

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERÍA

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE  
CONTENIDOS INTERACTIVOS PARA TV DIGITAL BASADA  
EN EL MIDDLEWARE GINGA DEL SISTEMA BRASILEÑO

JAVIER EDUARDO TORRES ALTAMIRANO

Sangolquí – Ecuador

2010

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado titulado: “DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE CONTENIDOS INTERACTIVOS PARA TV DIGITAL BASADA EN EL MIDDLEWARE GINGA, DEL SISTEMA BRASILEÑO”, ha sido desarrollado en su totalidad por el señor JAVIER EDUARDO TORRES ALTAMIRANO con CI: 180381629-5, bajo nuestra dirección.

Atentamente

---

Ing. Gonzalo Olmedo

DIRECTOR

---

Ing. Román Lara

CODIRECTOR

## RESUMEN

El presente proyecto de grado para la obtención del título de ingeniería, se centra en el estudio general del diseño y desarrollo de contenidos interactivos utilizando el middleware Ginga, mediante el uso de su motor de presentación denominado Ginga-NCL, cuyo lenguaje de desarrollo se denomina NCL; para lo cual previamente se realiza un análisis minucioso de los diferentes servicios de interactividad que se pueden brindar a través de la TV digital, así como de la arquitectura de referencia del middleware propuesta por las universidades brasileras PUC-Rio de Janeiro y la Universidad Federal de Paraíba.

El estudio realizado forma parte del proyecto para la implementación del Laboratorio de TV Digital Interactiva del Departamento de Eléctrica y Electrónica de la ESPE, además que sirve como precedente para futuras investigaciones y desarrollos en este campo, en beneficio no solo del país sino además de la comunidad Ginga latinoamericana de la cual forma parte la ESPE.

Como resultado de este proyecto se logró la obtención del primer programa interactivo de la Escuela, denominado INFO-ESPE el cual fue desarrollado en lenguaje NCL y en cuya estructura se implementa la utilización de una manera didáctica de los principales elementos del lenguaje. Además a través de este proyecto se logró posesionar a la ESPE como precursor en el desarrollo de la TV interactiva en el país, así como un miembro activo de la comunidad latinoamericana en este campo.

## DEDICATORIA

Este proyecto de tesis me gustaría dedicarlo principalmente a mis padres: Luis Eduardo y Lourdes del Rocio: a ti padre por ser siempre mi ejemplo y mi razón de superación, por haberme enseñado a jamás conformarme y siempre buscar más, por tu tenacidad y tu apoyo incondicional en cada etapa de mi vida; a ti madre por haberme dado la vida y darle sentido a cada uno de mis pasos, por enseñarme con tu ejemplo y abnegada labor a vencer cualquier obstáculo; a los dos por su amor infinito, muchas gracias, espero nunca haberlos defraudado.

A mis amigos del alma, que me han acompañado a lo largo de todo este camino, dándome sus consejos y extendiéndome siempre su mano, por ser mi soporte cuando más lo necesitaba.

A mis compañeros de aula, con los que compartimos momentos gratos al igual que difíciles, con quienes sufría o festejaba nuestros logros.

## AGRADECIMIENTOS

Agradezco a Dios por darme salud y vida para terminar con éxito mi carrera profesional, a mis padres su apoyo incondicional y amor infinito. De igual manera a todos mis amigos por acompañarme a lo largo de este trajinar que llamamos vida.

Por otra parte también agradezco muy sinceramente al Ing. Gonzalo Olmedo y al Ing. Román Lara, que mas allá de haberme guiado durante la elaboración de este proyecto y haber dedicado gran parte de su tiempo, de una manera desinteresada han sabido brindarme su amistad.

Finalmente agradezco a todos aquellos docentes que compartieron su conocimiento conmigo dentro y fuera de las aulas destacando de una manera especial al Ing. Freddy Acosta, al Ing. Julio Larco y al Ing. Paul Bernal, por ser más que una guía, grandes amigos.

En fin son muchas las personas especiales a las que me gustaría agradecer su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar en dónde estén quiero darles las gracias por formar parte de mí, por todo lo que me ha brindado y por todas sus bendiciones.

A todos ellos,

Muchas gracias de corazón.

# ÍNDICE

## CAPÍTULO I

INTRODUCCIÓN .....	1
1.1. Introducción General .....	1
1.1. Alcance .....	2
1.2. Objetivos .....	3
1.4. Organización del Documento .....	3

## CAPÍTULO II

INTERACTIVIDAD EN TV DIGITAL .....	5
2.1. Interactividad .....	5
2.1.1. Definición de Interactividad .....	5
2.1.2. Niveles de Interactividad .....	6
2.1.3. Televisión Digital Interactiva .....	6
2.1.3.1. Infraestructura de Red .....	7
2.1.3.2. Infraestructura de Software .....	8
2.2. Servicios Interactivos [4]. .....	9
2.2.1. Servicios Interactivos Autónomos (SIA) .....	9
2.2.1.1. Servicios de información: .....	10
2.2.1.2. Participación .....	11
2.2.1.3. Guías Electrónicas de Programas .....	11
2.2.1.4. Administración electrónica .....	11
2.2.1.5. Educación .....	11
2.2.1.6. Comercio electrónico .....	11
2.2.1.7. Telebanca .....	12
2.2.1.8. Juegos .....	12
2.2.1.9. Apuestas .....	12

2.2.1.10. Diversos .....	13
2.2.2. Servicios Interactivos Asociados a Programas (SIAP) .....	13
2.2.2.1. Servicios de información: .....	13
2.2.2.2. Participación.....	14
2.2.2.3. Educación .....	14
2.2.2.4. Comercio electrónico .....	14
2.2.2.5. Juegos.....	15
2.2.2.6. Apuestas .....	15
2.2.3. Programas Audiovisuales Interactivos .....	15
2.2.3.1. Video On Demand (VOD).....	16
2.2.3.2. Interacción Implicativa.....	16
2.2.3.3. Selección de ángulos de cámara .....	17
2.2.3.4. Multistream.....	17
2.2.3.5. Competencia .....	18
2.2.3.6. Hipervínculos .....	18
2.3. Middlewares para interactividad con TV Digital.....	18
2.3.1. Middleware .....	18
2.3.1.1. Multimedia Home Platform (MHP).....	20
2.3.1.2. Association of Radio Industries Businesses (ARIB) .....	23
2.3.1.3. DTV Applications <i>Software</i> Environment (DASE). .....	25
2.3.1.4. Advanced Common Application Platform (ACAP).....	26
2.3.1.5. Globally Executable MHP (GEM). .....	28

### **CAPÍTULO III**

ARQUITECTURA DEL MIDDLEWARE GINGA .....	31
3.1. Introducción.....	31
3.2. Arquitectura del Middleware Ginga .....	32
3.2.1. Ginga – NCL [15].....	38

3.2.1.1. El Lenguaje NCL .....	38
3.2.1.2. Objetos Multimedia NCL.....	41
3.2.1.3. Visión global de los elementos del Lenguaje NCL.....	41
3.2.1.4. Comandos de Edición Ginga-NCL.....	45
3.2.1.5. Núcleo Común de Procesamiento de Datos.....	46
3.2.1.6. Lua: El lenguaje de escritura de NCL .....	49
3.2.2. Ginga – J [16] .....	50
3.2.2.1. Arquitectura de Ginga-J.....	52
3.2.2.2. Especificaciones de Ginga-J .....	53
3.2.2.3. El API Ginga-J .....	53
3.2.2.4. Implementación de Referencia de Ginga-J.....	55
3.2.2.4.1. Componentes de Software de Ginga-J .....	55
3.2.2.4.1.1. Componentes de Acceso de flujo de bajo nivel.....	57
3.2.2.4.1.2. Componentes del procesamiento de flujos elementales ..	57
3.2.2.4.1.3. Componentes de interfaz de usuario .....	58
3.2.2.4.1.4. Componentes de Comunicación .....	58
3.2.2.4.1.5. Componentes de Gestión .....	59
3.2.2.4.1.6. Componentes de Persistencia .....	59
3.2.2.4.1.7. Componente de Acceso Condicional .....	60
3.2.3. El Puente entre Ginga-NCL y Ginga-J.....	60
3.2.4. Núcleo Común de Ginga (Comon-Core) [11].....	60

## **CAPÍTULO IV**

LENGUAJE NCL .....	63
4.1. Introducción al NCL.....	63
4.2. Estructura de un Documento Hipermedia .....	65
4.2.1. ¿Qué reproducir?.....	65
4.2.2. ¿Dónde reproducir? .....	66



4.2.3. ¿Cómo reproducir? .....	67
4.2.4. ¿Cuándo reproducir? .....	68
4.3. Estructura de un Documento NCL .....	70
4.3.1. Regiones.....	72
4.3.2. Descriptores.....	75
4.3.3. Nodo Multimedia (o Nodo de Contenido).....	79
4.3.4. Contextos.....	81
4.3.5. Puertas .....	82
4.3.6. Base de Conectores .....	83
4.3.7. Conectores .....	84
4.3.8. Enlaces .....	87
4.3.9. Anclas .....	88
4.3.10. Reglas.....	90
4.3.11. Switch .....	91
4.4. Herramientas.....	92
4.4.1. Composer .....	92
4.4.1.1. Visión Estructural.....	93
4.4.1.2. Visión Temporal.....	94
4.4.1.3. Visión de Diseño.....	94
4.4.1.4. Visión Textual .....	95
4.4.2. Emulador Ginga-NCL.....	95
4.4.3. Set Top Box Virtual .....	96
4.5. Desarrollo de la Aplicación NCL.....	97
4.5.1. Aplicación NCL INFO-ESPE .....	98
4.5.1.1. Requisitos y Modelamiento de la Aplicación.....	98
4.5.1.2. Programación y Pruebas de la Aplicación .....	100
4.5.1.3. Implementación .....	100

4.5.1.3.1. Diagrama NCM de la Aplicación INFO-ESPE .....	101
4.5.1.4. Resultados.....	105

### **CONCLUSIONES Y RECOMENDACIONES**

CONCLUSIONES.....	108
RECOMENDACIONES .....	112

### **REFERENCIAS**

REFERENCIAS BIBLIOGRÁFICAS .....	114
----------------------------------	-----

## ÍNDICE DE TABLAS

Tabla 4.1. Estructura Básica de un documento NCL.....	71
Tabla 4.2. Paramentáros que pueden ser utilizados en descriptor, de acuerdo al archivo multimedia.....	79
Tabla 4.3. Tipos de archivo multimedia.....	80
Tabla 4.4. URI válidas.....	81
Tabla 4.5. Papeles predefinidos de condición.....	86
Tabla 4.6. Papeles predefinidos de acción.....	86

## ÍNDICE DE FIGURAS

Figura 2.1. Estructura general de un terminal de acceso.....	19
Figura 2.2. Arquitectura de un receptor MHP.....	21
Figura 2.3. Versiones de MHP.....	21
Figura 2.4. Perfiles DVB-MHP.....	22
Figura 2.5. Arquitectura ARIB.....	24
Figura 2.6. Arquitectura DASE.....	26
Figura 2.7. Arquitectura ACAP.....	28
Figura 2.8. Concepto de GEM.....	30
Figura 3.1 Arquitectura de la TV Digital, con las tecnologías utilizadas en cada capa.....	33
Figura 3.2. Normas de referencia del Sistema Brasileño de TV Digital.....	34
Figura 3.3. Arquitectura Ginga.....	36
Figura 3.4. Ginga Common Core.....	37
Figura 3.5. Ejemplo de la transmisión de un Documento NCL.....	48
Figura 3.6. Contexto de Ginga-J.....	51
Figura 3.7. Arquitectura y ambiente de ejecución de Ginga-J.....	52
Figura 3.8. APIs de Ginga-J (Azul, Amarillo y Verde).....	54
Figura 4.1. Nodos e enlaces de un documento hipermedia común.....	63
Figura 4.2. Nodos, enlaces y nodos de composición.....	64
Figura 4.3. Representación de Archivos Multimedia.....	65
Figura 4.4. Representación de nodos multimedia y su composición.....	66
Figura 4.5. Representación de una Región.....	67

Figura 4.6. Representación de un descriptor asociado a una región.....	68
Figura 4.7. Representación de un descriptor asociado a un archivo multimedia que será presentado en una región.....	68
Figura 4.8. Puerta de un nodo de composición.....	69
Figura 4.9. Atributos de posición y tamaño de una región.....	74
Figura 4.10. Puerta plnicio como punto de entrada a un nodo interno de un contexto.....	82
Figura 4.11. Ilustración de un conector con tres nodos.....	84
Figura 4.12. Ilustración de un conector causal (elemento <causalConnector>) con papeles (role) de condición y acción.....	84
Figura 4.13. Ilustración de un enlace (elemento <link>).....	85
Figura 4.14. Máquina de estados de eventos.....	87
Figura 4.15. Herramienta de autoría Composer.....	93
Figura 4.15. Visión Estructural.....	93
Figura 4.16. Visión Temporal.....	94
Figura 4.17. Visión de Diseño.....	94
Figura 4.18. Visión Textual.....	95
Figura 4.19. Emulador GINGA-NCL.....	96
Figura 4.20. Set-Top Box Virtual.....	97
Figura 4.21. Ciclo de vida en cascada de la Aplicación implementada.....	98
Figura 4.22. Diagrama NCM de la Aplicación Principal.....	101
Figura 4.23. Diagrama NCM del Contexto General de las 10 opciones del submenú “Departamentos”.....	102
Figura 4.24. Diagrama NCM del Contexto del submenú “Admisión y Registro”.....	103
Figura 4.25. Diagrama NCM del Contexto del submenú “Sedes”.....	104
Figura 4.26. Primera opción de interacción a través del botón “INFO”.....	105

Figura 4.27. Menú Principal de Opciones.....	105
Figura 4.28. Pantalla principal, después de la selección la opción “ADMISION Y REGISTRO”.....	106
Figura 4.29. Submenú desplegado, después de la selección la opción “DEPARTAMENTOS”.....	106
Figura 4.30. Pantalla principal, después de la selección la opción “ELECTRICA Y ELECTRONICA” del submenú “DEPARTAMENTOS”.....	107
Figura 4.31. Pantalla principal, después de la selección la opción “SEDES”.....	107

## GLOSARIO

### A

**ATSC:** Advanced Television System Committee

**ARIB:** Association of Radio Industries Businesses

**ACAP:** Advanced Common Application Platform

**API:** Application Programming Interface

**ADSL:** Línea de Abonado Digital Asimétrica

### B

**BML:** Broadcast Markup Language

**BDTV:** Basic Digital TV Profile

### C

**CMS:** Content Management System

### D

**DTV:** Digital Television

**DVB:** Digital Video Broadcasting

**DAE:** Declarative Application Environment

**DASE:** DTV Applications *Software* Environment

**DSM-CC:** Digital Storage Media – Command and Control

### E

**EDTV:** Enhanced Digital TV Profile

## **G**

**GPRS:** Servicio General de Paquetes vía Radio

**GEM:** Globally Executable MHP

## **I**

**IPTV:** Internet Protocol Television

**ISDB-Tb:** International System for Digital Broadcast, Terrestrial, Brazilian version

## **J**

**JMF:** Java Media Framework

**JVM:** Java Virtual Machine

## **L**

**LMS:** Learning Management System

## **M**

**MHP:** Multimedia Home Platform

## **N**

**NCL:** Nested Conext Language

**NCM:** Nested Context Model

## **O**

**OCAP:** Open Cable Application Platform

## **P**

**PAI:** Programas Audiovisuales Interactivos.

**PAE:** Procedural Application Environment

**PDA:** Personal Digital Assistant

**PSTN:** Red Telefónica Pública Conmutada



## **S**

**STB:** Set Top Box

**SIA:** Servicios Interactivos Autónomos

**SIAP:** Servicios Interactivos Asociados a Programas

**SMIL:** Synchronized Multimedia Integration Language

**SBTVD:** Sistema Brasileiro de Televisión Digital

## **T**

**TVI:** Televisión Interactiva

**TS:** Transport Stream

**TDT:** Televisión Digital Terrestre

## **V**

**VSTB:** Virtual Set Top Box

## **W**

**Wi-Fi:** Wireless Fidelity

**WiMax:** Worldwide Interoperability for Microwave Access

## **X**

**XML:** Extensible Markup Language

**XHTML:** The Extensible HyperText Markup Language

# **CAPÍTULO I**

## **INTRODUCCIÓN**

### **1.1. Introducción General**

Al estar a pocos años el apagón analógico mundial de la Televisión Terrestre y al ser este medio, la principal fuente de información, entretenimiento y cultura de la mayor parte de la población a nivel mundial, este está sufriendo un cambio significativo, la migración del sistema tradicional de TV analógica hacia un sistema completamente digital, cuyo desarrollo ha ido creciendo durante el transcurso de los años, apareciendo distintos estándares para la Televisión Digital Terrestre, cada uno con características especiales y acordes a las necesidades y condiciones geográficas de cada uno de sus países en donde fueron desarrollados; a la par del suceso de este evento, surgió la necesidad de la aparición de nuevos servicios a través de la Televisión, gracias a las facilidades técnicas y tecnológicas que nos brinda cada uno de dichos estándares.

La Televisión Digital nos trae con ella muchas innovaciones con relación a la Televisión Analógica, aparte de sus ventajas ya conocidas como la impermeabilidad de la señal digital a las interferencias del medio, la transmisión de video en alta definición, así como la posibilidad de mejorar el aprovechamiento del espectro radioeléctrico, este medio hoy permite la difusión de aplicaciones y servicios de interactividad capaces de proporcionar tecnológicamente opciones como: la consulta de cuentas bancarias, el voto en directo en los concursos, la navegación por Internet, las compras telemáticas o la realización de gestiones con

las administraciones públicas, lo que permiten ampliar el nicho de negocios a las Televisoras, al igual que a las empresas proveedoras de servicios de valor agregado, además de brindar a los usuarios finales nuevas opciones y aplicaciones para su uso y deleite durante la transmisión de sus programas favoritos. Otra de las características importantes de las aplicaciones de interactividad a través de la TV Digital es el despliegue en servicios de educación, información, salud entre otros varios, que permiten además de entretener a la teleaudiencia, ayudarle a resolver problemas de manera inmediata al alcance de su control remoto.

En el caso de nuestro país luego de haber realizados la respectivas pruebas técnicas por parte de la Superintendencia de Telecomunicaciones en (fecha) Abril de 2010, se adoptó oficialmente el estándar Brasileiro ISDB-Tb, el cual es una modificación del estándar Japonés de TV Digital ISDB-T. Entre dichas modificaciones están el uso de MPEG-4 en lugar de MPEG-2 para la codificación de video y la aparición del middleware Ginga como plataforma de desarrollo y presentación de contenidos interactivos, mostrando en su arquitectura de referencia dos paradigmas de programación para la generación de contenidos interactivos, uno denominado Ginga-J (de procedimiento) y el otro Ginga-NCL (declarativo) el cual es el foco central de estudio del presente trabajo para la generación de contenidos interactivos.

### **1.1. Alcance**

En el presente proyecto se estudiara las características y arquitectura del middleware Ginga y se desarrollara una aplicación de interactividad para la Escuela Politécnica del Ejército, utilizando documentos hipermedia escritos en lenguaje NCL a través de la herramienta de software libre Composer, la que fue desarrollada en el laboratorio de TV Digital de la Pontificia Universidad Católica de Rio. La presentación del resultado final de la aplicación será presentada con ayuda del Set Top Box virtual de desarrollo de Ginga NCL, el cual es uno de los principales subsistemas de este middleware y que proporciona una infraestructura para la presentación de aplicaciones de interactividad.

## **1.2. Objetivos**

Para cumplir con el alcance propuesto en el presente proyecto se han planteado los siguientes objetivos:

El objetivo general del presente proyecto para la obtención del título de ingeniería es diseñar una aplicación de contenidos interactivos para TV Digital basada en el middleware Ginga, del Sistema Brasileño ISDB-Tb.

Los objetivos específicos son:

- Definir en qué consiste la interactividad para TV Digital.
- Investigar los diferentes estándares de middleware de interactividad para TV Digital.
- Investigar la arquitectura del Middleware Ginga.
- Determinar los requerimientos, tanto de hardware como de software, para implementar la plataforma de desarrollo de contenidos interactivos.
- Estudiar cómo estructurar un documento NCL.
- Analizar y estudiar las distintas herramientas de software existentes para la elaboración de contenidos interactivos, que utilicen el lenguaje NCL.
- Desarrollar una aplicación basada en documentos hipermedia escritos en lenguaje NCL mediante la instalación del subsistema lógico Ginga-NCL.

## **1.4. Organización del Documento**

El presente documento se encuentra dividido en cuatro capítulos, los mismos que a continuación serán detallados:

El capítulo uno presenta una breve introducción del contenido y enfoque de este proyecto, definiendo en él, los objetivos planteados para el mismo, así como el alcance del proyecto.

El capítulo dos presenta definiciones de interactividad con televisión digital, así como los diferentes servicios interactivos que se pueden brindar con la misma. Posteriormente se realiza un estudio breve de las diferentes plataformas de middleware de interactividad que ofrecen los principales sistemas de TV Digital existentes en el mundo.

En el capítulo tres se realiza un análisis sistemático y un estudio profundo de la arquitectura de referencia del middleware Ginga propuestos por la Universidad Católica de Rio de Janeiro y la Universidad Federal de Paraíba, las cuales han dividido el middleware en dos subsistemas principales, el primero el entorno de presentación Ginga-NCL que compete a la parte declarativa del middleware y la segunda la arquitectura de la parte Java, llamada Ginga-J, que en conjunto permiten la presentación de aplicaciones interactivas.

Finalmente en el capítulo cuatro se presenta un estudio detallado del lenguaje declarativo denominado NCL, así como de sus diferentes elementos que en conjunto permiten la elaboración de contenidos interactivos utilizando el Motor de presentación o Ginga-NCL, de igual manera se presentan las herramientas de software utilizadas tanto para la elaboración como la presentación de contenidos interactivos, para finalmente mostrar como resultado de este estudio la aplicación interactiva desarrollada, denominada INFO-ESPE con sus respectivos diagramas NCM que permiten un mejor entendimiento de cómo fue desarrollado.

## **CAPÍTULO II**

### **INTERACTIVIDAD EN TV DIGITAL**

#### **2.1. Interactividad**

##### **2.1.1. Definición de Interactividad**

En el mundo actual y globalizado se ha llevado la definición de interactividad a varios planos de aplicación, especialmente en la rama de la informática y en todo lo que se relacione con el campo tecnológica y dependiendo de su autor este término puede ser empleado de diversas maneras que en conclusión nos lleva directamente a un concepto común, como es el grado de relación entre un usuario o actor y un sistema pudiendo ser este informático, video, entre otros.

Con una perspectiva técnica Bettetini (1995) [1], al definir la interactividad destaca las siguientes características:

- La pluridireccionalidad del deslizamiento de las informaciones;
- El papel activo del usuario en la selección de las informaciones requeridas;
- El particular ritmo de la comunicación.

Desde este punto de vista, la palabra “interactividad” se está convirtiendo en un comodín extra de reciente aparición, utilizado con gran frecuencia, dándole este adjetivo un valor añadido al producto que lo hace máspreciado, actual e innovador.

Ahora el grado de interactividad que posea una aplicación dependerá de dos puntos principalmente: por una parte, implica la capacidad técnica de conceder el máximo de posibilidades y opciones de comunicación entre el usuario y la máquina y, por otra, implica conseguir que el tiempo de respuesta sea reducido, en relación a las acciones realizadas por el usuario [1].

### **2.1.2. Niveles de Interactividad**

Los niveles de interactividad serán definidos dependiendo de las finalidades para las que se hayan desarrollado los materiales y contenidos, los cuales mantendrán una estrecha relación con el tipo y calidad de los recursos tecnológicos y de programación informática que se hayan empleado en su implementación.

Desde esta perspectiva, se podrá considerar que un programa cuyo fin solo sea el hacer una presentación-demostración, en la que el usuario únicamente actúe como espectador, su nivel de interactividad será muy bajo. En cambio, en un programa en el que el usuario, con su actuación, esté en capacidad de modificar el valor de las variables que intervienen en un determinado fenómeno y pueda ver como el programa se ajusta a los valores asignados, visualizando el resultado mediante la ejecución del proceso, el grado de interactividad será elevado.

Considerando estos dos extremos, la implementación de los materiales, la gama y tipología de interactividad podrá ser muy diversa, dependiendo directamente de los objetivos a cumplir que se hayan formulado los diseñadores [2].

### **2.1.3. Televisión Digital Interactiva**

El concepto de Televisión Interactiva no es un tema nuevo. Ya en los años 80's se introdujo el sistema de teletexto en la televisión analógica, el que ofrecía una interacción limitada con nuestro televisor, la cual nos aportaba información sobre la programación televisiva, el tiempo etc. Actualmente este concepto se lo ha llevado a un nivel mucho más alto con el desarrollado de una nueva y mejorada forma de interacción con el usuario. Sin duda alguna la Televisión Digital Interactiva sobrepasa la televisión analógica en capacidad, optimización del

espectro radioeléctrico, inmunidad al ruido o interferencias, una más alta calidad de imagen y sonido, posibilidad de transmisión de datos simultáneamente y finalmente un ahorro de potencia en la transmisión. Gracias a la digitalización del sistema televisivo, la expansión de las redes de telecomunicaciones y al éxito de las experiencias con la Internet de alta velocidad, las posibilidades interactivas de la televisión se ven multiplicadas y mejoradas con la implantación de un canal de retorno; permitiendo al usuario, antes mero espectador, hoy desempeñar un papel más activo en el proceso de comunicación viendo incrementadas sus posibilidades de participación.

La interactividad a través de la televisión digital permite que el usuario sea un ente activo de la programación proporcionándole herramientas similares a las que brinda la Internet, tales como: la posibilidad de consultar o extender la información presentada, combinar contenidos multimedia (audio, video, texto), participación en foros de opinión y además controlar de cierta manera la secuencia de la información presentada [3].

#### **2.1.3.1. Infraestructura de Red**

Una infraestructura de red que permita la operabilidad de un Sistema de TV interactiva posee un mínimo de componentes, los cuales deben ser seleccionados bajo un estudio minucioso de alternativas, cuya selección se debe basar en algunos criterios tales como: bajo costo, orientado al tipo de aplicación y no a la prueba de señales, para el caso específico de este estudio la utilización de equipos de desarrollo y no comerciales, *software* de fuente abierta, y el uso de emuladores de elementos *hardware* [3].

- **Servidor de Contenidos y Aplicaciones:** Es el encargado del almacenamiento y adecuación de los contenidos y aplicaciones. Este equipo se comunica con el servidor de televisión, con los equipos de producción/desarrollo y con los equipos terminales a través de una conexión adecuada al tipo de red a desarrollar.
- **Servidor de Televisión:** Se encarga de dar soporte a los servicios de *Playout* (difusión) y modulación. Lleva a cabo la generación del flujo de



transporte MPEG-2 a través de la multiplexación de datos de diferentes fuentes como: los flujos de audio/video, la información de programa/servicio y el sistema de archivos del carrusel de objetos, tareas que demandan gran cantidad de procesamiento y un funcionamiento continuo.

- **Set Top Box (STB):** Es el encargado de la recepción y decodificación de una señal digital de televisión para ser presentada en un televisor analógico y/o digital de un estándar diferente al del sistema de *broadcasting* implementado en cada región o país. Para soportar características de interactividad son dotados de un *middleware*, como MHP, ARIB, DASE, GEM o Ginga, los cuales serán definidos más adelante.

### 2.1.3.2. Infraestructura de Software

A continuación presentan los elementos de *software* mínimos necesarios para soportar los servicios de Televisión Digital Interactiva [3].

- **Servidor de Aplicaciones:** Es el encargado de desempeñar las funciones de almacenamiento y servicio de las aplicaciones interactivas a través del canal de difusión, al igual que se encarga de dar respuesta a todas las peticiones de los televidentes que pueden llegar a través del canal de retorno.
- **Servidor de Contenidos:** Este componente desempeña funciones simples de almacenamiento de contenidos al igual que funciones mucho más complejas como actuar como un Sistema de Gestión de Aprendizaje (LMS) o Sistema de Gestión de Contenidos (CMS) pero con las respectivas adaptaciones para manejar material audiovisual.
- **Aplicación de TV:** Es un *software* comercial que permite el manejo de la señal de televisión y la ejecución de aplicaciones interactivas.
- **Servidor de *Playout*:** Es el componente encargado de integrar la codificación del audio y video; la generación del carrusel de objetos,

datos y eventos; la generación de la información de programas, servicios y aplicaciones, y la multiplexación de todos estos parámetros para la generación de un único flujo de transporte MPEG-2 que posteriormente será modulado para su transmisión.

- **STB Virtual:** Con fines de realización de prueba previas, resulta de gran utilidad la implementación de una herramienta en *software* que permita la pre-visualización de los contenidos interactivos, para lo cual se ha seleccionado el STB Virtual de desarrollo de Ginga NCL, el cual es uno de los principales subsistemas de este *middleware* y que proporciona una infraestructura para la presentación de aplicaciones de interactividad.
- **Herramienta para la creación de aplicaciones interactivas:** Es el *software* que permite la creación de interfaces de interactividad permitiendo adicionar texto, gráficos, audio y video pregrabado. Para este estudio se utilizaran la herramienta Composer desarrollada en el laboratorio de TV Digital de la Pontificia Universidad Católica de Rio de Janeiro.

## 2.2. Servicios Interactivos [4].

Los contenidos ofrecidos por los diferentes sistemas que soportan la TV Interactiva pueden definirse por varios criterios siendo uno de los más claros el establecido por la Comunidad Europea, ya que actualmente su desarrollo en este campo es el más evidente y con mayor despliegue en todo su continente; criterio que clasifica estos servicio en tres grandes grupos: Servicios Interactivos Autónomos (SIA), Servicios Interactivos Asociados a Programas (SIAP) y Programas Audiovisuales Interactivos (PAI).

### 2.2.1. Servicios Interactivos Autónomos (SIA)

Se define como Servicios Interactivos Autónomos (SIA), a aquellos productos audiovisuales con entidad propia, siempre disponibles para el telespectador, que funcionan independientemente del flujo televisivo y cuya autoría puede ser del

operador titular de la TVI o de un proveedor externo. Se puede acceder a ellos directamente desde la aplicación básica de la interface del sistema o por *walled gardens*. Buena parte de los SIA necesita un canal de retorno y además manejan una estructura multimedia (domina el uso de textos, gráficos y fotografías al uso de vídeo y/o sonido).

En general, este tipo de sistemas poseen tres etapas de acceso:

- Universal (todos los usuarios);
- De abonados (usuarios de la plataforma), y;
- De pago por servicio (usuarios suscritos a un contenido específico).

Frecuentemente, el acceso a estos contenidos se encuentra limitado y relacionado a suscripción o pago por evento.

