



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN

TELECOMUNICACIONES Y REDES

**ESTUDIO METODOLÓGICO PARA EL DISEÑO DE INTERFACES
ENTRE EL PC Y EL USUARIO UTILIZANDO ISDB.Tb Y
MIDDLEWARE GINGA**

TESIS DE GRADO

Previa a la obtención del título de:

INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

Presentado por:

María José López Montero

Karina Daniela Oleas Santillán

RIOBAMBA – ECUADOR

2012

Quiero agradecer a mi papá Dios por todo lo bueno y maravilloso que ha sido conmigo, porque sé que sin la ayuda de El nada de esto se habría logrado, a mi mami Sonia por su sacrificio y entrega, por cada lagrima que alguna vez ha salido de nuestros ojos, porque juntas hemos luchado hasta cruzar esta meta, a mi hermanito Sebastián por ser mi ejemplo y quien siempre me ha dado valor para seguir adelante, a todos los Docentes de mi querida escuela por sus enseñanzas que más allá de educarme en ciencia me han guiado en la universidad de la vida ,a mis siete amigos y compañeros de aula Karina, David, Pato, Gabriel, Tulio, Víctor y Luis por todas y cada una de las experiencias vividas, por ser más que mis compañeros, mis amigos y mi apoyo en muchos momentos, a mi tutor de Tesis, Ing. Paúl Romero por ser mi apoyo a lo largo de mi carrera, porque más allá de ser mi profesor en muchas ocasiones he sentido el cariño de padre y su mano extendida para levantarme cuando he necesitado de alguien, al personal administrativo y técnico porque sin ustedes nada sería igual, gracias por haberme brindado su amistad y ayuda todo este tiempo, son tantas las personas que han colaborado para hoy ver cristalizado este sueño que lo único que me queda es decirles un Dios le pague de corazón y que nuestro Padre Dios les guíe y bendiga siempre a lo largo de toda su vida.

Agradezco a Dios por darme salud y vida para terminar con éxito mi carrera profesional, a mis padres y hermanos 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 a los Ingenieros Paul Romero y William Calvopiña, que mas allá de haberme guiado durante la elaboración de la tesis 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, 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 han brindado y por todas sus bendiciones.

A todos gracias de corazón.

Este proyecto de tesis lo dedico a mi madre porque gracias a su amor, dedicación y ejemplo he logrado alcanzar este sueño que juntas lo construimos hace muchos años atrás, a mis dos amores chiquitos, a mi hija Vale Gordita Bebe y a mi princesa Renatita.

A ti Vale porque desde bebe me has regalado tu amor, por cada pequeña ocurrencia, porque de tus labios escuche susurrar por primera vez la palabra mamá, y porque quien más Tú, aun siendo tan pequeñita siempre confiaste en mi y en que puedo alcanzar mis sueños.

A mi princesa Renatita porque desde que naciste cambiaste el sentido de nuestras vidas, eres ese angelito que hacía falta en nuestro hogar, porque con cada cosa que haces o cada palabra que dices llenas nuestras vidas y nuestros corazones de alegría.

Este proyecto de tesis le dedico principalmente a mis padres: Arnulfo y Mariana :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.

A mi hija Marianita Dayanara la luz de mi existencia, el motor que me impulsa a seguir adelante

A mis hermanas y cuñados, 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.

| NOMBRE | FIRMA | FECHA |
|---|--------------|--------------|
| Ing. Iván Menes. DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA | ----- | ----- |
| Ing. Pedro Infante DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES | ----- | ----- |
| In. Paúl Romero DIRECTOR DE TESIS | ----- | ----- |
| Ing. William Calvopiña MIEMBRO DEL TRIBUNAL | ----- | ----- |
| Tlgo. Carlos Rodríguez. DIRECTOR DPTO. DOCUMENTACIÓN | ----- | ----- |
| NOTA DE LA TESIS: ----- | | |

RESPONSABILIDAD

“Yo, MARÍA JOSÉ LÓPEZ MONTERO, soy responsable de las ideas, doctrinas y resultados expuestos en esta: Tesis de Grado, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”

María José López Montero

RESPONSABILIDAD

“Yo, KARINA DANIELA OLEAS SANTILLAN, soy responsable de las ideas, doctrinas y resultados expuestos en esta: Tesis de Grado, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”

KARINA DANIELA OLEAS SANTILLÁN

ÍNDICE DE ABREVIATURAS

| ABREVIATURA | DESCRIPCIÓN |
|--------------------|---|
| A | |
| API: | Aplicación Programan Interface |
| B | |
| BDTV: | Basic Digital TV Profile |
| C | |
| CMS: | Content Management System |
| D | |
| DTV: | Digital Television |
| E | |
| E.S.P.O.CH. | Escuela Superior Politécnica de Chimborazo |
| EPG | Guía de Programación Electrónica |
| H | |
| HD | Alta definición |
| I | |
| IPTV: | Internet Protocol Television |
| ISDB-Tb: | International System for Digital Broadcast, Terrestrial, Brazilian version |
| J | |
| JVM: | Java Virtual Machine |

N

NCL: Nested Conext Language

P

PUC-Rio Pontificia Universidad Católica de Rio de Janeiro

S

STB: Set Top Box

SIA: Servicios Interactivos Autónomos

SBTVD: Sistema Brasileiro de Televisión Digital

T

TVI: Televisión Interactiva

TDT: Televisión Digital Terrestre

V

VSTB: Virtual Set Top Box

X

XML: Extensible Markup Language

XHTML: The Extensible HyperText Markup Language

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

LISTA DE ABREVIATURAS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ANTECEDENTES

JUSTIFICACIÓN DEL PROYECTO DE TESIS

OBJETIVOS

OBJETIVO GENERAL

OBJETIVOS ESPECIFICOS

HIPÓTESIS

CAPÍTULO I

MARCO TEÓRICO

| | | |
|-----|--|----|
| 1.1 | Introducción a la Televisión Digital Terrestre | 25 |
| 1.2 | Qué es la TDT? | 26 |
| 1.3 | Estructura de la TV Digital | 27 |
| 1.4 | Ventajas de la TDT | 28 |

| | | |
|---------|---|----|
| 1.5 | Interactividad | 31 |
| 1.5.1 | ¿Qué servicios interactivos ofrece la TDT? | 32 |
| 1.5.2 | ¿Qué aporta la interactividad? | 33 |
| 1.5.3 | Ventajas de la Interactividad | 33 |
| 1.5.4 | Agentes que intervienen en la Interactividad..... | 34 |
| 1.5.5 | Escenarios de uso de la interactividad | 34 |
| 1.5.5.1 | Interactividad local | 34 |
| 1.5.5.2 | Interactividad Completa | 35 |
| 1.6. | Servicios Interactivos | 36 |
| 1.7. | Estándares de TDT en el mundo | 39 |
| 1.8 | Desarrollo de Contenidos de TDT | 39 |

CAPÍTULO II

GINGA NCL

| | | |
|-------|-------------------------------------|----|
| 2.1 | Introducción | 40 |
| 2.2 | Paradigmas de la programación | 41 |
| 2.2.1 | Lenguaje Declarativo | 41 |

| | | |
|-----------|---|-------|
| 2.2.2 | Lenguaje Imperativo | 41 |
| 2.3 | Concepto de Ginga | 42 |
| 2.4 | ¿Por qué el nombre de Ginga? | 42 |
| 2.5 | ¿Por qué Ginga es un software libre? | 43 |
| 2.6. | Arquitectura del middleware Ginga | 44 |
| 2.7 | Subsistemas de Ginga | 45 |
| 2.7.1 | Ginga (Comon-Core) CC | 45 |
| 2.7.2 | <i>Ginga-J</i> | 50 |
| 2.7.3 | <i>Ginga-ncl</i> | 52 |
| 2.7.3.1 | Lenguaje de programación de ginga ncl | 53 |
| 2.7.3.2 | Componentes de Ginga ncl | 55 |
| 2.7.3.3 | Estructura básica de un módulo ncl | 57 |
| 2.7.3.4 | Ventajas del lenguaje ncl | 63 |
| 2.7.3.5 | Estructura de un documento ncl | 64 |
| 2.7.3.6 | Relación entre entidades de un documento ncl..... | 67 |
| 2.7.3.6.1 | ¿Qué Reproducir? | 68 |
| 2.7.3.6.2 | ¿Dónde Reproducir? | 70 |
| 2.7.3.6.3 | ¿Cómo Reproducir? | 71-72 |

2.7.3.6.4 ¿Cuándo Reproducir? 72

CAPITULO III

INSTALACIÓN VMWAREPLAYER Y ECLIPSE

3.1 Instalación Ginga ncl 74

3.1.1 Requerimientos de Hardware 75

3.1.2 Requerimientos de Software 75

3.2 Instalación de vmwareplayer 75

3.3 Eclipse ncl 78

3.3.1 Plugin ncl Eclipse 78

3.3.2 Instalación Eclipse ncl 79

CAPITULO IV

METODOLOGIA PARA EL DISEÑO DE INTERFACES

4.1 Desarrollo de la aplicación ncl 86

4.2 Aplicación ncl INFO- ESPOCH 87

4.3 Requisitos y modelamiento de la aplicación 88

4.4 Programación y pruebas de la aplicación 88

4.5 Implementación..... 89

4.6 Resultados 89

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMARY

ANEXOS

Anexo 1. Código de Implementación

BIBLIOGRAFÍA

..

ÍNDICE DE FIGURAS

| DESCRIPCIÓN | Página |
|---|---------------|
| Figura I.1. Estructura de la TDT | 27 |
| Figura I.2. Calidad de Imagen | 28 |
| Figura I.3. Dispositivos Múltiples en TDT | 29 |
| Figura I.4. Programación Múltiple en TDT | 29 |
| Figura I.5. Segmentación | 29 |
| Figura I.6. Transmisión de TV analógica a 4 kw | 30 |
| Figura I.7. Transmisión de TV digital a 1 kw | 30 |
| Figura I.8. Ejemplo de EPG | 30 |
| Figura I.9. Ejemplo de Alertas con TDT | 31 |
| Figura I.10. Interactividad | 32 |
| Figura I.11. Agentes de interactividad en TDT | 34 |
| Figura I.12. Interactividad Local | 35 |
| Figura I.13. Canal de Retorno | 35 |
| Figura I.14. Interactividad Completa | 36 |
| Figura I.15. Estándares TDT en el mundo | 39 |
| Figura II.16. Ginga Ecuador | 42 |
| Figura II.17. Movimiento Fundamental de la Capoeira | 43 |

| | |
|---|----|
| Figura II.18. Arquitectura de Ginga | 45 |
| Figura II.19. Componentes de Ginga – CommonCore | 46 |
| Figura II.20. APIs de Ginga-J | 52 |
| Figura II.21. Subsistema Ginga-NCL | 55 |
| Figura II.22. Estructura de un Documento NCL | 64 |
| Figura II.23. Estructura general de un Documento NCL | 67 |
| Figura II.24. Relación entre entidades de un documento NCL | 68 |
| Figura II.25. Representación de Archivos Multimedia | 69 |
| Figura II.26. Representación de nodos multimedia y su composición | 70 |
| Figura II.27. Representación de una Región | 70 |
| Figura II.28. Representación de un descriptor asociado a una región | 71 |
| Figura II.29. Representación de un descriptor asociado a un archivo multimedia que será presentado en una región | 72 |
| Figura II.30. Puerta de un nodo de composición | 73 |
| Figura III.31. Pantalla Principal de VmWare Workstation | 76 |
| Figura III.32. Pantalla ubuntu-server10.10-ginga | 76 |
| Figura III.33. Pantalla ginga-NCL STB | 77 |
| Figura III.34. Pantalla ubuntu-server10.10-ginga | 77 |
| Figura III.35. Pantalla principal de la maquina virtual de Ginga un vez ejecutada..... | 78 |

| | |
|---|----|
| Figura III.36. Pantalla descompresión eclipse-SDK-3.1.7-win32.zip | 79 |
| Figura III.37. Pantalla de ubicación de eclipse-SDK-3.1.7-win32.zip | 80 |
| Figura III.38. Pantalla ventana principal eclipse-SDK-3.1.7 | 80 |
| Figura III.39. Instalación nuevo Software eclipse-SDK-3.1.7 | 81 |
| Figura III.40. Localización de eclipse-SDK-3.1.7 | 81 |
| Figura III.41. Pantalla NCL eclipse 1.6.0 | 82 |
| Figura III.42. Licencia NCL eclipse 1.6.0 | 82 |
| Figura III.43. Instalación software NCL eclipse 1.6.0 | 83 |
| Figura III.44. Configuración de Window | 83 |
| Figura III.45. Configuración de Remote Ginga NCL | 84 |
| Figura III.46. Creación NCL Project | 84 |
| Figura III.47. NCL Project | 85 |
| Figura IV.48. Ciclo de vida en cascada de la Aplicación implementada | 87 |
| Figura IV.49. Menú Principal de Opción (1) | 89 |
| Figura IV.50. Menú Principal de Opción (2)..... | 89 |
| Figura IV.51. Pantalla principal, después e la selección la opción “AUTORIDADES” | 90 |
| Figura IV.52. Pantalla principal, después de la selección la opción “VISION” | 90 |
| Figura IV.53. Pantalla principal, después de la selección la opción | |

| | |
|---|----|
| “MISION” | 91 |
| Figura IV.54. Pantalla principal, después de la selección la opción | |
| “FACULTADES” | 91 |
| Figura IV.55. Pantalla principal, después de la selección la opción | |
| “FIE” del menú “FACULTADES” | 92 |
| Figura IV.56. Pantalla principal, después de la selección la opción | |
| “FADE” .del menú “FACULTADES” | 92 |
| Figura IV.57. Pantalla principal, después de la selección la opción | |
| “C.PECUARIAS” del menú “FACULTADES” | 93 |
| Figura IV.58. Pantalla principal, después de la selección la opción | |
| “R. NATURALES” del menú “FACULTADES” | 93 |
| Figura IV.59. Pantalla principal, después de la selección la opción | |
| “S.PUBLICA” del menú “FACULTADES” | 94 |
| Figura IV.60. Pantalla principal, después de la selección la opción | |
| “CIENCIAS” del menú “FACULTADES” | 94 |
| Figura IV.61. Pantalla principal, después de la selección la opción | |
| “MECANICA” del menú “FACULTADES” | 95 |

ÍNDICE DE TABLAS

| DESCRIPCIÓN | Página |
|---|---------------|
| TABLA I.I. Interactividad local vs Interactividad remota | 37 |
| TABLA II.II Elementos Head | 58 |
| TABLA II.III Elementos Body y Context | 58 |
| TABLA II.IV. Estructura Básica de un Documento NCL..... | 65 |

ANTECEDENTES

Al estar a pocos años del 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, se está sufriendo un cambio significativo, la migración del sistema tradicional de televisión 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. En el caso de nuestro país luego de haber realizado las respectivas pruebas técnicas por parte de la SUPERTEL, se adoptó oficialmente el estándar brasileño ISDB-Tb a la par de Middleware Ginga como plataforma de desarrollo y presentación de contenidos interactivos.

JUSTIFICACIÓN DEL PROYECTO DE TESIS

Hemos creído conveniente enfocar nuestro estudio hacia la televisión digital ya que esta nos trae muchas innovaciones y ventajas con relación a la televisión analógica, por ejemplo

- La transmisión de video en alta definición (HD)
- Podemos disfrutar de ambientes totalmente interactivos.
- La TDT innova toda la idea de TV Analógica, mediante sus dos tipos de canales: el simple y el canal de retorno, ya que estos nos permiten realizar muchas de las actividades que se pensaba que no estaban al alcance del usuario con la TV Analógica.
- La TDT nos brinda la posibilidad de mejorar el aprovechamiento del espectro radio eléctrico.
- Mejora la impermeabilidad de la señal digital a las interferencias del medio.
- Pero sobre todo la oportunidad de poder crear un entorno mediante el cual el usuario pueda interactuar de manera directa con esta nueva tecnología y familiarizarse rápidamente con ella, ya que podrá gozar de servicios de interactividad desplegados en:
 - Educación, Información.
 - Telemedicina, entre otros que simplifican el diario vivir del usuario. Todos estos además de entretener a la teleaudiencia le ayudan a resolver problemas de manera inmediata, y a minimizar el tiempo.

OBJETIVOS

OBJETIVO GENERAL

Realizar el estudio metodológico para la simulación del diseño de interfaces entre el PC y el Usuario utilizando el estándar ISDB-Tb y Middleware Ginga.

