ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA

# IMPLEMENTACIÓN DE UN TRANSMISOR DE PRUEBAS DE TV DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE APLICACIONES INTERACTIVAS

MARÍA AUGUSTA ILLESCAS ROBALINO

DIEGO FERNANDO VILLAMARÍN ZAPATA

SANGOLQUÍ – ECUADOR
2011

## CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado: “IMPLEMENTACIÓN DE UN TRANSMISOR DE PRUEBAS DE TV DIGITAL TERRESTRE ISDB-Tb, PARA LA EMISIÓN DE APLICACIONES INTERACTIVAS”, ha sido desarrollado en su totalidad por la señorita MARIA AUGUSTA ILLESCAS ROBALINO con CI: 171967379-8 y el señor DIEGO VILLAMARIN con CI: 171659811-3, bajo nuestra dirección.

---

Ing. Gonzalo Olmedo Ph.D.
**DIRECTOR**

---

Ing. Román Lara. M.Sc.
**CODIRECTOR**

# RESUMEN

En el presente proyecto se implementó un transmisor de televisión digital terrestre ISDB-Tb para pruebas de laboratorio, conformado por las etapas de generación, modulación y emisión de la señal de audio, video y datos para aplicaciones interactivas.

En la primera etapa, el flujo de datos (TS – Transport Stream) fue generado utilizando el software de distribución libre OpenCaster. Para la etapa de modulación se utilizo un equipo adecuado para esto. La transmisión se realiza por la interfaz que posee el equipo más una epata de amplificación.

El flujo de datos fue generado para dos canales con contenidos y aplicaciones diferentes, considerando definición alta y estándar. Para las pruebas de recepción se utilizaron un televisor LCD de alta calidad y un Set Top Box ISDB-Tb de desarrollo con middleware Ginga.

## DEDICATORIA

*Este proyecto de grado lo dedico con todo mi amor y cariño, a Dios por darme la oportunidad de vivir, regalarme una hermosa familia y darme fortaleza y sabiduría necesaria para terminar este proyecto, a mi familia por acompañarme en cada una de las etapas de mi vida, a mi madre por darme la vida, y quien con su simpleza me ha ensenado a ser una mujer luchadora, a mi hermano Edward por ser más que un hermano por brindarme su apoyo, confianza y amor y finalmente a todos mis amigos que han formado parte de mi vida durante todos estos años.*

**María Augusta Illescas Robalino**

*Este proyecto de tesis va dedicado con mucho amor para mi familia, especialmente a mis padres porque con éxito han sabido guiarme y apoyarme en el transcurso de mi formación profesional, y con mucho afecto a mis amigos y profesores que colaboraron en este proyecto.*

**Diego Fernando Villamarín Zapata**

# AGRADECIMIENTOS

Primero quiero agradecer a Dios por darme las fuerzas necesarias en los momentos en que más las necesité y bendecirme con años maravillosos de vida.

Para lograr culminar esta carrera y ahora mi proyecto de grado fue necesario el apoyo de muchas personas a las cuales me gustaría agradecer su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar donde estén quiero darles las gracias por formar parte de mí.

Mamita Luz, te agradezco primero por darme la vida, por brindarme tu apoyo aun cuando estabas lejos, por tenerme paciencia todos los días, y por no quitarme tu apoyo nunca a pesar de los momentos difíciles que hemos pasado.

Agradezco a mi hermano Edward de todo corazón por creer en mí, por ser como un padre y brindarme su apoyo moral y económico incondicional para lograr cumplir todas mis metas, por estar a mi lado apoyándome y aconsejándome siempre para escoger un buen camino, gracias a él soy la mujer que siempre me imagine ser.

A mi hermana y mi sobrina por estar siempre a mi lado, contagiándome con su alegría y sus ocurrencias por ser como son y por todo el cariño que me han brindado todos estos años.

A mi cuñada Olga por su confianza y apoyo durante varias etapas de mi vida y por inculcarme a seguir adelante con mis sueños.

A mis queridas amigas Fisihuek's, Gaby, Mell, Lily y Carlita por su amistad incondicional por estar en las buenas y en las malas siempre. Por todos estos años de una amistad sincera, sin su apoyo no hubiera podido culminar mi carrera y ahora cumplir esta meta.

A Diego mi compañero de tesis, por compartir buenos y malos momentos durante todo este tiempo mientras desarrollábamos la tesis, por tenerme paciencia y por hacerme sonreír casi todos los días.

A mis compañeros de toda la carrera David, Henry y Julio quienes me acompañaron en esta trayectoria de aprendizaje y conocimientos sobre todo por hacer de esta carrera más llevadera con su amistad.

A todos mis amigos que han sido parte de mi vida durante la vida universitaria, por sus consejos por su apoyo por estar pendientes de mí, gracias por su amistad los quiero mucho.

A mi director de tesis Gonzalo Olmedo, por depositar su confianza en Diego y en mi para el desarrollo de este proyecto, por enseñarnos hacer las cosas con honestidad y humildad.

A mi codirector de tesis Román Lara por su apoyo, y por sus consejos para que este proyecto haya sido realizado de la mejor manera.

**María Augusta Illescas Robalino**

Agradezco en primer lugar a Dios por darme sabiduría y humildad para luchar con perseverancia para alcanzar mis objetivos, y por la bendición de tener a mi familia.

A mis padres, por el amor, cariño, comprensión, motivación y ejemplo que siempre me han brindado y el apoyo incondicional en los momentos difíciles que he pasado en mi vida Universitaria.

A mis hermanos Mayra y Guillo por su ayuda y compañía.

A toda mi familia por el cariño brindado y el ejemplo que me incentivó para completar mis estudios.

A mis amigos “Chulla Vida” que compartieron este largo y difícil camino en esta carrera, por su ayuda y amistad sincera, a mi compañera de tesis Magus, a mis compañeros y en especial a Gaby por todo lo que compartimos y por el apoyo que me brindo.

A todos mis profesores por las enseñanzas y la amistad brindada, especialmente a mi orientador de tesis el Dr. Gonzalo Olmedo por la paciencia y confianza que ha depositado en mí.

**Diego Fernando Villamarín Zapata**

# PROLOGO

La TV digital promete revolucionar la forma de ver televisión, mejorar la calidad de imagen, interactividad con el televidente, más opciones de programas, y recepción en dispositivos móviles son algunas de las nuevas posibilidades ofrecidas. Para que estas posibilidades sean realidad, fue necesaria la utilización de nuevas teorías, tecnologías y herramientas. Este trabajo ofrece una referencia inicial para quienes quieran empezar en la generación y desarrollo de contenidos interactivos con la implementación de un transmisor de bajo costo.