Con respecto al tipo de contenido, los SIA se pueden presentar como:

#### **2.2.1.1. Servicios de información:**

**Servicios informativos:** Contenidos relacionados con actualidad periodística que permiten la consulta de noticias a diferentes grados de profundidad (titulares o desarrollo de la información).

**Información Especializada:** Contemplan servicios universales de información financiera, estado de la bolsa, recetas de cocina, educación infantil, industria automovilística. En algunas plataformas coexiste el acceso gratuito a la información básica con una modalidad de pago para obtener más información.

**Información de Servicio:** Conjunto de aplicaciones consideradas como información de utilidad o de interés, como por ejemplo el informe meteorológico o el estado del tránsito.

### **2.2.1.2. Participación**

Prestan generalmente un servicio de recolección de resultados de encuestas que indican o denotan la opinión y/o tendencias del telespectador con respecto a un producto, actor o programa así como sugerencias para los responsables de los programas (temas a tratar, opiniones sobre los contenidos o escoger personajes populares). Su uso más frecuente se distingue en aplicaciones SIAP.

### **2.2.1.3. Guías Electrónicas de Programas**

Son aquellos servicios que brindan información complementaria sobre programas, autores y protagonistas o valoraciones críticas de especialistas. También incluyen funciones de selección para la grabación de programas.

### **2.2.1.4. Administración electrónica**

A pesar de que existen pocas aplicaciones que permiten realizar trámites administrativos en la TVI, estos servicios van orientados a permitir la realización de los mismos como por ejemplo pedir citas médicas, realizar consultas a una enfermera entre otros varios tipos de diligencias personales.

### **2.2.1.5. Educación**

A pesar que este tipo de contenidos han sido desarrollados mucho en la televisión, ese conocimiento no se ha aplicado a la producción de SIA de formación con una intensidad deseable. Sin embargo, un ejemplo de las producciones de este tipo son el conjunto de juegos, eminentemente educativos.

### **2.2.1.6. Comercio electrónico**

Este tipo de contenidos han despertado en los operadores una nueva forma de ingresos gracias a la estrecha relación existente entre el comercio electrónico y el potencial publicitario de la televisión, permitiendo hacer inserciones publicitarias de distintos formatos en los SIA. La publicidad muestra unos productos o servicios

e, inmediatamente, una aplicación de compra asociada al anuncio los puede vender.

#### **2.2.1.7. Telebanca**

Presenta aplicaciones desarrolladas para entidades financieras con ciertas limitaciones en relación a las que las que desarrollan para Internet, estas aplicaciones permiten establecer contactos de los bancos con sus clientes mediante un canal de retorno y una correcta identificación del usuario dentro del servicio.

#### **2.2.1.8. Juegos**

Son aplicaciones de juegos a un nivel mucho menor que los disponibles en las consolas de videojuegos actuales. Un ejemplo muy habitual son las versiones televisivas de juegos populares como *Tetris* y *Pacman*. En una gran parte estos juegos no desarrollan una conexión a través del canal de retorno y son aplicaciones de interacción tras la descarga local. Sólo versiones avanzadas, en general previo pago telefónico, permiten juegos en conexión con los servicios centrales de la plataforma, con los proveedores de contenido o con otros jugadores de la misma plataforma.

#### **2.2.1.9. Apuestas**

Son algunas aplicaciones de juego televisivas se vinculan a actividades deportivas que permiten las apuestas y otras aplicaciones, como las del servicio de loterías y casino, que permiten el juego electrónico.

### 2.2.1.10. Diversos

Son aplicaciones variadas en su naturaleza o contexto, y por su dispersión, resulta difícil y poco práctico encuadrarlas genéricamente, como por ejemplo: horóscopo, esoterismo y citas, entre otros.

## 2.2.2. Servicios Interactivos Asociados a Programas (SIAP)

Los SIAP son servicios asociados y sincronizados a un programa televisivo lineal, como un complemento y le añaden un valor agregado al mismo; estos servicios se encuentran disponibles únicamente mientras dura el programa televisivo al cual están asociados y aparece durante uno o varios de sus segmentos, dependiendo de cómo fueron diseñados. La gran mayoría de SIAP utilizan diferentes modalidades de inserción textual, gráfica, fotográfica, ventanas de vídeo o sonido y, en menor medida, complementos videográficos y multimedia que no pertenecen al flujo del programa. En una mayor medida estos, no necesitan de un canal de retorno.

### 2.2.2.1. Servicios de información:

**Servicios informativos:** Son aplicaciones asociadas a programas informativos de actualidad que permite a los telespectadores profundizar en la información mediante textos, gráficos, videografía e infografía.

**Información de servicio:** Son aplicaciones relacionadas a información meteorología y tránsito, que frecuentemente están desarrolladas como aplicaciones SIA, pero también pueden ser desarrolladas como SIAP en las televisiones que disponen de estas emisiones como programas con entidad propia.

**Información de complemento de programas:** Son aplicaciones que permiten desplegar información complementaria a uno o varios eventos televisados y su mayor desarrollo se lo ha logrado en aplicaciones de programación deportiva, en las que permiten al telespectador mirar estadísticas y otros datos asociados al

evento. Se presentan en dos modalidades: de programas periódicos como por ejemplo el campeonato regular de fútbol y de acontecimientos puntuales tales como olimpiadas.

#### **2.2.2.2. Participación**

Son aplicaciones que disponen al telespectador como un elemento activo de la programación, permitiéndole emitir su criterio; la participación pagada puede ser presentada en distintas modalidades y una de las más frecuentes es la de votación asociada a *reality games*; otra modalidad es la del chat integrado al programa, el mismo que puede, o no, moderarse y editarse para su salida al aire, principalmente se asocian estas aplicaciones a *info-shows* y telenoticias. Dentro de la modalidad de participación pagada, también se puede incluir el envío de mensajes SMS que aparecen en pantalla o que son leídos por los presentadores, aplicación muy común dentro nuestro medio televisivo.

#### **2.2.2.3. Educación**

Este tipo de aplicaciones están comúnmente asociadas a programas de divulgación y en ocasiones, contienen cuestionarios de respuesta múltiple permitiéndole al telespectador medir su nivel de conocimientos, dándole algún tipo de gratificación.

#### **2.2.2.4. Comercio electrónico**

Son aplicaciones asociadas a la publicidad interactiva, permitiéndole al usuario la navegación, teniendo como objetivo final la compra de un producto.

### **2.2.2.5. Juegos**

Este tipo de aplicaciones están destinadas en su mayoría a que el telespectador juegue y se deleite a medida el programa se desarrolla, están asociadas a concursos televisivos.

### **2.2.2.6. Apuestas**

De la misma forma que sucede con los SIA, estas aplicaciones tienen la finalidad de que los telespectadores realicen sus apuestas de acuerdo al evento deportivo que se está transmitiendo, como por ejemplo previo pago, el usuario puede predecir los resultados de un partido de fútbol para ganar dinero o premios.

## **2.2.3. Programas Audiovisuales Interactivos**

Los PAI son programas de televisión diseñados para permitir que a medida que estos se vayan desarrollando, uno o varios espectadores intervengan y su estructura vaya siendo modificada, en este tipo de programas vale diferenciar dos tipos: Programas Audiovisuales Interactivos de Flujo (PAIF) y Programas Audiovisuales Interactivos de Stock (PAIS).

**Programas Audiovisuales Interactivos de Flujo (PAIF):** Son programas que pueden ser transmitidos en directo o en diferido, de modo que las acciones que ejecuten los espectadores sean realizadas sincrónicamente a la emisión, es decir debe existir coherencia entre el tiempo del discurso y el tiempo de consumo audiovisual, y las acciones de los telespectadores no alteran la relación. Dependiendo de la complejidad de su arquitectura, estos programas permitirían mayor o menor intervención del receptor y la necesidad, o no, de canal de retorno. Están concebidos para permitir su consumo lineal en caso de que un telespectador se inhiba de la interacción.

**Programas Audiovisuales Interactivos de Stock (PAIS):** Al contrario de los PAIF, los PAIS son siempre accesibles y disponibles para que el espectador pueda interactuar en el momento que lo desee. En estos programas, el tiempo del



discurso varía por los ritmos de interacción que marca el usuario y por las opciones que toma. Siempre necesitan canal de retorno, aunque sea para acceder al servicio. El ancho de banda del canal de retorno, es que permite determinar el nivel de interacción que puede darse, la misma que puede realizarse tanto de forma remota con el servidor del emisor o localmente con el *set top box* del telespectador.

En la actualidad existen relativamente pocas producciones que permitan que algún tipo de interacción por parte del telespectador afecte al desarrollo del programa, esto sucede principalmente a la complejidad y costo que implica la producción de programas tipo PAI; sin embargo, no se podrá hablar de una autentica evolución de la TVI hasta cuando la televisión involucre en su totalidad este tipo de programas y simplemente estaremos frente a experiencias de interacción con la llamada *Enhanced tv*, y no frente a la TVI propiamente dicha.

#### **2.2.3.1. Video On Demand (VOD)**

Es uno de los primeros desarrollos dentro de lo que es la televisión interactiva, ya que permite al usuario de operadoras de cable, plataformas de televisión digital o servicios de banda ancha; seleccionar un programa de un repertorio variado y decidir sobre la dimensión temporal del visionado, sus acciones alteran, únicamente, los tiempos de consumo (detener la narración, retrocederla o adelantarla) pero no permiten alterar el contenido o cooperar en el desarrollo del argumento.

#### **2.2.3.2. Interacción Implicativa**

Se presenta en programas que permiten al espectador tener una alta experiencia subjetiva de interacción, aunque la interacción técnica sea superficial. Un ejemplo claro puede ser un programa concurso en el cual los telespectadores pueda competir entre sí, o con los participantes presentes en el estudio, todo esto en tiempo real y bajo las mismas condiciones.

### 2.2.3.3. Selección de ángulos de cámara

Permite al usuario interactuar con el programa, seleccionado una o varias imágenes del mismo programa, de modo que pueda adoptar el papel de realizador u otra de sus grandes aplicaciones es la selección de ángulos de cámara en transmisiones en directo de partidos de fútbol, béisbol o hockey, en carreras de automóviles y en conciertos.

### 2.2.3.4. *Multistream*

El telespectador tiene acceso a un número variable de flujos de vídeo que pueden ser complementarios o alternativos. Dentro de esta categoría se presentan las siguientes modalidades:

**Conmutación binaria:** El usuario tiene acceso a dos flujos de vídeo en pantalla, los que pueden estar dispuestos en distintas proporciones dentro de la misma y el telespectador puede observar una, o las dos videos a la vez.

**Focus:** El usuario tiene acceso a un flujo de vídeo alternativo al principal; pero en este caso uno de estos flujos corresponde a textos sobre el programa que consume para que el telespectador focalice la atención sobre un aspecto.

**Menú:** El usuario puede abrir un menú sobrepuesto a la imagen en uno de los extremos de pantalla y navegar con las teclas del control remoto, así como acceder a información textual que aparece en primer término, mientras el programa emitido ocupa un cuarto de la superficie de pantalla.

**Mosaico:** El telespectador tiene acceso a un mosaico de tres flujos de vídeo y está en capacidad de elegir cuál de ellos quiere observar a pantalla completa, además en la pantalla se presenta la imagen en movimiento correspondiente a la emisión de cada uno de ellos. Uno de los flujos de vídeo es para el programa en su emisión lineal, mientras que los dos restantes son versiones alternativas.

**Videowall:** Se le presenta al telespectador un repertorio de varios videos en los cuales el *videowall* funciona como interface visual y permite seleccionar los flujos de interés, el volumen de información y el orden de visionado.

**Sinergia:** El telespectador tiene acceso a cuatro flujos de vídeos que se pueden relacionar entre ellos con herramientas de selección y opciones de información textual complementaria o de votación si existe canal de retorno.

#### **2.2.3.5. Competencia**

El telespectador puede disfrutar de uno de dos flujos de vídeo disponibles no por su selección sino por su competencia en el asunto es decir su conocimiento. El programa es un juego entre dos *streams*, uno está dedicado a la respuesta correcta y el otro a la incorrecta.

#### **2.2.3.6. Hipervínculos**

Son aplicaciones de interactividad con un nivel de complejidad muy elevada y se articulan sobre una arquitectura de secuencias que operan como pre-sentidos y que se agregan a la secuencia de emisión por decisión del telespectador. El establecimiento previo de una lógica predeterminada garantiza que la agregación de determinadas partes sea capaz de generar sentido. Sin embargo, cada telespectador hará su recorrido y finalizará su obra sin condicionar el consumo de otros usuarios.

### **2.3. Middlewares para interactividad con TV Digital**

#### **2.3.1. Middleware**

Para poder ejecutar aplicaciones interactivas en TV Digital, es necesario el uso de un terminal de acceso conocido con el nombre de *Set Top Box* (STB), a través de cual, los usuarios pueden controlar y manejar dichas aplicaciones.

Sin embargo, este proceso es posible gracias a la capa de *software* implementada en este terminal, que permite abstraer la complejidad del *hardware*, en otras palabras aquí es donde los componentes son los responsables de la ejecución de aplicaciones, dibujarlos en la pantalla del televisor, gestionar los eventos capturados por ellos, y supervisar todas las etapas de su ciclo de vida.

El *middleware* es una capa intermedia o API (*Application Programming Interface*) genérico, que permite el acceso a las aplicaciones y servicios interactivos siempre que sea de la misma manera, independientemente de la plataforma de *hardware* o de *software* donde se están ejecutando como se muestra en la Figura 2.1 [5].

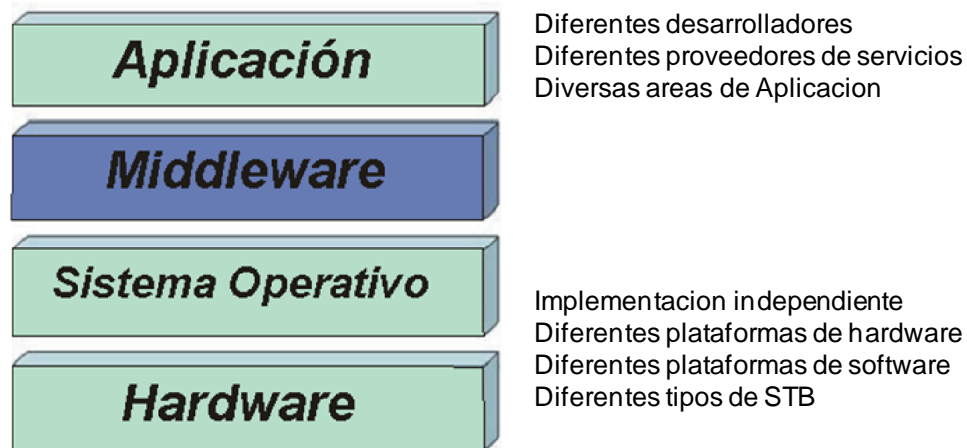


Figura 2.1. Estructura general de un terminal de acceso.

Actualmente en el mercado existen varios *middlewares*, pero entre ellos hay que destacar los cinco patrones de *middleware*, desarrollados en función del beneficio de la adopción de esta capa. En el caso de la norma europea DVB - *Digital Video Broadcasting Project* (DVB, 2003), utilizamos el MHP - *Multimedia Home Platform* desarrollado íntegramente en Java 6 (Juca, 2006). Para la norma de EE.UU. ATSC - *Advanced Television Systems Committee* (ATSC, 2006), el *middleware* líder es el DASE - *Software de aplicación para el Ambiente DTV* (ATSC, 2006). Para el caso de los japoneses ISDB - *Integrated Services Digital Broadcasting*, ARIB es - Asociación de Industrias de Radio y Negocios (ARIB, 1997) y finalmente los dos últimos desarrollados GEM - *Globally Executable MHP* que una derivación del MHP, basado en las especificaciones de *middleware* abierto desarrollado por el Proyecto DVB, siendo este un entorno de plataforma Independientes que se pueden adoptar a través de una serie de sistemas de entrega, incluyendo los medios de comunicación envasados y finalmente, Ginga el *Middleware* abierto del Sistema Brasileño de TV Digital (SBTVD, 2007), el que

está formado por un conjunto de tecnologías estandarizadas e innovaciones brasileñas que lo convierten en la especificación de *middleware* más avanzada [5].

### 2.3.1.1. Multimedia Home Platform (MHP).

MHP, o *Multimedia Home Platform*, es el nombre colectivo de un conjunto de especificaciones compatibles de *middleware* desarrolladas por el Proyecto DVB-MHP, fue diseñado para trabajar en todas las tecnologías de transmisión de DVB.

En una breve descripción, el entorno de ejecución de MHP, se basa en la utilización de una máquina virtual, es decir, las aplicaciones no serán desarrolladas en el código nativo del microprocesador, sino que serán clases Java independientes del *hardware*. Por tanto, una aplicación MHP (conocida como aplicación DVB-J o Xlet) va a ser el *bytecode* de un programa, escrito en Java y un conjunto de APIs, que serán interpretado por una máquina virtual Java que debe existir en el STB [6].

Por lo tanto se puede decir que el núcleo de la norma MHP está basado en una plataforma conocida como DVB-J, que incluye la máquina virtual Java (según la especificación de *Sun Microsystems*). Una entidad del *software* de sistema, denominada Gestor de Aplicaciones, la que se encarga de coordinar la ejecución de las aplicaciones y las comunicaciones con su entorno. Un conjunto de paquetes Java que proveen los interfaces entre las aplicaciones, las funciones de un receptor DVB y las redes de comunicación a las que está conectado. De igual forma, también se define en la norma, los formatos de los contenidos que el receptor debe poder gestionar, la torre de protocolos que debe implementar y la señalización adecuada para coordinar el correcto funcionamiento del conjunto (Figura 2.2) [7].

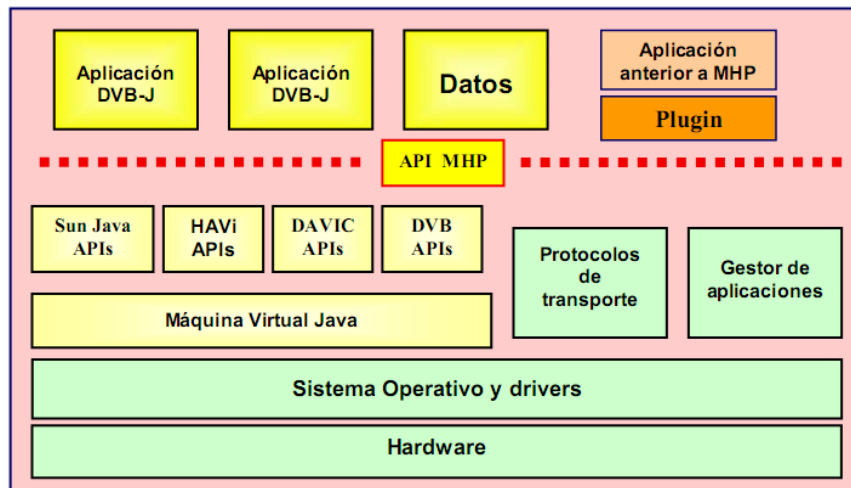


Figura 2.2. Arquitectura de un receptor MHP

Con el paso de los años y la aparición nuevas redes de banda ancha, se han desarrollado tres nuevas versiones de MHP, cada una de estas con nuevas características útiles para nuevos niveles y aplicaciones de interactividad. Estas versiones son MHP 1.0, MHP 1.1 y MHP 1.2, cuyas innovaciones se resumen en la figura 2.3:

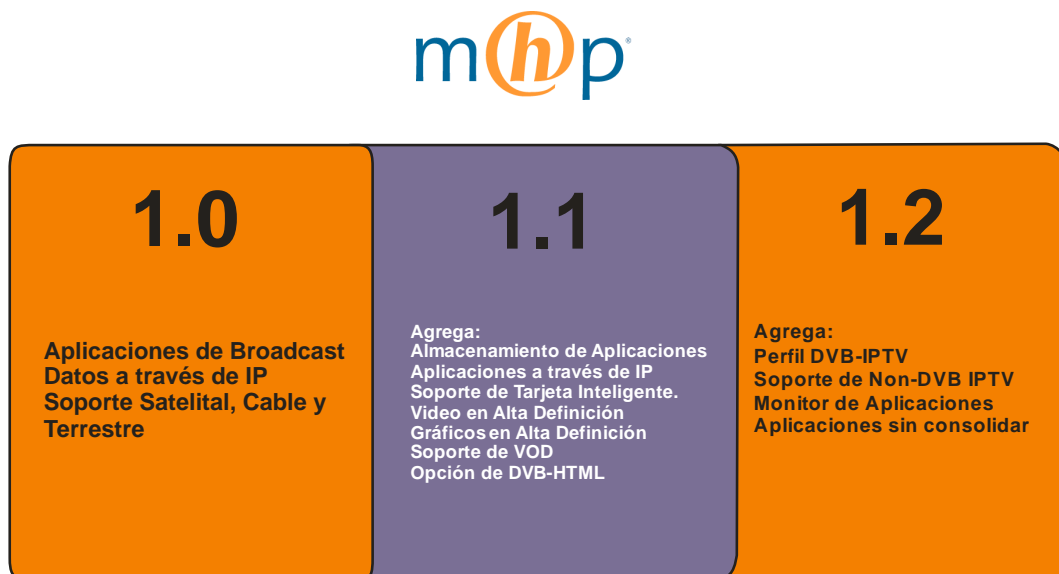


Figura 2.3. Versiones de MHP

Ya que no todas las versiones de MHP pueden soportar todas las áreas de aplicación, el sistema DVB introdujo el concepto de perfiles de soporte a la aplicación de las normas. Cada perfil se hace referencia a un área de aplicación específica y define los requisitos necesarios del STB. Estos perfiles son (Figura 2.4):

- **Enhanced Broadcast:** Este perfil requiere un STB, sin o con una capacidad de gestión limitada de canal de retorno.
- **Interactive Broadcast:** Este perfil permite el uso del canal de retorno (PSTN, ADSL, GPRS, Ethernet, etc.) para la ejecución de aplicaciones interactivas; además soporta la descarga de aplicaciones MHP a través del canal de retorno sólo en la versión 1.1, mientras que para los perfiles anteriores esto sólo es posible a través del canal de radiodifusión.
- **Internet Access:** Este perfil requiere un STB más complejo en términos de memoria y potencia de cálculo y permite la interactividad completa y el acceso a contenidos de Internet [8].

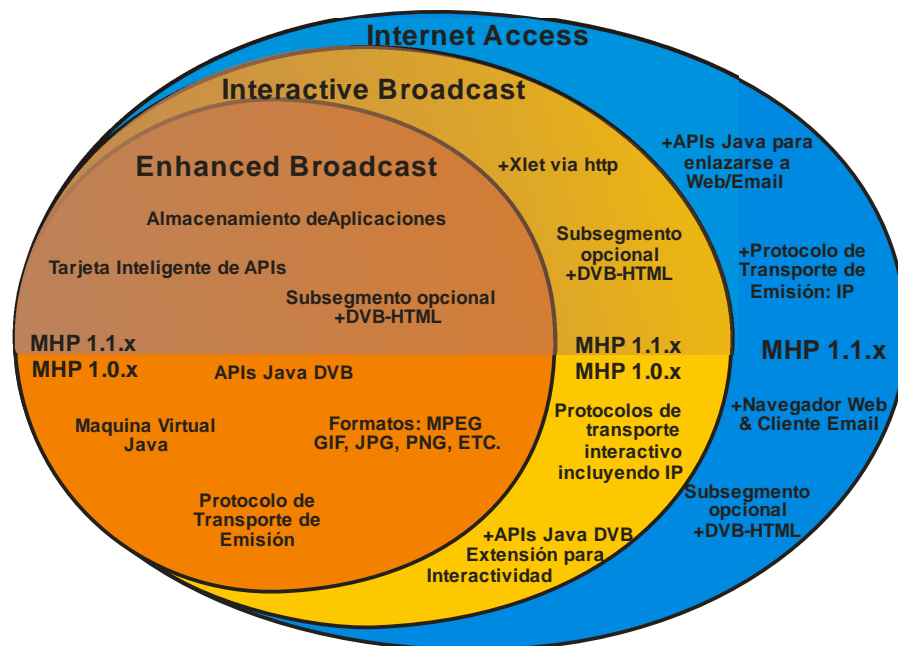


Figura 2.4. Perfiles DVB-MHP

### 2.3.1.2. Association of Radio Industries Businesses (ARIB)

ARIB (Asociación de Industrias y Negocios de Radiodifusión) es la principal entidad encargada de crear y mantener el estándar de TV Digital ISDB, esta organización reúne a varias empresas japonesas y extranjeras en el negocio de producir, financiar, fabricar, importar y exportar bienes de consumo relacionados con la radiodifusión.

En la parte de interactividad sobre TV Digital ARIB ha definido dos estándares principales, cada uno con las normativas necesarias para regular, manejar y controlar este proceso, siendo estas: ARIB STD-B24 y ARIB STD-23.

- **ARIB STD-B23 (*Application Execution EnginePlatform for Digital Broadcasting*)**: este es un estándar basado en la norma MHP, que define una plataforma para el motor ejecutor de aplicaciones para los servicios interactivos, ARIB STD-B23 busca establecer un núcleo común entre MHP, ACAP y actualmente con GEM, con el objetivo de poder compartir la misma plataforma, en el futuro.
- **ARIB STD-24 (*Data Coding and Transmission Specification for Digital Broadcasting*)**: este estándar entre otras cosas, especifica un lenguaje declarativo BML (*Broadcast Markup Language*). Desarrollado por la propia ARIB. El lenguaje BML se basa en XML (*Extensible Markup Language*), y se usa para especificar servicios multimedia para TV digital Interactiva. [9]

Por lo tanto el *middleware* que comúnmente se llama ARIB consta de una arquitectura formada por dos partes: presentación y ejecución.

El modelo de ejecución utiliza la máquina virtual Java para la interpretación de los *bytecodes*, que representan los procedimientos y/o funciones relacionadas con el contenido que se transmite. A su vez, la plantilla de presentación especifica la sintaxis del lenguaje de marcado BML (*Broadcast Markup Language*).

En la figura 2.5 se muestra la arquitectura de *middleware* japonés, en donde se muestra la división entre la parte responsable de la presentación y la responsable de la ejecución.



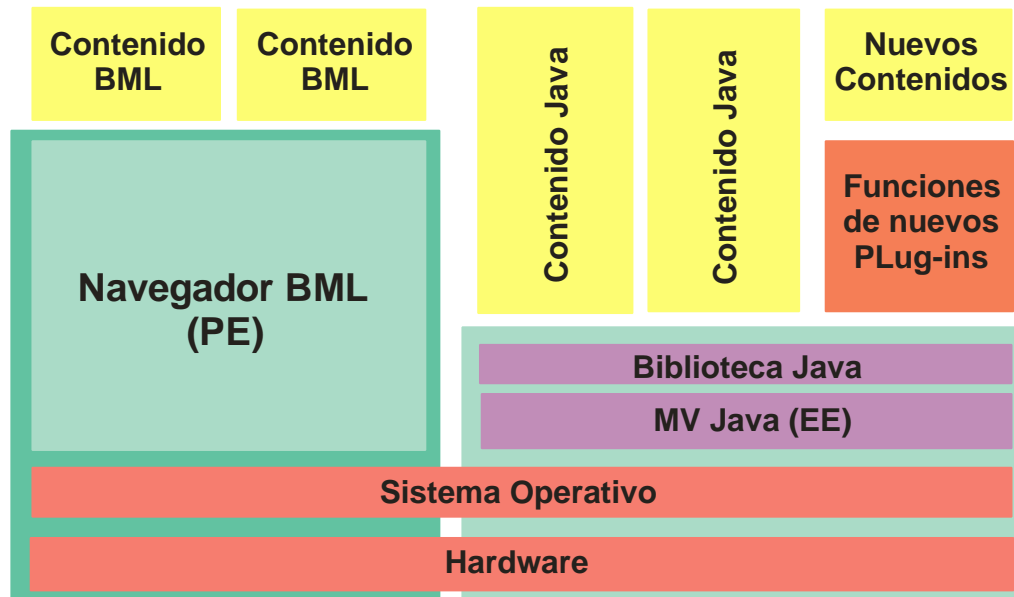


Figura 2.5. Arquitectura ARIB

El modelo de presentación corresponde al subsistema responsable de la tramitación de las aplicaciones declarativa. En el lenguaje BML se incluyen etiquetas y atributos utilizados para la autoría del contenido declarativo de TV Digital. El alcance de la aplicación estará acorde a los requerimientos de servicios multimedia. La codificación BML se define como un lenguaje basado en XML y además, la norma permite la especificación de aplicaciones a través del lenguaje Java con sus propias bibliotecas, conocida como ARIB-B23.

Una de las principales funciones del lenguaje BML es permitir o controlar la vista espacial y temporal, definir dónde, cuándo y qué mostrar en la pantalla. Otras de las tareas importantes, incluyen el mantenimiento de la relación entre la información y los medios de comunicación (visualización de los programas y las direcciones URL para el *streaming* de audio y vídeo), la estructuración de envío de la información (títulos, artículos, títulos, etc.) y la visualización de interfaces gráficas de usuario para la solicitud de información a través de la selección de iconos y otros objetos.

Po otra parte el modelo de ejecución está diseñado sobre la base de la utilización de una máquina virtual Java (instalado en el acceso de terminal) para interpretar los *bytecodes* de Java [10].

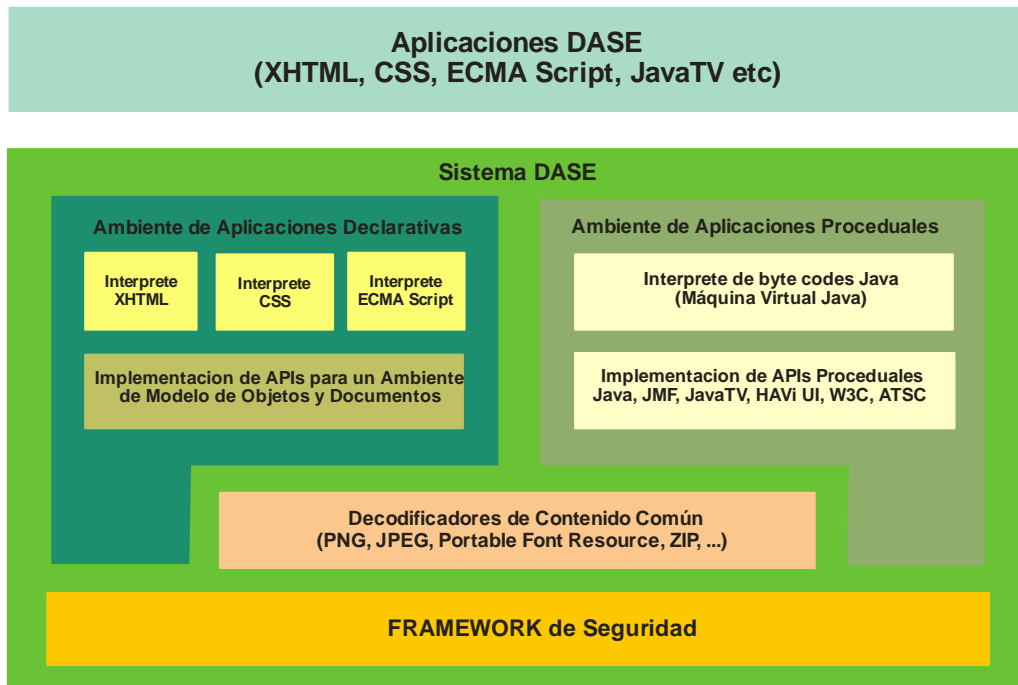
### **2.3.1.3. DTV Applications Software Environment (DASE).**

DASE es la norma del *middleware* americano, que en un comienzo fue diseñado como DASE nivel 1, este se limitó al nivel de interactividad local, porque no contemplaron la existencia de un canal de retorno. Actualmente, la versión DASE de nivel 3, más allá de solo contemplar interactividad, tiene por objeto la integración de Internet con la televisión, ofreciendo opciones correo electrónico y navegadores Web en el terminal de acceso.