OBJETIVOS ESPECIFICOS

- Estudiar la arquitectura del Middleware Ginga.
- Entender y programar Ginga NCL a través del software de soporte Eclipse.
- Determinar los requerimientos, tanto de hardware como de software, para implementar la plataforma virtual de desarrollo de contenidos interactivos.
- Desarrollar una aplicación virtual interactiva basada en Ginga-NCL para la ESPOCH.

- Familiarizarse con el Lenguaje de programación de Ginga NCL.
- Tomar en cuenta la importancia de la Metodología de Diseño en las interfaces interactivas de TDT.
- Socializar los resultados generados del proyecto de investigación.

HIPÓTESIS

El estudio metodológico para la simulación del diseño de interfaces entre el PC y el usuario utilizando el estándar ISDB-Tb y Middleware Ginga permitirá crear un entorno que se asemeje a la realidad del usuario de TDT lo cual le brindará la oportunidad de familiarizarse rápidamente con esta nueva tecnología y acceder a servicios de interactividad.

CAPÍTULO I

MARCO TEÓRICO

1.1. INTRODUCCIÓN A LA TELEVISIÓN DIGITAL TERRESTRE



La televisión digital terrestre (TDT) es la tecnología de transmisión y recepción de imagen y sonido, a través de señales digitales. En contraste con la televisión tradicional, que codifica los datos de manera analógica, la televisión digital codifica sus señales de forma binaria, habilitando así la posibilidad de crear vías de retorno entre consumidor y productor de contenidos, abriendo la posibilidad de crear aplicaciones interactivas, y la capacidad de transmitir varias señales en un mismo canal asignado.

La llegada de la televisión digital supone un cambio tan radical como el que supuso el paso del blanco y negro al color.

Se trata de conseguir imágenes mejores, pero no se queda ahí, sino que también se van a abrir las puertas a la futura introducción de servicios hasta ahora inimaginables, como la recepción móvil de televisión, la interactividad, la televisión a la carta o los servicios multimedia tan de moda hoy en día con la explosión de Internet.

1.2. ¿QUÉ ES LA TDT?

La **televisión digital** (o DTV, de sus siglas en inglés: *digital TV*) se refiere al conjunto de tecnologías de transmisión y recepción de imagen y sonido, a través de señales digitales.

En contraste con la televisión tradicional, que codifica los datos de manera analógica, la televisión digital codifica sus señales de forma binaria, habilitando así la posibilidad de crear vías de retorno entre consumidor y productor de contenidos, abriendo la posibilidad de crear aplicaciones interactivas, y la capacidad de transmitir varias señales en un mismo canal asignado, gracias a la diversidad de formatos existentes.

Un sistema integro de televisión digital, incorpora los siguientes actores:

- Cámaras de video digitales, que trabajan a resoluciones similares y más altas que las análogas.
- Transmisión digital.
- Pantallas digitales (por ejemplo, plasma, LCD, LEDS).

1.3. ESTRUCTURA DE LA TV DIGITAL

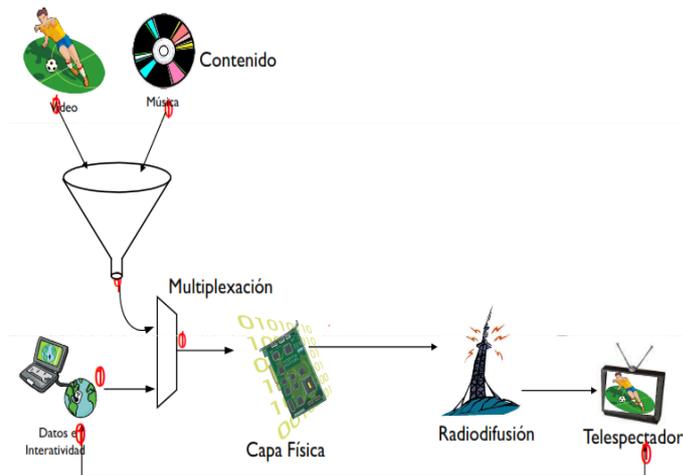


Figura I.1. Estructura de la TDT

La televisión digital es aquella en la cual se transmite, recibe y procesa señales de audio y video de manera discreta (1s y 0s), en contraste con la forma continua usada por la TV analógica. La digitalización de la televisión lleva consigo numerosas ventajas en la forma de entender y utilizar la televisión; la representación numérica permite el uso de compresores, filtros digitales, control de conexión local, detección y corrección de errores, canales de doble vía, etc. Una de las principales ventajas de la digitalización es poder aplicar técnicas de compresión de datos, logrando así una optimización del ancho de banda. Por ejemplo, un canal analógico tiene un ancho de banda típico de 6 MHz en América y 8 MHz en Europa. En este mismo ancho de banda pueden transmitirse hasta 5 canales de video y audio digital. El algoritmo más utilizado para la compresión de video y audio se llama MPEG-2 (Moving Pictures Experts Group). La digitalización también posibilita resoluciones superiores, efectos en el sonido, y sobre

todo, mejores usos del ancho de banda. Entre la información adicional que proporciona un canal digital de televisión se encuentra, fecha, hora, descripción del programa, subtítulos en varios idiomas (Closed Captioning), audio en varios idiomas, guía de programación en tiempo real, búsquedas, etc.

Quizá la característica más importante de la TV digital, para los padres de familia, es la clasificación de los programas. Esto permite controlar la televisión para que a infantes y adolescentes les sean restringidos programas no aptos para su edad. Con la televisión analógica no se puede efectuar esta función.

1.4. VENTAJAS DE LA TDT

- Mejor Calidad de recepción

La TDT permite una mejora en la calidad de la recepción y amplía la oferta disponible tanto en número de canales como en versatilidad del sistema.

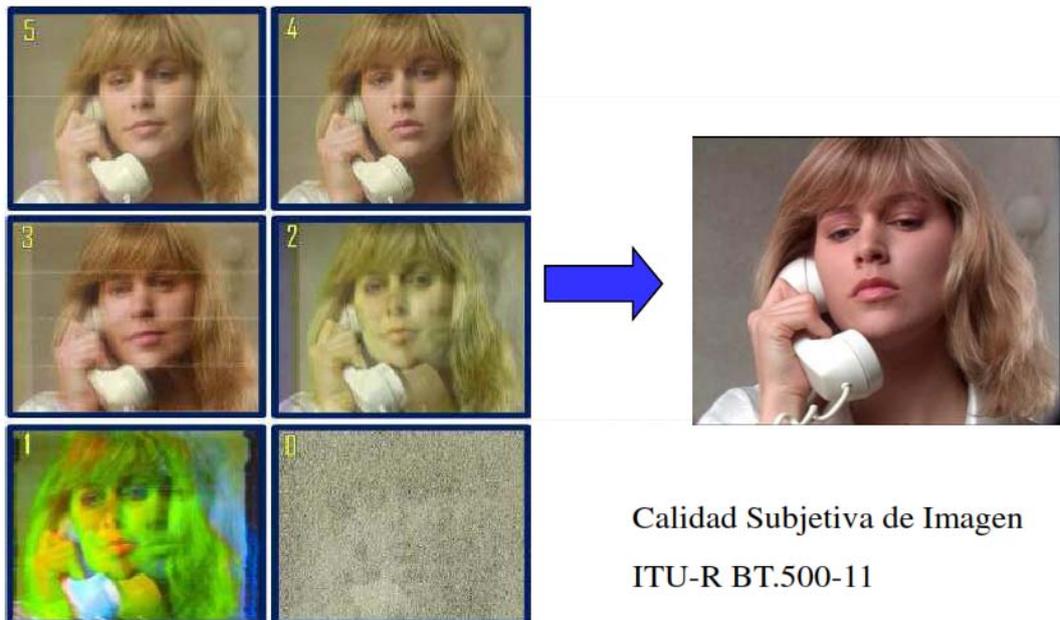


Figura I.2. Calidad de Imagen

- Múltiples Dispositivos

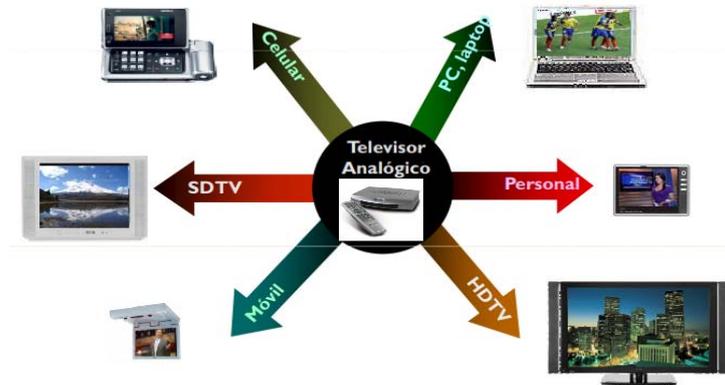


Figura I.3. Dispositivos Múltiples en TDT

- Múltiple Programación

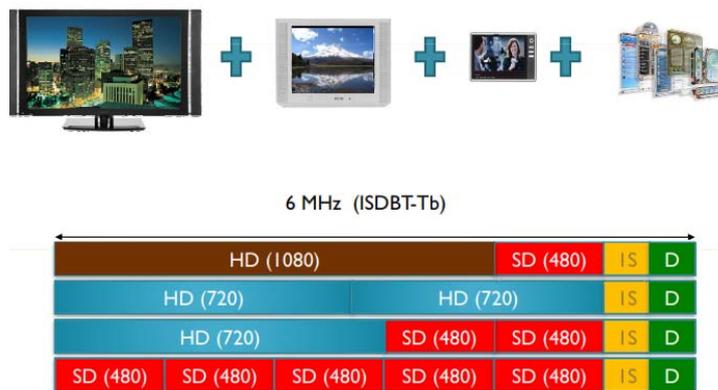


Figura I.4. Programación Múltiple en TDT

- Segmentación

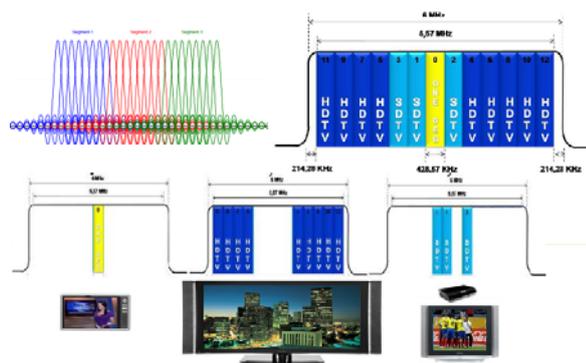


Figura I.5. Segmentación

- **Menor Potencia Media**

Canal: 644 MHz

Ganancia de antena de 10 dB



Figura I.6. Transmisión de TV analógica a 4 kw

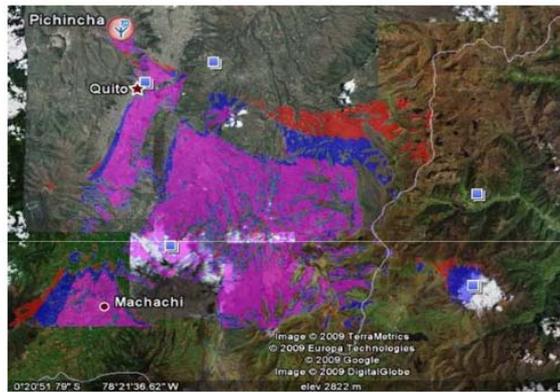


Figura I.7. Transmisión de TV digital a 1 kw

- **Guía de Programación Electrónica EPG**



Figura I.8. Ejemplo de EPG

- Prevención de Desastres – Alertas de emergencia

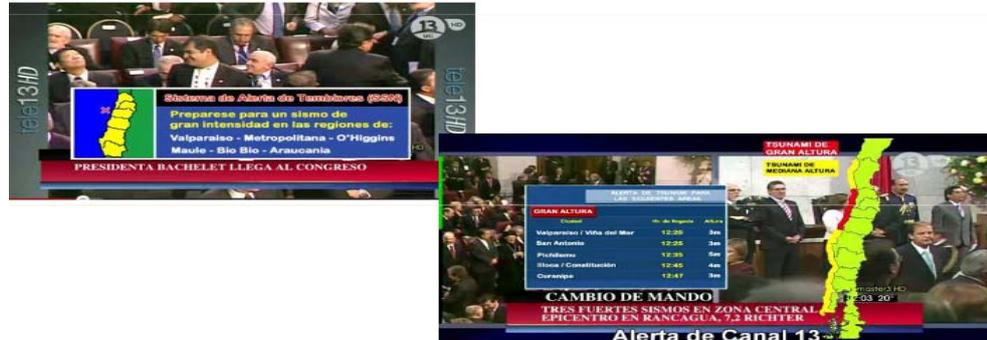


Figura I.9. Ejemplo de Alertas con TDT

- Mejor calidad de Audio y Video.
- Permite multiprogramación.
- Optimización del Espectro Radioeléctrico.
- Interactividad local y completa.
- Teletexto CC (Close Caption)
- Menor potencia media en el transmisor

1.5. INTERACTIVIDAD

La interactividad es la **capacidad de ofrecer contenidos adicionales a los programas de televisión, permitiendo al usuario ver informaciones asociadas al contenido audiovisual**, la programación de los canales, participar en concursos, votaciones, comprar productos o servicios, e incluso participar en los propios programas de televisión con el mando a distancia. La interactividad es **posible gracias a aplicaciones que complementan la programación**, siendo el usuario el que decide si quiere o no verlos, y cuando verlos.

La interactividad, le ofrece al espectador la posibilidad de **personalizar el contenido que muestra su televisor**, bien sea accediendo a información enviada durante el proceso de emisión, pero que solo se hace visible si el espectador lo desea, o bien accediendo a servidores con los que puede intercambiar información, a través de un canal de retorno utilizando el televisor como interfaz de salida.

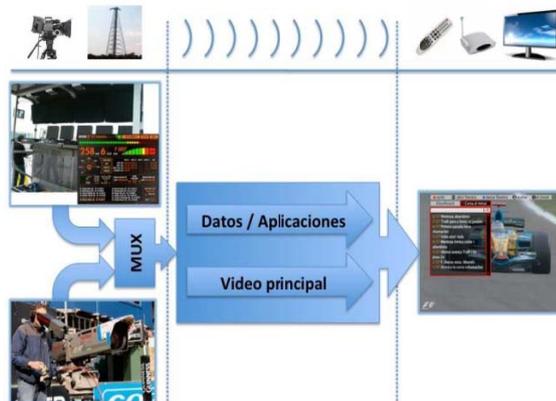


Figura I.10. Interactividad

1.5.1 ¿QUÉ SERVICIOS INTERACTIVOS OFRECE LA TDT?

En sí, la TDT **no ofrece ningún servicio interactivo**. Dispone de atractivos menús, guías de programación y teletextos mejorados. No obstante, **diversas televisiones y decodificadores** (los de gama alta, a partir de 250€), **sí permitirán esta interacción**, al incluir en muchos de los casos **discos duros** internos y un **módem** que **se comuniquen con el exterior**, los denominados MHP. Con este tipo de aparatos, que serán los más populares, **sí se permitirán los servicios interactivos, como votar en nuestro reality favorito o participar en chats.**

1.5.2 ¿QUÉ APORTA LA INTERACTIVIDAD?

La interactividad va a permitir a los canales de televisión ofrecer un importante conjunto de **servicios al ciudadano**, que permitan explorar nuevas formas de hacer televisión, incorporando funciones avanzadas de comunicación y participación, y servicios sociales para el desarrollo de la sociedad de la información. Por el lado de los usuarios, la interactividad va a permitir **acceder a nuevos contenidos**, a una televisión mucho más rica y completa, con la posibilidad de **participar e influir en los programas de televisión**.

1.5.3 VENTAJAS DE LA INTERACTIVIDAD

La interactividad permite **complementar los contenidos de televisión**, tanto a través de servicios públicos (ayuntamientos, gobiernos, sanidad, sectores desprotegidos, etc.) como servicios comerciales o de entretenimiento (votaciones, concursos, publicidad interactiva, etc.) que hasta ahora solo eran accesibles a través de otros medios como ordenador o teléfono móvil.

