



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO EN:

“INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES”

Título:

**“IMPLEMENTACIÓN DE UN LABORATORIO DE PRUEBAS DE TV
DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE
APLICACIONES INTERACTIVAS PARA LA ESCUELA DE
INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES DE LA
UNACH”**

Autor:

JOSÉ ANDRÉS PEÑAHERRERA OBREGÓN

Director:

ING. Anibal Llanga

RIOBAMBA – ECUADOR

Los miembros del Tribunal de Graduación del proyecto de investigación de título: IMPLEMENTACIÓN DE UN LABORATORIO DE PRUEBAS DE TV DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE APLICACIONES INTERACTIVAS PARA LA ESCUELA DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES DE LA UNACH, presentado por: José Andrés Peñaherrera Obregón y dirigida por: Ing. Aníbal Llanga.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Yessenia Cevallos

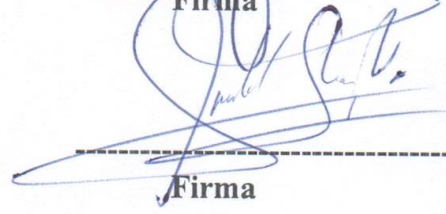
Presidente del Tribunal



Firma

Ing. Aníbal Llanga

Director del Tribunal



Firma

Ing. Daniel Santillán

Miembro del Tribunal



Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a: José Andrés Peñaherrera Obregón e Ing. Aníbal Llanga; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

AGRADECIMIENTO

Agradezco a Dios por concederme la vida, darme capacidad y entendimiento para poder cumplir con mi meta deseada, a mis padres y hermano que me han apoyado en el transcurso de mis estudios con sus palabras de aliento y respaldo, a mis maestros catedráticos quienes, con mística profesional me han entregado sus conocimientos, en especial a mi tutor Ing. Anibal Llanga quien con su vocación y profesionalismo me ha orientado para realizar esta tesis, a los funcionarios responsables de los laboratorios de la facultad por haberme brindado su confianza y predisposición, a mis compañeros y amigos que me apoyaron dandome ánimo para culminar este proyecto.

José Peñaherrera O.

DEDICATORIA

Esta tesis va dedicada a mi familia que me ha brindado aliento positivo para que pueda seguir adelante en mis estudios y triunfe en mi propósito de superación, para poderme desenvolver como un buen profesional al servicio de mi patria y a mi querida Universidad forjadora de profesionales capaces con mística de trabajo y entrega a sus estudiantes.

José Peñaherrera O.

RESUMEN

El presente proyecto tiene como objetivo la implementación de un laboratorio de pruebas de televisión digital terrestre ISDB-Tb, para la emisión de aplicaciones interactivas.

En la emisión se necesita el desarrollo de interactividad, codificación ISDB-Tb, multiplicación y modulación toda instalada y configurada en dos PC que intercambian los datos vía Ethernet.

La codificación de aplicaciones interactivas se lo realiza mediante el uso de NCL Eclipse que es un plug-in de Eclipse para el desarrollo textual de NCL.

La codificación ISDB-Tb se lo efectúa en Linux-Ubuntu donde mediante la instalación de varios software libres como ffmpeg, tovid y OpenCaster se codifica en formato mpeg 2 y se crea el (TS – Transport Stream) que adecua la señal previo a su modulación.

Para la modulación y transmisión se utiliza la tarjeta PCI modelo DTA-2111 que es un modulador Multi-Standard VHF/UHF, esta tarjeta cuenta con un software llamado streamxpress que permite configurar los parámetros de transmisión y su configuración para por ultimo transmitir mediante una antena UHF.

En la recepción del canal de prueba se trabaja con un decodificador digital llamado set top box compatible con el formato ISDB-tb y una televisión con entrada de audio y video.



Lic. Geovanny Armas P.

9 de abril del 2014

SUMMARY

This project aims to implement an ISDB-Tb terrestrial digital TV testing laboratory for the broadcasting of interactive applications.

In the broadcasting of DTT, the interactivity development, the ISDB-Tb coding, the multiplexing and modulation, all of them installed and configured in two PCs to exchange data via Ethernet are carried out.

The coding is done by interactive applications using NCL Eclipse is an Eclipse plugin for developing textual NCL.

The ISDB-Tb coding is carried out in Linux-Ubuntu in which by installing some free software like ffmpeg, tovid and OpenCaster, the mpeg 2 format is coded and the (TS – Transport Stream) is created, it adapts the signal prior to modulation.

For modulation and transmission, the PCI Card, DTA-2111 model is used, this card is a VHF/UHF Multi-Standard modulator and has software called streamxpress to configure transmission parameters and settings to finally transmit via a UHF antenna.

For receiving the test channel it is necessary to work with a digital decoder called set top box which is compatible with the ISDB-Tb format and a television with video and audio input.

Geovanny Armas P.

CENTRO DE IDIOMAS



INDICE GENERAL

RESUMEN	vi
SUMMARY	vii
ACRÓNIMOS	xiv
INTRODUCCIÓN	1
CAPITULO I	2
1. FUNDAMENTACIÓN TEÓRICA	2
1.1. TELEVISIÓN DIGITAL TERRESTRE	2
1.2. TELEVISIÓN DIGITAL TERRESTRE EN ECUADOR	3
1.2.1. ESTACIONES AUTORIZADAS PARA LA OPERACIÓN	3
1.3. DIFERENCIAS ENTRE ISDB-T E ISDB-Tb	4
1.3.1. COMPENSACIÓN DE MOVIMIENTO CON BLOQUES DE TAMAÑO VARIABLE.	5
1.3.2. CARACTERÍSTICAS DEL SISTEMA DE CODIFICACIÓN Y MODULACIÓN	6
1.3.3. FLUJO DE TRANSPORTE (TS) MPEG-2	8
1.3.4. MULTIPLEXACIÓN	8
1.3.4.1. FUNCIONAMIENTO BÁSICO DE UN MULTIPLEXOR	8
1.3.4.2. TABLAS PSI	9
1.3.4.3. TABLAS SI	10
1.3.5. MODIFICACIONES E IMPLEMENTACIONES BRASILEÑAS	11
1.3.6. TV INTERACTIVA – GINGA	11
1.4. REQUISITOS DE CODIFICACIÓN DE MEDIA Y MÉTODOS DE TRANSMISIÓN REFERIDOS EN DOCUMENTOS NCL	12
1.4.1. MÉTODOS DE CODIFICACIÓN Y TRANSMISIÓN DE VIDEO – DATOS DE VIDEO REFERIDOS EN ELEMENTOS <media>	12
1.4.1.1. TRANSMISIÓN DE VIDEO MPEG-4 Y H.264 MPEG-4 AVC	12
1.4.1.2. TRANSMISIÓN EN SECCIONES MPEG-2	12
1.4.2. MÉTODOS DE CODIFICACIÓN Y TRANSMISIÓN DE AUDIO – DATOS DE AUDIO REFERIDOS EN ELEMENTOS <MEDIA>	12
1.4.2.1. TRANSMISIÓN DE AUDIO MPEG-4	12
1.4.2.2. TRANSMISIÓN EN SECCIONES MPEG-2	13
1.4.3. FORMATO TS PARA TRANSMISIÓN DE VIDEO/AUDIO MPEG	13
1.4.3.1. ESPECIFICACIÓN DE LA CODIFICACIÓN DE DATOS	13
1.4.4. TRANSMISIÓN EN SECCIONES MPEG-2	14
1.4.4.1. RESTRICCIONES EN LA REPRODUCCIÓN	14
1.4.5. ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DE IMÁGENES ESTÁTICAS Y GRÁFICOS DE BITMAP REFERIDOS POR ELEMENTOS <MEDIA>	14
1.4.5.1. TRANSMISIÓN DE MPEG-2 I-FRAME, MPEG-4 I-VOP Y H.264 MPEG-4 AVC I-PICTURE	14
1.4.5.2. TRANSMISIÓN EN SECCIONES MPEG-2 PARA REPRODUCCIÓN INTERACTIVA	14

1.4.5.3.	TRANSMISIÓN DE IMAGEN ESTÁTICA JPEG	15
1.4.5.4.	ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DEL BITMAP PNG	15
1.4.5.5.	ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DE LA ANIMACIÓN MNG	15
1.5.	SOFTWARE LINUX	16
1.5.1.	SISTEMA OPERATIVO UBUNTU (LINUX)	16
1.5.1.1.	AYUDA DE COMANDOS UBUNTU	17
1.5.2.	SAMBA (UBUNTU-LINUX)	18
1.5.3.	TOVID (UBUNTU-LINUX)	18
1.5.4.	FFMPEG (UBUNTU-LINUX)	19
1.5.5.	OPENCASER (UBUNTU-LINUX)	20
1.6.	SOFTWARE WINDOWS	21
1.6.1.	MEDIA PLAYER CLASSIC (WINDOWS)	21
1.6.2.	ECLIPSE NCL (WINDOWS)	21
1.6.2.1.	CARACTERÍSTICAS PRINCIPALES:	21
1.6.3.	GINGA EMULADOR (WINDOWS)	23
1.6.4.	STREAMXPRESS SOFTWARE (WINDOWS)	23
1.7.	HARDWARE	24
1.7.1.	ANTENA MICROSTRIP PARA LA BANDA UHF	24
1.7.2.	MODULADOR DTA-2111	25
1.7.2.1.	CARACTERÍSTICAS DEL DTA-2111	25
1.7.2.2.	REQUISITOS DEL PC	26
1.7.2.3.	DETALLES TÉCNICOS	26
1.7.3.	SET TOP BOX	26
1.7.3.1.	DECODIFICADOR DIGITAL (MUNDY HOME)	27
CAPITULO II		28
2.	METODOLOGÍA	28
2.1.	TIPO DE ESTUDIO	28
2.2.	POBLACION Y MUESTRA	28
2.3.	OPERACIONALIZACIÓN DE VARIABLES	29
2.4.	OPERALIZACION DE HIPÓTESIS	30
2.4.1.	ANÁLISIS PREVIO PARA EL PLANTEAMIENTO HIPÓTESIS	32
2.4.2.	PLANTEAMIENTO DE HIPÓTESIS	34
2.4.3.	VALIDACION CHI CUADRADO	34
2.5.	PROCEDIMIENTOS	37
2.5.1.	ETAPA DE GENERACIÓN Y TRANSMISIÓN BTS EN FORMATO ISDB-TB.	37
2.5.1.1.	PC B DESARROLLO DE APLICACIONES GINGA NCL Y LUA Y PRUEBAS DE LA BTS.	38
2.5.1.2.	GESTION EN PC A PARA EL SERVIDOR LINUX	39
a.	INSTALACION DE UBUNTU LINUX	39
b.	CONFIGURACION DE LA RED EN UBUNTU	39
c.	USAR APT-GET Y EL RESTO DE SISTEMA POR PROXY	41
d.	INSTALACION DE OPENCASER2.4	42

e. INSTALACIÓN DE COMPLEMENTOS LINUX PARA GENERACIÓN ISDB-Tb	43
f. SCRIPT PARA GENERAR LAS TABLAS PSI/SI DEL TS EN FORMATO ISDB-TB	44
g. CODIFICACIÓN (ES), PAQUETIZACIÓN (PES), Y FORMACIÓN DE PAQUETES TS.	44
CODIFICACION (ES).	44
h. MULTIPLEXACIÓN MPEG2 DEL TRANSPORT STREAM CON UN SERVICIO PARA INTERACTIVIDAD	49
i. COMPARTIR ARCHIVOS VIA ETHERNET EN UBUNTU	51
2.5.1.3. COMPARTIR EL ARCHIVO NCL CREADO EN ECLIPSE DESDE PC B EN WINDOWS	52
a. RECIEN INSTALADO ECLIPSE	52
b. DESPUES DE INSTALADO ECLIPSE	52
2.5.1.4. INSTALACIÓN DEL SOFTWARE DTC- 300 -SP	53
2.5.1.5. AJUSTES DE MODULACIÓN ISDB -T	54
2.5.2. ETAPA DE RECEPCIÓN DE UN SERVICIO DE TV DIGITAL CON INTERACTIVIDAD.	64
2.5.2.1. ANTENAS UHF	64
2.5.2.2. EL CONECTOR UTILIZADO PARA ESTA ANTENA	66
2.5.2.3. RECEPCION EN EL SET-TOP BOX	66
CAPITULO III	67
3. RESULTADOS	67
3.1. PRUEBAS DE ARCHIVOS TS Y DE LA BTS MULTIPLEXADA ANTES DE SER MODULADA.	67
3.2. PRUEBAS DE APLICACIONES INTERACTIVAS	67
3.3. ANALISIS DE LA SEÑAL DIGITAL MEDIANTE EL ANALIZADOR DE ESPECTROS.	70
3.4. RECEPCION EN EL SET-TOP BOX DE LA SEÑAL DECODIFICADA EN FORMATO ISDB-Tb.	71
4. DISCUSIÓN	72
5. CONCLUSIONES Y RECOMENDACIONES	73
5.1. Conclusiones	73
5.2. Recomendaciones	74
6. PROPUESTA	75
6.1. Título de la propuesta	75
6.2. Introducción	75
6.3. OBJETIVOS	76
6.4. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA	76
6.4.1. GINGA	76

6.4.2. Eclipse – GINGA NCL	77
6.4.3. OPENCASTER	77
6.4.4. Dta-2111	77
6.5. DISEÑO ORGANIZACIONAL.	78
6.6. MONITOREO Y EVALUACIÓN DE LA PROPUESTA	78
7. Bibliografía	79
8. APÉNDICES Y ANEXOS	83

INDICE DE FIGURAS

FIGURA 1: DIFUSIÓN DE LOS SISTEMAS DE TDT MUNDIAL	2
FIGURA 2: RESOLUCIONES ISDB-TB	5
FIGURA 3: SISTEMA DE TRANSMISIÓN ISDB-TB	5
FIGURA 4: PARTICIONES	6
FIGURA 5: EJEMPLO DE SELECCIÓN OPTIMIZADA DE PARTICIÓN DE MACROBLOQUES	6
FIGURA 6: ESQUEMA ESTRUCTURA DEL SEGMENTO DEL ESPECTRO DE ISDB-T	7
FIGURA 7: 13 SEGMENTOS DE ISDB-T	7
FIGURA 8: MODULACIÓN	7
FIGURA 9: CODIFICACIÓN, PAQUETIZACIÓN Y MULTIPLEXACIÓN	8
FIGURA 10: ORDENAMIENTO DEL CUADRO MÚLTIPLEX	8
FIGURA 11: BTS	9
FIGURA 12: INFORMACIÓN PSI ENVIADA COMO INFORMACIÓN ÚTIL EN EL TS.....	9
FIGURA 13: ARQUITECTURA GINGA	11
FIGURA 14: LOGO UBUNTU	16
FIGURA 15: LOGO SAMBA-UBUNTU.....	18
FIGURA 16: LOGO TOVID	18
FIGURA 17: LOGO FFMPEG	19
FIGURA 18: LOGO OPENCASTER	20
FIGURA 19: LOGO MPC	21
FIGURA 20: LOGO ECLIPSE	21
FIGURA 21: SUGERENCIA CÓDIGO CONTEXTUALES	22
FIGURA 22: ERRORES DE MARCADO	22
FIGURA 23: EJECUTAR LA APLICACIÓN.....	22
FIGURA 24: LOGO GINGA	23
FIGURA 25: LOGO STREAMXPRESS	23
FIGURA 26: ANTENA MICROSTRIP.....	24
FIGURA 27: ANTENA MICROSTRIP RECTANGULAR	24
FIGURA 28: MONTAJE	25
FIGURA 29: MULTI-STANDARD VHF/UHF MODULATOR	25
FIGURA 30: DECODIFICADOR DIGITAL	27
FIGURA 31: CURVA CHI-CUADRADO	35
FIGURA 32: DIGRAMA DEL LABORATORIO DE TDT ISDB-TB	37
FIGURA 33: SERVIDORES ISDB-TB.....	38
FIGURA 34: CONFIGURACION UBUNTU 1	40
FIGURA 35: CONFIGURACION UBUNTU 3	41
FIGURA 36: INTERFAZ GRÁFICA UBUNTU	43
FIGURA 37: SALIDA EN EL TERMINAL TOVID-IDVID	45
FIGURA 38: MULTIPLEXACIÓN	49
FIGURA 39: WORKSPACE ECLIPSE.....	52
FIGURA 40: CON LICENCIA.....	54
FIGURA 41: SIN LICENCIA.....	54
FIGURA 42: MODULACION	56
FIGURA 43: PARÁMETROS DE MODULACIÓN	57
FIGURA 44: CAPAS.....	58
FIGURA 45: PARAMETROS DE TRANSMISION	59

FIGURA 46: PARAMETROS DE CAPAS	61
FIGURA 47: ERROR ASIGNACIÓN BITRATE	63
FIGURA 48: TRANSMISIÓN	63
FIGURA 49: TARJETA DEKTEC CON ANTENA UHF 75 Ω	64
FIGURA 50: ANTENA PASIVA DE VARILLA PARA TDT	65
FIGURA 51: ANTENA MICROSTRIP UHF	65
FIGURA 52: ELEMENTOS DEL ACOPLE ANTENA UHF-SET-TOP BOX	66
FIGURA 53: SET-TOP BOX	66
FIGURA 54: CONTROL SET-TOP BOX VIRTUAL	67
FIGURA 55: APLICACIÓN INTERACTIVA GINGA NCL	68
FIGURA 56: APLICACIÓN INTERACTIVA GINGA NCL	68
FIGURA 57: APLICACIÓN INTERACTIVA GINGA NCL	69
FIGURA 58: APLICACIÓN INTERACTIVA GINGA NCL	69
FIGURA 59: APLICACIÓN INTERACTIVA GINGA NCL	70
FIGURA 60: ANALISIS ESPECTRAL	70
FIGURA 61: ESPECTRO DE LA PORTADORA DE LA SEÑAL DIGITAL TRANSMITIDA	71
FIGURA 62: RECEPCIÓN DIGITAL ISDB-TB	71
FIGURA 63: RECEPCIÓN DIGITAL ISDB-TB	72

INDICE DE TABLAS

TABLA 1: ESTACIONES AUTORIZADAS ECUADOR	3
TABLA 2: TABLAS PSI	9
TABLA 3: TABLAS SI	10
TABLA 4: PRINCIPALES ALTERACIONES RELACIONADAS A LA NORMA	11
TABLA 5: AYUDA DE COMANDOS UBUNTU	17
TABLA 6: DETALLES TÉCNICOS	26
TABLA 7: OPERACIONALIZACIÓN DE VARIABLES	29
TABLA 8: OPERALIZACION DE HIPÓTESIS	31
TABLA 9: LISTADO DE APRENDIZAJE DE PERSONAS	32
TABLA 10: PROYECCION DE DESARROLLO DE INTERACTIVIDAD	33
TABLA 11: TABLA DE HIÓTESIS	34
TABLA 12: TABLA DE CALCULOS x2	35
TABLA 13: NUMERO DE MUESTRAS POR TRAMA DE AUDIO	48
TABLA 14: AJUSTES DE MODULACIÓN	55
TABLA 15: COMPONENTES MPEG-2 TS DE CAPA DE ASIGNACIÓN	58
TABLA 16: PARAMETROS ISBB-T 1	59
TABLA 17: PARAMETROS ISBB-T 2	59
TABLA 18: PARÁMETROS DE CAPAS	61
TABLA 19: CENTRO DE MENSAJES	63

ACRÓNIMOS

AC: (Auxiliary Channel). Canal Auxiliar.

AAC: (Advanced Audio Coding). Codificación Avanzada de Audio.

AIT: (Application Information Table). Tabla de Información de Aplicaciones.

AVI: (Audio Video Interleaved). Formato de Audio y Video Entrelazado.

ABNT: (Brazilian Association of Technical Standards). Asociación Brasileira de Normas Técnicas.

ATSC: (Advanced Television System Committee). Comité de Sistema de Televisión Avanzado.

AU: (Access Unit). Unidad de Acceso.

BIT: (Broadcast Information Table). Tabla de información del Radiodifusor.

BST: (Band Segmented Transmission). Transmisión de Banda Segmentada.

BTS: (Broadcast Transport Stream). Flujo de Transporte de Radiodifusión.

CAT: (Conditional Access Table). Tabla de Acceso Condicional.

COFDM: (Coded Orthogonal Frequency Division Multiplexing). Multiplexación por División de Frecuencia Ortogonal Codificada.

CP: (Continuous Pilot). Pilotos Continuos.

DDB: (Download Data Block). Download de Bloque de Datos.

DQPSK: (Differential Quadrature Phase Shift Keying). Modulación por desplazamiento de fase diferencial.

DTMB: (Digital Terrestrial Multimedia Broadcast). Radiodifusión Digital Terrestre de Multimedia.

DSMCC: (Digital Storage Media Command and Control). Medio de Almacenamiento Digital para Control y Comando.

DTS: (Decoding Time Stamp). Marca de Tiempo de Decodificación.

DVB-T: (Digital Video Broadcasting – Terrestrial). Difusión Digital de Video Terrestre.

EDTV: (Enhanced Definition Television). Televisión Digital Mejorada.

ES: (Elementary Stream). Flujo Elemental.

EIT: (Event Information Table). Tabla de Información de Eventos.

EPG: (Electronic Program Guide). Guía de Programación electrónica.

FPS: (Frames per Second). Cuadros por Segundo.

GIF: (Graphic Interchange Format). Gráfico de format intercambiable.

HD: (High-definition). Alta Definición.

HDTV: (High Definition Television). Televisión de Alta Definición.

ISDB-Tb: (International System for Digital Broadcast, Terrestrial, Brazilian Version). Transmisión Digital de Servicios Integrados, versión Brasileña.

LDTV: (Low Definition Television). Televisión de Baja Definición.

LD: (Low-definition). Baja Definición.

LIFIA: (Laboratory Research and Training in Advanced Computing). Laboratorio de Investigación y Formación en Informática Avanzada.

MPEG: (Moving Picture Experts Group). Grupo de Expertos en Imágenes Móviles.

NCL: (Nested Context Language). Lenguaje de Contextos Anidados.

NIT: (Network Information Table). Tabla de Información de Red.

OFDM: (Orthogonal frequency-division multiplexing). Multiplexación por División de Frecuencias Ortogonales.

PAT: (Program Association Table). Tabla de Asociación de Programa.

PCI: (Peripheral Component Interconnect). Interconexión de Componentes Periféricos.

PES: (Packetized Elementary Stream). Flujo Elemental Paquetizado.

PID: (Program Identifier). Identificador de Programa.

PMT: (Program Map Table). Tabla de Mapeamiento de Programas.

PNG: (Portable Network Graphics). Gráficos de red portátiles.

PSI: (Program Specific Information). Tabla de Información de los Programas.

PS: (Program Stream). Flujo de Programa.

QAM: (Quadrature Amplitude Modulation). Modulación de Amplitud en Cuadratura.

QPSK: (Quadrature Phase Shift Keying). Modulación por Desplazamiento de Fase en Cuadratura.

RF: (Radio Frequency). Radio Frecuencia.

SATVD: (Argentine System of Digital Television). Sistema Argentino de Televisión Digital.

SBTVD: (Brazilian Digital Television System). Sistema Brasileiro de Televisión Digital.

SDT: (Service Descriptor Table). Tabla de Descriptor de Servicio.

SDTV: (Standard Definition Television). Televisión de Definición Estándar.

SI: (Service Information). Información de Servicio.

STB: (Set top Box). Receptor decodificador de televisión digital.

TDT: (digital terrestrial television). Televisión Digital Terrestre.

TMCC: (Transmission Multiplexing Configuration Control). Control de Configuración de Transmisión de Multiplexación.

TS: (Transport Stream). Flujo de Transporte.

TOT: (Time Offset Table). Tabla de Cambio de Horario.

UIT: (International Telecommunication Union). Unión Internacional de Telecomunicaciones.

UHF: (Ultra High Frequency). Frecuencias Ultra Altas.

VBV: (Video Buffer Verifier). Verificador del Buffer de Video.

VSB: (Vestigial Side Band). Modulación de banda lateral vestigial.

INTRODUCCIÓN

Ecuador ya tiene en funcionamiento varias estaciones de TDT con la norma ISDB-tb pero aun se encuentran operando en modo de prueba.

La TDT es la transmisión de audio, video y datos, todo codificado en forma digital y comprimido en paquetes de formato digital con un tamaño pequeño y fijo, lo que posibilita la transmisión en definiciones alta, estándar y baja en una sola frecuencia y con un ancho de banda de 6Mhz. La transmisión en alta definición no seria posible sin una compresión de datos ya que esta necesita un ancho de banda uy grande.

La parte mas importante de TDT es el poder transmitir datos (Interactividad) lo que permite interaccion directa con el televidente.

El motivo que incentivó el desarrollo del proyecto radica en la reciente adopción del estándar brasileño de Televisión Digital que será implementado en un futuro próximo en nuestro país, por lo cual es de gran importancia conocer cómo trabaja este estándar y los beneficios que ofrece. Por otro lado, también es necesario analizar los elementos que intervienen tanto en el desarrollo de contenidos interactivos, una de las mayores atracciones de TV Digital, como en la generación del flujo de transporte a transmitir y el equipamiento que se requiere para poder visualizarlos a nivel de usuario.

Además actualmente la Facultad de Ingeniería Electrónica y telecomunicaciones no dispone de un laboratorio de este tipo, que servirá como un complemento para el estudio de Televisión Digital en la materia de Optativa.

CAPITULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. TELEVISIÓN DIGITAL TERRESTRE¹

La TDT cuenta con múltiples estándares, cada uno de estos con características tecnológicas diferentes como consecuencia de parámetros de codificación, paquetización, multiplexación y modulación establecidos de acuerdo a las necesidades de cada región del mundo.

La televisión digital terrestre es la tecnología que posibilita optimizar el espectro radioeléctrico en sus parámetros de compresión de datos que lo componen y en consecuencia el ancho de banda. La codificación digital permite una compresión de la señal que a su vez genera ventajas como el poder transmitir múltiples señales de audio, video y datos en HD, SD y LD con la agrupación de portadoras en un número de segmentos fijo que se establecen en cada norma de televisión digital.

En el mundo la TDT se establece de acuerdo a normas adoptadas por cada región, esto se lo puede apreciar en la Figura 1.

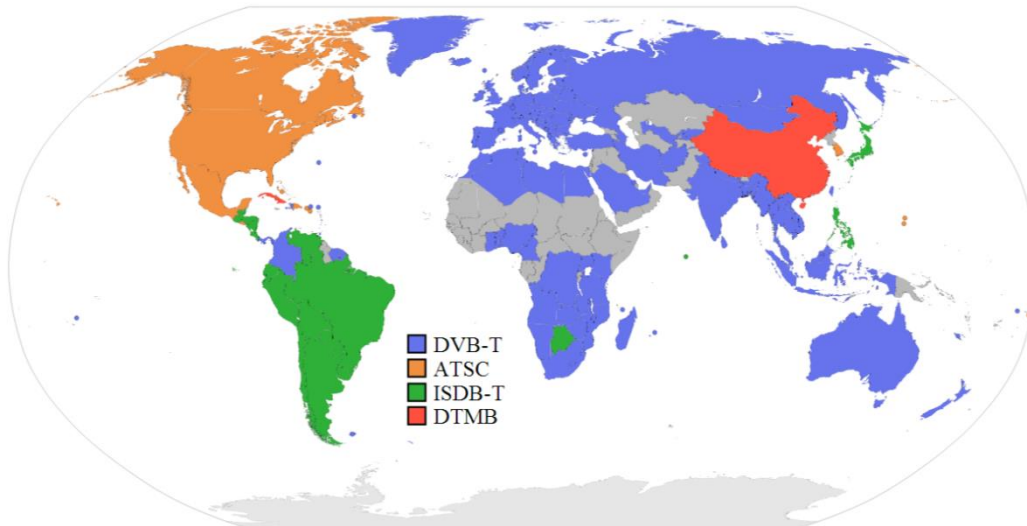


Figura 1: DIFUSIÓN DE LOS SISTEMAS DE TDT MUNDIAL

Fuente: (WIKIPEDIA, Televisión digital terrestre, 2014)

¹ WIKIPEDIA. (3 de Enero de 2014). Televisión digital terrestre. Recuperado el 2014, de <http://es.wikipedia.org/wiki/TDT>

1.2. TELEVISIÓN DIGITAL TERRESTRE EN ECUADOR

En el Ecuador la TDT se encuentra aun en etapa de desarrollo, la norma que se adopto el 26 de marzo del 2010 es ISDB-Tb (Servicio de broadcasting digital integrado - terrestre brasileño). El apagón analógico se efectuara en el transcurso de los años 2016 2017 y 2018.

1.2.1. ESTACIONES AUTORIZADAS PARA LA OPERACIÓN²

Marca	Ciudad de origen	Ciudades de cobertura	Operado por	Al aire	Canal Virtual	Canal Físico	Fecha de solicitud**
Canal UNO ^{4,5}	Guayaquil	Guayaquil, Eloy Alfaro (Durán), Yaguachi Nuevo, Samborondón	RELAD S.A.	Si	12.1 HD / 12.2 SD (en pruebas***)	N/D	18/10/2012
		Quito, Sangolquí	Canal Uno S.A.	No			19/03/2013
Ecuador TV HD ^{6,7}	Quito	Quito, Sangolquí, Machachi	Televisión y Radio de Ecuador E.P.	Si	HD 7.1 / SD 7.2	26 UHF	18/10/2012
		Guayaquil, Eloy Alfaro (Durán), Yaguachi Nuevo, Samborondón		Si	HD 7.1 / SD 7.3	21 UHF	
		Cuenca		No			
Ecuavisa HD ^{9,10}	Guayaquil	Guayaquil, Eloy Alfaro (Durán), Yaguachi Nuevo, Samborondón	Corporación Ecuatoriana de Televisión S.A.	Si	HD 2.1	23 UHF	18/10/2012
		Quito, Tabacundo, Cayambe, Sangolquí	Televisora Nacional Canal 8 C.A.	Si	HD 8.1	36 UHF	
Espol TV ¹¹	Santa Elena	Santa Elena, Salinas, La Libertad	Escuela Politécnica del Litoral	No			25/01/2013
Gama TV ¹²	Quito	Quito, Tabacundo, Cayambe, Sangolquí	Compañía Televisión del Pacífico Teledos S.A.	Si	2.1 HD / 2.2 SD (en pruebas***)	30 UHF	18/10/2012
OK TV - Tevecorp ¹³	Machala	Machala, Paccha, El Gusbo, Pasaje, Piñas, Santa Rosa, Zaruma	Tevecorp S.A.	No			18/10/2012
Oromar HD ¹⁴	Manta	Manta, Portoviejo, Montecristi, Santa Ana de Vuelta Larga, Rocafuerte, Jaramijo	Sistemas Globales de Comunicación HC GLOBAL S.A.	Si	HD 41.1	N/D	18/10/2012
		Quito		N/D***			
		Guayaquil		Si	HD 28.1	N/D	
RTS ^{15,16}	Guayaquil	Guayaquil, Samborondón, Yaguachi Nuevo, Milagro, Eloy Alfaro (Durán)	Telecastro Guayaquil C.A.	Si	4.1 HD / 4.2 SD (en pruebas***)	38 UHF	18/10/2012
		Quito, Cayambe, Sangolquí, Tabacundo	Organización Ecuatoriana de Televisión Ortel S.A.	Si	5.1 HD / 5.2 SD	34 UHF	
RTU ^{17,18,19}	Quito	Quito, Sangolquí	48 UHF ABC	Si	SD 48.1 / SD 48.2 (18.9)	43 UHF	19/03/2013
		Guayaquil, Samborondón, Yaguachi Nuevo, Milagro, Eloy Alfaro (Durán)	Costanera	Si	HD 99.1 (en pruebas***)	N/D	25/01/2013
		Santo Domingo de los Colorados, El Carmen, Pedro Vicente Maldonado	Compañía Radio Hit S.A.	No			18/10/2012
TC HD ²⁰	Guayaquil	Guayaquil, Eloy Alfaro (Durán), Milagro, Samborondón, Yaguachi Nuevo	Cadena Ecuatoriana de TV S.A.	Si	HD 10.1 / SD 10.2	29 UHF	18/10/2012
Teleamazonas	Quito	Quito	Centro de Radio y Televisión Cratel S.A.	Si	HD 4.1 / SD 4.2	32 UHF	18/10/2012
		Guayaquil	Teleamazonas Guayaquil S.A.	Si	HD 5.1 / SD 5.2	N/D	
Telesucosos HD ²¹	Quito	Quito - Distrito Metropolitano, Sangolquí	Compusud C.A.	Si	SD 29.1	41 UHF	18/10/2012
TVS.FHD.RS	Guayaquil	Quito	Televisión Satelital S.A.	Si	HD 25.1	39 UHF	18/10/2012
		Guayaquil		Si	HD 36.1	N/D	
Tevemés ²²	Guayaquil	Guayaquil, Samborondón, Yaguachi Nuevo, Eloy Alfaro (Durán)	Perone S.A.	No			18/10/2012
TV Color ²³	Latacunga	Latacunga, Ambato, Saquisilí, Pujilí, San Miguel, Pillaro, Tisaleo, Cevallos, Quero, Mocha	Caicedo Alvarez Fredy Francisco	No			18/10/2012
Unimax ²⁴	Ambato	Ambato, Latacunga, San Miguel, Tisaleo, Cevallos, Quero, Pelileo, Pillaro	Muessa C.A.	No			18/10/2012

Tabla 1: ESTACIONES AUTORIZADAS ECUADOR
Fuente: (WIKIPEDIA, Televisión Digital Terrestre en Ecuador, 2014)

*Marca usada para las operaciones en la TDT.

**Las operadoras tienen un periodo de hasta 9 meses para iniciar sus transmisiones a partir de la fecha de solicitud.

***Las señales en pruebas pueden estar no siempre disponibles para su sintonización o pueden presentar señal intermitente.

****Oromar HD funciono en modo de pruebas hasta finales de septiembre de 2013 sobre la frecuencia 47 UHF.

² WIKIPEDIA. (7 de Enero de 2014). *Televisión Digital Terrestre en Ecuador*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Televisi%C3%B3n_Digital_Terrestre_en_Ecuador

1.3. DIFERENCIAS ENTRE ISDB-T E ISDB-Tb

CARACTERÍSTICAS ISDB-T

- codificación de video en formato mpeg 2
- codificación audio en formato ACC
- una presentación de hasta 25 cuadros por segundo
- modulación OFDM-BST configurada como , 16-QAM, 64-QAM,QPSK o DQPSK
- Ancho de banda de 6 MHZ con portadoras agrupadas en 13 segmentos
- SFN (Single Frequency Network)
- Transmision de datos con interactividad
- Robustes de la señal contra perdidas multitrayectoria, desvanecimiento y demás interferencias.

CARACTERÍSTICAS ISDB-Tb

- codificación de video en formato mpeg 4
- codificación audio en formato HE-ACC
- una presentación de hasta 30 cuadros por segundo
- modulación OFDM-BST configurada como , 16-QAM, 64-QAM,QPSK o DQPSK
- Utilizacion del midleware Ginga
- Ancho de banda de 6 MHZ con portadoras agrupadas en 13 segmentos
- SFN (Single Frequency Network)
- Transmision de datos con interactividad
- Robustes de la señal contra perdidas multitrayectoria, desvanecimiento y demás interferencias.

La recepción de televisión digital en formato ISDB-T no es compatible con el formato ISDB-Tb, pero una señal en formato ISDB-Tb es compatible en audio y video, mas no en datos con el formato ISDB-T.

Resoluciones de la TV Digital

SDTV- 4:3 y 640 x 480	
EDTV - 16:9 y 1280 x 720	
HDTV - 16:9 y 1920 x 1080	
LDTV- 4:3 y 320 x 240	

Figura 2: RESOLUCIONES ISDB-TB

Fuente: (Bernal & Mejía , 2013)

Este tipo de características hacen posible transmitir audio, video y datos en HD, SD, LD para dispositivos portátiles por vía terrestre, cable o satélite en forma óptima.

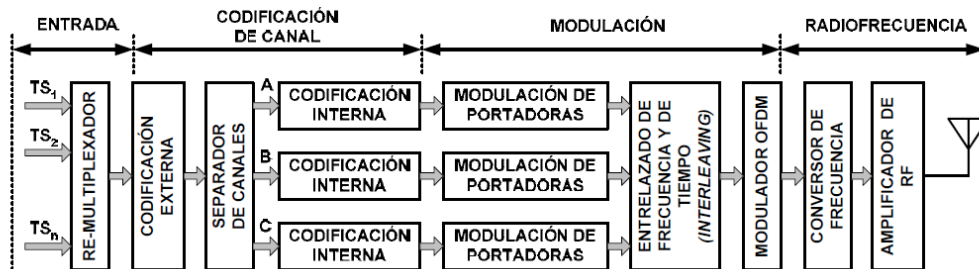


Figura 3: SISTEMA DE TRANSMISIÓN ISDB-TB

Fuente: (Pisciotta)

1.3.1. COMPENSACIÓN DE MOVIMIENTO CON BLOQUES DE TAMAÑO VARIABLE.

El objetivo de la partición de la imagen en macrobloques es mejorar la resolución de la imagen en su desplazamiento o movimiento al ser decodificada y presentada.

En estándares diferentes a MPEG-2 se presentan cuadros divididos en macrobloques con un tamaño de 16x16. En MPEG-2 los cuadros de 16x16 se dividen en cuatro bloques de 8x8. En MPG-4 los 4 cuadros de 8x8 se dividen en 2 bloques con tamaño de 8x16 o 16x8, o en 4 bloques de 8x8 y a su vez estos pueden ser divididos en 2 bloques de 4x8 o 8x4, o en 4 bloques de 4x4.