Las aplicaciones DASE pueden ser tanto de entorno declarativo y de procedimiento. Sus aplicaciones declarativas están representadas por documentos hipermedia y multimedia escrito por medio de un lenguaje de marcado basado en XHTML, por ejemplo, entre otros. En cuanto a las aplicaciones de procedimiento, son programas que utilizan la tecnología Java para la ejecución de instrucciones, agregándole un mayor poder de procesamiento. La máquina virtual Java recibió extensiones que le permiten un desenvolvimiento más adecuado en el mundo de la televisión digital.

Cabe señalar que una aplicación DASE no es sólo declarativa o de procedimiento. Generalmente, las aplicaciones de *middleware* declarativas, a menudo hacen uso de *scripts*. Además, una aplicación de procedimiento puede hacer referencia a un contenido declarativo.

En la figura 2.6 se puede apreciar la arquitectura general de DASE, la capa superior de esta arquitectura representa las aplicaciones que pueden ser declarativas o de procedimiento.



**Figura 2.6. Arquitectura DASE**

El sistema DASE está formado por los módulos de Ambiente de Aplicación Declarativo (*Declarative Application Environment* o DAE) y el Ambiente de Aplicación de Procedimiento (*Procedural Application Environment* o PAE). Un DAE es un subsistema lógico que procesa lenguajes de marcado, hojas de estilos y *scripts*. Su principal componente es el DCDE (*Declarative Content Decoding Engine*), que tiene por finalidad hacer un análisis sintáctico del documento. Por otro lado, un PAE es el módulo encargado de procesar el contenido de los objetos activos, o ejecutables, su principal componente es el PCDE (*Procedural Content Execution Engine*), que por ejemplo, contiene la maquina virtual Java [10].

#### **2.3.1.4. Advanced Common Application Platform (ACAP).**

Con el objetivo de establecer un único estándar de *middleware*, y que fuese común para todos los terminales de acceso tanto de TDT (Televisión Digital Terrestre), como por cable en los Estados Unidos, un grupo de investigadores de ATSC y de empresas ligadas a la estandarización de la TV Digital americana se

reunió para armonizar sus entornos de aplicaciones original: DASE-1 y OCAP (*Open Cable Application Platform*).

El estándar OCAP, desarrollado principalmente por CableLabs, se deriva de MHP (*Multimedia Home Platform*), y adaptado a las características técnicas y de negocios, de las empresas difusión por cable en los Estados Unidos.

ACAP sigue las especificaciones definidas en el estándar GEM y utiliza el mismo conjunto de APIs Java y el modelo de aplicación que se utiliza en MHP. La obligación de un canal de retorno, el soporte a las aplicaciones almacenadas y la versión del carrusel de objetos de DSM-CC, son algunas de las diferencias existentes entre ACAP y MHP.

Las aplicaciones ACAP también se clasifican como declarativas y de procedimiento. Aquellas que contengan sólo contenido de procedimiento, escrito en Java y que pueden combinar gráficos, vídeos e imágenes, son denominadas como aplicaciones ACAP-J. Las aplicaciones declarativas se denominan ACAP-X, y son representadas por documentos multimedia escritos a través de un lenguaje de marcas como XHTML, reglas de estilo, *scripts*, videos y audios. Vale resaltar que una aplicación ACAP no tiene que ser completamente de procedimiento o declarativa, por lo tanto una aplicación ACAP-J puede hacer referencia a un contenido declarativo, así como una aplicación ACAP-X puede, por ejemplo, hacen uso de *scripts*.

En la figura 2.7 se distingue claramente la arquitectura del *middleware* ACAP, para los sistemas que soportan aplicaciones declarativas y de procedimiento. Aquí se puede observar que hay una gran similitud con la arquitectura de DASE. Mientras que el entorno ACAP-X es responsable de interpretar el contenido de una aplicación declarativa ACAP, el entorno ACAP-J es el responsable del tratamiento de los contenidos de procedimiento a través de la máquina virtual Java y de los APIs de extensión.

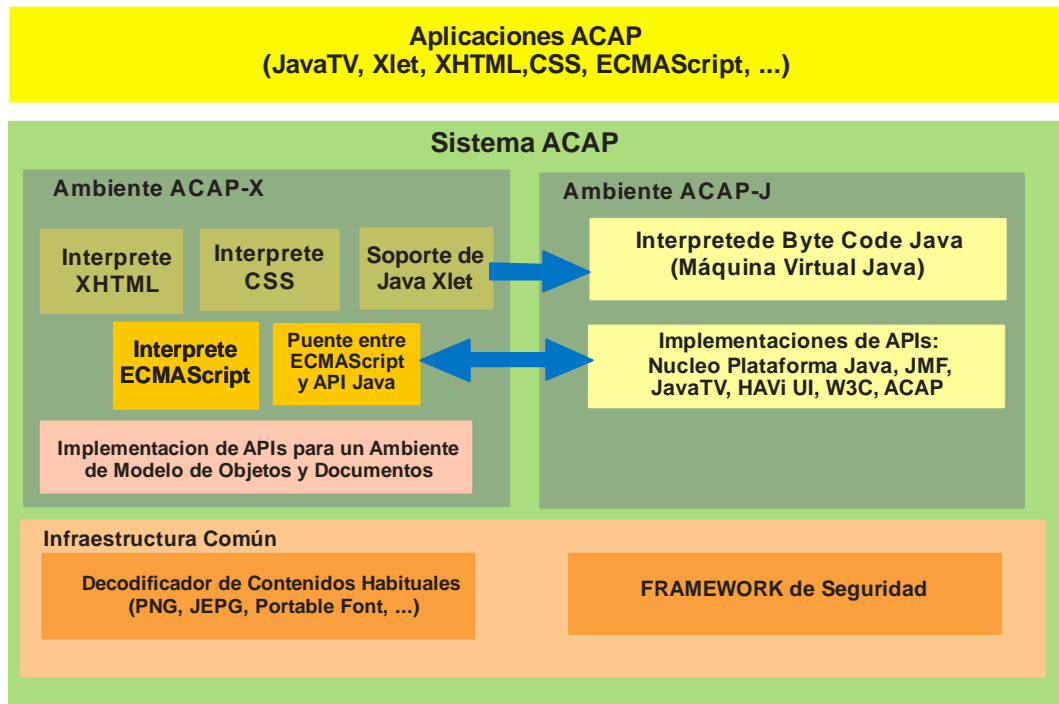


Figura 2.7. Arquitectura ACAP

Las definiciones de una aplicación de procedimiento y de entorno ACAP-J se basan en los estándares GEM y OCAP. Una aplicación de entorno ACAP-J soportan los mismos archivos de clase Java definidos por MHP, y algunas API's extendidas.

Las definiciones de una aplicación declarativa de entorno ACAP-X, se basan en el *middleware* DASE. Algunas modificaciones son necesarias para mantener la compatibilidad con el estándar GEM [10].

### 2.3.1.5. Globally Executable MHP (GEM).

Debido al rápido desarrollo y popularización del *middleware* MHP entre los diferentes países que utilizan la norma DVB, surgió la iniciativa para su implementación sobre otras plataformas internacionales. Las entidades encargadas de la normalización de los sistemas de televisión digital de fuera de Europa han expresado su interés en participar en este esfuerzo, dispuesto a ejecutar partes del *middleware* MHP en su especificación como una forma de

aprovechar el desarrollo tecnológico y para establecer compatibilidad entre ellos, de modo que todas las aplicaciones puedan ser utilizadas y presentadas en los terminales de acceso de otros estándares.

Para que las aplicaciones MHP puedan ser utilizadas sobre otras plataformas, como lo exigen los órganos tales como CableLabs (EE.UU.), ARIB (Japón), el grupo DVB decidió proponer una única especificación llamada GEM (*Globally executable MHP*). Además de capturar todas las interfaces y semántica definida por el MHP, GEM también incluye los requisitos impuestos por otras normas internacionales.

Formalmente, la especificación GEM no puede considerarse como una especificación completa para terminales de acceso, lo correcto es decir que GEM es un *framework* a partir del cual la implementación del *middleware* de un terminal de acceso puede ser instanciada. El estándar define el conjunto de APIs, garantías de semántica y formatos de contenido con el que las aplicaciones puedan contar, para la construcción de los servicios interactivos ofrecidos sobre cualquier estándar internacional de televisión digital. Para la creación de *middlewares* compatibles, se hace necesario citar de forma completa la especificación GEM, con el fin de satisfacer todas sus necesidades.

Las siguientes plataformas definidas, están basadas sobre o extendidas de GEM (Figura 2.8):

- Multimedia Home Platform (MHP), multiplataforma de *middleware*, desarrollada por el proyecto DVB;
- OpenCable Application Platform (OCAP), que es la capa de *middleware* de TVI para las cableras americanas;
- ARIB B.23, la especificación de la Asociación de Industrias y Negocios de radiodifusión de Japón;
- Advanced Common Application Platform (ACAP), el estándar de *middleware* de ATSC para televisión terrestre;
- BD-J, la plataforma Java para discos Blu-ray;

- Ginga-J, *framework* del *middleware* Brasileño.

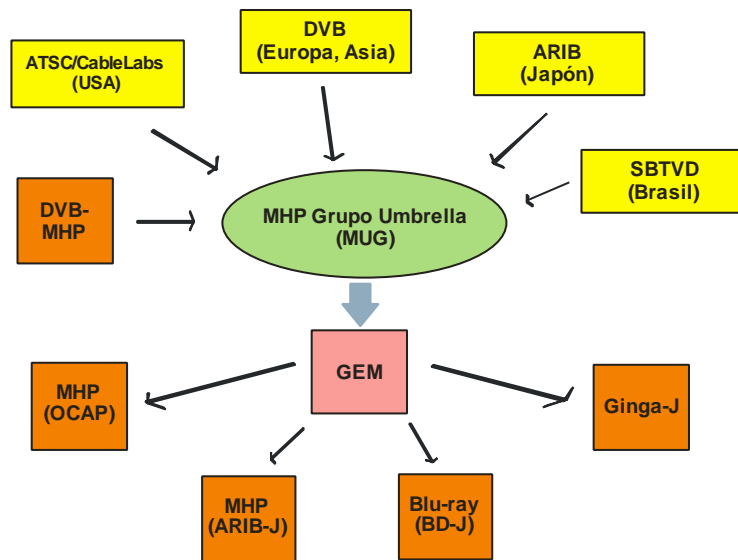


Figura 2.8. Concepto de GEM

Todas estas plataformas están basadas sobre el núcleo común de GEM, y esto permite la escritura de aplicaciones Java con interoperabilidad sobre todos los sistemas.

## CAPÍTULO III

### ARQUITECTURA DEL MIDDLEWARE GINGA

#### 3.1. Introducción

Ginga es el nombre del *middleware* abierto del Sistema Brasileño de TV Digital (SBTV-D). El nombre fue escogido en reconocimiento a la cultura, arte y continua lucha por libertad e igualdad del pueblo brasileño [11].

Ginga es una capa de software intermedio (*middleware*), entre el *hardware*/Sistema Operativo y las aplicaciones, que ofrece una serie de facilidades para el desenvolvimiento de contenidos y aplicaciones para TV Digital, permitiendo la posibilidad de poder presentar los contenidos en distintos receptores independientemente de la plataforma de *hardware* del fabricante y el tipo de receptor (TV, celular, PDAs, etc.).

*Middleware* es un neologismo creado para designar capas de *software* que no constituyen directamente aplicaciones, sino que facilitan el uso de ambientes ricos en tecnología de información. La capa de *middleware* concentra servicios como identificación, autenticación, autorización, directorios, certificados digitales y otras herramientas de seguridad. En el contexto de la TV Digital, el *middleware* viene a ser un *software* que controla sus principales facilidades (grado de programación, menús de opciones), inclusive la posibilidad de ejecutar aplicaciones, dando soporte a la interactividad.

Las aplicaciones ejecutadas sobre Ginga son clasificadas en dos categorías, dependiendo de la forma en la que son escritas. Las Aplicaciones de



Procedimiento son escritas usando el lenguaje Java y las Aplicaciones Declarativas son escritas usando el lenguaje NCL.

El *middleware* abierto Ginga esta subdividido en dos subsistemas principales entrelazados, que permiten el desenvolvimiento de aplicaciones siguiendo dos paradigmas de programación diferentes. Dependiendo de las funcionalidades requeridas en cada aplicación, un paradigma será más adecuado que el otro. Estos dos subsistemas son llamados Ginga-J para aplicaciones de procedimiento Java y Ginga-NCL para aplicaciones declarativas NCL.

Todas las propuestas del sistema de Televisión Digital especifican middlewares sobre los cuales las aplicaciones de TV Interactiva pueden ser ejecutadas.

En comparación con los sistemas de *middleware* concebidos para los otros estándares de TV Digital, algunas de las funcionalidades de Ginga son innovadoras, desarrolladas específicamente para la realidad brasileña. En Brasil, el uso de la TV como medio de inclusión digital es prioridad para el gobierno, desde este punto de vista, el desenvolvimiento de aplicaciones interactivas para TV Digital ocupa un lugar destacado, lo que no ocurre en otros países donde ya tienen implantado un Sistema de TV Digital hace mucho más tiempo.

### **3.2. Arquitectura del *Middleware* Ginga**

El *middleware* Ginga a diferencia de los sistemas hipermedia convencionales, en donde prevalece el modelo de servicio tipo *pull*, esto es, donde es responsabilidad del programa intérprete solicitar un nuevo documento y buscar el contenido de los objetos, como ocurre por ejemplo con los navegadores web, en TV un modelo de servicio es de tipo *push*. En este caso, la emisora proporciona para la difusión flujos de audio y vídeo multiplexados con otros datos, algunos objetos multimedia pueden ser recibidos bajo demanda, más un servicio de tipo *push* es predominante.

Además de invertir un paradigma de servicio, los usuarios pueden comenzar a mirar un programa ya iniciado, más aun todavía, los usuarios pueden querer cambiar de canal, y por consiguiente salir y entrar a programas ya en proceso.

Otro aspecto desafiante es la edición de documentos durante su exhibición, en los programas en vivo y en programas modificados por retransmisoras, esta posibilidad es bastante interesante.

Similar a trabajar en un proyecto de construcción, la mejor forma de trabajar con un sistema complejo, como es el caso de un sistema de TV digital interactiva, es a través de la representación de su arquitectura. Una arquitectura muestra los principales elementos de un sistema, expresando claramente sus interacciones y escondiendo los detalles menos importantes bajo el punto de vista adoptado. En la Figura 3.1 se muestra la arquitectura de la TV digital, representada en capas de todas las tecnologías ya existentes [11].

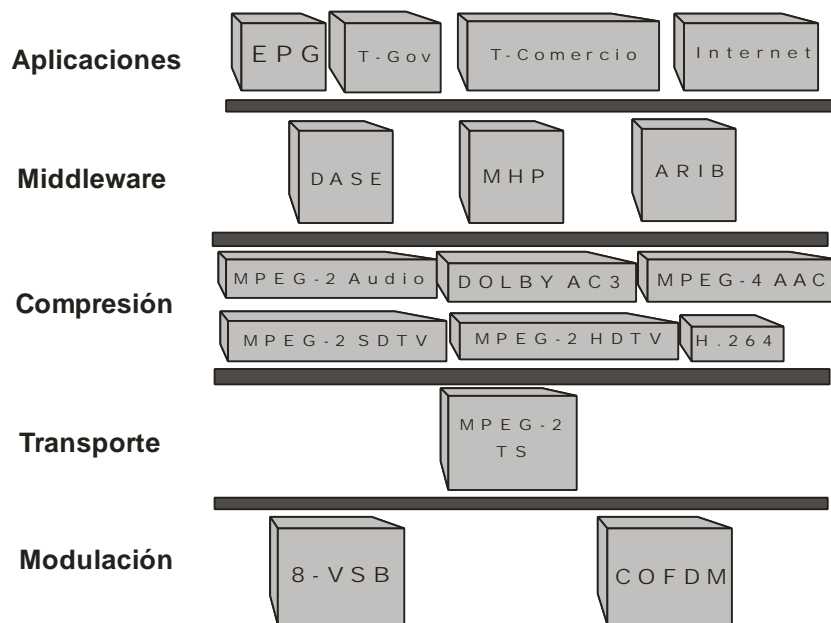


Figura 3.1 Arquitectura de la TV Digital, con las tecnologías utilizadas en cada capa

En la figura 3.2 se presentan las normas de referencia del Sistema Brasileño de TV digital terrestre, incluido su *middleware* llamado Ginga.

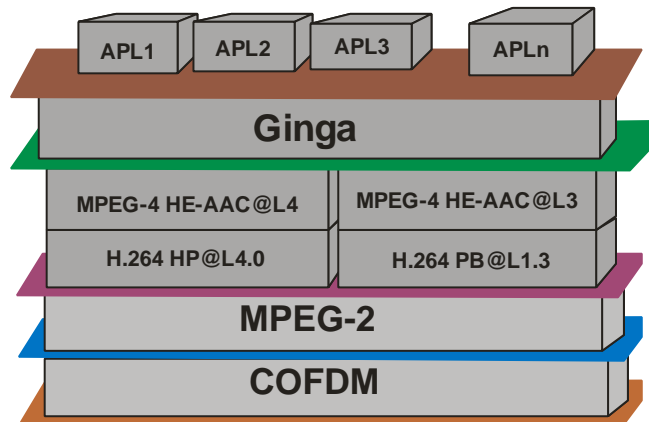


Figura 3.2. Normas de referencia del Sistema Brasileño de TV Digital

La idea central de la arquitectura en capas, está en que cada una ofrezca servicios para la capa superior y use los servicios ofrecidos por la inferior. De esta manera, las aplicaciones que se ejecutan en TV Digital interactiva usen la capa del *middleware*, que intermedia toda la comunicación entre las aplicaciones y el resto de los servicios ofrecidos por las capas inferiores.

La finalidad de la capa del *middleware* o capa intermedia es ofrecer un servicio estandarizado para las aplicaciones (capa superior), escondiendo las peculiaridades y heterogeneidades de las capas inferiores (tecnologías de compresión, de transporte y de modulación). El uso del *middleware* facilita la portabilidad de las aplicaciones, permitiendo que sean transportadas para cualquier receptor digital (o *set top box*) que soporte el *middleware* adoptado. Esa portabilidad es primordial en sistemas de TV digital, pues es muy complicado considerar como premisa que todos los receptores digitales sean exactamente iguales.

Cuando se busca los requisitos de un *middleware*, teniendo por base las aplicaciones que se serán desenvueltas en un sistema de TV Digital, cuatro puntos llaman la atención:

- Soporte sincronización de archivos multimedia.
  - Sincronización basada en estructuras

- Soporte de canal de retorno
- Soporte múltiples dispositivos de visualización.
- Soporte el desarrollo de programas en vivo (en tiempo de exhibición).
- Soporte la adaptación del contenido y de la forma en cómo el contenido es presentado.

En el caso especial de Brasil, el *middleware* también ofrece un buen soporte al desenvolvimiento de aplicaciones destinadas a la inclusión social, como aplicaciones para educación, salud, etc.

Enfocado a esos requisitos, el universo de las aplicaciones Ginga pueden ser particionadas en un conjunto de aplicaciones declarativas y un conjunto de aplicaciones de procedimiento.

Los lenguajes declarativos son más intuitivos es decir de más alto nivel y es por esto que resultan más fáciles de utilizar, normalmente no exigen de un programador experto. Sin embargo, los lenguajes declarativos tienden a ser definidos con un enfoque específico. Cuando el enfoque de la aplicación no se relaciona con el del lenguaje, el uso de un lenguaje de procedimiento es apenas ventajoso, mas se hace necesario. Un uso del lenguaje de procedimiento usualmente requiere un muy buen nivel de programación. Una aplicación híbrida es aquella cuyo conjunto de entidades posee tanto contenidos de tipo declarativos como de procedimiento. Una aplicación, entretanto, no necesariamente debe ser puramente declarativa o de procedimiento; sin temor a equivocación se puede afirmar que en los sistemas de TV digital, ambos tipos de aplicaciones coexisten, siendo pertinente que los dispositivos receptores integren soporte para ambos tipos en su *middleware*. Esto ocurre en todos los *middlewares* de todos los sistemas incluido el *middleware* Ginga.

En particular, las aplicaciones declarativas frecuentemente hacen usos de *scripts*, cuyo contenido es de naturaleza de procedimiento, además una aplicación declarativa puede hacer referencia a un código *JavaTV Xlet* incorporado. De igual manera, una aplicación de procedimiento puede hacer referencia a una aplicación declarativa, conteniendo, por ejemplo, contenido gráfico, o puede construir o

iniciar las aplicaciones con contenidos declarativos. Por lo tanto, ambos tipos de aplicaciones Ginga pueden utilizar las facilidades de los ambientes de aplicación declarativo y de procedimiento.

Una arquitectura de implementación de referencia del *middleware* Ginga puede ser dividida en tres grandes módulos: Ginga-CC (*Common Core*), el ambiente de presentación Ginga-NCL (declarativo) y el ambiente de ejecución Ginga-J (de procedimiento); esta arquitectura de muestra en la figura 3.3.

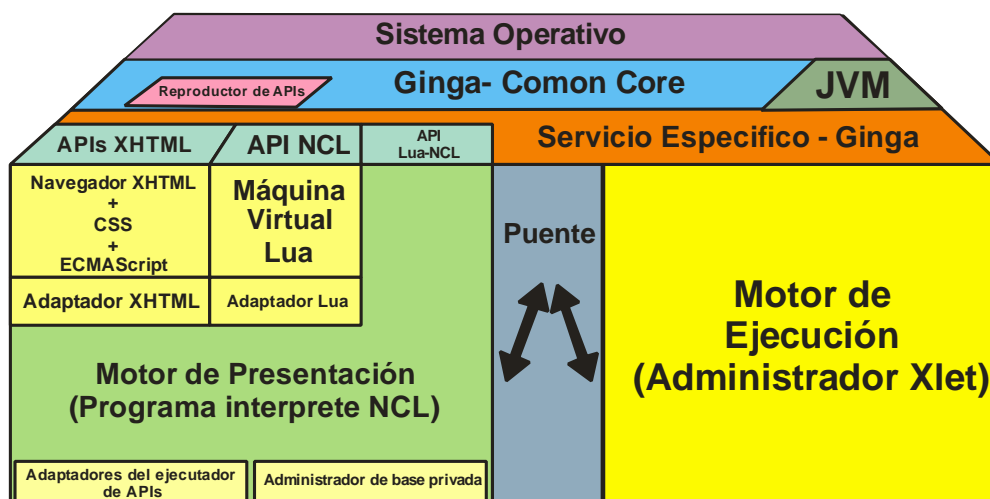


Figura 3.3. Arquitectura Ginga

Ginga-NCL es el subsistema lógico del sistema Ginga que procesa documentos NCL. Un componente clave de Ginga-NCL es el motor decodificador de contenidos declarativos (Programa intérprete NCL llamado Maestro). Otro de los módulos importante es el agente de usuario del modelo XHTML, el cual incluye un *stylesheet* (CSS), el interpretador *ECMAScript*, y la máquina Lua, la que es responsable de interpretar los *Scripts* Lua. La especificación de este subsistema se base en las normas ABNT NBR 15606-2 [12] y ABNT NBR 15606-5 [13].

Ginga-J es el subsistema lógico del sistema Ginga que procesa el contenido de los objetos Xlet. Un componente clave del ambiente de aplicaciones de procedimiento es el motor de ejecución de contenidos de procedimiento,

compuesta por la máquina virtual de Java. La especificación de este subsistema se basa en la norma ABNT NBR 15606-4 [14].

Ginga-NCL y Ginga-J son construidos sobre los servicios ofrecidos por el módulo del núcleo común de Ginga (*Ginga-Common Core*), cuya composición se muestra en la figura 3.4.

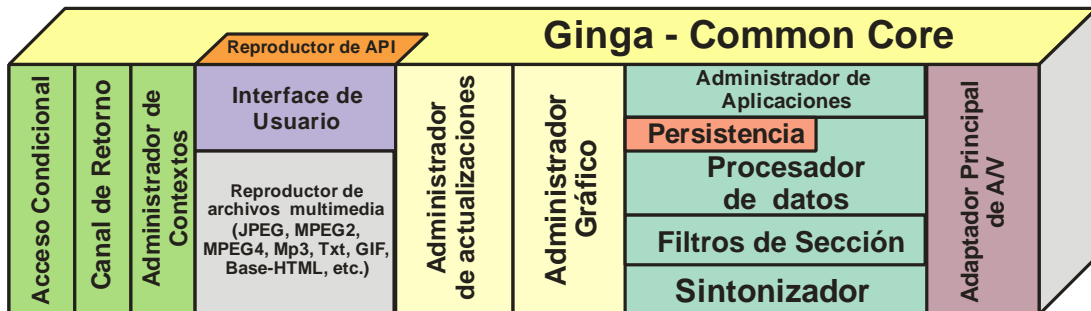


Figura 3.4. Ginga Common Core

Los decodificadores de contenido común sirven tanto para las necesidades de decodificación de las aplicaciones declarativas como de las de procedimiento, así también para la presentación de contenidos comunes como PNG, JPEG, MPEG y otros formatos. El núcleo común de Ginga está compuesto por decodificadores de contenido común y procedimientos para obtener contenidos, transportados en un flujo de MPEG-2 o a través del canal de retorno. El núcleo común de Ginga debe también soportar el modelo de despliegue conceptual especificado por el estándar brasileño de TV Digital.

La arquitectura y facilidades de Ginga son proyectadas a ser aplicadas a sistemas de radiodifusión y receptores terrestres (medio aire). Sin embargo, la misma arquitectura y facilidades pueden ser aplicadas a otros sistemas de transporte (como satélite, TV por cable, y sistemas de IPTV).

En general, Ginga no discrimina cualquier aplicación nativa que también puede escoger compartir el plano de gráficos. Esto incluye pero no se limita a: subtítulos para sordos, sistema de mensajes de acceso condicional, menús del receptor, y guía de programas nativos. Las aplicaciones nativas pueden tomar precedencia

sobre aplicaciones Ginga. Los subtítulos para sordo y los mensajes de emergencia deben tomar precedencia sobre el sistema Ginga.

Algunas aplicaciones nativas, como el encabezamiento cerrado, presentan un caso especial donde la aplicación nativa puede estar activa por largos periodos concurrentemente con aplicaciones Ginga.

### **3.2.1. Ginga – NCL [15]**

#### **3.2.1.1. El Lenguaje NCL**

Todas las maquinas de presentación de los tres principales sistema de TV Digital (BML, DVB-HTML y ACAP-X) se basan en lenguaje XHTML.

XHTML es un lenguaje declarativo de base multimedia que propone la estructura definida por la relación entre los objetos XHTML (documentos XHTML u objetos encerrados en documentos XHTML) la que es encajada en los contenidos multimedia del documento. XHTML puede así ser clasificado como un lenguaje de *markup*: un formalismo que describe una clase de documentos que emplean el *markup* en orden a delinear la estructura del documento, apariencia y otros aspectos.

Las relaciones de referencia definidas por los enlaces XHTML son el enfoque del lenguaje declarativo XHTML. Otro tipo de relación, como la relación de sincronización de espacio-tiempo y relaciones alternativas (adaptabilidad multimedia), son usualmente definidas utilizando lenguajes imperativos (por ejemplo *ECMAScript*); así ellos no pueden tomar ventaja de la forma fácil de autoría ofertada en otros lenguajes declarativos, como NCL y SMIL (*Synchronized Multimedia Integration Language*).

Al contrario de HTML o XHTML, NCL y SMIL tienen una estricta separación entre contenido y estructura, y ello les provee un control no invasivo de vinculación a la presentación y diseño

El enfoque del lenguaje declarativo NCL es más amplio que el de su similar XHTML. Generaliza la sincronización espacio-tiempo por medio de los enlaces

(links) NCL; adaptabilidad, definida por los elementos NCL *switch* y *switch* de descriptores; y soporte a múltiples dispositivos de reproducción, definidos por las regiones NCL, son el enfoque del lenguaje declarativo NCL. La interacción del usuario es tratado simplemente como un caso especial de la sincronización temporal. Las mismas facilidades se pueden encontrar en SMIL, pero estas están obligadas a generar el documento antes de empezar la presentación. La generación de documentos en vivo no es soportada por SMIL como lo hace NCL.

Es más, SMIL fue todavía restringido a la Web y no puede ser soportado por un ambiente de radiodifusión aún. Como NCL tiene una separación estricta entre contenidos y estructura, NCL no puede definir algún archivo multimedia por sí mismo.

En cambio, este define el pegamento que mantiene unidos a los archivos multimedia en la presentación multimedia. Por consiguiente, un documento NCL solo define como estructurar y relacionar los objetos multimedia, en tiempo y espacio. Como un “lenguaje pega”, este no restringe ni prescribe el tipo de contenido de los objetos multimedia. En este sentido, se pueden tener objetos de imágenes (GIF, JPEG, PNG, etc.), objetos de vídeo (MPEG, MOV, etc.), objetos de audio (MP3, WMA, etc.), objetos de texto (TXT, PDF, etc.), objetos de ejecución (Xlet, Lua, etc.), etc., como objetos multimedia NCL. El tipo de objetos multimedia que soporta NCL depende del reproductor multimedia que es incorporado en el programa intérprete NCL (NCL Player), de hecho los reproductores multimedia comunes son ofertados por el modulo de núcleo común de Ginga. Uno de estos reproductores es, por supuesto, el decodificador/reproductor implementado en *hardware* por el receptor de TVD. De esta forma, se puede notar que el flujo principal de audio y vídeo es tratado como todos los otros objetos multimedia que pueden ser relacionados al uso de NCL.