La adopción del estándar ISDB-Tb por Ecuador, ofrece una excelente oportunidad para que el desarrollo se lo realice en conjunto entre el estado, la universidad y la industria. Con este trabajo se busca tener un incentivo para que eso suceda.

En los próximos años un proceso de evolución debe suceder en las emisoras de televisión para que estas de acoplen a la nueva era de televisión que se inician con la transmisión de tv digital abierta en el país, para esto la universidad debe estar preparada con el fin de poder apoyar con accesorias y pruebas de equipos en el tiempo que dure la transición.

La Escuela Politécnica del Ejercito desde el año 2006 ha venido contribuyendo en el tema de TV Digital en el Ecuador, gracias a sus investigadores, por tal motivo se creó un transmisor de pruebas con el fin de continuar con la investigación y desarrollo del sistema de televisión digital. Siendo así los pioneros en realizar pruebas reales de transmisión con aplicaciones interactivas, bajo el estándar ISDB-Tb en el Ecuador.

# INDICE GENERAL

## CAPÍTULO 1

### INTRODUCCIÓN


1.1 MOTIVACIÓN....21

1.2 CONTRIBUCIÓNES....23

1.3 ESTRUCTURA DEL PROYECTO DE GRADO....23


## CAPÍTULO 2

### IMPLEMENTACION DEL LABORATORIO DE TV DIGITAL


2.1 REQUERIMIENTOS DE HARDWARE Y SOFTWARE....

2.1.1 Generación TS....

2.1.2 Modulación TS....

2.1.3 Radio Frecuencia....

2.1.4 Receptor....

2.2 INSTALACIÓN DE SOFTWARE....

2.2.1 Instalación OpenCaster2.4....

2.2.2 Instalación StreamXpress TS Player....


## CAPÍTULO 3

### GENERACIÓN DEL TRANSPORT STREAM


3.1 TRANSPORT STREAM (TS)....25

3.2 ESTRUCTURA DEL TRANSPORT STREAM....26

3.2.1 Codificación....27

3.2.2 Paquetización....27

3.2.3 Formación de Paquetes TS....28

3.2.4 Multiplexación....30

3.2.5 Tablas de Información del Sistema .....30
3.2.6 Tablas PSI/SI .....31
3.3 GENERACION DEL TRANSPORT STREAM .....34
3.3.1 Formatos de Audio y Video .....34
3.3.2 Resolución y Relación de Aspecto de Pantalla .....35
3.3.3 Codificador .....36
3.3.4 Transport Stream con un Servicio .....39
3.3.5 Transport Stream con un Servicio e Interactividad .....54
3.3.6 Transport Stream con dos Servicios e Interactividad .....

# CAPÍTULO 4

## TRANSMISIÓN Y PRUEBAS DE RECEPCIÓN

4.1 TRANSMISIÓN EN EL ESTANDAR ISDB-Tb .....63
4.1.1 Re-Multiplexor .....66
4.1.2 Segmentación del Canal y Capas .....66
4.1.3 Modos de Operación .....68
4.1.4 Aleatorizador o Dispersión de Energía .....68
4.1.5 Ajuste de Retraso .....69
4.1.6 Entrelazador Externo .....69
4.1.7 Código Interno .....70
4.1.8 Entrelazador de Bits .....72
4.1.9 Mapeamiento .....72
4.1.10 Combinador De Capas Jerárquicas .....75
4.1.11 Entrelazador de Tiempo .....76
4.1.12 Entrelazador de Frecuencia .....77
4.1.13 Señal OFDM .....77
4.1.14 Capacidad de Transmisión del Sistema .....81
4.2 Configuración del Transmisor .....
4.2.1 Transmisión de un Servicio de TV Digital .....
4.2.2 Transmisión de un Servicio de TV Digital con Interactividad .....

4.2.3 Transmisión de dos servicios de TV Digital con Interactividad .....
4.3 PRUEBAS DE RECEPCIÓN .....
4.3.1 Configuración del Set Top Box (STB) .....
4.3.2 Recepción de un Servicio De Tv Digital .....
4.3.3 Recepción de un Servicio de Tv Digital con Interactividad .....83
4.3.4 Recepción de dos Servicios de Tv Digital con Interactividad .....86

# CAPÍTULO 5

## CONCLUCIONES Y RECOMENDACIONES

5.1 CONCLUCIONES .....88
5.2 RECOMENDACIONES .....89

# INDICE DE FIGURAS

Figura. 2.1. Estructura del Sistema de TV Digital en el Laboratorio .....
Figura. 2.2. Instalacion en Servidor .....
Figura. 2.3. Generador de Señales .....
Figura. 2.4. Analizador de Espectros, pérdida del cable de medida. .....
Figura. 2.5. Potencia de salida .....
Figura. 2.6. Ganancia real a la salida del transmisor .....
Figura. 2.7 Diagrama de potencias en la transmisión .....
Figura. 2.8 Gráfica de la Distancia Vs la Sensibilidad del Receptor .....
Figura. 2.9. Descompresión de archivos de instalación .....
Figura. 2.10. Modificación de archivos .....
Figura. 2.11. Compilación de archivos .....
Figura. 2.12. Instalación de archivos .....
Figura. 2.13. Comprobación de la Instalación .....
Figura. 2.14. Archivo descomprimido StreamXpressSetup.exe .....
Figura. 2.15. Instalación .....

Figura. 2.16. Datos del usuario.....
Figura. 2.17. Tipo de Instalación .....
Figura. 2.18. Confirmar la Instalación .....
Figura. 2.19. Proceso de Instalación.....
Figura. 2.20. Instalación Finalizada .....
Figura. 2.21. Icono de .....
Figura. 2.22. Interface Gráfica .....
Figura. 2.23. Interface Gráfica .....
Figura. 2.24. Interface Gráfica .....
Figura. 3.1. Estructura *Transport Stream* .....25
Figura. 3.2. Formación de ES, PES y TS.....26
Figura. 3.3. Cabecera paquete TS.....28
Figura. 3.4. Multiplexación de dos TS .....30
Figura. 3.5. Estructura de un TS definido por las Tablas PAT y PMT.....32
Figura. 3.6. Propiedades de Video .....37
Figura. 3.7 Información del video fuente.....38
Figura. 3.8 Transformación a formato .AVI.....39
Figura. 3.9 Proceso de generación de Transport Stream con un servicio.....39
Figura. 3.10 Codificación de Audio y Video por separado .....40
Figura. 3.11 Paquetización del Video.....42
Figura. 3.12 Paquetización del Audio.....43
Figura. 3.13 Proceso de Multiplexación.....52
Figura. 3.14. Generación Transport Stream con un Servicio e Interactividad .....54
Figura. 3.15 Proceso de Multiplexación con Interactividad.....62
Figura. 3.16. Generación Transport Stream con dos Servicios e Interactividad .....
Figura. 3.17 Proceso de Multiplexación con Interactividad.....
Figura. 4.1. Capa física de los sistemas de TV Digital. .....63
Figura. 4.2. Diagrama de bloques de un Transmisor ISDB-Tb.....66
Figura. 4.3. Re-multiplexor y Codificador RS.....66
Figura. 4.5. Transmisión Jerárquica Tres Capas Activas.....67
Figura. 4.6. Generador de secuencias pseudo-aleatorias. .....69
Figura. 4.7. Estructura del Entrelazador de bytes.....70