La principal ventaja de la interactividad en televisión, radica en la posibilidad de acceder a un amplio conjunto de **servicios públicos o privados a través del televisor**, con un único terminal, y un mando a distancia. Otra ventaja de la interactividad radica en que es **el propio usuario, si quiere o no ver los servicios interactivos y los contenidos asociados a la interactividad** (por ejemplo, si quiere o no ver los mensajes que los usuarios envían a los programas tipo SMS). Finalmente, la interactividad en televisión permite

ofrecer **servicios adaptados a las necesidades** de los diferentes colectivos que conforman la sociedad, independientemente de la edad, y la localización.

1.5.4 AGENTES QUE INTERVIENEN EN LA INTERACTIVIDAD

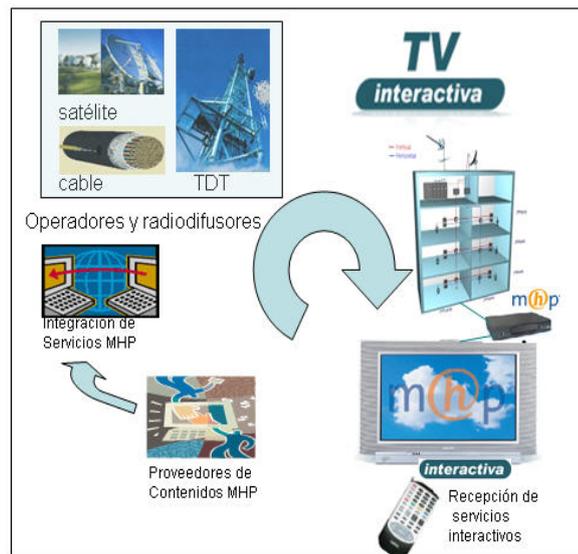


Figura I.11. Agentes de interactividad en TDT

1.5.5 ESCENARIOS DE USO DE LA INTERACTIVIDAD

La interactividad puede ser de dos tipos:

1.5.5.1 INTERACTIVIDAD LOCAL:

El **espectador interactúa con la información que está almacenada en el receptor**, la cual se renueva con cierta periodicidad.

Con la interactividad local, el **usuario puede acceder a contenidos interactivos pero no puede enviar datos de vuelta**. Ejemplos de aplicaciones interactivas locales son las Guías

de Programación, el Teletexto Digital, o información sobre los participantes en un programa.

Únicamente trabaja con aplicaciones transmitidas por *Broadcast* y descargadas en el *Set-top-Box*.



Figura I.12. Interactividad Local

1.5.5.2 INTERACTIVIDAD COMPLETA

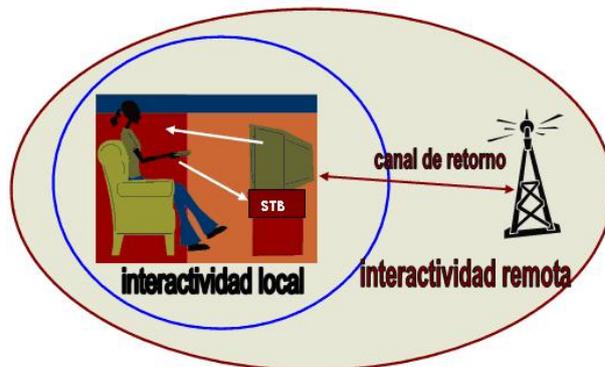


Figura I.13. Canal de Retorno

El espectador interactúa con un proveedor de servicios exterior, al que se conecta mediante un canal de retorno.

La interactividad con canal de retorno permite no solo ver contenidos adicionales a la programación y navegar por ellos, sino también enviar respuestas por parte de los usuarios, e incluso comunicarse con otros usuarios. La interactividad con canal de retorno es la que permite a los usuarios participar en concursos, votar, o enviar mensajes a otros usuarios.

El espectador interactúa con un proveedor de servicios exterior, al que se conecta mediante un canal de retorno.



Figura I.14. Interactividad Completa

1.6. SERVICIOS INTERACTIVOS

Se pueden establecer tres categorías de servicios interactivos:

- **Servicios de información:**

Son aquellos que ofrecen una información independiente de la programación audiovisual que se está emitiendo en ese momento.

- **Servicios ligados a la programación:**

Son aquellos que complementan con información suplementaria la programación audiovisual emitida.

| TIPOS DE SERVICIO | INTERACTIVIDAD LOCAL | INTERACTIVIDAD REMOTA |
|-------------------------------------|---|--|
| Servicios de información | <ul style="list-style-type: none">• Guía electrónica de Programación• Teletexto mejorado• juegos | |
| Servicios ligados a la programación | <ul style="list-style-type: none">• Estadísticas deportivas• Información ampliada sobre el programa• Noticias desarrolladas | <ul style="list-style-type: none">• Participación en concursos• Encuestas• Preguntas y entrevistas a invitados• Pago por visión |
| Servicios transaccionales | | <ul style="list-style-type: none">• Reserva de plazas• Consultas bancarias• T-administración• Compras |

Tabla I.1. Interactividad Local vs Interactividad Remota

- **Servicios transaccionales:** Son aquellos que ofrecen la posibilidad de enviar y recibir información de forma personalizada y exclusiva.

Los servicios interactivos **se implementan con aplicaciones interactivas**. Podemos definirlos como aquellos **programas adicionales a los contenidos de televisión que pueden verse a través de un decodificador interactivo**. El usuario es el que decide si quiere o no ver las aplicaciones interactivas mediante una acción simple con el mando a

distancia. Con el fin de comunicar al usuario la posibilidad de acceso a aplicaciones interactivas los operadores o canales de televisión presentan un pequeño menú interactivo que indica al usuario que puede ver una aplicación o un grupo de aplicaciones interactivas

Las aplicaciones interactivas pueden ser de muy diversos tipos, pero de manera general se pueden dividir en:

- **Servicios públicos**

- Servicios de información (estado del tráfico, tiempo, farmacias de guardia, teléfonos de interés, información de aeropuertos, trenes, autobuses, etc.)
- Servicios avanzados como servicios sanitarios, gestión de impuestos, informaciones y servicios de ayuntamientos y administraciones públicas, servicios sociales a colectivos específicos (tercera edad, etc.).
- Servicios de entretenimiento o comerciales.
- Fidelización a contenidos o programas, concursos, votaciones, publicidad interactiva, venta por impulso, compra de eventos (fútbol, cine, conciertos, etc.)
- Informaciones sobre concursantes, actores, etc., y pueden tanto independientes del contenido, como integradas y sincronizadas con el mismo (concurso interactivo al mismo tiempo que el programa).

1.7 ESTÁNDARES DE TDT EN EL MUNDO



Figura I.15. Estándares TDT en el mundo

En marzo del 2010 Ecuador adoptó el estándar de televisión digital terrestre japonés – brasileño ISDB -Tb.

1.8 DESARROLLO DE CONTENIDOS DE TDT

Existen algunas herramientas para el desarrollo de Contenidos Interactivos para TV Digital entre las cuales citaremos las siguientes:

- *Ginga-NCL*
- *Ginga-Lua*
- *Ginga-Java*

Mediante el uso de estas herramientas se ofrece la oportunidad de divulgar la creatividad y habilidad en el desarrollo de contenidos interactivos en varios campos como :

- Tele Educación
- Tele Salud
- Prevención de Catástrofes
- E- learning

CAPÍTULO II

GINGA NCL

2.1 INTRODUCCIÓN

El Ginga-NCL fue desarrollado por la Pontificia Universidad Católica de Rio de Janeiro PUC-Rio, provee una infraestructura de presentación para aplicaciones interactivas de tipo declarativas escritas en el lenguaje NCL (Nested Context Lenguaje).

NCL es una aplicación de XML (eXtensible Markup Language) con facilidades para los aspectos de interactividad, sincronismo, espacio-temporal entre objetos de mídia, adaptabilidad, soporte a múltiples dispositivos y soporte a la producción de programas interactivos en vivo no-lineares.

El NCL es un lenguaje del tipo basado en la estructura que define una separación bien demarcada entre el contenido y la estructura de un aplicativo, permitiendo definir objetos de media estructurados y relacionados tanto en tiempo y espacio.

2.2 PARADIGMAS DE LA PROGRAMACIÓN

2.2.1 LENGUAJE DECLARATIVO

Provee facilidades para especificar aspectos de interactividad, entre objetos de multimedia, y soporte para múltiples dispositivos, es decir construir aplicaciones

El Lenguaje Declarativo es un modelo de programación que está basado en el desarrollo de programas especificando o “declarando” un conjunto de condiciones, preposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución.

2.2.2 LENGUAJE IMPERATIVO

Es el estilo de programación convencional, donde las aplicaciones se descomponen en pasos de cálculo que nos dan una especificación de algoritmos.

A este lenguaje se lo conoce como lenguaje procedimental, donde los programas son realizados mediante funciones, devuelve un nuevo estado de resultado y recibe como entrada el resultado de otras funciones.

Se caracteriza porque en ellos indican al ordenador que debe hacer y como debe hacerse.

2.3 CONCEPTO DE GINGA

Middleware es un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y sistema operativo). Un middleware para aplicaciones de TV digital consta de máquinas de ejecución de los lenguajes ofrecidos y librerías de funciones, que permiten el desarrollo rápido y fácil de aplicaciones.



Figura II.16: Ginga Ecuador

Ginga es el nombre del middleware de la Recomendación ITU-T para servicios de IPTV y del Sistema Nipo-Brasileño de TV Digital Terrestre (ISDB-T_B). Está formado por un conjunto de tecnologías estandarizadas e innovaciones brasileñas que lo convierten en la especificación de middleware más avanzada.

2.4 ¿POR QUÉ EL NOMBRE DE GINGA?

Ginga es una cualidad, casi indefinible, de movimiento y actitud que poseen los brasileños y que es evidente en todo lo que hacen. La forma en la que caminan, hablan, bailan y se relacionan con todo en sus vidas. La **ginga** es un movimiento fundamental de la **capoeira**, una forma de lucha por la libertad y la igualdad. El nombre **Ginga** fue escogido en

reconocimiento a la cultura, el arte y la continua lucha por la libertad y la igualdad del pueblo brasileño.

Esa misma lucha estuvo presente en el proceso de desarrollo de **Ginga**, tanto en los varios años de arduo trabajo en la PUC-Rio y en la UFPB, cuanto recientemente rompimos varias barreras para tornarlo la única innovación brasileña que compone el Sistema Nipo-Brasileño de TV Digital.



Figura II.17: Movimiento Fundamental de la Capoeira

2.5 ¿POR QUÉ GINGA ES SOFTWARE LIBRE?

Desde su concepción, **Ginga** tuvo en consideración la necesidad de la inclusión social/digital y la obligación de compartir el conocimiento de forma libre.

Ginga es una tecnología que le da al ciudadano todos los medios para que obtenga acceso a la información, educación a distancia y servicios sociales a través de su TV, el medio de comunicación omnipresente.

Ginga tiene en consideración la importancia de la televisión, presente en la totalidad de los hogares, como un medio complementario para la inclusión social/digital, es una especificación abierta, de fácil aprendizaje, permitiendo que todos produzcan contenido

interactivo, lo que dará un nuevo impulso a las TVs comunitarias y a la producción de contenido por las grandes emisoras.

El entorno declarativo de **Ginga**, llamado **Ginga-NCL** tiene también una implementación de referencia en código abierto, desarrollada por el Laboratorio TeleMídia de la PUC-Rio.

Adoptando la licencia GPLv2, el laboratorio TeleMídia garantiza el acceso permanente a toda la evolución del código publicado en la Comunidad Ginga, sean cuales fueren sus aplicaciones y autores de aquí en adelante.

2.6 ARQUITECTURA DEL MIDDLEWARE GINGA

Middleware es un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y sistema operativo). Un middleware para aplicaciones de TV digital consta de máquinas de ejecución de los lenguajes ofrecidos y librerías de funciones, que permiten el desarrollo rápido y fácil de aplicaciones.

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 se muestra en la (figura II.18)

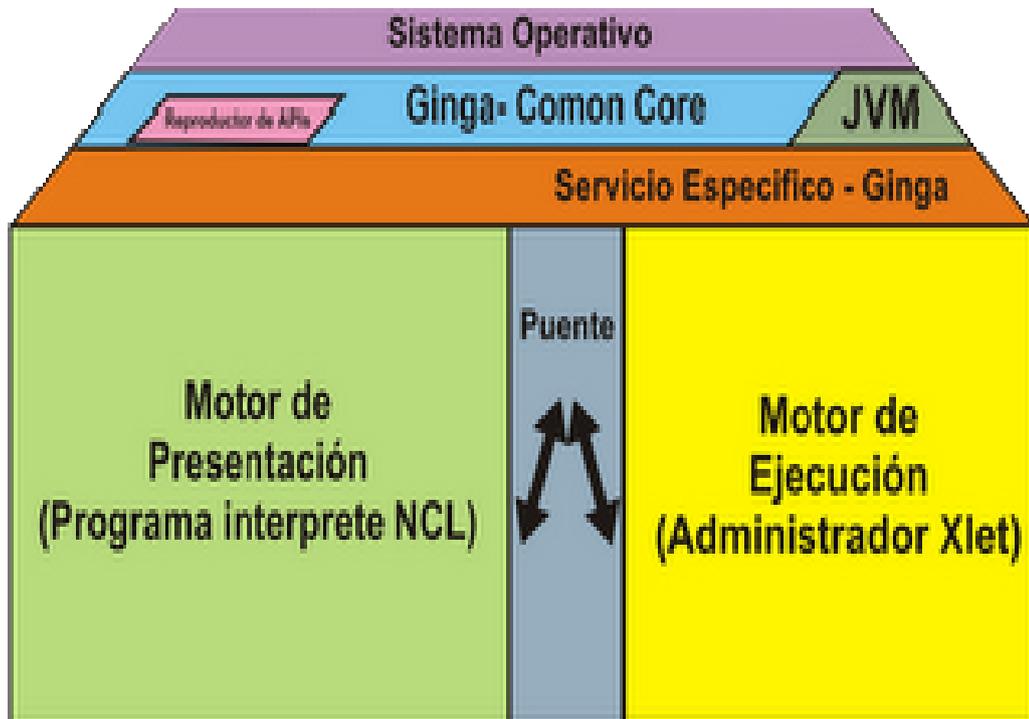


Figura II.18 Arquitectura de Ginga

2.7 SUBSISTEMAS DE GINGA

El middleware abierto Ginga se subdivide en tres subsistemas principales interrelacionados, que permiten el desarrollo de aplicaciones siguiendo dos paradigmas de programación diferentes. Dependiendo de las funcionalidades requeridas en cada proyecto de aplicación, un paradigma será más adecuado que otro. Estos tres subsistemas se llaman: Núcleo Común de Ginga (Common Core), Ginga-J (para aplicaciones procedurales Java) y Ginga-NCL (para aplicaciones declarativas NCL).

2.7.1 Ginga (Comon-Core)

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.

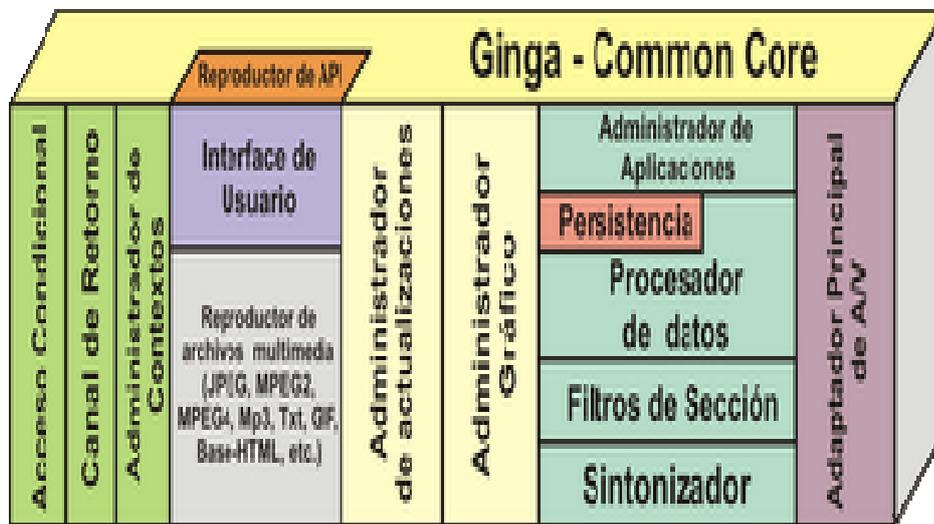


Figura II.19. Componentes de Ginga – Common Core

En la Figura II.19, 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.