Otro objeto NCL multimedia que puede ciertamente ser soportado, es el objeto multimedia del modelo HTML. Por lo tanto, NCL no substituye, pero incorpora documentos del modelo HTML (u objetos). Como con otros objetos multimedia, el lenguaje del modelo HTML puede tener soporte en el programa intérprete NCL como una opción de implementación, y por lo tanto podría depender sobre, que



navegador HTML actuara como su reproductor multimedia incorporado en el programa intérprete NCL.

Como una consecuencia, es posible tener un navegador BML, navegador DVB-HTML y navegador ACAP-X incorporados en un reproductor de documentos NCL. Esto incluso es posible para recibir un código de navegador a través de *datacasting* e instalarlo como un *plug-in* (usualmente un *plug-in* Java).

Debe enfatizarse que, con el afán de soportar todas las facilidades de los navegadores del modelo XHTML definidos por otros estándares de TVD, todas las especificaciones de la TV digital terrestre brasileña relacionadas a *datacasting* deben soportar estas facilidades, como las del flujo de transporte de eventos por ejemplo.

Aunque un navegador del modelo XHTML debe ser soportado, el uso de los elementos XHTML para definir una relación (incluyendo enlaces XHTML) deben ser disuadido cuando se autoría documentos NCL. La estructura basada en autoría debe ser enfatizada por razones bien conocidas largamente reportadas en la literatura.

Durante la exhibición de contenidos de objetos multimedia, varios eventos son generados. Ejemplos de eventos son la presentación de segmentos marcados de contenidos de objetos multimedia, la selección de una marca de un segmento de contenido, etc.

Los eventos pueden generar acciones sobre otros objetos multimedia, como poner en marcha o parar su presentación. En consecuencia los eventos deben ser reportados por el reproductor multimedia del programa intérprete NCL que, en su giro, puede generar acciones para ser aplicadas a este u otros reproductores. Ginga-NCL define un adaptador de API para estandarizar la interfaz entre el programa intérprete NCL, y cada reproductor específico, como muestra la figura 3.4.

Cuando algún reproductor multimedia, en particular un navegador del modelo XHTML, es integrado al programa intérprete de Ginga-NCL, este debe soportar el adaptador de API. Para algunos reproductores multimedia, incluyendo al

navegador del modelo XHTML, un módulo del adaptador puede ser necesariamente para lograr la integración.

Finalmente, para la edición en vivo, Ginga-NCL también ha definido flujos de eventos en orden para soportar eventos generados en vivo en flujos multimedia, en particular el flujo de vídeo del programa principal. Estos eventos son generalizados con el mismo concepto encontrado en otros estándares, como por ejemplo los eventos-b de BML. Nuevamente, aunque un navegador basado en XHTML debe ser soportado, el uso de elementos XHTML para definir relaciones (incluyendo el flujo de eventos) debe ser disuadido en la autoría de documentos NCL. Por todas las razones mencionadas, NCL se ha vuelto la opción natural para el estándar de TV Digital Brasileño.

### **3.2.1.2. Objetos Multimedia NCL**

Como se mencionó, NCL no restringe o prescribe los tipos de contenidos de objetos multimedia. En este sentido, podemos tener objetos de imágenes, objetos de vídeo, objetos de audio, objetos de texto, y también objetos del modelo XHTML, que pueden contener insertados objetos de escritura (*script*). En el caso del *middleware* Ginga-NCL, el único código *script* permitido para ser insertado en objetos del modelo XHTML es el código *ECMAScript*.

NCL también permite objetos multimedia cuyo contenido es un código de procedimiento especificados en algunos lenguajes de procedimiento.

Sin embargo, el lenguaje de escritura de Ginga-NCL es Lua. Ginga-NCL también ofrece soporte a objetos multimedia que llevan códigos Xlet.

### **3.2.1.3. Visión global de los elementos del Lenguaje NCL**

NCL es una aplicación XML que permite un acceso por módulos. El acceso por módulos se ha usado en varias recomendaciones de lenguaje W3C. Un módulo es la una colección de relaciones semánticas de elementos XML, atributos y valores de atributos que presentan una unidad de funcionalidad.

Los módulos son definidos en conjuntos coherentes. Un perfil del lenguaje es una combinación de módulos, para el estándar Brasileño de TVD, Ginga-NCL define dos perfiles de lenguaje: el Perfil EDTV (*Enhanced Digital TV Profile*) y el Perfil BDTV (*Basic Digital TV Profile*).

En esta sección se describirán las principales definiciones hechas por Ginga-NCL. La estructura básica de un módulo NCL lo definen el elemento raíz, llamado <ncl>, y sus elementos hijo, el elemento <head> y el elemento <body>, siguiendo la terminología adoptada por otro estándar W3C.

El elemento <head> puede contener los siguientes elementos hijo: <importedDocumentBase>, <ruleBase>, <transitionBase>, <regionBase>, <descriptorBase>, <connectorBase>, <meta> y <metadata>.

El elemento <body> puede contener los elementos hijo <port>, <attribute>, <media>, <context>, <switch> y <link>. El elemento <body> es tratado como un nodo de contexto NCM (*Nested Context Model*). NCM, es el modelo conceptual de NCL, donde un nodo puede ser un contexto, un *switch* o un objeto multimedia. Los nodos de contexto pueden contener otros nodos NCM y enlaces (*links*). Un nodo *switch* contiene otros nodos NCM, los nodos NCM están representados por sus correspondiente elementos NCL.

El elemento <media> define un objeto multimedia específico, su tipo y su localización. Otros tipos de elementos <media> son las de tipo “*application/x-ginga-settings*”, que especifica un objeto cuyos atributos son variables globales definidas por el documento original o son variables de ambiente reservadas que pueden ser manipuladas por el procesamiento del documento NCL; y las de tipo “*application/x-ginga-time*”, que especifican un elemento <media> especial cuyo contenido es el Tiempo del Meridiano de Greenwich (GTM).

El elemento <context> es el responsable de la definición de nodos de contexto. Un nodo de contexto NCM es un tipo particular de un nodo compuesto NCM y este define como será contenido un grupo de nodos y un grupo de enlaces como ya se menciona. Al igual que el elemento <body>, un elemento <context> puede tener elementos hijo <port>, <attribute>, <media>, <context>, <switch>, y <link>.

El elemento <switch> permite la definición de nodos de documentos alternativos (representados por los elementos <media>, <context> y <switch>) para ser escogidos durante su tiempo de presentación. Las reglas de prueba utilizadas en la selección del componente *switch* a ser presentadas se definen por el elemento <rule> o <compositeRule>, los que son agrupados por el elemento <ruleBase>, definido como un elemento hijo del elemento <head>.

Las interfaces NCL funcionalmente permiten la definición de nodos de interface a ser utilizado en relación con otras interfaces. El elemento <area> permite la definición de anclas de contenidos representando porciones espaciales, opciones temporales, u opciones temporales y espaciales de un objeto multimedia (<media>).

El elemento <port> especifica un nodo puerta compuesto (<context>, <body> o <switch>) con su respectivo mapeo a una interface de uno de sus componentes hijo. El elemento <attribute> es usado para definir un nodo de atributo o un grupo de nodos de atributo como uno de los nodos de interface. El elemento <switchPort> permite la creación de interfaces de elementos <switch> que son mapeadas para un grupo de interfaces alternativas de nodos de *switches* internos.

El elemento <descriptor> especifica la información temporal y espacial necesaria para presentar cada componente del documento. El elemento puede referirse a un elemento <region> en el caso que se quiera definir su la posición inicial de la presentación de un elemento <media> en algún dispositivo de salida. Los elementos <descriptor> deben ser definidos dentro del elemento <head> del documento, el elemento <regionBase> define un grupo de elementos <region>, cada uno de los cuales puede contener anidados otros elementos <region>, y así sucesivamente, las regiones definen las áreas de presentación que se utilizaran en el dispositivo y son referenciadas por los descriptores, como ya se mencionó.

Un elemento <causalConnector> representa una relación que puede ser usada para la creación de elementos <link> en el documento. En una relación causal, una condición debe ser satisfecha para activar una acción. Un elemento <link> enlaza (a través de los elementos <bind>) un nodo de interface con los roles del conector, definiendo una relación espacio-temporal entre los objetos NCL.

El elemento <descriptorSwitch> contiene un grupo de descriptores alternativos para ser asociados con un objeto NCL. Similar al elemento <switch>, un <descriptorSwitch> la selección se realiza durante la presentación del documento, utilizando las reglas de prueba definidas por el elemento <rule> o <compositeRule>.

Con el fin de permitir una entidad base para incorporar otra base ya definida, se puede utilizar el elemento <importBase>. Adicionalmente, un documento NCL puede ser importado a través del elemento <importNCL>. El elemento <importedDocumentBase> especifica un grupo de documento NCL importados, y debe también ser definido como un elemento hijo del elemento <head>.

Algunos atributos de elementos NCL importantes son definidos en otros módulos NCL. El modulo de entidad de reuso (EntityReuse) permite la reutilización de un documento NCL. Este módulo define al atributo *refer*, el que hace referencia a un elemento URI que puede ser reutilizado. Solo los elementos <media>, <context>, <body> y <switch> pueden ser reutilizados. El módulo de navegación por teclas (*KeyNavigation*) provee las extensiones necesarias para describir las operaciones de movimientos de foco utilizando un dispositivo de control como el control remoto. Básicamente, este módulo define atributos que pueden ser incorporados por elementos <descriptor>.

Algunas funcionalidades de SMIL son también incorporadas por NCL. El elemento <transition> y algunos atributos de transición son definidos en el modulo de Transiciones Básicas (*BasicTransitions*) y el módulo de Modificación de Transiciones (*TransitionModifiers*) del SMIL. El elemento <transitionBase> de NCL especifica un grupo de efectos de transición, definidos por el elemento <transition>, y debe ser definido como un elemento hijo del elemento <head>.

Finalmente, el módulo SMIL de Meta Información (*MetaInformation*) es también incorporado. Este módulo no contiene información que sea usada o mostrada durante la presentación. En cambio, este contiene información sobre contenidos que son usados o mostrados. El módulo de Meta Información posee dos elementos que permiten describir documentos NCL.

El elemento <meta> especifica un solo par de propiedades o valores. El elemento <metadata> contiene información que es también relacionado a la meta información del documento. Esto actúa como el elemento ruta de un árbol RDF: el elemento RDF y su subsistema de elementos.

#### **3.2.1.4. Comandos de Edición Ginga-NCL**

El núcleo del motor de presentación de Ginga-NCL está compuesto por el programa Intérprete NCL y el Administrador de Base Privada. El Programa Intérprete NCL está a cargo de recibir un documento NCL y controlar su presentación, tratando de garantizar que las relaciones especificadas entre los objetos multimedia sea respetada. El Programa Intérprete trata con documentos que son reunidos dentro de una estructura de datos conocida como base privada. Ginga asocia una base privada con un canal de TV.

El Administrador de Base Privada está encargado de recibir comandos de edición de documentos NCL y mantener los documentos NCL activos (los documentos son presentados). Los comandos de edición NCL del modelo XML [ISDTV-T N06 V02] están divididos en tres subgrupos.

El primer subgrupo está enfocado sobre la base privada de activación y desactivación (comandos: openBase, activateBase, deactivateBase, saveBase y closeBase). Los documentos NCL pueden ser añadidos a una base privada y entonces ser arrancados, pausados, resumidos, parados y removidos, a través de comandos bien definidos que componen el segundo subgrupo.

El tercer subgrupo define comandos para edición en vivo, permitiendo a los elementos NCL ser añadidos o removidos, y además permitiendo valores para modificar los elementos <attribute> de NCL.

Al agregar comandos, estos siempre tienen elementos NCL como sus argumentos. Sea que los elementos especificados existan o no, la consistencia del documento debe mantenerse por el Programa Intérprete, en el sentido que todos los atributos del elemento clasifiquen, deben ser definidos como requeridos. El elemento es definido usando una notación de sintaxis del modelo XML idéntica

a la utilizada por la definición de documentos NCL (definidos por los esquemas NCL).

El DSM-CC (*Digital Storage Media – Command and Control*) es adoptado en Ginga para el acarreo de comandos de edición en un flujo elemental de MPEG-2 TS (*Transport Stream*).

### **3.2.1.5. Núcleo Común de Procesamiento de Datos**

El Núcleo Común de Ginga está compuesto por decodificadores de contenido común, y además debe soportar el modelo conceptual de despliegue especificado por el estándar de TV Digital Brasileño [ISDTV-T N06 V01].

Otras de las funciones del Núcleo Común es establecer los procedimientos para obtener los contenidos transportados en un flujo de transporte MPEG-2 o por un canal de retorno, y para reconstruir la estructura de datos requerida para el proceso del documento NCL.

El DSM-CC es adoptado en Ginga para llevar los comandos de edición en flujos elementales de MPEG-2 TS. Los eventos de flujo DSM-CC y el protocolo de carrusel de objetos DSM-CC son la base para manejar documentos en el Motor de Presentación de Ginga.

Como es usual, los descriptores de eventos de flujo DSM-CC deben tener una estructura, compuesta básicamente por un identificador (id), un tiempo de referencia y un campo de datos privado. La identificación únicamente determina el evento de flujo como un comando de edición. La referencia de tiempo, indica el momento exacto cuando el evento será activado.

Una referencia de tiempo igual a cero informa que el evento debe ser activado inmediatamente después de ser recibido (los eventos llevan este tipo de referencia de tiempo que son comúnmente conocidas como eventos “hágalo ahora”). El campo de datos privado provee soporte para los parámetros de evento; y los parámetros del comando de edición para el caso del descriptor de evento de flujo. Si un parámetro de comando NCL del modelo XML es lo suficientemente corto, este es transportado directamente en el evento de flujo del

descriptor de carga útil (*payload*). Por otra parte, el acarreo de carga útil solo hace referencia al parámetro del comando del modelo XML (un archivo XML y es asociado a archivos de contenido multimedia, si es necesario) que es puesto en el sistema de archivos para ser transportados por el carrusel de objetos DSM-CC.

El protocolo de carrusel de objetos DSM-CC permite la transmisión cíclica de objetos de evento y sistemas de archivos. Un generador de carrusel DSM-CC es utilizado para unir el sistema de archivos y los objetos de evento de flujo en un flujo elemental.

Los objetos de evento trazan los nombres del evento de flujo en los identificadores de evento de flujo. Los objetos de evento se emplean además para informar a Ginga sobre los eventos de flujo DSM-CC que debe recibir.

Los nombres de evento permiten especificar los tipos de eventos, ofreciendo un nivel de abstracción más alto para las aplicaciones del *middleware* cuando se registra y manipula eventos de flujo DSM-CC.

El Administrador de Base Privada, así como los objetos de ejecución NCL (por ejemplo NCLua, NCLet, etc.), deben utilizar nombres de evento para registrarse como oyentes de los eventos de flujo que manejen. El Administrador de Base Privada debe registrarse como oyente (*listener*) del nombre de evento de flujo “*gingaEditingCommand*”.

La transmisión de archivos del documento NCL puede hacerse a través de un carrusel de objeto o a través de un protocolo de canal interactivo [ISDTV-T Standard 07].

Cuando los parámetros de comando (*command parameters*) NCL del modelo XML (documentos NCL o nodos NCL a ser incluidos en un documento NCL) son transmitidos a través de un carrusel de objeto, el Núcleo Común de Ginga es responsable de acoplar cada archivo de organización del sistema que compone el parámetro del modelo XML al dispositivo de recepción. En el mismo carrusel de objeto que lleva el documento NCL o especificación de nodo, un objeto de evento debe ser transmitido con el fin de trazar el nombre “*gingaEditingCommand*” al evento “*eventId*” del descriptor de evento de flujo DSM-CC, el cual debe llevar el correspondiente comando de edición. El descriptor de eventos de flujo debe llevar



un grupo de pares de referencia {uri, ior}. El parámetro “uri” del primer par debe tener el esquema “x-isdtv” y la ruta local absoluta del documento NCL o la especificación del nodo NCL (la ruta en el servidor de datos). En lo que corresponde al parámetro “ior”, en el se debe referenciar al documento NCL o especificar el IOR (*carouselId, moduleId, objectKey*) del nodo NCL en el carrusel de objeto [ISDTV-T N06 V03]. Si otros sistemas de archivo deben ser transmitidos utilizando carruseles de objeto con el fin de completar el comando “*addDocument*” o “*addNode*” con contenido multimedia, otro par {uri, ior} debe ser presentado en el comando. En este caso, el parámetro “uri” debe tener el esquema “x-isdtv” y la ruta local absoluta de la raíz del sistema de archivo (la ruta en servidor de datos), y su correspondiente parámetro “ior” en el par debe referenciar el IOR de cualquier archivo de raíz hija o directorio en el carrusel de objeto (el IOR de la entrada de servicio de carrusel). En la figura 3.5 se presenta un ejemplo de la transmisión de un documento NCL a través de un carrusel de objeto.

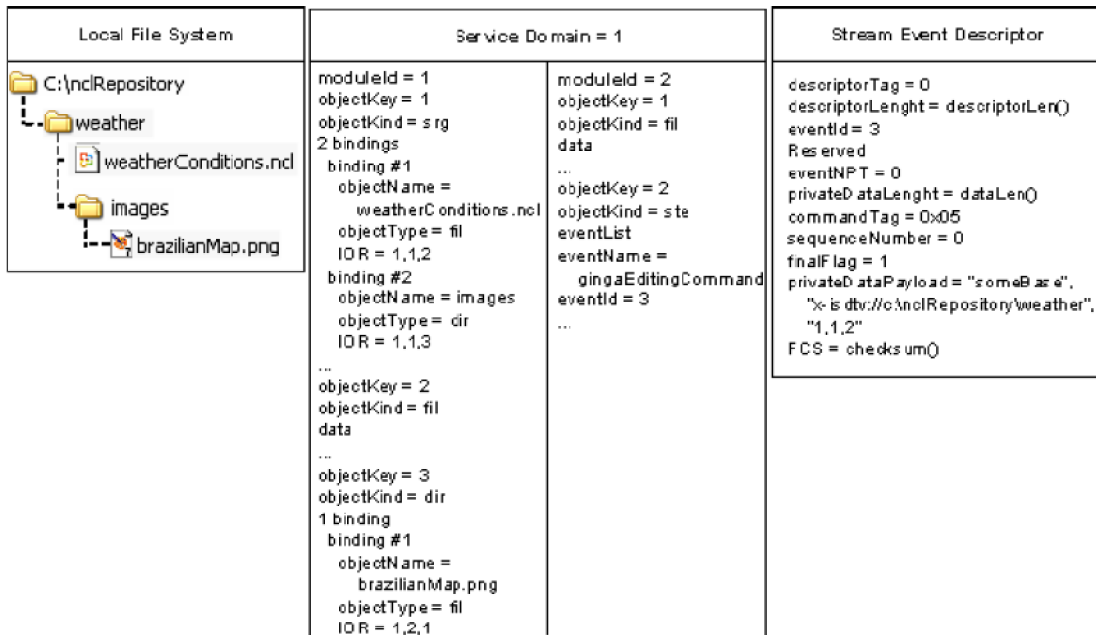


Figura 3.5. Ejemplo de la transmisión de un Documento NCL

En este ejemplo, un proveedor de contenidos quiere transmitir un programa interactivo llamado “interactiveProgram.ncl” almacenado en uno de sus

servidores de datos (llamado Local File System en la Fig. 3.5). Un carrusel de objeto debe entonces ser generado (Service Domain=1, en la Fig. 3.5) llevando todos los contenidos del programa interactivo (el archivo NCL y todos los archivos multimedia) y también un objeto de evento (moduleId=2 y objectKey=2, en la Fig. 3.5) trazando el nombre “gingaEditingCommand” al valor del evento “eventId” (de valor “3”, en la Fig. 3.5). Un descriptor de evento de flujo debe también ser transmitido con un valor apropiado de eventId, para este ejemplo su valor es “3”, y el valor de “commandTag” especificando un comando “addDocument” (código hexadecimal 0x05). El parámetro “uri” debe tener el esquema “x-isdtv” y la ruta absoluta del documento NCL (“C:/nclRepository/weather”, en la Fig. 3.5). Finalmente, el IOR del documento NCL en el carrusel de objeto se lleva en el parámetro xmlDocument (carouselId=1, moduleId=1, objectKey=2, en la Fig. 3.5).

Por otro lado, si un protocolo de canal interactivo es usado para descargar un documento NCL a través del canal de retorno, la descarga debe ser requerida a por medio del parámetro de documento “uri” pasado en el comando de edición “addDocument”. Además, al menos un programa relacionado al carrusel de objeto debe ser transmitido, llevando el trazado de evento de objeto el nombre de “gingaEditingCommand” para el evento “eventId” del evento de flujo DSM-CC, el cual debe llevar el comando de edición “addDocument”.

### **3.2.1.6. Lua: El lenguaje de escritura de NCL**

El lenguaje de escritura adoptado por Ginga-NCL para implementar objetos de procedimiento insertados en documentos NCL es Lua (elementos <media> de tipo “application/x-ginga-NCLua”).

NCLua proporciona una librería “nclCommand” que ofrece un grupo de funciones para soportar comandos de edición de NCL, con la misma semántica descrita en el lenguaje NCL, con una facilidad adicional. Todas las funciones pueden recibir una referencia de tiempo como un parámetro opcional que puede ser usado para especificar el momento exacto cuando el comando de edición debe ser ejecutado.

Este parámetro tiene la misma función fija del tiempo de referencia presente en el evento de flujo.

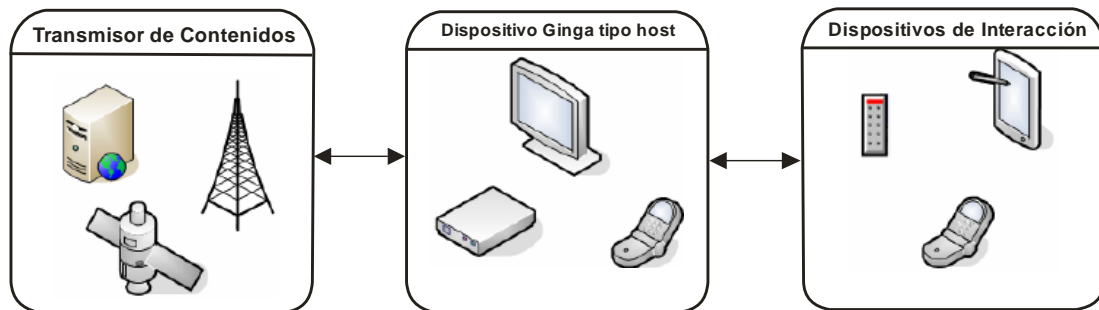
Si el parámetro opcional de tiempo de referencia no se proporciona en el llamado de la función, el comando de edición debe ser ejecutado inmediatamente. Cuando es proporcionado este parámetro puede poseer dos tipos diferentes de valores, con diferentes significados. Si este es un valor numérico, define la cantidad de tiempo en segundos, de cuánto tiempo debe ser pospuesto. Sin embargo, este parámetro puede también especificar el momento exacto para ejecutar el comando en términos de valores absolutos. Para eso, este parámetro debe ser una tabla con los siguientes campos: year (cuatro dígitos), month (1-12), day (1-31), hour (0-23), min (0-59), sec (0-61), y isdst (daylight saving flag, un boolean).

NCLua también proporciona una librería “nclExtended” que ofrece un grupo de funciones para arrancar, parar, pausar o resumir una interface definida en NCL para ejecutar objetos multimedia NCLua. Esta función resultante puede ser usada como condición, evaluada por el Programa Interprete NCL, para activar acciones sobre otros objetos NCL del mismo documento. Además del evento comandado de la transición de estado, un código de procedimiento NCLua puede también registrarse como un oyente de las transiciones de estado de algún evento NCL.

Finalmente, dependiendo en la configuración del *middleware*, es posible tener acceso en Lua al mismo API proporcionado por Ginga-J, el ambiente de procedimiento de Ginga, con el fin de tener acceso a algunos recursos del set-top box y facilidades Ginga. El API proporcionado en Lua debe seguir las mismas especificaciones presentadas en el Foro de TVD Brasileira.

### 3.2.2. Ginga – J [16]

En la figura 3.6 se muestra el contexto en el cual se ejecuta la pila de *software* de Ginga-J. En esta figura, la pila de *software* de Ginga-J reside sobre el dispositivo Ginga, que puede ser, por ejemplo, un *set top box*, un aparato de TV digital, un teléfono celular, etc.



**Figura 3.6. Contexto de Ginga-J**

El *software* Ginga-J tiene acceso a flujos de vídeo, audio, datos y otros activos multimedia. Estos flujos son distribuidos por medio aire, recibidos y procesados por las aplicaciones, que pueden presentar los contenidos a los espectadores. El espectador puede interactuar con la aplicación a través de los dispositivos de interacción de entrada y salida adjuntos o asociados con el dispositivo Ginga. El dispositivo Ginga recibirá acciones por parte de los espectadores a través del dispositivo de interacción, como el control remoto o teclado.

En respuesta a la acción del espectador, el dispositivo Ginga presentará una respuesta visual, así como salidas de audio utilizando su propia pantalla y altavoces o pantallas y altavoces de los dispositivos de interacción. Un solo dispositivo puede tener la capacidad de entrada y salida simultáneamente.

Un ejemplo de dispositivo de interacción puede ser una PDA conectada a la plataforma Ginga a través de una red inalámbrica. Con este dispositivo de interacción, un espectador puede enviar comandos a la plataforma a través del teclado de la PDA y las aplicaciones de la plataforma pueden enviar el contenido visual que se presentará en la pantalla de la PDA.

Muchos espectadores pueden interactuar al mismo tiempo con la plataforma Ginga. En este caso, cada espectador puede tener un dispositivo de interacción y la plataforma debe distinguir los comandos enviados por y para cada dispositivo.

### 3.2.2.1. Arquitectura de Ginga-J

El modelo Ginga-J distingue entre entidades de *hardware* o de recursos, *software* del sistema, aplicaciones como se muestra en la figura 3.7.

Las aplicaciones nativas pueden llevarse a cabo sin utilizar las funcionalidades estándar, provistas por el sistema operativo del dispositivo Ginga o por la aplicación Ginga. Las aplicaciones nativas también pueden utilizar las APIs estandarizados de Ginga-J. Las aplicaciones Xlet deben utilizar las APIs estandarizadas por Ginga-J.

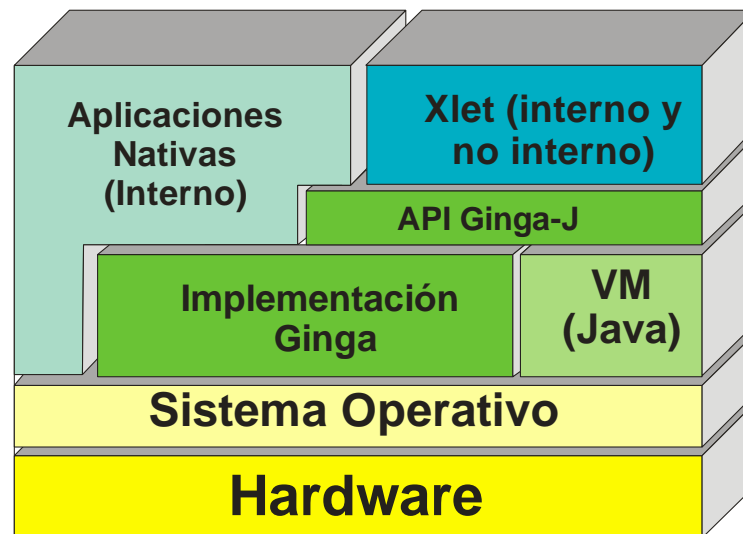


Figura 3.7. Arquitectura y ambiente de ejecución de Ginga-J

En general, Ginga desconoce de las aplicaciones nativas, pero no está limitado para: subtítulos para sordos, acceso condicional (CA) a los mensajes del sistema, menús del receptor, y guías electrónicas nativas de programas.

Las aplicaciones nativas pueden tener prioridad sobre las aplicaciones Ginga. Como por ejemplo, los subtítulos para sordos y los mensajes de emergencia tendrán prioridad sobre el sistema Ginga.

### 3.2.2.2. Especificaciones de Ginga-J

Ginga-J es parte de procedimiento del *middleware* Ginga, basado en tecnologías Java (algunas APIs Java y su máquina virtual), que incorpora varias innovaciones, pero manteniendo compatibilidad con la mayoría de *middlewares* de TV digital actuales, desde que se unió a la norma GEM.

El estándar Brasileño de TV Digital se creó considerando la convergencia digital como un requisito obligatorio, ya que el estándar de modulación empleado por el Sistema Brasileño tiene como objetivo transmitir la señal de TV digital de manera simultánea a diferentes dispositivos tales como receptores fijos de televisión de alta definición y definición estándar, dispositivos móviles y portátiles, como teléfonos celulares.

La especificación Ginga-J también incluye soporte para la comunicación con dispositivos que utilizan *Bluetooth*, *Wi-Fi*, infrarrojo, Línea de Poder, Ethernet o cualquier tecnología de red normalmente utilizadas. Las aplicaciones pueden proporcionar soporte para la interacción de múltiples usuarios, ya el receptor de TV puede ser accesado por diferentes dispositivos simultáneamente.

### 3.2.2.3. El API Ginga-J

Algunos requisitos importantes identificados en el contexto Brasileño no cumplían algunas de las funciones correspondientes a las definiciones de *middleware* internacionales establecidas. Con el fin de satisfacer los requerimientos específicos de Brasil y al mismo tiempo mantener la compatibilidad internacional con el API de GEM, Ginga se basa en tres grupos de APIs llamados: Verde, Amarillo y Azul (Figura 3.8). Las APIs Verde de Ginga son las APIs compatibles con GEM, Las APIs Amarillo fueron aplicaciones compuestas para cumplir los requisitos específicos de Brasil que pueden ser implementadas mediante el uso de un software de adaptación utilizando las APIs Verde. Las APIs Azul no son compatibles con las APIs de GEM. De esta manera, las aplicaciones que solo utilizan las APIs Verde pueden ser ejecutadas en los *middlewares* Ginga, MHP, OCAP, ACAP y ARIB SDT-23. Las aplicaciones que utilizan las APIs Verde y Amarillo solo pueden ser ejecutadas en MHP, ACAP, OCAP y ARIB SDT-23, si el

software de adaptación es transmitido y ejecutado conjuntamente a la aplicación. Las aplicaciones que utilizan las APIs Azul solo se ejecutarán en ambientes del *middleware* Ginga.