Figura. 4.8. Codificador convolucional de tasa 1/2 ..... 71
Figura. 4.9. Estructura de los entrelazadores de bits ..... 72
Figura. 4.10. Diagrama de constelación DQPSK ..... 73
Figura. 4.11. Diagrama de constelación QPSK ..... 73
Figura. 4.12. Diagrama de constelación 16QAM ..... 74
Figura. 4.13. Diagrama de constelación 64QAM ..... 74
Figura. 4.14. Ejemplo de organización de datos en el combinador ..... 75
Figura. 4.15. Ejemplo del entrelazamiento temporal sobre un ruido impulsivo ..... 76
Figura. 4.16. Ejemplo del entrelazamiento en frecuencia ..... 77
Figura. 4.17. Tiempo de guarda ..... 78
Figura. 4.18. Formación en bits del cuadro TMCC ..... 80
Figura. 4.19. Transmisión de un servicio de tv digital .....
Figura. 4.20. Parámetros para emitir un servicio de tv digital .....
Figura. 4.21. Transmisión de un servicio de tv digital con interactividad .....
Figura. 4.22. Parámetros para emitir un servicio de tv digital con interactividad .....
Figura. 4.23. Transmisión de dos servicio de tv digital con interactividad .....
Figura. 4.24. Parámetros para emitir dos servicios de tv digital con interactividad .....
Figura. 4.25. Menú del STB .....
Figura. 4.26. Submenú canales .....
Figura. 4.27. Proceso de Sintonización .....
Figura. 4.28. Recepción de un servicio de TV digital .....
Figura. 4.29. Proceso de carga de una aplicación interactiva ..... 84
Figura. 4.30. Lista de Aplicaciones Interactivas disponibles ..... 85
Figura. 4.31. Aplicación Interactiva Guía Hospitalaria ..... 85
Figura. 4.32. Lista de Canales Disponibles ..... 86
Figura. 4.33. Aplicación APP_GINGA2 disponible en el canal ESPETV2 ..... 86
Figura. 4.34. Guía Turística de la Ciudad de Ibarra ..... 87

# INDICE DE TABLAS

Tabla. 2.1. Requerimientos Generación TS .....
Tabla. 2.2. Requerimientos Modulación TS .....

Tabla. 2.3. Requerimientos Radio Frecuencia.....
Tabla. 2.4. Requerimientos Receptor.....
Tabla. 2.5. Archivos OpenCaster .....
Tabla. 3.1. Tabla de PID's utilizados para la transmisión de Tablas del Sistema .....31
Tabla. 3.2. Tablas PSI principales utilizadas en el sistema ISDB-Tb.....33
Tabla. 3.3. Tablas SI principales utilizadas en el sistema ISDB-Tb.....33
Tabla. 3.4 Formatos de video en ISDB-Tb .....36
Tabla. 3.5 Parámetros codificación video .....40
Tabla. 3.6 Parámetros codificación audio .....41
Tabla. 3.7 Parámetros esvideo2pes .....42
Tabla. 3.8 Parámetros pesvideo2pes .....43
Tabla. 3.9 Parámetros esaudio2pes .....44
Tabla. 3.10 Parámetros esaudio2pes .....44
Tabla. 3.11. Script de la tabla para un servicio de televisión digital .....45
Tabla. 3.12 Secciones y Paquetes TS generados por las tablas.....51
Tabla. 3.13. Parámetros tscbrmuxer .....53
Tabla. 3.14. Script de la tabla para un servicio de televisión digital con interactividad .....54
Tabla. 3.15. Parámetros para la generación de interactividad .....61
Tabla. 3.16. Asignación de PID's para los programas.....
Tabla. 3.17. Script de la tabla para dos servicio de televisión digital con interactividad.....
Tabla. 4.1. Ejemplo de configuración recomendada.....67
Tabla. 4.2. Modos de Operación .....68
Tabla. 4.3. Ajuste de atraso en Bits .....69
Tabla. 4.4. Punteo Codificador Convolucional .....71
Tabla. 4.5. Atrasos en el entrelazamiento temporal.....76
Tabla. 4.6. Duración de símbolo y de cuadro OFDM.....79
Tabla. 4.7. TMCC 102 bits de información del sistema .....81
Tabla. 4.8. Tasa útil de datos en 13 segmentos .....82
Tabla. 4.9. Resumen parámetros del trasmisor ISDB-T.....82

# GLOSARIO

**AAC** *Advanced Audio Coding* - Tipo de Codificación de Audio.

**AC** *Auxiliary Channel* – Canal Auxiliar.

**AIT** Aplication Information Table – Tabla de Información de Aplicaciones.

**AVI** *Audio Video Interleaved* - Formato de Audio y Video Entrelazado

**ATSC** *Advanced Television System Committee* – Comité de Sistemas de Televisión Avanzados.

**BIT** *Broadcast Information Table* – Tabla de información del Radiodifusor.

**BML** Broadcast Markup Language – Lenguaje de Marcas 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 Frecuencias Ortogonales Codificadas

**CP** *Continuous Pilot* - Pilotos Continuos.

**CRC** Cyclic Redundancy Check – Verificación de Redundancia Cíclica

**DSMCC** *Digital Storage Media Command and Control* – Medios digitales de almacenamiento de Mando y Control

**DQPSK** *Differential Quadrature Phase Shift Keying* – Modulación por desplazamiento de fase diferencial.

**DTMB** *Digital Terrestrial Multimedia Broadcast* – Radiodifusión Digital Terrestre de Multimedia.

**DVB** *Digital Video Broadcasting* - Radiodifusión de Vídeo Digital.

**ES** *Elementary Stream* - Flujo Elemental.

**EDTV** *Enhanced Definition Television* – Televisión Digital Mejorada

**EIT** *Event Information Table* - Tabla de Información de Eventos.

**EPG** *Electronic ProgramGuide* – Guía de Programación electrónica.

**FTP** *File Transfer Protocol* – Protocolo de transferencia de archivos.

**HD** *High-definition* - Alta Definición.

**HDTV** *High Definition Television* - Televisión de Alta Definición.

**IFFT** *Inverse Fast Fourier Transform* - Transformada Rápida Inversa de Fourier.

**IIP** *ISDB-TInformation Packet* – Paquete de Informaciones de ISDB-T.

**IPTV** *Internet Protocol Television*- Televisión sobre un protocolo de Internet.

**ISDB-T** *Integrated Services Digital Broadcasting-Terrestrial* - Servicios Integrados de Radiodifusión Digital - Terrestre.