La figura 5 muestra las particiones posibles.

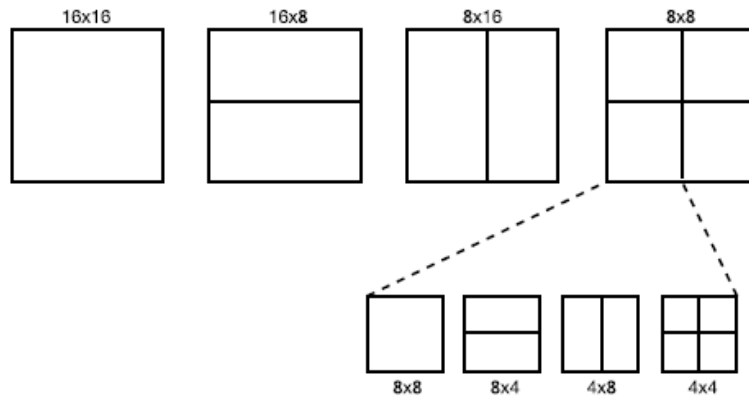


Figura 4: PARTICIONES
Fuente: (Sandoval)

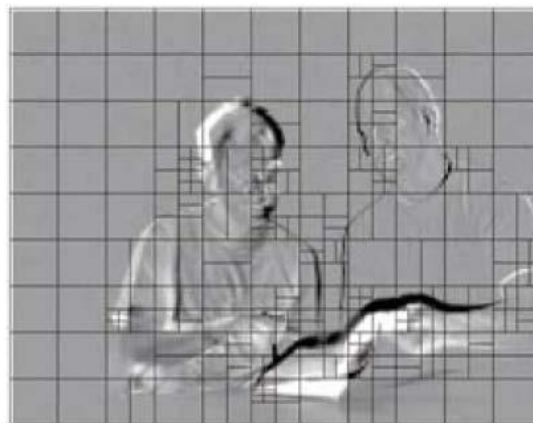


Figura 5: EJEMPLO DE SELECCIÓN OPTIMIZADA DE PARTICIÓN DE MACROBLOQUES
Fuente: (Sandoval)

1.3.2. CARACTERÍSTICAS DEL SISTEMA DE CODIFICACIÓN Y MODULACIÓN

La amodulación OFDM cuenta con una banda útil del canal dividida en 13 segmentos de 428 y agrupadas en 3 camadas diferentes para una transmisión gerarquica con diferentes transport stream TS de audio, video y datos denominado transmisión jerárquica, en que cada camada puede ser modulada con diferentes programas. Además mediante símbolos de corta duración se consigue permite que la señal sea resistente a distorciones e interferencias. El efecto Doppler presente en la recepción móvil es un problema contrarrestado con la configuración SFN. En OFDM se configura la separación de frecuencias en tres modos 1,2 y3 dosde caben 108 portadoras, 216 portadoras y 432 portadoras respectivamente con intervalos de guarda: 1/4, 1/8, 1/16 y 1/32.

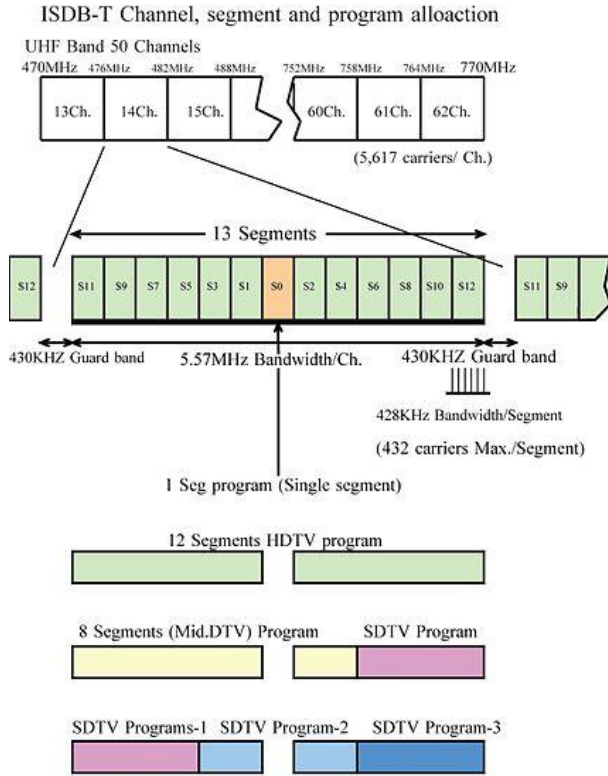


Figura 6: ESQUEMA ESTRUCTURA DEL SEGMENTO DEL ESPECTRO DE ISDB-T
 Fuente: (WIKIPEDIA, ISDB-T, 2013)

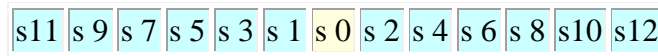


Figura 7: 13 SEGMENTOS DE ISDB-T
 Fuente: (WIKIPEDIA, ISDB-T, 2013)

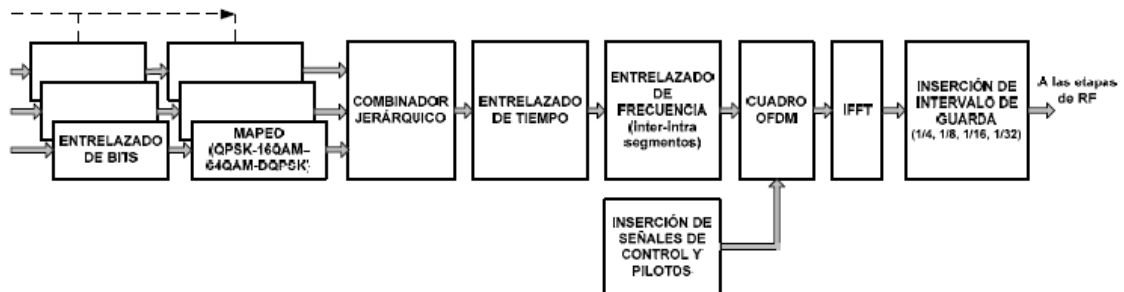


Figura 8: MODULACIÓN
 Fuente: (Sandoval)

1.3.3. FLUJO DE TRANSPORTE (TS) MPEG-2

La transmisión se la realiza jerárquicamente por capas A, B y C.

Los requerimientos necesarios para la configuración del flujo de transporte son:

- La codificación o creación de ES (Elementary Stream).
- La paquetización ES de tamaño variable.
- La formación del TS (Transport Stream) de longitud fija 188 bytes.

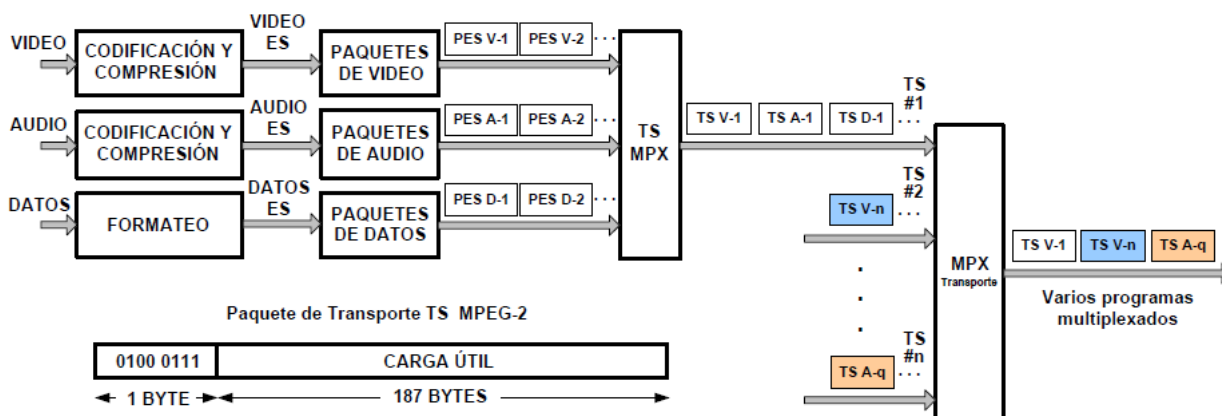


Figura 9: CODIFICACIÓN, PAQUETIZACIÓN Y MULTIPLEXACIÓN
Fuente: (Pisciotta)

1.3.4. MULTIPLEXACIÓN

1.3.4.1. FUNCIONAMIENTO BÁSICO DE UN MULTIPLEXOR

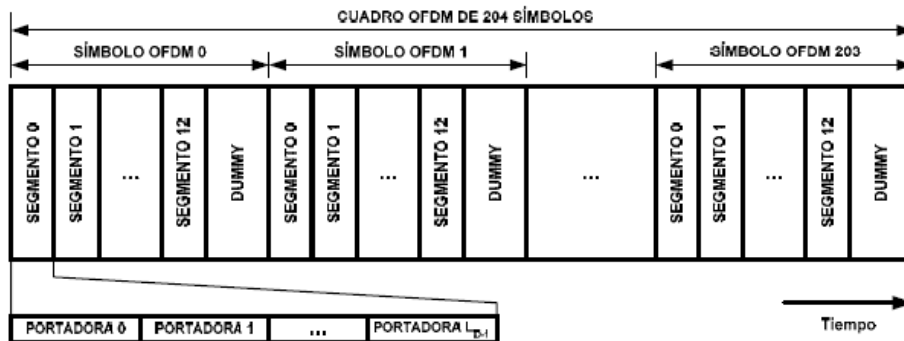


Figura 10: ORDENAMIENTO DEL CUADRO MÚLTIPLEX
Fuente: (Pisciotta)

El multiplexor agrupa las señales HD, SD y LD (ONE-SEG) y las convierte en una señal multiplexada denominada BTS (Broadcast Transport Stream) con una longitud de 32,507936 Mbps, en paquetes de 204 bytes de los cuales 188 bytes corresponden a la información útil y 16 bytes a la configuración para modulación y paridad.



Figura 11: BTS
Fuente: El Autor

El multiplexor señala e identifica diferentes señales a su entrada, esto gracias a la utilización de tablas PSI/SI que contienen toda la información requerida para multiplexar la obtención de un programa.

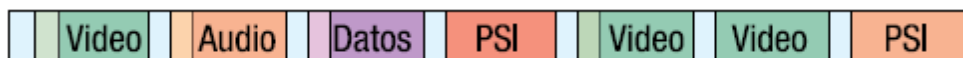


Figura 12: INFORMACIÓN PSI ENVIADA COMO INFORMACIÓN ÚTIL EN EL TS
Fuente: (Sandoval)

1.3.4.2. TABLAS PSI

Son tablas que contienen información, señalización, codificación de acceso condicional o información de red de los Elementary Streams a ser multiplexados de acuerdo al estándar MPEG-2.

Nombre	Significado/ Función
PAT	Tabla de Asociación de programas Asocia cada programa a un PID de la PMT.
PMT	Tabla de Mapeo de Programas Especifica los componentes de un programa o servicio.
CAT	Tabla de acceso condicional Información sobre el acceso condicional a los programas.

Tabla 2: TABLAS PSI
Fuente: (Olmedo , 2012)

1.3.4.3. TABLAS SI

Las tablas SI permiten enviar información de servicios adicionales como fecha y hora, aplicaciones interactivas, entre otros parámetros.

Nombre	Función
AIT	Tabla de información de aplicación. Transmite información y control sobre las aplicaciones transmitidas en el flujo.
BIT	Tabla de información de radiodifusor. Designa las unidades radiodifusoras y los parámetros SI para cada una de ellas.
EIT	Tablas de información de eventos. Transporta información sobre los eventos existentes en cada servicio y su orden cronológica de exhibición.
NIT	Tabla de información de red. Transporta información sobre la formación del TS y sobre su medio de transmisión
RST	Tabla de estado de evento. Permite la actualización del estado de eventos, por ejemplo cuando hay alteraciones en el horario de programación.
SDT	Tabla de descripción de servicios. Lista e identifica los servicios existentes en un TS.
TDT	Tabla de fecha y horario. Informa la hora y la fecha al sistema.
TOT	Tabla de cambio de fecha y hora. Informa al receptor la hora, fecha y huso horario.

Tabla 3: TABLAS SI
Fuente: (Olmedo , 2012)

1.3.5. MODIFICACIONES E IMPLEMENTACIONES BRASILEÑAS

Item	Adecuaciones realizadas
Descriptores de vídeo	Adaptación del sistema de multiplexación a fin de permitir el tráfico del vídeo H.264 no solamente para servicios de one-seg, sino también para servicios HD de acuerdo a lo especificado por la ABNT NBR 15602-1.
Descriptores de audio	Adaptación del sistema de multiplexación permitiendo señalar el formato de audio AAC y HE-AAC MPEG-4 LATM/LOAS adoptado por la SBTVD, bien como todas sus variaciones definidas en la norma ABNT NBR 15602-2
Descriptores de datos	Adaptación del sistema de multiplexador permitiendo la correcta señalización de los servicios de datos definidos en la norma ABNT NBR 15606
Canal Virtual	Utilización del campo <i>remote_control_key</i> a fin de permitir que el receptor mantenga la misma secuencia de canales utilizada en la TV Analógica.
Horario	Adecuación de las tablas responsables por enviar el horario para el televidente, a fin de permitir la señalización de los diversos husos horarios existentes en Brasil, bien como el horario de verano.
Clasificación indicativa	Adecuación de la tabla de forma a atender la Portaria nº. 1220 del Ministerio de Justicia, de 11.07.2007,
Tablas de Género y subgénero	Adecuación para atender a los géneros y subgéneros de los programas exhibidos en Brasil
EPG (<i>Electronic Program Guide</i>)	Adecuación de los descriptores responsables por enviar informaciones a los televidentes a fin de mantener la armonía entre lo que fue especificado en las diversas normas do SBTVD
Código de área (<i>area code</i>)	Estandarización de los códigos de área para o SBTVD
Identificadores de la emisora	Estandarización de los identificadores que deberán ser utilizados por las generadoras para no generar duplicidad en las señales transmitidas en una misma región.

Tabla 4: PRINCIPALES ALTERACIONES RELACIONADAS A LA NORMA
Fuente: (Sandoval)

1.3.6. TV INTERACTIVA – GINGA

Ginga es la tecnología desarrollada por Brasil y que consta de sistemas y sub-sistemas relacionados entre si para desarrollar y presentar interactividad. Consta de dos sub-sistemas, Ginga J (aplicaciones Java) y Ginga NCL (presentación multimedia). La programación esta basada en aplicaciones NCL y LUA las cuales.

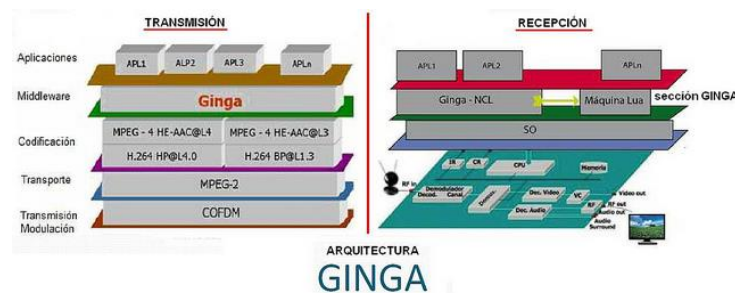


Figura 13: ARQUITECTURA GINGA
Fuente: (Gomes Soares)

1.4. REQUISITOS DE CODIFICACIÓN DE MEDIA Y MÉTODOS DE TRANSMISIÓN REFERIDOS EN DOCUMENTOS NCL³

1.4.1. MÉTODOS DE CODIFICACIÓN Y TRANSMISIÓN DE VIDEO – DATOS DE VIDEO REFERIDOS EN ELEMENTOS <media>

1.4.1.1. TRANSMISIÓN DE VIDEO MPEG-4 Y H.264|MPEG-4 AVC

Para poder transmitir video en formato MPEG-4 como un ES, los datos deben transmitirse como MPEG-2 packetized elementary stream (video PES), con el tipo de flujo especificado de acuerdo con la atribución de tipos de flujos ISO/IEC 13818-1 (valor 0x10 para video de acuerdo con H.264|MPEG-4 AVC).

1.4.1.2. TRANSMISIÓN EN SECCIONES MPEG-2

Para transmitir datos de video MPEG-4 ó H.264|MPEG-4 AVC por medio de secciones MPEG-2 específicas (ver atribución de tipos de flujos para secciones MPEG-2 en ISO/IEC 13818-1), uno de los métodos de transmisión siguientes se debe usar obligatoriamente:

- a) Como un archivo de flujo elemental de video MPEG-4 (o H.264|MPEG-4 AVC);
- b) Como un archivo de flujo multiplexado en el formato TS especificado en 12.4.

1.4.2. MÉTODOS DE CODIFICACIÓN Y TRANSMISIÓN DE AUDIO – DATOS DE AUDIO REFERIDOS EN ELEMENTOS <MEDIA>

1.4.2.1. TRANSMISIÓN DE AUDIO MPEG-4

Para transmitir contenido de audio MPEG-4 como un flujo elemental de audio, los datos del audio se deben transmitir obligatoriamente como MPEG-2 packetized elementary stream (audio PES), con el tipo de flujo especificado en conformidad con la atribución de tipos de flujos ISO/IEC 13818-1 (valor 0x11 para audio ISO/IEC 14496-3).

³ NBR, A. (19 de Noviembre de 2009). *NORMA BRASILEÑA* . Recuperado el 2014, de [http://www.advicom.ec/userFiles/files/Normas/ABNTNBR%2015606-2_2007Esp_2008Vc_2009-Codificacion%20Video,Audio%20y%20Multiplexacion%20\(Parte%202\).pdf](http://www.advicom.ec/userFiles/files/Normas/ABNTNBR%2015606-2_2007Esp_2008Vc_2009-Codificacion%20Video,Audio%20y%20Multiplexacion%20(Parte%202).pdf)

1.4.2.2. TRANSMISIÓN EN SECCIONES MPEG-2

Para transmitir datos de audio MPEG-4 por medio de secciones MPEG-2 específicas (ver atribución de tipos de flujos para secciones MPEG-2 en ISO/IEC 13818-1), uno de los métodos siguientes de transmisión se debe usar obligatoriamente:

- a) como un archivo de flujo elemental de audio MPEG-4;
- b) como un archivo de flujo multiplexado en el formato TS especificado en 12.4.

1.4.3. FORMATO TS PARA TRANSMISIÓN DE VIDEO/AUDIO MPEG

1.4.3.1. ESPECIFICACIÓN DE LA CODIFICACIÓN DE DATOS

a. TRANSMISIÓN DE VIDEO Y AUDIO MULTIPLEXADOS

Para transmitir datos de video MPEG-1/2/4 ó H.264|MPEG-4 AVC junto con datos de audio MPEG-1/2/4 ó AC3 en archivos multiplexados en secciones MPEG-2 específicas, cada archivo de video/audio multiplexado se codifica en un formato TS, conforme definido en la ISO/IEC 13818-1.

b. PSI REQUERIDO

Una tabla PAT se debe describir obligatoriamente. Cualquier PAT se debe describir obligatoriamente con el `program_number` cuyo valor es diferente de 0 y este valor debe representar obligatoriamente un PID de la PMT. Los valores disponibles de `program_number` serán definidos en un reglamento de estándar operativo. Una tabla PMT se debe describir obligatoriamente. Cualquier descriptor de identificación de flujo que indique un segundo loop debe contener obligatoriamente un descriptor PMT. En caso contrario, se puede insertar un descriptor conforme sea necesario. Se recomienda que los valores disponibles para `component_tag` y sus reglas de ocurrencia en descriptores ES y PMT defaults en un segundo loop sean equivalentes a un reglamento operativo estándar dedicado al flujo principal del tipo de media responsable por la transmisión del flujo en cuestión. En una implementación en la cual un flujo de transporte se decodifica desde un archivo que fue transmitido con base en la especificación de la codificación de datos definida en esta sección y se entrega en una interfaz digital de alta velocidad, una tabla SIT se debe describir obligatoriamente (ver la ABNT NBR 15606-1). En otros casos, las SIT no son necesarias, salvo especificación explícita en contrario.

Cualquier tabla diferente de PAT, PMT y SIT (por ejemplo: CAT, NIT, SDT, BAT, EIT, RST, TDT, TOT, PCAT, SDTT y St – ver la ABNT NBR 15601) obligatoriamente no debe ser descrita. Una tabla PAT debe ocurrir obligatoriamente en un flujo a una frecuencia no menor que una vez a cada 100 ms. Una tabla PMT debe ocurrir obligatoriamente en un flujo a una frecuencia no menor que una vez cada 100 ms. Por toda la duración de un archivo de formato TS, las tablas PAT y PMT obligatoriamente no deben ser modificadas o actualizadas.

1.4.4. TRANSMISIÓN EN SECCIONES MPEG-2

Para transmitir un archivo codificado con la especificación de codificación de datos de acuerdo con 12.6 en secciones MPEG-2 específicas, la transmisión debe estar de acuerdo obligatoriamente con ABNT NBR 15606-3.

1.4.4.1. RESTRICCIONES EN LA REPRODUCCIÓN

Para, al mismo tiempo, recibir un servicio por difusión y reproducir un archivo TS recibido a través de secciones MPEG-2 específicas, son necesarios dos sistemas de procesamiento de flujo de transporte separados. Las restricciones en la integración y coordinación de un contenido/evento recibido por un servicio de difusión con un archivo TS no son descritas en esta Norma.

1.4.5. ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DE IMÁGENES ESTÁTICAS Y GRÁFICOS DE BITMAP REFERIDOS POR ELEMENTOS <MEDIA>

1.4.5.1. TRANSMISIÓN DE MPEG-2 I-FRAME, MPEG-4 I-VOP Y H.264|MPEG-4 AVC I-PICTURE

Para transmitir una imagen estática en cuadros MPEG-2 I por medio de un componente video PES, el esquema de codificación debe obligatoriamente cumplir las convenciones definidas en la ABNT NBR 15606-1. El componente PES se debe transmitir obligatoriamente como un flujo cuyo valor de tipo es igual a 0x02.

Para transmitir una imagen estática en MPEG-4 I-VOP por medio de un componente video PES, el esquema de codificación debe obligatoriamente cumplir las convenciones definidas en la ABNT NBR 15606-1. El componente PES se debe transmitir obligatoriamente como un flujo cuyo valor de tipo es igual a 0x10. Para transmitir una imagen estática en H.264|MPEG-4 AVC I-picture por medio de un componente video PES, el esquema de codificación debe obligatoriamente cumplir las convenciones definidas en la ABNT NBR 15606-1. El componente PES se debe transmitir obligatoriamente como un flujo cuyo valor de tipo es igual a 0x1B.

1.4.5.2. TRANSMISIÓN EN SECCIONES MPEG-2 PARA REPRODUCCIÓN INTERACTIVA

Para transmitir una imagen estática en cuadros MPEG-2 I por medio de secciones MPEG-2, el esquema de codificación debe obligatoriamente estar de conformidad con la ABNT NBR 15606-1. La imagen estática se debe transmitir obligatoriamente como un archivo en la sección MPEG-2. Para transmitir una imagen estática en MPEG4-I-VOP por medio de secciones MPEG-2, el esquema de codificación debe obligatoriamente estar de

conformidad con la ABNT NBR 15606-1. La imagen estática se debe transmitir obligatoriamente como un archivo en la sección MPEG-2. Para transmitir una imagen estática en H.264|MPEG-4 AVC I-picture por medio de secciones MPEG-2, el esquema de codificación debe obligatoriamente estar conforme con las convenciones en la ABNT NBR 15606-1. La imagen estática se debe transmitir obligatoriamente como un archivo en la sección MPEG-2. En esos casos, el valor del tipo de flujo de la sección MPEG-2 debe obligatoriamente estar de acuerdo con la ISO/IEC 13818-1.

1.4.5.3. TRANSMISIÓN DE IMAGEN ESTÁTICA JPEG

Las imágenes estáticas JPEG se deben transmitir obligatoriamente a través de secciones MPEG-2 específicas (ver la atribución de tipos de flujos para secciones MPEG-2 en la ISO/IEC 13818-1)

1.4.5.4. ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DEL BITMAP PNG

Para los datos de bitmap PNG que son exhibidos solamente bajo el control de datos CLUT especificados por separado de esta Norma, los datos de la paleta, dentro de los datos PNG, pueden ser abreviados. El gráfico de bitmap PNG se debe transmitir obligatoriamente a través de secciones MPEG-2 específicas (ver la atribución de tipos de flujos para secciones MPEG-2 en la ISO/IEC 13818-1).

1.4.5.5. ESQUEMA DE CODIFICACIÓN Y TRANSMISIÓN DE LA ANIMACIÓN MNG

Para que los datos de bitmap PNG en el formato de animación MNG que son exhibidos solamente bajo el control de datos CLUT especificados por separado de esta Norma, los datos de la paleta dentro de los datos PNG pueden ser omitidos. El gráfico de animación de bitmap MNG se debe transmitir obligatoriamente a través de secciones MPEG-2 específicas (ver atribución de tipos de flujos para secciones MPEG-2 en la ISO/IEC 13818-1).

Esquema de codificación y transmisión de datos y animación de gráficos GIF Los datos de gráficos y animaciones GIF se deben transmitir obligatoriamente a través de secciones MPEG-2 específicas (ver atribución de tipos de flujos para secciones MPEG-2 en la ISO/IEC 13818-1).

1.5. SOFTWARE LINUX

1.5.1. SISTEMA OPERATIVO UBUNTU (LINUX)⁴

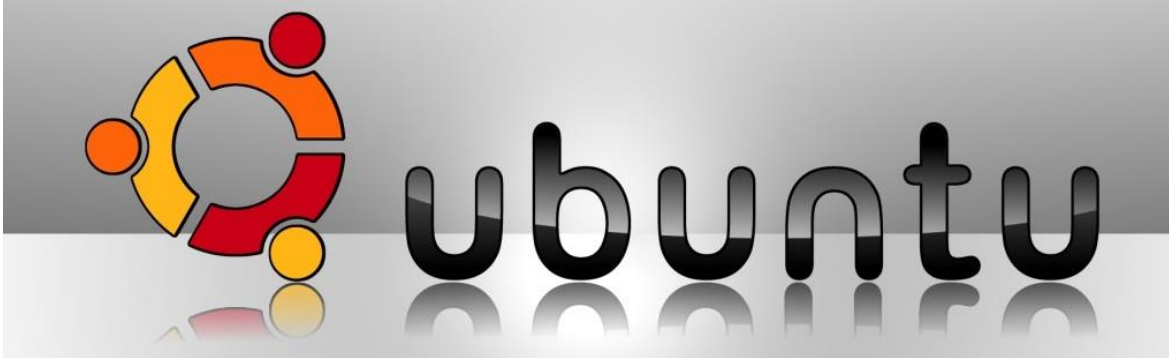


Figura 14: LOGO UBUNTU

Fuente: Autor

Ubuntu es un sistema operativo que se basa en Linux, es un software libre y cuenta con actualizaciones periódicas de sus complementos y características, casi todos los software de complemento para Ubuntu Linux son gratuitos, unos poseen interface grafica y otros no, pero todas estas características son importantes ya que permiten emplearlo como servidor, a pesar de que es un software orientado al usuario.

⁴ WIKIPEDIA. (4 de Enero de 2014). *Ubuntu*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/Ubuntu>

1.5.1.1. AYUDA DE COMANDOS UBUNTU

Ubuntu Reference		FOSSwire
<p>Privileges</p> <p>sudo <i>command</i> - run <i>command</i> as root</p> <p>sudo -s - open a root shell</p> <p>sudo -s -u user - open a shell as <i>user</i></p> <p>sudo -k - forget sudo passwords</p> <p>gksudo <i>command</i> - visual sudo dialog (GNOME)</p> <p>kdesudo <i>command</i> - visual sudo dialog (KDE)</p> <p>sudo visudo - edit <i>/etc/sudoers</i></p> <p>gksudo nautilus - root file manager (GNOME)</p> <p>kdesudo konqueror - root file manager (KDE)</p> <p>passwd - change your password</p>	<p>Network</p> <p>ifconfig - show network information</p> <p>iwconfig - show wireless information</p> <p>sudo iwlist scan - scan for wireless networks</p> <p>sudo /etc/init.d/networking restart - reset network for manual configurations</p> <p>(file) <i>/etc/network/interfaces</i> - manual configuration</p> <p>ifup interface - bring <i>interface</i> online</p> <p>ifdown interface - disable <i>interface</i></p>	
<p>Display</p> <p>sudo /etc/init.d/gdm restart - restart X and return to login (GNOME)</p> <p>sudo /etc/init.d/kdm restart - restart X and return to login (KDE)</p> <p>(file) <i>/etc/X11/xorg.conf</i> - display configuration</p> <p>sudo dexconf - reset <i>xorg.conf</i> configuration</p> <p>Ctrl+Alt+Bksp - restart X display if frozen</p> <p>Ctrl+Alt+FN - switch to tty <i>N</i></p> <p>Ctrl+Alt+F7 - switch back to X display</p>	<p>Special Packages</p> <p>ubuntu-desktop - standard Ubuntu environment</p> <p>kubuntu-desktop - KDE desktop</p> <p>xubuntu-desktop - XFCE desktop</p> <p>ubuntu-minimal - core Ubuntu utilities</p> <p>ubuntu-standard - standard Ubuntu utilities</p> <p>ubuntu-restricted-extras - non-free, but useful</p> <p>kubuntu-restricted-extras - KDE of the above</p> <p>xubuntu-restricted-extras - XFCE of the above</p> <p>build-essential - packages used to compile programs</p> <p>linux-image-generic - latest generic kernel image</p> <p>linux-headers-generic - latest build headers</p>	
<p>System Services¹</p> <p>start service - start job <i>service</i> (Upstart)</p> <p>stop service - stop job <i>service</i> (Upstart)</p> <p>status service - check if <i>service</i> is running (Upstart)</p> <p><i>/etc/init.d/service start</i> - start <i>service</i> (SysV)</p> <p><i>/etc/init.d/service stop</i> - stop <i>service</i> (SysV)</p> <p><i>/etc/init.d/service status</i> - check <i>service</i> (SysV)</p> <p><i>/etc/init.d/service restart</i> - restart <i>service</i> (SysV)</p> <p>runlevel - get current runlevel</p>	<p>Firewall¹</p> <p>ufw enable - turn on the firewall</p> <p>ufw disable - turn off the firewall</p> <p>ufw default allow - allow all connections by default</p> <p>ufw default deny - drop all connections by default</p> <p>ufw status - current status and rules</p> <p>ufw allow port - allow traffic on <i>port</i></p> <p>ufw deny port - block <i>port</i></p> <p>ufw deny from ip - block <i>ip</i> address</p>	
<p>Package Management¹</p> <p>apt-get update - refresh available updates</p> <p>apt-get upgrade - upgrade all packages</p> <p>apt-get dist-upgrade - upgrade with package replacements; upgrade Ubuntu version</p> <p>apt-get install pkg - install <i>pkg</i></p> <p>apt-get purge pkg - uninstall <i>pkg</i></p> <p>apt-get autoremove - remove obsolete packages</p> <p>apt-get -f install - try to fix broken packages</p> <p>dpkg --configure -a - try to fix broken packages</p> <p>dpkg -i pkg.deb - install file <i>pkg.deb</i></p> <p>(file) <i>/etc/apt/sources.list</i> - APT repository list</p>	<p>Application Names</p> <p>nautilus - file manager (GNOME)</p> <p>dolphin - file manager (KDE)</p> <p>konqueror - web browser (KDE)</p> <p>kate - text editor (KDE)</p> <p>gedit - text editor (GNOME)</p>	
	<p>System</p> <p>Recovery - Type the phrase "REISUB" while holding down Alt and SysRq (PrintScr) with about 1 second between each letter. Your system will reboot.</p> <p>lsb_release -a - get Ubuntu version</p> <p>uname -r - get kernel version</p> <p>uname -a - get all kernel information</p>	
<p><small>¹ Prefix commands with sudo to run. Ubuntu is a trademark of Canonical Ltd. Licensed under CC-BY-SA 3.0. Free to redistribute; see creativecommons.org for details.</small></p>		

Tabla 5: AYUDA DE COMANDOS UBUNTU
Fuente: (Peddicord, 2008)

1.5.2. SAMBA (UBUNTU-LINUX)⁵



Figura 15: LOGO SAMBA-UBUNTU
Fuente: Autor

Samba es el complemento y software libre de Linux que cuenta con interfaz gráfica y sirve para compartir datos en red, es la característica que hace de Ubuntu un servidor. Es compatible con sistemas operativos del mismo tipo y diferentes como Windows.

1.5.3. TOVID (UBUNTU-LINUX)⁶⁷



Figura 16: LOGO TOVID
Fuente: Autor

Es un complemento Linux sin interface gráfica que permite convertir archivos .AVI a otros formatos específicos, ya que tovid cuenta con una la función de mostrar los características de archivos codificados, es de gran utilidad en el proceso de codificación con su comando idvid.

⁵ WIKIPEDIA. (12 de Diciembre de 2013). *Samba (programa)*. Recuperado el 2014, de [http://es.wikipedia.org/wiki/Samba_\(programa\)](http://es.wikipedia.org/wiki/Samba_(programa))

⁶ Ubuntu, G. (4 de Noviembre de 2007). *Tovid*. Recuperado el 2014, de <http://www.guia-ubuntu.com/index.php?title=Tovid>

⁷ COMMONS, O. (s.f.). *Ubuntu Hacks/Multimedia*. Recuperado el 2014, de http://commons.oreilly.com/wiki/index.php/Ubuntu_Hacks/Multimedia

```
$ idvid sample.avi
```

```
-----  
script de identificación de vídeo idvid
```

```
Versión 0.25
```

```
Escrito en 2004 por Eric Pierce
```

```
-----  
La recopilación de información de vídeo. Esto puede tardar varios minutos,  
así que por favor sea paciente ...
```

```
=====
```

```
Archivo: sample.avi
```

```
Ancho: 512 píxeles
```

```
Altura: 384 píxeles
```

```
...
```

1.5.4. FFMPEG (UBUNTU-LINUX)⁸



Figura 17: LOGO FFMPEG

Fuente: Autor

Ffmpeg es un programa sin interfaz gráfica que permite establecer parámetros de codificación al formato MPEG-2 requerido en el manejo del ES Elementary Stream, permite configurar todas las características del estándar ISDB-Tb.

⁸ Ubuntu, G. (s.f.). *Ffmpeg*. Recuperado el 2014, de <http://www.guia-ubuntu.com/index.php/Ffmpeg#2281>

1.5.5. OPENCASTER (UBUNTU-LINUX)⁹



OpenCaster para SATVD-T

Figura 18: LOGO OPENCASTER
Fuente: (LIFIA, OpenCaster para SATVD, 2010)

OpenCaster desarrollado por AVALPA, es un software libre para generar TS en formato MPEG-2. Este software genera BTS con características del estándar DVB-T y no tiene soporte para el formato ISDB-Tb.

LIFIA que pertenece a la Universidad Nacional de la Plata en Argentina modificó OpenCaster para agregarle características específicas para el formato ISDB-Tb las cuales están incompletas, pero son suficientes para poder multiplexar paquetes TS y obtener una BTS compatible con la norma.

El software necesita los siguientes requisitos para su correcto funcionamiento:

- Un sistema operativo GNU/Linux
- El compilador de C de GCC
- Python

⁹ Venegas Picón, L. A. (22 de Mayo de 2012). *GENERACIÓN DE UNA TRAMA BROADCAST TRANSPORT STREAM (BTS) USANDO EL SOFTWARE LIBRE OPENCASTER*. Recuperado el 2014, de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1393/VENEGAS_PICON_LUIS_BTS_OPENCASTER.pdf?sequence=1

1.6. SOFTWARE WINDOWS

1.6.1. MEDIA PLAYER CLASSIC (WINDOWS)¹⁰



Figura 19: LOGO MPC
Fuente: Autor

Es un software compatible con múltiples formatos de audio y video para su reproducción, específicamente para reproducir archivos BTS en formato .TS, con lo que se puede apreciar el archivo multiplexado en MPEG-2 y ver si está listo para su modulación y transmisión.

1.6.2. ECLIPSE NCL (WINDOWS)¹¹



Figura 20: LOGO ECLIPSE
Fuente: (Systems, 2009)

Eclipse es el software de Windows compatible con el lenguaje Ginga NCL y LUA por medio de la instalación de los plugins correspondientes, cuenta con un entorno de desarrollo fácil de manipular y cuenta con todas las características necesarias que son compatibles con Ginga.