El conjunto de APIs Verde, está compuesto por los paquetes Sun JavaTV, DAVIC, HAVi y DVB, todos incluidos en el marco de especificaciones GEM. El conjunto de APIs Amarillo está conformado por el API JMF 2.1, que es necesario para el desarrollo de aplicaciones avanzadas, con captura de sonido por ejemplo, una extensión de la API de Presentación de GEM, con funcionalidades para soportar las especificaciones de flujo de video definidas en el estándar Ginga-J, una extensión para la API del canal de retorno de GEM, que permite el envío de mensajes asíncronos; y una extensión de la API de Servicios de Información del ISDB ARIB SDT-23. El conjunto de APIs Azul está compuesto por un API de integración de dispositivos, que permite al receptor de TV digital la comunicación con cualquier dispositivo con una interfaz compatible (conexión por cable, como Ethernet o PLC, o redes inalámbricas como infrarrojo o *Bluetooth*), que puede ser utilizada como un dispositivo de entrada o de salida, una API multiusuario, que utiliza la API de integración de dispositivos para permitir que varios usuarios puedan interactuar simultáneamente con aplicaciones de TV digital; una API puente a NCL, que permite el desarrollo de aplicaciones Java que contengan aplicaciones NCL.

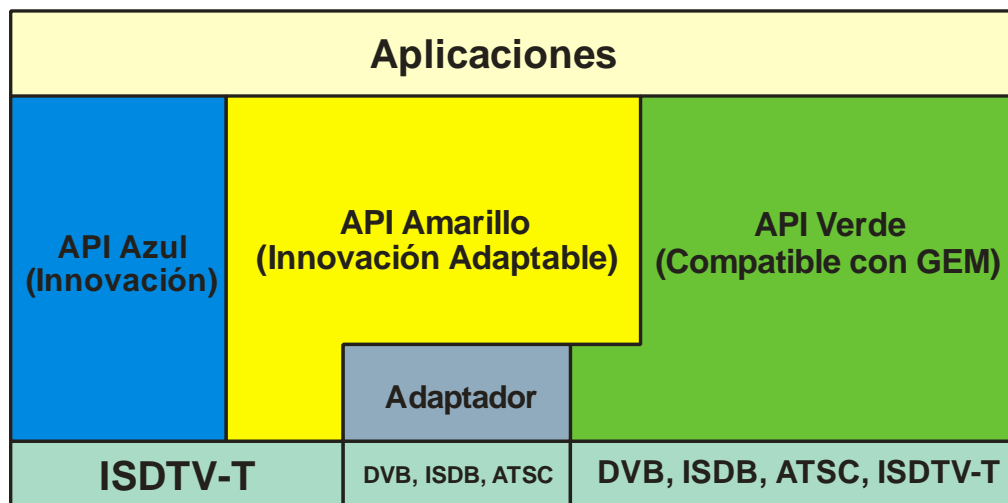


Figura 3.8. APIs de Ginga-J (Azul, Amarillo y Verde)

### 3.2.2.4. Implementación de Referencia de Ginga-J

Una implementación de referencia de Ginga-J está siendo desarrollada para validar las APIs y las especificaciones de Ginga-J. La implementación del sistema ahora tiene más de 800 mil líneas de código fuente. El desarrollo de la implementación de referencia utiliza actualmente plataformas Intel-Kalaheo como banco de pruebas.

La implementación de referencia de Ginga-J fue diseñada con el fin de ser el *middleware* de procedimiento para un receptor universal de TV digital, ya que incorpora tres características importantes:

- **Es multi-red.-** La especificación Ginga define una API de sintonización que es compatible con todas las redes utilizadas actualmente para la transmisión de TV digital (terrestre, cable, satélite e IPTV).
- **Es multi-sistema.-** El Sistema ISDTV-T se adhiere a la especificación del Servicio de Información ISDB ARIB B.10, por lo que la especificación Ginga-J define un API de Servicio de Información compatible con ARIB B.23. Sin embargo, la implementación de referencia de Ginga-J incluye la tabla de procesamiento del Servicio de Información de DVB, ATSC y ARIB, utilizando esta API de Servicio de Información como la salida de datos.
- **Es de aplicación compatible.-** Ya que la especificación Ginga-J, define el grupo de APIs Verde, la implementación de referencia Ginga-J es capaz de ejecuta la mayoría de las aplicaciones diseñadas para ejecutarse sobre los *middlewares* compatibles con GEM.

#### 3.2.2.4.1. Componentes de Software de Ginga-J

La implementación de referencia de Ginga-J utiliza un enfoque basado en componentes de software. Este enfoque facilita la evolución funcional temporal del *middleware*, permitiendo la incorporación de nuevas funcionalidades a través de la adición de nuevos componentes, y también permite la reutilización de algunos de los componentes en otras implementaciones de *middlewares*, por ejemplo, para PDAs, teléfonos móviles, y receptores de TV de alta calidad (con



muchos recursos, más costosos), receptores de baja calidad (con menores recursos, más baratos), etc. Aquellos componentes pueden ser reunidos sobre diferentes perfiles, en función de algunas características de la plataforma.

Los principales elementos de la arquitectura de Ginga-J pueden ser agrupados por sus funcionalidades utilizando siete denominaciones:

- **Componentes del acceso de flujo de bajo nivel.-** contiene los elementos responsables para acceder a los flujos de transporte, y para su procesamiento y demultiplexación en los distintos flujos elementales que construyen.
- **Componentes del procesamiento de flujos elementales.-** contiene los elementos responsables del tratamiento de los flujos elementales, su decodificación y facilitación para otros componentes.
- **Componentes de interfaz de usuario.-** contiene los elementos responsables de permitir la interacción del usuario a través de la presentación de elementos audiovisuales, así como la administración de eventos generados de usuario (puede incluir dispositivos externos).
- **Componentes de Comunicación.-** son los elementos responsables de permitir la comunicación entre las aplicaciones que se ejecutan sobre el *middleware*, así como otras aplicaciones.
- **Componentes de Gestión.-** son los elementos responsables de la gestión del *middleware* (gestión de contextos, actualizaciones del *middleware*, etc.) y para la gestión de aplicaciones (control del ciclo de vida, control de fallos, etc.).
- **Componentes de Persistencia.-** elementos responsables del almacenamiento de datos persistentes.
- **Componente de Acceso Condicional.-** elementos responsables de la seguridad del acceso restringido de los contenidos transmitidos por proveedor de contenidos.

#### 3.2.2.4.1.1. Componentes de Acceso de flujo de bajo nivel

- **El Sintonzador:** Es el componente responsable de la selección del canal físico, así como de uno de los flujos de transporte, que está siendo transmitido por el canal seleccionado.
- **Servidor de Información de Flujo:** Es el responsable de identificar cuáles son los flujos elementales presentes en el flujo de transporte seleccionado por el Sintonzador y el proporcionar información relacionada (programación, opciones disponibles de audio y subtítulos, etc.) de los otros componentes del *middleware*. Este componente fue desarrollado para manejar múltiples tipos de flujos pequeños de información, de las normas principales del *middleware*, lo que hace Ginga-J independientemente de la norma de transmisión.
- **Demultiplexor:** Es el componente responsable de proporcionar los flujos elementales presentes en el flujo de transporte de los otros componentes del *middleware*. Estos flujos primarios pueden estar disponibles para cualquier componente Ginga-J, por una aplicación que se ejecuta sobre el *middleware* o algún componente de hardware (por ejemplo el decodificador de audio o vídeo).

#### 3.2.2.4.1.2. Componentes del procesamiento de flujos elementales

- **Controlador de Procesamiento Multimedia:** Es el componente responsable del control del procesamiento de los elementos multimedia y para proporcionarlos a los otros componentes del *middleware*. A través de esta API es posible seleccionar qué flujo de audio y vídeo debe ser decodificado, qué subtítulo se debe utilizar, para iniciar y detener el proceso de decodificación de los archivos multimedia, etc. Este componente contiene analizadores y procesadores multimedia específicos para cada tipo de multimedia soportado. La API de este controlador se basa en JMF 2.1, que es compatible con JMF 1.0, actualmente utilizado por los sistemas de *middleware* compatibles con GEM.

- **Procesador de Flujo de Datos:** Es el componente responsable de acceder, procesar y proporcionar a los otros componentes del *middleware* flujos de datos elementales como: el carrusel de datos, paquetes IP transmitidos a través de broadcast, etc. El Procesador también es responsable de notificar a los otros componentes de eventos como la llegada de aplicaciones, eventos sincrónicos o asincrónicos DSM-CC, etc.

#### 3.2.2.4.1.3. Componentes de interfaz de usuario

- **Controlador de Presentación Multimedia:** Es el componente responsable de la presentación de los flujos multimedia (audio, vídeo, imágenes, etc.).
- **Administrador de Eventos de Usuario:** Es el componente responsable de capturar los eventos generados por el usuario, tales como oprimir una tecla del control remoto, los comandos de un dispositivo de interacción (como un teléfono móvil) y pasar estos eventos a las aplicaciones registradas y/o componentes del *middleware*.
- **Elementos Gráficos:** Son componentes compuestos por los elementos gráficos principales utilizados para desarrollar una aplicación. Estos elementos gráficos incluyen botones, cajas de textos, etc.

#### 3.2.2.4.1.4. Componentes de Comunicación

Canal de Interacción: Es el componente responsable de proporcionar interfaces que puedan ser utilizadas por otros componentes del *middleware* para acceder al canal de interacción, que es un canal de datos bidireccional que puede ser utilizado por las aplicaciones locales para comunicarse con las aplicaciones remotas. Este componente fue desarrollado para soportar múltiples tipos de redes (PTSN, Ethernet, GSM, WiMax, etc.) y también incluye algunas características que permiten la programación de datos para ser enviados en un tiempo específico.

Comunicación entre Aplicaciones: Las APIs proporcionadas por este componente pueden ser utilizadas para permitir que las aplicaciones que se ejecutan sobre Ginga-J puedan comunicarse entre sí.

#### 3.2.2.4.1.5. Componentes de Gestión

- **Gestor de Aplicaciones** : Es un elemento de *software* responsable de: cargar, configurar, instalar y ejecutar las aplicaciones sobre Ginga-J proporcionando una API para controlar el ciclo de vida de la aplicación; identificar y prevenir el fallo de la aplicación; y también para gestionar la utilización de recursos y el control de acceso.
- **Gestor de *Middleware***: Es el componente responsable de actualizar el código del *middleware* en tiempo de ejecución. Esto permite a los componentes del *middleware* ser substituidos por la corrección de errores o debido al cambio o mejora de la funcionalidad. Este componente también es responsable de proporcionar la información relativa al contexto actual del receptor de TV digital. Este contexto se define en términos de capacidad, uso y disponibilidad de los recursos de la CPU, memoria, dispositivos de persistencia, etc.

#### 3.2.2.4.1.6. Componentes de Persistencia

- **Gestor de Perfil**: El principal objetivo de este componente es proporcionar, a todos los componentes del *middleware* o a las aplicaciones que se ejecutan sobre Ginga-J, un conjunto de datos definidos por el receptor de TV digital (preferencias definidas por el usuario). Estas definiciones pueden ser relacionadas con el control parental, autorización de servicios generales, etc.
- **Persistencia**: Es el componente diseñado para permitir a un objeto almacenarse después que el proceso que lo creó termine.

### 3.2.2.4.1.7. Componente de Acceso Condicional

**Modulo de Acceso Condicional:** Es el modulo del *middleware* responsable de controlar el acceso a contenidos restringidos transmitidos al receptor. Estos contenidos pueden ser recibidos tanto desde el canal de broadcast como desde el canal de retorno (IPTV).

### 3.2.3. El Puente entre Ginga-NCL y Ginga-J

El puente bidireccional entre Ginga-NCL y Ginga-J se hace:

- En un sentido, a través de relaciones NCL, definidas en los elementos <link> que hacen referencia a elementos <media> que representan a códigos Xlet (de tipo “*application/x-ginga-NCLet*”) soportados por Ginga-J, y a través de scripts Lua (elementos <media> de tipo “*application/xginga-NCLua*”) que hace referencia a métodos Ginga-J;
- En sentido contrario, a través de funciones Ginga-J que pueden monitorear cualquier evento NCL y también pueden ordenar cambios en los elementos NCL, a través de las relaciones definidas en los elementos <link> o a través de comandos de edición NCL, similar a lo que hace por los objetos NCLua.

### 3.2.4. Núcleo Común de Ginga (Comon-Core) [11]

El núcleo común de Ginga concentra servicios necesarios tanto para Motor de Presentación (declarativo) como para el Motor de Ejecución (de procedimiento). Este subsistema es la interfaz directa con el sistema operativo, haciendo un puente estrecho con el hardware. Aquí es donde se accede al sintonizador de canales, sistema de archivos, terminal gráfico, entre otros.

Está compuesto por los decodificadores de contenido común y por procedimientos para obtener contenidos transportados en flujos de transporte MPEG-2 y a través del canal de interactividad. Los Decodificadores de contenido común sirven tanto a las aplicaciones de procedimiento como a las declarativas que necesiten

decodificar y presentar tipos comunes de contenidos como PNG, JPEG, MPEG y otros formatos. El núcleo común de Ginga debe obligatoriamente también ser compatible con el modelo conceptual de exhibición, como se describe en la norma ABNT NBR 15606-1. En la Figura 3.5 se muestran los componentes básicos del Núcleo Común descritos a continuación:

- **El Sintonizador:** Es el modulo responsable de sintonizar un canal, seleccionando un canal físico y los flujos de transporte que están siendo enviados por este canal.
- **Filtros de Selección:** Una vez sintonizado el canal, el *middleware* debe ser capaz de acceder a partes específicas del flujo de transporte. Para esto, existe un Filtro de Selección, el mismo que es capaz de buscar en el flujo, la parte exacta que las APIs necesitan para su ejecución. Funcionando exactamente como un filtro, dejando pasar apenas la información requerida por la API.
- **Procesador de Datos:** Es el elemento responsable de acceder, procesar y transferir los datos recibidos por la capa física. También es responsable de notificar a los otros componentes, sobre cualquier evento que se ha recibido.
- **Persistencia:** Ginga es capaz de guardar archivos, incluso después que ha finalizado el proceso que los creó, para que este pueda ser abierto en otra ocasión.
- **Administrador de Aplicaciones:** Es el módulo responsable de cargar, configurar, inicializar y ejecutar cualquier aplicación en cualquier entorno ya sea declarativo o de procedimiento. También es responsable de controlar el ciclo de vida de las aplicaciones, eliminarlas cuando sea necesario, además de controlar los recursos utilizados por esas APIs.
- **Adaptador Principal de A/V:** Con el Adaptador Principal de A/V, las aplicaciones consiguen ver el flujo de audio y vídeo. Esto es necesario cuando una aplicación necesita controlar sus acciones, de acuerdo con lo que se está transmitiendo.

- **Administrador de Gráficos:** Las normas del *middleware* definen como se presentan al usuario las imágenes, videos, datos, etc., administrando las presentaciones de la misma manera que está definida en el estándar ARIB [ARIB B-24, 2004].
- **Administrador de Actualizaciones:** Es el componente que gestiona las actualizaciones del sistema controlando, descargando las actualizaciones del *middleware* siempre que sea necesario, para corregir los errores encontrados en versiones anteriores. Esto de ser hecho en tiempo de ejecución, sin perturbar el uso normal de la TV por parte del usuario.
- **Reproductor de Archivos Multimedia:** Son las herramientas necesarias para presentar los archivos multimedia recibidos, como por ejemplo archivos de tipo MPEG, JPEG, TXT, MP3, GIF, HTML, etc.
- **Interface de Usuario:** Este módulo es responsable de captar e interpretar los eventos generados por los usuarios, tales como, comandos del control remoto y notificar a los otros módulos interesados.
- **Administrador de Contextos:** Es el responsable de captar las preferencias del usuario, notificando a los otros componentes interesados esas preferencias. Esta información puede ser por ejemplo el horario en que el usuario mira la TV, o el bloquear y desbloquear canales, entre otros.
- **Canal de Retorno:** Proporciona la interfaz de las capas superiores con el canal de interacción (o canal de retorno). Además, debe gestionar el canal de retorno de modo que los datos sean transmitidos cuando el canal esté disponible o forzar la transmisión en caso de que el usuario o una aplicación tengan definido un horario exacto.
- **Acceso Condicional:** Este componente está encargado de restringir contenidos inapropiados recibidos por los canales de programación, proporcionando así seguridad para el *middleware*.

## CAPÍTULO IV

### LENGUAJE NCL

#### 4.1. Introducción al NCL

En este capítulo se describirá y estudiara el lenguaje NCL (*Nested Context Language*) versión 3.0, en donde se presentaran y explicaran los elementos que intervienen en el desarrollo y construcción de documentos hipermedia, con sincronización entre archivos multimedia y la interacción del usuario.

Un Documento hipermedia generalmente está compuesto de nodos (*nodes*) y enlaces (*links*) (Figura 4.1).

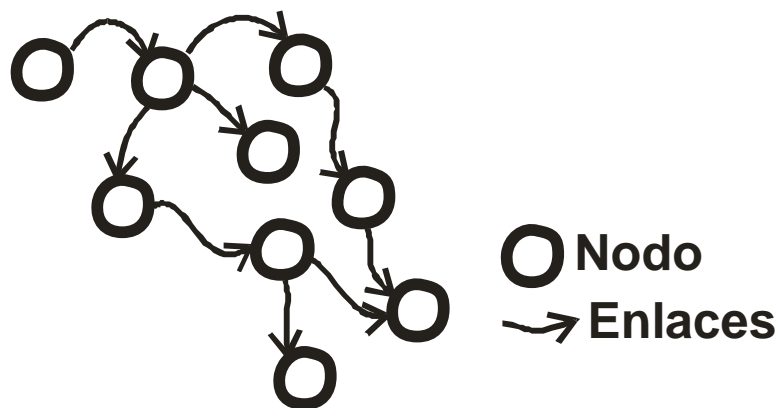


Figura 4.1. Nodos e enlaces de un documento hipermedia común

El Modelo de Contextos Anidados - NCM (*Nested Context Model*), modelo subyacente del lenguaje NCL, va más allá de la gráfica anterior para representar



hipermedia. En NCM, los elementos multimedia pueden ser anidados, permitiendo segmentar y estructura el documento hipermedia según sea necesario. Esto se hace a través de nodos de composición, que son nodos compuestos como los que se muestran en la figura 4.2. Un nodo compuesto también es llamado de contexto.

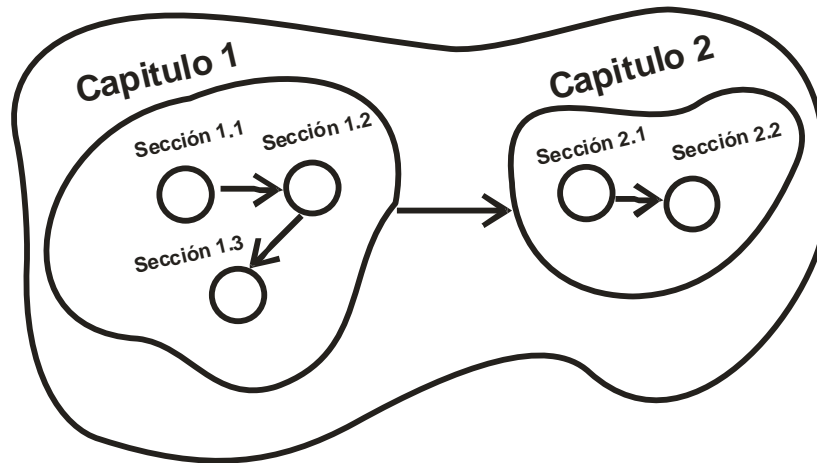


Figura 4.2. Nodos, enlaces y nodos de composición

Por lo tanto, en NCM, un nodo (*node*) puede ser de dos tipos:

- Nodo de contenidos o de archivos multimedia: asociados a un elemento de archivos multimedia tales como video, audio, imagen, texto, aplicación, etc.  
o
- Nodo de composición o contexto (nodo compuesto o de contexto, del cual un switch es un caso particular, como se verá más adelante).

En el caso de la Figura 4.2, el capítulo 1 y capítulo 2 son contextos (nodos de composición), mientras que cada una de las secciones, es un nodo de contenido.

El NCM al igual que todo lenguaje declarativo está basado en un modelo conceptual de datos, en el que se representa los conceptos estructurales de los datos así como los eventos relacionados a los mismos. Además en él, se definen las reglas estructurales y las operaciones sobre los datos de modo que se pueda manipular y actualizar las estructuras.

Para ayudar a la construcción de documentos hipermedia siguiendo el modelo NCM, se desarrollo el lenguaje NCL, el que se utilizará en este estudio, para la construcción de documentos hipermedia. Los elementos del lenguaje se presentaran en las siguientes secciones [19].

## 4.2. Estructura de un Documento Hipermedia

Para construir un documento hipermedia, es necesario definir qué se va a reproducir, dónde (es decir, la región de la pantalla que se va utilizar), cómo (por ejemplo, el volumen, con o sin borde, con qué reproductor) y cuándo (antes o después de presionar un botón, que archivo multimedia será reproducido).

### 4.2.1. ¿Qué reproducir?

En general, lo primero que se debe considerar cuando comenzamos a estructurar un programa audiovisual interactivo es su contenido. Dicho contenido está representado a través de nodos multimedia. Para un mejor entendimiento se representaran en este trabajo gráficamente los nodos multimedia a través de círculos, como lo ilustra la Figura 4.3.

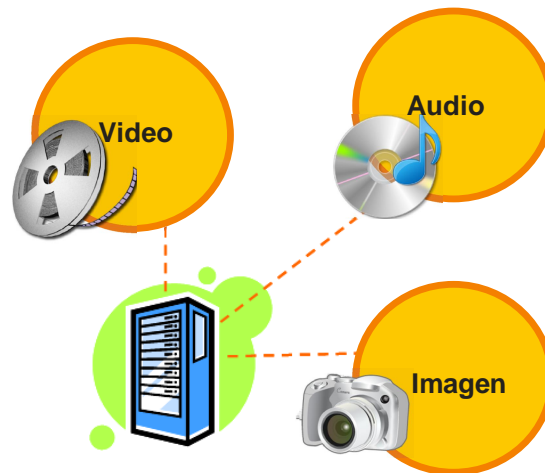


Figura 4.3. Representación de Archivos Multimedia

Cada nodo multimedia se define dentro de un contexto. En NCL, el cuerpo (body), es el elemento del contexto que contiene todos los nodos en el documento, que son los nodos multimedia o contextos. En la Figura 4.4 se muestra un documento

con cuatro nodos multimedia, tres de los cuales están dentro de de un contexto (ctx1) anidado al cuerpo (body).

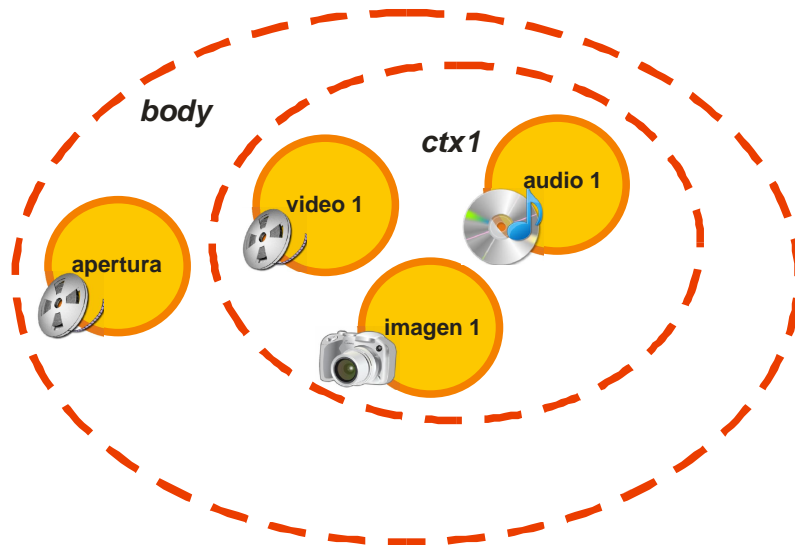


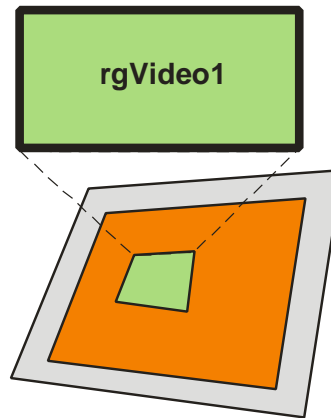
Figura 4.4. Representación de nodos multimedia y su composición

#### 4.2.2. ¿Dónde reproducir?

A medida que se define el contenido de un programa, también se debe comenzar a definir las áreas en dónde cada uno de los archivos multimedia (video, audio, imagen, etc.) se mostraran en la pantalla, por medio de los elementos llamados regiones.

Una región básicamente indica la posición y las dimensiones de un área donde un archivo multimedia se presentará, en otras palabras, una región sirve para inicializar la posición de los nodos multimedia en una ubicación específica.

En este trabajo, se representará a una región gráficamente por rectángulos, como se ilustra en la Figura 4.5.



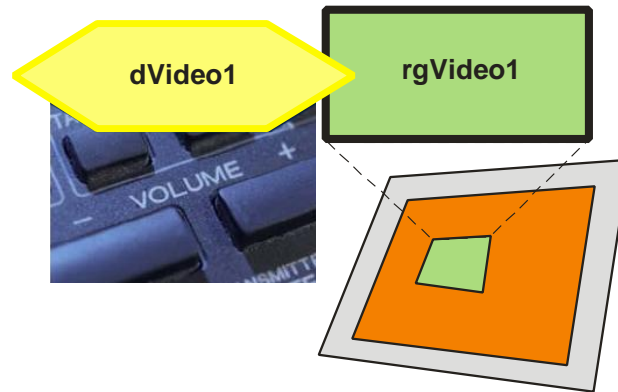
**Figura 4.5. Representación de una Región**

Es importante observar que una región define dónde los archivos multimedia podrán ser mostrados, mas no está asociada a un archivo multimedia en particular. Esta asociación se la realiza a través de un descriptor, los mismos que se revisaran más adelante.

#### **4.2.3. ¿Cómo reproducir?**

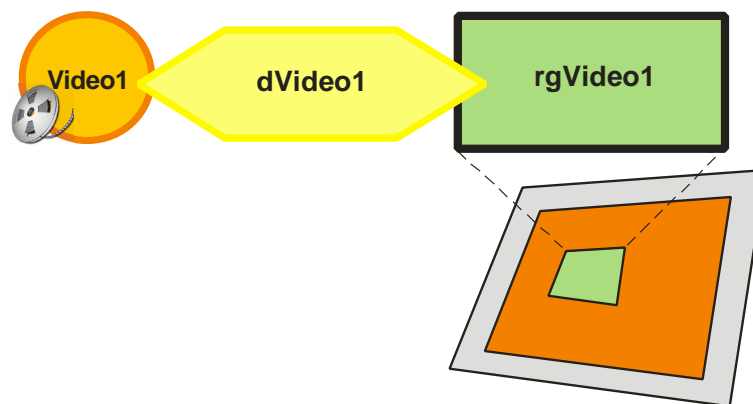
La asociación de un archivo multimedia a una región se define a través de un descriptor. Los descriptores también se utilizan para definir la forma de cómo un archivo multimedia debe ser presentado. Por ejemplo, un descriptor de un archivo de audio puede ajustar su volumen, o el de una imagen, definir el su nivel de transparencia, y el de un texto, podrá definir si este, será presentado visualmente o será leído por un sintetizador de voz, etc.

Al definir un descriptor, es necesario definir la región a la que está asociada (Figura 4.6). Todos los archivos multimedia que utilizan dicho descriptor están asociados con la región correspondiente.



**Figura 4.6. Representación de un descriptor asociado a una región**

Incluso si no se quiere cambiar la forma de cómo un archivo multimedia será presentado, es necesario un descriptor para asociar el archivo multimedia a la región donde debe ser presentado. En la Figura 4.7 se ilustra un descriptor dVideo1 utilizado por un archivo multimedia que se será presentado en la región rgVideo1, sin ningún cambio en la forma de cómo será presentado.



**Figura 4.7. Representación de un descriptor asociado a un archivo multimedia que será presentado en una región**

#### 4.2.4. ¿Cuándo reproducir?

Para definir cuál es el primer nodo del documento a ser presentado, se debe crear una puerta dentro del contexto de este nodo. En caso de existir más de una

puerta dentro del contexto, los nodos mapeados por todas las puertas son iniciados en paralelo.

El identificar por donde un documento puede comenzar a ser presentado es apenas una característica específica de una puerta. De hecho, las puertas son necesarias para dar acceso a los nodos (son nodos de archivos multimedia o contextos) internos a un contexto cualquiera, y no sólo del cuerpo. En la Figura 4.8, el nodo video1 del contexto ctx1 sólo puede ser accesado fuera del contexto ctx1, a través de la puerta pVideo1, mientras que los nodos audio1 e imagen1 no pueden ser accesados fuera del contexto ctx1.

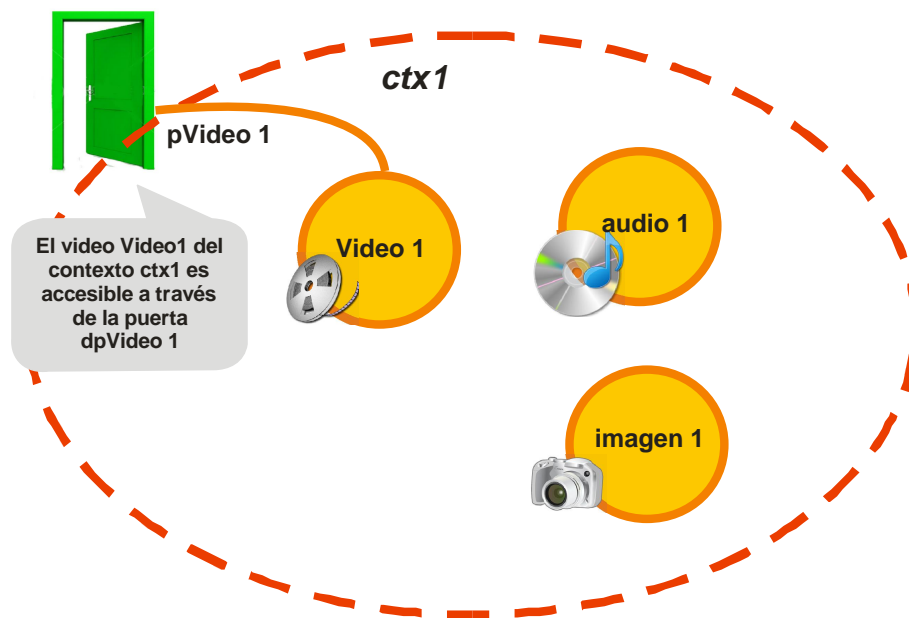


Figura 4.8. Puerta de un nodo de composición





Para definir cuando un nodo multimedia será presentado en relación a otros, se crean enlaces, que son utilizados para establecer el sincronismo entre los nodos y para definir la interactividad del programa [18].