**ISDB-TB** *Integrated Services Digital Broadcasting-Terrestrial Brazil* - Servicios Integrados de Radiodifusión Digital - Terrestre Brasil.

**LCD** *Liquid Crystal Display* – Pantalla de cristal liquido.

**LD** *Low-definition* - Baja Definición.

**LDT** *Linked Descriptor Table* – Tabla de descriptores de enlaces.

**LDTV** *Low-definition television* - Televisión de Baja Definición.

**MPEG** *Moving Picture Experts Group* - Grupo de Expertos en Imágenes Dinámicas.

**NBIT** *Network Board Information Table* – Tabla de Información de la tarjeta de Red

**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 Programas.

**PCAT** Partial Content Announcement Table – Tabla de anuncio de contenido parcial.

**PCI** *Peripheral Component Interconnect* – Interconexión de Componentes Periféricos

**PCR** *Program Clock Reference* - Referencia de Reloj para los Programas.

**PES** *Packetized Elementary Stream* - Flujo Elemental Empaquetado.

**PID** *Packet Identifier* - Identificador do Paquetes.

**PMT** *Program Map Table* – Tabla de Mapeamiento de Programas

**PSI** *Program Specific Information* - Tabla de Información de los Programas.

**PTS** *Presentation Time Stamps* - Estampas de Tiempo de Presentación.

**QAM** *Quadrature Amplitude Modulation* - Modulación en Amplitud y Cuadratura.

**QPSK** *Quadrature Phase Shift Keying* - Modulación de Fase en Cuadratura.

**RAM** *Random Access Memory* - Memoria de Acceso Aleatorio.

**RF** *Radio Frequency* – Radio Frecuencia.

**RS** *Reed Solomon* - Tipo de Código Corrector de Error.

**RST** *Running Status Table* – Tabla del estado que se está ejecutando.

**SATVD** Sistema Argentino de Televisión Digital.

**SDT** *Service Description Table* - Tabla de Descripción de Servicios.

**SDTV** *Standard Definition Television* - Televisión de Definición Estándar.

**SI** *Service Information* - Tablas de Información del Sistema.

**SP** *Scattered Pilots* - Pilotos Dispersos.

**STB** *Set Top Box* – Decodificador para TV Digital

**TCP** *Transfer Control Protocol* - Protocolo de Control de Transferencia.

**TDT** *Time and Date Table* – Tabla de Fecha y Hora

**TMCC** *Transmission and Multiplexing Configuration Control* – Control y Configuración de Multiplexación y Transmisión.

**TOT** *Time Offset Table* - Tabla de Cambio de Horario.

**TS** *Transport Stream* - Flujo de Transporte.

**UIT** Unión Internacional de Telecomunicaciones.

**UHF** *Ultra High Frequency* –Frecuencias Ultra Altas

**VSB** *Vestigial Side Band* – Modulación de banda lateral vestigial.

# CAPÍTULO 1

## INTRODUCCIÓN

Según la forma de recepción de la señal de televisión digital en el hogar, se puede hablar de Televisión Digital por Satélite, Televisión Digital por Cable o Televisión Digital Terrestre. Las dos primeras tecnologías están presentes en Ecuador, únicamente brindadas por empresas privadas.

El 26 de marzo del 2010 se adoptó oficialmente el estándar japonés-brasileño ISDB-Tb (*InternationalSystem Digital Broadcasting, Terrestrial, Brazilian version*) para la televisión digital terrestre en nuestro país, ofreciendo televisión abierta gratuita con mejores prestaciones[1].

La televisión digital implica una forma distinta de transmitir la señal televisiva, en referencia a la televisión analógica actual.

La imagen, el audio y los datos se convierten en información digital, permitiendo que los errores en la transmisión se corrijan y no existan interferencias ni distorsiones en pantalla. De esta manera, queda garantizada una óptima calidad de imagen y sonido y además de habilitarse la transmisión de servicios de alta definición (HD – *High Definition*) [2].

Las normas de la televisión digital consiguen aumentar la oferta de programas y permiten ofrecer servicios interactivos a través de los cuales se puede interactuar con la programación televisiva, acceder a información acerca de los contenidos

transmitidos con múltiples señales de audio, teletexto, guía electrónica de programas (EPG – *Electronic Program Guide*), canales de radio, servicios interactivos, imagen panorámica y también permite desarrollar plataformas avanzadas para atender temáticas vinculadas con la educación, la salud, los derechos, la cultura, la religión, el entretenimiento, etc. [3]

La televisión digital terrestre está operando mundialmente con cuatro estándares.La plataforma usada en Norteamérica y algunos países centroamericanos es ATSC; ISDB-T en Japón y Filipinas.ISDB-Tb (variante del ISDB-T) en Brasil y la mayoría de los países latinoamericanos: Perú, Argentina, Chile, Venezuela, Ecuador, Costa Rica, Paraguay, Bolivia, Nicaragua y Guatemala, con la excepción de Colombia, Guyana, Suriname, Panamá, Honduras, El Salvador y México. DTMB en la República Popular China, Hong Kong y Macau. DVB-T en los países europeos, Australia, partes de África y países de América Latina: Colombia, y Panamá. El resto del mundo aún no se ha decidido.

A partir del 2000, el gobierno de Brasil a través de las universidades del país desarrolló un estudio sobre la televisión digital y las ventajas y retos que conlleva realizar esta transición. En noviembre del 2003 se creó el Sistema Brasilero de TV Digital (SBTVD), actualmente denominado ISDB-Tb**,** que inició sus servicios comerciales y públicos el 2 de diciembre de 2007 en la ciudad de Sao Paulo - Brasil [4].Se diferencia básicamente de la norma japonesa por el uso del códec MPEG-4 (H.264) para compresión de vídeo, en lugar de MPEG-2 (*Moving Picture ExpertsGroup*) usado en ISDB-T, compresión de audio con HE-AAC (*High-Efficiency Advanced Audio Coding),* modulación en BST-OFDM-TI, presentación de 30 cuadros por segundo incluso en dispositivos portátiles, a diferencia de los 15 cuadros por segundo para equipos móviles en la norma ISDB-T e interacción utilizando el middleware o software de soporte de aplicaciones distribuidas o intermediario, desarrollado en Brasil y denominado Ginga, compuesto por los módulos *Ginga-NCL*, usado para exhibir documentos en lenguaje NCL (*Nested Context Language*) y *Ginga-J* para aplicaciones escritas en lenguaje Java. En el

caso de la norma original ISDB-T, este software es el BML (*Broadcast Markup Language*). Esto ocasiona que los receptores ISDB-T no sean compatibles con las señales desarrolladas para la norma ISDB-Tb, aunque éstos últimos si son compatibles con los de la versión original. Además, es posible utilizar SBTVD/ISDB-Tb en 6 Mhz, 7 MHz o 8 MHz si es requerido porque el sistema es totalmente compatible.