2.7.2 Ginga-J

Provee una infraestructura de ejecución de aplicaciones procedurales Java y extensiones específicamente adaptadas al ambiente de la TV

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.

Por otro lado, ya que 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 2.5). 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 SunJavaTV, 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.

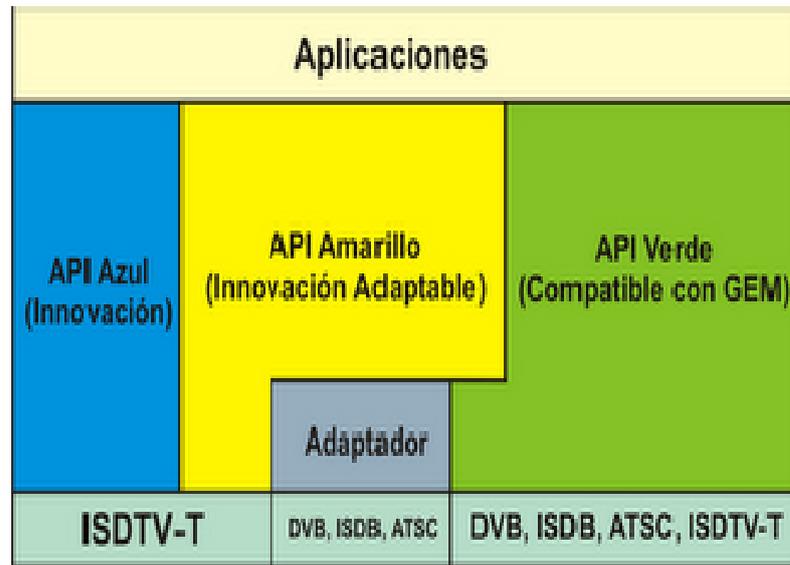


Figura II.20. APIs de Ginga-J

2.7.3 GINGA-NCL

Es un entorno de presentación multimedia para aplicaciones declarativas escritas en NCL y su lenguaje de scripting Lua.

NCL es una aplicación XML (eXtensibleMarkup Language), con facilidades para especificar los aspectos de la interactividad, en tiempo-espacio entre los objetos de los medios de comunicación, la adaptabilidad, soporte a múltiples dispositivos y producción en vivo de programas interactivo no lineares.

Siendo una aplicación XML, el lenguaje de NCL posee una estricta separación entre el contenido y la estructura, NCL no define los medios de comunicación por sí mismos. En cambio, mantiene los medios de comunicación, junto con una presentación multimedia. Sin embargo, un documento de NCL sólo define cómo los objetos de los medios de

comunicación están estructurados y relacionados en el tiempo y el espacio. [El video (MPEG, etc.), audio (AAC, etc.), imagen (JPEG, GIF, etc.) y texto(TXT, HTML, etc.), son ejemplos de objetos de medios de comunicación.

Entre estos objetos se destacan los objetos de vídeo y audio en SBTVD , Otro objeto importante es aquel basado en XHTML.

NCL no sustituye a XHTML sino que complementa lo que es incapaz de cumplir como un lenguaje declarativo. A diferencia de XHTML el lenguaje NCL no mezcla la definición de contenido de un documento con su estructura. El ambiente declarativo es en sí muy limitado. Las aplicaciones que utilizan un lenguaje declarativo debe tener su enfoque sobre el sincronismo, siendo el foco de la lengua NCL exactamente eso, no la interactividad, ya que la interacción es tratada como resultado de la sincronización. En NCL, un documento XHTML es un tipo de elemento de los medios de comunicación.

De forma semejante, lenguajes imperativos pueden ser adicionados y utilizados como medios de comunicación.

2.7.3.1 LENGUAJE DE PROGRAMACION DE GINGA NCL

Es un lenguaje declarativo, es decir, en contraposición a los lenguajes imperativos estos lenguajes describen el problema a solucionar y no recetan un algoritmo a seguir paso a paso, como es el caso de los lenguajes imperativos, sino que la máquina mediante métodos internos de inferencia, deduce información y toma decisiones respecto a la descripción del problema mediante ecuaciones y sentencias de restricciones.

En general el objetivo es ofrecer un lenguaje que sea lo suficientemente simple como para dar masividad al uso de la tecnología, junto con no limitar la creatividad de diseño. Muchas veces simplificando mucho un lenguaje, se limitan las posibilidades que se pueden lograr con el, por lo que es importante encontrar un equilibrio o una forma de explotar estas dos ideas importantísimas.

Es por estas razones que se opta por darle preferencia al lenguaje declarativo Ginga-NCL, un lenguaje más simple con el que muchas problemáticas son resueltas internamente por el sistema.

Uno de los objetivos centrales de esta tecnología, junto con converger el mundo analógico al mundo digital, e intentar manejar todo de manera centralizada; es darle interactividad con el usuario a la televisión.

Ginga abre las puertas para que el usuario maneje lo que ve de múltiples maneras, como por ejemplo, visualizar dos programas al mismo tiempo y canalizar el audio por distintas vías para dos usuarios simultáneos.

También se pueden repetir escenas, grabar con facilidad, e incluso conectarse a internet e interactuar online, como por ejemplo comprando artículos publicitados, ropa, comida, y accesorios de lujo.

2.7.3.2 COMPONENTES DE GINGA NCL

Los componentes de este subsistema son (figura II.21):



Figura II.21 Subsistema Ginga-NCL

A continuación se define los elementos principales de Ginga-NCL:

- **Formateador (Formatter):**

Quien se encarga de recibir y controlar las aplicaciones multimedia escritas en NCL.

Dichas aplicaciones son entregadas al Formateador por el Ginga-CC.

- **Analizador de XML (XML Parser), Convertidor (Converter):**

Realizan la traducción de la aplicación NCL en la estructura interna de datos de Ginga-NCL para controlar la aplicación. Estos componentes son solicitados por el Formateador.

- **Programador (Scheduler):**

Es iniciado para organizar el orden de la presentación del documento NCL (antes que inicie los objetos de media, se evalúan las condiciones de los enlaces y la

programación correspondiente a las relaciones de las acciones que guiaran el flujo de la presentación). El componente Programador es responsable para dar la orden al componente Administrador de la Reproducción (Player Manager) para iniciar la reproducción apropiada del tipo de contenido de media para exhibirlo en el momento indicado.

- **Base Privada (Private Base):**

El Motor de Presentación (Presentation Engine) lidia con un conjunto de aplicaciones NCL que están dentro de una estructura conocida como Base Privada.

- **Administrador de la Base Privada (Private Base Manager):**

Este componente está a cargo de recibir los comandos de edición de los documentos NCL y el darle mantenimiento a los documentos NCL presentados.

Estos comandos de edición están divididos en tres subgrupos:

- 1er Grupo de Comandos, responsable por la activación y desactivación de una base privada, o sea, la habilitación de una determinada aplicación NCL.
- 2do Grupo de Comandos, responsable de iniciar, pausar, resumir, detener, remover las aplicaciones NCL.
- 3er Grupo de Comandos, responsable de la actualización de aplicaciones en tiempo real, permitiendo el agregar o remover elementos NCL y permite que se asignen valores a las propiedades de los objetos de media.

- **Administrador del Diseño (Layout Manager):**

El Motor de Presentación soporta múltiples dispositivos de presentaciones a través del componente Administrador del Diseño, el cual es responsable de mapear todas las regiones definidas en una aplicación NCL.

2.7.3.3 ESTRUCTURA BÁSICA DE UN MÓDULO 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 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).

Aquí 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:

| Elemento | Elemento hijo |
|----------|---|
| <head> | <importedDocumentBase>, <ruleBase>, <transitionBase>, <regionBase>, <descriptorBase>, <connectorBase>, <meta> y <metadata>. |

TABLA II.2. Elementos <Head>

El elemento <body> puede contener los elementos hijo

| Elemento | Elemento hijo |
|-------------------------|---|
| <body> <context> | <port>, <attribute>, <media>, <context>, <switch> y <link>. |

TABLA II. 3. Elementos <body>, <context>

El elemento **<body>** es tratado como un nodo de contexto NCM. 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 menciono. Al igual que el elemento <body>, un elemento <context> puede tener elementos hijo

<port>, <attribute>, <media>, <context>, <switch>, y <link>.(Ver **TABLA II. 3**)

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.

La 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 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 switchs 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 <causal Connector> 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 <descriptor Switch> contiene un grupo de descriptores alternativos para ser asociados con un objeto NCL. Similar al elemento <switch>, un <descriptor Switch> la selección se realiza durante la presentación del documento, utilizando las reglas de prueba definidas por el elemento <rule> o <composite Rule>.

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 módulo 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 módulo 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.

2.7.3.4 VENTAJAS DEL LENGUAJE NCL

- ✓ Compatibilidad con otros sistemas.
- ✓ El Ginga-NCL permite la adición de players BML, DVB-HTML y ACAP/OCAP XHTML, permitiendo de esta manera ser extendido (inclusive vía broadcast) para funcionar en otros países, con sistemas diferentes.
- ✓ Hace viable la Interactividad, ya que define una interfaz estandarizada que garantiza que las aplicaciones y los contenidos interactivos enviados vía Broadcast sean visualizados correctamente en cualquier aparato receptor.
- ✓ Flexibilidad para que el Radiodifusor pueda crear y moldar servicios avanzados de acuerdo con sus necesidades de negocio.
- ✓ Portabilidad de las aplicaciones residentes, tornando posible el desarrollo de aplicaciones de TV Digital para ser embarcadas en cualquier receptor independiente de la plataforma, modelo y/o fabricante.
- ✓ Permite la interacción simultánea de varios usuarios utilizando diferentes dispositivos;
- ✓ Permite la integración del receptor con otros dispositivos que puedan estar a su
- ✓ alrededor.

2.7.3.5 ESTRUCTURA DE UN DOCUMENTO NCL

Un Documento NCL está formado básicamente de la siguiente manera



FIGURA II.22. 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;
- Una sección de encabezado de programa (sección head), donde se definen las regiones, los descriptores, los conectores y las reglas utilizadas por el programa;
- Un cuerpo de programa (sección body), donde se definen los contextos, nodos multimedia, enlaces y otros elementos que definen el contenido y la estructura del programa;

- Por lo menos una puerta que indica por donde el programa comenzara a ser exhibido (portInicio);y
- La conclusión del documento.

La Tabla II.4. Presenta la estructura básica de un documento NCL.

| | | |
|--|---|---|
| <p>Encabezado del Archivo NCL</p> | <p>1:<?xmlversion="1.0"encoding="ISO-8859-1"?></p> <p>2:<nclid="ejemplo01"xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile"</p> <p>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd"></p> | <p style="text-align: center;">1</p> |
| <p>Encabezado del Programa</p> | <p>3:<head></p> | |
| | <p>Base de Regiones</p> <p>4: <región Base></p> <p>5: <!--regiones de la pantalla en donde se presentaran los archivos multimedia--></p> <p>6: </región Base></p> |  |
| | <p>Base de Descriptores</p> <p>7: <descriptor Base></p> <p>8: <!--descriptores que define como los archivos multimedia Serán presentados--></p> <p>9: </descriptor Base></p> | |
| | <p>Base de Conectores</p> <p>10: <connector Base></p> <p>11: <!--conectores que define como los enlaces son activado y las acciones que ellos ejecutan--></p> <p>12: </connector Base></p> | |
| | <p>13:</head></p> | |

| | | | | |
|--------------------------------|--|--|-----|--|
| Cuerpo del Programa | 14:<body> | | | |
| Puerta de Entrada en el | 15: <port id="plnicio" component="ncPrincipal" interface="Inicio"/> | | | |
| Contenido del Programa | 16: <!-- contextos, nodos multimedia y sus anclas, enlaces y otros elementos --> | | 6 7 | |
| | 17:</body> | | | |
| Fin | 18:</ncl> | | | |

TABLA II.4. ESTRUCTURA 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 (descriptor Base);
- 4) El contenido (nodos multimedia-media) y la estructura (contextos-context) 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 (área y atributte);
- 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).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="ejemplo1" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>