1.6.2.1. CARACTERÍSTICAS PRINCIPALES:

¹⁰ WIKIPEDIA. (24 de Septiembre de 2013). *Media Player Classic*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Media_Player_Classic

¹¹ Systems, L. o. (2009). *NCL Eclipse*. Recuperado el 2014, de http://laws.deinf.ufma.br/ncleclipse/pt-br:start?redirect=1#.Us8G6tIW3_G

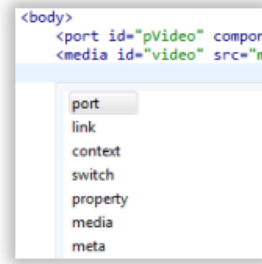


Figura 21: SUGERENCIA CÓDIGO CONTEXTUALES
Fuente: (Systems, 2009)



Figura 22: ERRORES DE MARCADO
Fuente: (Systems, 2009)

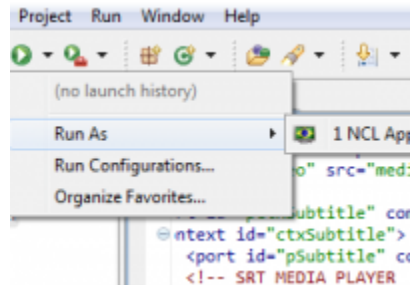


Figura 23: EJECUTAR LA APLICACIÓN
Fuente: (Systems, 2009)

1.6.3. GINGA EMULADOR (WINDOWS)¹²



Figura 24: LOGO GINGA
Fuente: (Brasileiro)

Es el software del middleware Ginga imprescindible en el desarrollo de aplicaciones interactivas, ya que permite la reproducción de las mismas una vez desarrolladas en eclipse en formato .NCL y admite configuraciones extras de reproducción.

1.6.4. STREAMXPRESS SOFTWARE (WINDOWS)¹³



Figura 25: LOGO STREAMXPRESS
Fuente: Autor

El DTC- 300 -SP StreamXpress es una herramienta fácil de utilizar de Windows TM paquete de software diseñado para proporcionar la reproducción en tiempo real de MPEG-2 flujos de transporte compatible y SD -SDI archivos grabados. StreamXpress está destinado a ser cargado por el usuario final en cualquier PC o portátil de calificación y el trabajo en conjunto con la mayoría de los dispositivos de salida DekTec. StreamXpress está diseñado para leer un transporte con formato correcto secuencia o archivo SD- SDI desde el disco duro del PC local y conectarlo a un dispositivo de salida para DekTec en streaming en tiempo real. StreamXpress soporta una gran variedad de interfaces de transmisión incluyendo ASI, ToSIP , RF modulada y salidas SD- SDI . StreamXpress proporciona

¹² Brasileiro, P. d. (s.f.). *Ginga*. Recuperado el 2014, de http://www.softwarepublico.gov.br/ver-comunidade?community_id=1101545

¹³ Dektec. (2013). *DTC-300 StreamXpress*. Recuperado el 2014, de <http://www.dektec.com/Products/Apps/DTC-300/>

información básica de flujo de transporte incluyendo: Contenido de la corriente de transporte, PID, así como información de archivo. Proporciona iniciar, detener, pausar, looping continuo y muchas más capacidades. Estas características, cuando se combina con la variedad de dispositivos de salida DekTec disponibles, dan al usuario una reproducción de alto valor potente y solución portátil. El DTC- 300 -SP es un producto de software con licencia de DekTec.

1.7. HARDWARE

1.7.1. ANTENA MICROSTRIP PARA LA BANDA UHF¹⁴

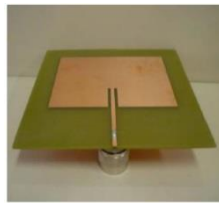


Figura 26: ANTENA MICROSTRIP

Fuente: (Garcia)

La radiación se hace presente en las discontinuidades y circuitos abiertos de la estructura, especialmente si su tamaño es comparable a la longitud de onda. No obstante el efecto de bordes se manifiesta en el contorno de la estructura y depende del grosor y la permitividad del dieléctrico.

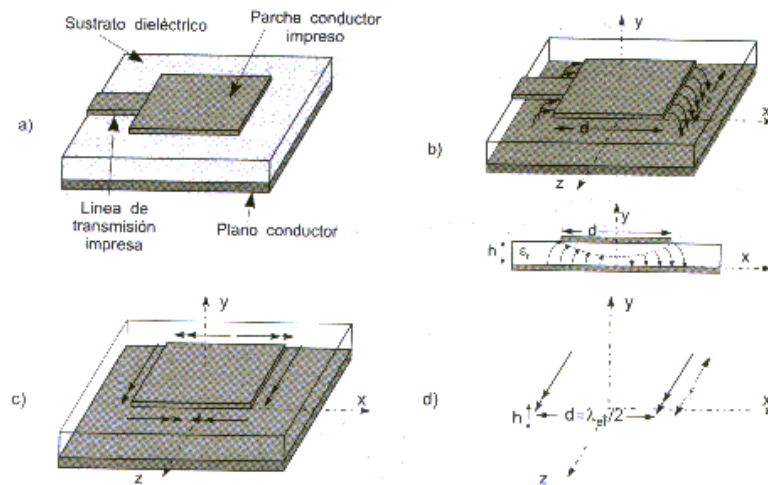


Figura 27: ANTENA MICROSTRIP RECTANGULAR

Fuente: (Garcia)

¹⁴ Garcia, F. (s.f.). *DISEÑO DE ANTENA MICROSTRIP PARA LA BANDA DE UHF*. Recuperado el 2014, de <http://www.docentes.unal.edu.co/jcgarciia/docs/Papers/ELE018.pdf>

El parche equivale a dos ranuras de dimensiones $W \times \Delta L$. La longitud L se elige para que haya una inversión de fase y la radiación de ambas ranuras se suma en fase, donde $L = \lambda/2$. La radiación existente en los flancos laterales del parche se cancela entre sí. El circuito equivalente es una línea de transmisión de longitud L , con dos impedancias que simulan las pérdidas de radiación y la capacidad de la discontinuidad y el circuito abierto



Figura 28: MONTAJE
Fuente: (Garcia)

1.7.2. MODULADOR DTA-2111¹⁵

Multi-Standard VHF/UHF
Modulator for PCI Express



Figura 29: MULTI-STANDARD VHF/UHF MODULATOR
Fuente: (Dektec, DTA-2111 Multi-Standard VHF/UHF Modulator for PCI Express, 2011)

1.7.2.1. CARACTERÍSTICAS DEL DTA-2111

- Modulador multi-estándar para PCI Express con soporte para la mayoría de los estándares de QAM-, y modulación OFDM basado en VSB.

¹⁵ Dektec. (2011). *DTA-2111 Multi-Standard VHF/UHF Modulator for PCI Express*. Recuperado el 2014, de <http://www.dektec.com/Products/PCIe/DTA-2111/>

- Conversión digital para una excelente calidad de señal sin necesidad de calibración.
- Soporta todas las constelaciones y los modos de modulación para cada estándar soportado.
- Upconverter Todos los Canales 36 .. 1002MHz totalmente ágil sobre la banda VHF y UHF.

1.7.2.2. REQUISITOS DEL PC

- Mínimo P4 @ 2GHz o equivalente AMD.
- Core 2 @ 2 GHz mínimo para ISDB-T, CMMB, DTMB, DVB-C2, DVB-T2 y simulación del canal.
- Mínimo de 512 MB de RAM.

1.7.2.3. DETALLES TÉCNICOS

KEY ATTRIBUTES		MODULATION STANDARDS	
Parameter	Value	Modulation	Standard
RF connector	75-Ω F female	ATSC VSB	ATSC A/53E
Frequency range	36 .. 1002MHz ±3ppm	CMMB*	GY/T 220.1/2-2006
Bandwidth (max)	8.0MHz	DTMB*	GB 20600-2006
I/Q sample rate	4.7 .. 9.375MHz	DVB-C	EN 300 429
Level	Range	DVB-C2*	EN 302 769
		DVB-T / DVB-H	EN 300 744
		DVB-T2*	EN 302 755
	Step size	I/Q*	Arbitrary I/Q samples
	Accuracy	ISDB-T*	ARIB STD-B31
MER	>40dB	QAM	J.83 Annex A/B/C
Adjacent channel	-54dB (QAM) -52dB (OFDM)	* Option	
Phase noise	<-95dBc @ 100kHz	ORDERING INFORMATION	
Spectral purity	>50dB @ -9dBm	Type	Description
Wideband noise	35dBμV (B=1MHz)	DTA-2111-SP	VHF/UHF modulator for PCI Express with <i>StreamXpress</i>
PCI Express	v1.1; PCIe 1x	DTA-2111-GOLD	DTA-2111 with all options
OS	XP/2k3/Vista/2k8/7 Linux 2.6	Please refer to www.dektec.com for the latest pricing and a list of distributors and resellers.	

Tabla 6: DETALLES TÉCNICOS

Fuente: (Dektec, DTA-2111 Multi-Standard VHF/UHF Modulator , 2011)

1.7.3. SET TOP BOX¹⁶

El STB es un receptor o decodificador de TDT configurado en el estándar original. Este dispositivo es necesario en televisores analógicos que no cuentan con un sintonizador digital integrado.

El set-top box trata el audio, video y datos por separado, analiza permisos con los cuales desifrar información y por medio de la interacción del usuario son ejecutados los datos que representan las aplicaciones o servicios interactivos.

¹⁶ WIKIPEDIA. (1 de Diciembre de 2013). *Set-top box*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Set-top_box

Puede tener un canal de retorno via Ethernet para televisión IP.
Para sintonizar datos en la norma ISDB-Tb se debe contar con el middleware Ginga incorporado al STB el mismo que detecte los servicios de interactividad transmitidos.

1.7.3.1. DECODIFICADOR DIGITAL (MUNDY HOME)¹⁷



Figura 30: DECODIFICADOR DIGITAL
Fuente: (Pionner, 2014)

Este pequeño decodificador permite la visualización de canales que sean transmitidos en señal digital en Televisores que no posean el decodificador instalado de fábrica con la norma Japonesa / Brasileña, la cual fue adoptada como estándar en el Ecuador.

¹⁷ Pionner. (2014). *Decodificador Digital*. Recuperado el 2014, de https://www.laesquinapioneer.ec/3-135-decodificador_digital.html%E2%80%8E

CAPITULO II

2. METODOLOGÍA

Métodos y técnicas para el desarrollo de la implementación.

Método Analítico.

Este método se utilizó para poder entender la estructura de una red de tv digital terrestre, así como su funcionamiento al transmitir y recibir generando tramas BTS mediante el control y análisis con software disponibles como StreamXpress, imponiendo a cada aplicación condiciones particulares y parámetros.

Método Comparativo.

Este tipo de método ayudará a realizar las diferentes comparaciones que se realizan al utilizar software y hardware de bajo costo con los de alto costo, para conseguir establecer una correcta implementación y funcionamiento de la red de tv digital terrestre.

Técnica de la Entrevistas.

Se utilizara este tipo de técnica para poder recopilar información de personas vinculadas al área de estudio como Ingenieros y técnicos.

2.1. TIPO DE ESTUDIO

El presente estudio está orientado a la línea de investigación de telecomunicaciones para sistemas de transmisión y recepción digital.

2.2. POBLACION Y MUESTRA

La implementación de un laboratorio de pruebas de tv digital terrestre isdb-tb, para la emisión de aplicaciones interactivas se realizara en la ciudad de Riobamba específicamente en la Universidad Nacional de Chimborazo.

2.3. OPERACIONALIZACIÓN DE VARIABLES

VARIABLES	CONCEPTO	CATEGORÍA	IDICADOR
Laboratorio de tv digital terrestre ISDB-Tb.	Red de telecomunicaciones de tv digital terrestre ISDB-Tb para pruebas de laboratorio	Generación, modulación y emisión.	<ul style="list-style-type: none">• Bit, Bytes• QAM, OFDM• MPEG-2• MHz
Desarrollo de aplicaciones interactivas.	Aplicaciones interactivas desarrolladas en GINGA NCL	Estándar ISDB-tb	<ul style="list-style-type: none">• Paquetes de datos de interactividad

Tabla 7: OPERACIONALIZACIÓN DE VARIABLES
Fuente: El Autor

2.4. OPERALIZACION DE HIPÓTESIS

CATEGORIAS	VARIABLES	INDICADORES	INDICES
Laboratorio de Tv digital terrestre ISDB-Tb	Transmisión de TDT	<ul style="list-style-type: none"> ➤ El 90% al 100% de la transmisión de una trama BTS en una hora con una resolución de 1920x1080 píxeles, una relación de aspecto de 16:9, contenido de audio y video codificado en MPEG-2 con un ancho de banda de 6 MHz. ➤ El 70% al 80% de la transmisión de una trama BTS en una hora con una resolución de 1280x720 píxeles, una relación de aspecto de 4:3, contenido de audio y video codificado en MPEG-2 con un ancho de banda de 6 MHz. ➤ El 60% de la transmisión de una trama BTS en una hora con una resolución de 720x480 píxeles, una relación de aspecto de 4:3 , contenido de audio estéreo y video codificado en MPEG-1 u otro formato con un ancho de banda de 6MHz. 	<p>ALTO (Alta definición calidad de DVD, aspecto panorámico o de cine. Se podrán enviar más programas y contenido por el mismo canal. Efecto de sonido envolvente permite al televidente involucrarse más con los contenidos.)</p> <p>MEDIO (Definición estándar calidad de DVD, Imagen de aspecto convencional. Se podrán enviar más programas y contenido por el mismo canal Efecto de sonido envolvente permite al televidente involucrarse más con los contenidos.</p> <p>BAJO (Resolución de televisión analógica, Imagen de aspecto Convencional, Se podrán enviar más programas y contenido por el mismo canal, Sonido convencional de calidad promedio</p>

<p>APLICACIONES INTERACTIVAS</p>	<p>Desarrollo de aplicaciones interactivas</p>	<ul style="list-style-type: none"> ➤ Desarrollo de un manual con 2 aplicaciones interactivas en middleware Ginga que los estudiantes de quinto año lo desarrollaran en 6 meses. ➤ Desarrollo de un manual que contenga 1 aplicacion interactiva en middleware Ginga que los estudiantes de quinto año lo desarrollaran en 6 meses. ➤ Ninguna. 	<p>ALTO (Sistemas de interactividad que representa un punto de encuentro entre la informática y la televisión.)</p> <p>MEDIO (Sistemas de interactividad que representa un punto de encuentro entre la informática y la televisión.)</p> <p>BAJO (Sistemas de interactividad que representa un punto de encuentro entre la informática y la televisión.)</p>
----------------------------------	--	--	---

Tabla 8: OPERALIZACION DE HIPÓTESIS
Fuente: El Autor

2.4.1. ANÁLISIS PREVIO PARA EL PLANTEAMIENTO HIPÓTESIS

Este cuadro representa la muestra del porcentaje de aprendizaje de las 31 personas que siguieron los manuales de prácticas para el desarrollo de aplicaciones interactivas.

PERSONAS	% DE APRENDIZAJE EN 1 SEMANA DE CLASES EN LA MATERIA DE OPTATIVA
AGUIRRE ALVAREZ CRISTIAN RAUL	4,67
ANILEMA GUADALUPE JUAN CARLOS	4,65
ARMIJOS CABEZAS FREDY DAVID	4,67
ASTUDILLO AMAN DARIO JAVIER	4,67
AUQUILLA LONDO DENIS PATRICIO	4,67
BUENAÑO CARRILLO DANIEL ALEJANDRO	4,67
CHACHA PILCO ALEX DAVID	4,67
CONTERO RAMOS ANDRES FELIPE	4,67
CONYA YAUTIBUG NELSON ROLANDO	4,65
CORDOVA ALVAREZ NADEZHDA ROCIO	4,67
CUJANO ORTEGA MAYRA VIVIANA	4,67
GAIBOR MELENDRES LUIS ALBERTO	4,67
HINOJOSA LUZURIAGA CRISTIAN RUBEN	4,66
LEMA YEPEZ MILTON EFRAIN	4,67
MACAS CANDO VERONICA ALEXANDRA	4,67
MARTINEZ CARRERA JOSE LUIS	4,66
MAYORGA MALA WILSON FERNANDO	4,67
MERA DEL PINO CINTHIA ELIZABETH	4,66
MORA HIDALGO FREDDY FERNANDO	4,66
PARRA CHAVEZ MARCO VINICIO	4,66
PIZANAN CARDENAS CHRISTIAN HUMBERTO	4,65
QUISI GUALLI FRANKLIN GEOVANNY	4,67
SILVA GUADALUPE VERONICA ELIZABETH	4,66
TINGO TIERRA MILTON MARCELO	4,66
VALDIVIEZO NARANJO SANTIAGO GUILLERMO	4,67
VERA GAIBOR CATHERINE IRALDA	4,67
YAGOS VILEMA CRISTHIAN MAURICIO	4,65
OLIVAREZ YACHIMBA JOSÉ VINICIO	4,65
PAZMIÑO JARA CRISTIAN	4,67
MEDINA ERIKA	4,67
Ing. ANIBAL LLANGA	4,67
PROMEDIO	4,67

Tabla 9: LISTADO DE APRENDIZAJE DE PERSONAS
Fuente: El Autor

Este cuadro es la proyección del estudio previo sobre del tiempo con respecto al promedio de aprendizaje de las 31 personas que siguieron los manuales de prácticas para el desarrollo de aplicaciones interactivas.

CUADRO DE PROYECCIÓN DE DESARROLLO DE INTERACTIVIDAD			
MESES	SEMANAS	PORCENTAJES	TOTAL
MES 1	SEMANA 1	4.67	18.68
	SEMANA 2	4.67	
	SEMANA 3	4.67	
	SEMANA 4	4.67	
MES2	SEMANA 1	4.67	18.68
	SEMANA 2	4.67	
	SEMANA 3	4.67	
	SEMANA 4	4.67	
MES 3	SEMANA 1	4.67	18.68
	SEMANA 2	4.67	
	SEMANA 3	4.67	
	SEMANA 4	4.67	
MES 4	SEMANA 1	4.67	18.68
	SEMANA 2	4.67	
	SEMANA 3	4.67	
	SEMANA 4	4.67	
MES 5	SEMANA 1	4.67	18.68
	SEMANA 2	4.67	
	SEMANA 3	4.67	
	SEMANA 4	4.67	
MES 6	SEMANA 1	4.67	4.67
	SEMANA 2		
	SEMANA 3		
	SEMANA 4		
			98%

Tabla 10: PROYECCION DE DESARROLLO DE INTERACTIVIDAD

Fuente: El Autor

2.4.2. PLANTEAMIENTO DE HIPÓTESIS

HIPÓTESIS NULA

Las pruebas TDT ISDB-Tb dependen de la Transmisión de TDT ISDB-Tb y del desarrollo de aplicaciones interactivas.

HIPÓTESIS ALTERNATIVA

Las pruebas TDT ISDB-Tb no dependen de la Transmisión de TDT ISDB-Tb y del desarrollo de aplicaciones interactivas.

Variable independiente Variable dependiente	Transmisión de TDT	Desarrollo de aplicaciones interactivas	TOTAL
PRUEBAS TDT ISDB-Tb % esperado	95	95	190
PRUEBAS TDT ISDB-Tb % obtenido	100	98	198
TOTAL	195	193	388

Tabla 11: TABLA DE HIÓTESIS
Fuente: El Autor

2.4.3 VALIDACION CHI CUADRADO

a) Calculo ni^* : $ni^* = n(p)$

$$n1^* = 195\left(\frac{190}{388}\right) = 95,489$$

$$n2^* = 195\left(\frac{198}{388}\right) = 99,51$$

$$n3^* = 193\left(\frac{190}{388}\right) = 94,51$$

$$n4^* = 193\left(\frac{198}{388}\right) = 98,489$$

b) Cálculo $(x)^2$

n_i	n_i^*	$n_i - n_i^*$	$(n_i - n_i^*)^2$	$(\frac{(n_i - n_i^*)^2}{n_i^*})$
95	95,489	-0,489	0,2391	$2,50 \times 10^{-3}$
100	99,51	0,49	0,2401	$2,4128 \times 10^{-3}$
95	94,51	0,49	0,2401	$2,54 \times 10^{-3}$
98	98,489	-0,489	0,2391	$2,427 \times 10^{-3}$
TOTAL				0,00987 $\Sigma(\frac{(n_i - n_i^*)^2}{n_i^*})$

Tabla 12: TABLA DE CALCULOS $(x)^2$
Fuente: El Autor

c)

PROBABILIDAD

$\alpha = 5\% = 0.05$

GRADOS DE LIBERTAD

$V = (k-1)(j-1)$

$V = (2-1)(2-1) = 1$

d)

$(x)^2_{0.05, 1} = 3,84$

e)

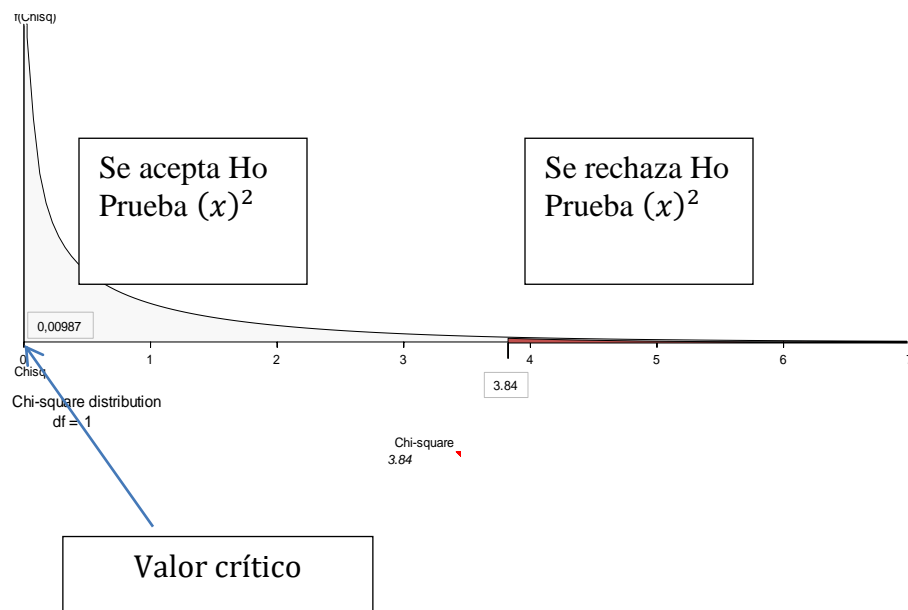


Figura 31: CURVA CHI-CUADRADO
Fuente: El Autor

$P(0,00987) > 5\%$ **HIPÓTESIS NULA VÁLIDA**

CONCLUSION

De acuerdo al análisis de la gráfica de validación CHI cuadrado se puede decir que la hipótesis nula se valida ya que la probabilidad de que esta ocurra es alta con respecto a la hipótesis alternativa.

La razón debe a que el laboratorio desarrollado cumple con los parámetros de transmisión además del desarrollo óptimo de aplicaciones interactivas por parte de los estudiantes que cursan la materia de optativa.

2.5. PROCEDIMIENTOS

La implementación del laboratorio de pruebas de tv digital terrestre isdb-tb, para la emisión de aplicaciones interactivas se lo realiza en dos etapas que son:

- **ETAPA DE GENERACIÓN Y TRANSMISIÓN BTS EN FORMATO ISDB-TB.**
- **ETAPA DE RECEPCIÓN DE UN SERVICIO DE TV DIGITAL CON INTERACTIVIDAD.**

2.5.1. ETAPA DE GENERACIÓN Y TRANSMISIÓN BTS EN FORMATO ISDB-TB.

Para esta etapa se emplean dos PCs A y B. En la A se instala el software para la codificación (ES), paquetización (PES), Formación de Paquetes TS y multiplicación. En la pc B se instala el software y hardware para el desarrollo de aplicación interactivas NCL y modulación.

Las dos PCs representan una sola etapa vinculadas entre si vía Ethernet.

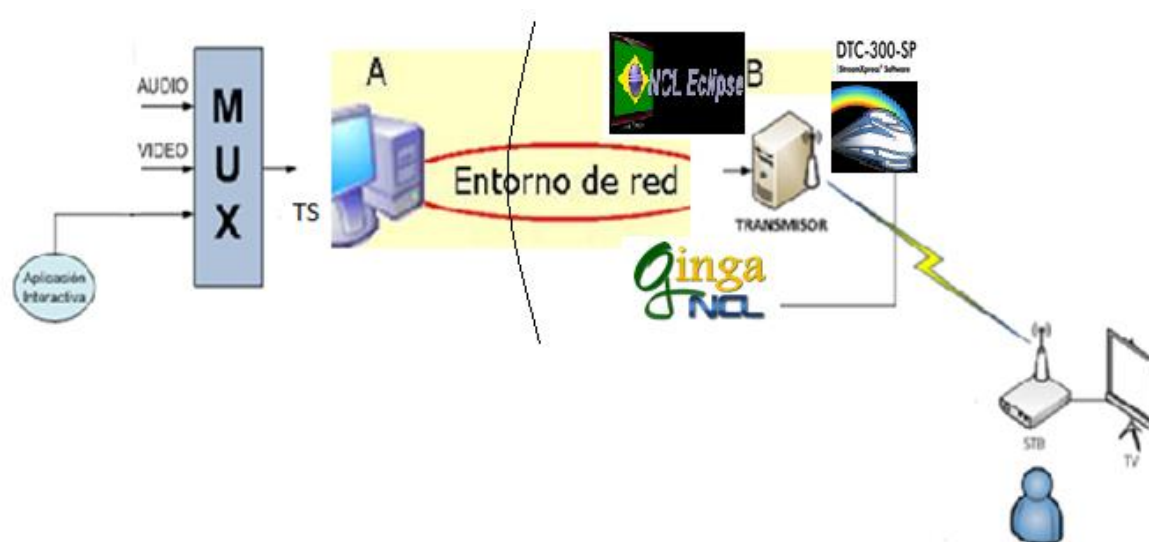


Figura 32: DIGRAMA DEL LABORATORIO DE TDT ISDB-TB

Fuente: El Autor



Figura 33: SERVIDORES ISDB-TB
Fuente: El Autor

2.5.1.1. PC B DESARROLLO DE APLICACIONES GINGA NCL Y LUA Y PRUEBAS DE LA BTS.

En esta pc se debe instalar el software necesario para el desarrollo de aplicaciones ginga ncl y lua. Para la programación en Ginga NCL o LUA se utiliza “eclipse” el mismo que se instala en Windows y se descarga de la dirección: <http://www.eclipse.org/downloads/>. Cabe indicar que para completar la instalación de eclipse, se deben instalar los plugins NCL y LUA necesarios para el desarrollo de aplicaciones interactivas estos se descargan de esta dirección <https://marketplace.eclipse.org/search/site/>.

Para la simulación de las aplicaciones desarrolladas en eclipse, se utiliza el Emulador de Ginga (Ginga4Windows) que es la versión para windows y se descarga de la dirección: <http://www.gingancl.org.br/pt-br/ferramentas>.

Para realizar las pruebas de multiplexacion(BTS) se debe instalar el reproductor”Media Player Classic” este reproductor es capaz de reproducir archivos .ts, por lo que servirá para realizar las pruebas de desarrollo en OpenCaster. El link de descarga de este software es el siguiente: <http://mpc-hc.org/>.

2.5.1.2. GESTION EN PC A PARA EL SERVIDOR LINUX

a. INSTALACION DE UBUNTU LINUX¹⁸

INICIANDO LA INSTALACIÓN DE UBUNTU

Para instalar Ubuntu, es necesario seguir una guía la misma que ayuda a quienes no tienen mucha práctica en el tema y se lo puede encontrar en:

<http://carlosmoncadasalazar.blogspot.com/2013/10/como-instalar-ubuntu-12.html>

Una vez iniciada la instalación es necesario particionar el disco, lo cual se lo debe hacer así:

Se deben crear 4 particiones: Una para /boot, otra para /, otra para /home, y otra para swap. El directorio /boot contiene todo lo necesario para que funcione el proceso de arranque del sistema, además almacena los datos que se utilizan antes de que el kernel comience a ejecutar programas en modo usuario, / se utiliza para almacenar las aplicaciones que se instalan, /home generalmente se usa para datos (ficheros e imágenes) y swap es un espacio temporal de intercambio.

En este caso los tamaños asignados serán de:

- /boot 500 Mb
- / 50000 Mb
- Swap 5000 Mb
- /home el resto de la memoria

b. CONFIGURACION DE LA RED EN UBUNTU¹⁹

Lo primero es buscar el icono de red y entrar a la opción “Editar las conexiones”, o ir directamente a “conexiones de red”.

¹⁸ Moncada Salazar, C. (22 de Octubre de 2013). *Como Instalar Ubuntu 12.04*. Recuperado el 2014, de <http://carlosmoncadasalazar.blogspot.com/2013/10/como-instalar-ubuntu-12.html>

¹⁹ González, D. (25 de Agosto de 2013). *[Tip] como cambiar el servidor DNS y la dirección IP en Ubuntu 12.04 / 12.10*. Recuperado el 2014, de <http://planetubuntu.es/post/tip-como-cambiar-el-servidor-dns-y-la-direccion-ip-en-ubuntu-12-04-12-10>

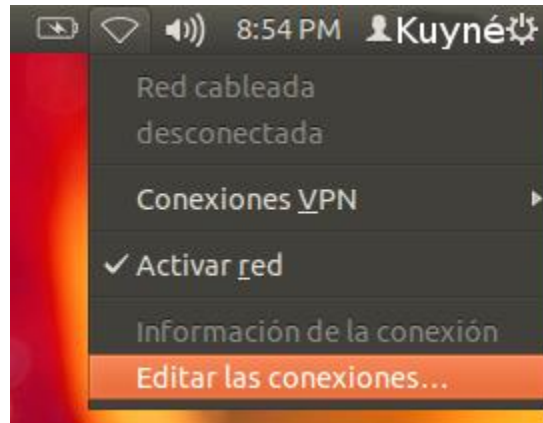


Figura 34: CONFIGURACION UBUNTU 1
Fuente: (González , 2013)

Una vez dentro, se selecciona la conexión cableada, luego se oprime el botón editar



Figura 49: CONFIGURACION UBUNTU 2
Fuente: (González , 2013)

Luego de ello se va a la pestaña “Ajustes de ipv4” y se selecciona la opción “manual”, finalmente se llenan los campos correspondientes al servidor dns y para cambiar la dirección IP solo aprieta el botón añadir y se llenan los campos dirección, mascara de red y puerta de enlace.



Figura 35: CONFIGURACION UBUNTU 3
Fuente: (González , 2013)

c. USAR APT-GET Y EL RESTO DE SISTEMA POR PROXY²⁰

En caso de que no estén configurados los proxy del sistema, se deben configurar para poder utilizar el comando apt-get de Linux el cual permite instalar paquetes y aplicaciones además de otras funciones como actualizaciones desinstalaciones comprobaciones etc.

Para esto se necesita editar el archivo de texto apt.conf en cual se ejecuta desde el terminal con el siguiente comando.

sudo gedit /etc/apt/apt.conf

Luego agregamos las siguientes líneas y guardamos.

```
Acquire::http::proxy "http://direccion_del_proxy:puerto/";
Acquire::https::proxy "https:// direccion_del_proxy:puerto/";
Acquire::ftp::proxy "ftp:// direccion_del_proxy:puerto/";
Acquire::socks::proxy "socks:// direccion_del_proxy:puerto/";
```

Ejemplo:

```
Acquire::http::Proxy "http://192.168.0.45:8080";
```

²⁰ Always, U. (11 de Diciembre de 2012). *Como usar apt-get y el resto de sistema por Proxy.*

Recuperado el 2014, de <http://ubuntualways.wordpress.com/2012/12/11/como-usar-apt-get-y-el-resto-de-sistema-por-proxy/>

d. INSTALACION DE OPENCASER2.4²¹

PAQUETES REQUERIDOS PARA EJECUTAR OPENCASER.

La instalación de estos paquetes se debe efectuar anteponiendo el comando de Linux apt-get install para cada caso.

Python2.7, Python-dev, Binutils, gcc, gcc-4.3, libc6-dev, libgomp1, LINUX-libc-dev, make, libpcap-dev, libpcap0.8, libpcap0.8-dev, zlib1g-dev.

BAJANDO LAS FUENTES

Las fuentes se pueden bajar del sitio original del OpenCaster en <http://www.avalpa.com/>. Hay que tener un usuario registrado, pero el registro es libre. Es recomendable bajar también el manual de OpenCaster del sitio original. Muchos de los detalles del uso de OpenCaster no se desarrollan en este documento, solo lo particular para el SATVD. La distribución que LIFIA modificó está versionada como 2.4. Las fuentes de dicha versión están disponibles también en <ftp://tvd.lifia.info.unlp.edu.ar/OpenCaster2.4/>.

Allí se encuentran los archivos:

OpenCaster.2.4.tgz Contiene la distribución original de OpenCaster 2.4 según distribuida por AVAL- PA

OpenCaster2.4-lifia-rev362.patch Es el archivo usado para aplicar los cambios de LIFIA a la distribución.

README.OpenCaster2.4-lifia-rev362.txt Contiene una descripción de instalación y los cambios hechos a la distribución original.

Se deben descargar todos los archivos a algún directorio temporal para su instalación. En adelante, consideramos que se bajan en:

`/home/tvd/OpenCaster/`

PREPARADO DE FUENTES Y COMPILACIÓN

Se deben seguir los siguientes pasos:

1. Descomprimir la versión original de los fuentes:

```
$ cd /home/tvd/OpenCaster/
```

```
$ tar xzf OpenCaster.2.4.tgz
```

2. Aplicar el patch de LIFIA :

²¹ LIFIA. (3 de Mayo de 2011). *OpenCaster para SATVD-T*. Recuperado el 2014, de <http://wiki.ginga.org.ar/lib/exe/fetch.php?media=lifia:guiaopencaster2.pdf>

```
$ cd OpenCaster2.4/  
$ patch -p 1 < ./ OpenCaster2.4-lifia-rev362.patch  
3. Compilar:  
$ make  
Esto deja las herramientas listas para instalar.
```

INSTALACIÓN

Esta parte de la instalación requiere el uso de la cuenta root.

```
$ sudo su  
# make install
```

Una vez que el software está instalado, ya se pueden generar transport streams. Para probar la instalación, ejecutar el siguiente comando:

```
$ python -c "from dvbobjects.PSI.PAT import *"
```

El comando debería ejecutar sin mostrar ningún error en la salida

e. INSTALACIÓN DE COMPLEMENTOS LINUX PARA GENERACIÓN ISDB-Tb

FFMPEG Y TOVID PARA LA CODIFICACIÓN (ES), PAQUETIZACIÓN (PES), Y FORMACIÓN DE PAQUETES TS.

Para instalar estos complementos de Linux se puede utilizar una herramienta muy útil denominada Centro de software de Ubuntu, donde se busca lo que se desea instalar o si no se lo puede hacer mediante el comando de linux apt-get install.

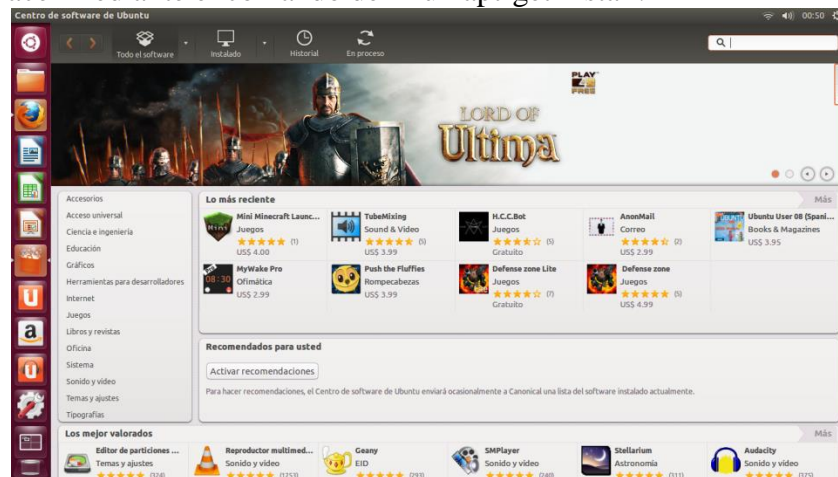


Figura 36: INTERFAZ GRÁFICA UBUNTU
Fuente: Autor

f. SCRIPT PARA GENERAR LAS TABLAS PSI/SI DEL TS EN FORMATO ISDB-TB

A este script que generara las tabla PSI/SI explicado en el capítulo 1 se lo nombra como gtables1.py y consta de un servicio por medio de la tabla PMT y una aplicación interactiva por medio de la tabla AIT, la extensión .py significa que será compilado con python para posteriormente ser multiplexado por OpenCaster en formato ISDB-tb.

El script del archivo gtables1.py se muestra en el anexo 1

g. CODIFICACIÓN (ES), PAQUETIZACIÓN (PES), Y FORMACIÓN DE PAQUETES TS.²²²³

CODIFICACION (ES).

En la codificación ES es importante que se tome como base un archivo mpeg4 y que este sea convertido a formato avi que es un formato compatible con FFmpeg, manteniendo las características del archivo base como tasa de bits o bit rate en caso del video, y la frecuencia de muestreo o sampling rate en el caso del audio.

Los parámetros y características del archivo para su codificación se los puede consultar con el comando (TOVID ID nombre del archivo), utilidad del complemento Linux TOVID.