En enero de 2009, la Agencia Brasileño-Japonesa, grupo de estudio para la televisión digital, terminó y publicó un documento de adhesión a la especificación ISDB-T con el brasileño SBTVD, resultando en una especificación que ahora se llama ISDB-T Internacional o ISDB-Tb [5].

El 29 de abril de 2009 ISDB-Tb fue certificado oficialmente por la Unión Internacional de Telecomunicaciones (UIT) tanto el módulo de Ginga-NCL como el lenguaje NCL/Lua, desarrollado por la Universidad Católica de Río, como primera recomendación internacional para entornos multimedia interactivos para TV Digital y IPTV-Recomendación H.761 [6].

Actualmente en nuestro país, debido a la donación de un transmisor digital de parte del gobierno de Japón, el canal del estado ECUADOR TV, está transmitiendo en la frecuencia del canal 47 que está en el rango de 668MHz a 674MHz, dos canales digitales en calidad estándar SD (Standard Definition) y un canal para receptores móviles 1SEG. (*One Segment*), para recepción en aparatos móviles; el primer canal de definición estándar y el de 1SEG., tiene la programación de ECUADOR TV y el segundo canal de definición estándar, presenta programas educativos japoneses.

## 1.1 MOTIVACIÓN

En la actualidad se está viviendo la transformación global del servicio de televisión terrestre radiodifundida, en donde la televisión digital representa la

revolución en la transmisión de programas junto a una gran flexibilidad en los contenidos emitidos, siendo posible mezclar un número determinado de canales de video, audio y datos en una sola señal. Permitiendo al televidente interactuar con su televisión ya sea en equipos fijos, portátiles o móviles, con la característica que el televidente ahora puede disfrutar de una señal mucho más robusta, sin ruidos, interferencias, ni doble imagen, perfeccionando de este modo los contenidos que el espectador está visualizando.

El desarrollo de la televisión digital terrestre posibilitará el acceso de los sectores de población de escasos recursos económicos, a redes y servicios de telecomunicaciones a los cuales no es posible acceder actualmente.

En nuestro país el índice de penetración de la televisión es altísimo ya que el 90% de la población ecuatoriana accede a la televisión abierta [7].

Por las razones antes mencionadas, la Escuela Politécnica del Ejército previo a la adopción del estándar de televisión digital terrestre para Ecuador, participó con el grupo de investigación de televisión digital, en las pruebas realizadas con los diferentes estándares.

La ESPE como apoyo a este grupo de investigación, adquirió equipos con los cuales se realizaran pruebas tanto de transmisión como de recepción de la señal de televisión digital.

En el presente proyecto de grado se presenta la implementación del transmisor de televisión digital con el estándar ISDB-Tb para el laboratorio de TV Digital de la ESPE.

La ESPE se ha transformado en la universidad pionera del país, en lo que se refiere a realizar estudios y pruebas con el estándar ISDB-Tb. Además, de ser la primera universidad en tener un transmisor de televisión digital, permitiendo llegar

masivamente a la comunidad politécnica con información mediante la TV. Adicionalmente, se amplía un abanico único de posibilidades para desarrollar *T-Learning*(*Transformative Learning*) como potencial educativo que ofrece la televisión digital interactiva.

## 1.2 CONTRIBUCIÓNES

Este proyecto contempla el estudio de las modificaciones, requisitos y características tanto de software como de hardware y la implementación de un transmisor de laboratorio para televisión digital interactiva bajo el estándar ISDB-Tb, mediante el uso de un equipo especial adquirido por la ESPE. Se finalizará realizando pruebas reales de la recepción, dela emisión de un flujo de datos que contenga aplicaciones interactivas desarrolladas en GINGA NCL, en un STB (Set Top Box) que admite el estándar ISDB-Tb.

## 1.3 ESTRUCTURA DEL PROYECTO DE GRADO

Este trabajo está estructurado en 5 capítulos, a continuación se presenta un breve resumen de cada uno:

En el Capítulo 2 se detallan todos los equipos que forman parte de la implementación del Laboratorio de Televisión Digital en la ESPE, se describen los requerimientos de hardware y de software para cada uno y finalmente se explica la instalación de los softwares más relevantes para el proceso de generación del flujo de transporte (*TS - Transport Stream*).

En el Capítulo 3 se realiza un análisis de lo que contiene el flujo de datos para el transporte de audio, video y datos. Posteriormente se procede a explicar de forma prácticael proceso de generación del TS, mediante un software dedicado que es OpenCaster, donde se explica con ejemplos los diferentes tipos de servicio que puede tener un TS.

El en Capitulo 4 se explica la configuración de parámetros del transmisor para poder emitir la señal tipo *broadcast* al aire y tener una recepción óptima. En la parte de recepción se realizan pruebas para verificar los servicios y la usabilidad de las aplicaciones interactivas.

En el Capítulo 5 se encuentran las conclusiones y recomendaciones finales.

# CAPITULO 3

## GENERACIÓN DEL TRANSPORT STREAM

### 2.1 TRANSPORT STREAM (TS)

Es TS es el protocolo de comunicación para audio, video y datos definido para los sistemas que trabajan con el estándar MPEG-2 ISO/IEC 13818-1 [11], que genera un flujo de datos que se adaptan para comunicar o almacenar uno o más programas de televisión junto con datos adicionales, la función principal del TS es permitir la multiplexación síncrona de video, audio y datos.