### 4.3. Estructura de un Documento NCL

Un documento NCL es un archivo escrito en XML. Todo documento NCL posee la siguiente estructura:

- Un encabezado de archivo NCL (líneas 1 y 2);
- Una sección de encabezado de programa (sección head, líneas 3 – 13), donde se definen las regiones, los descriptores, los conectores y las reglas utilizadas por el programa;
- Un cuerpo de programa (sección body, líneas 14-17), donde se definen los contextos, nodos multimedia, enlaces y otros elementos que define el contenido y la estructura del programa;
- Por lo menos una puerta que indica por donde el programa comenzara a ser exhibido (port plnicio, línea 15); y
- La conclusión del documento (línea 18).

La Tabla 4.1. Presenta la estructura básica de un documento NCL.

<p><b>Encabezado del Archivo NCL</b></p>	<p>1: &lt;?xml version="1.0" encoding="ISO-8859-1"?&gt;</p> <p>2: &lt;ncl id="ejemplo01" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile"</p> <p style="padding-left: 40px;">xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</p> <p style="padding-left: 40px;">xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile</p> <p style="padding-left: 40px;">http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd"&gt;</p>		
<p><b>Encabezado del Programa</b></p>	<p>3: &lt;head&gt;</p>		
<p><b>base de regiones</b></p>	<p>4: &lt;regionBase&gt;</p> <p>5: &lt;!--regiones de la pantalla en donde se presentaran los archivos multimedia --&gt;</p> <p>6: &lt;/regionBase&gt;</p>		
<p><b>base de descriptores</b></p>	<p>7: &lt;descriptorBase&gt;</p> <p>8: &lt;!-- descriptores que define como los archivos multimedia</p>		

		serán presentados -->		
	<b>base de conectores</b>	9: </descriptorBase>		8
		10: <connectorBase> 11: <!-- conectores que define como los enlaces son activado y las acciones que ellos ejecutan --> 12: </connectorBase>		
		13: </head>		
	<b>Cuerpo del Programa</b>	14: <body>		
	<b>Puerta de entrada en el programa</b>	15: <port id="plnicio" component="ncPrincipal" interface="ilnicio"/>	5	
	<b>contenido del programa</b>	16: <!-- contextos, nodos multimedia y sus anclas, enlaces y otros elementos -->	6 7 4	
		17: </body>		
	<b>Fin</b>	18: </ncl>		

**Tabla 4.1. Estructura Básica de un documento NCL**

Generalmente, los pasos para construir un documento NCL deben definir:

- 1) Los encabezados básicos del archivo NCL y del Programa;
- 2) Las regiones de la pantalla en donde se presentaran los elementos visuales (región Base);
- 3) Como y donde los nodos multimedia serán presentados, a través de descriptores (descriptorBase);
- 4) El contenido (nodos multimedia - media) y la estructura (contextos - contex) del documento (sección body), asociados a los descriptores;



- 5) La puerta de entrada al programa, apuntando al primer nodo que va a ser presentado, así como las puertas para los contextos, con el propósito de desarrollar enlaces entre contextos y nodos multimedia (port);
- 6) Anclas para los nodos multimedia, con el propósito de construir los enlaces entre nodos multimedia (area y attributte);
- 7) Enlaces para el sincronismo e interactividad entre los nodos multimedia y contextos (link); y
- 8) Los conectores que especifica el comportamiento de los enlaces del documento (connectorBase).

#### 4.3.1. Regiones

Una región (region) se define como un área en el dispositivo de salida, donde un nodo multimedia puede ser presentado. Para organizar el diseño de las distintas partes del documento hipermedia, las regiones pueden estar anidadas. En NCL, las regiones son definidas en el encabezado de programa (<head>), en la sección de las regiones base (<regionBase>).

Todo documento NCL posee por lo menos una región, que define la dimensión y las características del dispositivo donde uno o más nodos multimedia serán presentados. Una región sirve para analizar la posición de los nodos multimedia en un lugar específico [18].

Por ejemplo si creamos dos regiones, a la primera la denominaremos rgTV, en la que definiremos las dimensiones de la pantalla de la TV, y la segunda rgVideo1, que no servirá para presentar un video en un lugar específico de la TV (asumiendo que las dimensiones del televisor son 1024x576 pixeles y vamos a reproducir un video de 640x480 pixeles de dimensión):

```
<region id="rgTV" width="1024" height="576">  
    <region id="rgVideo1" left="192" top="48" width="640" height="480" />  
</region>
```

En NCL, para una región se definen los siguientes atributos:

- **id\*:** es el identificador único, utilizado para referenciar una región (por ejemplo, los descriptores de los archivos multimedia que serán representados en la región).
- **title (título):** Es el título de una región. En caso de ser una región exhibida como una moldura, este título será el que aparece como título de la ventana correspondiente.
- **left (izquierda):** Hace referencia a la coordenada “x” del lado izquierdo de la región, con relación a la coordenada del lado izquierdo de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).
- **top (tope):** Hace referencia a la coordenada “y” del lado superior de la región, con relación a la coordenada del lado superior de la región (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).
- **right (derecha):** Hace referencia a la coordenada “x” del lado derecho de la región, con relación al lado derecho de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).
- **bottom (base):** Hace referencia a la coordenada “y” del lado inferior de la región, con relación al lado inferior de la región principal (o al borde exterior de la pantalla, en caso de que la región no esté anidada a ninguna otra).
- **width (anchura) y height (altura):** Son las dimensiones horizontal y vertical de una región. Cabe observar que como autor se puede especificar las dimensiones de una región según su conveniencia. Por ejemplo, en ciertos casos puede ser mejor definir los atributos *right*, *bottom*, *width* y *height*. En otros casos, puede ser más apropiado especificar los atributos *top*, *left*, *width* y *height*.
- **zIndex:** Indica la posición de una región en el eje “z” y es utilizado generalmente para indicar, en el caso de existir regiones sobrepuestas, que región se presenta sobre las otras. Las capas con índice mayor serán presentadas sobre las capas de índice menor.

En la Figura 4.9 se ilustra los atributos *top*, *left*, *right*, *bottom*, *width*, *height* e *zIndex*:

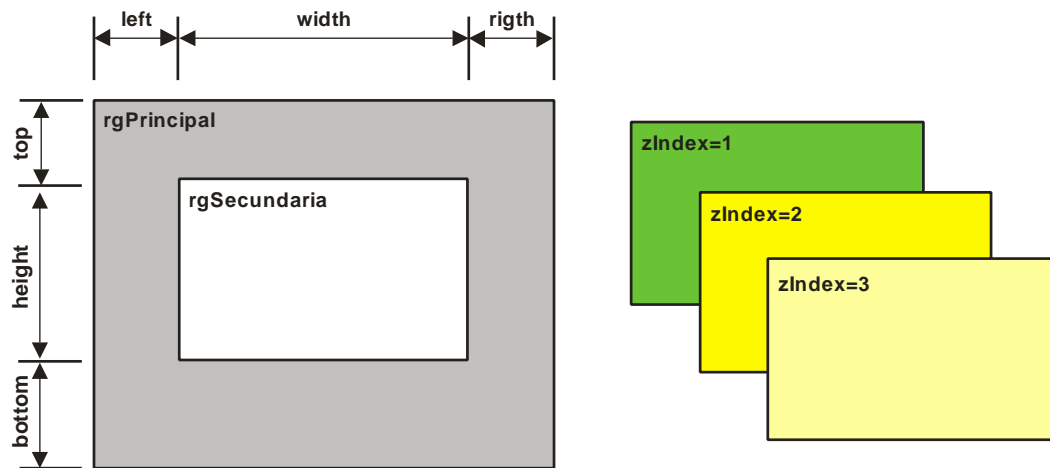


Figura 4.9. Atributos de posición y tamaño de una región

Es importante señalar que si todos los atributos de posición y el tamaño se especifican, los atributos *left* (izquierda) y *width* (ancho) tienen prioridad sobre el atributo *right* (derecho), así como los atributos *top* (superior) y *height* (altura) tienen prioridad sobre el atributo *bottom* (fondo).

Por otra parte, cuando dos o más regiones tienen el mismo valor *zIndex*, y en cada región que se presenta un archivo multimedia, existen dos posibilidades: el caso que un archivo multimedia sea presentado antes que otro, un archivo multimedia que se ha iniciado después se va a sobreponer a la que fue iniciada anteriormente (orden temporal). En caso de que los dos archivos multimedia sean iniciados al mismo tiempo, el orden será seleccionado aleatoriamente por el programa interprete.

La base de regiones posee los siguientes atributos:

- **id:** identificador único de la base de regiones.
- **device:** Define el dispositivo al cual la región está asociada. Puede ser de la forma *systemScreen(i)*, *systemAudio(i)*, de acuerdo a los dispositivos disponibles.

### 4.3.2. Descriptores

Un descriptor es el elemento encargado de definir como será presentado un nodo multimedia, asociándolo a una región (*region*). En NCL, todos los descriptores deben ser definidos en el encabezado del programa (<head>), en la sección llamada base de descriptores (<descriptorBase>). Por lo tanto, todo nodo multimedia que será presentado, debe tener asociado un descriptor [20].

En NCL se definen los siguientes atributos para un descriptor:

- **id\***: identificador único, empleado en las referencias de un descriptor (por ejemplo, los archivos multimedia/contenidos presentados por el descriptor).
- **player**: identifica la herramienta de presentación que se empleara para mostrar el objeto multimedia asociado al descriptor. Este atributo es opcional. Cuando se omite, el programa interprete procura buscar esta herramienta por defecto en función al tipo de objeto multimedia.
- **explicitDur**: determina la duración ideal de un objeto multimedia asociado a un descriptor. El valor de este atributo debe ser expresado en segundos, escribiendo primero el valor numérico seguido de la letra “s” (por ejemplo, 9.9s). Cuando el atributo *explicitDur* no se especifica, el objeto multimedia asociado a este descriptor será presentado con su duración por defecto. Los archivos multimedia como textos e imágenes estáticas poseen una duración infinita por defecto. Para el programa interprete, este atributo no se considera para archivos multimedia continuos; en un programa interprete con soporte de ajuste elástico del tiempo, el valor de ese atributo es usado como referencia para modificar el tiempo de duración del nodo (permitiendo, por ejemplo, que algunos cuadros de un video de 30s dejen de ser exhibidos para que solo dure 29s).
- **region**: Identifica la región asociada a un descriptor. Al utilizar un descriptor, el objeto multimedia será presentado en su región correspondiente. Este atributo solo precisa ser especificado para objetos que poseen un contenido visual a ser presentado.
- **freeze**: identifica lo que sucede después de la presentación del objeto multimedia asociado a un descriptor. En un video, el valor “true” indica que el último cuadro debe ser congelado.

- **focusIndex:** define un índice de navegación para un objeto multimedia asociado a un descriptor. En el caso de no ser definido un índice, el objeto multimedia no podrá recibir el foco de navegación. En un objeto, el foco inicial estará con el menor *focusIndex*. Cabe observar que el valor que se le asigna a un *focusIndex* no es necesariamente un número y en ese caso, “menor” significa lexicográficamente menor.
- **focusBorderColor:** define el color del rectángulo que debe aparecer sobrepuesto en la región de este descriptor cuando el objeto a él asociado recibe el foco. Puede ser uno de los siguientes valores: *white, black, silver, gray, red, maroon, fuchsia, purple, lime, green, yellow, olive, blue, navy, aqua, o teal*.
- **focusBorderWidth:** define el espesor en pixeles del rectángulo que debe aparecer sobrepuesto a la región de este descriptor cuando el elemento a él asociado recibe un foco. En caso de que este sea igual a 0, ningún borde será presentado. Un valor positivo indica que el borde estará fuera del contenido del objeto, mientras que un valor negativo indica que el borde se formara sobre el contenido del objeto.
- **focusBorderTransparency:** define el porcentaje de transparencia de un borde. Este recibe un valor real entre 0 y 1, donde 0 significa totalmente opaco y 1 significa totalmente transparente.
- **focusSrc:** define un archivo multimedia alternativo que debe ser presentado cuando el elemento asociado a este descriptor esté con el foco.
- **focusSelSrc:** define un archivo multimedia alternativo que debe ser presentad cuando es presionado el botón “Ok” o “Enter” mientras el elemento asociado a este descriptor esté con el foco.
- **selBorderColor:** define un color de borde que debe ser exhibido cuando sea presionado el botón “Ok” o “Enter” mientras el elemento asociado a este descriptor ande con el foco.
- **moveLeft:** identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para la izquierda del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo *focusIndex* del elemento E).

- **moveRight:** identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para la derecha del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo *focusIndex* del elemento E)
- **moveUp:** identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para arriba del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo *focusIndex* del elemento E).
- **moveDown:** identifica el índice de navegación del elemento E que debe recibir el foco si es presionada la flecha para abajo del control remoto mientras el elemento asociado a este descriptor esté con el foco (definido por el atributo *focusIndex* del elemento E).
- **transIn:** define la transición que será ejecutada al iniciar la presentación del objeto multimedia.
- **transOut:** define la transición que será ejecutada al terminar la presentación del objeto multimedia.

En NCL se define aún el siguiente elemento opcional contenido en un elemento descriptor:

- **descriptorParam:** define un parámetro del descriptor como un par <propiedad,valor>. Las propiedades y sus respectivos valores dependen del programa de presentación del archivo multimedia asociado al descriptor.

Cada descriptor puede contener diversos elementos *descriptorParam*, definidos en el formato:

```
<descriptorParam name="nome_do_parametro" value="valor_do_parametro" />
```

Para indicar que archivo multimedia correspondiente debe ser reproducido con un volumen del 90% del máximo:

```
<descriptor id="dVideo1" region="rgVideo1">
  <descriptorParam name="soundLevel" value="0.9" />
</descriptor>
```

El uso de parámetros de un descriptor promueve un alto grado de flexibilidad. Le corresponde a cada programa de presentación de objetos multimedia (player) interpretar esas propiedades de la forma adecuada. Actualmente, no se puede definir parámetros de un descriptor en el Composer. En NCL, están reservados los parámetros que se describen en la Tabla 4.2:

Tipos de	Parámetros	Descripción
<b>Archivos Multimedia</b>		
<b>Objetos con Audio</b>	soundLevel, balanceLevel, trebleLevel, bassLevel	Valores entre 0 y 1. En el caso de soundLevel, 0 = mute; 0.5 = volumen a 50%; y 1 = volumen máximo
	location	Posición del objeto multimedia. Se trata de dos números separados por una coma, en el orden <left,top> en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	size	dimensiones del objeto multimedia. Se trata de dos números separados por una coma, en el orden <width,height> , en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	bounds	posición y dimensiones del objeto multimedia. Se trata de cuatro números separados por una coma, en el orden <left, top, width, height>, en uno de los siguientes formatos: a) números reales entre 0 y 100, seguidos del símbolo de porcentaje; o b) números enteros no negativos que especifiquen un valor en pixeles.
	background	Nombres de colores reservados: white, black, silver, gray, red, maroon, fuchsia, purple, lime, green, yellow, olive, blue, navy, aqua, o teal. También puede ser transparent, para el caso de imágenes con transparencia, como algunos GIFs
	visible	true o false
	transparency	número real entre 0 y 1 indicando transparencia: 0 significa totalmente opaco y 1 significa totalmente transparente
	fit	toma uno de los siguientes valores: fill, hidden, meet, meetBest o

<b>Objetos</b>	slice, donde:
<b>Visuales</b>	<ul style="list-style-type: none"> <li>• fill = redimensiona el contenido del objeto multimedia para que toque todos los bordes de la región;</li> <li>• hidden: si la altura intrínseca del contenido del archivo multimedia es más pequeña que el atributo height, el objeto necesita ser renderizado a partir del tope y rellenar su altura restante con el color de background; si es mayor, el sobrante debe ser cortado. Idéntico para el ancho y la izquierda.</li> <li>• meet = redimensiona el contenido del objeto multimedia manteniendo sus proporciones hasta alcanzar uno de los bordes de la región. Si haya un espacio vacío a la derecha o en la parte de bajo, debe ser llenado con el color del background.</li> <li>• meetBest = semejante al meet, pero el objeto multimedia no es ampliado en más del doble de las dimensiones originales.</li> </ul> <p>slice = redimensiona el contenido del objeto multimedia manteniendo sus proporciones hasta que toda la región sea llenada. Parte del contenido puede ser cortado a la derecha o en la parte inferior del contenido.</p>
scroll	Toma uno de los siguientes valores: none, horizontal, vertical, both o automatic.
style	Localización de un archivo de hoja de estilo.

**Tabla 4.2. Paramentaros que pueden ser utilizados en descriptor, de acuerdo al archivo multimedia.**

Los descriptores no aparecen en las visiones estructural, de disposición o temporal en el Composer. Para verificar si un descriptor fue creado correctamente, se debe consultar la visión textual.

#### 4.3.3. Nodo Multimedia (o Nodo de Contenido)

Un nodo multimedia o de contenido (*media node* o *content node*) define el objeto multimedia propiamente dicho: video, audio, texto, etc. Cada definición de un nodo multimedia debe presentar, además del archivo de origen, el descriptor que regulará la presentación de un objeto específico [17].

Un nodo multimedia posee los siguientes atributos:

- **Id:** identificador único del nodo multimedia, empleado para hacer referencia al nodo (por ejemplo, en las puertas de los nodos de contexto que se dirigen hacia los archivos multimedia)



- **Type:** tipo de archivo multimedia, especificado como MIME, según la tabla 4.3:

Valor de type	Extensión de archivo de atributo src
Text/html	.htm, .html
Text/css	.css
Text/xml	.xml
Image/bmp	.bmp
Image/png	.png
Image/gif	.gif
Image/jpeg	.jpg
Audio/basic	.wav
Audio/mp3	.mp3
Audio/mp2	.mp2
Audio/mpeg4	.mp4, .mpg4
Video/mpeg	.mpeg, .mpg
Application/x-ginga-NCLua	.lua
Application/x-ginga-NCLet	.xlt, .xlet, .class
Application/x-ginga-settings	No tiene archivo asociado (no se definió el atributo <b>src</b> ). Se trata de un nodo de atributos globales para ser utilizado en normas y <i>switches</i> .
Application/x-ginga-time	No tiene archivo asociado (no se define el atributo src)

Tabla 4.3. Tipos de archivo multimedia

- **Src:** fuente del objeto multimedia. Se trata del camino para el archivo multimedia. Ese camino puede ser relativo, a partir del directorio donde se encuentra el archivo NCL, o absoluto, a través de una URI. Las URI válidas son las que se presentan en la tabla 4.4:

Esquema	formato	Uso
<b>File:</b>	///camino_archivo/#id_fragmento	Archivos locales
<b>http:</b>	//id_servidor/camino_archivo/# id_fragmento	Archivos remotos bajados del canal de retorno utilizando el protocolo http
<b>Rstp:</b>	//id_servidor/camino_archivo/# id_fragmento	<i>Streams</i> bajados del canal de retorno utilizando el protocolo rstp
<b>Rtp:</b>	//id_servidor/camino_archivo/# id_fragmento	<i>Streams</i> bajados del canal de retorno utilizando el protocolo rtp
<b>Isdtv-ts:</b>	//id_programa	Streams recibidos del <i>transport stream</i>

Tabla 4.4. URI válidas

- **Descriptor:** identificador del descriptor que controla la presentación del objeto multimedia.
- **Refer:** referencia a otro nodo multimedia previamente definido, como forma de reutilización de nodo (utiliza los atributos del nodo multimedia referenciado, excepto el id)
- **newInstance:** establece si un nodo que se refiere a otro genera una nueva instancia del objeto en el programa intérprete o se reutiliza la instancia previamente creada.

#### 4.3.4. Contextos

El elemento *body* es un caso particular de contexto, representando el documento como un todo. Los Contextos o nodos de composición son utilizados para estructurar un documento hipermedia, los mismos que pueden ser anidados, con el objetivo de reflejar la estructura del documento y ayudarle al autor a organizar de mejor manera los segmentos del programa audiovisual interactivo.

Un contexto es definido de la siguiente forma:

```
<context id=" ctxNome">
```

```
...
```

```
</context >
```

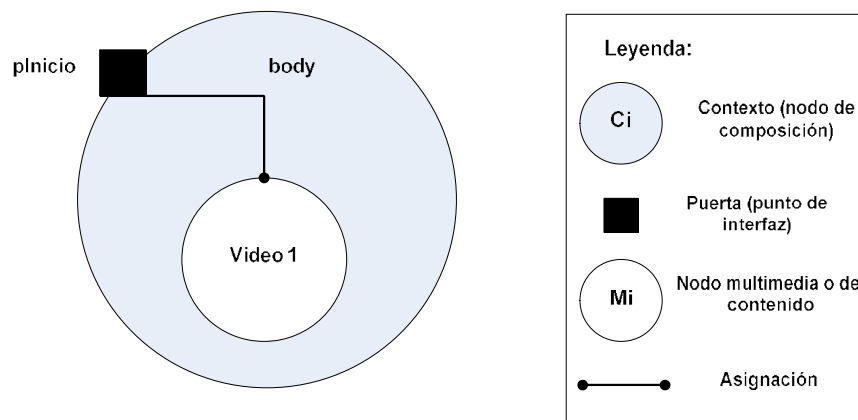
Los atributos de un contexto son:

- **Id:** identificador único de contexto
- **Refer:** referencia a otro contexto previamente definido (utiliza los atributos del contexto referenciado, excepto el id)

El contexto *body* de un documento NCL hereda el id del propio documento.

#### 4.3.5. Puertas

Una puerta (*port*) es un punto de interfaz (*interface point*) de un contexto, que ofrecen acceso externo al contenido (nodos internos) de un contexto. En otras palabras, para que un enlace apunte a un nodo interno al contexto, este contexto debe poseer una puerta que lo dirija hacia el nodo interno deseado. (Figura 4.10).



**Figura 4.10.** Puerta *plnicio* como punto de entrada a un nodo interno de un contexto

En todo documento, debe haber por lo menos una puerta de entrada (*port* en la sección *body*) indicando cual es el nodo multimedia o contexto inicial.

```
<port id="plnicio" componet="video1" >
```

Cabe observar que la puerta *plnicio* del *body* generalmente se asigna al video principal *video1*. Esto significa que, al iniciar el documento hipermedia, el programa intérprete seguirá la puerta *plnicio*, la misma que se dirige hacia el nodo de contenido *video1*, que será entonces presentado [18].

Una puerta posee los siguientes atributos:

- **Id:** identificador único de la puerta, utilizado en las referencias para la misma.  
**Component:** nodo de media o contexto al cual la puerta se asigna.  
 En caso de que el **component** sea un contexto, se debe definir el atributo **interface**, haciendo la asignación a una puerta o ancla de aquel contexto.  
 En caso de que el **component** sea un nodo multimedia, se puede definir el atributo **interface** como una ancla del nodo. Si el atributo **interface** fuera omitido, todo el nodo será considerado como asignado a aquella puerta.
- **Interface:** nombre del punto de interfaz (puerta) de destino en el contexto o nombre de la ancla de destino en el nodo multimedia o contexto.

#### 4.3.6. Base de Conectores

Todos los conectores son definidos en una base de conectores (elemento <connectorBase>, que posee como único atributo un identificador (id) de base.

Una base de conectores contener los siguientes elementos hijos:

- <causalConnector>: define un conector propiamente.
- <importBase>: permite importar una base de conectores de algún otro archivo.

Una base de conectores puede ser definida conforme las siguientes estructuras:

```
<connectorBase id="menusConectores">
  <importBase .../>
  <importBase .../>
  <causalConnector id="onBeginStar">
    ...
  </causalConnector>
  <causalConnector id=" " >
    ...
  </causalConnector id=" " >
    ...
  <causalConnector id=" " >
    ...
</connectorBase>
```

### 4.3.7. Conectores

En NCM y en NCL, el sincronismo no está hecho por marcas de tiempo (*timestamps*), pero si por mecanismos de causalidad y restricción definidos en los conectores (*connectors*). El conector define los papeles (*roles*) que los nodos de origen y de destino ejercen en los enlaces que utiliza el conector. La Figura 4.11 ilustra un conector con tres nodos.

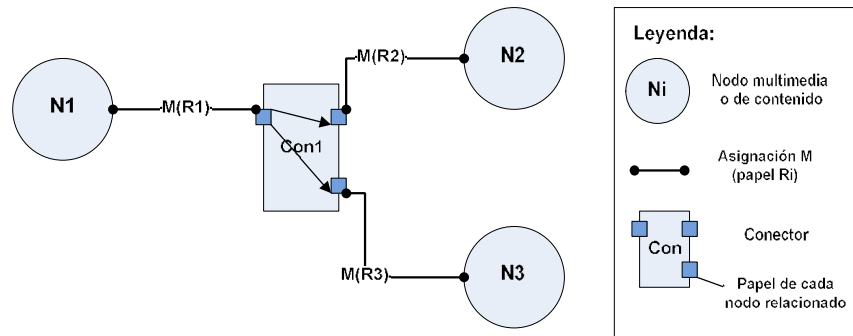


Figura 4.11. Ilustración de un conector con tres nodos.

En NCL 3.0, existe apenas un tipo de conector: o conector causal (*causal Connector*). Un conector causal define las condiciones (*condition*) bajo las cuales el enlace <link> puede ser activado, y las acciones (*action*) que serán realizadas cuando el mismo sea activado. Un conector causal debe poseer por lo menos condición y una acción. Cada condición u acción está asociada a un papel (role), punto de interfaz que participa de las asignaciones del enlace (Figura 4.12).

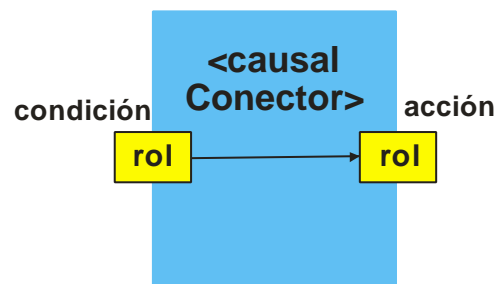


Figura 4.12. Ilustración de un conector causal (elemento <causalConnector>) con papeles (role) de condición y acción.

El elemento `<link>` hace referencia a un `<causalConector>` y define un conjunto de asignaciones (elementos `<bind>` hijos del elemento `<link>`), que asocian cada extremo del enlace (interface de objeto) a un papel de conector utilizado, como ilustra la figura 4.13:

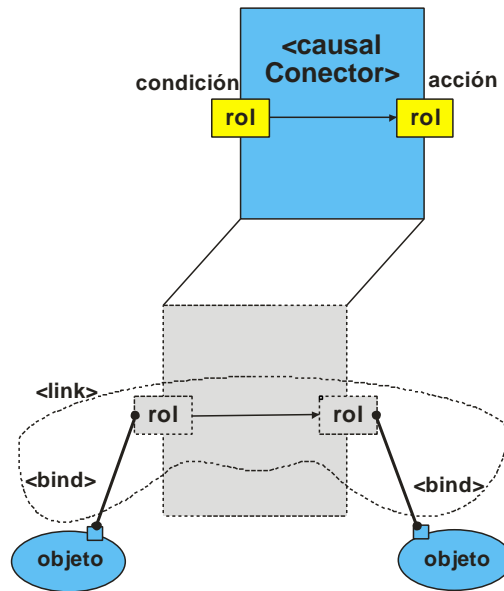


Figura 4.13. Ilustración de un enlace (elemento `<link>`)

Los elementos hijo de un `<causalConector >` son:

- **`<connectorParam>`**: define parámetros de conector cuyos valores deberían ser definidos por los enlaces que utilizan un conector.
- **`<simpleCondition>`** y **`<compoundCondition>`**: definen las condiciones simples o compuestas de activación de un enlace que utiliza un conector;
- **`<simpleAction>`** y **`<compundAction>`**: definen las acciones simples o compuestas que se realizarán cuando un enlace que utiliza un conector sea activado.

Los siguientes papeles de condición son predefinidos (Tabla 4.5):

papel	Descripción (el enlace será activado cuando...)
<b>onBegin</b>	...la presentación del nodo multimedia asociado al papel <b>onBegin</b> será iniciada
<b>onEnd</b>	...la presentación del nodo multimedia asociado al papel <b>onEnd</b> será terminada (naturalmente o por un evento <b>stop</b> )
<b>onAbort</b>	...la presentación del nodo multimedia asociado al papel <b>onAbort</b> será abortada
<b>onPause</b>	...la presentación del nodo multimedia asociado al papel <b>onPause</b> será pausada
<b>onResume</b>	...la presentación del nodo multimedia asociado al papel <b>onResume</b> será retomada (después de una pausa)
<b>onSelection</b>	...una tecla <key> será presionada, o cuando la tecla OK sea presionada mientras un nodo multimedia este corriendo.
<b>onAtribution</b>	...un valor <valor> será atribuido

**Tabla 4.5. Papeles predefinidos de condición**

Los siguientes papeles de acción son predefinidos (Tabla 4.6):

papel	Descripción (cuando el enlace sea activado...)
<b>Start</b>	...inicia la presentación del nodo multimedia asociado al papel <b>start</b>
<b>Stop</b>	...termina la presentación del nodo multimedia asociado al papel <b>stop</b>
<b>Abort</b>	...aborta la presentación del nodo multimedia asociado al papel <b>abort</b>
<b>Pause</b>	...pausa la presentación del nodo multimedia asociado al papel <b>pause</b>
<b>Resume</b>	...retoma la presentación del nodo multimedia asociado al papel <b>resume</b> (en caso de que este en pausa)
<b>Set</b>	...establece un valor <valor> para el ancla asociada al papel <b>set</b>

**Tabla 4.6. Papeles predefinidos de acción**

Tanto los papeles de condición como los de acción están asociados a transiciones de estados en una máquina de eventos, ilustrada en la Figura 4.14.

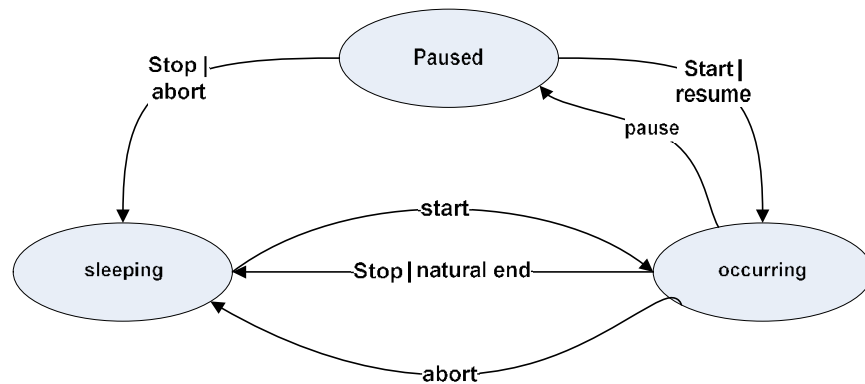


Figura 4.14. Máquina de estados de eventos.