  </head>

  <body>

  </body>
</ncl>
```

FIGURA II.23 ESTRUCTURA GENERAL DE UN DOCUMENTO NCL

2.7.3.6 RELACIÓN ENTRE ENTIDADES DE UN DOCUMENTO NCL

Para construir un documento hipermedia, es necesario definir **qué se va a reproducir** (Objeto Media), **dónde los vamos a mostrar** (es decir, la región de la pantalla que se va utilizar), **como** (por ejemplo, el volumen, con o sin borde, con qué reproductor,

Descriptores), **Cuando** (Links y Conectores, antes o después de presionar un botón, que archivo multimedia será reproducido) y finalmente **Como reusamos código** (Contexto).



FIGURA II.24 RELACIÓN ENTRE ENTIDADES DE UN DOCUMENTO NCL

2.7.3.6.1 ¿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 II.25.



Figura II.25. 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 II.26 se muestra un documento con cuatro nodos multimedia, tres de los cuales están dentro de un Contexto (ctx1) anidado al cuerpo (body).



Figura II.26. Representación de nodos multimedia y su composición

2.7.3.6.2 ¿Dónde reproducir?

A medida que se define el contenido de un programa, también se debe comenzar a definir las **áreas** en donde 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.

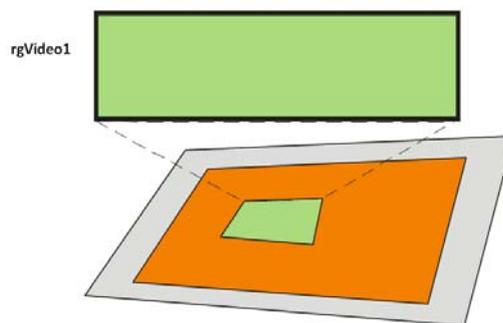


Figura II.27. Representación de una Región

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.

2.7.3.6.3 ¿Cómo reproducir?

La asociación de un archivo multimedia a una región se define a través de un descriptor. Los descriptores también se utilizan para definirla 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 II.28). Todos los archivos multimedia que utilizan dicho descriptor están asociados con la región correspondiente.

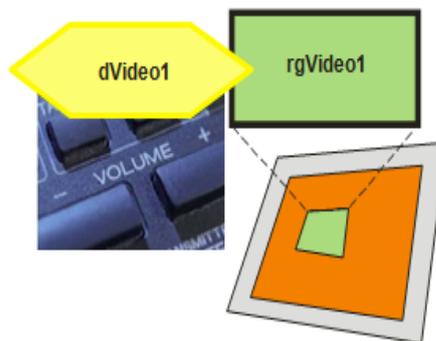


Figura II.28. Representación de un descriptor asociado a una región

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 II.29 se ilustra un descriptor dVideo1 utilizado por un archivo multimedia que será presentado en la región rgVideo1, sin ningún cambio en la forma de cómo será presentado.

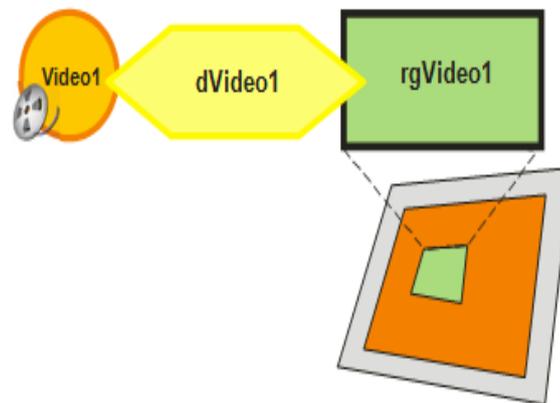


Figura II.29. Representación de un descriptor asociado a un archivo multimedia que será presentado en una región

2.7.3.6.4 ¿CUANDO 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 II.29. el nodo video1 del contexto ctx1 solo 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.

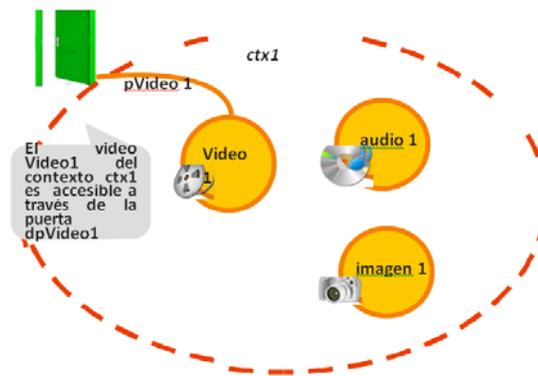


Figura II.30. Puerta de un nodo de composición

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.

CAPÍTULO III

INSTALACIÓN GINGA NCL Y ECLIPSE

3.1 INSTALACIÓN GINGA NCL

El emulador Ginga-NCL 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.

3.1.1 REQUERIMIENTOS DE HARDWARE

Computador PC con Sistema Operativo Windows (XP, Vista, WIN7)

- 32 o 64 bits.
- Mínimo P4.
- Mínimo 512 MB en RAM.
- 5 GB de Espacio Libre en Disco Duro(Recomendado)

3.1.2 REQUERIMIENTOS DE SOFTWARE

- VMWare Player 3.1 o superior.
- Máquina Virtual de GINGA.
- Eclipse 3.2 o superior.
- NCL Eclipse.

3.2 INSTALACIÓN DE VMWare Player

La instalación de imagen de la máquina virtual de Ginga es el procedimiento más sencillo para iniciar con la programación en NCL.

Además de la máquina virtual se requiere asistentes para cargar los archivos NCL a la misma en nuestro caso hemos utilizado ECLIPSE.EXE.

Los Pasos a seguir para la instalación son los siguientes.

Paso 1.

Se descarga la máquina virtual que ofrece la PUC-Rio de la siguiente dirección:

www.ncl.org.br/ferramentas/ubuntu-server10.10-ginga-i386.zip y descomprimirla en

cualquier lugar del disco duro.

Paso 2.

Se ejecuta VmWare donde tendremos una interfaz como muestra la figura

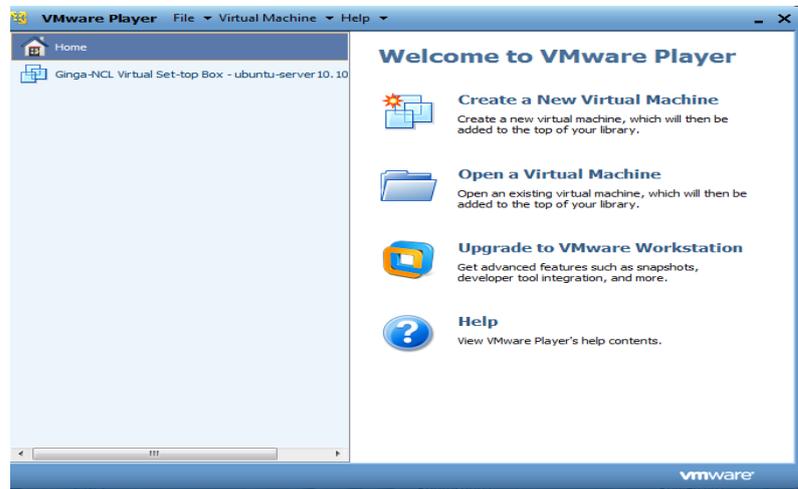


Figura III.31. Pantalla Principal de VmWare Workstation

Paso 3. CARGAR LA MAQUINA VIRTUAL GINGA

Seleccionamos el archivo "ubuntu-server10.10-ginga-i386.vmx" que descomprimimos

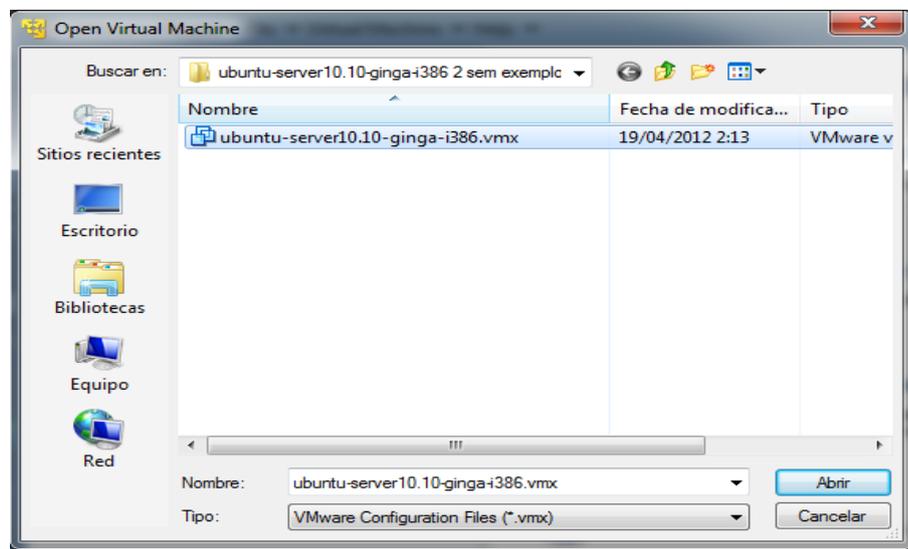


Figura III.32. Pantalla ubuntu-server10.10-ginga

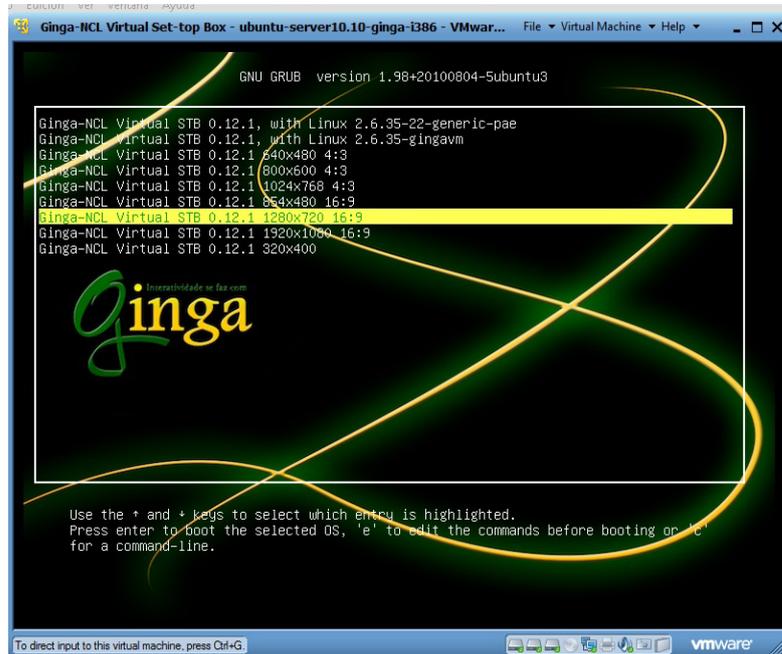


Figura III.33. Pantalla ginga-NCL STB

Paso 4. Una vez seleccionado el archivo ubuntu-server10.10-ginga-i386.vmx se tendrá la siguiente ventana (Figura III.34.), se debe hacer clic en Play virtual machine para iniciar la máquina virtual.

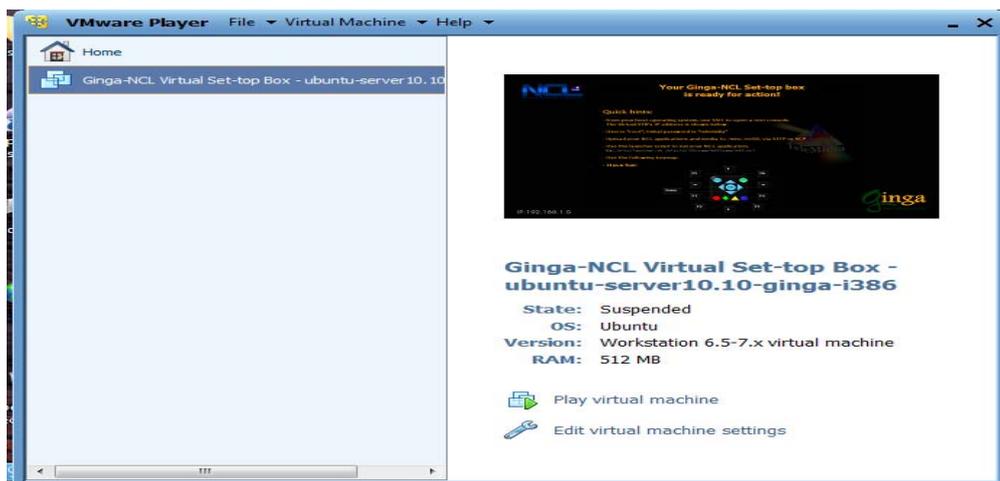


Figura III.34. Pantalla ubuntu-server10.10-ginga

Finalmente se mostrará la siguiente pantalla, como resultado se tendrá el Ginga-NCL Virtual Set Top Box ejecutándose, listo para la carga de aplicaciones en entorno NCL



Figura III.35. Pantalla principal de la maquina virtual de Ginga una vez ejecutada

3.3 Eclipse NCL

El entorno de desarrollo integrado (IDE) Eclipse permite crear y editar aplicaciones NCL

3.3.1 Plugin NCL Eclipse

Provee funcionalidades tales como:

- ✚ Creación de documentos .ncl
- ✚ Coloreado de sintaxis
- ✚ Autocompletado y autoformato de código
- ✚ Indicación de errores

3.3.2 INSTALACIÓN Eclipse NCL.

Su instalación es muy sencilla, podemos descargarlo de www.eclipse.org en forma de archivo ZIP y solo tenemos que descomprimir eclipse-SDK-3.1.7-win32.zip en la carpeta donde queramos tenerlo instalado(Figura III. 36.).

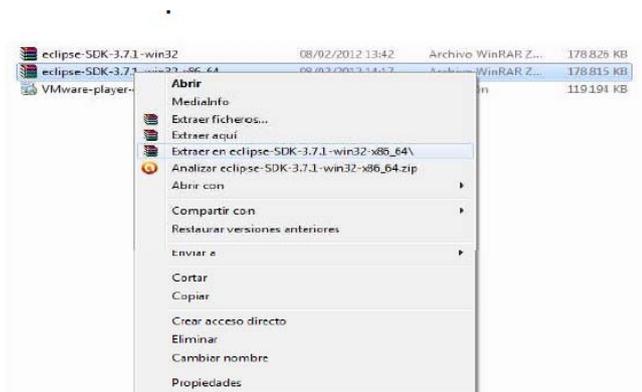


Figura III.36. Pantalla descompresión eclipse-SDK-3.1.7-win32.zip

Para ejecutarlo solo hay que arrancar el fichero Eclipse.exe. Una vez arrancado lo único que nos pedirá es que le demos la ruta por defecto donde queramos que eclipse nos vaya guardando los proyectos que creamos:

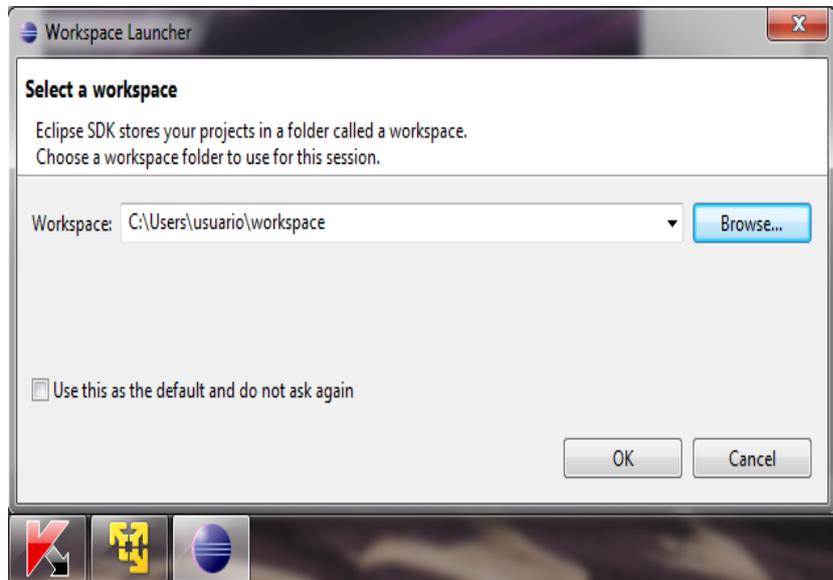


Figura III.37. Pantalla de ubicación de eclipse-SDK-3.1.7-win32.zip

Después de esto nos aparecerá la ventana principal de Eclipse:

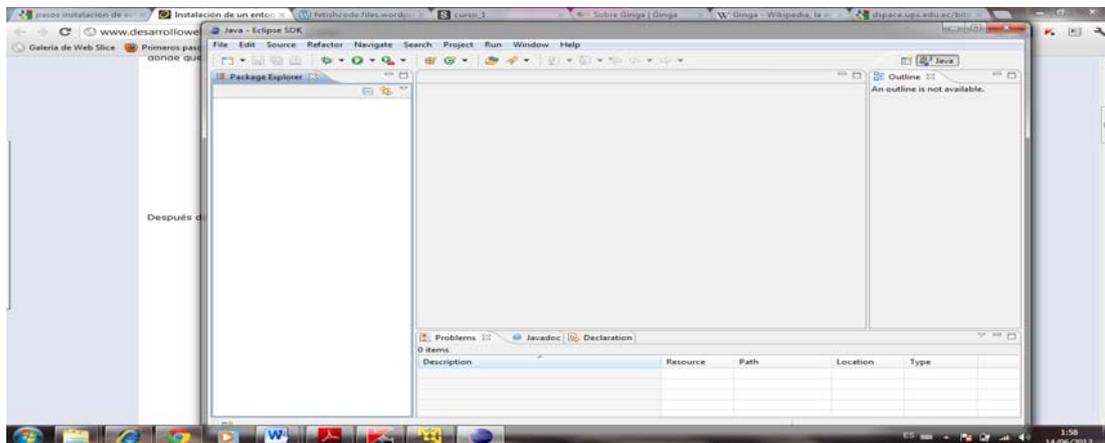


Figura III.38. Pantalla ventana principal eclipse-SDK-3.1.7

A continuación dentro de la ventana principal de Eclipse abrimos la ventana Help>Install New Software(Figura III.39)

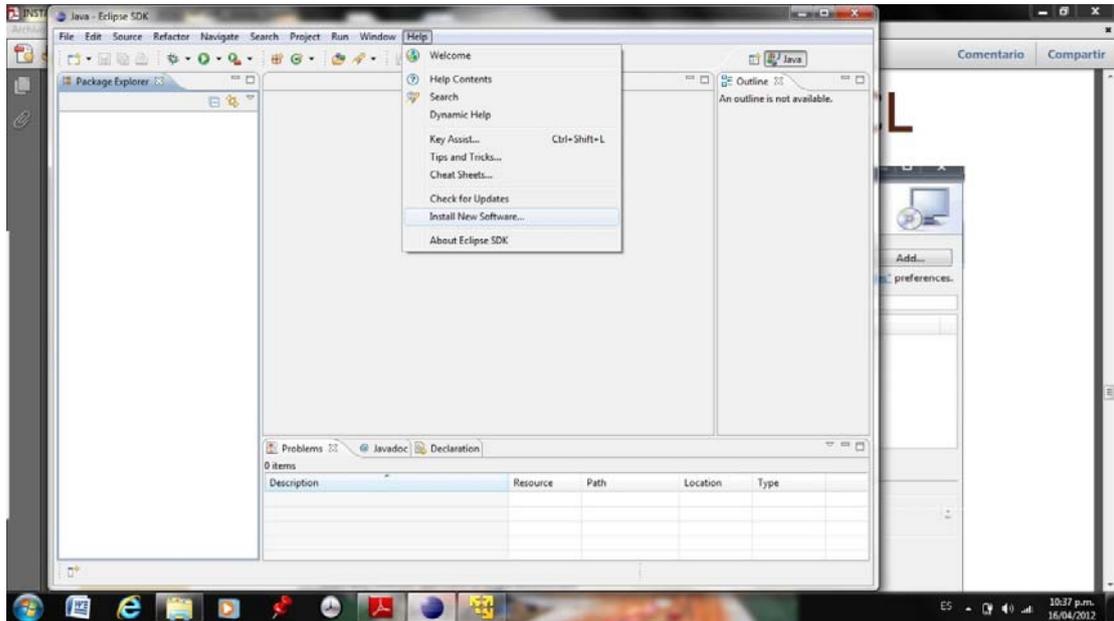


Figura III.39. Instalación nuevo Software eclipse-SDK-3.1.7

Continuamos con la configuración

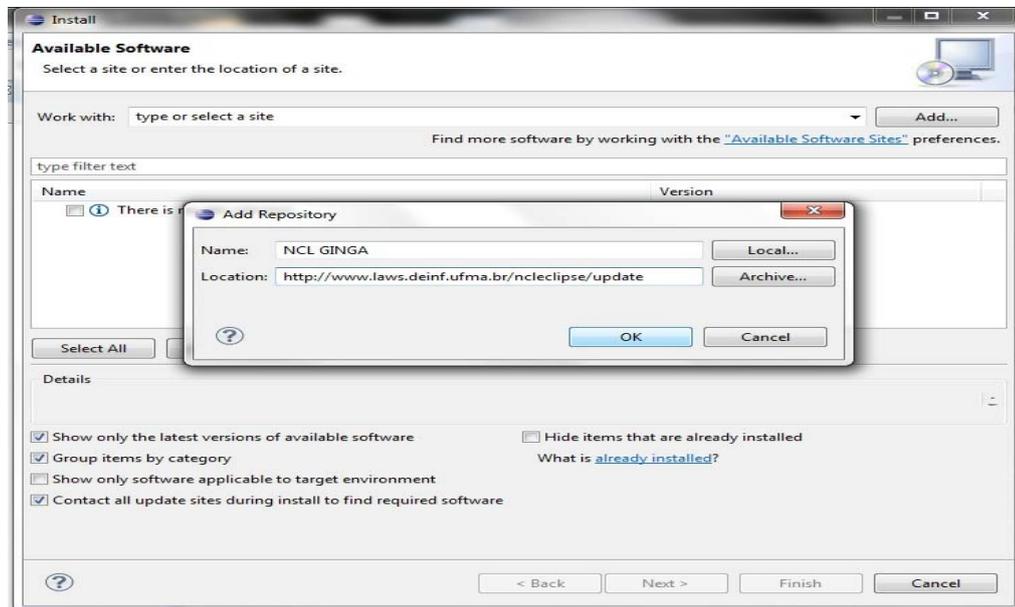


Figura III.40. Localización de eclipse-SDK-3.1.7

E instalamos NCL Eclipse (Figura III.41.)

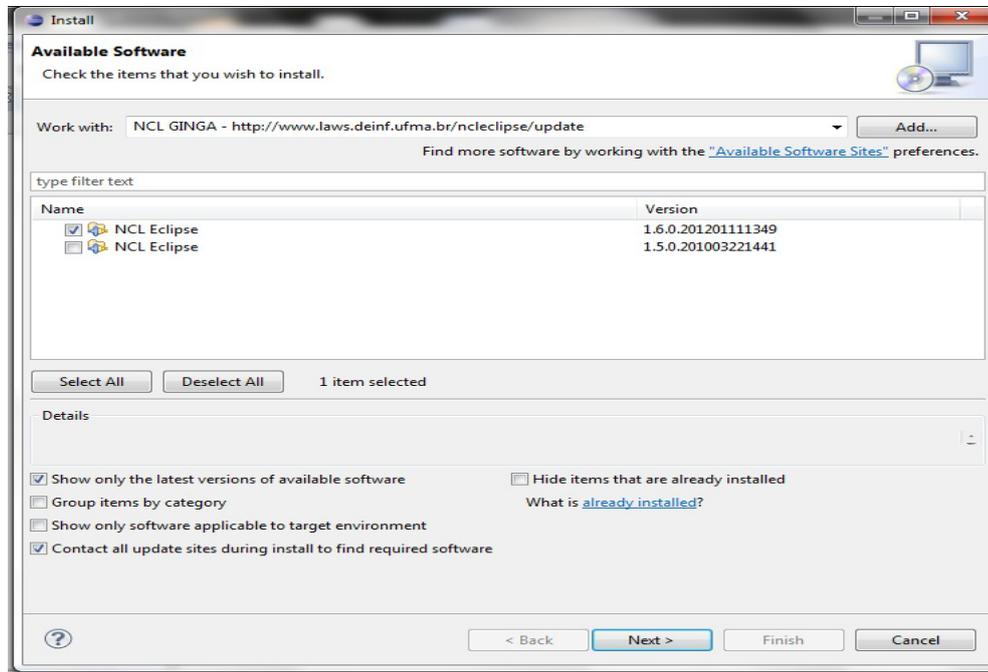


Figura III.41. Pantalla NCL eclipse 1.6.0

Aceptamos los términos de la licencia.

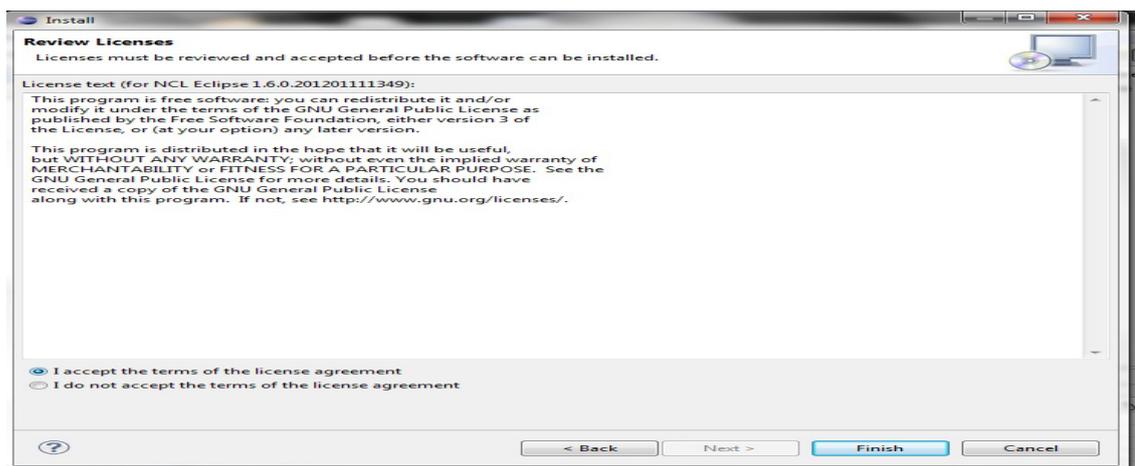


Figura III.42 Licencia NCL eclipse 1.6.0

E inmediatamente se realiza la instalación del software.(Figura III.43.)

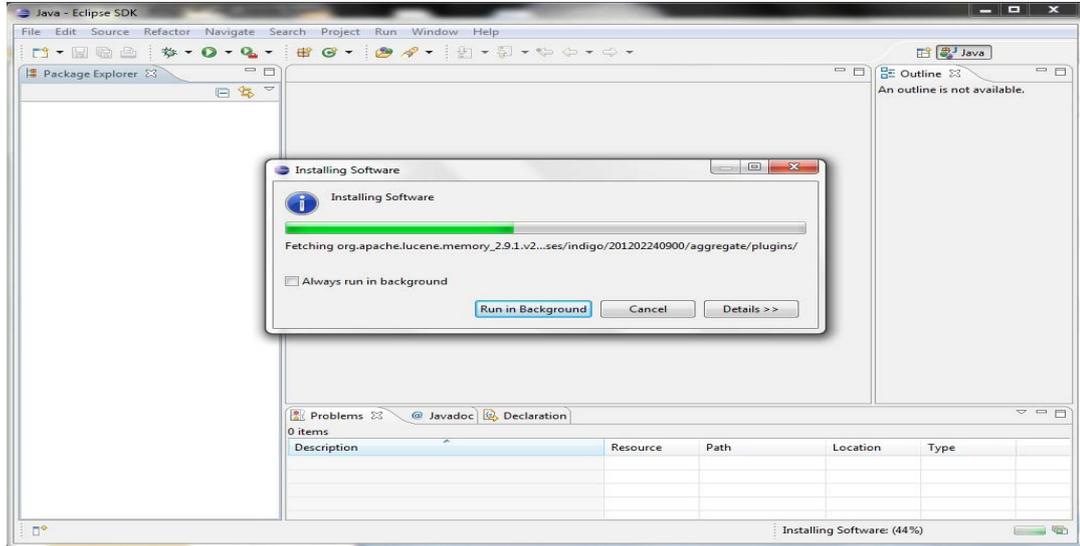


Figura III.43. Instalación software NCL eclipse 1.6.0

En la ventana principal de eclipse , ingresamos a **Window**>Preferences

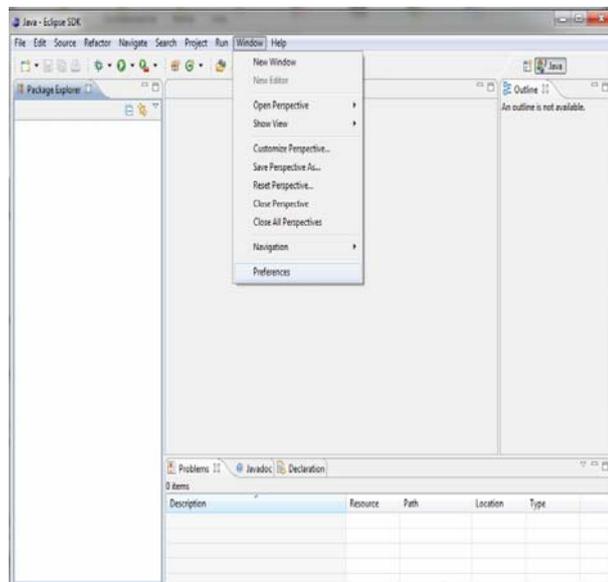


Figura III.44. Configuración de Window

Dentro de Preferences seleccionamos >NCL >Remote Ginga NCL Play

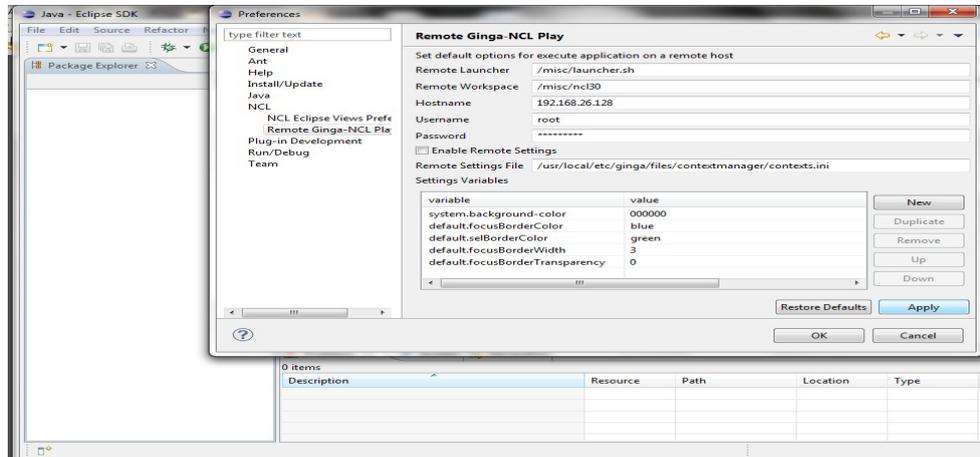


Figura III.45. Configuración de RemoteGinga NCL

Se debe tener muy en cuenta que el hostname que posee la maquina virtual debe coincidir con el hostname que posee eclipse NCL.

Con las configuraciones realizadas el programa se encuentra listo para crear un proyecto ncl.(Figura III.46.)

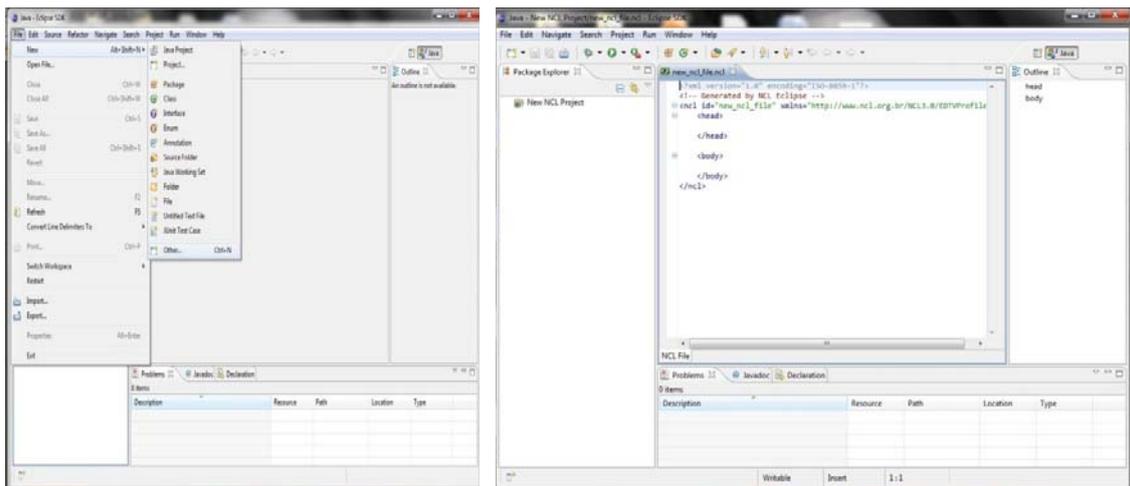


Figura III.46. Creación NCL Project

Finalmente programas la aplicación deseada.(Figura III.47)

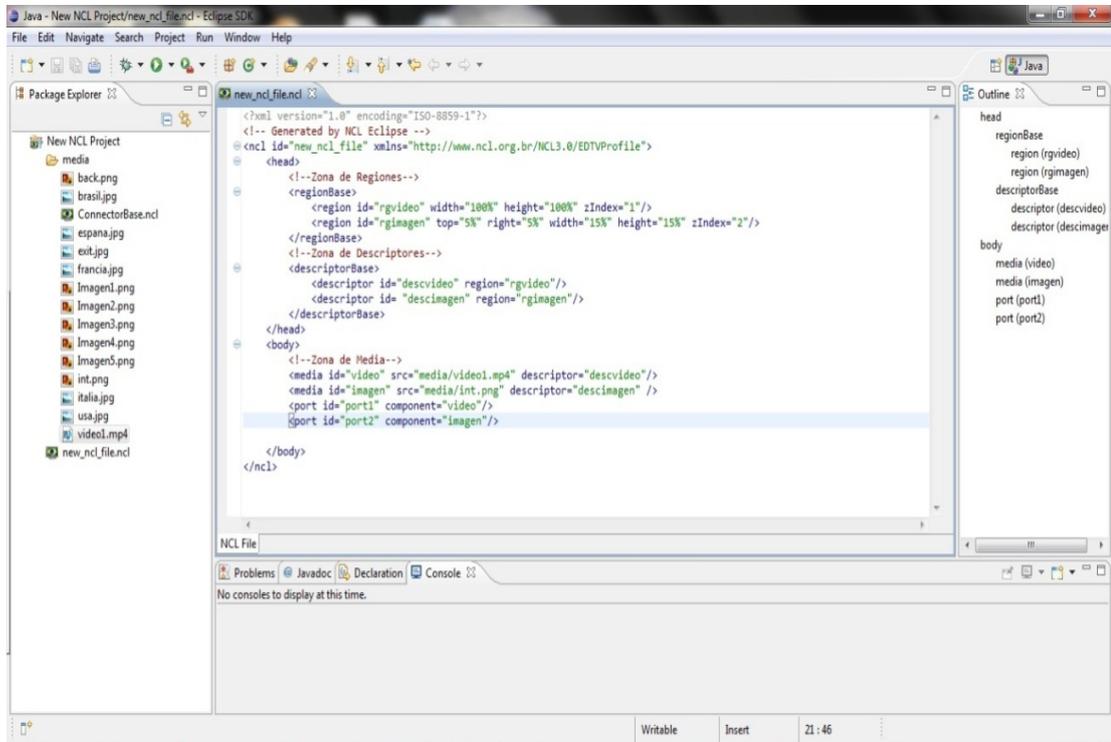


Figura III.47. NCL Project

CAPÍTULO IV

METODOLOGÍA PARA EL DISEÑO DE INTERFACES

4.1 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 IV.48.)

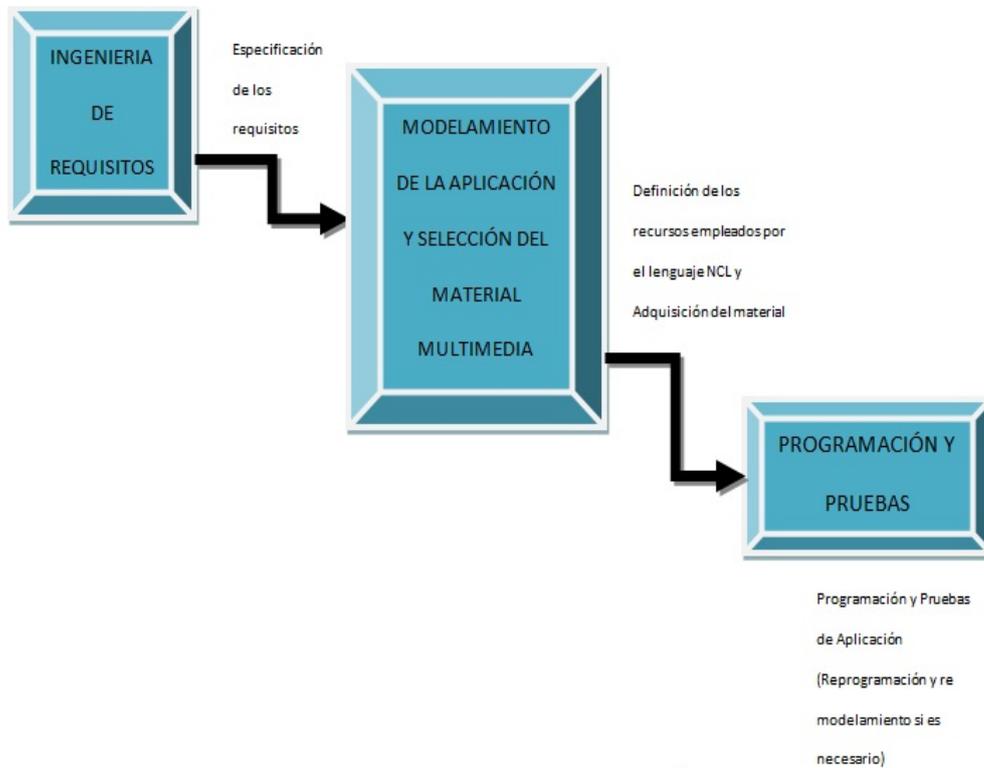


Figura IV.48. Ciclo de vida en cascada de la Aplicación implementada

4.2 APLICACIÓN NCL INFO-ESPOCH

Bajo la necesidad de manipular material multimedia libre de derechos de autor y por ser la Escuela Superior Politécnica de Chimborazo la institución para la cual se desarrolló este proyecto, se implementó la Aplicación Interactiva denominada "INFO-ESPOCH", la que pretende promocionar e informar sobre su visión, misión, autoridades, facultades, mientras el espectador observa su video institucional.

4.3 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 se presiona da la opción "INFO", se desplegará un menú de opciones de interacción; la primera se denominará "VISION MISION Y AUTORIDADES INSTITUCIONALES" y su acceso se hará a través de la interacción con el botón "AMARILLO" del control remoto; la segunda se denominará "FACULTADES" y su acceso se hará a través de la interacción con el botón "VERDE" del control remoto; finalmente la tercera opción se denominará "FACULTADES" y su acceso se hará a través de la interacción con el botón "AMARILLO" del control remoto.