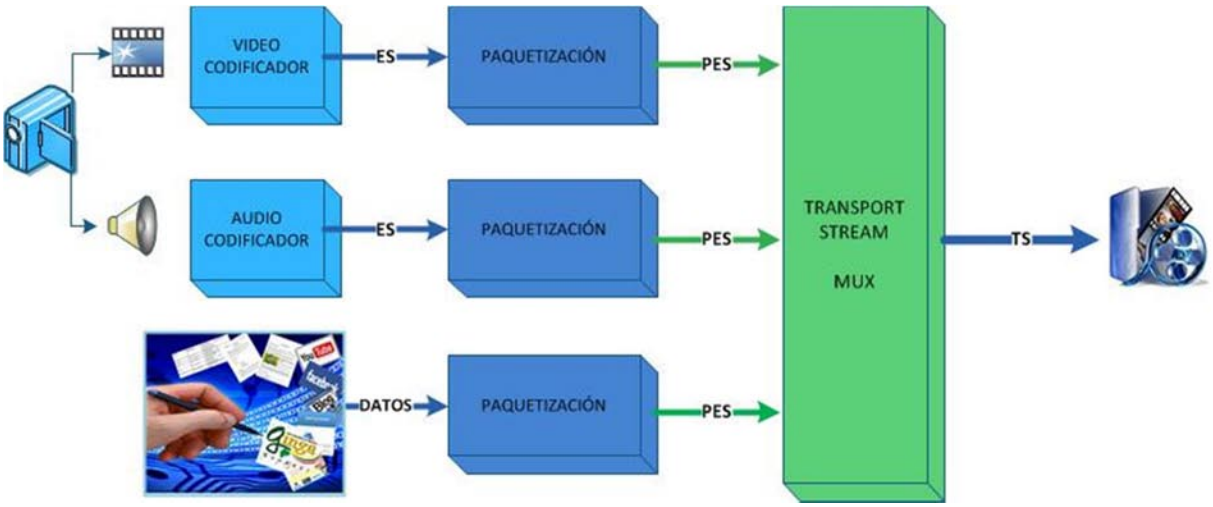


**Figura. 3.1. Estructura *Transport Stream***

En la Figura 3.1 se presenta el diagrama de bloques donde se puede observar el proceso del flujo de datos para el obtener el TS.

En el sistema ISDB-T la compresión de video y audio se hace con MPEG-2, en la versión brasileña que es el ISDB-Tb se usa MPEG-4 para la codificación del video y HE-AAC para el audio, pero para la parte de modulación y transporte se mantiene la paquetización en MPEG-2.

## 2.2 ESTRUCTURA DEL TRANSPORT STREAM

La Figura. 3.1 se muestra las etapas de codificación, paquetización y multiplexación del proceso de generación de un TS, para un servicio completo que contiene audio, video y datos.

En la Figura 3.2 se observa la formación de los flujos de datos ES (*Elementary Stream*), PES (*Packetized Elementary Stream*) y TS, los que serán explicados más adelante, y la forma en cómo están constituidos cada uno a partir del otro. Este proceso es el mismo para el video como para el audio pero de manera independiente.

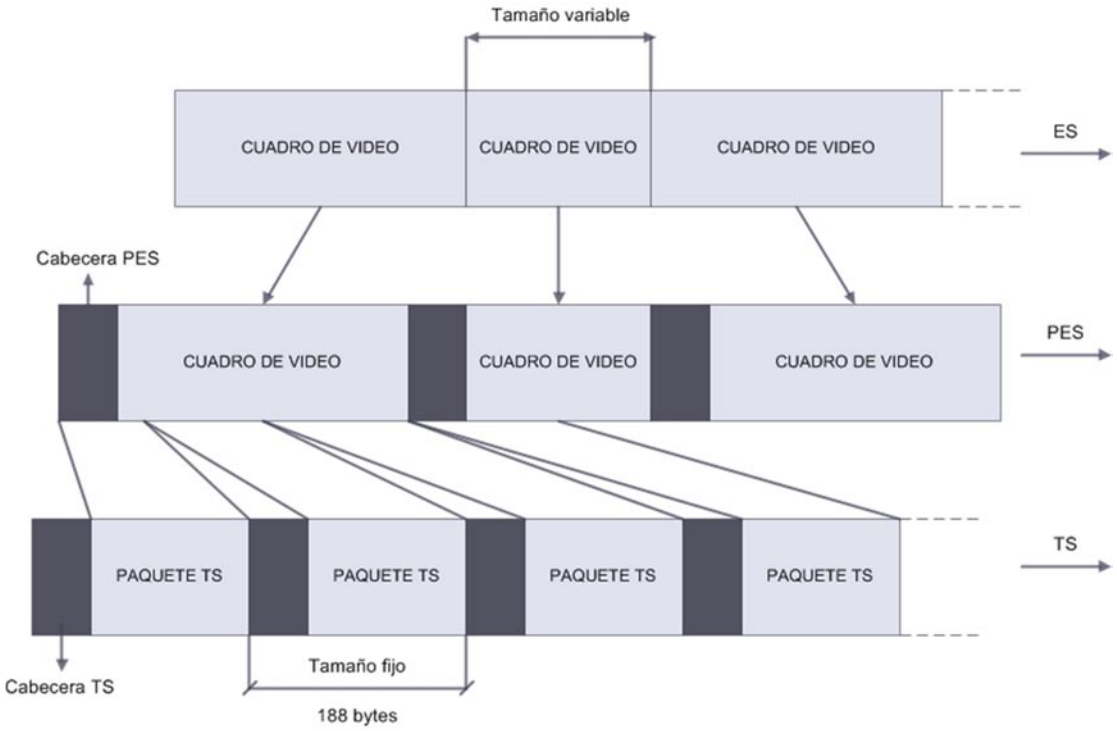


**Figura. 3.2. Formación de ES, PES y TS**

El TS es un flujo de la capa de transporte y está formado por paquetes de tamaño fijo de 188 Bytes, de los cuales 4 son de cabecera y 184 de carga; los cuales transportan diferentes tipos de información, como partes de PES (cuya compresión de video y audio se lo hace con MPEG-4 y ACC respectivamente), información sobre servicios, datos e información de rellenado (paquetes nulos) utilizados para obtener una tasa de transmisión fija.

### 2.2.1 Codificación

En la etapa de codificación el audio y video se comprimen de manera independiente formando cada uno de ellos un flujo de datos que es el *Elementary Stream* (ES), el cual está organizado en paquetes de tamaño variable que dependen de la calidad del video y audio.

### 2.2.2 Paquetización

En esta etapa se obtienen los bits provenientes del codificador de audio o video para estructurarlos en forma de paquetes llamados PES. Estos paquetes tienen una cabecera inicial seguidos por la carga (audio, video o datos). Esta cabecera transporta información de datos presentes en la carga, tamaño del paquete PES y el tiempo para informar al decodificador cuando decodificar, o cuando presentar la información del video y audio.

La compresión del paquete PES está relacionada con el tiempo del cuadro, y como la tasa de compresión de la señal de video y audio varia instantáneamente, esto hace que el tamaño del paquete PES sea variable.

### 2.2.3 Formación de Paquetes TS

En esta etapa el paquete PES es dividido en paquetes menores con un tamaño fijo de 188 *bytes* dando así origen al flujo de datos TS.

