mayo 2008. [En línea]. Available: <http://es.scribd.com/doc/99109337/Herramientas-Desarrollo-MHP-para-TDT#scribd>. [Último acceso: 5 febrero 2015].
- [18 «Norma ISO-9126 para análisis de software,» Septiembre 2014. [En línea]. Available:
] http://www.cuatroorios.org/index.php?option=com_content&view=article&id=163:norma-iso-9126-para-an%C3%A1lisis-de-software&catid=39:blogsfeeds. [Último acceso: 20 marzo 2015].
- [19 «Televisión Digital Terrestre en el Ecuador,» 22 Marzo 2015. [En línea]. Available:
] <http://www.telecomunicaciones.gob.ec/television-digital-terrestre-en-el-ecuador/>.

- [20 Supertel, «Estadísticas de Radiodifusión y Televisión abierta y pagada,» [En línea].
] Available: <http://www.supertel.gob.ec/index.php/Estadisticas>.
- [21 «Programación extrema: “Metodología para desarrollo ágil de aplicaciones”,» 11
] Junio 2012. [En línea]. Available:
http://www.uv.mx/universo/486/infgral/infgral_15.html.
- [22 «Interactividad,» Diciembre 2014. [En línea]. Available:
] <http://www.televisiondigital.gob.es/tecnologias/Interactividad/Paginas/interactividad.aspx>.
- [23 «Televisores otros : Definicion de ATSC,» Marzo 2012. [En línea]. Available:
] <http://skp.samsungsportal.com/integrated/popup/FaqDetailPopupMobile.jsp?cdsite=latin&seq=143913>.
- [24 A. Martinez P, «MHP: UNA OPORTUNIDAD PARA LA CONVERGENCIA TV-
] INTERNET,» Junio 2005. [En línea]. Available:
http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1316-48212005000200007&lang=es.
- [25 «Entorno de Desarrollo Integrado (IDE),» 25 Enero 2013. [En línea]. Available:
] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.
- [26 R. Sanz de Acedo, «Herramientas Desarrollo MHP para TDT,» Mayo 2008. [En
] línea]. Available: <http://es.scribd.com/doc/99109337/Herramientas-Desarrollo-MHP-para-TDT#scribd>.
- [27 «Norma ISO-9126 para análisis de software,» Septiembre 2014. [En línea].
] Available:
http://www.cuatrorios.org/index.php?option=com_content&view=article&id=163:norma-iso-9126-para-an%C3%A1lisis-de-software&catid=39:blogsfeeds.