²² Gutiérrez Tapia, A. G. (8 de Abril de 2013). *DISEÑO DE UN LABORATORIO DE TELEVISIÓN DIGITAL PARA LA TRANSMISIÓN DE SEÑALES CON MULTIPROGRAMACIÓN, CONTENIDOS INTERACTIVOS Y GUÍA ELECTRÓNICA DE PROGRAMACIÓN (EPG)*. Recuperado el 2014, de <http://dspace.ucuenca.edu.ec/bitstream/123456789/372/1/tesis.pdf>

²³ Illescas Robalino, M. A., & Villamarín Zapata, D. F. (2011). *IMPLEMENTACIÓN DE UN TRANSMISOR DE PRUEBAS DE TV DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE APLICACIONES INTERACTIVAS*. Recuperado el 2014, de <http://repositorio.espe.edu.ec/bitstream/21000/4289/1/T-ESPE-032586.pdf>

```
prueba@prueba: ~/tvd/servicio2
Not compliant with (S)VCD or DVD
This video does not seem to be compliant with (S)VCD or DVD
standards. If you burn it to a video disc, it may not work.
=====
prueba@prueba:~/tvd/servicio2$ tovid id output3.avi
Read options from /home/prueba/.tovid/tovid.ini:
-----
tovid id
Identify video files
Version 0.33
http://tovid.wikia.com
-----
Analyzing file: 'output3.avi'...
=====
File: output3.avi
Width: 720 pixels
Height: 480 pixels
Aspect ratio: 1.33:1
Frames: 5452
Duration: 00:03:01 hours/mins/secs
Framerate: 30.000 frames per second
Video format: FMP4
Video bitrate: 1432648 bits per second
-----
Audio track 1 (Stream 0.1, AID 0):
-----
Codec: ac3
Bitrate: 160000 bits per second
Sampling rate: 44100 Hz
=====
Audio is compliant with the following formats:
Not compliant with (S)VCD or DVD
Video is compliant with the following formats:
Not compliant with (S)VCD or DVD
This video does not seem to be compliant with (S)VCD or DVD
standards. If you burn it to a video disc, it may not work.
=====
prueba@prueba:~/tvd/servicio2$
```

Figura 37: SALIDA EN EL TERMINAL TOVID-IDVID
Fuente: El Autor

CODIFICACION DEL ARCHIVO BASE MPEG4 A .AVI

COMANDO EN TERMINAL

En caso de tener un archivo mp4 incompatible con la norma ISDB-Tb, puede variarse sus parámetros como son el bit rate de video y de audio, aspecto y resolución.

```
$ ffmpeg -i video.mp4 -b 2973040 -r 25 -s 720x480 -aspect 4:3 -ab 128k vid2.avi
```

DESCRIPCION

- Fuente de video (-i): video.mp4
- Bit rate de video (-b): 2973040 bps
- Frames por Segundo (-r): 25 fps
- Resolución de Video (-s): 720x480
- Aspecto de Pantalla (-aspect): 4:3
- Bit rate de audio (ab): 180k
- Salida de video (-f): video.avi

OBTENCION DEL ARCHIVO ES DE VIDEO MEDIANTE LA CODIFICACION DEL ARCHIVO .AVI A .MPEG2.

COMANDO EN TERMINAL

```
$ ffmpeg -i vid2.avi -an -vcodec mpeg2video -f mpeg2video -s 720x480 -r 25 -aspect 4:3 -deinterlace -b 2973k -maxrate 2973k -minrate 2973k -bf 2 -bufsize 1835008 vid2.m2v
```

DESCRIPCION

- -i: Archivo fuente de audio y video en formato .avi
- -an: ignora a la señal de audio
- -vcodec: mpeg2video mpeg2video
- -f: Formato de salida mpeg2video
- -s: Resolución del video en pixeles (EXPLICADO EN EL CAPÍTULO 1)
- -r: número de frames por segundo (Sincronia de tiempo en audio y video) 25 fps (PARAMETRO FFmpeg OpenCaster)²⁴
- -aspect: aspecto de la presentación en la pantalla
- -deinterlace: opción para desentrelazar imágenes
- -b: tasa de bits de la señal codificada Y para mantenerla constante ponemos este mismo parámetro en maxrate y minrate, que son las tasas máximas y mínimas respectivamente
- -bf: Este parámetro indica “bidirectionally predictive coded picture” el cual contiene diferente información de la trama anterior o siguiente (I o P) dentro de un Group of Pictures (GOP). Para este ejemplo se define un valor de 2.
- -bufsize: Es el tamaño del buffer en bits. Esta cantidad está directamente relacionada con el buffer VBV (video buffer verifier) mediante la siguiente expresión: $vbv_buffersize * 1024 * 16 = buffersize = 1835008$, donde el vbv_buffersize es 112 (el valor del Video Buffer Verifier para MPEG-2 puede ser 112 Kbps o 224 Kbps).
- Vid2.m2v: Nombre del archivo generado

OBTENCION DEL ARCHIVO ES DE AUDIO MEDIANTE LA CODIFICACION DEL ARCHIVO .AVI A .MPEG2.

²⁴ <http://www.avalpa.com/assets/freesoft/opencaster/AvalpaBroadcastServerUserManual-v2.1.pdf>

COMANDO EN TERMINAL

```
ffmpeg -i vid2.avi -vn -ac 2 -acodec mp2 -f mp2 -ab 160000 -ar 48000  
aud2.mp2
```

DESCRIPCION

- -i: Archivo fuente de audio y video en formato .avi
- -vn: ignora a la señal de video
- -ac: Define el número de canales. Por defecto es igual a 1
- -acodec: códec de audio mp2 compatible con el Transport Stream.
- -f : Formato de salida (mp2)
- -ab: tasa de bits de la señal de audio
- -ar: frecuencia de muestreo de la señal de audio la cual puede ser de 32000 Hz, 44100 Hz o de 48000 Hz dependiendo del archivo.
- Aud2.mp2: Nombre del archivo generado

PAQUETIZACIÓN (PES)

PAQUETIZACIÓN (PES) DEL ARCHIVO ES DE VIDEO

COMANDO EN TERMINAL

```
esvideo2pes vid2.m2v > vid2.pes
```

DESCRIPCION

- vid2.m2v: Video ES a ser convertido
- vid2.pes: Video convertido a PES

PAQUETIZACIÓN (PES) DEL ARCHIVO ES DE AUDIO

COMANDO EN TERMINAL

```
aud2.mp2 1152 48000 480 3600 > aud2.pes
```

DESCRIPCION

- aud2.mp2: Audio ES a ser convertido
- aud2.pes: Audio convertido a PES

- 1152: Numero de muestras por frame

	MPEG 1	MPEG 2 (LSF)	MPEG 2.5 (LSF)
Capa 1	384	384	384
Capa 2	1152	1152	1152
Capa 3	1152	576	576

Tabla 13: NUMERO DE MUESTRAS POR TRAMA DE AUDIO

Fuente: (Gutiérrez Tapia, 2013)

- 48000: Frecuencia de muestreo, debe ser la misma que se definió en el ES.
- 480: Tamaño del frame de audio.
- 3600: PTS (Presentation TimeStamp), se define para poder sincronizar con el video .

FORMACIÓN DE PAQUETES TS

PAQUETIZACIÓN TS DE VIDEO

COMANDO EN TERMINAL

```
pesvideo2ts 2065 25 112 3418996 0 vid2.pes > vid2.ts
```

DESCRIPCION

- 2065: Numero de PID asignado al video
- 25: Numero de Frames por Segundo
- 112: valor VBV (Video Buffer Verifier) definido en el estándar MPEG-2
- 3418996 ancho de banda de la señal de Transport Stream 15% más grande que la tasa de bits de la señal
- 0: Indica que el video no va a estar en un loop
- vid2.pes: Video PES a ser convertido
- vid2.ts: Video de salida ya en formato TS

PAQUETIZACIÓN TS DE AUDIO

```
pesaudio2ts 2075 1152 48000 480 0 aud2.pes > aud2.ts
```

DESCRIPCION

- 2075 Numero de PID asignado al audio

- 1152 Numero de muestras por frame, se debe mantener el valor que se definió en el PES.
- 48000 Frecuencia de muestreo, se debe mantener el valor que se definió en el PES.
- 480 Tamaño de frame de audio, se debe mantener el valor que se definió en el PES.
- 0 Indica que el audio no va estar en un loop.
- aud2.pes Audio PES a ser convertido.
- aud2.ts Audio de salida ya en formato TS

CREACION DE CARROUSEL DE OBJETOS (app_ginga.ts)

Para este propósito se utiliza el siguiente comando:

```
$ oc-update.sh app_ginga 0x0C 1 2004 2
```

h. MULTIPLEXACIÓN MPEG2 DEL TRANSPORT STREAM CON UN SERVICIO PARA INTERACTIVIDAD²⁵

Para poder efectuar la multiplicación se debe crear un archivo vacío null.ts el mismo que guardara paquetes nulos que completaran el ancho de banda de 6 MHz.

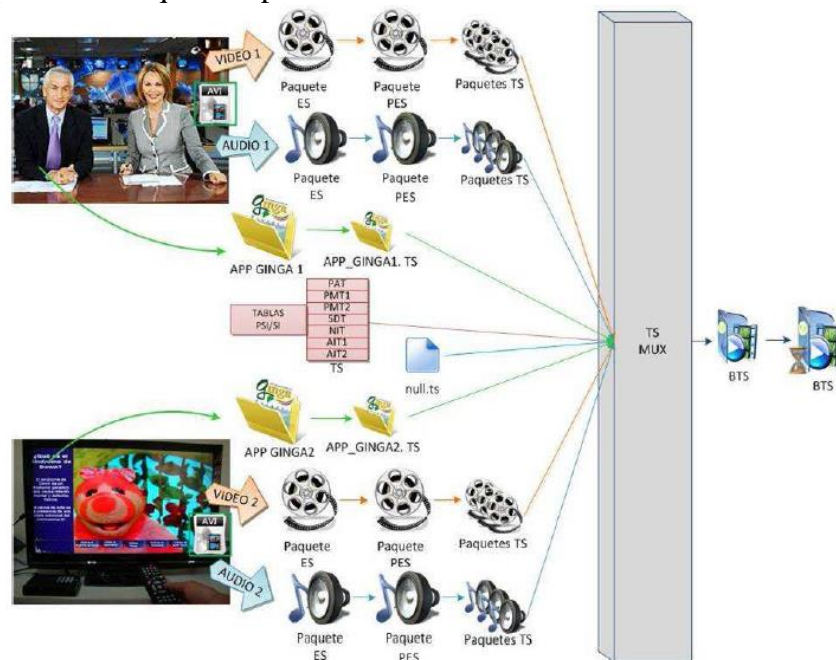


Figura 38: MULTIPLEXACIÓN
Fuente: (Olmedo , 2012)

²⁵ LIFIA. (3 de Mayo de 2011). *OpenCaster para SATVD-T*. Recuperado el 2014, de <http://wiki.ginga.org.ar/lib/exe/fetch.php?media=lifia:guiaopencaster2.pdf>

MULTIPLEXADO DEL TRANSPORT STREAM

Multiplexando con la aplicación:

```
$ tsbrmuxer \  
3620000 \  
b:15040 pat.ts \  
b:15040 pmt_sd.ts \  
b:3008 sdt.ts \  
b:3008 nit.ts \  
b:3008 ait.ts \  
b: 600000 app_ginga.ts \  
b: 2973040 video.ts \  
b: 160000 audio.ts \  
b: 2618 null.ts > audvid2.ts  
$ tsstamp audvid2.ts 29958294 > prueba.fixed.ts
```

Un detalle a tener en cuenta es el tamaño de la aplicación. Si la aplicación es muy grande, el archivo de carousel generado (app_ginga.ts) será grande también. Como la velocidad de transmisión está fija en 400.000 bps, hay que asegurar que la cantidad de segundos multiplexados son suficientes para transmitir todo el carousel al menos una vez. Si esto no sucede, el carousel va a estar incompleto y por lo tanto la aplicación nunca se va a poder bajar. A modo de ejemplo, si el archivo es de 1200 KiB (kilobytes), su tamaño es 9.830.400 bits. A 400.000 bps, el archivo tardará 24,576 segundos para transmitirse por completo, casi los 30 segundos para un archivo de 1,2 MiB.

Si quiere transmitir archivos más grandes, puede o bien aumentar la velocidad de transmisión, o generar un multiplexado de mayor duración.

Un poco de detalle acerca de los números:

3620000 Es la cantidad de paquetes a multiplexar. El sistema ISDB-T transmite cerca de 20000 paquetes por segundo, con lo cual, el archivo generado es de cerca de 3.1 minutos.

b:15040 Tanto la PAT como la PMT deben ser enviadas al menos 10 veces por segundo. Sabiendo

que cada una de las tablas entra en un solo paquete de 188 bytes, tenemos que enviar 10 paquetes por segundo. Y como cada paquete es de 188 bytes $8 = 1504$ bits, queremos que el ancho de banda sea de $1504 \text{ bits} \times 10 = 15040$ bps. Un análisis similar se necesita para la NIT y SDT.

b: 2973040, b: 160000 Son el ancho de banda del audio y del video respectivamente. Más detalle se puede encontrar en el manual de OpenCaster.

b: 26186150 Es el ancho de banda de paquetes nulos. El sistema ISDB-T tiene un ancho de banda fijo de 29.958.294 bps, y como estamos usando:

$15040*2+3008*3+2973040+160000+600000 = 3772144$ en total, tenemos que completar el Transport stream con paquetes nulos. El ancho de banda requerido se deriva de: $29958294-3772144 = 26186150$ bps.

Tener en cuenta que el ancho de banda del sistema ISDB-T es muy grande. Generar Transport streams muy largos es una de las formas más rápidas que conozco para agotar el espacio en disco.

Ahora se obtiene un archivo prueba1.ts que tiene el primer multiplexado. Sin embargo, hay un problema con el PCR. El PCR es la referencia del reloj del sistema que trabaja en 27 Mhz en el que viaja en el stream de video.

Como se cambió la posición de los paquetes del video, la llegada de los mismos al STB es imprecisa. Para arreglar esto existe una herramienta llamada tsstamp expresada así:

```
$ tsstamp prueba1.ts 29958294 > prueba.fixed.ts
```

Al ingresar este comando streamxpress muestra el error “Bit rate of select components excedes later capacity”, por esta razón se debe fijar un valor diferente en el ancho de banda de paquetes nulos, variando sus decimales.

i. COMPARTIR ARCHIVOS VIA ETHERNET EN UBUNTU²⁶

INSTALACION Y CONFIGURACION DE SAMBA PARA COMPARTIR LA CARPETA “APP_ginga” DESDE LA PC A HACIA LA PC B.

INSTALACIÓN

Para instalar Ubuntu se lo debe hacer desde el centro de software Ubuntu, ya q es un software libre su instalación no es complicada.

Una vez instalado Ubuntu se lo debe configurar, para poder compartir archivos en red con Windows, la configuración de samba se lo puede obtener del siguiente link de internet:

<http://blog.desdelinux.net/instalacion-y-configuracion-de-samba-en-ubuntu-12-04/>

En esta configuración se debe vincular el usuario de Ubuntu del servidor que es “prueba”, con el directorio prueba/tvd/servicio2/app_ginga o prueba/tvd/servicio2/servicio2/app_ginga.

²⁶ DesdeLinux. (s.f.). *Instalación y configuración de SAMBA en Ubuntu 12.04*. Recuperado el 2014, de <http://blog.desdelinux.net/instalacion-y-configuracion-de-samba-en-ubuntu-12-04/>

2.5.1.3. COMPARTIR EL ARCHIVO NCL CREADO EN ECLIPSE DESDE PC B EN WINDOWS

Para compartir un archivo desde eclipse en Windows solo se debe seleccionar la ubicación de workspace para que se guarde siempre el archivo creado, para esto hay dos formas, cuando recién se instala eclipse y después de instalado por si la dirección inicial no fue la correcta.

a. RECIEN INSTALADO ECLIPSE

Aparecerá una interfaz gráfica al inicio de la instalación de eclipse donde se selecciona la ubicación de destino ubicándonos en browse, en este caso en red se selecciona prueba.

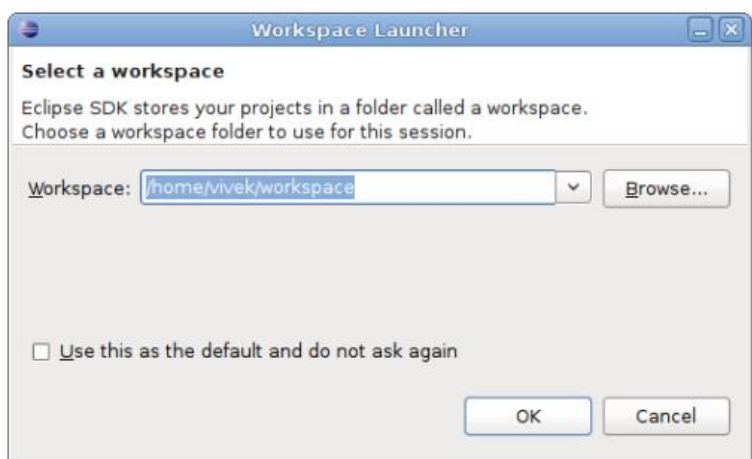


Figura 39: WORKSPACE ECLIPSE
Fuente: Autor

b. DESPUES DE INSTALADO ECLIPSE

Se puede cambiar la ubicación del Workspace seleccionando preferences, General/Startup and Shutdown/Workspaces.

Una vez realizada la multiplexación, el archivo BTS está listo para pasar a la etapa de modulación vía ethernet donde mediante el software streamxpress, se configura todos los parámetros necesarios de modulación digital en formato ISDB-T del BTS para poder ser transmitido mediante la tarjeta Dektec DTA-2111 empleando en la misma una antena UHF adecuada para el rango de frecuencias a las que funciona.

2.5.1.4. INSTALACIÓN DEL SOFTWARE DTC- 300 -SP²⁷

El software StreamXpress DTC- 300 se puede instalar en cualquier PC de Microsoft Windows adecuado. Al instalar el software StreamXpress, el paquete de instalación instalará el StreamXpress aplicación y el controlador de tarjetas USB y PCI.

Descargar la versión más reciente de StreamXpress del sitio web de DekTec en <http://www.dektec.com/Downloads/Applications.asp>.

Descomprimir el software en un directorio temporal

NOTA: Se debe estar conectado como administrador para instalar el software.

Cerrar cualquier otro software que ya pueda estar utilizando hardware DekTec.

EJECUCION STREAMXPRESS “SETUP.EXE”

Tener en cuenta que si está ejecutando Win7 se necesita hacer clic derecho sobre el ejecutable y seleccionar “Ejecutar como administrador ‘. El no hacerlo puede dar lugar a una mala instalación, incluso si se tiene los derechos de administrador.

Seguir las instrucciones en la pantalla hasta que aparezca seleccionar Completa o Personalizada.

Completa se instalará el software StreamXpress y los controladores para USB DekTec y PCI / PCIe dispositivos. Esto sobrescribirá los controladores existentes en el PC.

Instalación personalizada permite al usuario elegir qué elementos se instalará y la ubicación de la instalación.

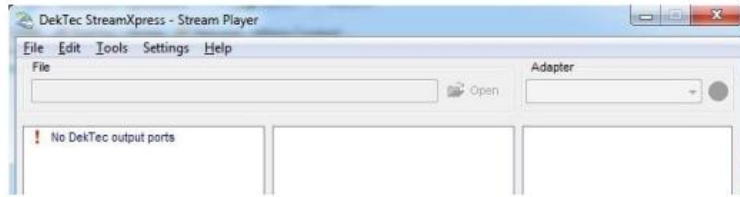
Por defecto utilizar la instalación completa.

Cuando la instalación se haya completado, está listo para comenzar a utilizar el StreamXpress DTC- 300 –SP software.

LICENCIAS DE SOFTWARE

Con el fin de operar el software StreamXpress DTC- 300 -SP, el PC debe estar conectado a un dispositivo DekTec que posea una licencia DTC- 300 -SP válida. Si no se detecta ningún producto de hardware DekTec, la Interfaz StreamXpress mostrará esto:

²⁷ Dektec. (Agosto de 2012). *DTC-300-SP StreamXpress® Software*. Recuperado el 2014, de <http://www.dektec.com/Products/Apps/DTC-300/Downloads/DTC-300-SP%20StreamXpress%20Manual.pdf>

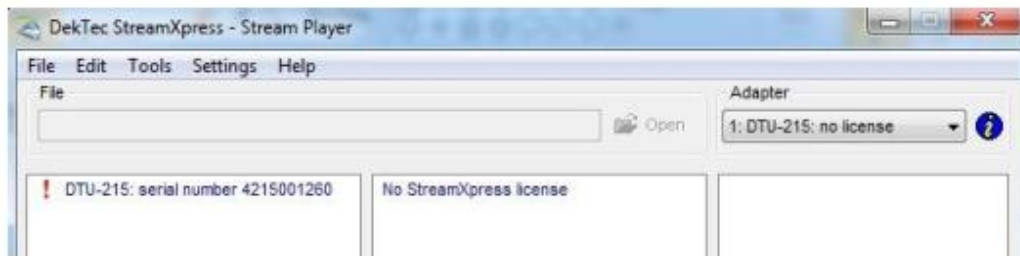


If no license is detected on a valid DekTec product, the StreamXpress interface will show this:

Figura 40: CON LICENCIA

Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

Si no se detecta ninguna licencia sobre un producto DekTec válido, la interfaz StreamXpress mostrará esto:



Use DtInfo to add a license or contact DekTec to purchase a valid StreamXpress license.

Figura 41: SIN LICENCIA

Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

Utilizar DtInfo, añadir una licencia o contacto DekTec, comprar una licencia StreamXpress válida.²⁸

2.5.1.5. AJUSTES DE MODULACIÓN ISDB -T²⁹

Si el adaptador DekTec tiene licencia para apoyar la modulación de ISDB -T (Integrated Services Digital Broadcasting -Terrestrial), el usuario puede seleccionar el sistema ISDB -T modulación y cambiar sus parámetros.

²⁸ <http://www.dektec.com/Products/Apps/DTC-300/Downloads/DTC-300-SP%20StreamXpress%20Manual.pdf>

²⁹ Dektec. (Agosto de 2012). *DTC-300-SP StreamXpress® Software*. Recuperado el 2014, de <http://www.dektec.com/Products/Apps/DTC-300/Downloads/DTC-300-SP%20StreamXpress%20Manual.pdf>

ISDB -T es el estándar utilizado para la difusión terrestre donde ISDB -T ha sido adoptado (Sobre todo Japón, Brasil y muchos otros países de América del Sur).

El formato ISDB -T es un formato de modulación muy versátil y permite hasta tres canales de transmisiones en un solo canal de RF. Los diferentes canales pueden ser móvil, HDTV y SDTV, por ejemplo. Hay 13 segmentos en una transmisión ISDB -T que se asigna de diferentes maneras dependiendo de las aplicaciones. Iseg se utiliza normalmente para la radiodifusión de televisión móvil.

Uso ISDB -T UHF 470 MHz -770 MHz, ancho de banda de 300 MHz, para repartir los 50 canales del espectro aire sobre, a saber ch.13 -ch. 62, cada canal es de 6 MHz de ancho. Estos 50 canales, ch.13 - ch.62, se les llama " canal físico.

ISDB -T modulador acepta 188 bytes MPEG- 2 TS, pero muchos de ellos también se puede aceptar el formato TMCC. El formato TMCC es un formato TS 204 de paquetes de bytes. Los primeros 188 bytes están reservados para los MPEG- 2 TS y los 16 bytes adicionales llevan la información de modulación. El StreamXpress es capaz de la transmisión de la corriente de TMCC codificada y el uso de la información de modulación para modular la MPEG -2 TS. DekTec ofrece una utilidad llamada TMCCENC.exe para crear TMCC corrientes codificadas.

Variable	Type	Descriptio	Details	Diag
ISDB-T Parameters	Display	Display the ISDB modulation information if file is TMCC encoded	#1	1
CH#	Display/ Selectio	Select the pre allocated RF output channel	#2	2
Frequency	Drop Down	Set the bandwidth of the RF channel	#3	3
Use TMCC Info	Tick Box	Will appear when file open has TMCC encoded information. Tick to use the file settings	#4	4
Params...	Button	Select this to change the ISDB-T modulation parameters	#5	5

Tabla 14: AJUSTES DE MODULACIÓN
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)



Figura 42: MODULACION
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

1 Parámetros de ISDB -T:

Si un archivo está abierto con el archivo TMCC, el StreamXpress mostrará los parámetros de modulación que contiene en el archivo. Tener en cuenta que la casilla " utilizar TMCC Info" necesita ser revisado para utilizar estos parámetros para la transmisión.

2 CH:

El StreamXpress permite la modulación a cualquier canal ágil RF utilizando el campo de la frecuencia del canal.

Para ISDB -T el plan de canales que ya se conoce y el usuario puede seleccionar un canal de RF " estándar" para playout. Hay 2 tipos de canales RF disponibles en el StreamXpress: Canal 1 a 62 representan canales de emisión de RF a través del aire. C31 a C62 representa el plan de canales para la ISDB sobre el cable.

3 Frecuencia:

El StreamXpress permite la modulación en un 6MHz o ancho de banda de 8 MHz. Tener en cuenta que el ancho de banda es 6MHz prevalente para ISDB -T.

4 Use TMCC Info:

Si el archivo abierto es un archivo TMCC y contiene parámetros de modulación ISDB -T, esta casilla de verificación se aparecerá. Si se activa, los parámetros de modulación se derivan del archivo TMCC y se sobreponen los conjuntos de parámetros.

5 Params:

Se selecciona esta opción para cambiar los parámetros de modulación ISDB -T. Los parámetros de modulación ISDB -T están organizados en 4 secciones:

- 1-MPEG-2 TS componentes de capa de asignación
- 2- Parámetros ISDB-T
- 3- Parámetros de capa
- 4- Configuraciones de centros de mensaje y preestablecidos

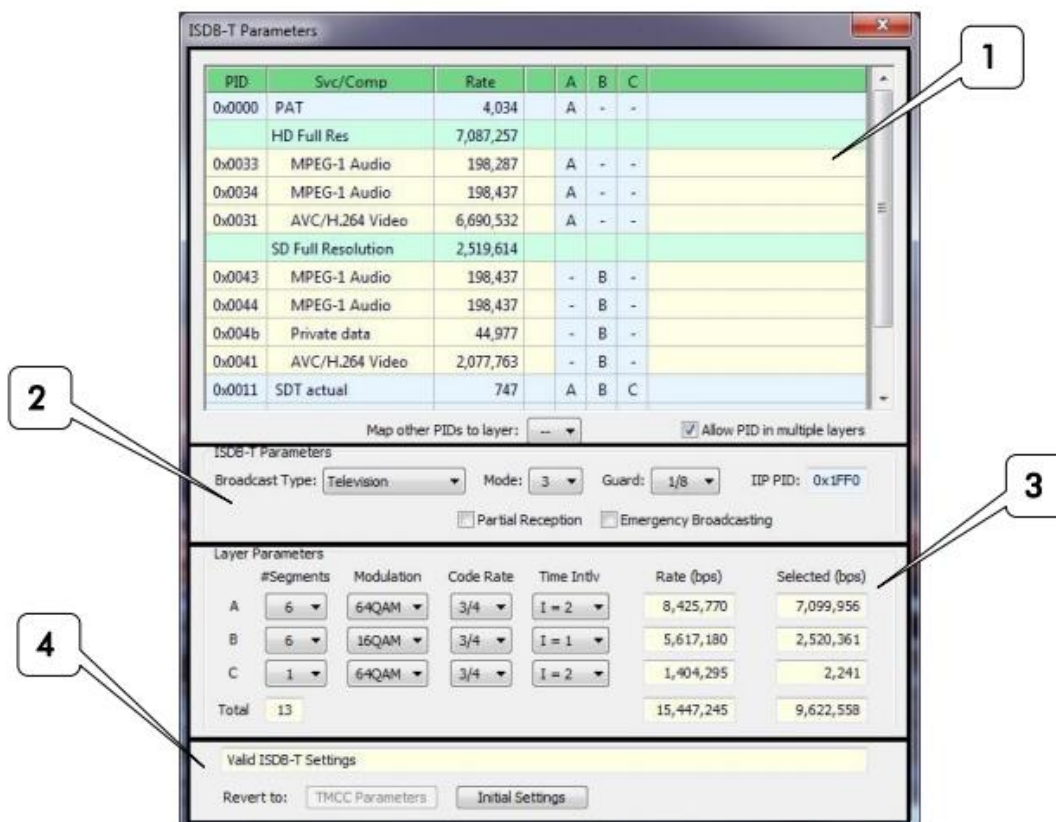


Figura 43: PARÁMETROS DE MODULACIÓN
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

COMPONENTES MPEG-2 TS DE CAPA DE ASIGNACIÓN

Esta sección permite seleccionar qué componente del flujo de transporte se realizará en el que la capa.

Normalmente, un programa completo se llevará en una sola capa.

Variable	Type	Description	Details	Diag
PID-SVC-	Display	Display information about the opened Transport Stream	#1	1
Active layer	Selection	Select what active layer to assign the PID	#2	2
Map other PIDS to Layer	Drop down	Assign other PID to a specific layer	#3	3
Allow PIDS in multiple	Tick Box	Allow to duplicate the same PID in multiple layers	#4	4

Tabla 15: COMPONENTES MPEG-2 TS DE CAPA DE ASIGNACIÓN
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

PID	Svc/Compo	Rate	A	B	C
0x0000	PAT	4,034	A	-	-
	HD Full Res	7,087,257			
0x0033	MPEG-1 Audio	198,287	A	-	-
0x0034	MPEG-1 Audio	198,437	A	-	-
0x0031	AVC/H.264 Video	6,690,532	A	-	-
	SD Full Resolution	2,519,614			
0x0043	MPEG-1 Audio	198,437	-	B	-
0x0044	MPEG-1 Audio	198,437	-	B	-
0x004b	Private data	44,977	-	B	-
0x0041	AVC/H.264 Video	2,077,763	-	B	-
0x0011	SDT actual	747	A	B	C

Figura 44: CAPAS

Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

1 PID-SVC-Bitrate:

El StreamXpress lee la información de TS desde el archivo abierto y muestra el PID, Servicio tipo de componente y el bitrate de los servicios. La ayuda en pantalla bitrate sirve para garantizar que el componente del servicio se ajuste al ancho de banda asignado a la capa.

2 Capa activa:

Para cada componente / servicio, el usuario puede seleccionar qué ISDB-T capa para asignar el componente. Seleccionar haciendo clic en la columna A, B, y C. A menos que la opción "Permitir PID en múltiples capas" está marcada, cada componente sólo puede ir en 1 capa.

3 Mapa otro PIDS a Capa:

Si un componente, no se ha asignado el usuario puede seleccionar A, B, C como una asignación de capa por defecto.

4 Dejar PID en múltiples capas:

Si está habilitado, un servicio o PID se pueden utilizar en múltiples capas.

PARAMETOS ISDB-T

En esta sección se muestran los parámetros para la modulación ISDB-T.

Variable	Type	Descriptio	Details	Diag
Broadcast Type	Drop down	Select the type of broadcast	#1	1
Mode	Drop down	Select the mode for the ISDB-T broadcast	#2	2
Guard	Drop down	Select the level of the guard interval rate	#3	3

Tabla 16: PARAMETROS ISBB-T 1
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

Variable	Type	Descriptio	Details	Diag
PID-SVC-	Display	Display information about the opened Transport Stream	#1	1
Active layer	Selection	Select what active layer to assign the PID	#2	2
Map other PIDS to Layer	Drop down	Assign other PID to a specific layer	#3	3
Allow PIDS in multiple	Tick Box	Allow to duplicate the same PID in multiple layers	#4	4

Tabla 17: PARAMETROS ISBB-T 2
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

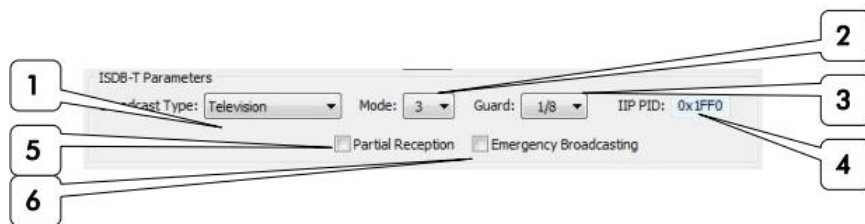


Figura 45: PARAMETROS DE TRANSMISION
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

1 Tipo de transmisión:

El ISDB -T ofrece 3 tipos de radiodifusión: televisión digital utilizando los 13 segmentos. Utilizar este menú desplegable para seleccionar el tipo de servicio deseado.

2 Modo:

El ISDB -T ofrece 3 modos para la radiodifusión: el modo 1, 2 y 3.

Modo 1 usa 108 portadoras por segmento

Modo 2 usa 216 portadoras por segmento

Modo 3 usa 432 portadoras por segmento.

Utilice este menú desplegable para seleccionar el modo deseado.

3 intervalo de guarda:

El intervalo de guarda es una sección en tiempo redundante de información que añade una copia de la última parte de un símbolo para la "entrada principal " del símbolo con el objetivo de absorber la interferencia de multipath- ondas de retraso.

El intervalo de guarda se puede ajustar a $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$.

4 PII PID:

IIP significa ISDB -T Paquete de Información. PII no se pasan al modulador. PII contienen una tabla de datos importante que se llama TMCC (Transmisión y Control de Configuración Multiplex). Más significativamente , los datos TMCC incluye los parámetros de temporización precisas requeridas para permitir al modulador sincronizarse con precisión con otros moduladores de una red de frecuencia única (SFN) y los parámetros de modulación que el modulador debe utilizar para cada capa jerárquica. Este campo permite establecer el PID que se utiliza para transferir estos datos al modulador.

5 Recepción parcial:

Este modo de transmisión se utiliza para los receptores de banda estrecha. Al limitar la gama de frecuencia intercalado dentro de un segmento en sí, es posible separar un segmento de forma independiente a partir de los restantes segmentos en la señal transmitida. De tal manera, la recepción parcial de los servicios contenida en un canal de transmisión puede ser obtenida usando un receptor de banda estrecha que tiene un ancho de banda de un segmento OFDM. Cabe señalar que la recepción parcial utiliza un segmento dedicado, que es posicionado para ser el central entre los segmentos OFDM. Si se utiliza la recepción parcial, la capa A debe tener de 1 segmento.

6 Emisión de emergencia:

Cuando se establece esta casilla permitirá una alarma de radiodifusión (1 bit) en los datos TMCC.

PARÁMETROS DE CAPAS

Esta sección permite la selección de la modulación y la asignación de segmento para cada capa.

Variable	Type	Descriptio	Details	Diag
# Segment	Drop down	Select the number of segment for each Layer (A,B and C)	#1	1
Modulation	Drop down	Select the modulation mode for each Layer (A,B and C)	#2	2
Code rate	Drop down	Select the code rate for each Layer (A,B and C)	#3	3
Time Intlv	Drop down	Select the time interleaver for each Layer (A,B and C)	#4	4
Rate (bps)	Display	Displays the total available bitrate for each Layer (A,B and C)	#5	5
Selected (Bps)	Display	Displays the total selected component bitrate for each Layer (A,B	#6	6
Total	Display	Display the total number of segments: (A+B+C)	#7	7

Tabla 18: PARÁMETROS DE CAPAS

Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

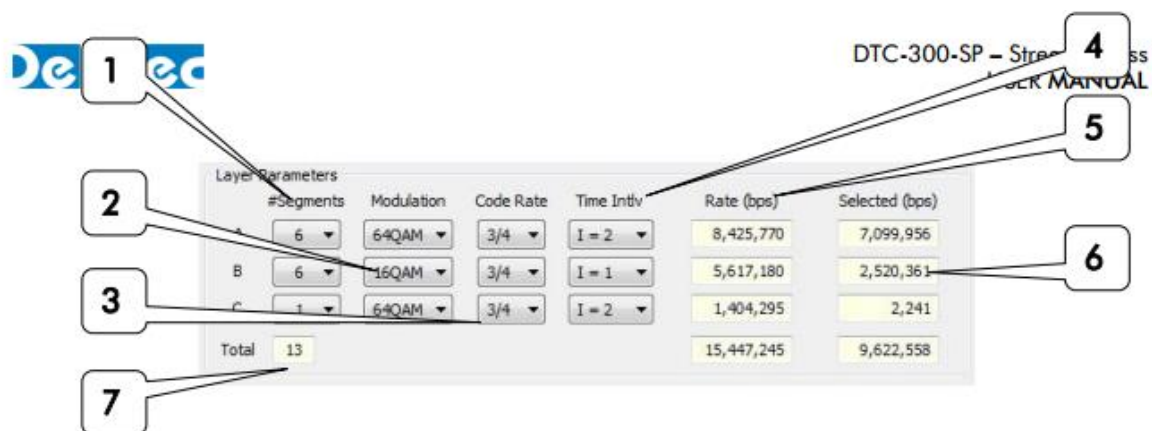


Figura 46: PARAMETROS DE CAPAS

Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

1 Número de segmentos:

El segmento ISDB -T cuenta con 13 segmentos. Estos segmentos se pueden asignar a cualquiera de las 3 capas si se transmite en el modo normal. Esta caída hacia abajo permite la asignación de segmento para cada capa A, B , y la capa C .

2 de la modulación:

Dependiendo de la aplicación, una capa puede utilizar diferente de modulación. Los modos de modulación disponibles son DQPSK, QPSK, 16QAM y 64QAM. Cada capa se puede usar una modulación diferente.

3 Tasa Código:

Dependiendo de la aplicación, una capa puede usar diferente tipo de código. Los tipos de código disponibles son de $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, $\frac{5}{6}$, $\frac{7}{8}$. Cada capa puede utilizar una velocidad de código diferente.

4 Tiempo Intlv:

Intercalador tiempo es única para ISDB -T y es muy eficaz para mejorar tanto la robustez frente a ruido de impulso y el rendimiento para la recepción móvil / portátil. El intercalador de tiempo depende del

Modo de emisión de ISDB -T.

Para el modo 1, los valores son 0, 4, 8,16.

Para el modo 2, los ajustes son 0, 2, 4,8.

Para el modo 3, los ajustes son 0, 1, 2,4.

NOTA: Se recomienda el uso de un valor medio.