Con paquetes de tamaño fijo es más fácil la detección de errores, y la multiplexación de varios programas en un mismo flujo de datos. Por lo tanto el flujo de datos del TS transporta las señales de audio y video, multiplexados y comprimidos junto con información adicional sobre el flujo de datos.

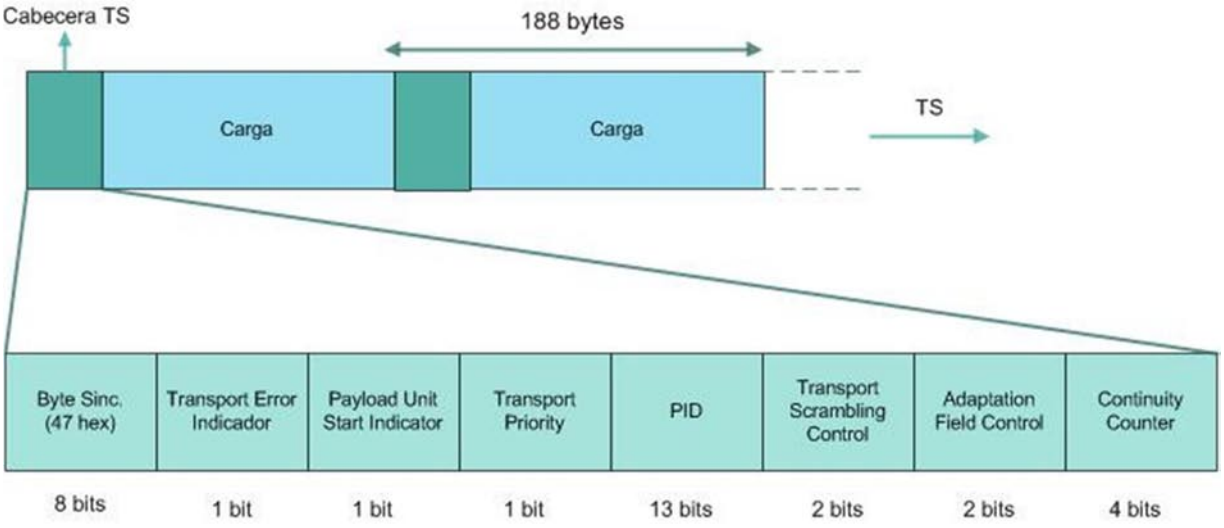


**Figura. 3.3. Cabecera paquete TS**

A continuación se explicara brevemente cada parte que compone la cabecera de un paquete TS como se muestra en la Figura. 3.3.

- *Byte de sincronismo:* Es utilizado como referencia del inicio del paquete TS, tiene un valor fijo de 47H (0100 011 bits).

- *Transport error indicator*: Está formado por 1 *bit* el cual indica si el paquete tiene errores, ocasionados durante la transmisión. Tiene valor de 1 cuando al menos 1 *bit* esta errado, caso contrario tiene un valor de 0 [12].

- *Payload unit start indicator*: En los paquetes nulos tiene un valor de ‘0’, para los paquetes que transportan las tablas del sistema asume ‘1’, cuando los datos de formación de las tablas se inician en este paquete es ‘0’. Cuando el paquete transporta información de audio y video, tendrá un valor de ‘1’, si el paquete estuviera transportando el inicio de un paquete PES es ‘0’.

- *PID- Packet identifier*: Está formado por 13 *bits* que identifica cada paquete del TS. El valor del PID depende del tipo de información que está transportando en su carga.

- *Transport priority*: Cuando este *bit* tiene valor de ‘1’ indica que el paquete que está transportando tiene mayor prioridad que los demás paquetes con el mismo PID.

- *Transport scrambling control*: Indica si el contenido del paquete sufrió algún tipo de criptografía.

- *Adaptation field control*: Está formado por 2 *bits* que indican si entre la cabecera inicial y la carga del paquete existe un campo especial llamado *adaptation field*. Dentro de la información que es transportada se encuentra el *Program Clock Reference* (PCR), esta referencia de tiempo hace que se tenga una correcta y síncrona decodificación al momento de la presentación de las señales de audio y video.

- *Continuity counter*: Formado por 4 *bits* que transporta el valor de un contador que es incrementado por cada paquete enviado con el mismo PID.

### 2.2.4 Multiplexación

El proceso de multiplexación del TS es cuando se une informaciones contenidas en dos o más flujos diferentes. En la multiplexación de diferentes flujos se debe garantizar el orden de envío de los paquetes de cada TS, no sea alterado tal como se indica en el ejemplo de la Figura. 3.4. Otro punto importante cuando se multiplexan diferentes TS, las tablas del nuevo flujo formado deben indicar correctamente todos los programas y componentes resultantes del proceso de multiplexación.

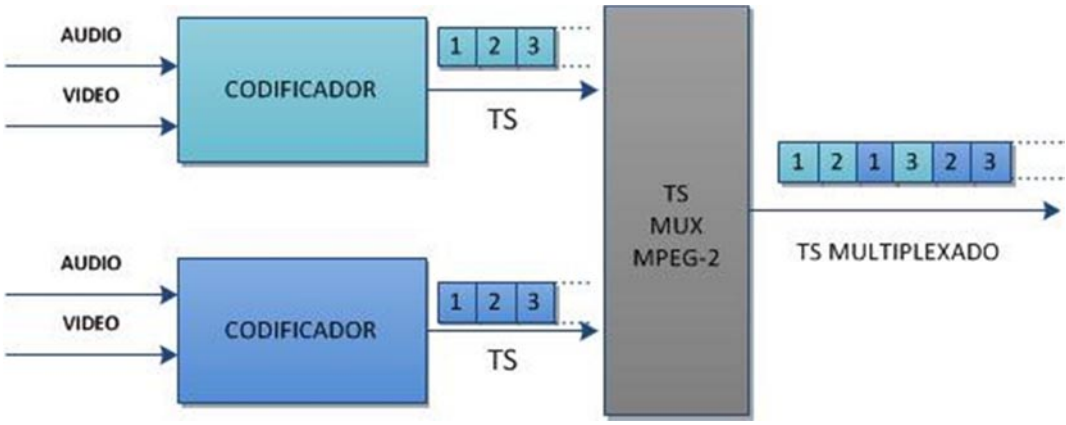


**Figura. 3.4. Multiplexación de dos TS**

### 2.2.5 Tablas de Información del Sistema

Cada paquete que forma el TS posee una identificación individual, PID. Por lo tanto no existe un valor de PID predeterminado, para el video o el audio. Existen algunos paquetes que transportan información adicional de cómo está estructurado el flujo, identificando cuantos programas están siendo transmitidos, a que programa pertenecen los videos y audios, lo que cada PID transporta entre otras informaciones.