En nuestro caso por ser un Set Top Box Virtual se utilizaran las teclas F1, F2, F3, F4 respectivamente.

4.3 PROGRAMACIÓN Y PRUEBAS DE LA APLICACIÓN

El desarrollo de la Aplicación se realizó utilizando la herramienta de desarrollo NCL llamada Eclipse, y sus pruebas con la ayuda de la herramienta de presentación STB Virtual todas estas de distribución libre.

4.4 IMPLEMENTACIÓN

La implementación de esta aplicación posee un nivel de complejidad alto, y el código llegó a tener más de 500 líneas.

4.5. RESULTADOS

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

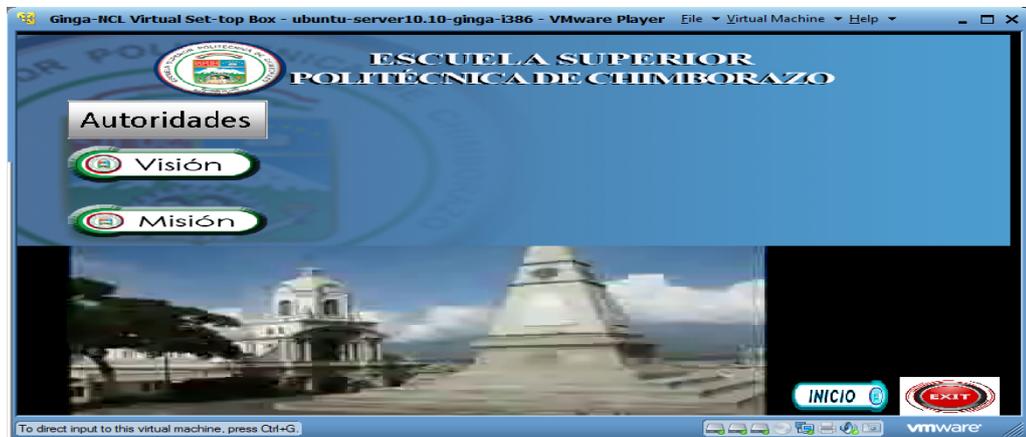


Figura IV.49. Menú Principal de Opciones



Figura IV.50. Menú Principal de Opciones



Figura IV.51. Pantalla principal, después de la selección la opción "AUTORIDADES"



Figura IV.52. Pantalla principal, después de la selección la opción "VISION"



Figura IV.53. Pantalla principal, después de la selección la opción "MISION"



Figura IV.54. Pantalla principal, después de la selección la opción "FACULTADES"



Figura IV.55. Pantalla principal, después de la selección la opción “FIE” del menú “FACULTADES”



Figura IV.56. Pantalla principal, después de la selección la opción “FADE” del menú “FACULTADES”



Figura IV.57. Pantalla principal, después de la selección la opción “C.PECUARIAS” del menú “FACULTADES”



Figura IV.58. Pantalla principal, después de la selección la opción “R.NATURALES” del menú “FACULTADES”



Figura IV.59. Pantalla principal, después de la selección la opción “S. PUBLICA” del menú “FACULTADES”



Figura IV.60. Pantalla principal, después de la selección la opción “CIENCIAS” del menú “FACULTADES”



Figura IV.61 Pantalla principal, después de la selección la opción “MECANICA” del menú “FACULTADES”

CONCLUSIONES

1. La TDT abre las puertas de la interactividad y permite a los usuarios trabajar en un entorno más fácil y de rápida accesibilidad.
2. Con la llegada de la TDT se logra realizar tareas que jamás se imaginó alcanzar con la TV analógica, dentro de las cuales podemos citar: E- learning, Telemedicina, Portales de Compras, Entretenimiento, entre otros.
3. Se crean nuevas fuentes de trabajo y con esto crece el desarrollo del país en lo social, tecnológico, y económico.
4. El lenguaje NCL nos permite simular la TDT , a la par de Eclipse que es la herramienta donde programamos nuestro entorno interactivo.
5. Gracias a la ayuda de la VMWare Station virtualizamos lo que en la realidad sería los equipos utilizados para tener una señal de TDT entre los cuales se encuentra el Set Top Box y el mando (Control Remoto).
6. Mediante esta implementación virtual hemos podido constatar que la interactividad es una realidad en TV Digital.
7. Al realizar la aplicación para la ESPOCH utilizando middleware Ginga y Eclipse NCL hemos cumplido con el objetivo de nuestro estudio ya que se logró implementar el portal INFO-ESPOCH el cual brinda la oportunidad de conocer a nuestras autoridades, la misión, visión y cada una de las facultades que conforman nuestra Politécnica.

RECOMENDACIONES

1. Debido a la complejidad que existe hasta llegar a familiarizarse con el Ginga NCL se recomienda antes de iniciar la programación para la interactividad, tener en cuenta todos los conceptos básicos del mismo. Además de la estructura de un programa NCL.
2. 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.
3. 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.
4. Por ser Ginga de código abierto se recomienda aportar con las investigaciones y aplicaciones realizadas con Ginga, ya que de esta manera construiremos una base sólida para que esta nueva tecnología tenga el impacto que se espera en nuestro país y en Latinoamérica.

RESUMEN

El Estudio Metodológico para el diseño de interfaces entre el Pc y el usuario utilizando ISDB-Tb y middleware Ginga fue desarrollado pensando en la necesidad de brindar a la comunidad politécnica un entorno de interactividad mediante el cual pueda acceder de forma rápida y didáctica a diferente información de nuestra institución.

En el desarrollo investigativo de ese sistema se utilizó el entorno integrado (IDE) Eclipse para la creación de archivos ncl, igualmente que el middleware Ginga-NCL el mismo que provee una infraestructura de presentación para aplicaciones interactivas de tipo declarativas escritas en el lenguaje NCL (Nested Context Lenguaje). Además se utilizó para la implementación de esta aplicación un computador con los siguientes requerimientos de HW : RAM de 2,5 GB, Procesador Dual-core 1.73 GHz , disco duro de 120 GB. En el cual se instaló los softwares : VMWare Player 3.1 , maquina virtual de Ginga y los asistentes para cargar los archivos NCL como es ECLIPSE 3.2.

Como resultado de este proyecto se logró la obtención del primer programa interactivo de la ESPOCH, denominado INFO-ESPOCH . Además a través de este proyecto se logró posesionar a la ESPOCH como precursor en el desarrollo de la TV interactiva. Por lo cual se recomienda, realizar nuevas investigaciones y aplicaciones, que ayuden a la ESPOCH a ser un miembro activo de la comunidad latinoamericana en este campo, ya que de esta manera construiremos una base solida para que esta nueva tecnología tenga el impacto que se espera en nuestro país y en Latinoamérica.

SUMMARY

The Study Methodology for the design of interfaces between the PC and the user using the standard ISDB-Tb and Middleware Ginga was developed with the need to provide the community polytechnic environment interaction whereby you can access quickly and didactic to different information from our institution.

In the research development of this system was used integrated development environment (IDE) Eclipse, to create ncl files, also the middleware Ginga-NCL, it provides an infrastructure for interactive presentation of declarative type, written in language NCL (Nested Context language). Also used for the implementation of this application a computer with the following hardware requirements: 2.5 GB RAM, Dual-core Processor 1.73 GHz, 120 GB hard disk.

In which you installed the software virtual machine image of Ginga and attendees to upload files NCL as eclipse.exe.

As a result of this project was achieved obtaining the first interactive ESPOCH, called INFO-ESPOCH. Also through this project was achieved ESPOCH Place the positive to the forerunner in the development of interactive TV. Whether as recommended further research and applications that help ESPOCH to be an active member of the Latin American community in this field, because in this manner build a solid foundation for this new technology has the expected impact in our country and Latin America.

ANEXO

ANEXO 1.CODIGO DE LA IMPLEMENTACIÓN EN Eclipse NCL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="kariprueba" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>

    <!--TIPOS DE TRANSCISIONES -->
    <transitionBase>
        <!--duracion de la transcision comando dur-->
        <transition id="transima1" type="barWipe" dur="2s"/>
        <transition id="transima2" type="irisWipe" dur="3s"/>
        <transition id="transima3" type="clockWipe" dur="4s"/>
        <transition id="transima4" type="snackWipe" dur="1s"/>
        <transition id="transima5" type="clockWipe" dur="6s"/>
    </transitionBase>
    <!--Zona de Regiones-->
    <regionBase>
        <region id="rgvideo" width="100%" height="100%" zIndex="1"/>
        <region id="rgimagen" top="5%" right="5%" width="15%"
            height="10%" zIndex="1"/>
    <!--CREO LAS REGIONES Y VOY POSICIONANDO LOS BOTONES CON SUS
    DIMENSIONES DE ALTO Y ANCHO FACULTADES-->
        <region id="rg13" top="1%" right="1%" width="99%" height="55%"
            zIndex="2"/>
        <region id="rg9" top="2%" right="0.5%" width="20%" height="10%"
            zIndex="2"/>
        <region id="rg8" top="1%" right="48%" width="80%" height="55%"
            zIndex="1"/>
```

```
<region id="rg1" top="10%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg2" top="20%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg3" top="30%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg4" top="38%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg5" top="49%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg6" top="60%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg7" top="71%" right="0.5%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg10" top="18%" right="75%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg11" top="30%" right="75%" width="20%"  