5 velocidad en bps:

La tasa de transporte total disponible para la emisión en una capa depende del número de segmentos asignado, parámetros modulaciones etc .Este campo muestra el total de la tasa de bits disponible para la capa.

6 Bps seleccionada:

Esta pantalla representa la tasa de bits total para todos los componentes del TS seleccionados en la sección de componentes.

NOTA: La velocidad de bits total del componente seleccionado debe ser inferior a la velocidad del canal para cada segmento. Si el tipo de componente seleccionado es mayor que la velocidad del canal, esta pantalla se volverá roja y se generará un error.

7 Total:

Esta pantalla representa el número total de segmentos utilizados para la transmisión. Se añade a todos los segmentos seleccionados para la capa A, B y C.

CONFIGURACIÓN PREESTABLECIDA Y CENTRO DE MENSAJES

En esta sección muestra un error y permite al usuario seleccionar la configuración preestablecida.

Variable	Type	Descriptio	Details	Diag
Message Center	Display Field	Displays is the configuration is valid and point to any error		1
TMCC Parameters	Drop down	Load the TMCC parameters present in the 204 byte packet TS file (If present)		2
Initial settings	Drop down	Load the original settings of the modulation		3

Tabla 19: CENTRO DE MENSAJES
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)



Figura 47: ERROR ASIGNACIÓN BITRATE
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

Una vez configurados todos los parámetro de modulación ya se puede transmitir seleccionando play en la sección de playout.



Figura 48: TRANSMISIÓN
Fuente: (Dektec, DTC-300-SP StreamXpress® Software, 2012)

2.5.2. ETAPA DE RECEPCIÓN DE UN SERVICIO DE TV DIGITAL CON INTERACTIVIDAD.

Para esta etapa de la implementación se utilizan dos antenas de UHF, un set-top box de la marca MUNDY HOME que decodifica en formato ISDB-Tb y una televisión con entradas de audio y video.

2.5.2.1. ANTENAS UHF³⁰

- La antena de UHF para la tarjeta moduladora que se emplea tiene las siguientes características.



Figura 49: TARJETA DEKTEC CON ANTENA UHF 75 Ω
Fuente: El Autor

³⁰ Alibaba. (s.f.). *uhf pasiva antena de varilla para la tv digital para el uso en casa*. Recuperado el 2014, de <http://spanish.alibaba.com/product-free/uhf-passive-rod-antenna-for-digital-tv-for-home-application-112064581.html>



Figura 50: ANTENA PASIVA DE VARILLA PARA TDT
Fuente: El Autor

La recepción de rango de frecuencia: 470-860mhz uhf pasiva antena de varilla

1. recibir rango de frecuencia: uhf 470-860mhz
2. impedancia: ohm 75 typ.
3. de aseguramiento de la operación rango de temperatura: -20 ~ +60 grados
4. de almacenamiento de garantía de rango de temperatura: -40 ~ +85 grados
6. coaxial longitud: 1m
7. coaxial cable: f/iec/sma/mcx terminal
8. dimensión: 29x29x139mm

- **la antena de UHF para el set-top box a emplearse cumple con las características necesarias para la recepción en la Banda de 6MHz de canales digitales, específicamente en frecuencias de 670 a 672 MHz .**



Figura 51: ANTENA MICROSTRIP UHF
Fuente: El Autor

2.5.2.2. EL CONECTOR UTILIZADO PARA ESTA ANTENA

Para conectar la antena microstrip al decodificador se emplea un conector BNC-RF el cual adapta las impedancias de la antena y el decodificador a 75 ohm.



Figura 52: ELEMENTOS DEL ACOPLE ANTENA UHF-SET-TOP BOX
Fuente: Autor

2.5.2.3. RECEPCION EN EL SET-TOP BOX

Para recibir la señal digital se debe realizar una programación automática la cual se encargara de detectar el canal transmitido en este caso CANAL UNACH. Este canal ya viene configurado con una resolución y aspecto específico, pero también se tiene la opción para configurar estos parámetros.

Este decodificador no cuenta con el middleware Ginga por lo que solo recibe el audio y video en formato ISDB-Tb



Figura 53: SET-TOP BOX
Fuente: El Autor

CAPITULO III

3. RESULTADOS

Las pruebas realizadas se efectúan en el proceso de desarrollo del laboratorio, y al final de su desarrollo, para esto se utiliza varios hardware y software que permiten verificar el funcionamiento de la interactividad y de la señal digital en formato ISDB-Tb.

3.1. PRUEBAS DE ARCHIVOS TS Y DE LA BTS MULTIPLEXADA ANTES DE SER MODULADA.

Para efectuar estas pruebas se utiliza el software media player classic el cual es capaz de reproducir archivos con extensión .ts y por tanto puede mostrar cómo se está efectuando la decodificación de la señal en parámetros de: aspecto, resolución, sincronismo de la señal, tamaño, duración, fotogramas o cuadros por segundo y formato de audio y video que es lo que interesa conocer como resultado final.

3.2. PRUEBAS DE APLICACIONES INTERACTIVAS

Gracias al emulador Gingga4windows se puede ver el resultado del desarrollo de aplicaciones interactivas en eclipse, este emulador cuenta con todo lo necesario para realizar estas pruebas.

Para simular el control remoto del set top box, se utiliza el teclado del PC de la siguiente manera:

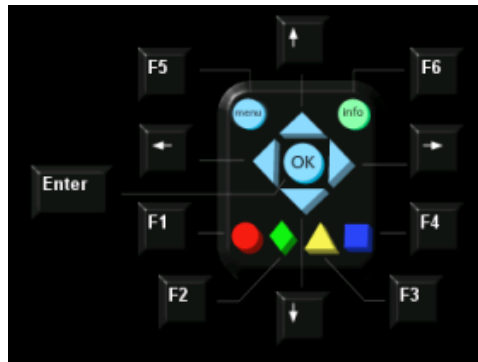


Figura 54: CONTROL SET-TOP BOX VIRTUAL
Fuente: (Lopez, Carrasco, & Fuentes, 2010)



Figura 55: APLICACIÓN INTERACTIVA GINGA NCL
Fuente: El Autor

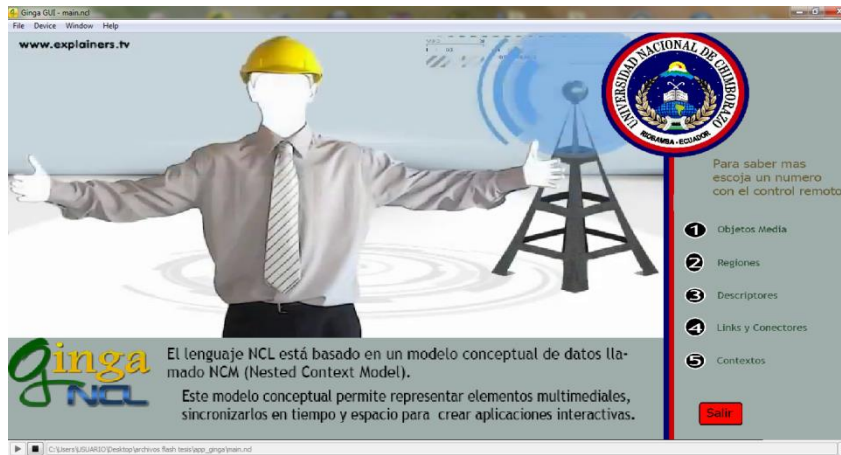


Figura 56: APLICACIÓN INTERACTIVA GINGA NCL
Fuente: El Autor



Figura 57: APLICACIÓN INTERACTIVA GINGA NCL
Fuente: El Autor



Figura 58: APLICACIÓN INTERACTIVA GINGA NCL
Fuente: El Autor

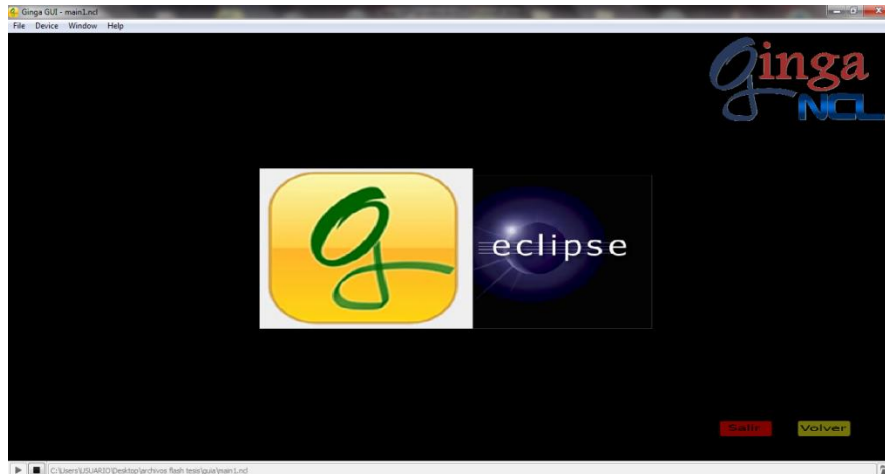


Figura 59: APLICACIÓN INTERACTIVA GINGA NCL
Fuente: El Autor

3.3. ANALISIS DE LA SEÑAL DIGITAL MEDIANTE EL ANALIZADOR DE ESPECTROS.

Para analizar la señal digital recibida se emplea un analizador de espectros, este muestra las características espectrales como son: nivel de potencia, frecuencia, ancho de banda que es lo que interesa para evaluar el funcionamiento del modulador Dektec DTA-2111.



Figura 60: ANALISIS ESPECTRAL
Fuente: El Autor

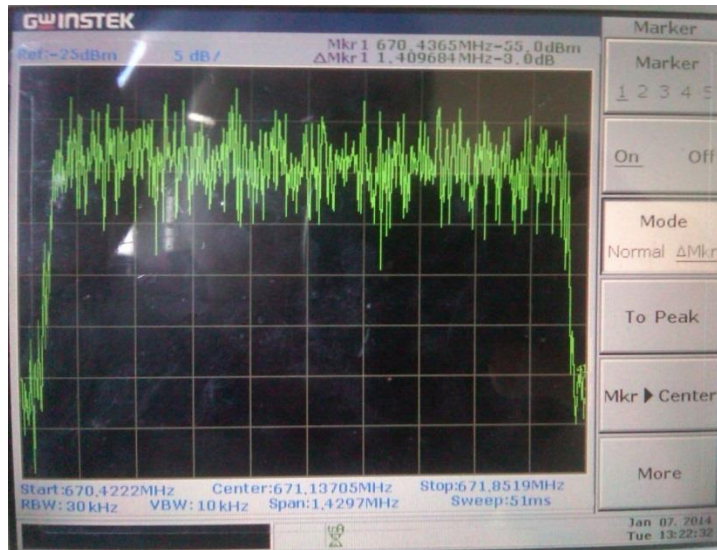


Figura 61: ESPECTRO DE LA PORTADORA DE LA SEÑAL DIGITAL TRANSMITIDA
Fuente: El Autor

3.4. RECEPCION EN EL SET-TOP BOX DE LA SEÑAL DECODIFICADA EN FORMATO ISDB-Tb.



Figura 62: RECEPCIÓN DIGITAL ISDB-TB
Fuente: El Autor



Figura 63: RECEPCIÓN DIGITAL ISDB-TB
Fuente: El Autor

4. DISCUSIÓN

El motivo que incentivó el desarrollo del proyecto radica en la reciente adopción del estándar brasileño de Televisión Digital que será implementado en un futuro próximo en nuestro país, por lo cual es de gran importancia conocer cómo trabaja este estándar y los beneficios que ofrece. Por otro lado, también es necesario analizar los elementos que intervienen tanto en el desarrollo de contenidos interactivos, una de las mayores atracciones de TV Digital, como en la generación del flujo de transporte a transmitir y el equipamiento que se requiere para poder visualizarlos a nivel de usuario.

Además actualmente la Facultad de Ingeniería Electrónica y telecomunicaciones no dispone de un laboratorio de este tipo, que servirá como un complemento para el estudio de Televisión Digital en la materia de Optativa.

La implementación de un laboratorio de pruebas de tv digital terrestre ISDB-TB, para la emisión de aplicaciones interactivas posibilitará una vinculación directa de los estudiantes con la digitalización de la comunicación lo cual será beneficioso para los mismos en un futuro con respecto a esta tecnología que está próxima a implementarse en nuestro país y aumentará sus posibilidades de inserción laboral en esta área que posibilitará obtener el nivel más elevado posible de interactividad y el desarrollo de nuevas aplicaciones que brinden entretenimiento, educación, cultura y además contribuyan para la formación de una sociedad capaz de enfrentar los desafíos actuales, donde la información y el conocimiento son cada vez más impredecibles para el progreso económico y el bienestar social.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Mediante la implementación del laboratorio se logra analizar en forma práctica los tipos de codificación, paquetización, multiplexación y modulación, además analizar la señal digital en su transmisión y recepción con una aplicación interactiva que desarrolla también conocimientos de programación.
- Mediante la implementación del laboratorio se logra analizar en forma práctica los tipos de codificación, paquetización, multiplexación y modulación, además analizar la señal digital en su transmisión y recepción con una aplicación interactiva que desarrolla también conocimientos de programación.
- Para este proyecto se utilizó software libre, como los principales están Ffmpeg y OpenCaster con lo que se consiguió minimizar costos con respecto a la codificación y multiplexación, esto se lo puede demostrar poniendo como ejemplo al codificador ENVIVIO 4CASTER M2 MOBILE TV VIDEO SDI IN H.263 H.264 MPEG4 MPEG-4 ENCODER con un costo de US \$1,190.00 y el modulador MX-1500 MPEG-2-TS Multiplexer For Terrestrial Digital Broadcasting con un costo aproximado de US \$20,000.00. Además se empleó la tarjeta moduladora Dektec DTA-2111 que con la opción de modulación ISDB-T tiene un costo aproximado de 1705 € en comparación a otros equipos moduladores del mercado como por ejemplo el MO-370 LE Modulador ISDB-T/Tb con un costo de 5000 €.
- El sistema operativo Linux Ubuntu tiene la ventaja de ser un software libre y orientado al usuario con lo que mediante el conocimiento y empleo de comandos básicos y configuraciones se lo puede utilizar como un servidor que interactúe a su vez con otro de su mismo tipo o diferente como en este caso Windows complementándose como un solo bloque de desarrollo de TDT.
- La programación en lenguaje Glinga NCL tiene una estructura que consta de objetos multimedia que se sincronizan en base a programación de declaraciones que especifican un conjunto de afirmaciones, condiciones, restricciones y proporciones.

5.2. Recomendaciones

- Este laboratorio de pruebas constituye una solución escalable ya que su estudio puede ser ampliado en un futuro.
- Un aspecto muy importante para futuros estudios de desarrollo del tema planteado es contar con las instalaciones y el espacio físico adecuado para el funcionamiento del laboratorio de TDT, ya que en este momento se encuentra ubicado en un área improvisada donde se desarrollan otras actividades que pueden perjudicar la implementación.
- Ya que este laboratorio se basa en archivos creados temporalmente la transmisión no se la realiza en tiempo real como la de un canal de TDT profesional, por lo que se recomienda el estudio de este factor para mejorar este estudio, una alternativa sería emplear el reproductor VLC ya que es un software libre y cuenta con esta opción por lo que podría ser una alternativa viable.
- Para la realización de pruebas más completas es recomendable la adquisición de un set-top box de desarrollo que cuente con el middleware Ginga ya que actualmente el factor económico del tesista es una limitación para su empleo.
- Se pueden realizar estudios técnicos que permitan ampliar el área de cobertura de la señal de televisión digital, lo que ayudara al desarrollo de la universidad, como lo conseguido por la UCSG de Guayaquil con su canal analógico.
- Sería de gran ayuda poder contar con una interfaz gráfica para el manejo de Open Caster, lo que facilitaría al usuario su uso, por lo que convendría contar con el apoyo del área de Ingeniería en Sistemas de la UNACH.

6. PROPUESTA

6.1. Título de la propuesta

Implementación de un Laboratorio de pruebas de TV DIGITAL TERRESTRE ISDB-tb, para la emisión de aplicaciones interactivas para la escuela de ingeniería electrónica y telecomunicaciones de la unach.

6.2. Introducción

Previamente se analiza el problema a resolver, es así como se escoge el no disponer de un laboratorio que sirva como complemento de aprendizaje en el área de las telecomunicaciones digitales, su causante son múltiples factores, dentro de los cuales se destaca el bajo desarrollo tecnológico de esta área en nuestro país.

Hasta el día de hoy las herramientas con las que cuenta la universidad para el estudio de las telecomunicaciones están directamente relacionadas a la práctica con señales analógicas mediante equipos como el analizador de espectro, osciloscopios, generadores, transmisores y receptores de señal analógica, por lo que el estudio de la señal digital se basa solo en la parte teórica, lo que constituye una falencia muy grande al momento de trabajar en esta área, donde se relaciona directamente con la práctica de las telecomunicaciones.

La solución planteada representa una alternativa para suplir la falta de equipos digitales con los cuales poder realizar prácticas de laboratorio, poner en práctica la teoría, además de complementar el aprendizaje en el área digital con programación interactiva que es una nueva área en las Telecomunicaciones que es indispensable para una Ingeniería para poder irrumpir en esta área.

Con la práctica de esta tecnología los estudiantes tendrán más oportunidades de incursionar en ámbito laboral teniendo mayor confianza en sus conocimientos y poniéndolos en práctica ya que en el Ecuador el apagón analógico es próximo a implementarse.

6.3. OBJETIVOS

OBJETIVO GENERAL

Implementar un laboratorio de pruebas de TV DIGITAL TERRESTRE ISDB-Tb, para la emisión de aplicaciones interactivas para la Escuela de Ingeniería Electrónica y Telecomunicaciones de la UNACH.

OBJETIVOS ESPECÍFICOS

- Determinar mediante un estudio de factibilidad los distintos tipos de software libre y hardware (tarjetas moduladoras y sintonizadoras en estándar ISDB-Tb) más apropiadas para este proyecto.
- Implementar una red de tv digital terrestre isdb-tb con equipos y dispositivos de bajo costo, con software libre que permitan investigar y conseguir resultados en lo referente a los parámetros de interactividad.
- Desarrollar un manual de prácticas de laboratorio en relación al funcionamiento y estructura de la red de tv digital terrestre ISDB-Tb.
- Desarrollar aplicaciones interactivas mediante un manual de guía interactivo para poder realizarlas.

6.4. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA

La investigación realizada se ha completado con el uso de las siguientes tecnologías.

6.4.1. GINGA³¹

Ginga es la tecnología desarrollada por Brasil y que consta de sistemas y sub-sistemas relacionados entre si para desarrollar y presentar interactividad. Consta de dos sub-sistemas, Ginga J (aplicaciones Java) y Ginga NCL(presentación multimedia). La programación esta basada en aplicaciones NCL y LUA las cuales.

³¹ WIKIPEDIA. (22 de Julio de 2013). *Ginga*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/Ginga>

6.4.2. Eclipse – GINGA NCL ³²

Eclipse es el software de Windows compatible con el lenguaje GINGA NCL y LUA por medio de la instalación de los plugins correspondientes, cuenta con un entorno de desarrollo fácil de manipular y cuenta con todas las características necesarias que son compatibles con GINGA.

6.4.3. OPENCASER³³

OpenCaster desarrollado por AVALPA, es un software libre para generar TS en formato MPEG-2. Este software genera BTS con características del estándar DVB-T y no tiene soporte para el formato ISDB-Tb.

LIFIA que pertenece a la Universidad Nacional de la Plata en Argentina modificó OpenCaster para agregarle características específicas para el formato ISDB-Tb las cuales están incompletas, pero son suficientes para poder multiplexar paquetes TS y obtener una BTS compatible con la norma.

6.4.4. Dta-2111³⁴

Modulador multi estándar VHF/UHF. Sus características son:

- modulador multi-estándar para PCI Express con soporte para la mayoría de los estándares de QAM-, y modulación OFDM basado en VSB.
- conversión digital para una excelente calidad de señal sin necesidad de calibración.
- Soporta todas las constelaciones y los modos de modulación para cada estándar soportado.
- Todos los Canales 36 .. 1002MHz totalmente ágil sobre la banda VHF y UHF.

³² Americana, R. I. (2011). *Recursos*. Recuperado el 2014, de http://www.rediberoamericanatdt.org/?page_id=677

³³ Venegas Picón, L. A. (22 de Mayo de 2012). *GENERACIÓN DE UNA TRAMA BROADCAST TRANSPORT STREAM (BTS) USANDO EL SOFTWARE LIBRE OPENCASER*. Recuperado el 2014, de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1393/VENEGAS_PICON_LUIS_BTS_OPENCASER.pdf?sequence=1

³⁴ Dektec. (2011). *DTA-2111 Multi-Standard VHF/UHF Modulator for PCI Express*. Recuperado el 2014, de <http://www.dektec.com/Products/PCIe/DTA-2111/>

- opción simulador de canal digital.
- Salida de RF para la conexión directa a la entrada de la antena de un receptor digital.

6.5. DISEÑO ORGANIZACIONAL.



6.6. MONITOREO Y EVALUACIÓN DE LA PROPUESTA

El monitoreo de la implementación deberá estar a cargo del profesor que imparta la materia de Optativa el mismo que administrara y analizará posibles cambios.

La evaluación estará a cargo del tutor ya que sabe todo respecto a esta implementación.

7. Bibliografía

- Galabay Toalongo, P. T., & Vivar Espinoza, F. R. (Junio de 2012). *Manejo del software Ginga para el desarrollo de aplicaciones interactivas para televisión digital, basado en el estándar Brasileño ISDB-Tb*. Recuperado el 2014, de <http://dspace.ups.edu.ec/bitstream/123456789/2228/13/UPS-CT002411.pdf>
- Alibaba. (s.f.). *uhf pasiva antena de varilla para la tv digital para el uso en casa*. Recuperado el 2014, de <http://spanish.alibaba.com/product-free/uhf-passive-rod-antenna-for-digital-tv-for-home-application-112064581.html>
- Always, U. (11 de Diciembre de 2012). *Como usar apt-get y el resto de sistema por Proxy*. Recuperado el 2014, de <http://ubuntualways.wordpress.com/2012/12/11/como-usar-apt-get-y-el-resto-de-sistema-por-proxy/>
- Americana, R. I. (2011). *Recursos*. Recuperado el 2014, de http://www.rediberoamericanatdt.org/?page_id=677
- AVALPa. (1 de Marzo de 2011). *Avalpa Broadcast Server*. Recuperado el 2014, de <http://www.avalpa.com/assets/freesoft/opencaster/AvalpaBroadcastServerUserManual-v2.1.pdf>
- Bernal , I., & Mejía , D. (2013). *Herramientas de Desarrollo para TV Digital Orientadas al estándar ISDB-Tb*. Recuperado el 2014, de <http://ticec.cedia.org.ec/dmdocuments/Documentacion/Dia2/ponenci6.ppt>
- Brasileiro, P. d. (s.f.). *Ginga*. Recuperado el 2014, de http://www.softwarepublico.gov.br/ver-comunidade?community_id=1101545
- COMMONS, O. (s.f.). *Ubuntu Hacks/Multimedia*. Recuperado el 2014, de http://commons.oreilly.com/wiki/index.php/Ubuntu_Hacks/Multimedia
- Dektec. (Mayo de 2011). *DTA-2111 Multi-Standard VHF/UHF Modulator*. Recuperado el 2014, de <http://www.dektec.com/Products/PCle/DTA-2111/Downloads/DTA-2111%20Leaflet.pdf>
- Dektec. (2011). *DTA-2111 Multi-Standard VHF/UHF Modulator for PCI Express*. Recuperado el 2014, de <http://www.dektec.com/Products/PCle/DTA-2111/>
- Dektec. (Agosto de 2012). *DTC-300-SP StreamXpress® Software*. Recuperado el 2014, de <http://www.dektec.com/Products/Apps/DTC-300/Downloads/DTC-300-SP%20StreamXpress%20Manual.pdf>
- Dektec. (2013). *DTC-300 StreamXpress*. Recuperado el 2014, de <http://www.dektec.com/Products/Apps/DTC-300/>

- DesdeLinux. (s.f.). *Instalación y configuración de SAMBA en Ubuntu 12.04*. Recuperado el 2014, de <http://blog.desdelinux.net/instalacion-y-configuracion-de-samba-en-ubuntu-12-04/>
- García, F. (s.f.). *DISEÑO DE ANTENA MICROSTRIP PARA LA BANDA DE UHF*. Recuperado el 2014, de <http://www.docentes.unal.edu.co/jcgarciaa/docs/Papers/ELE018.pdf>
- Gomes Soares, L. F. (s.f.). *FORO DEL SISTEMA BRASILEÑO*. Recuperado el 2014, de http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_04_soares.pdf
- González, D. (25 de Agosto de 2013). *[Tip] como cambiar el servidor DNS y la dirección IP en Ubuntu 12.04 / 12.10*. Recuperado el 2014, de <http://planetubuntu.es/post/tip-como-cambiar-el-servidor-dns-y-la-direccion-ip-en-ubuntu-12-04-12-10>
- Gutiérrez Tapia, A. G. (8 de Abril de 2013). *DISEÑO DE UN LABORATORIO DE TELEVISIÓN DIGITAL PARA LA TRANSMISIÓN DE SEÑALES CON MULTIPROGRAMACIÓN, CONTENIDOS INTERACTIVOS Y GUÍA ELECTRÓNICA DE PROGRAMACIÓN (EPG)*. Recuperado el 2014, de <http://dspace.ucuenca.edu.ec/bitstream/123456789/372/1/tesis.pdf>
- Illescas Robalino, M. A., & Villamarín Zapata, D. F. (2011). *IMPLEMENTACIÓN DE UN TRANSMISOR DE PRUEBAS DE TV DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE APLICACIONES INTERACTIVAS*. Recuperado el 2014, de <http://repositorio.espe.edu.ec/bitstream/21000/4289/1/T-ESPE-032586.pdf>
- LIFIA. (13 de Septiembre de 2010). *OpenCaster para SATVD*. Recuperado el 2014, de <http://wiki.ginga.org.ar/lib/exe/fetch.php?media=lifia:guiaopencaster.pdf>
- LIFIA. (3 de Mayo de 2011). *OpenCaster para SATVD-T*. Recuperado el 2014, de <http://wiki.ginga.org.ar/lib/exe/fetch.php?media=lifia:guiaopencaster2.pdf>
- Lopez, F., Carrasco, C., & Fuentes, L. (2010). *Ginga*. Recuperado el 2014, de <http://profesores.elo.utfsm.cl/~agv/elo323/2s10/projects/LopezCarrascoFuentes/Uso-Ginga.html>
- Moncada Salazar, C. (22 de Octubre de 2013). *Como Instalar Ubuntu 12.04*. Recuperado el 2014, de <http://carlosmoncadasalazar.blogspot.com/2013/10/como-instalar-ubuntu-12.html>
- Muñoz Fernández, F., & Ansorena González, P. (28 de Octubre de 2009). *IMPLEMENTACIÓN DE UNA SOLUCIÓN PARA LA GESTIÓN DE UN CANAL DE TELEVISIÓN DIGITAL TERRESTRE*. Recuperado el 2014, de <http://academica-e.unavarra.es/bitstream/handle/2454/1791/577020.pdf?sequence=1>
- NBR, A. (19 de Noviembre de 2009). *NORMA BRASILEÑA*. Recuperado el 2014, de [http://www.advicom.ec/userFiles/files/Normas/ABNTNBR%2015606-2_2007Esp_2008Vc_2009-Codificacion%20Video,Audio%20y%20Multiplexacion%20\(Parte%20\).pdf](http://www.advicom.ec/userFiles/files/Normas/ABNTNBR%2015606-2_2007Esp_2008Vc_2009-Codificacion%20Video,Audio%20y%20Multiplexacion%20(Parte%20).pdf)

- Olmedo , G. (Noviembre de 2012). *Sistema de Transmisión del Estándar ISDB-Tb*. Recuperado el 2014, de https://docs.google.com/file/d/0B3Hp29uy3q7_Y1pXX1pKNC1CRU0/edit?pli=1
- Peddicord, J. (22 de Abril de 2008). *FOSSwire*. Recuperado el 2014, de <http://fosswire.com/post/2008/04/ubuntu-cheat-sheet/>
- Pionner. (2014). *Decodificador Digital*. Recuperado el 2014, de https://www.laesquinapioneer.ec/3-135-decodificador_digital.html%E2%80%8E
- Pisciotta, N. O. (s.f.). *Sistema ISDB-Tb (Primera parte)*. Recuperado el 2014, de <http://www.eradigital.com.ar/blog/wp-content/uploads/2010/09/Sistema-ISDB-Tb-Primera-parte.pdf>
- Sandoval, F. (s.f.). *ISDB-T e ISDB-Tb*. Recuperado el 2014, de <http://es.scribd.com/doc/104745307/7isdb-110907154157-phpapp01>
- Systems, L. o. (2009). *NCL Eclipse*. Recuperado el 2014, de http://laws.deinf.ufma.br/ncleclipse/pt-br:start?redirect=1#.Us8G6tIW3_G
- Ubuntu, G. (4 de Noviembre de 2007). *Tovid*. Recuperado el 2014, de <http://www.guia-ubuntu.com/index.php?title=Tovid>
- Ubuntu, G. (s.f.). *Ffmpeg*. Recuperado el 2014, de <http://www.guia-ubuntu.com/index.php/Ffmpeg#2281>
- Venegas Picón, L. A. (22 de Mayo de 2012). *GENERACIÓN DE UNA TRAMA BROADCAST TRANSPORT STREAM (BTS) USANDO EL SOFTWARE LIBRE OPENCASTER*. Recuperado el 2014, de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1393/VENEGAS_PICO_N_LUIS_BTS_OPENCASTER.pdf?sequence=1
- WIKIPEDIA. (6 de Octubre de 2013). *ATSC*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/ATSC>
- WIKIPEDIA. (2 de Noviembre de 2013). *DVB-T*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/DVB-T>
- WIKIPEDIA. (22 de Julio de 2013). *Ginga*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/Ginga>
- WIKIPEDIA. (30 de Diciembre de 2013). *ISDB-T*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/ISDB-T>
- WIKIPEDIA. (24 de Septiembre de 2013). *Media Player Classic*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Media_Player_Classic
- WIKIPEDIA. (12 de Diciembre de 2013). *Samba (programa)*. Recuperado el 2014, de [http://es.wikipedia.org/wiki/Samba_\(programa\)](http://es.wikipedia.org/wiki/Samba_(programa))

WIKIPEDIA. (1 de Diciembre de 2013). *Set-top box*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Set-top_box

WIKIPEDIA. (5 de Enero de 2014). *DTMB*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/DTMB>

WIKIPEDIA. (3 de Enero de 2014). *Televisión digital terrestre*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/TDT>

WIKIPEDIA. (7 de Enero de 2014). *Televisión Digital Terrestre en Ecuador*. Recuperado el 2014, de http://es.wikipedia.org/wiki/Televisi%C3%B3n_Digital_Terrestre_en_Ecuador

WIKIPEDIA. (4 de Enero de 2014). *Ubuntu*. Recuperado el 2014, de <http://es.wikipedia.org/wiki/Ubuntu>

8. APÉNDICES Y ANEXOS

ANEXO 1

Programación en Python del archivo gtables1.py para la configuración de OpenCaster.

```
#!/usr/bin/python
# coding: utf-8

import os
from dvbobjects.PSI.PAT import *
from dvbobjects.PSI.NIT import *
from dvbobjects.PSI.SDT import *
from dvbobjects.PSI.PMT import *
from dvbobjects.MHP.AIT import *
from dvbobjects.MHP.Descriptors import *
from dvbobjects.SBTVD.Descriptors import *

id_ts_unach      = 0x073b # ID de red.
id_red_unach     = 0x073b # ID de red original.
frecuencia       = 671    # Frecuencia de transmisión
id_control_remoto = 0x05   # Tecla de control remoto.
id_servicio      = 0xe760 # ID de servicio de TV Digital.
pid_pmt_servicio = 1031   # PID de la PMT del servicio.

                                #DEFINICION DE LA RED: NIT

nit = network_information_section(
network_id = id_red_unach,
network_descriptor_loop = [
network_descriptor(network_name = "CANAL UNACH",),
system_management_descriptor(
    broadcasting_flag = 0,
    broadcasting_identifier = 3,
    additional_broadcasting_identification = 0x01,
    additional_identification_bytes = [],
)
],
transport_stream_loop = [
    transport_stream_loop_item(
        transport_stream_id = id_ts_unach,
        original_network_id = id_red_unach,
        transport_descriptor_loop = [
            service_list_descriptor(
                dvb_service_descriptor_loop = [
                    service_descriptor_loop_item (
```

```

        service_ID = id_servicio,
        service_type = 1,
    ),
],
),
terrestrial_delivery_system_descriptor(
    area_code = 1341,
    guard_interval = 0x01,
    transmission_mode = 0x02,
    frequencies = [
        tds_frequency_item( freq=frecuencia )
    ],
),
partial_reception_descriptor (
    service_ids = []
),
transport_stream_information_descriptor (
    remote_control_key_id = id_control_remoto,
    ts_name = "CANAL UNACH",
    transmission_type_loop = [
        transmission_type_loop_item(
            transmission_type_info = 0x0F,
            service_id_loop = [
                service_id_loop_item(
                    service_id=id_servicio
                ),
            ]
        ),
        transmission_type_loop_item(
            transmission_type_info = 0xAF,
            service_id_loop = [],
        ),
    ],
)
],
),
),
],
),
version_number = 0,
section_number = 0,
last_section_number = 0,
)

```

DEFINICION DE LOS SERVICIOS: SDT

```

sdt = service_description_section(
    transport_stream_id = id_ts_unach,
    original_network_id = id_red_unach,
    service_loop = [
        service_loop_item(
            service_ID = id_servicio,
            EIT_schedule_flag = 0,
            EIT_present_following_flag = 0,
            running_status = 4,
            free_CA_mode = 0,
        )
    ]
)

```

```

        service_descriptor_loop = [
            service_descriptor(
                service_type = 1,
                service_provider_name = "UNACH",
                service_name = "CANAL UNACH",
            ),
        ],
    ),
],
version_number = 0,
section_number = 0,
last_section_number = 0,
)

        # DEFINICION DEL MAPA DE PROGRAMAS: PAT

pat = program_association_section(
    transport_stream_id = id_ts_unach,
    program_loop = [
        program_loop_item(
            program_number = 0,
            PID = 16,
        ),
        program_loop_item(
            program_number = id_servicio,
            PID = pid_pmt_servicio,
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)

        # DEFINICION DE LA TABLA: PMT

pmt_sd = program_map_section(
    program_number = id_servicio,
    PCR_PID = 2065,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2,
            elementary_PID = 2065,
            element_info_descriptor_loop = [
            ]
        ),
        stream_loop_item(
            stream_type = 3,
            elementary_PID = 2075,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5,
            elementary_PID = 2001,
            element_info_descriptor_loop = [
                data_component_descriptor (

```

```

        data_component_id = 0xA3,
        additional_data_component_info = ait_identifier_info(
            application_type = GINGA_NCL_application_type,
            ait_version = 0
        ).bytes(),
    ),
    application_signalling_descriptor(
        application_type = 9,
        AIT_version = 1,
    ),
]
),
stream_loop_item(
    stream_type = 0x0B,
    elementary_PID = 2004,
    element_info_descriptor_loop = [
        association_tag_descriptor(
            association_tag = 0x0C,
            use = 0,
            selector_lenght = 0,
            transaction_id = 0x80000000,
            timeout = 0xFFFFFFFF,
            private_data = "",
        ),
        stream_identifier_descriptor(
            component_tag = 0x0C,
        ),
        carousel_identifier_descriptor(
            carousel_ID = 2,
            format_ID = 0,
            private_data = "",
        ),
        data_component_descriptor (
            data_component_id = 0xA0,
            additional_data_component_info = additional_ginga_j_info(
                transmission_format = 0x2,
                document_resolution = 0x5,
                organization_id = 0x0000000A,
                application_id = 0x0064,
                carousel_id = 2,
            ).bytes(),
        ),
    ]
)
],
version_number = 0,
section_number = 0,
last_section_number = 0,
)

```

*****APLICACIÓN GINGA*****

```
ait = application_information_section(  
    application_type = 0x0009,  
    common_descriptor_loop = [],  
    application_loop = [  
        application_loop_item(  
            organisation_id = 0x0000000A,  
            application_id = 0x64,  
            application_control_code = 0x01,  
            application_descriptors_loop = [  
                transport_protocol_descriptor(  
                    protocol_id = 0x0001,  
                    transport_protocol_label = 0,  
                    remote_connection = 0,  
                    component_tag = 0x0C,  
                ),  
            ],  
            application_descriptor(  
                application_profile = 0x0001,  
                version_major = 1,  
                version_minor = 0,  
                version_micro = 0,  
                service_bound_flag = 1,  
                visibility = 3,  
                application_priority = 1,  
                transport_protocol_labels = [ 0 ],  
            ),  
            application_name_descriptor(  
                application_name = "app_ginga"  
            ),  
            ginga_ncl_application_descriptor(  
                parameters = [ ]  
            ),  
            ginga_ncl_application_location_descriptor (  
                base_directory = "/home/tvd/servicio2/app_ginga",  
                class_path_extension = "",  
                initial_class = "main.ncl",  
            ),  
        ],  
    ],  
    version_number = 0,  
    section_number = 0,  
    last_section_number = 0,  
)
```

```
# ESCRIBIENDO LAS TABLAS A ARCHIVOS
```

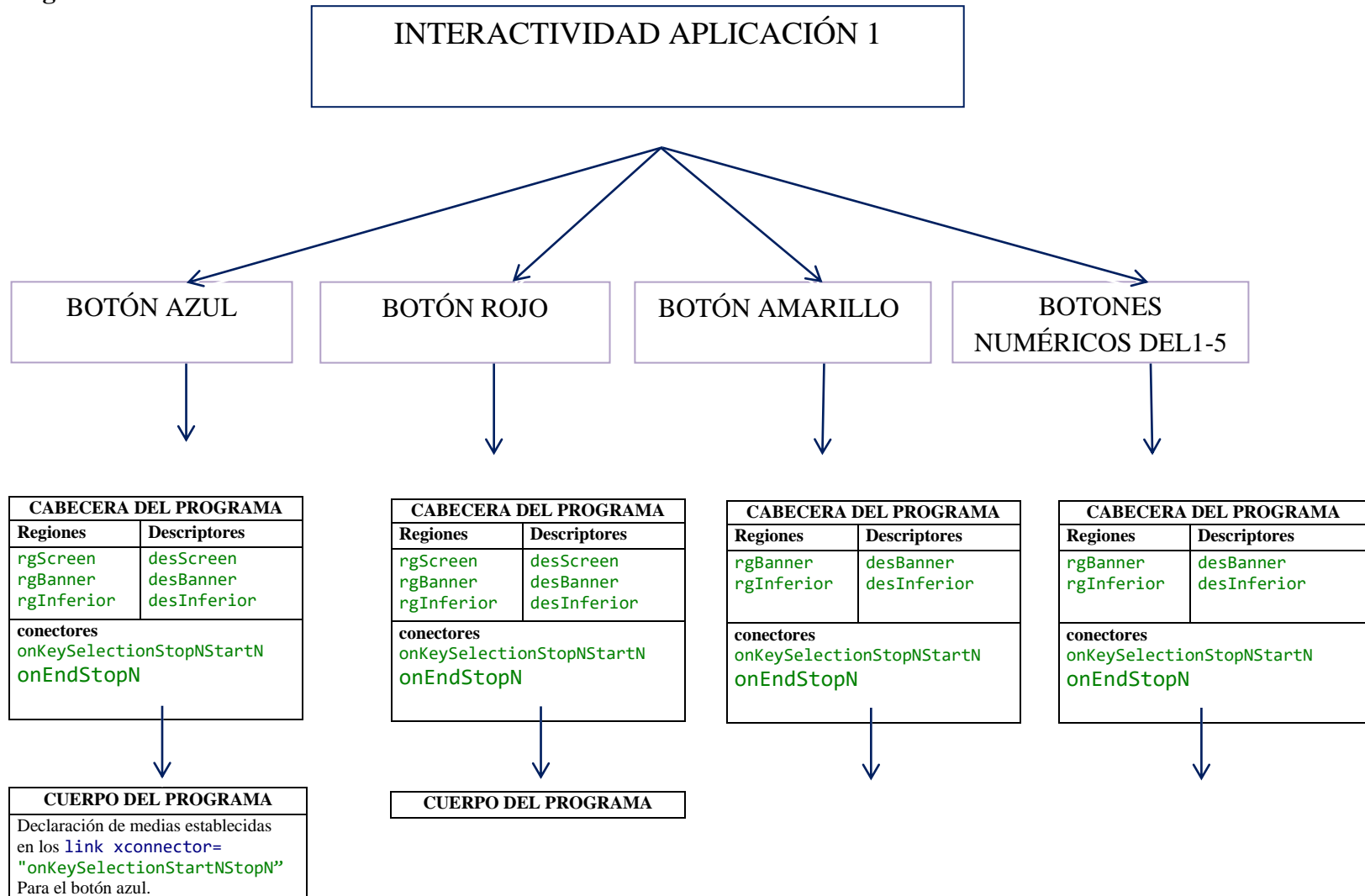
```
out = open("./nit.sec", "wb")
out.write(nit.pack())
out.close()
os.system("sec2ts 16 < ./nit.sec > ./nit.ts")
out = open("./pat.sec", "wb")
out.write(pat.pack())
out.close()
os.system("sec2ts 0 < ./pat.sec > ./pat.ts")
out = open("./sdt.sec", "wb")
out.write(sdt.pack())
out.close()
os.system("sec2ts 17 < ./sdt.sec > ./sdt.ts")
out = open("./pmt_sd.sec", "wb")
out.write(pmt_sd.pack())
out.close()
os.system("sec2ts " + str(pid_pmt_servicio) +
" < ./pmt_sd.sec > ./pmt_sd.ts")
```

```
GINGA
```

```
out = open("./ait.sec", "wb")
out.write(ait.pack())
out.close()
os.system("sec2ts "+ str(2001) + " < ./ait.sec > ./ait.ts")
```

ANEXO 2

Manual guía de laboratorio 1



Declaración de medias establecidas en los `link xconnector=`
`"onKeySelectionStartNStopN"`
Para el botón rojo.