Tanto los papeles de condición “*onBegin*”, “*onEnd*”, “*onAbort*”, “*onPause*” y “*onResume*”, así como los papeles de acción “*start*”, “*stop*”, “*abort*”, “*pause*” y “*resume*” están relacionados a posibles transiciones de estados de presentación de anclas de contenidos como se ilustra en la figura 4.14.

Un papel de condición “*onSelection*” está relacionado a eventos de selección y ligado a interactividad a través de dispositivos de entrada, como el control remoto de la TV [19].

#### 4.3.8. Enlaces

Los enlaces (*links*) asocian nodos a través de conectores (*connectors*), que definen la semántica de la asociación entre los nodos. NCL define los siguientes atributos:

- **Id**: identificador único de enlace
- **xconnector**: identificador de conector asociado al enlace

NCL define los siguientes elementos contenidos en un elemento de tipo enlace:

- **LinkParam**: define un parámetro del enlace como un par <propiedad, valor>. Las propiedades y sus respectivos valores dependen de la definición del conector al cual el enlace está asociado. Un enlace puede contener diversos elementos linkParam.
- **Bind**: indica un componente (nodo multimedia o de contexto) involucrado en el enlace, indicando su papel (role) en el mismo, conforme la semántica del conector.



En algunos casos se debe indicar también el punto de interfaz (interface) del nodo, al cual el enlace está ligado (puerta del contexto o ancla de un nodo multimedia). Un enlace puede contener diversos elementos bind, y debe contener por lo menos un bind para cada papel definido en el conector.

El elemento *bind* puede contener una o más instancias del siguiente elemento como elementos hijos:

- **bindParam:** define un parámetro específico del *bind* como un par <propiedad, valor>. Las propiedades y sus respectivos valores dependen de la definición del conector al cual el enlace está asociado.

A continuación se muestra el esquema básico de conformación de un enlace (<link>):

```
<link id="nombre_de_link" xconnector="id_de_Link">
  <bind role="id_del_papel_de_condicion" component="id_de_un_objeto"
                                     interface="id_de_una_interface"/>
  <bind role="id_del_papel_de_accion" component="id_de_un_objeto"
                                     interface="id_de_una_interface"/>
</link>
```

#### 4.3.9. Anclas

Las anclas son puntos de entrada para los nodos multimedia o contextos. El objetivo de utilizar anclas es emplear segmentos o propiedades de un nodo multimedia o contexto, sea como origen o destino de los enlaces. Una ancla puede ser del tipo ancla de contenido (*content anchor*) o ancla de propiedad (*property anchor*).

Un **ancla de contenido** define un segmento multimedia (intervalo de tiempo y/o región en el espacio) que pudiera ser utilizado como punto de activación de los enlaces. Un segmento de multimedia es una selección contigua de **unidades de información** (*information units*) de un nodo. La definición de esas unidades de información depende del tipo de archivo multimedia presentado por el nodo. Las

unidades de información de un video, por ejemplo, pueden ser los *frames* de video.

Un ancla de contenido es definida como un elemento *area* dentro del elemento *media*.

En el siguiente ejemplo, son definidas tres anclas de contenido para un nodo de video:

```
<media type="video" id="video1" src="media/video1.mpg" descriptor="dVideo1">
  <area id="a01" begin="5s" end="9s"/>
  <area id="a02" begin="10s" end="14s"/>
  <area id="a03" begin="15s" end="19s"/>
</media>
```

NCL define los siguientes atributos de ancla de contenido:

- **id:** identificador único de ancla
- **coords:** coordenadas en pixeles del ancla espacial (atributo válido para archivos multimedia visuales), en el formato "**X,Y,width,height**"
- **begin:** inicio del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)
- **end:** termino del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)
- **dur:** duración del ancla, en segundos, en el formato "99.9s" (atributo válido para archivos multimedia continuos)
- **first:** cuadro/muestra del archivo multimedia definiendo el inicio del ancla (atributo válido para archivos multimedia continuos)
- **last:** cuadro/muestra del archivo multimedia definiendo el fin del ancla (atributo válido para archivos multimedia continuos)
- **text:** texto del ancla en el archivo de origen (atributo válido para archivos multimedia de texto)
- **position:** posición del texto del ancla en el archivo de origen (atributo válido para archivos multimedia de texto)
- **anchorLabel:** identificador del ancla en el archivo de origen, tal como es interpretado por la herramienta de exhibición.

Las **anclas de propiedad** se refieren a propiedades de un nodo de origen o de destino, que pueden ser manipuladas por los enlaces. Ejemplos de propiedades

son: volumen del audio de un nodo de audio o video, coordenadas y dimensiones de exhibición de un nodo multimedia visual, entre otros.

Un ancla de propiedad es definida como un elemento *property* dentro del elemento *media* o *context*. En el siguiente ejemplo se definen cuatro anclas de propiedad para un nodo de video, además de un ancla de contenido:

```
<media type=" video" id="video1" src="media/video1.mpg" descriptor="dVideo1">
  <!--anclas de propiedad que seran manipuladas por los enlaces -->
  <property name="top"/>
  <property name="left"/>
  <property name="width"/>
  <property name="height"/>
  <!--anclas de contenido en el video que debe sincronizar la imagen -- >
  <area id="aVideo1Imagem1" begin="3s" end="8s"/>
</media>
```

#### 4.3.10. Reglas

Las reglas simples de un documento NCL son definidas en la sección *ruleBase*, con base en una propiedad, operador y valor, como en el ejemplo a continuación:

```
<ruleBase>
  <rule id="rEn" var="idioma" comparator="eq" value="en" />
  <rule id="rPt" var="idioma" comparator="eq" value="pt" />
</ruleBase>
```

Los siguientes operadores pueden ser utilizados como comparador en la definición de reglas:

- eq (*equal* – igual a);
- ne (*not equal* –diferente de);
- gt (*greater than* – mayor que);
- ge (*greater than or equal to* – mayor que o igual a);
- lt (*less than* – menor que);
- le (*less than o equal to* – menor que o igual a).

Una forma de almacenar las propiedades que serán utilizadas en las reglas es utilizar el nodo **settings**. Se trata de un nodo de propiedades globales, como en el siguiente ejemplo:

```
<media type="application/x-ginga-settings" id="nodeSettings">
  <property name="idioma" />
</media>
```

#### 4.3.11. Switch

Un *switch* es un contexto con nodos alternativos, es decir, dentro de los cuales apenas uno será activado. La decisión acerca de que nodo será activado es dada por reglas. En la definición de un switch, además de los nodos que lo componen, deben ser definidas las asignaciones de las reglas para esos nodos y sus anclas, tal como se indica en el siguiente ejemplo:

```
<switch id="switchAudioIdioma">
  <bindRule rule="rEn" constituent="audioEn" />
  <bindRule rule="rPt" constituent="audioPt" />
  <media type="audio" id="audioEn" src="media/audioEn.mp3" descriptor="dAudio1" />
  <media type="audio" id="audioPt" src="media/audioPt.mp3" descriptor="dAudio1" />
</switch>
```

En ese ejemplo, la media *audioEn* será reproducida en caso de que la regla *rEn* sea válida, es decir, en el caso de que *idioma* posea el valor "en". Si esa regla no fuera válida, la próxima asignación es valorada, y así sucesivamente, hasta que una regla válida sea alcanzada o hasta terminar el conjunto de asignaciones de aquel *switch* [19].

En el caso de que el *switch* sea compuesto de contextos, es necesario definir uno o más elementos *switchPort* para indicar la puerta de destino de cada asignación de las reglas, como en el siguiente ejemplo:

```
<switch id="switchNivel">
  <switchPort id="pNivel">
    <mapping component="ctxBasico" interface="pBasico" />
```

```
    <mapping component="ctxAvancado" interface="pAvancado" />
</switchPort>
<bindRule rule="rBasico" component="ctxBasico" />
<bindRule rule="rAvancado" component="ctxAvancado" />
<context id="ctxBasico">
    <port id="pBasico" component="videoBasico" />
    <!--nodos e enlaces del context ctxBasico -->
</context>

<context id="ctxAvancado">
    <port id="pAvancado" component="videoAvancado" />
    <!--nodos e enlaces del context ctxAvancado -->
</context>
</switch>
```

## 4.4. Herramientas

### 4.4.1. Composer

Composer es un herramienta de autoría hipermedia desarrollada por el Laboratorio TeleMídia del Departamento de Informática de la PUC-Rio. Con esta herramienta es posible construir programas audiovisuales interactivos con un mediano conocimiento del lenguaje NCL.

Actualmente esta herramienta se encuentra en la versión 2.2.1 NCL 3.0 y es posible encontrarla para diversos sistemas operativos de 32 bits como Microsoft Windows, Linux y Mac OS X.

La herramienta Composer es dinámica y bastante completa ya que posee una gran variedad de formas en la que se puede visualizar el proyecto. En la figura 4.15 se presenta el ambiente grafico de trabajo de Composer.

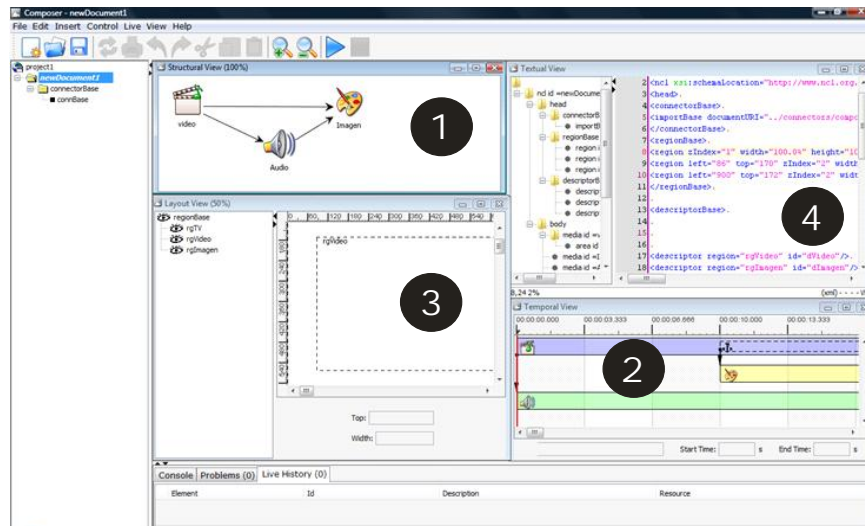


Figura 4.15. Herramienta de autoría Composer

Además en la figura 4.15 se puede observar que un documento hipermedia puede ser considerado a través de diversas visiones. La versión actual de Composer permite al usuario trabajar con: 1) Visión Estructural, 2) Visión Temporal, 3) Visión de Diseño, 4) Visión Textual.

#### 4.4.1.1. Visión Estructural

La visión estructural (*Structural View*) permite visualizar los todos los nodos multimedia que contenga nuestro documento hipermedia así como los enlaces de sincronismo entre los mismos. Además en esta visión es posible crear nodos multimedia, contextos y enlaces, al igual que definir todas sus propiedades (Figura 4.15).

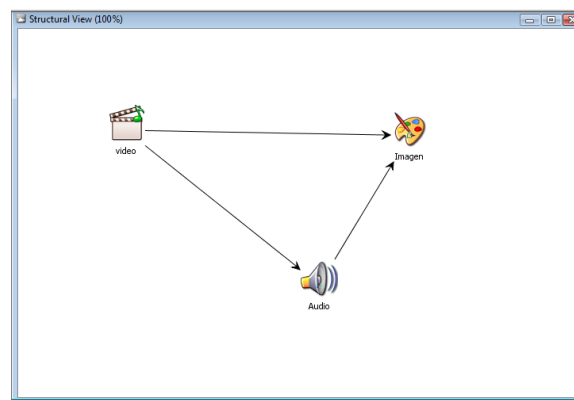


Figura 4.15. Visión Estructural

#### 4.4.1.2. Visión Temporal

La visión Temporal (*Temporal View*) ilustra el sincronismo temporal entre los nodos multimedia y los momentos de interactividad. En la Figura 4.16, se muestra la visión temporal de un documento hipermedia que exhibe una imagen en un determinado intervalo de la duración total de un archivo de video, al igual se puede observar la reproducción simultanea del archivo de video y un de audio.

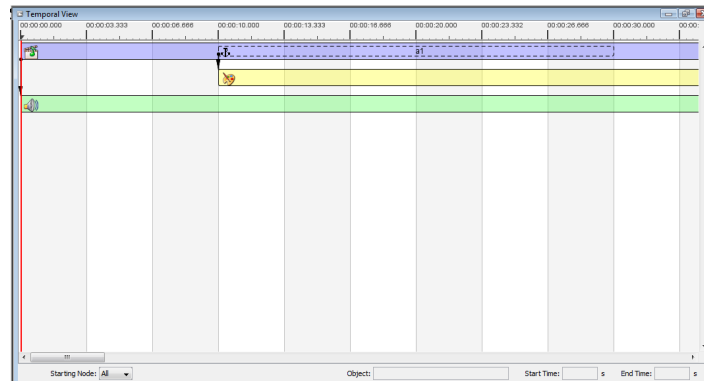


Figura 4.16. Visión Temporal

#### 4.4.1.3. Visión de Diseño

La visión de Diseño (*Layout View*) muestra las regiones de pantalla en donde los archivos multimedia pertenecientes al documento van a ser presentados. En la Figura 4.17 se muestra la visión de diseño de un documento hipermedia que contiene 3 regiones: la de pantalla interna (rgTV), un area para exhibir un video (rgVideo) y un area para presentar una imagen (rgImagen).

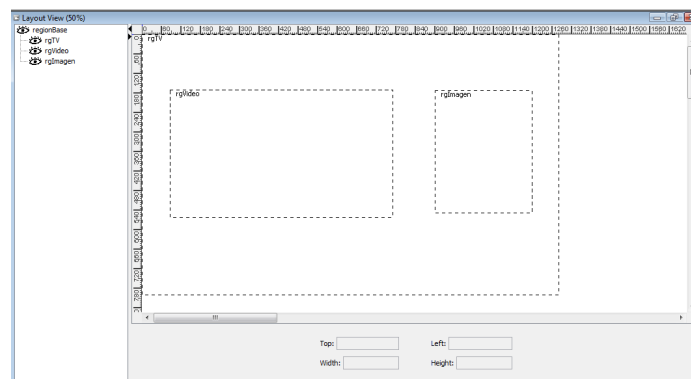


Figura 4.17. Visión de Diseño

#### 4.4.1.4. Visión Textual

La Visión Textual (*Textual View*) muestra el código NCL como se ve en la Figura 4.18. En esta visión, el usuario puede editar directamente el código NCL como en un editor de texto, permitiéndole así poder manipular todos los elementos NCL al igual que sus respectivos atributos que no puede ser determinados a través de las herramientas graficas de Composer, dándole de esta manera un recurso potencial en el momento de desarrollar aplicaciones más complejas.

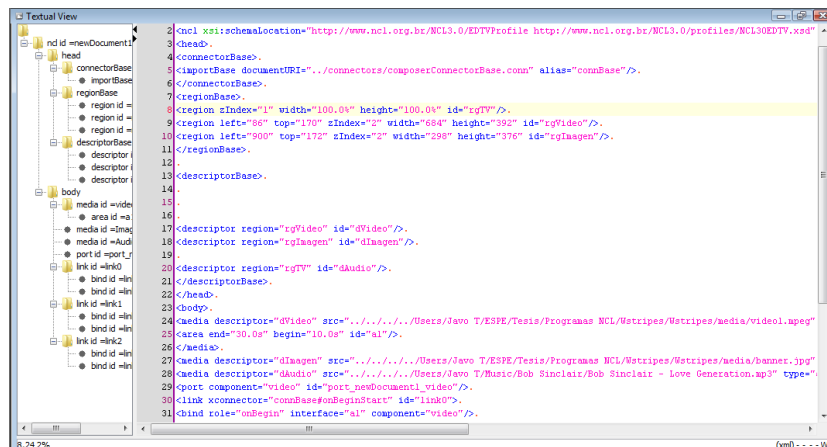


Figura 4.18. Visión Textual

Las Visiones de Composer son sincronizadas, es decir que cuando se realiza alguna modificación en una de ellas, automáticamente el resto de visiones son actualizadas al cambio.

#### 4.4.2. Emulador Ginga-NCL

El emulador Ginga-NCL (Figura 4.19) tiene una implementación Java del middleware Brasileño, al igual que un conjunto de herramientas que permiten simular un *set top box*. Para la ejecución del emulador es necesario tener instalada la Máquina Virtual de Java (JVM). Esta herramienta muestra todos los errores en caso de existir, a través de consola (tanto en ambiente Linux, Windows y MacOS), además esta herramienta posee un control remoto interactivo que puede ser utilizado a través del mouse.



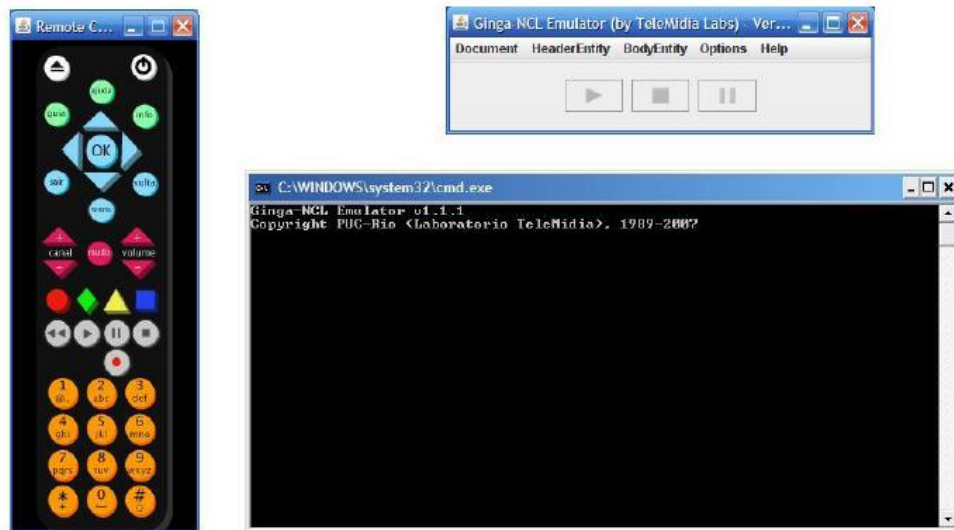


Figura 4.19. Emulador GINGA-NCL

#### 4.4.3. Set Top Box Virtual

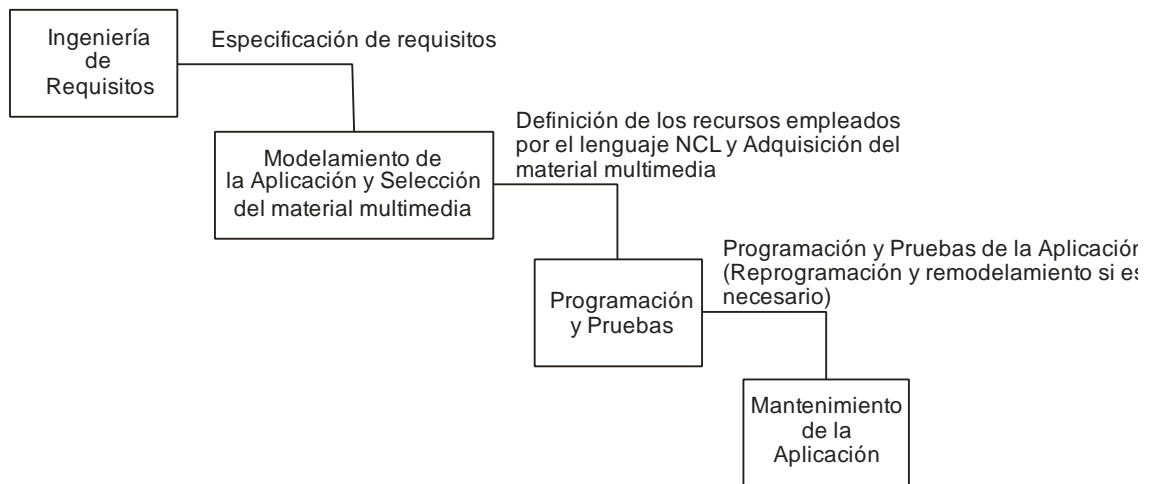
El *Set Top Box* Virtual (VSTB) (Figura 4.20) es en realidad una máquina virtual para VMWare que posee instalada una imagen de Fedora Linux. El VSTB simula fielmente este ambiente, porque con recurso de procesamiento y memoria reducidos como sucede en los STB reales, posee una cantidad muy reducida de recursos y paquetes instalados. Entre los paquetes instalados en la máquina virtual de Fedora está una implementación en C++ del middleware Ginga. Como un kit de herramientas y protocolos para interface, Fedora utiliza DirectFB cuya instalación ocupa cerca de 25 megabytes, este además permite la renderización de videos de una manera eficiente y utiliza una cantidad muy pequeña de memoria. Para la aceleración del hardware utiliza un Frame Buffer vesafb, el cual soporta una resolución máxima de 640 pixeles por 480 pixeles. Para ejecutar una aplicación sobre el VSTB, se debe abrir una conexión utilizando el protocolo Secure Shell (SSH).



Figura 4.20. Set-Top Box Virtual

#### 4.5. Desarrollo de la Aplicación NCL

Como parte final de este estudio, se hace necesaria la aplicación de todos los conceptos e información adquiridos a lo largo del mismo. Para esto, se desarrolló una aplicación que incluye todos los principales elementos del Lenguaje NCL, de tal manera que se verifica su uso, así como el de sus propiedades en el momento de la presentación de la aplicación. Con el fin de optimizar el tiempo del desarrollo de este proyecto, se utilizó un ciclo de vida en cascada como se puede observar en la figura 4.21.



**Figura 4.21. Ciclo de vida en cascada de la Aplicación implementada**

#### **4.5.1. Aplicación NCL INFO-ESPE**

Bajo la necesidad de manipular material multimedia libre de derechos de autor y por ser la Escuela Politécnica del Ejército la institución para la cual se desarrolló este proyecto, se implementó la Aplicación Interactiva denominada “INFO-ESPE”, la que pretende promocionar e informar sobre sus carreras, sedes y periodos de admisión, mientras el espectador observa su video institucional.

##### **4.5.1.1. Requisitos y Modelamiento de la Aplicación**

Esta Aplicación debe permitir una fácil interacción con el espectador, para lo cual deberá presentar una opción de acceso al menú de información que la aplicación presentara bajo la interacción del usuario, para esto se definirá el primer botón interactivo llamado “INFO”, el cual se presentará paralelamente al video institucional.

Una vez que sea presionada la opción “INFO”, se desplegará un menú con tres opciones de interacción; la primera se denominará “ADMISION y REGISTRO” y su acceso se hará a través de la interacción con el botón “ROJO” del control remoto; la segunda se denominará “DEPARTAMENTOS” y su acceso se hará a través de la interacción con el botón “VERDE” del control remoto; finalmente la

tercera opción se denominará “SEDE” y su acceso se hará a través de la interacción con el botón “AMARILLO” del control remoto.

Al seleccionar la opción “ADMISION Y REGISTRO”, se ejecutarán tres eventos principales, el primero es la presentación de una máscara que contiene tres regiones de presentación, el video institucional redimensionará sus propiedades a través del elemento “*bounds*”, se presentarán las primeras páginas de navegación vertical y horizontal respectivamente y por último se presentara la opción de retorno a la opción principal llamada “INFO” y el redimensionamiento a su tamaño original del video, a través de la selección del botón “MENU” del control remoto. Una vez ingresada a esta pantalla de navegación el usuario tendrá la posibilidad de navegar por tres páginas horizontales a través de las teclas de navegación horizontal (“LEFT”, “RIGHT”) del control remoto, permitiéndole al usuario regresar o adelantar de página; la primera página tendrá información sobre “Requisitos para la inscripción”, la segunda página tendrá información sobre “Información para los aspirantes” y la tercera página presentará dos cuadros sobre las “Actividades” que se realizarán con sus respectivas fechas. Otra de las opciones disponibles será la navegación vertical a través de cinco páginas que presentaran información referente a la ESPE como visión, misión, convenios, ubicación de la institución, todo esto con ayuda de las teclas de navegación vertical (“UP”, “DOWN”).

Al seleccionar la opción “DEPARTAMENTOS”, se desplegará un submenú de navegación vertical que presentará diez opciones de selección referentes a los departamentos actualmente definidos en la Escuela (Ciencias de la Computación, Seguridad y Defensa, Ciencias Económicas y Administrativas, Eléctrica y Electrónica, Ciencias Exactas, Ciencias Humanas y Sociales, Lenguas, Ciencias de la Energía y Mecánica, Ciencias de la Tierra y Construcción y Ciencias de la Vida), la navegación sobre este submenú se hará a través de las teclas de navegación vertical (“UP”, “DOWN”), y la selección de cualquiera de ellas a través del botón “OK” del control remoto. Una vez seleccionada cualquiera de las diez opciones, se ejecutaran los mismos tres eventos ya explicados en la primera opción del menú principal, la diferencia será notoria en el número de páginas de navegación horizontal así como su contenido; para este caso se presentaran cuatro páginas de navegación horizontal en las cuales distribuidamente se

presentara información referente a cada uno de los departamentos así como información de los responsables de cada uno de ellos y de las carreras que representan.

La tercera y última opción denominada “SEDES” ejecutara los tres eventos antes descritos, y ahora las páginas de navegación horizontal se incrementaran en un número de veintidós páginas, de las cuales las primeras veinte presentarán fotografías del campus y servicios de las distintas sedes (Sangolquí, Latacunga, Héroe del Cenepa, Idiomas y IASA I y II) y las dos últimas información sobre los contactos de los distintos campus.

Para generar todos estos contenidos, el Departamento de Marketing de la Escuela facilitará el material multimedia, cuyo contenido se respaldará con la información existente en la página web de la ESPE.

#### **4.5.1.2. Programación y Pruebas de la Aplicación**

El desarrollo de la Aplicación se realizó utilizando la herramienta gráfica de desarrollo NCL llamada Composer, y sus pruebas con la ayuda de dos herramientas de presentación: el emulador de GINGA-NCL (Figura 4.19) y el STB Virtual (Figura 4.20), todas estas de distribución libre.

#### **4.5.1.3. Implementación**

La implementación de esta aplicación posee un nivel de complejidad alto, y el código llegó a tener más de 1500 líneas. Además el programa tuvo que ser reestructurado a la idea planteada originalmente, de manera que su ejecución se pudiera realizar en cualquiera de los dos simuladores anteriormente descritos, las modificaciones principalmente que se realizaron fueron en los botones de interacción, debido a que el VSTB no posee todas las opciones de teclas que posee el emulador Ginga-NCL.