height="10%" zIndex="3"/>
```

```
<region id="rg12" top="45%" right="75%" width="20%"  
height="10%" zIndex="3"/>
```

```
<!--CREO LA REGION DE LAS ESCUELAS-->
```

```
<region id="faculfie1" height="39%" width="50%" left="20%"  
top="18%" zIndex="2"/>
```

```
<region id="faculfade2" height="39%" width="60%" left="20%"  
top="18%" zIndex="2"/>
```

```
<region id="faculcpecuarias3" height="39%" width="50%"  
left="20%" top="18%" zIndex="2"/>
```

```
<region id="faculrnaturales4" height="39%" width="50%"
left="20%" top="18%" zIndex="2"/>
<region id="faculspublica5" height="39%" width="50%" left="20%"
top="18%" zIndex="2"/>
<region id="faculciencias6" height="39%" width="50%" left="20%"
top="18%" zIndex="2"/>
<region id="faculmecanica7" height="39%" width="50%" left="20%"
top="18%" zIndex="2"/>
<region id="fotoautoridades" height="39%" width="60%"
left="30%" top="18%" zIndex="2"/>
<region id="fotovision" height="39%" width="70%" left="30%"
top="18%" zIndex="2"/>
<region id="fotomision" height="39%" width="70%" left="30%"
top="18%" zIndex="2"/>
```

```
<!--CREO REGIONES DE LOS BOTONES SALIR Y REGRESAR-->
```

```
<region id="rgoculta" height="10%" width="10%" left="77%"
top="90%" zIndex="2"/>
<region id="rgsalir" height="10%" width="10%" left="88%"
top="90%" zIndex="2"/>
```

```
</regionBase>
```

```
<!--Zona de Descriptores-->
```

```
<descriptorBase>
```

```
<descriptor id="descvideo" region="rgvideo"/>
```

```
<!--tiempo para que aparezca la imagen comando explicitDur-->
```

```
<descriptor id="descimagen" region="rgimagen">
```

```
<!--LA TRANSPARENCIA SE CREA DENTRO DEL DESCRIPTOR SIEMPRE-->
```

```
<descriptorParam name="transparency" value="80%"/>
```

```
<!--CREA LOS BOTONES Y LA SECUENCIA DE LAS FLECHAS -->
</descriptor>
<descriptor id="descmen1" region="rg1" focusIndex="1"
moveDown="2" moveUp="7" focusBorderColor="red"/>
<descriptor id="descmen2" region="rg2" focusIndex="2"
moveDown="3" moveUp="1" focusBorderColor="yellow"/>
<descriptor id="descmen3" region="rg3" focusIndex="3"
moveDown="4" moveUp="2" focusBorderColor="green"/>
<descriptor id="descmen4" region="rg4" focusIndex="4"
moveDown="5" moveUp="3" focusBorderColor="aqua"/>
<descriptor id="descmen5" region="rg5" focusIndex="5"
moveDown="6" moveUp="4" focusBorderColor="fuchsia"/>
<descriptor id="descmen6" region="rg6" focusIndex="6"
moveDown="7" moveUp="5" focusBorderColor="maroon"/>
<descriptor id="descmen7" region="rg7" focusIndex="7"
moveDown="1" moveUp="6" focusBorderColor="olive"/>
<descriptor id="descmen10" region="rg10" focusIndex="10"
moveDown="11" moveUp="12" focusBorderColor="fuchsia"/>
<descriptor id="descmen11" region="rg11" focusIndex="11"
moveDown="12" moveUp="10" focusBorderColor="maroon"/>
<descriptor id="descmen12" region="rg12" focusIndex="12"
moveDown="10" moveUp="11" focusBorderColor="olive"/>
<descriptor id="descmen8" region="rg8" />
<descriptor id="descmen9" region="rg9" />
<descriptor id="descmen13" region="rg13" />

<descriptor id="desfac1" region="faculfie1" explicitDur="0:00:02"
transIn="transima2" transOut="transima5"/>
```

```
<descriptor id="desfac2" region="faculfade2" explicitDur="0:00:01"
transIn="transima2" transOut="transima5"/>
    <descriptor id="desfac3" region="faculcpecuarias3"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
        <descriptor id="desfac4" region="faculrnaturales4"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
            <descriptor id="desfac5" region="faculpublica5"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                <descriptor id="desfac6" region="faculciencias6"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                    <descriptor id="desfac7" region="faculmecanica7"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                        <descriptor id="descaut8" region="fotoautoridades"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                            <descriptor id="desvision9" region="fotovision"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                                <descriptor id="desmision" region="fotomision"
explicitDur="0:00:01" transIn="transima2" transOut="transima5"/>
                                    <descriptor id="desrgo" region="rgoculta"/>
                                        <descriptor id="desrgsa" region="rgsalir"/>
                                            </descriptorBase>
<!-- ZONA DE CONECTORES-->
<connectorBase>
    <!--Creo mi base de conectores -->
    <importBase documentURI="ConnectorBase.ncl" alias="conector"/>
    <!--Importo la base que ya esta hecha-->
</connectorBase>
```

```
</head>
<body>
  <!--Zona de Media-->
  <media id="video" src="media/ESPOCH-_Riobamba.mp4"
  descriptor="descvideo">
    <area id="area1" begin="10s"/>
    <property name="bounds"/>
    <!--creo una area -->
  </media>
  <media id="imagen" src="media/escudo.gif" descriptor="descimagen">
    <property name="explicitDur" value="20s"/>
  </media>
  <media id="oculta" src="media/inicio.png" descriptor="desrgo"/>
  <media id="exit" src="media/exit.jpg" descriptor="desrgsa"/>
  <!--contextos-->
    <!--menu1-->
  <context id="menu1">
    <port id="p9" component="op9"/>
    <port id="p8" component="op8"/>
    <port id="p1" component="op1"/>
    <port id="p2" component="op2"/>
    <port id="p3" component="op3"/>
    <port id="p4" component="op4"/>
    <port id="p5" component="op5"/>
    <port id="p6" component="op6"/>
    <port id="p7" component="op7"/>
```

```
<media id="op9" src="media/facultades.png"
descriptor="descmen9"/>
<media id="op8" src="media/fondoo.png" descriptor="descmen8"/>
<media id="op1" src="media/fie.png" descriptor="descmen1"/>
<media id="op2" src="media/fade.png" descriptor="descmen2"/>
<media id="op3" src="media/pecuarias.png"
descriptor="descmen3"/>
<media id="op4" src="media/rnaturales.png"
descriptor="descmen4"/>
<media id="op5" src="media/saludpublica.png"
descriptor="descmen5"/>
<media id="op6" src="media/ciencias.png"
descriptor="descmen6"/>
<media id="op7" src="media/mecanica.png"
descriptor="descmen7"/>
```

```
<!--          FACULTADES -->
<media id="ifie" src="media/fietexto.jpg" descriptor="desfac1"/>
<media id="ifade" src="media/fadetextocopy.jpg"
descriptor="desfac2"/>
<media id="ipecuarias" src="media/pecuariastextocopy.jpg"
descriptor="desfac3"/>
<media id="irnaturales" src="media/rnaturalestextocopy.jpg"
descriptor="desfac4"/>
<media id="ispublica" src="media/saludpublicatexto.jpg"
descriptor="desfac5"/>
<media id="iciencias" src="media/cienciastextocopy.jpg"
descriptor="desfac6"/>
```

```
<media id="imecanica" src="media/mecanicatextocopy.jpg"
descriptor="desfac7"/>
```

```
<link xconnector="conector#onKeySelectionStart">
  <bind role="onSelection" component="op1">
    <bindParam name="keyCode" value="VK_ENTER"/>
  </bind>
  <bind role="start" component="ifie"/>
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
  <bind role="onSelection" component="op2">
    <bindParam name="keyCode" value="VK_ENTER"/>
  </bind>
  <bind role="start" component="ifade"/>
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
  <bind role="onSelection" component="op3">
    <bindParam name="keyCode" value="VK_ENTER"/>
  </bind>
  <bind role="start" component="ipecuarias"/>
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
  <bind role="onSelection" component="op4">
    <bindParam name="keyCode" value="VK_ENTER"/>
  </bind>
  <bind role="start" component="irnaturales"/>
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
```

```
<bind role="onSelection" component="op5">
    <bindParam name="keyCode" value="VK_ENTER"/>
</bind>
<bind role="start" component="ispublica"/>
</link>
<link xconnector="conector#onKeySelectionStart">
    <bind role="onSelection" component="op6"><bindParam
name="keyCode" value="VK_ENTER"/>
    </bind>
    <bind role="start" component="iciencias" />
</link>
<link xconnector="conector#onKeySelectionStart">
    <bind role="onSelection" component="op7"><bindParam
name="keyCode" value="VK_ENTER"/>
    </bind>
    <bind role="start" component="imecanica" />

</link>

<link xconnector="conector#onKeySelectionStartNStopN">
    <bind role="onSelection" component="op1"/>
    <bind role="start" component="ifie"/>
    <bind role="stop" component="ifade"/>
    <bind role="stop" component="ipecuarias"/>
    <bind role="stop" component="irnaturales"/>
    <bind role="stop" component="ispublica"/>
    <bind role="stop" component="iciencias"/>
    <bind role="stop" component="imecanica"/>
```