CUERPO DEL PROGRAMA
Declaración de medias establecidas en los <code>link xconnector=</code> <code>"onKeySelectionStartNStopN"</code> Para el botón amarillo.

CUERPO DEL PROGRAMA
Declaración de medias establecidas en los <code>link xconnector=</code> <code>"onKeySelectionStartNStopN"</code> Para cada botón.

LABORATORIO 1 DE PRÁCTICA-GUIA NCL.

ESTRUCTURA DEL PROGRAMA

CABECERA	CUERPO	CABECERA	CUERPO
<ul style="list-style-type: none"> <head> 	<ul style="list-style-type: none"> <body> 	<ul style="list-style-type: none"> <head> 	<ul style="list-style-type: none"> <body>
<p>Región Lugar donde se va a mostrar</p>	<p>Puerta Lugar por donde se ingresa</p>	<p><regionBase> </regionBase></p>	<p><port id="" component="" /></p>
<p>Descriptor Como se va a mostrar</p>	<p>Media Que se va a mostrar</p>	<p><descriptorBase> </descriptorBase></p>	<p><media id=""> </media></p>
<p>Conector Cuando se va a mostrar</p>	<p>Enlace Cuando se va a mostrar</p>	<p><connectorBase> </connectorBase></p>	<p><link xconnector=""> </link></p>
<ul style="list-style-type: none"> </head> 	<ul style="list-style-type: none"> </body> 	<ul style="list-style-type: none"> </head> 	<ul style="list-style-type: none"> </body>

OBJETIVO:

Lograr reproducir un video en la pantalla, junto con una imagen (Icono de interactividad), que inicie medias simultáneamente cuando otros objetos medias terminen y viceversa, mediante la interacción con el usuario.

DECLARACION DE LA CABECERA DEL PROGRAMA.

PASO 1.

DEFINICIÓN DE REGIONES EN LA PANTALLA.

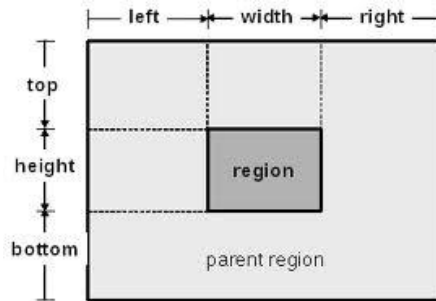
El área en donde se va a mostrar cada elemento en la pantalla, por medio de elementos llamados regiones. Una región representa la posición y tamaño de la zona donde ciertos objetos media serán visualizados, es decir una región sirve para inicializar la posición de los medios en una ubicación específica.

Elementos:	Atributos:	Contenido:
regionBase	<i>id, device, region</i>	(importBase region)+
region	<i>id, title, left, right, top, bottom, height, width, zIndex</i>	(region)*

Las regiones son el área de la pantalla donde se van a mostrar los elementos multimedia, la declaración de las regiones se lo debe hacer dentro de la cabecera del programa entre <head> y </head>. Las declaraciones de la región se lo debe efectuar dentro de <regionBase> y </regionBase> se debe declarar el nombre que identifica la region que es "región id",

³⁵ (Galabay Toalongo & Vivar Espinoza, 2012)

características como ancho (*width*), alto(*height*), posición(*left*, *right*, *top* y *bottom*), y la superposición en los planos de la pantalla(*Zindex*).



DECLARACION DE REGIONES

```

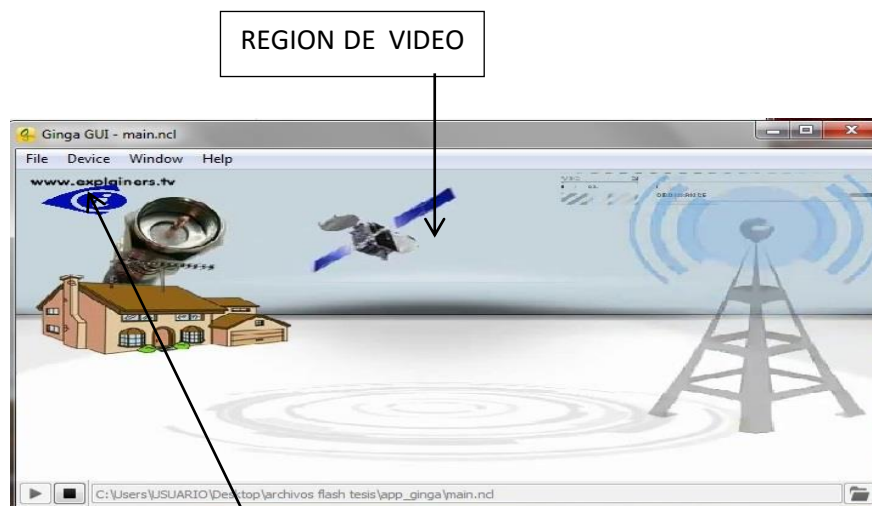
<head>
    <regionBase>
        #REGION DE VIDEO
        <region id="rgScreen" height="100%" width="100%"/>

        #REGION DEL ICONO DE INTERACTIVIDAD
        <region id="rgBoton" left="3%" top="5%" height="10%"
width="10%"/>

        #REGION IMAGENES LADO DERECHO
        <region id="rgBanner" left="70%" height="100%" width="30%"/>

        #REGION IMAGENES PARTE INFERIOR
        <region id="rgInferior" top="75%" height="25%" width="78%"/>
    </regionBase>

```



REGION DEL ICONO DE INTERACTIVIDAD

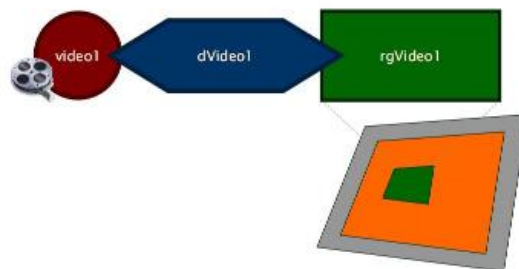


REGION IMÁGENES PARTE INFERIOR

REGION IMÁGENES LADO DERECHO

PASO 2.

DEFINICIÓN DE DESCRIPTORES.



La función del elemento descriptor es especificar los atributos que tendrá la región en la pantalla como duración, transparencia, color etc. Además el descriptor se constituye como un enlace entre los elementos multimedia y su región correspondiente con su id.

El elemento debe estar declarado dentro de la cabecera del programa entre `<head>` y `</head>`. Las declaraciones de se lo debe efectuar dentro de `<descriptorBase>` y `</descriptorBase>` se debe declarar el nombre que identifica el descriptor que es “descriptor id” el mismo que va a servir de enlace, y la región a la que se esta refiriendo “región”, estas dos declaraciones son la base y las declaraciones de atributos es opcional.

Elementos:	Atributos:	Contenido:
descriptor	<i>id, player, explicitDur, region, freeze, moveLeft, moveRight, move Up, moveDown, focusIndex, focusBorderColor, focusBorderWidth, focusBorderTransparency, focusSrc, focusSelSrc, selBorderColor, transIn, transOut</i>	(descriptorParam)*
descriptorParam	<i>name, value</i>	
descriptorBase	<i>id</i>	(importBase descriptor descriptorSwitch)+

DECLARACION DE DESCRIPTORES

<descriptorBase>

#DESCRIPTOR DE VIDEO

<descriptor id="desScreen" region="rgScreen"/>

DESCRIPTOR DEL ICONO DE INTERACTIVIDAD

<descriptor id="desBanner" region="rgBanner"/>

DESCRIPTOR IMAGENES LADO DERECHO

<descriptor id="desBoton" region="rgBoton"/>

DESCRIPTOR IMAGENES PARTE INFERIOR

<descriptor id="desInferior" region="rgInferior"/>

</descriptorBase>

PASO 3.






DEFINICION DE CONECTORES.

Los conectores están directamente asociado a los link y elementos media, establecen las acciones que se va a efectuar automáticamente, o mediante la interactividad con el usuario empleando botones del control remoto de Ginga.

Los conectores deben estar declarados en la cabecera del programa entre `<head>` y `</head>`.

Los componentes de un conector son los siguientes:

- `<connectorParam>`: para definir parámetros.
- `<compoundCondition>` y `<simpleCondition>`: condiciones del conector.
- `<simpleAction>` y `<compoundAction>`: acciones del conector.
- `onBegin`: Activa elementos multimedia declarados.
- `onEnd`: Termina la presentación de elementos multimedia declarados.
- `onAbort`: Aborta la presentación de elementos multimedia declarados.
- `onPause`: Pausa elementos multimedia vinculados al conector.
- `onResume`: Activa nuevamente la presentación de los elementos multimedia luego de una pausa.
- `onSelection`: Opcion que vincula la interaccion con el usuario y los botones del control remoto declarados, para efectuar una acción determinada.

Valores para la propiedad key:	Correspondencia con botones del control remoto:
RED	F1 
GREEN	F2 
YELLOW	F3 
BLUE	F4 
MENU	F5 
INFO	F6 
ENTER	
CURSOR_LEFT	
CURSOR_UP	
CURSOR_RIGHT	
CURSOR_DOWN	

- `onBeginAttribution`: Determina la activación de atributos de elementos multimedia.
- `onEndAttribution`: Determina la desactivación de atributos de elementos multimedia.

Importando bases de archivos externos

Empeado para importar un archivo .ncl en otro. Se lo declara dependiendo del tipo de archivo en el área a la que corresponda como por ejemplo si se importan conectores debería declararse entre `<conectorBase>` y `</conectorBase>`. La declaración `importBase documentURI`, sirve para declara el nombre del archivo que se quiere importar, mientras que la declaración `alias` se utiliza para declarar un nombre alternativo que permita utilizar el archivo importado en la parte que se requiera.

```
<conectorBase>
<importBase documentURI="ConnectorBase.ncl" alias="conEx"/>
</conectorBase>
```

El archivo que se importa comúnmente es la base de datos de conectores que se la puede encontrar directamente en internet, ejemplo <http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/Part%208%20-%20NCL3.0-DTV.pdf>.

EJEMPLO CONECTOR

```
<link xconnector="conEx#onEndStop">
<bind role="onEnd" component="animation" interface="segCred"/>
<bind role="stop" component="menu"/>
<bind role="stop" component="interactivity"/>
</link>
```

Nombre:	<i>onEndStopN</i>
Condición:	Termina la exhibición del nodo (ligado con la función <i>onBegin</i>).
Acción:	Cierra la presentación de uno o más en nodos (relacionadas con la función <i>stop</i>)
Ilustración:	
Código NCL:	<pre><causalConnector id="onEndStopN"> <simpleCondition role="onEnd"/> <simpleAction role="stop" max="unbounded"/> </causalConnector></pre>
Lectura:	Cuando el nodo ligado a la función <i>onEnd</i> inicia, terminan los nodos ligados a la función <i>stop</i> .

DECLARACION DE CONECTORES

```
<connectorBase>

    <causalConnector id="onKeySelectionStartNStopN">
        <connectorParam name="bounds"/>

        <connectorParam name="keyCode"/>
        <simpleCondition role="onSelection" key="$keyCode"/>
        <compoundAction operator="seq">
            <simpleAction role="set" value="$bounds"/>
            <simpleAction role="start" max="unbounded"
qualifier="par"/>
            <simpleAction role="stop" max="unbounded"
qualifier="par"/>
        </compoundAction>
    </causalConnector>

    <importBase documentURI="ConnectorBase.ncl" alias="conEx"/>

</connectorBase>
</head>
```

DECLARACION DEL CUERPO DEL PROGRAMA

PASO 4.

DEFINICIÓN PUERTOS.

Los puertos son declaraciones que sirven para presentar en forma sincronizada elementos multimedia inmediatamente al iniciar la reproducción de el documento ncl programado, la declaración de puertos se los debe efectuar luego de declarar el elemento multimedia asociado a este.

La declaración de los puertos se los realiza dentro del cuerpo del programa `<body>` y `</body>`, debe tener un nombre o identificación con `port id` y su vínculo a un elemento multimedia con `component`.

DECLARACION DE PUERTOS

```
<body>

<port id="portVideo" component="video"/>
<port id="portboton" component="boton"/>
```

PASO 5.

DEFINICIÓN DE ELEMENTOS MEDIA.

Los elementos multimedia o media, es lo que se va a mostrar en la pantalla como pueden ser: videos, imágenes, texto, archivos Lua, html etc.

Las medias se deben declarar en el cuerpo del programa entre `<body>` y `</body>` y su declaración debe estar compuesta de el identificador o nombre del elemento multimedia `media id`, el descriptor que enlazara la media con la región `descriptor` y la ruta de ubicación de la media en el programa `src`.

Elementos:	Atributos:	Contenido:
media	<i>id, src, refer, instance, type, descriptor</i>	(area property)*

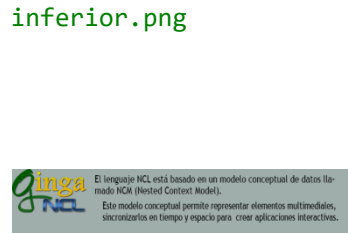
DECLARACION DE MEDIAS

```
<media id="video" descriptor="desScreen" src="media/backup.mp4">
  <property name="bounds" />
</media>
<media id="boton" descriptor="desBoton" src="media/info.png"/>
<media id="inferior" descriptor="desInferior"
src="media/inferior.png"/>
  <media id="inferiorOmedia" descriptor="desInferior"
src="media/inf_omedia.png"/>
  <media id="inferiorRegiones" descriptor="desInferior"
src="media/inf_regiones.png"/>
  <media id="inferiorDescriptores" descriptor="desInferior"
src="media/inf_descriptores.png"/>
  <media id="inferiorLinksconectores" descriptor="desInferior"
src="media/inf_linksconectores.png"/>
  <media id="inferiorContextos" descriptor="desInferior"
src="media/inf_contextos.png"/>
  <media id="lista" descriptor="desBanner" src="media/lista.png"/>
  <media id="omedia" descriptor="desBanner"
src="media/lat_omedia.png"/>
```

```

<media id="regiones" descriptor="desBanner"
src="media/lat_regiones.png"/>
<media id="descriptores" descriptor="desBanner"
src="media/lat_descriptores.png"/>
<media id="linksconectores" descriptor="desBanner"
src="media/lat_linksconectores.png"/>
<media id="contextos" descriptor="desBanner"
src="media/lat_contextos.png"/>

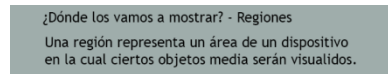
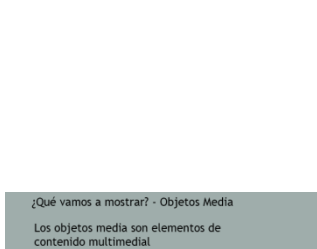
```



inf_omedia.png

lat_omedia.png

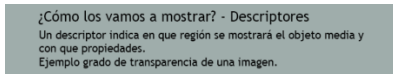
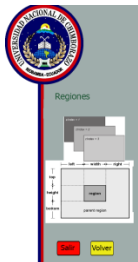
inf_regiones.png



lat_regiones.png

inf_descriptores.png

lat_descriptores.png

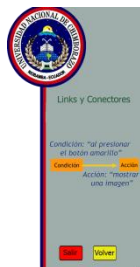


inf_linksconectores.png

lat_linksconectores.png

inf_contextos.png

¿Cuándo los vamos a mostrar? - Links y Conectores
Permiten especificar acciones sobre objetos media a partir de la ocurrencia de eventos.



Contexto
Permite agrupar elementos NCL y reutilizarlos.

lat_contextos.png



PASO 6.

DEFINICIÓN DE LINK.

Área funcional Linking.

El Link es el elemento del programa ncl que vincula un elemento multimedia con una acción, en otras palabras es la vinculación o enlace entre el conector y el elemento media. Para declarar un link se lo debe hacer dentro del cuerpo del programa `<body>` y `</body>` y debe estar compuesto de el nombre del conector `link xconnector` y elementos `bind` para establecer las acciones a efectuarse entre los conectores y los elementos media. El orden de declaración de los conectores es así:

- El primero se denomina role, que se utiliza para hacer referencia a un papel del conector
- El segundo se denomina component, que se utiliza para identificar el nodo
- El tercero es un atributo opcional denominado interfaz, usado para hacer referencia a una interfaz del nodo
- El cuarto es un atributo opcional denominado descriptor, usado para hacer referencia a un descriptor a ser asociado con el nodo

Elementos:	Atributos:	Contenido:
bind	<i>role, component, interface, descriptor</i>	(bindParam)*
bindParam	<i>name, value</i>	
linkParam	<i>name, value</i>	
link	<i>id, xconnector</i>	(linkParam*, bind+)

DECLARACION DE LINKS

```
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
```

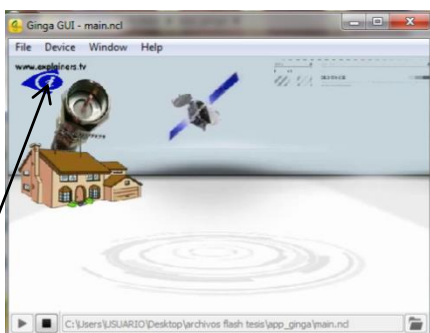
Declaración de parámetros link para iniciar interactividad de usuario al presionar el boton azul del control (F4) que vincula la presentación de un menú compuesto por los elementos media lista e inferior

```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="boton">
    <bindParam name="keyCode" value="BLUE"/>
  </bind>
  <bind role="stop" component="boton"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>

  <bind component="video" interface="bounds" role="set">
    <bindParam name="bounds" value="0,0,78%,75%" />
    <!-- left, top, width, height -->
  </bind>

</link>
```

```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,78%,75%" />
```



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



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

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

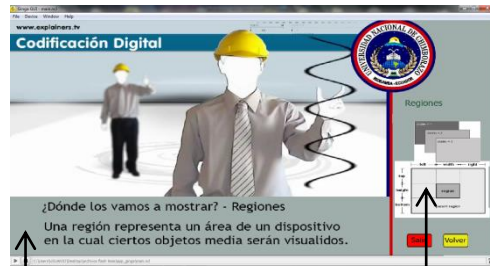


```
<bind role="stop" component="inferior"/>
<bind role="start" component="inferiorRegiones"/>
```

```
</link>
```



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



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

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

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

Declaración de elementos link que identifican el elemento media (lista) como un vínculo de interactividad al presionar el botón (3) del control, lo cual termina la ejecución de dos medias (lista e inferior) e inicia la presentación de dos medias (descriptores e inferiorDescriptores).

```
<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="lista">
    <bindParam name="keyCode" value="3"/>
  </bind>
  <bind role="stop" component="lista"/>
  <bind role="start" component="descriptores"/>
  <bind role="stop" component="inferior"/>
  <bind role="start" component="inferiorDescriptores"/>
</link>
```



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



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

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

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

Declaración de elementos link que identifican el elemento media (lista) como un vínculo de interactividad al presionar el botón (4) del control, lo cual termina la ejecución de dos medias (lista e inferior) e inicia la presentación de dos medias (linksconectores e inferiorLinksconectores).

```
<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="lista">
    <bindParam name="keyCode" value="4"/>
  </bind>
  <bind role="stop" component="lista"/>
  <bind role="start" component="linksconectores"/>

  <bind role="stop" component="inferior"/>
  <bind role="start" component="inferiorLinksconectores"/>

</link>
```



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



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

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

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

Declaración de elementos link que identifican el elemento media (lista) como un vínculo de interactividad al presionar el botón (5) del control, lo cual termina la ejecución de dos medias (lista e inferior) e inicia la presentación de dos medias (contextos e inferiorContextos).

```
<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="lista">
    <bindParam name="keyCode" value="5"/>
  </bind>
  <bind role="stop" component="lista"/>
  <bind role="start" component="contextos"/>
  <bind role="stop" component="inferior"/>
  <bind role="start" component="inferiorContextos"/>
</link>
```




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



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

```
<bind role="stop" component="inferior"/> <x-->
```

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

Declaración de elementos link que identifican el elemento media (lista) como un vínculo de interactividad al presionar el botón Rojo del control (F1), lo cual termina la ejecución de dos medias (lista e inferior) e inicia la presentación de dos medias (video (con atributos) e Icono de interactividad (botón)).

```

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="lista">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="lista"/>
  <bind role="stop" component="inferior"/>
  <bind role="start" component="boton"/>

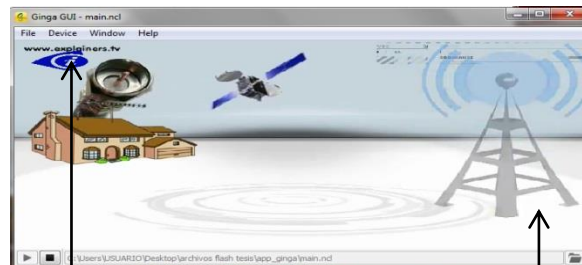
  <bind component="video" interface="bounds" role="set">
    <bindParam name="bounds" value="0,0,133.33%,133.5%" />
    <!-- left, top, width, height -->
  </bind>
</link>

```



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

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

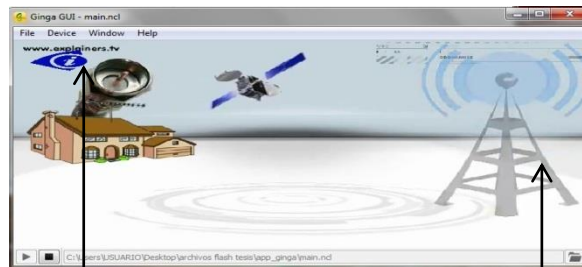


```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
</>
```

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


Declaración de elementos link que identifican el elemento media (omedia) como un vínculo de interactividad al presionar el botón Rojo del control (F1), lo cual termina la ejecución de dos medias (omedia e inferiorOmedia) e inicia la presentación de dos medias (video (con atributos) e Icono de interactividad (botón)).

```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="omedia">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="omedia"/>
  <bind role="stop" component="inferiorOmedia"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
<!-- left, top, width, height -->
</bind>
</link>
```



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

```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%"
/>
```

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

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

Declaración de elementos link que identifican el elemento media (regiones) como un vínculo de interactividad al presionar el botón Rojo del control (F1), lo cual termina la ejecución de dos medias (regiones e inferiorRegiones) e inicia la presentación de dos medias (video (con atributos) e Icono de interactividad (botón)).

```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="regiones">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="regiones"/>
  <bind role="stop" component="inferiorRegiones"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
<!-- left, top, width, height -->
</bind>
</link>
```



```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%"
/>
```

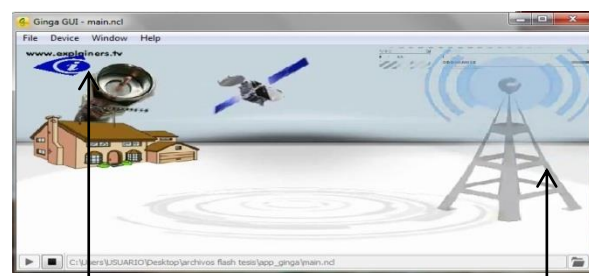
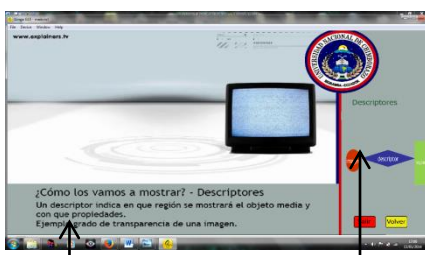
```
<bind role="stop" component=" regiones " />
```

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

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

Declaración de elementos link que identifican el elemento media (descriptores) como un vínculo de interactividad al presionar el botón Rojo del control (F1), lo cual termina la ejecución de dos medias (descriptores e inferiorDescriptores) e inicia la presentación de dos medias (video (con atributos) e Icono de interactividad (botón)).

```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="descriptores">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="descriptores"/>
  <bind role="stop" component="inferiorDescriptores"/>
  <bind role="start" component="boton"/>
  <bind component="video" interface="bounds" role="set">
    <bindParam name="bounds" value="0,0,133.33%,133.5%" />
  <!-- left, top, width, height -->
</bind>
```



```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%"
/>
```

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

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

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

Declaración de elementos link que identifican el elemento media (linksconectores) como un vínculo de interactividad al presionar el botón Rojo del control (F1), lo cual termina la ejecución de dos medias (linksconectores e inferiorLinksconectores) e inicia la presentación de dos medias (video (con atributos) e Icono de interactividad (botón)).

```

</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="linksconectores">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="linksconectores"/>
  <bind role="stop" component="inferiorLinksconectores"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
<!-- left, top, width, height -->
</bind>

```



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

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

```
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%"
 />
```

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



```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="omedia">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="omedia"/>
  <bind role="stop" component="inferiorOmedia"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

```



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

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



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

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

Declaración de elementos link que identifican el elemento media (regiones) como un vínculo de interactividad al presionar el botón Amarillo del control (F3), lo cual termina la ejecución de dos medias (regiones e inferiorRegiones) e inicia la presentación de dos medias (lista e inferior).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="regiones">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="regiones"/>
  <bind role="stop" component="inferiorRegiones"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

```



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



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

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

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

Declaración de elementos link que identifican el elemento media (descriptores) como un vínculo de interactividad al presionar el botón Amarillo del control (F3), lo cual termina la ejecución de dos medias (descriptores e inferiorDescriptores) e inicia la presentación de dos medias (lista e inferior).

```
<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="descriptores">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="descriptores"/>
  <bind role="stop" component="inferiorDescriptores"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>
```

```
</link>
```



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



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

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

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

Declaración de elementos link que identifican el elemento media (linksconectores) como un vínculo de interactividad al presionar el botón Amarillo del control (F3), lo cual termina la ejecución de dos medias (linksconectores e inferiorLinksconectores) e inicia la presentación de dos medias (lista e inferior).


```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="linksconectores">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="linksconectores"/>
  <bind role="stop" component="inferiorLinksconectores"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

```



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

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

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

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

Declaración de elementos link que identifican el elemento media (contextos) como un vínculo de interactividad al presionar el botón Amarillo del control (F3), lo cual termina la ejecución de dos medias (contextos e inferiorContextos) e inicia la presentación de dos medias (lista e inferior).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="contextos">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="contextos"/>
  <bind role="stop" component="inferiorContextos"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

```


CÓDIGO DE PROGRAMACION EN GINGA NCL DE main.ncl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <regionBase>
      <region id="rgScreen" height="100%" width="100%"/>
      <region id="rgBoton" left="3%" top="5%" height="10%"
width="10%"/>

      <region id="rgBanner" left="70%" height="100%" width="30%"/>
      <region id="rgVideo" height="75%" width="78%"/>
      <region id="rgInferior" top="75%" height="25%" width="78%"/>

    </regionBase>
    <descriptorBase>
      <descriptor id="desVideo" region="rgVideo"/>
      <descriptor id="desScreen" region="rgScreen"/>
      <descriptor id="desBanner" region="rgBanner"/>
      <descriptor id="desBoton" region="rgBoton"/>
      <descriptor id="desInferior" region="rgInferior"/>

    </descriptorBase>
    <connectorBase>

      <causalConnector id="onKeySelectionStartNStopN">
        <connectorParam name="bounds"/>

        <connectorParam name="keyCode"/>
        <simpleCondition role="onSelection" key="$keyCode"/>
        <compoundAction operator="seq">
          <simpleAction role="set" value="$bounds"/>
          <simpleAction role="start" max="unbounded"
qualifier="par"/>
          <simpleAction role="stop" max="unbounded"
qualifier="par"/>
        </compoundAction>
      </causalConnector>

      <importBase documentURI="ConnectorBase.ncl" alias="conEx"/>

    </connectorBase>
  </head>

  <body>
    <media id="video" descriptor="desScreen" src="media/backup.mp4">
      <property name="bounds" />
    </media>

    <port id="portVideo" component="video"/>
    <port id="portboton" component="boton"/>
  </body>
</ncl>
```

```

        <media id="boton" descriptor="desBoton" src="media/info.png"/>
        <media id="inferior" descriptor="desInferior"
src="media/inferior.png"/>
        <media id="inferiorOmedia" descriptor="desInferior"
src="media/inf_omedia.png"/>
        <media id="inferiorRegiones" descriptor="desInferior"
src="media/inf_regiones.png"/>
        <media id="inferiorDescriptores" descriptor="desInferior"
src="media/inf_descriptores.png"/>
        <media id="inferiorLinksconectores" descriptor="desInferior"
src="media/inf_linksconectores.png"/>
        <media id="inferiorContextos" descriptor="desInferior"
src="media/inf_contextos.png"/>

```

```

        <media id="lista" descriptor="desBanner" src="media/lista.png"/>
        <media id="omedia" descriptor="desBanner"
src="media/lat_omedia.png"/>
        <media id="regiones" descriptor="desBanner"
src="media/lat_regiones.png"/>
        <media id="descriptores" descriptor="desBanner"
src="media/lat_descriptores.png"/>
        <media id="linksconectores" descriptor="desBanner"
src="media/lat_linksconectores.png"/>
        <media id="contextos" descriptor="desBanner"
src="media/lat_contextos.png"/>

```

```
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
```

```

<link xconnector="onKeySelectionStartNStopN">
    <bind role="onSelection" component="boton">
        <bindParam name="keyCode" value="BLUE"/>
    </bind>
    <bind role="stop" component="boton"/>
    <bind role="start" component="lista"/>
    <bind role="start" component="inferior"/>

    <bind component="video" interface="bounds" role="set">
    <bindParam name="bounds" value="0,0,78%,75%" />
    <!-- left, top, width, height -->
</bind>

```

```
</link>
```

```
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
```

```

<link xconnector="conEx#onKeySelectionStartNStopN">
    <bind role="onSelection" component="lista">
        <bindParam name="keyCode" value="1"/>

```

```

        </bind>
        <bind role="stop" component="lista"/>
        <bind role="start" component="omedia"/>
    <bind role="stop" component="inferior"/>
        <bind role="start" component="inferiorOmedia"/>
</link>

<link xconnector="conEx#onKeySelectionStartNStopN">
    <bind role="onSelection" component="lista">
        <bindParam name="keyCode" value="2"/>
    </bind>
    <bind role="stop" component="lista"/>
    <bind role="start" component="regiones"/>

    <bind role="stop" component="inferior"/>
    <bind role="start" component="inferiorRegiones"/>

</link>
<link xconnector="conEx#onKeySelectionStartNStopN">
    <bind role="onSelection" component="lista">
        <bindParam name="keyCode" value="3"/>
    </bind>
    <bind role="stop" component="lista"/>
    <bind role="start" component="descriptores"/>
    <bind role="stop" component="inferior"/>
    <bind role="start" component="inferiorDescriptores"/>
</link>

<link xconnector="conEx#onKeySelectionStartNStopN">
    <bind role="onSelection" component="lista">
        <bindParam name="keyCode" value="4"/>
    </bind>
    <bind role="stop" component="lista"/>
    <bind role="start" component="linksconectores"/>

    <bind role="stop" component="inferior"/>
    <bind role="start" component="inferiorLinksconectores"/>

</link>

<link xconnector="conEx#onKeySelectionStartNStopN">
    <bind role="onSelection" component="lista">
        <bindParam name="keyCode" value="5"/>
    </bind>
    <bind role="stop" component="lista"/>
    <bind role="start" component="contextos"/>
    <bind role="stop" component="inferior"/>
    <bind role="start" component="inferiorContextos"/>
</link>

```

```

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="lista">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="lista"/>
  <bind role="stop" component="inferior"/>
  <bind role="start" component="boton"/>

  <bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
  <!-- left, top, width, height -->
</bind>
</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="omedia">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="omedia"/>
  <bind role="stop" component="inferiorOmedia"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
  <!-- left, top, width, height -->
</bind>
</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="regiones">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="regiones"/>
  <bind role="stop" component="inferiorRegiones"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
  <!-- left, top, width, height -->
</bind>
</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="descriptores">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="descriptores"/>
  <bind role="stop" component="inferiorDescriptores"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />

```

```

<!-- left, top, width, height -->
</bind>
</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="linksconectores">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="linksconectores"/>
  <bind role="stop" component="inferiorLinksconectores"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
<!-- left, top, width, height -->
</bind>
</link>

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="contextos">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="stop" component="contextos"/>
  <bind role="stop" component="inferiorContextos"/>
  <bind role="start" component="boton"/>
<bind component="video" interface="bounds" role="set">
  <bindParam name="bounds" value="0,0,133.33%,133.5%" />
<!-- left, top, width, height -->
</bind>
</link>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="omedia">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="omedia"/>
  <bind role="stop" component="inferiorOmedia"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="regiones">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="regiones"/>
  <bind role="stop" component="inferiorRegiones"/>
  <bind role="start" component="lista"/>
  <bind role="start" component="inferior"/>
</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="descriptores">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

```