4.5.1.3.1. Diagrama NCM de la Aplicación INFO-ESPE

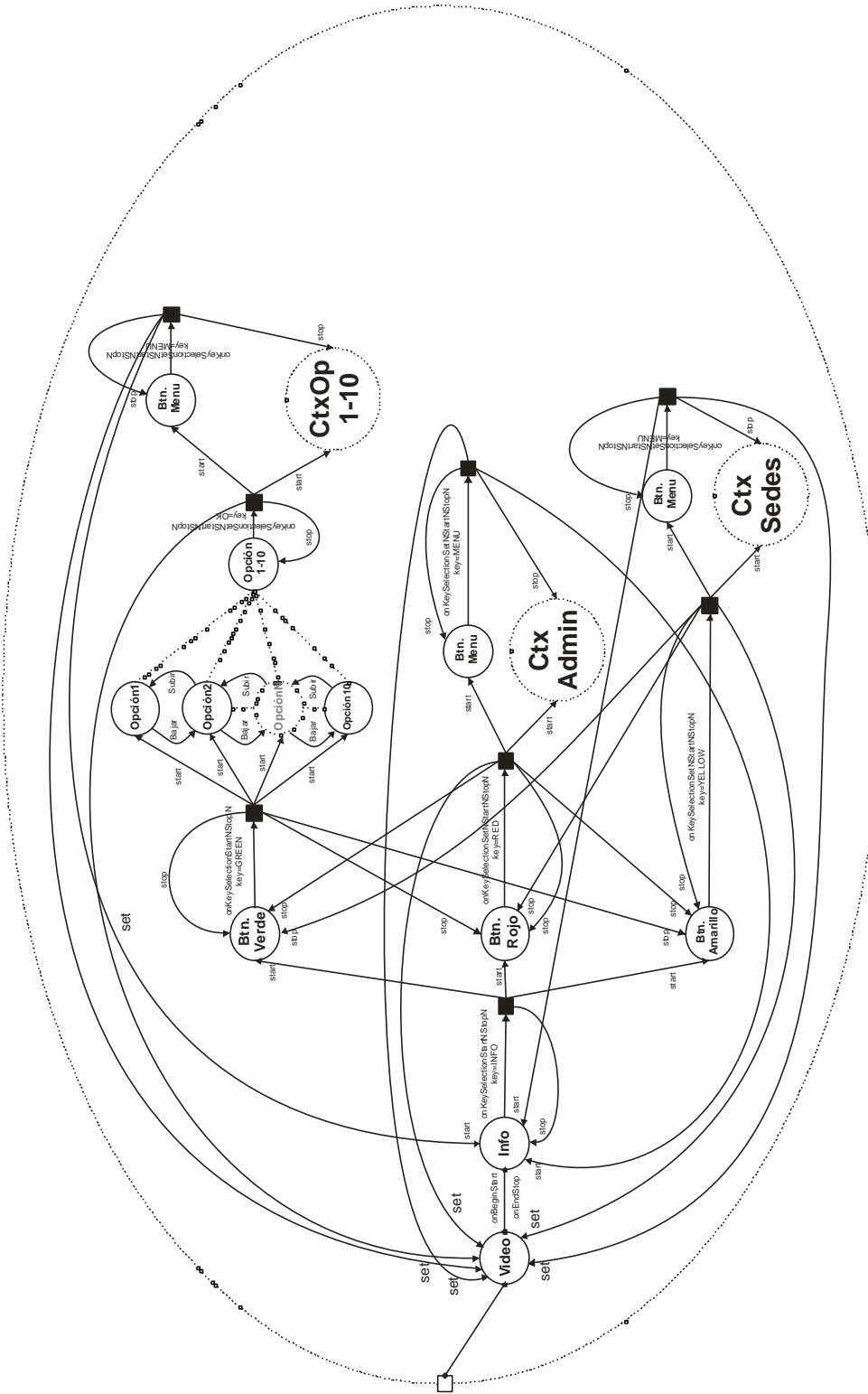


Figura 4.22. Diagrama NCM de la Aplicación Principal

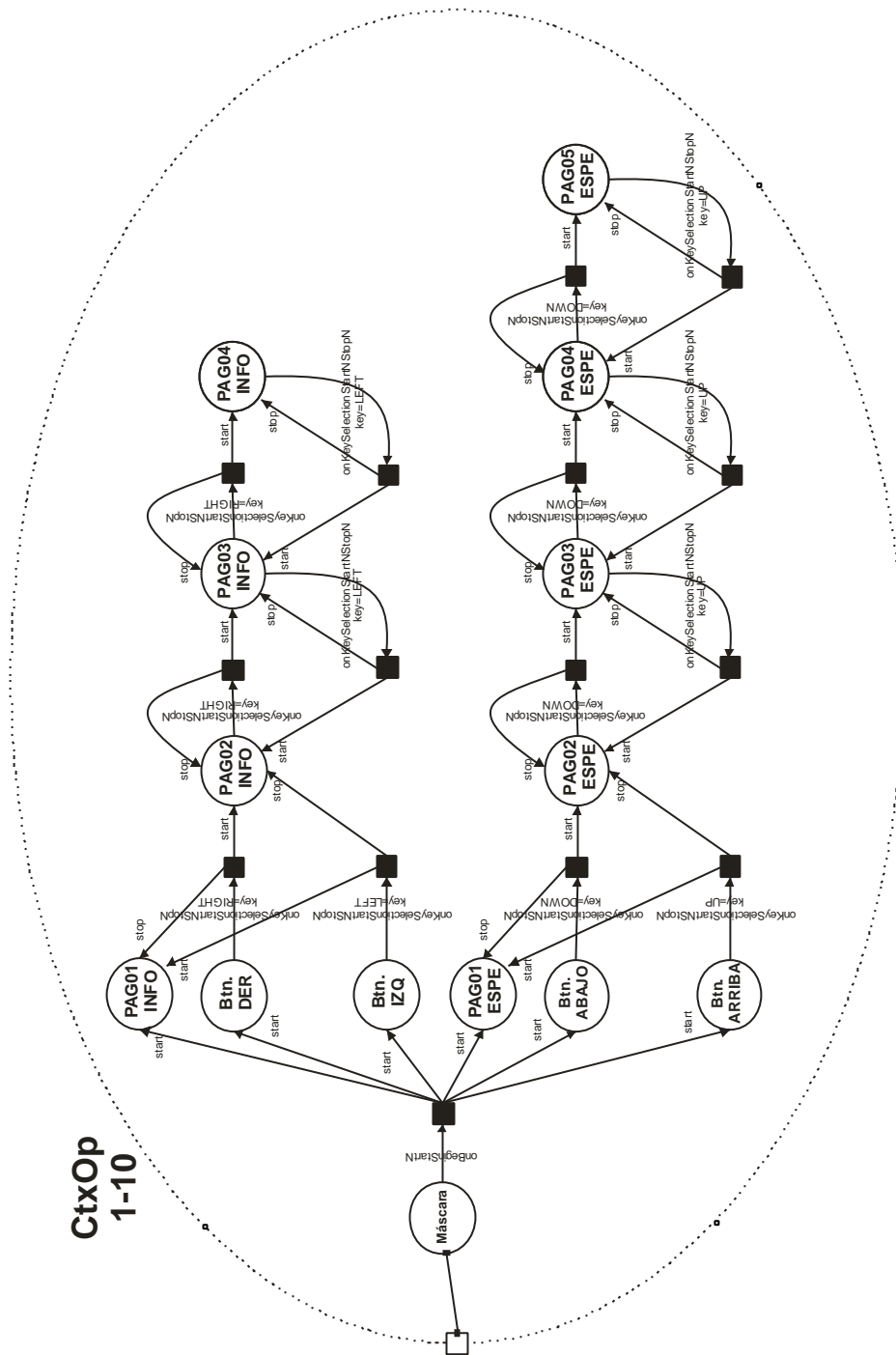


Figura 4.23. Diagrama NCM del Contexto General de las 10 opciones del submenú “Departamentos”

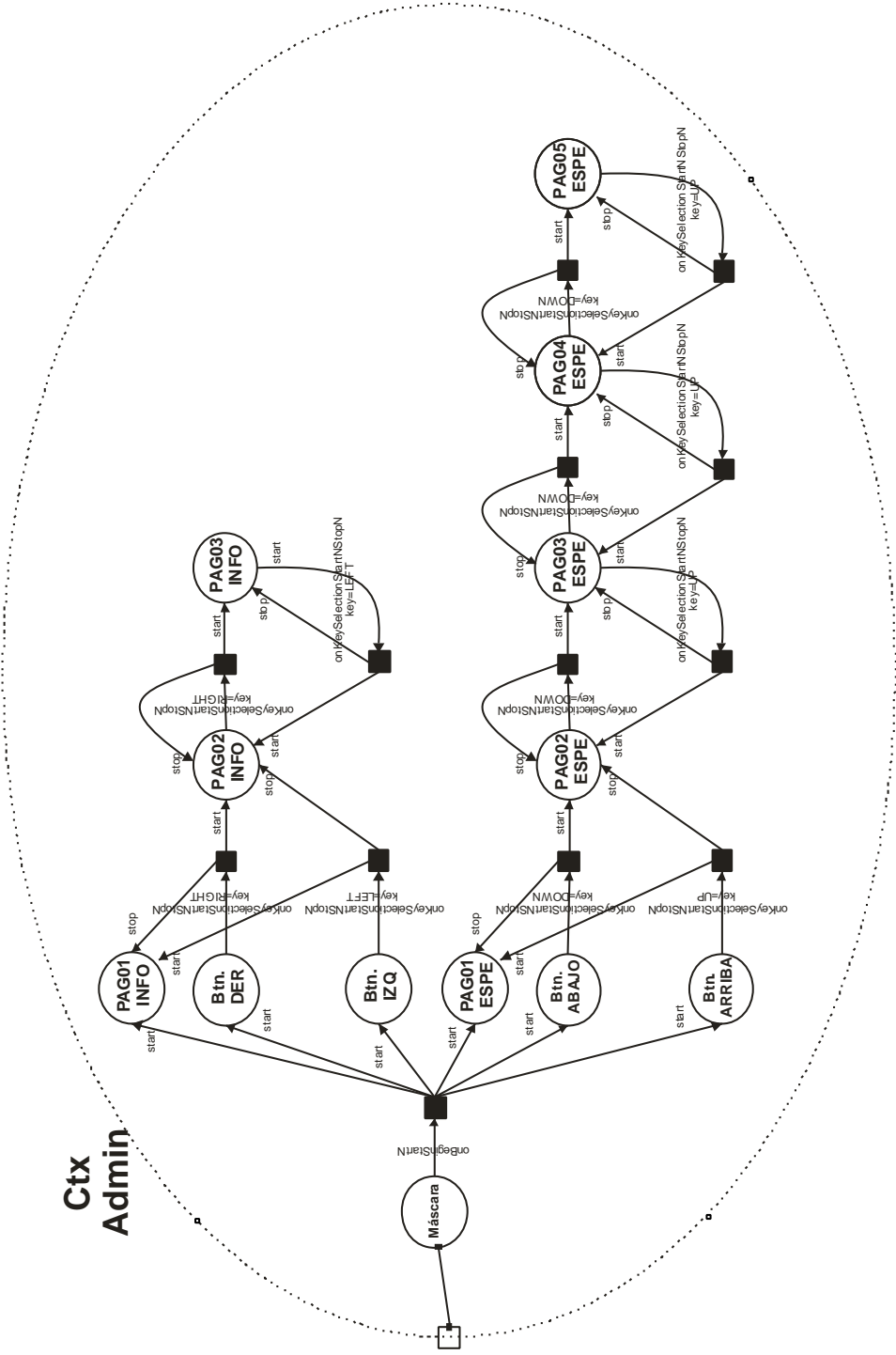


Figura 4.24. Diagrama NCM del Contexto del submenú “Admisión y Registro”



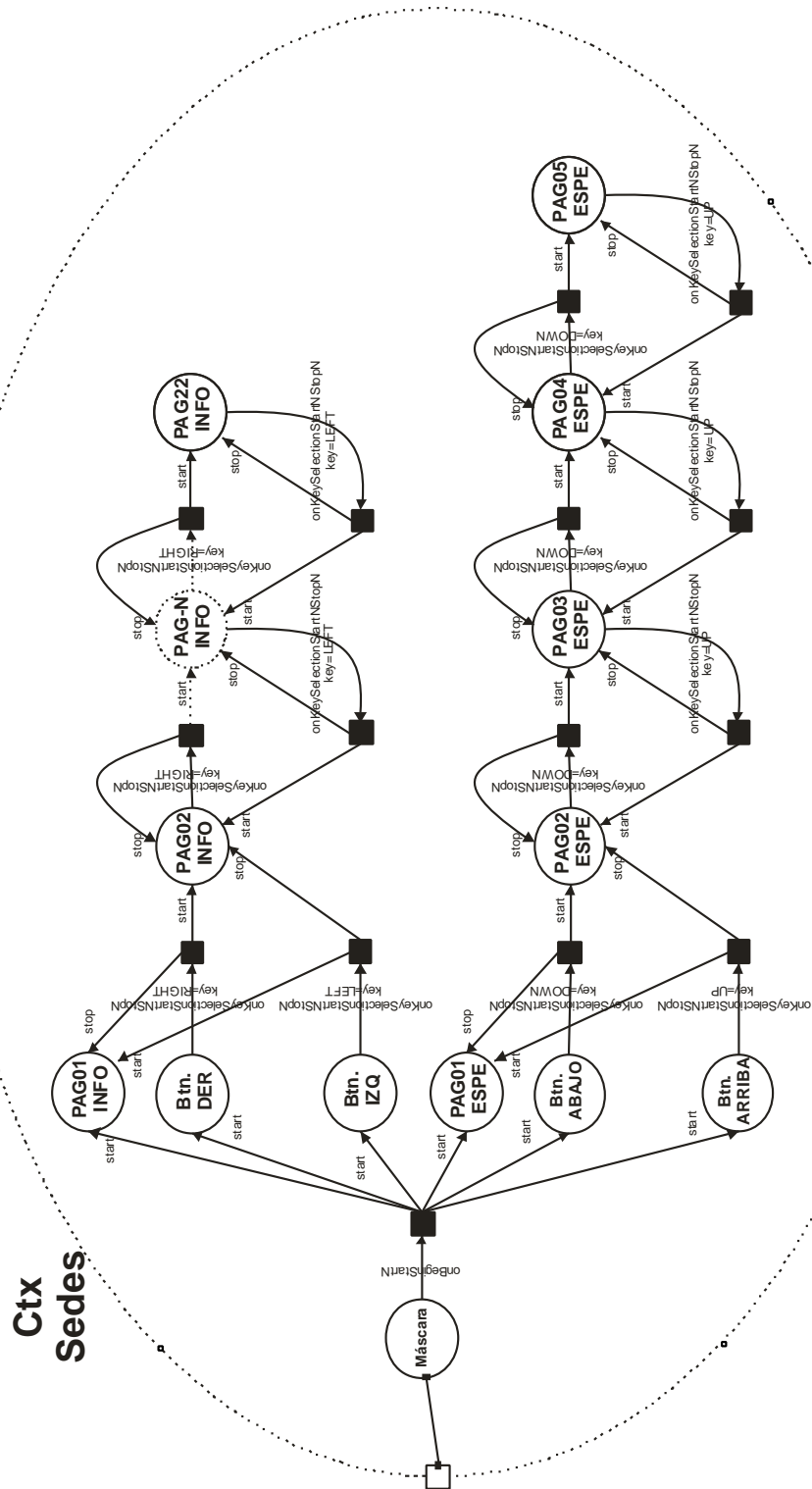


Figura 4.25. Diagrama NCM del Contexto del submenú "Sedes"

**4.5.1.4. Resultados**

Al finalizar la implementación de la Aplicación INFO-ESPE, el resultado final se puede evidenciar en las figuras 4.26- 4.31:



Figura 4.26. Primera opción de interacción a través del botón “INFO”



Figura 4.27. Menú Principal de Opciones

The screenshot displays the main interface of the ESPE website. On the left, there is a navigation menu with options like 'BANCA Y FINANZAS', 'INFORMACION Y MARKETING', 'MERCADOTECNIA', 'MARKETING Y PUBLICIDAD', and 'COMERCIO EXTERIOR'. Below this is the ESPE logo and a descriptive paragraph about the institution. On the right, there are two tables detailing academic activities for February-July 2010 and March-August 2010. At the bottom, there are navigation buttons for 'REGRESAR', 'menu', 'ANTERIOR', and 'SIGUIENTE'.

ACTIVIDAD	FEBRERO-JULIO 2010
Inscripciones Modalidad Presencial	Del 11 al 15 de enero
Pruebas de Admisión Modalidad Educación Presencial	20 de enero
Publicación de Resultados de Exámenes	19 de febrero
Matriúlas Ordinarias Mod. Presencial	Del 22 de febrero al 5 de marzo
Matriúlas Extraordinarias Mod. Presencial	Del 8 al 10 de marzo
Inicio de Clases Modalidad Presencial	15 de marzo
Último día de clases Modalidad Presencial	17 de julio

PERIODO ACADÉMICO MARZO-AGOSTO 2010	
ACTIVIDAD	
Temas de propuestas	A partir del 5 de enero 2010
Inscripciones	Del 11 al 15 de enero 2010
Pruebas de Admisión Modalidad Distancia	25 al 28 de enero 2010
Publicación de Resultados de Exámenes	19 de febrero 2010
Matriúlas Ordinarias Modalidad Distancia (estudiantes nuevos)	Del 22 de febrero al 5 de marzo 2010
Matriúlas Extraordinarias Modalidad Distancia (estudiantes nuevos)	Del 8 al 20 de marzo 2010
Matriúlas Ordinarias Modalidad Distancia (estudiantes antiguos)	Del 15 al 19 de marzo 2010
Matriúlas Extraordinarias Modalidad Distancia (estudiantes antiguos)	Del 23 al 25 de marzo 2010
Jornadas académicas	Sábado 20 de marzo 2010
Inicio del periodo académico	22 de marzo 2010
Finalización del periodo académico	Agosto de 2010

Figura 4.28. Pantalla principal, después de la selección la opción “ADMISION Y REGISTRO”

The screenshot shows a sub-menu for departments. On the left, a list of academic areas is displayed, with 'C. DE ENERGIA Y MECANICA' highlighted in red. On the right, there is a background image of students in uniform looking at a document.

- CIENCIAS DE LA VIDA
- C. TIERRA Y CONSTRUCCION
- C. DE ENERGIA Y MECANICA**
- C. HUMANAS Y SOCIALES
- CIENCIAS EXACTAS
- ECONOMICAS Y ADMINISTRATIVAS
- CIENCIAS DE LA COMPUTACION
- LENGUAS
- SEGURIDAD Y DEFENSA
- ELECTRICA Y ELECTRONICA

Figura 4.29. Submenú desplegado, después de la selección la opción “DEPARTAMENTOS”



Figura 4.30. Pantalla principal, después de la selección la opción “ELECTRICA Y ELECTRONICA” del submenú “DEPARTAMENTOS”



Figura 4.31. Pantalla principal, después de la selección la opción “SEDES”

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

Del análisis de la realidad actual de la televisión interactiva que vive la comunidad europea y de las previsiones de desarrollo, el tipo de contenidos interactivos han sido agrupados en tres grandes bloques: Los Servicios Interactivos Autónomos (SIA), Servicios Interactivos Asociados a Programas (SIAP) y Programas Audiovisuales Interactivos (PAI), cada uno de ellos con características propias.

Se pudo determinar además que actualmente debido a la complejidad de factores que conlleva la producción de PAI, se puede suponer que en un futuro inmediato la oferta de SIA y de SIAP crezca con mayor rapidez que la de los PAI a nivel mundial.

Se estableció que la consecución de una completa interactividad televisiva se logrará en la medida en que los programas ofrezcan alternativas a los telespectadores para que puedan alterar la estructura narrativa del contenido ofertado y configuren un discurso propio adaptado a sus preferencias.

Se comprobó que independientemente de la plataforma de hardware, la capa de middleware ofrece un servicio normalizado para las aplicaciones interactivas, ocultando las peculiaridades y diferencias de las capas más bajas,

ofreciendo una gran variedad de facilidades para el desarrollo de contenidos y aplicaciones de TV Digital.

Se demostró que mediante la integración de sus dos motores: el presentación y el de ejecución, Ginga puede dar soporte a aplicaciones tanto de tipo declarativo como de procedimiento, ofreciendo a los desarrolladores de aplicaciones interactivas la flexibilidad de trabajar con cualquiera de estos dos tipos de lenguaje de acuerdo a sus necesidades.

Se indicó que las implementaciones de referencia de Ginga-NCL y Ginga-J dan soporte a varias normativas como son la sincronización de archivos multimedia, la recepción en múltiples dispositivos exhibición, el desenvolvimiento de programas en vivo (en tiempo de exhibición), a la adaptación de contenidos y como se los presenta, entre otros.

La limitación en el desarrollo de aplicaciones interactivas con el cien por ciento de posibilidades que especifica la norma, es debido a que sus primeras implementaciones poseen una inmensa lista de ítems especificados que aun no han sido implementados, pero que por otro lado abre un amplio campo de desarrollo para las universidades, centros de investigación y empresas privadas interesadas en el tema.

Con la finalidad de permitir la compatibilidad de Ginga con los middlewares de otros sistemas de televisión digital y por lo tanto permitir el intercambio de contenidos interactivos entre los países que adoptan estos sistemas, Ginga se vinculó a GEM, a partir de la implementación de la parte Java del middleware o Ginga-J.

El lenguaje NCL posee un enfoque más amplio que el de su similar XHTML, en lo que corresponde al desarrollo de contenidos interactivos ya que este generaliza la sincronización espacio-tiempo por medio de los enlaces (links) NCL; adaptabilidad, definida por los elementos NCL *switch* y *switch* de descriptores; y el soporte a múltiples dispositivos de reproducción, definidos por

las regiones NCL, tratando a la interacción del usuario simplemente como un caso especial de la sincronización temporal.

El lenguaje NCL define como estructurar y relacionar los objetos multimedia, en tiempo y espacio; es decir este cumple la función de un “lenguaje pega”, el cual no restringe ni prescribe el tipo de contenido de los objetos multimedia, es decir que se pueden tener objetos de imagen, video, audio, texto, de ejecución, como objetos multimedia NCL. El tipo de objetos multimedia que soporta NCL dependerá del reproductor multimedia que es incorporado en el programa intérprete NCL (NCL Player), de hecho los reproductores multimedia comunes son ofertados por el modulo de núcleo común de Ginga.

El lenguaje NCL define dos perfiles: el Perfil BDTV (*Basic Digital TV Profile*) y el Perfil EDTV (*Enhanced Digital TV Profile*) con la finalidad que el lenguaje pueda trabajar en dispositivos con un mayor poder de procesamiento así como en dispositivos con recursos más limitados, los mismos que actualmente son soportado por las principales herramientas de software de desarrollo como Eclipse y Composer.

Se encontró una limitante en la presentación de contenidos con interactividad completa, es decir que utilicen el canal de interactividad a través del uso de scripts Lua, debido que estos eventos aun no están implementados en las herramientas de presentación.

Composer es un herramienta de autoría hipermedia desarrollada que construir programas audiovisuales interactivos con un mediano conocimiento del lenguaje NCL gracias a su amigable interfaz gráfica en la que se puede observar que un documento hipermedia a través de diversas visiones de desarrollo.

El programa interactivo en este trabajo desarrollado y presentado, es el resultado final de la investigación del presente proyecto en el cual se aplicaron todos los conocimientos adquiridos en el aprendizaje del lenguaje NCL, de modo que en su construcción y programación se utilizaron todos los elementos

principales del lenguaje, para que el mismo sea una muestra didáctica de lo que se puede aplicar y realizar con la parte declarativa del middleware Ginga.



## RECOMENDACIONES

Concluido el presente estudio se recomienda bajo las experiencias adquiridas a lo largo del mismo:

Para obtener un rendimiento óptimo al trabajar con la herramienta Composer versión 2.2.1 NCL 3.0 se recomienda trabajar bajo las versiones de sistema operativo como Windows XP o en Linux Ubuntu en versiones menores a la 10.04.

Para desarrolladores de aplicaciones interactivas que están empezando con el aprendizaje del lenguaje NCL, se recomienda iniciar su capacitación con el aprendizaje del software Composer, ya que este permite de una manera muy didáctica la comprensión de los distintos elementos del lenguaje por medio de su amigable interfaz gráfica.

El momento de desarrollar aplicaciones interactivas declarativas verificar las especificaciones del middleware tales como manejo de propiedades de los archivos multimedia ya desarrolladas por las implementaciones de referencia del middleware, propuestas actualmente por la PUC-Rio y el LIFIA de Argentina.

Para los desarrolladores que trabajan sobre software libre, específicamente sobre Linux Ubuntu, la comunidad Ginga de Argentina ha desarrollado un archivo ejecutable .deb del middleware Ginga, permitiendo al usuario evitar la utilización de una maquina virtual para correr del STBV, mejorando de esta manera el rendimiento de los recursos de hardware de nuestro computador de desarrollo.

Para potencializar el desarrollo de aplicaciones interactivas declarativas, una vez comprendido la lógica de programación y estructuración de un documento NCL a través de Composer, se recomienda la migración a la herramienta de desarrollo Eclipse la cual cuenta con los plugins necesarios tanto para NCL y Lua.

Por otro lado se recomienda a los investigadores, trabajar conjuntamente con la comunidad Ginga de Latinoamérica con toda confianza, para compartir y solventar las dudas que vayan apareciendo a medida que se desarrollen nuevos proyectos de investigación relacionados al presente.

Para programadores expertos se recomienda analizar el código fuente del middleware y embeberlo en diferentes tipos de receptores, lo que permitirá mejorar y realizar nuevas implementaciones del middleware como un aporte a la comunidad Ginga.

Finalmente, a pesar de que la televisión digital interactiva ofrece nuevas oportunidades tanto de negocios como de acceso a la información, los desarrolladores de contenidos deberán tomar muy en cuenta factores como la realidad de consumo y necesidades de nuestra sociedad, así como la visión y misión bajo las cuales fue creado el middleware Ginga.

## REFERENCIAS BIBLIOGRÁFICAS

[1] MERITXELL, Estebanell, ***“Interactividad e Interacción”***, Universidad de Girona – España, <http://web.udg.edu/pedagogia/images/gretice/INTERACT.pdf>, revisado en octubre de 2009.

[2] AYMERICH, Laura, ***“Los nuevos formatos de publicidad interactiva en televisión”***, Grupo de investigación consolidado del Departamento de Comunicación Audiovisual y Publicidad de la UAB, [http://www.griss.org/curriculums/aymerich/publicaciones/aeic\\_aymerich.pdf](http://www.griss.org/curriculums/aymerich/publicaciones/aeic_aymerich.pdf), revisado en octubre de 2009.

[3] ARCINIEGAS, José, AMAYA, Juan, URBANO, Franco, CAMPO, Wilmar, EUSCATEGUI, Robert, GARCÍA, Alexander, GARCÍA, Xabiel, ***“EDiTV: Educación virtual basado en televisión interactiva para soportar programas a distancia”***, Universidad del Cauca – Colombia, Universidad Autónoma de Occidente – Colombia, Universidad de Oviedo – España, [www.renata.edu.co/.../48-educacion-virtual-basada-en-television-interactiva-para-apoyar-procesos-educativos-a-distancia.html](http://www.renata.edu.co/.../48-educacion-virtual-basada-en-television-interactiva-para-apoyar-procesos-educativos-a-distancia.html), revisado en noviembre de 2009.

[4] PRADO, Emili, FRANQUET, Rosa, SOTO, María, RIBES, Xavier, FERNÁNDEZ, David, ***“Tipología funcional de la televisión interactiva y de las aplicaciones de interacción con el televisor”***, <http://www.ehu.es/zer/zer25/ze25-1-prado.pdf>, revisado en octubre de 2009.

[5] SIMIONI, Adriano, ***“Un framework para el desenvolvimiento de aplicaciones interactivas para Televisión Digital”***, Centro de Ciencias exactas y Tecnológicas, Universidad de Vale do Rio dos Sinos, [www.sbc.org.br/bibliotecadigital/download.php?paper=959](http://www.sbc.org.br/bibliotecadigital/download.php?paper=959), revisado en octubre de 2009.

[6] Pagina Oficial de MHP, <http://www.mhp.org>, Marzo 2010.

[7] GIL SOLLA, Alberto, PAZOS ARIAS, José, LÓPEZ NORES, Martín, BLANCO FERNÁNDEZ, Yolanda, ***“Experiencias sobre una Implementación Libre y Abierta del Estándar MHP para TV Digital Interactiva”***, Departamento de

Ingeniería Telemática, Universidad de Vigo, <http://idtv.det.uvigo.es/pdfs/jitel03.pdf>, revisado en enero de 2010.

[8] AMATLLER, Jordi, BALDO, David, BENELLI, Giuliano, DAINO, Giovanni, Zambon, Ricardo, "**Televisión Digital Terrestre Interactiva, el desafío de la interoperabilidad en Brasil**", Departamento de la Ingeniería en Informática de la Universidad de Siena, <http://downloads.hindawi.com/journals/ijdmb/2009/579569.pdf>, revisado en enero de 2010.

[9] PÉREZ GARCÍA, Nelson, "**Televisión Digital Terrestre e Interactividad**", Universidad de los Andes, <http://gitel.ing.ula.ve/Curso-TDT-CENDIT-Bloque2-Parte2.pdf>, revisado en noviembre de 2009.

[10] LAIOLA GUIMARÃES, Rodrigo, DE RESENDE COSTA, Romualdo, "**Interactividad y Sincronización en TV Digital**", Departamento de Informática, PUC de Rio, [http://rodrigo.laiola.com.br/academic/laiola\\_monografia\\_interatividade.pdf](http://rodrigo.laiola.com.br/academic/laiola_monografia_interatividade.pdf), revisado en enero de 2010

[11] RIBEIRO, Jean, "**Middleware Ginga**", Departamento de Ingeniería, Universidad Federal Fluminense, <http://www.midiacom.uff.br/~debora/fsmm/trab-2008-2/middleware.pdf>, revisado en diciembre de 2009.

[12] **ABNT NBR 15606-2** – Asociación Brasileira de Normas Técnicas – "Televisión Digital Terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 2: Ginga-NCL para receptores fijos y móviles – Lenguaje de aplicación XML para codificación de aplicaciones" Sistema Brasileiro de TV Digital Terrestre, [http://www.abnt.org.br/imagens/Normalizacao\\_TV\\_Digital/ABNTNBR15606-2\\_2007Vc\\_2008.pdf](http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNBR15606-2_2007Vc_2008.pdf), 2008.

[13] **ABNT NBR 15606-5** – Asociación Brasileira de Normas Técnicas – "Televisión Digital Terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 5: Ginga-NCL para receptores portátiles – Lenguaje de aplicación XML para codificación de aplicaciones" Sistema Brasileiro de TV Digital Terrestre,

[http://www.abnt.org.br/imagens/Normalizacao\\_TV\\_Digital/ABNTNBR15606-5\\_2008Ed1.pdf](http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNBR15606-5_2008Ed1.pdf), 2008.

[14] **ABNT NBR 15606-4** – Asociación Brasileira de Normas Técnicas – “Televisión digital terrestre – Codificación de datos y especificaciones de transmisión para radiodifusión digital – Parte 4: Ginga-J – Ambiente para ejecución de aplicaciones de procedimiento” Sistema Brasileiro de TV Digital Terrestre, [http://www.dtv.org.br/download/pt-br/ABNTNBR15606-4\\_2010Ed1.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15606-4_2010Ed1.pdf), 2010.

[15] GOMES, Luiz, FERREIRA Rogéiro, FERREIRA, Márcio, “**Ginga-NCL: El Ambiente Declarativos del Sistema de TV Digital Brasileira**”, Departamento de Informática, Pontificia Universidad Católica de Rio de Janeiro, [ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/journalpapers/2007\\_04\\_jbcs\\_lfgs.pdf](ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/journalpapers/2007_04_jbcs_lfgs.pdf), revisado en septiembre de 2009.

[16] LEMOS DE SOUZA, Guido, CUNHA, Luiz, COELHO Carlos, “**Ginga-J: El Middleware de Procedimiento del Sistema de TV Digital Brasileira**”, Departamento de Informática, Universidad Federal de Paraíba, [http://www.tvdi.inf.br/upload/artigos/ginga-j\\_the\\_procedural\\_middleware\\_for\\_brazilian\\_digital\\_tv\\_system.pdf](http://www.tvdi.inf.br/upload/artigos/ginga-j_the_procedural_middleware_for_brazilian_digital_tv_system.pdf), revisado en septiembre de 2009.

[17] GOMES SOARES, Luiz, JUNQUEIRA BARBOSA, Simone, “**TV digital interactiva en Brasil se hace con Ginga**”, Departamento de Informática, Pontificia Universidad Católica de Rio de Janeiro, <http://www.ncl.org.br/documentos/JAI2008.pdf>, revisado en diciembre de 2009.

[18] SOARES, Carlos, GOMES, Luiz, FERREIRA, Rogerio, JUNQUERIA, Simone, “**Construyendo programas audiovisuales interactivos utilizando NCL 3.0 y la herramienta Composer**”, Laboratorio de Sistemas Multimedia TeleMídia, Pontificia Universidad Católica de Rio de Janeiro, <http://www.ncl.org.br/documentos/TutorialNCL3.0-2ed.pdf>, revisado en diciembre de 2009.

[19] GOMES SOARES, Luiz, JUNQUEIRA BARBOSA, Simone, “**Programando en NCL 3.0**”, Editorial Elsevier, Tercera Edición, 2009.

[20] CARVALHO, Rafael, FERREIRA, Joel, RIBEIRO, Jean, VARANDA, Julia, MUCHALUAT, Debora, ***“Introducción a los Lenguajes NCL y Lua: Desarrollando Aplicaciones Interactivas para TV Digital”***, Laboratorio MídiaCom, Universidad Federal Fluminense, <http://www.midiacom.uff.br/gtvd/files/apostila.pdf>, revisado en enero de 2010.

[21] Página Oficial de Ginga Argentina: <http://www.ginga.org.ar/>, 2010.

[22] Página Oficial de Ginga Brasil: <http://www.ginga.org.br/>, 2010.

[23] Página Oficial de Ginga Perú: <http://www.ginga.org.pe/>, 2010.

[24] Página Oficial de Software Público de Brasil: <http://www.softwarepublico.gov.br/>, 2010.

[25] Página Oficial del Club NCL de Brasil: <http://clube.ncl.org.br/>, 2010.

[26] Página Oficial del Laboratorio TeleMídia <http://www.telemidia.puc-rio.br>, 2010.

[27] Página Oficial del Laboratorio LAVID: <http://www.lavid.ufpb.br/>, 2010.

[28] Página Oficial de la Comunidad Ginga Argentina: <http://comunidad.ginga.org.ar/>, 2010.

[29] Página Oficial de NCL Brasil: <http://www.ncl.org.br>, 2010.