</link>

<link xconnector="conector#onKeySelectionStartNStopN">

<bind role="onSelection" component="op2"/>

<bind role="start" component="ifade"/>

<bind role="stop" component="ipecuarias"/>

<bind role="stop" component="irnaturales"/>

<bind role="stop" component="ispublica"/>

<bind role="stop" component="iciencias"/>

<bind role="stop" component="imecanica"/>

<bind role="stop" component="ifie"/>

</link>

<link xconnector="conector#onKeySelectionStartNStopN">

<bind role="onSelection" component="op3"/>

<bind role="start" component="ipecuarias"/>

<bind role="stop" component="irnaturales"/>

<bind role="stop" component="ispublica"/>

<bind role="stop" component="iciencias"/>

<bind role="stop" component="imecanica"/>

<bind role="stop" component="ifie"/>

<bind role="stop" component="ifade"/>

</link>

<link xconnector="conector#onKeySelectionStartNStopN">

<bind role="onSelection" component="op4"/>

<bind role="start" component="irnaturales"/>

<bind role="stop" component="ispublica"/>

```
<bind role="stop" component="iciencias"/>  
<bind role="stop" component="imecanica"/>  
<bind role="stop" component="ifie"/>  
<bind role="stop" component="ifade"/>  
<bind role="stop" component="ipecuarias"/>
```

```
</link>
```

```
<link xconnector="conector#onKeySelectionStartNStopN">  
  <bind role="onSelection" component="op5"/>  
  <bind role="start" component="ispublica"/>  
  <bind role="stop" component="iciencias"/>  
  <bind role="stop" component="imecanica"/>  
  <bind role="stop" component="ifie"/>  
  <bind role="stop" component="ifade"/>  
  <bind role="stop" component="ipecuarias"/>  
  <bind role="stop" component="irnaturales"/>
```

```
</link>
```

```
<link xconnector="conector#onKeySelectionStartNStopN">  
  <bind role="onSelection" component="op6"/>  
  <bind role="start" component="iciencias"/>  
  <bind role="stop" component="imecanica"/>  
  <bind role="stop" component="ifie"/>  
  <bind role="stop" component="ifade"/>  
  <bind role="stop" component="ipecuarias"/>  
  <bind role="stop" component="irnaturales"/>  
  <bind role="stop" component="ispublica"/>
```

```
</link>
```

```
<link xconnector="conector#onKeySelectionStartNStopN">
```

```
  <bind role="onSelection" component="op7"/>
```

```
  <bind role="start" component="imecanica"/>
```

```
  <bind role="stop" component="ifie"/>
```

```
  <bind role="stop" component="ifade"/>
```

```
  <bind role="stop" component="ipecuarias"/>
```

```
  <bind role="stop" component="irnaturales"/>
```

```
  <bind role="stop" component="ispublica"/>
```

```
  <bind role="stop" component="iciencias"/>
```

```
</link>
```

```
</context>
```

```
<!--contextos-->
```

```
  <!--menu2-->
```

```
<context id="menu2">
```

```
  <port id="p10" component="op10"/>
```

```
  <port id="p11" component="op11"/>
```

```
  <port id="p12" component="op12"/>
```

```
  <port id="p13" component="op13"/>
```

```
  <media id="op13" src="media/fondoo.png"  
  descriptor="descmen13"/>
```

```
  <media id="op10" src="media/autoridadessss.png"  
  descriptor="descmen10"/>
```

```
  <media id="op11" src="media/vision.png"  
  descriptor="descmen11"/>
```

```
  <media id="op12" src="media/mision.png"  
  descriptor="descmen12"/>
```

```
<!--autoridades -->
```

```
<media id="iautoridades" src="media/autoridades.png"  
descriptor="descout8"/>
```

```
<media id="ivision" src="media/visiontextocopy.jpg"  
descriptor="desvision9"/>
```

```
<media id="imision" src="media/misiontextocopy.jpg"  
descriptor="desmision"/>
```

```
<link xconnector="conector#onKeySelectionStart">
```

```
<bind role="onSelection" component="op10"><bindParam  
name="keyCode" value="VK_ENTER"/>
```

```
</bind>
```

```
<bind role="start" component="iautoridades" />
```

```
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
```

```
<bind role="onSelection" component="op11"><bindParam  
name="keyCode" value="VK_ENTER"/>
```

```
</bind>
```

```
<bind role="start" component="ivision" />
```

```
</link>
```

```
<link xconnector="conector#onKeySelectionStart">
```

```
<bind role="onSelection" component="op12"><bindParam  
name="keyCode" value="VK_ENTER"/>
```

```
</bind>
```

```
<bind role="start" component="imision" />

</link>
<link xconnector="conector#onKeySelectionStartNStopN">
  <bind role="onSelection" component="op10"/>
  <bind role="start" component="iautoridades"/>
  <bind role="stop" component="ivision"/>
  <bind role="stop" component="imision"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
  <bind role="onSelection" component="op11"/>
  <bind role="stop" component="iautoridades"/>
  <bind role="start" component="ivision"/>
  <bind role="stop" component="imision"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
  <bind role="onSelection" component="op12"/>
  <bind role="stop" component="iautoridades"/>
  <bind role="stop" component="ivision"/>
  <bind role="start" component="imision"/>
</link>
</context>

<!--ZONA DE PUERTAS-->
<port id="port1" component="video"/>
<!--esta es la unica puerta de entrada que en este caso es mi video-->
<link xconnector="conector#onBeginStart" id="inicia el video">
  <!--Inicializo mi conector-->
```

```
<bind role="onBegin" component="video"/>
<bind role="start" component="imagen"/>
</link>
<link xconnector="conector#onBeginStart">
  <bind role="onBegin" component="video" interface="area1"/>
  <bind role="start" component="imagen"/>
</link>
<!-- BOTON VERDE (F2)-->
<link xconnector="conector#onKeySelectionStartNSetNStopN">
  <bind role="set" component="video" interface="bounds">
    <bindParam name="var" value="5%,55%,70%,60%"/>
  </bind>
  <bind role="onSelection" component="imagen"/>
  <linkParam name="keyCode" value="GREEN"/>
  <bind role="start" component="menu2"/>
  <bind role="start" component="exit"/>
  <bind role="start" component="oculta"/>
  <bind role="stop" component="imagen"/>

  </link>
<link xconnector="conector#onBeginStart">
  <bind role="onBegin" component="video" interface="area1"/>
  <bind role="start" component="imagen"/>
</link>
<!-- BOTON AMARILLO (F3)-->
<link xconnector="conector#onKeySelectionStartNSetNStopN">
  <bind role="set" component="video" interface="bounds">
    <bindParam name="var" value="5%,55%,70%,60%"/>
  </bind>
  <bind role="onSelection" component="imagen"/>
  <linkParam name="keyCode" value="YELLOW"/>
  <bind role="start" component="menu2"/>
  <bind role="start" component="exit"/>
  <bind role="start" component="oculta"/>
  <bind role="stop" component="imagen"/>

  </link>
```

```
</bind>
<bind role="onSelection" component="imagen"/>
<linkParam name="keyCode" value="YELLOW"/>
<bind role="start" component="menu1"/>
<bind role="start" component="exit"/>
<bind role="start" component="oculta"/>
<bind role="stop" component="imagen"/>
```

```
</link>
```

```
<!-- BOTON AZUL (F4)-->
<link xconnector="conector#onKeySelectionStartNSetNStopN">
  <bind role="onSelection" component="oculta">
    <bindParam name="keyCode" value="BLUE"/>
  </bind>
  <bind role="start" component="imagen"/>
  <bind role="stop" component="menu1"/>
  <bind role="stop" component="menu2"/>
  <bind role="stop" component="oculta"/>
  <bind role="stop" component="exit"/>
  <bind role="set" component="video" interface="bounds">
    <bindParam name="var" value="00%,00%,200%,200%"/>
  </bind>
```

```
</link>
```

```
<!-- BOTON ROJO (F1)-->
<link xconnector="conector#onKeySelectionStopN">
  <bind role="onSelection" component="exit">
    <bindParam name="keyCode" value="RED"/>
```

```
</bind>
<bind role="stop" component="imagen"/>
<bind role="stop" component="menu1"/>
<bind role="stop" component="menu2"/>
<bind role="stop" component="video"/>
<bind role="stop" component="oculta"/>
<bind role="stop" component="exit"/>
<bind role="stop" component="video"/>
</link>
<link xconnector="conector#onEndStopN">
  <bind role="onEnd" component="video"/>
  <bind role="stop" component="imagen"/>
  <bind role="stop" component="menu1"/>
  <bind role="stop" component="menu2"/>
  <bind role="stop" component="oculta"/>
  <bind role="stop" component="exit"/>
</link>
</body>
</ncl>
```

BIBLIOGRAFÍA

- 1.- **GOMES SOARES, L.M y JUNQUEIRA BARBOSA, S.**, *Programando en NCL 3.0.*, 2da. ed., s.l., Elsevier., 2010., pp.28-60.
- 2.- **SEN, J.A.**, *Análisis y diseño de sistemas de TDT.*, Traducido del inglés por Paul Lara., 2da. ed., México., McGraw Hill., 2010., pp 547-568.

BIBLIOGRAFÍA DE INTERNET

- 3.- **Blanco González, V.**, *Instalación de un entorno de desarrollo integrado (IDE): Eclipse.*,
<http://www.desarrolloweb.com/articulos/1692.php> ., 2012-06-10
- 4.- **CARVALHO, R. et al.**, *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> .,
2012-03-22.

5.- Eclipse-descarga e instalación.,

<http://www.karmany.net/index.php/programacion-web/36-eclipse/78-eclipse-descarga-e-instalacion> ., 2012-06-10

6.- **GOMES, L., JUNQUEIRA , S.**, Ginga-NCL:El

Ambiente Declarativos del Sistema de TV Digital Brasileira.,
Departamento de Informática., Pontificia Universidad Católica de
Rio de Janeiro., ftp://ftp.telemedia.puc-rio.br/~lfgs/docs/journalpapers/2007_04_jbcs_lfgs.pdf.,2012-01-12

7.- **GOMES, L., JUNQUEIRA, S.**, **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>., 2012-01-17

8.- **HERRERA HERMIDA, H.**,. Instalación y uso de VMWare player.,

<http://www.bujarra.com/instalacion-y-uso-de-vmware-player>.,
2012 -05-25

9.- INFORMACIÓN GENERAL., www.esPOCH.edu.ec., 2012-05-20.

10.- LEMOS DE SOUZA, G., CUNHA, L., COELHO C., 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](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)., 2012-02-26

11.- Página Oficial de Ginga Brasil., [http://www.gingancl.org.br/.](http://www.gingancl.org.br/)

2012-04-10

12.- Página Oficial de Ginga Ecuador., [http://www.gingancl.org.ec/.](http://www.gingancl.org.ec/)

2012-04-10

13.- Página Oficial de de Software Público de Brasil.,

[http://www.softwarepublico.gov.br/.](http://www.softwarepublico.gov.br/), 2012-04-10

14.- Página Oficial del Club NCL de Brasil .,

[http://clube.ncl.org.br/.](http://clube.ncl.org.br/), 2012-04-15.

15.- Página Oficial del Laboratorio Tele Mídia .,

<http://www.telemidia.puc-rio.br.>, 2012-04-15.

16.- SOARES, C. et al., Construyendo programas audiovisuales interactivo

utilizando NCL 3.0., Laboratorio de Sistemas Multimedia

TeleMídia., Pontificia Universidad Católica de Rio de Janeiro,

<http://www.ncl.org.br/documentos/TutorialNCL3.0-2ed.pdf.>,

2012- 02- 20.

17.- VELASQUEZ DIAZ, C., Informe Preliminar: Estado del Arte de

Receptores Set-Top-Box – Aplicaciones.,

<http://es.scribd.com/doc/70106948/23/Emulador-GINGA-NCL-Set->

[Top-Box- Virtual.](#),