```

        </bind>
        <bind role="stop" component="descriptores"/>
        <bind role="stop" component="inferiorDescriptores"/>
        <bind role="start" component="lista"/>
        <bind role="start" component="inferior"/>
</link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="linksconectores">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>
        <bind role="stop" component="linksconectores"/>
        <bind role="stop" component="inferiorLinksconectores"/>
        <bind role="start" component="lista"/>
        <bind role="start" component="inferior"/>
</link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="contextos">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>
        <bind role="stop" component="contextos"/>
        <bind role="stop" component="inferiorContextos"/>
        <bind role="start" component="lista"/>
        <bind role="start" component="inferior"/>
</link>

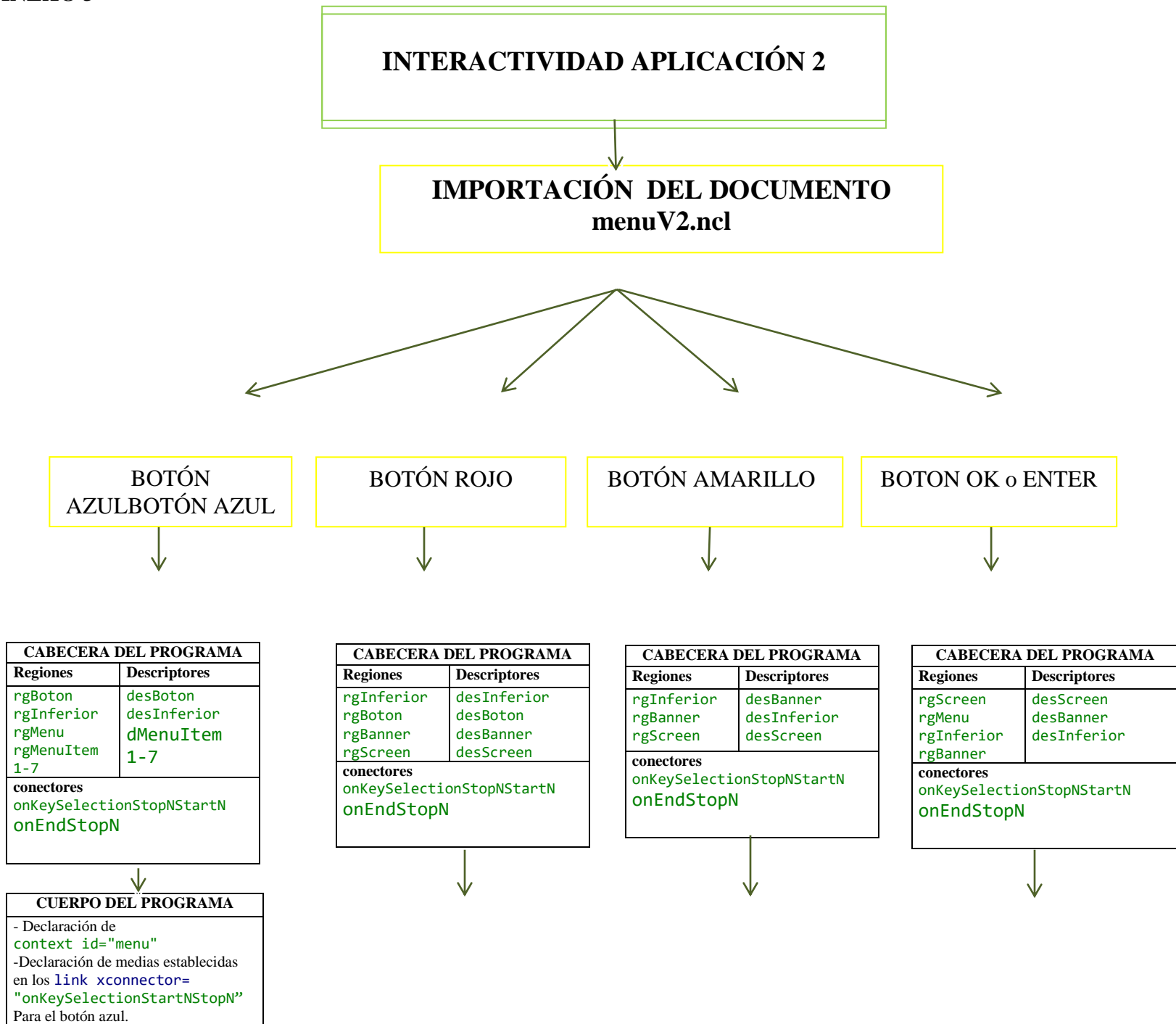
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
    <link xconnector="conEx#onEndStopN">
        <bind role="onEnd" component="video"/>

        <bind role="stop" component="regiones"/>
        <bind role="stop" component="descriptores"/>
        <bind role="stop" component="linksconectores"/>
        <bind role="stop" component="contextos"/>
        <bind role="stop" component="omedia"/>
        <bind role="stop" component="inferiorContextos"/>
        <bind role="stop" component="inferiorLinksconectores"/>
        <bind role="stop" component="inferiorDescriptores"/>
        <bind role="stop" component="inferiorRegiones"/>
        <bind role="stop" component="inferiorOmedia"/>
        <bind role="stop" component="lista"/>
        <bind role="stop" component="boton"/>
        <bind role="stop" component="inferior"/>

</link>

```

```
</body>  
</nc1>
```





CUERPO DEL PROGRAMA
Declaración de medias establecidas en los <code>link xconnector=</code> <code>"onKeySelectionStartNStopN"</code> Para el botón rojo.

CUERPO DEL PROGRAMA
Declaración de medias establecidas en los <code>link xconnector=</code> <code>"onKeySelectionStartNStopN"</code> Para el botón amarillo.

CUERPO DEL PROGRAMA
- Declaración de <code>context id="menu"</code> -Declaración de medias establecidas en los: <code>link xconnector=</code> <code>"onKeySelectionStartNStopN"</code> Y utilizacion de: <code>interface=pMenuItem 1-7</code> Para cada caso.

LABORATORIO 2 DE PRÁCTICA-GUIA NCL.

Para información adicional sobre estructura, objetivo, definiciones, referirse al manual 1.

DECLARACION DE LA CABECERA DEL PROGRAMA

PASO 1. EMPLEO DE UN DOCUMENTO NCL IMPORTADO PARA COMPLEMENTO.

En el laboratorio 1 ya se importó un archivo ncl con contenido de la base de conectores. Esta sección aparte de importar la base de conectores, **se definirá la importación de un documento que contiene programación para poder navegar mediante las flechas del control, en un menú de medias.** Para permitir que una base de entidades sea incorporada a otra ya existente, el módulo Import define el elemento `<importBase>`, que tiene dos atributos: `DocumentURI` y `alias`. El atributo `documentURI` se refiere a un URI correspondiente al documento NCL conteniendo la base a ser importada. El atributo `alias` especifica un nombre a ser utilizado como código cuando sea necesario referirse a elementos de esa base importada. EL nombre del atributo `alias` debe ser obligatoriamente único en un documento y su alcance está sup editado al documento que lo definió.

Elementos:	Atributos:	Contenido:
<code>importBase</code>	<i>alias, documentURI, región</i>	
<code>imported DocumentBase</code>	<i>id</i>	<code>(importNCL)+</code>
<code>importNCL</code>	<i>alias, documentURI</i>	

`<head>`

```

    <importedDocumentBase>
<importNCL documentURI="menuV2.ncl" alias="menuID"/>
</importedDocumentBase>

```

ESTRUCTURA DEL DOCUMENTO menuV2.ncl

```

<ncl id="menuV1" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>
<regionBase>
<region id="rgMenu" left="2%" top="20%" width="400"
height="70%" zIndex="99">

```

DECLARACION DE REGIONES

```

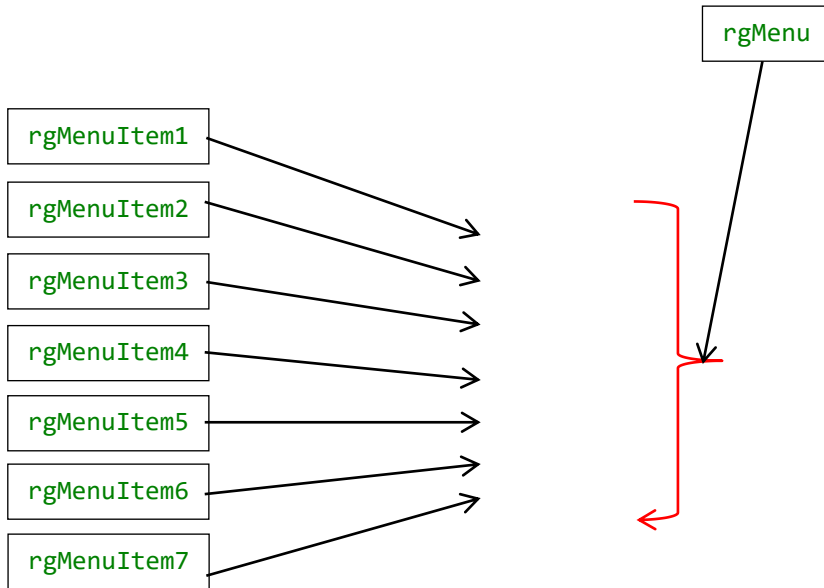
<region id="rgMenuItem1" top="0" height="50" zIndex="2" />
<region id="rgMenuItem2" top="60" height="50" zIndex="2" />
<region id="rgMenuItem3" top="120" height="50" zIndex="2"/>

```

³⁶ (Galabay Toalongo & Vivar Espinoza, 2012)

```
<region id="rgMenuItem4" top="180" height="50" zIndex="2" />
<region id="rgMenuItem5" top="240" height="50" zIndex="2" />
<region id="rgMenuItem6" top="300" height="50" zIndex="2" />
<region id="rgMenuItem7" top="360" height="50" zIndex="2" />

</region>
</regionBase>
```



DECLARACION DE DESCRIPTORES

Para declarar estos descriptores se usaran otros atributos que darán una presentación adecuada al menú de medias expuestas.

- **focusIndex:** establece un índice a ser utilizado para la navegación entre los objetos multimedia presentados en la pantalla. Si un descriptor no define éste, el objeto multimedia no podrá recibir el foco de navegación. En el inicio de ejecución del programa, el foco pasa para el elemento asociado al descriptor de menor índice. Una observación, el valor del focusIndex puede no ser numérico, en cuyo caso será elegido el índice lexicográficamente menor.
- **moveUp:** establece el descriptor, a través de su índice, que recibirá el foco cuando la "Flecha arriba" del control remoto es presionado. Esta asignación sólo se realiza cuando el descriptor que lo define esté con el foco.
- **moveDown:** establece el descriptor, a través de su índice, que recibirá el foco cuando la Flecha abajo" del control remoto es presionado. Esta asignación sólo se realiza cuando el elemento asociado a este descriptor este con el foco
- **focusBorderWidth:** define el ancho del borde en píxeles del rectángulo cuando el elemento a él asociado recibe el foco. El espesor puede tomar valores positivos y negativos. En el caso de ser positivo se presentará fuera de la región, si no, el borde se presentará dentro de la región
- **focusBorderColor:** define el color del borde (rectángulo) de la región de este descriptor cuando el objeto a él asociado recibe el foco. El atributo puede tener uno de los siguientes valores: white, black, silver, gray, red, marrón, fucsia, purple, lime, green, yellow, olive, blue, navy, aqua (blanco, negro, plata, gris, rojo, fucsia, marrón, morado, verde lima, amarillo, aceite de oliva, azul, azul marino, azul turquesa, o verde azulado).
- **focusSrc:** define un archivo multimedia alternativo que debe ser presentado cuando el elemento asociado a este descriptor esté con el foco.
- **selBorderColor:** define un color de borde que debe ser exhibido cuando sea presionado el botón Ok o Enter mientras el elemento asociado a este descriptor este con el foco.

```
<descriptorBase>
<descriptor id="dMenuItem1" region="rgMenuItem1"
focusIndex="1" moveUp="7" moveDown="2"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_herra_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem2" region="rgMenuItem2"
focusIndex="2" moveUp="1" moveDown="3"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_objetos_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem3" region="rgMenuItem3"
focusIndex="3" moveUp="2" moveDown="4"
```

```

focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_regiones_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem4" region="rgMenuItem4"
focusIndex="4" moveUp="3" moveDown="5"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_descriptores_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem5" region="rgMenuItem5"
focusIndex="5" moveUp="4" moveDown="6"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_linksconectores_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem6" region="rgMenuItem6"
focusIndex="6" moveUp="5" moveDown="7"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_contextos_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem7" region="rgMenuItem7"
focusIndex="7" moveUp="6" moveDown="1"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_resultado_on.png" selBorderColor="aqua"/>
</descriptorBase>
</head>

```

dMenuItem1

dMenuItem2

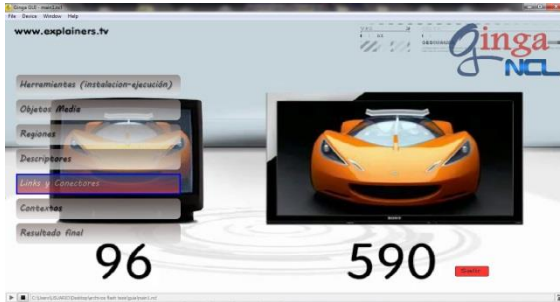


dMenuItem3

dMenuItem4



dMenuItem5



dMenuItem6



dMenuItem7



DECLARACION DEL CUERPO DEL PROGRAMA

DECLARACION DE EL CONTEXTO

El elemento body es un caso particular de contexto, presentando el documento como un todo. En NCL, un nodo de composición o contexto está representado por el elemento <context> y se utiliza para estructurar un nodo hipermedia, los mismos que pueden ser anidados, con el objetivo de reflejar la estructura del documento y ayudarle al programador a organizar de mejor manera los segmentos del programa audiovisual interactivo. Éste puede tener otros elementos, incluyendo otros contextos.

Los atributos de un contexto son:

- id: identificador único de contexto
- refer: referencia a otro contexto previamente denido, utiliza los atributos del contexto referenciado, excepto el id que debe ser de la forma: alias#id_del_body_documento_importado.
- El id del body importado debe ser diferente al del documento importador.

```
<body>  
  <context id="menu">
```

DECLARACIÓN DE PUERTOS.

<property> : es una propiedad de la media que es utilizado cuando algún atributo de una media es necesario.

value: indica el valor que será comparado con la propiedad

```
<port id="pMenuItem1" component="menuItem1" />  
<port id="pMenuItem2" component="menuItem2" />  
<port id="pMenuItem3" component="menuItem3" />  
<port id="pMenuItem4" component="menuItem4" />  
<port id="pMenuItem5" component="menuItem5" />  
<port id="pMenuItem6" component="menuItem6" />  
<port id="pMenuItem7" component="menuItem7" />  
<media id="menuItem1" src="boton_herra_off.png"
```

DECLARACIÓN DE MEDIAS

```
<media id="menuItem1" src="boton_herra_off.png"  
descriptor="dMenuItem1" >  
<property name="transparency" value="30 %"/>  
</media>  
<media id="menuItem2" src="boton_objetos_off.png"  
descriptor="dMenuItem2" >  
<property name="transparency" value="30 %"/>  
</media>  
<media id="menuItem3" src="boton_regiones_off.png"  
descriptor="dMenuItem3" >
```

```

<property name="transparency" value="30 %"/>
</media>
<media id="menuItem4" src="boton_descriptores_off.png"
descriptor="dMenuItem4" >
<property name="transparency" value="30 %"/>
</media>
<media id="menuItem5" src="boton_linksconectores_off.png"
descriptor="dMenuItem5" >
<property name="transparency" value="30 %"/>
</media>
<media id="menuItem6" src="boton_contextos_off.png"
descriptor="dMenuItem6" >
<property name="transparency" value="30 %"/>
</media>
<media id="menuItem7" src="boton_resultado_off.png"
descriptor="dMenuItem7" >
<property name="transparency" value="30 %"/>
</media>
</context>
</body>
</ncl>

```

boton_herra_off.png

Herramientas (instalacion-ejecución)

boton_objetos_off.png

Objetos Media

boton_regiones_off.png

Regiones

boton_descriptores_off.png

Descriptores

boton_linksconectores_off.png

Links y Conectores

boton_contextos_off.png

Contextos

boton_resultado_off.png

Resultado final

PASO 2

DECLARACION DE REGIONES

#REGIÓN DE VIDEOS


```

<regionBase>
  <region id="rgScreen" height="100%" width="100%"
zIndex="0"/>

      #REGIÓN DEL ICONO DE INTERACTIVIDAD

      <region id="rgBoton" left="3%" top="5%" height="10%"
width="10%" zIndex="2" />

      #REGIÓN DEL LOGO-GINGA (logo, logo1)

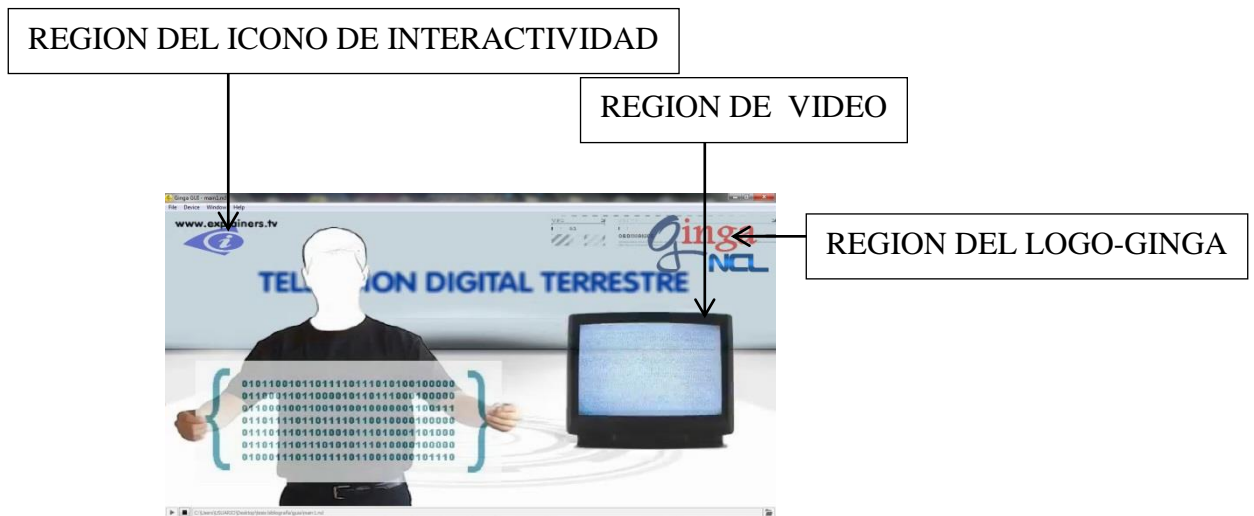
      <region id="rgBanner" right="1%" top="1%" height="20%"
width="20%" zIndex="2"/>

      #REGIÓN DEL BOTÓN SALIR Y SALIRVOLVER

      <region id="rgInferior" right="5%" bottom="5%" height="5%"
width="15%" zIndex="1"/>

</regionBase>

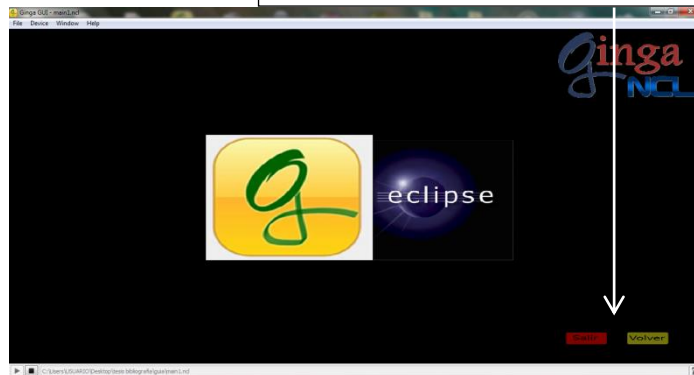
```





REGION DEL BOTON SALIR

REGION DEL BOTON SALIRVOLVER



PASO 3

DECLARACION DE DESCRIPTORES

```
<descriptorBase>
  <descriptor id="desScreen" region="rgScreen"/>
  <descriptor id="desBanner" region="rgBanner"/>
  <descriptor id="desBoton" region="rgBoton"/>
  <descriptor id="desInferior" region="rgInferior"/>
</descriptorBase>
```

PASO 4

DECLARACION DE CONECTORES

```
<connectorBase>
```

```

    <causalConnector id="onKeySelectionStartNStopN">
        <connectorParam name="bounds"/>

        <connectorParam name="keyCode"/>
        <simpleCondition role="onSelection" key="$keyCode"/>
        <compoundAction operator="seq">
            <simpleAction role="set" value="$bounds"/>
            <simpleAction role="start" max="unbounded"
qualifier="par"/>
            <simpleAction role="stop" max="unbounded"
qualifier="par"/>
        </compoundAction>
    </causalConnector>

```

IMPORTACION DE CONECTORES

```

    </causalConnector>

    <importBase documentURI="ConnectorBase.nc1" alias="conEx"/>

    </connectorBase>
</head>

```

PASO 5

DECLARACION DE PUERTOS Y MEDIAS

El atributo refer hace referencia a un elemento id que será reusado. Sólo <media>, <context>, <body> o <switch> se pueden reutilizar. Un elemento que hace referencia a otro elemento no puede ser reutilizado; es decir, su id no puede ser el valor de un atributo refer.

```

<body>

    <context id="menu" refer="menuID#menu"/>
    <media id="video" descriptor="desScreen" src="media/backup.mp4">
        <property name="bounds" />
    </media>
    <media id="video1" descriptor="desScreen" src="japo6.mpg">
        <property name="bounds" />
    </media>
    <media id="video2" descriptor="desScreen" src="JAP02.mpg">
        <property name="bounds" />
    </media>
    <media id="video3" descriptor="desScreen" src="japo.mpg">
        <property name="bounds" />
    </media>

```

```

<media id="video4" descriptor="desScreen" src="Japo1.mpg">
  <property name="bounds" />

</media>
<media id="video5" descriptor="desScreen" src="JAP03.avi">
  <property name="bounds" />

</media>
<media id="video6" descriptor="desScreen" src="JAP04.mpg">
  <property name="bounds" />

</media>
<media id="video7" descriptor="desScreen" src="JAP05.mpg">
  <property name="bounds" />

</media>

<port id="portVideo" component="video"/>
<port id="portlogo" component="logo"/>
<port id="portboton" component="boton"/>

<media id="boton" src="media/info.png"
descriptor="desBoton" >
<property name="transparency" value="50 %"/>
</media>
<media id="salir" src="media/boton_salir.png"
descriptor="desInferior" >
<property name="transparency" value="20 %"/>
</media>
<media id="salirvolver" src="media/boton_salirvolver.png"
descriptor="desInferior" >
<property name="transparency" value="50 %"/>
</media>
<media id="logo" src="logo-ginga.png"
descriptor="desBanner" >
<property name="transparency" value="20 %"/>
</media>

```

#La razón que obliga a declarar una media con la misma fuente (`logo-ginga.png`) pero con distinto nombre (`logo1`), es porque debido a la declaración del link final de que con el video terminen varias imágenes, no se puede establecer la misma media (`logo`) para que inicie la siguiente interacción una vez que finalice la presentación del video porque representa una declaración ilógica.

```

<media id="logo1" src="logo-ginga.png"
descriptor="desBanner" >
<property name="transparency" value="20 %"/>
</media>

```

PASO 6

DECLARACION DE LINKS

```

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
# Declaración de parámetros link para iniciar interactividad de usuario al
presionar el botón azul del control (F4) que vincula la presentación de un menú
compuesto por los elementos media menú y salir.

```

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="boton">
    <bindParam name="keyCode" value="BLUE"/>
  </bind>
  <bind role="stop" component="boton"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="salir"/>

```

```

</link>

```

```

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

```



```

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

```



```

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

```


Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.nc1` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución del elemento medias (video) e inicia la presentación de tres medias (video2, salirvolver y logo1).

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem2"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video2"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

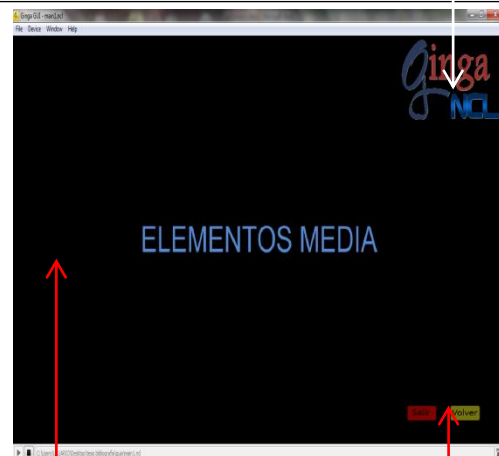
</link>

```

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



```
<bind role="start" component="logo1"/>
```



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

```
<bind role="start" component="video2"/>
```

Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.ncl` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución de la media (video) e inicia la presentación de tres medias (video3, salirvolver y logo1).

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem3"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video3"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

</link>

```

```

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

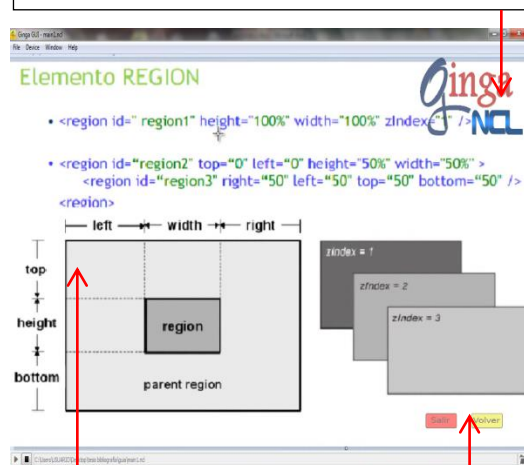
```



```

<bind role="start" component="logo1"/>

```



```

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

```

```

<bind role="start" component="video3"/>

```


Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.ncl` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución de la media (video) e inicia la presentación de tres medias (video4, salirvolver y logo1).

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem4"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video4"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

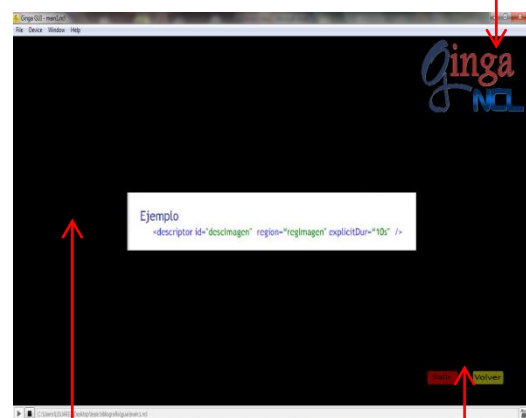
</link>

```

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



```
<bind role="start" component="logo1"/>
```



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

```
<bind role="start" component="video4"/>
```

Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.ncl` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución de la media (video) e inicia la presentación de tres medias (video5, salirvolver y logo1).

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem5"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video5"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

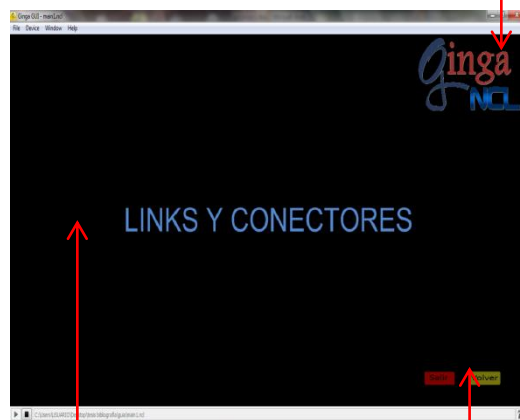
</link>

```

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



```
<bind role="start" component="logo1"/>
```



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

```
<bind role="start" component="video5"/>
```

Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.ncl` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución de la media (video) e inicia la presentación de tres medias (video6, salirvolver y logo1).

```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem6"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video6"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

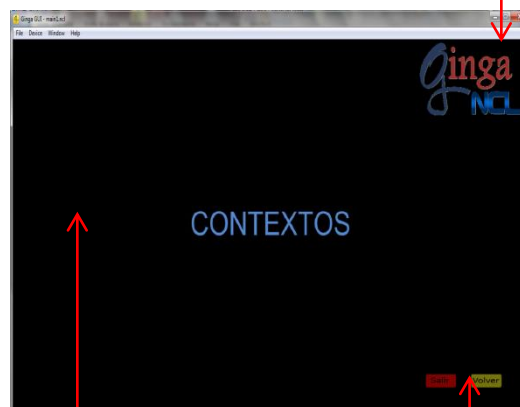
</link>

```

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



```
<bind role="start" component="logo1"/>
```



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

```
<bind role="start" component="video6"/>
```

Declaración de elementos link que identifican mediante component el contexto importado por medio del archivo `menuV2.nc1` compuesto por puertos que sirven como enlaces para por medio de otro archivo reutilizar sus componentes configurados.

Además se emplea La identificación del elemento media (menu) como un vínculo de interactividad al presionar el botón (OK o ENTER) del control, lo cual termina la ejecución de la media (video) e inicia la presentación de tres medias (video7, salirvolver y logo1).

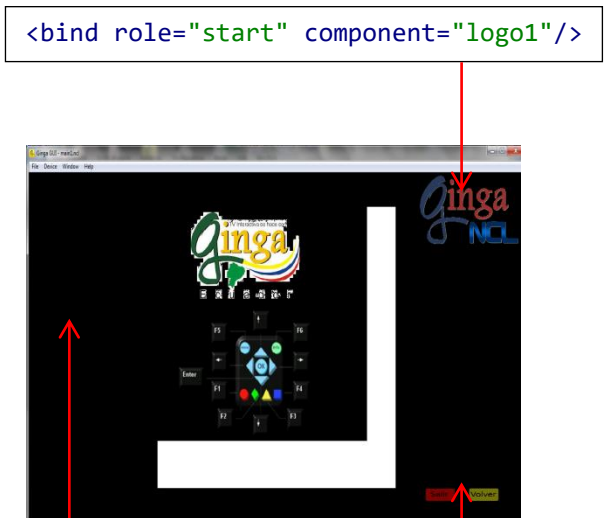
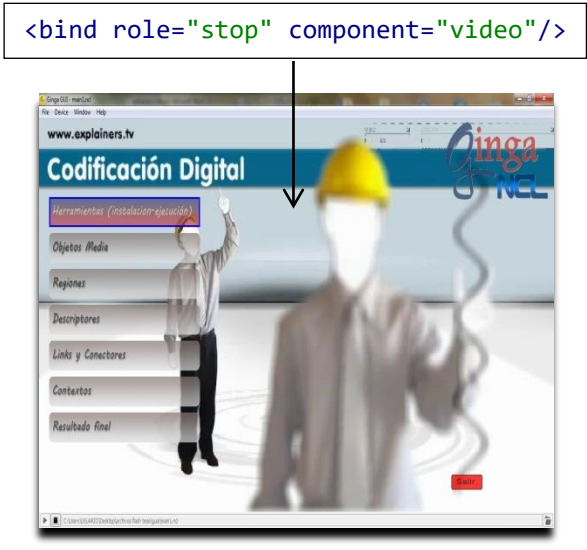
```

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem7"/>

  <bind role="stop" component="video"/>
  <bind role="start" component="video7"/>
  <bind role="start" component="salirvolver"/>
  <bind role="start" component="logo1"/>

</link>

```



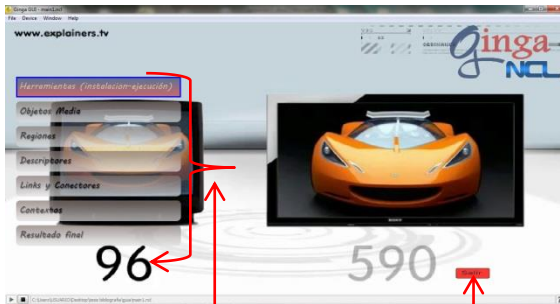
`<bind role="start" component="salirvolver"/>`

`<bind role="start" component="video7"/>`

```
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
```

Declaración de elementos link que identifican el elemento media (logo) como un vínculo de interactividad al presionar el botón (ROJO o F1) del control, lo cual termina la ejecución de dos medias (salir y menu) e inicia la presentación de la media (boton).

```
<link xconnector="onKeySelectionStartNStopN">  
  <bind role="onSelection" component="logo">  
    <bindParam name="keyCode" value="RED"/>  
  </bind>  
  
  <bind role="stop" component="salir"/>  
  <bind role="stop" component="menu"/>  
  <bind role="start" component="boton"/>  
  
</link>
```



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

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



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

Declaración de elementos link que identifican el elemento media (salirvolver) como un vínculo de interactividad al presionar el botón (ROJO o F1) del control, lo cual termina la ejecución de tres medias (video1, salirvolver y logo1) e inicia la presentación de tres medias (boton, video y logo).

El atributo role se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

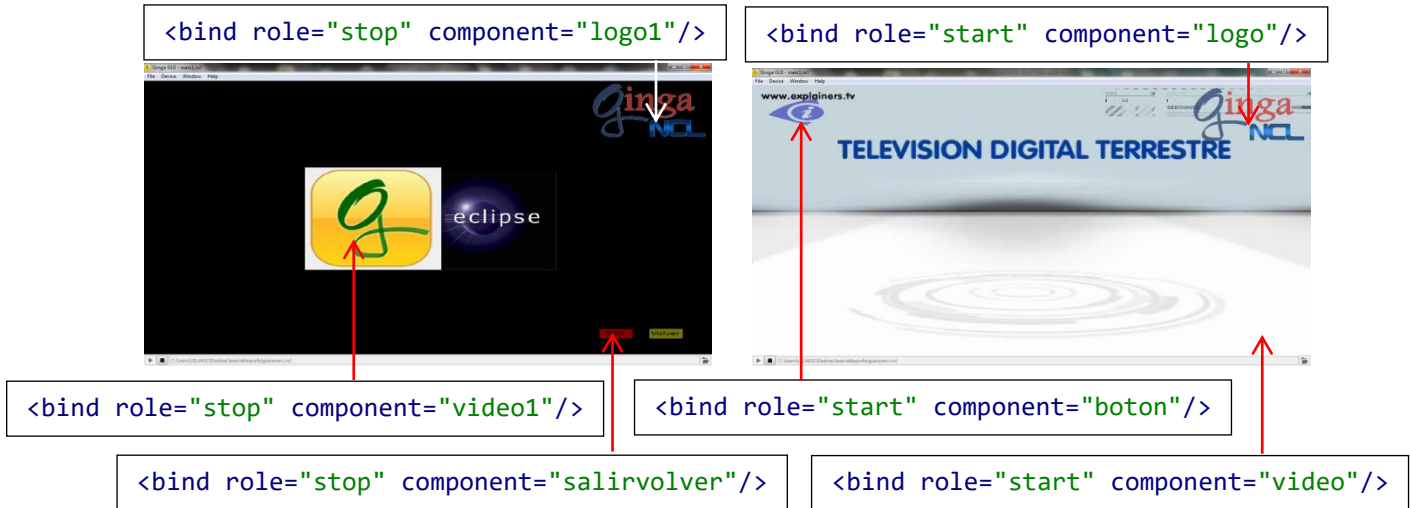
```
<link xconnector="onKeySelectionStartNStopN">  
  <bind role="onSelection" component="salirvolver">  
    <bindParam name="keyCode" value="RED"/>  
  </bind>  
  
  <bind role="stop" component="logo1"/>
```

```

<bind role="stop" component="video1"/>
<bind role="stop" component="salirvolver"/>
<bind role="start" component="boton"/>
<bind role="start" component="video"/>
<bind role="start" component="logo"/>
<bind component="video" interface="bounds" role="set"/>

</link>

```



Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (ROJO o F1) del control, lo cual termina la ejecución de tres medias (`video2`, `salirvolver` y `logo1`) e inicia la presentación de dos medias (botón, video y logo). El atributo `role` se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

```

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video2"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

```

```
<bind role="stop" component="logo1"/>
```

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



```
<bind role="stop" component="video2"/>
```

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

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

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

Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (ROJO o F1) del control, lo cual termina la ejecución de tres medias (`video3`, `salirvolver` y `logo1`) e inicia la presentación de tres medias (botón, video y logo). El atributo `role` se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

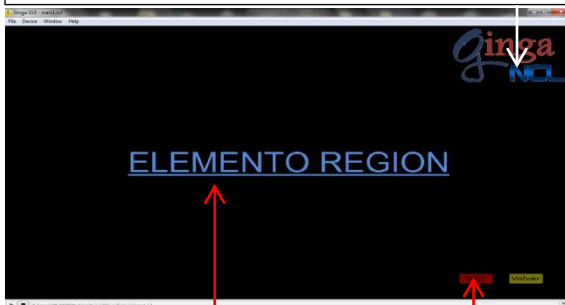
```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video3"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>
```

```
<bind role="stop" component="logo1"/>
```

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



```
<bind role="stop" component="video3"/>
```

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

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

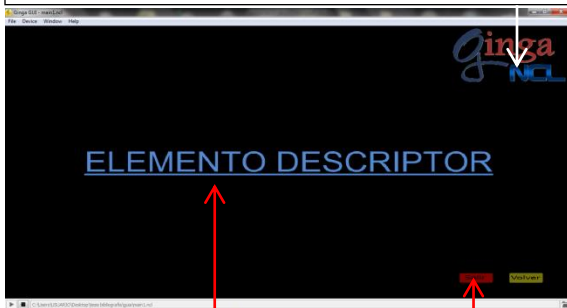
```
<bind role="start" component="video"/>
```

Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (ROJO o F1) del control, lo cual termina la ejecución de tres medias (`video4`, `salirvolver` y `logo1`) e inicia la presentación de tres medias (botón, `video` y `logo`). El atributo `role` se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

```
<link xconnector="onKeySelectionStartNStopN">  
  <bind role="onSelection" component="salirvolver">  
    <bindParam name="keyCode" value="RED"/>  
  </bind>  
  
  <bind role="stop" component="logo1"/>  
  <bind role="stop" component="video4"/>  
  <bind role="stop" component="salirvolver"/>  
  <bind role="start" component="boton"/>  
  <bind role="start" component="video"/>  
  <bind role="start" component="logo"/>  
  <bind component="video" interface="bounds" role="set"/>  
  
</link>
```

```
<bind role="stop" component="logo1"/>
```

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



```
<bind role="stop" component="video4"/>
```

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

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

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

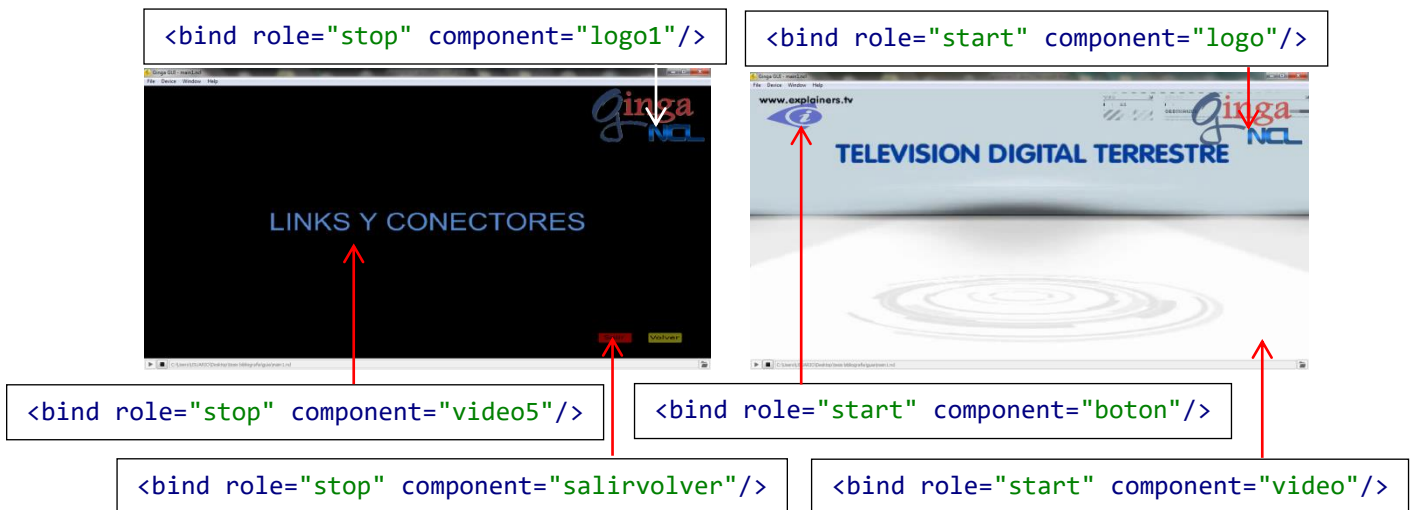
Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (ROJO o F1) del

control, lo cual termina la ejecución de tres medias (video5, salirvolver y logo1) e inicia la presentación de tres medias (botón, video y logo). El atributo role se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

```
<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video5"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>
```



Declaración de elementos link que identifican el elemento media (salirvolver) como un vínculo de interactividad al presionar el botón (ROJO o F1) del

control, lo cual termina la ejecución de tres medias (video6, salirvolver y logo1) e inicia la presentación de tres medias (botón, video y logo). El atributo role se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

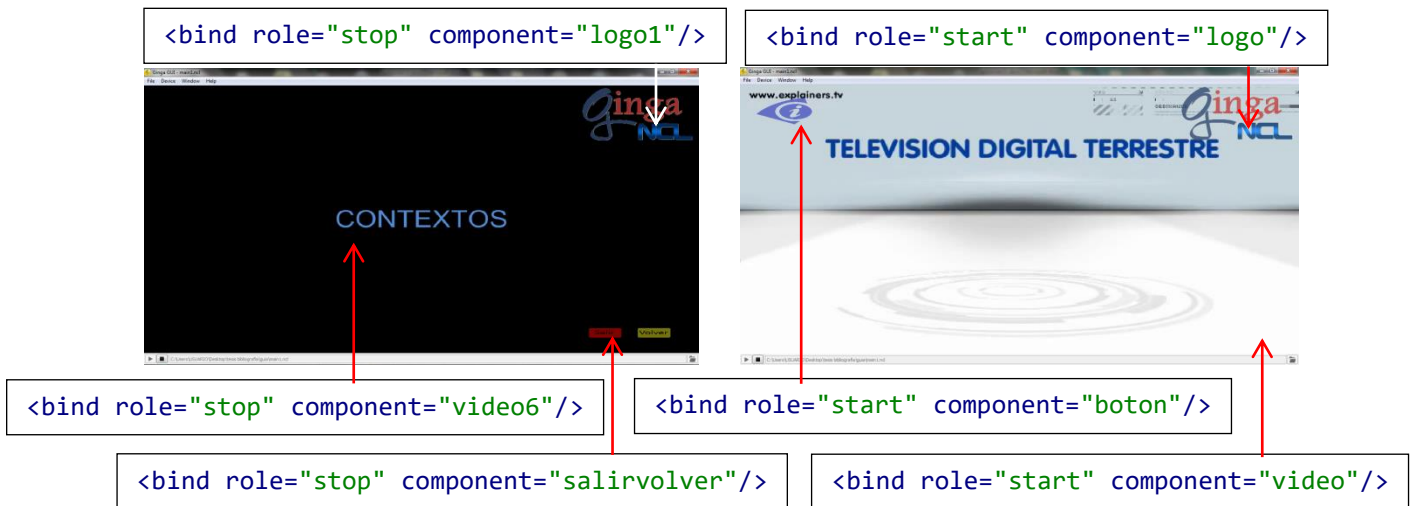
```

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video6"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

```



Declaración de elementos link que identifican el elemento media (salirvolver) como un vínculo de interactividad al presionar el botón (ROJO o F1) del

control, lo cual termina la ejecución de tres medias (video7, salirvolver y logo1) e inicia la presentación de tres medias (botón, video y logo). El atributo role se utiliza para hacer referencia a un papel del conector en este caso reiniciar las propiedades del video.

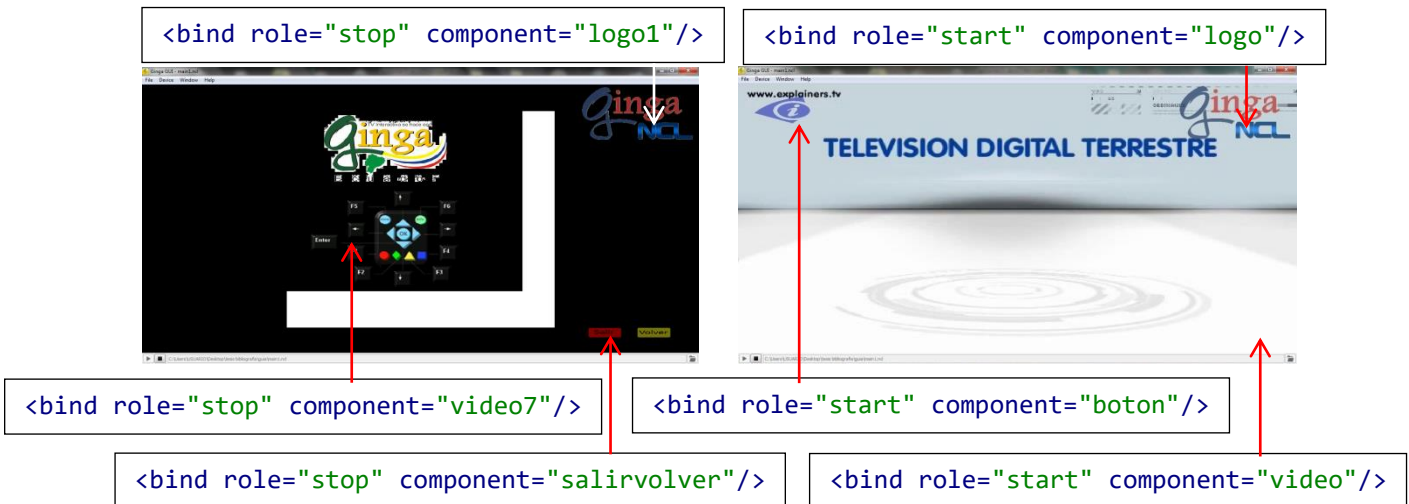
```

<link xconnector="onKeySelectionStartNStopN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video7"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

```



```

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

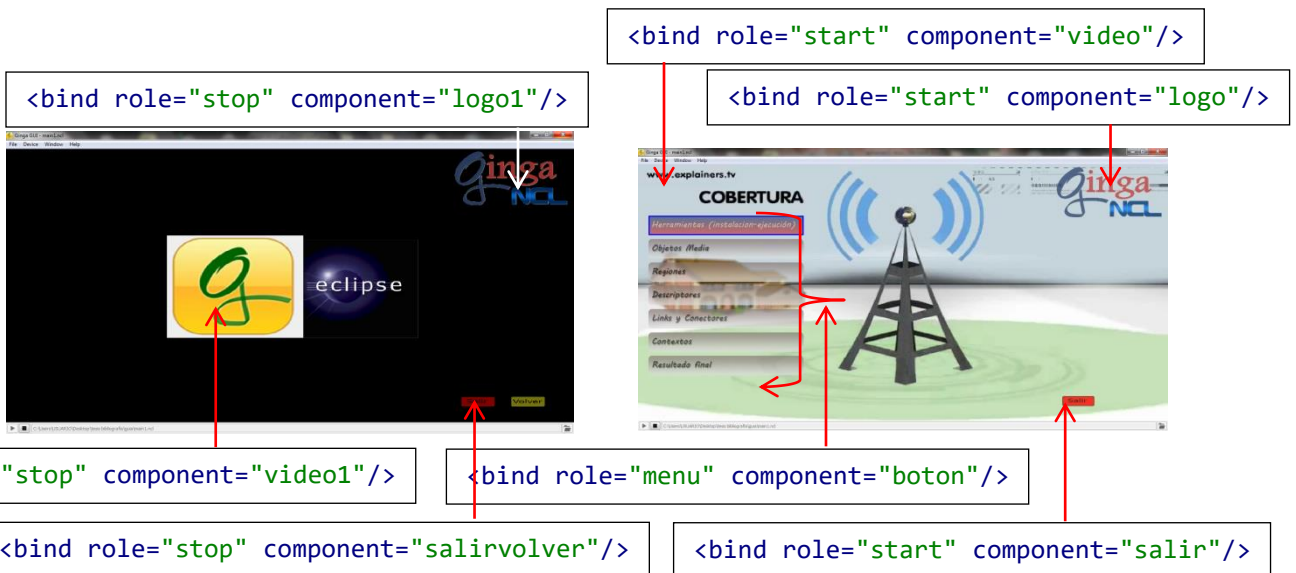
```

Declaración de elementos link que identifican el elemento media (salirvolver) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (video1,salirvolver y logo1) e inicia la presentación de cuatro medias (menu, video, logo y salir).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video1"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



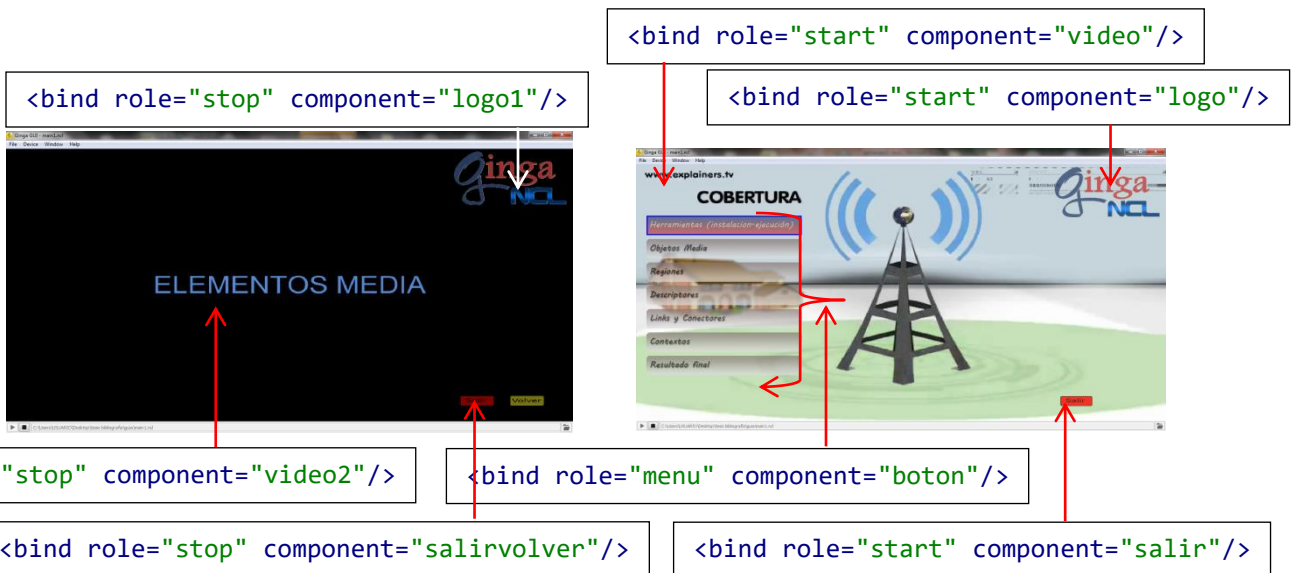
Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (`video2`, `salirvolver` y `logo1`) e inicia la presentación de cuatro medias (`menu`, `video`, `logo` y `salir`).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video2"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



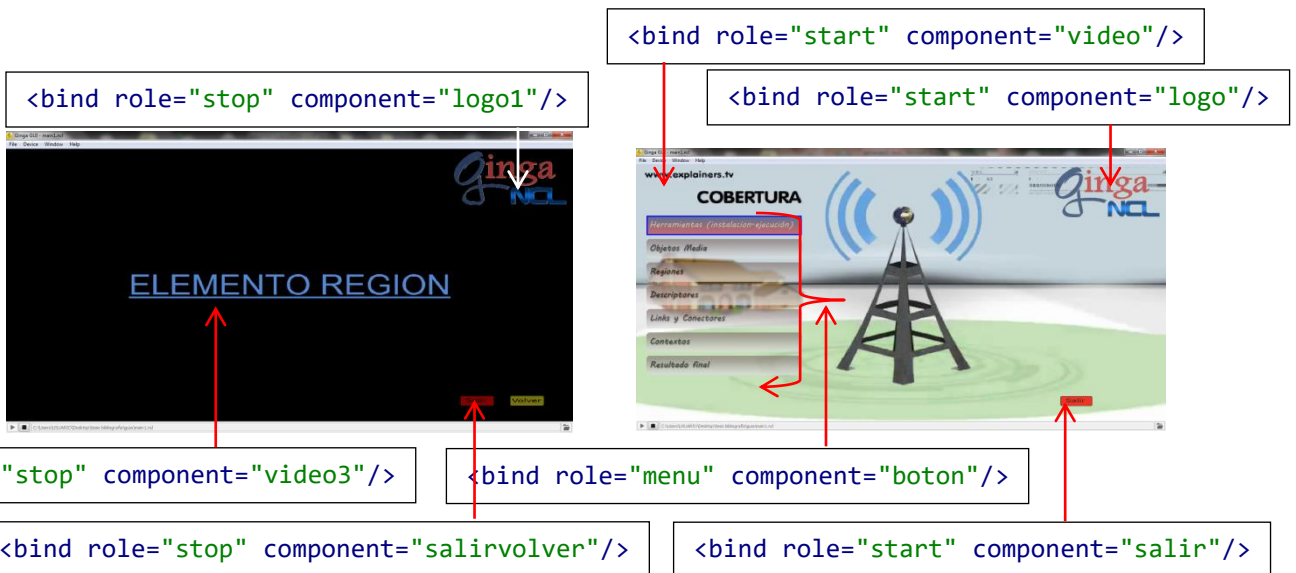
Declaración de elementos link que identifican el elemento media (**salirvolver**) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (**video3**, **salirvolver** y **logo1**) e inicia la presentación de cuatro medias (**menu**, **video**, **logo** y **salir**).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video3"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



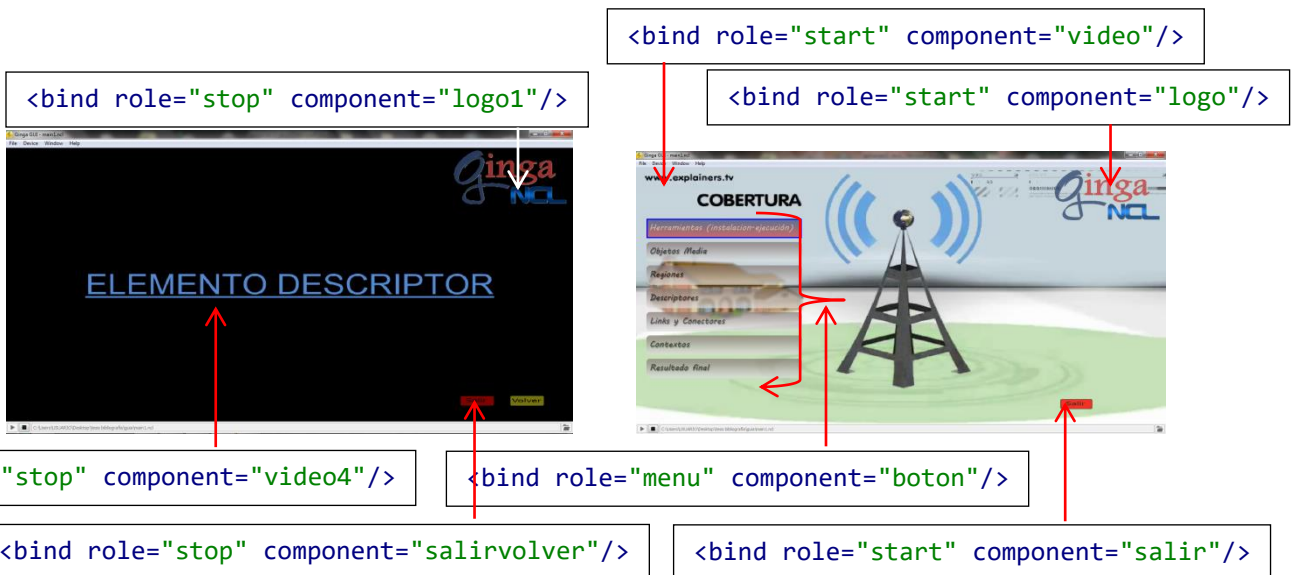
Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (`video4`, `salirvolver` y `logo1`) e inicia la presentación de cuatro medias (`menu`, `video`, `logo` y `salir`).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video4"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (amarillo o F3) del

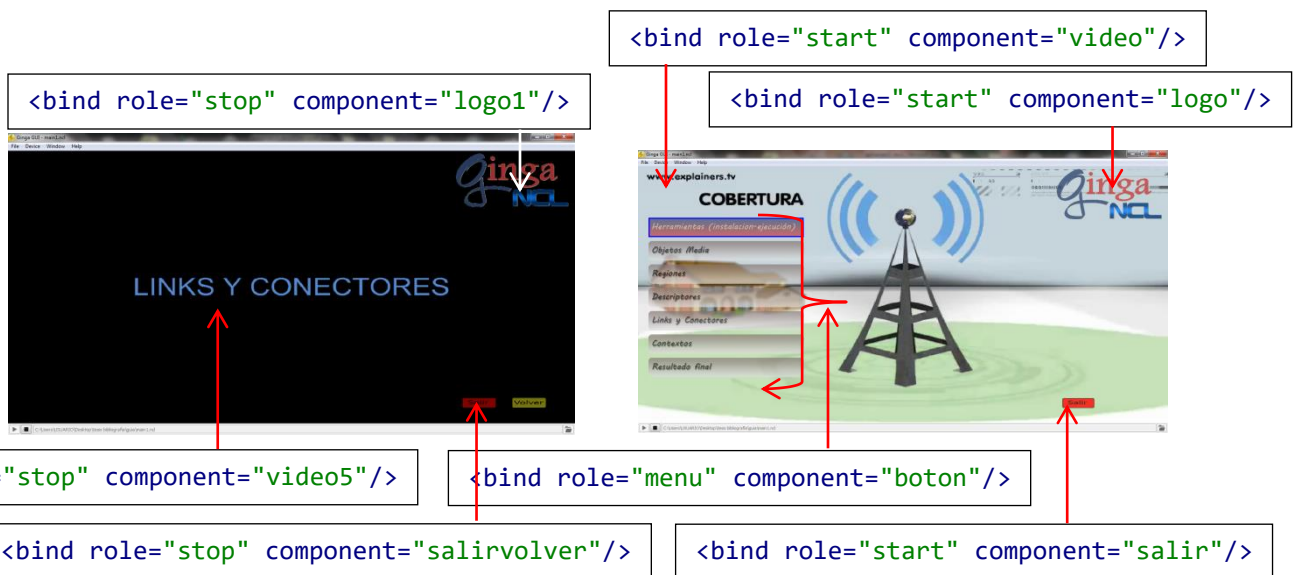
control, lo cual termina la ejecución de tres medias (video5,salirvolver y logo1) e inicia la presentación de cuatro medias (menu, video, logo y salir).

```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video5"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



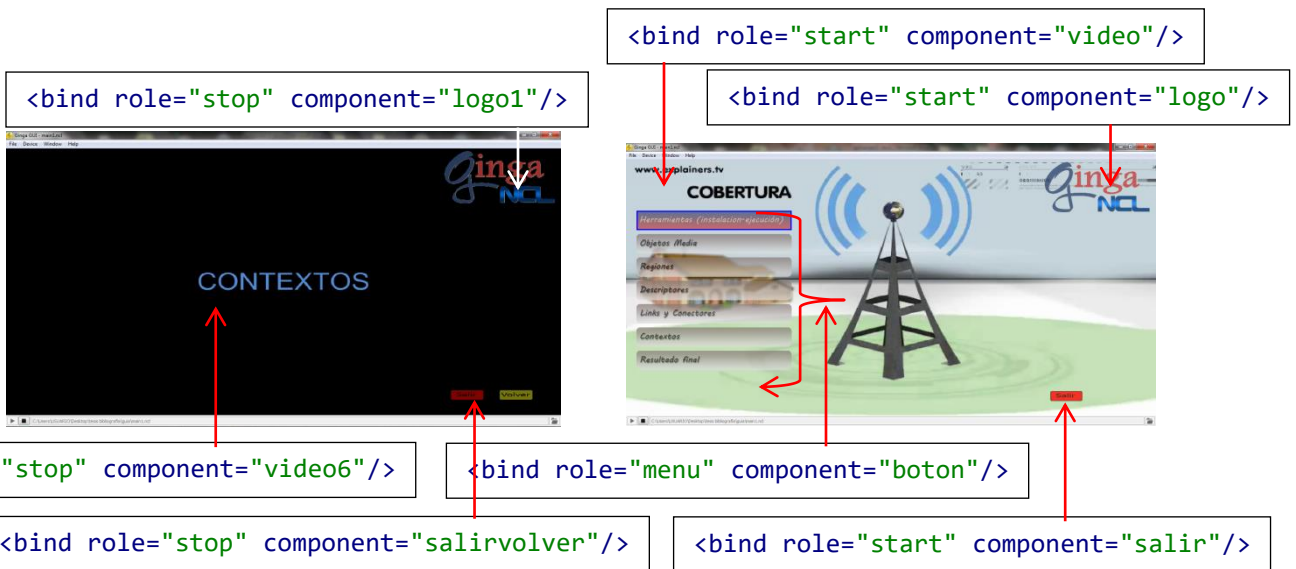
Declaración de elementos link que identifican el elemento media (salirvolver) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (video6,salirvolver y logo1) e inicia la presentación de cuatro medias (menu, video, logo y salir).


```

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video6"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind role="start" component="salir"/>
</link>

```



Declaración de elementos link que identifican el elemento media (`salirvolver`) como un vínculo de interactividad al presionar el botón (amarillo o F3) del control, lo cual termina la ejecución de tres medias (`video7`, `salirvolver` y `logo1`) e inicia la presentación de cuatro medias (`menu`, `video`, `logo` y `salir`).

Antes de declarar las imágenes que se desea que terminen al finalizar el video, hay que tener claro cuáles son las que inician con la siguiente interacción por ejemplo video1, logo1 y salirvolver, estas son medias que no se pueden declarar en esta sección.

```

        <link xconnector="conEx#onEndStopN">
            <bind role="onEnd" component="video"/>

        <bind role="stop" component="logo"/>
        <bind role="stop" component="menu"/>
        <bind role="stop" component="boton"/>
        <bind role="stop" component="salir"/>

        </link>
    </body>
</ncl>

```

CÓDIGO DE PROGRAMACION EN GINGA NCL DE main1.ncl

```

<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
    <head>
        <importedDocumentBase>
        <importNCL documentURI="menuV2.ncl" alias="menuID"/>
        </importedDocumentBase>
        <regionBase>
            <region id="rgScreen" height="100%" width="100%"
zIndex="0"/>
            <region id="rgBoton" left="3%" top="5%" height="10%"
width="10%" zIndex="2" />
            <region id="rgBanner" right="1%" top="1%" height="20%"
width="20%" zIndex="1"/>
            <region id="rgInferior" right="5%" bottom="5%" height="5%"
width="15%" zIndex="1"/>

        </regionBase>
        <descriptorBase>
            <descriptor id="desScreen" region="rgScreen"/>
            <descriptor id="desBanner" region="rgBanner"/>
            <descriptor id="desBoton" region="rgBoton"/>
            <descriptor id="desInferior" region="rgInferior"/>

        </descriptorBase>

```

```

<connectorBase>

  <causalConnector id="onKeySelectionStartNStopN">
    <connectorParam name="bounds"/>
    <connectorParam name="keyCode"/>
    <simpleCondition role="onSelection" key="$keyCode"/>
    <compoundAction operator="seq">
      <simpleAction role="set" value="$bounds"/>
      <simpleAction role="start" max="unbounded"
qualifier="par"/>
      <simpleAction role="stop" max="unbounded"
qualifier="par"/>
    </compoundAction>
  </causalConnector>

  <importBase documentURI="ConnectorBase.ncl" alias="conEx"/>

</connectorBase>
</head>
<body>
  <context id="menu" refer="menuID#menu"/>
  <media id="video" descriptor="desScreen" src="media/backup.mpg">
    <property name="bounds" />
  </media>
  <media id="video1" descriptor="desScreen" src="japo6.mpg">
    <property name="bounds" />
  </media>
  <media id="video2" descriptor="desScreen" src="JAP02.mpg">
    <property name="bounds" />
  </media>
  <media id="video3" descriptor="desScreen" src="japo.mpg">
    <property name="bounds" />
  </media>
  <media id="video4" descriptor="desScreen" src="Japo1.mpg">
    <property name="bounds" />
  </media>
  <media id="video5" descriptor="desScreen" src="JAP03.mpg">
    <property name="bounds" />
  </media>
  <media id="video6" descriptor="desScreen" src="JAP04.mpg">
    <property name="bounds" />
  </media>
  <media id="video7" descriptor="desScreen" src="JAP05.mpg">
    <property name="bounds" />
  </media>

```

```

</media>

<port id="portVideo" component="video"/>
<port id="portlogo" component="logo"/>
<port id="portboton" component="boton"/>

<media id="boton" src="media/info.png"
descriptor="desBoton" >
<property name="transparency" value="50 %"/>
</media>
<media id="salir" src="media/boton_salir.png"
descriptor="desInferior" >
<property name="transparency" value="20 %"/>
</media>
<media id="salirvolver" src="media/boton_salirvolver.png"
descriptor="desInferior" >
<property name="transparency" value="50 %"/>
</media>
<media id="logo" src="logo-ginga.png"
descriptor="desBanner" >
<property name="transparency" value="20 %"/>
</media>
<media id="logo1" src="logo-ginga.png"
descriptor="desBanner" >
<property name="transparency" value="20 %"/>
</media>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="boton">
    <bindParam name="keyCode" value="BLUE"/>
  </bind>
  <bind role="stop" component="boton"/>
  <bind role="start" component="menu"/>
  <bind role="start" component="salir"/>

</link>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

<link xconnector="conEx#onKeySelectionStartNStopN">
  <bind role="onSelection" component="menu"
interface="pMenuItem1"/>

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

```

```

        <bind role="start" component="video1"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>
    </link>

    <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem2"/>

        <bind role="stop" component="video"/>
        <bind role="start" component="video2"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

    </link>
    <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem3"/>

        <bind role="stop" component="video"/>
        <bind role="start" component="video3"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

    </link>

    <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem4"/>

        <bind role="stop" component="video"/>
        <bind role="start" component="video4"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

    </link>

    <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem5"/>

        <bind role="stop" component="video"/>
        <bind role="start" component="video5"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

    </link>

    <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem6"/>

```

```

        <bind role="stop" component="video"/>
        <bind role="start" component="video6"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

</link>

        <link xconnector="conEx#onKeySelectionStartNStopN">
        <bind role="onSelection" component="menu"
interface="pMenuItem7"/>

        <bind role="stop" component="video"/>
        <bind role="start" component="video7"/>
        <bind role="start" component="salirvolver"/>
        <bind role="start" component="logo1"/>

</link>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

<link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="logo">
                <bindParam name="keyCode" value="RED"/>
        </bind>

        <bind role="stop" component="salir"/>
        <bind role="stop" component="menu"/>
        <bind role="start" component="boton"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
                <bindParam name="keyCode" value="RED"/>
        </bind>

        <bind role="stop" component="logo1"/>
        <bind role="stop" component="video1"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="boton"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind component="video" interface="bounds" role="set"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
                <bindParam name="keyCode" value="RED"/>
        </bind>

```

```

    <bind role="stop" component="logo1"/>
    <bind role="stop" component="video2"/>
    <bind role="stop" component="salirvolver"/>
    <bind role="start" component="boton"/>
    <bind role="start" component="video"/>
    <bind role="start" component="logo"/>
    <bind component="video" interface="bounds" role="set"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video3"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video4"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
  <bind role="onSelection" component="salirvolver">
    <bindParam name="keyCode" value="RED"/>
  </bind>

  <bind role="stop" component="logo1"/>
  <bind role="stop" component="video5"/>
  <bind role="stop" component="salirvolver"/>
  <bind role="start" component="boton"/>
  <bind role="start" component="video"/>
  <bind role="start" component="logo"/>
  <bind component="video" interface="bounds" role="set"/>

</link>

```



```

<link xconnector="conEx#onKeySelectionStopNStartN">
<bind role="onSelection" component="salirvolver">
  <bindParam name="keyCode" value="RED"/>
</bind>

<bind role="stop" component="logo1"/>
<bind role="stop" component="video6"/>
<bind role="stop" component="salirvolver"/>
<bind role="start" component="boton"/>
<bind role="start" component="video"/>
<bind role="start" component="logo"/>
<bind component="video" interface="bounds" role="set"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
<bind role="onSelection" component="salirvolver">
  <bindParam name="keyCode" value="RED"/>
</bind>

<bind role="stop" component="logo1"/>
<bind role="stop" component="video7"/>
<bind role="stop" component="salirvolver"/>
<bind role="start" component="boton"/>
<bind role="start" component="video"/>
<bind role="start" component="logo"/>
<bind component="video" interface="bounds" role="set"/>

</link>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->
<link xconnector="conEx#onKeySelectionStopNStartN">
<bind role="onSelection" component="salirvolver">
  <bindParam name="keyCode" value="YELLOW"/>
</bind>

<bind role="stop" component="logo1"/>
<bind role="stop" component="video1"/>
<bind role="stop" component="salirvolver"/>
<bind role="start" component="menu"/>
<bind role="start" component="video"/>
<bind role="start" component="logo"/>
<bind role="start" component="salir"/>

</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
<bind role="onSelection" component="salirvolver">
  <bindParam name="keyCode" value="YELLOW"/>
</bind>

<bind role="stop" component="logo1"/>

```

```

        <bind role="stop" component="video2"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="menu"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind role="start" component="salir"/>
    </link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>

        <bind role="stop" component="logo1"/>
        <bind role="stop" component="video3"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="menu"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind role="start" component="salir"/>
    </link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>

        <bind role="stop" component="logo1"/>
        <bind role="stop" component="video4"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="menu"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind role="start" component="salir"/>
    </link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>

        <bind role="stop" component="logo1"/>
        <bind role="stop" component="video5"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="menu"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind role="start" component="salir"/>
    </link>

    <link xconnector="conEx#onKeySelectionStopNStartN">
        <bind role="onSelection" component="salirvolver">
            <bindParam name="keyCode" value="YELLOW"/>
        </bind>

```

```

        <bind role="stop" component="logo1"/>
        <bind role="stop" component="video6"/>
        <bind role="stop" component="salirvolver"/>
        <bind role="start" component="menu"/>
        <bind role="start" component="video"/>
        <bind role="start" component="logo"/>
        <bind role="start" component="salir"/>
</link>

<link xconnector="conEx#onKeySelectionStopNStartN">
    <bind role="onSelection" component="salirvolver">
        <bindParam name="keyCode" value="YELLOW"/>
    </bind>

    <bind role="stop" component="logo1"/>
    <bind role="stop" component="video7"/>
    <bind role="stop" component="salirvolver"/>
    <bind role="start" component="menu"/>
    <bind role="start" component="video"/>
    <bind role="start" component="logo"/>
    <bind role="start" component="salir"/>
</link>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-->

    <link xconnector="conEx#onEndStopN">
        <bind role="onEnd" component="video"/>

        <bind role="stop" component="logo"/>
    <bind role="stop" component="menu"/>
    <bind role="stop" component="boton"/>
    <bind role="stop" component="salir"/>

    </link>
</body>
</ncl>

```

CÓDIGO DE PROGRAMACION EN GINGA NCL DE menuV2.ncl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="menuV1" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>
<regionBase>
<region id="rgMenu" left="2%" top="20%" width="400"
height="70%" zIndex="99">
<region id="rgMenuItem1" top="0" height="50" zIndex="2" />
<region id="rgMenuItem2" top="60" height="50" zIndex="2" />
<region id="rgMenuItem3" top="120" height="50" zIndex="2"/>
<region id="rgMenuItem4" top="180" height="50" zIndex="2" />
<region id="rgMenuItem5" top="240" height="50" zIndex="2" />
<region id="rgMenuItem6" top="300" height="50" zIndex="2" />
<region id="rgMenuItem7" top="360" height="50" zIndex="2" />

</region>
</regionBase>
<descriptorBase>
<descriptor id="dMenuItem1" region="rgMenuItem1"
focusIndex="1" moveUp="7" moveDown="2"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_herra_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem2" region="rgMenuItem2"
focusIndex="2" moveUp="1" moveDown="3"
```

```

focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_objetos_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem3" region="rgMenuItem3"
focusIndex="3" moveUp="2" moveDown="4"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_regiones_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem4" region="rgMenuItem4"
focusIndex="4" moveUp="3" moveDown="5"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_descriptores_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem5" region="rgMenuItem5"
focusIndex="5" moveUp="4" moveDown="6"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_linksconectores_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem6" region="rgMenuItem6"
focusIndex="6" moveUp="5" moveDown="7"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_contextos_on.png" selBorderColor="aqua"/>
<descriptor id="dMenuItem7" region="rgMenuItem7"
focusIndex="7" moveUp="6" moveDown="1"
focusBorderWidth="-3" focusBorderColor="blue"
focusSrc="boton_resultado_on.png" selBorderColor="aqua"/>
</descriptorBase>
</head>
<body>
  <context id="menu">
    <port id="pMenuItem1" component="menuItem1" />
    <port id="pMenuItem2" component="menuItem2" />
    <port id="pMenuItem3" component="menuItem3" />
    <port id="pMenuItem4" component="menuItem4" />
    <port id="pMenuItem5" component="menuItem5" />
    <port id="pMenuItem6" component="menuItem6" />
    <port id="pMenuItem7" component="menuItem7" />
    <media id="menuItem1" src="boton_herra_off.png"
descriptor="dMenuItem1" >
    <property name="transparency" value="30 %"/>
    </media>
    <media id="menuItem2" src="boton_objetos_off.png"
descriptor="dMenuItem2" >
    <property name="transparency" value="30 %"/>
    </media>
    <media id="menuItem3" src="boton_regiones_off.png"
descriptor="dMenuItem3" >
    <property name="transparency" value="30 %"/>
    </media>
    <media id="menuItem4" src="boton_descriptores_off.png"
descriptor="dMenuItem4" >
    <property name="transparency" value="30 %"/>
    </media>
    <media id="menuItem5" src="boton_linksconectores_off.png"
descriptor="dMenuItem5" >
    <property name="transparency" value="30 %"/>
    </media>
    <media id="menuItem6" src="boton_contextos_off.png"
descriptor="dMenuItem6" >

```

```
<property name="transparency" value="30 %"/>
</media>
<media id="menuItem7" src="boton_resultado_off.png"
descriptor="dMenuItem7" >
<property name="transparency" value="30 %"/>
</media>
</context>
</body>
</ncl>
```