Esas informaciones sobre el flujo son llamadas las tablas de información del sistema, poseen una estructura de formación bien definida y son transportadas en la carga de los paquetes. Cada tabla posee un valor de PID asociado, conforme se muestra en la Tabla 3.1 [12].

**Tabla. 3.1. Tabla de PID's utilizados para la transmisión de Tablas del Sistema**

| VALOR | DESCRIPCIÓN |
|---|---|
| 0x0000 | Paquetes que transportan a la tabla PAT *(Program Association Table*) |
| 0x0001 | Paquetes que transportan a la tabla CAT (*Conditional Access Table*) |
| 0x0003-0x000F | Valores de PID's Reservados |
| 0x0010 | Paquetes que transportan a la tabla NIT (*Network Information Table*) |
| 0x0011 | SDT (*Service Description Table*) y BAT (*Bouquet Information Table*) |
| 0x0012, 0x0026 y 0x0027 | EIT (*Event Information Table*) |
| 0x0013 | RST (*Running Status Table*) |
| 0x0014 | TDT (*Time and Date Table*) y TOT (*Time Offset Table*) |
| 0x0011 – 0x1FFE | Otras tablas, excluyéndose PAT, CAT, NIT y paquetes nulos |
| 0x0022 | PCAT (*Partial Content Announcement Table*) |
| 0x0024 | BIT (*Broadcaster Information Table*) |
| 0x0025 | NBIT (*Network Board Information Table*) e LDT (*Linked Description Table*) |
| 0x1FFF | Identificación de los paquetes nulos |

### 2.2.6 Tablas PSI/SI

Las talas PSI/SI (*Program Specific Information* / *Service Information*) son un conjunto de tablas que permiten al receptor identificar y decodificar correctamente los programas que forman un TS. Por ejemplo la tabla PAT, asocia cada flujo de programa a un PID de los paquetes que transportan a su PMT. La PMT a su vez es una tabla que enlista los PID's de los paquetes de audio, video, datos, PCR, entre otras, que están asociadas a un determinado programa. En la figura 3.2 se muestra un ejemplo de la estructura de un TS basado en las informaciones recogidas de las

dos tablas, en este ejemplo, la PAT informa que existen dos programas en el flujo y asocia a cada una de ellas a un valor de PID de la tabla PMT, y as u vez la PMT informa de los PID's de los paquetes asociados a el programa en cuestión.

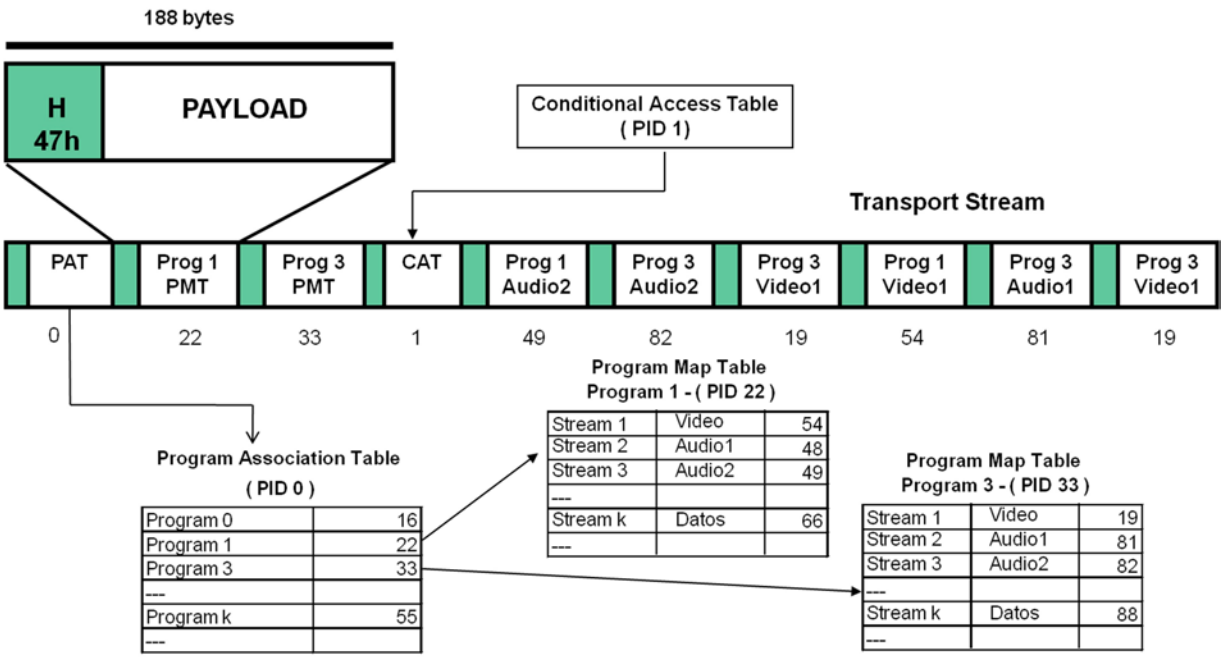


**Figura. 3.5. Estructura de un TS definido por las Tablas PAT y PMT**

Las tablas son transmitidas con una frecuencia relativamente alta para que sea posible que un receptor encendido en un instante cualquiera pueda rápidamente identificar y decodificar los datos recibidos. Por ejemplo las tablas PAT y PMT, deben ser enviadas en promedio de cada 100ms[12].

Las tablas pueden sufrir alteraciones a cualquier instante y tales alteraciones son informadas a través de un control de versión (*Version Numbrer*) de la tabla e indica que versión de la tabla debe ser aplicada (*Current Next Indicador*) en el momento.

Las tablas son formadas por secciones que tienen tamaño variable y que son transportadas dentro de la carga del paquete del TS. Para que el receptor identifique correctamente el tamaño de la tabla, es informado en cuantas secciones de la tabla

fue dividida (*Last Section Number*), además de la identificación de sección transportada en cada paquete (*Section Number*).

Para poder identificar errores en el receptor al finalizar la tabla se incrementa 4 bytes de CRC (*Cyclic Redundancy Check*).

En las siguientes tablas se muestra las funciones de las principales tablas PSI y SI.

**Tabla. 3.2. Tablas PSI principales utilizadas en el sistema ISDB-Tb.**

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

**Tabla. 3.3. Tablas SI principales utilizadas en el sistema ISDB-Tb.**

| Nombre | Significado/ Función |
|---|---|
| AIT | Tabla de información de aplicación.<br>Transmite información y control sobre las aplicaciones transmitidas en el flujo. |
| BIT | Tabla de información de radiodifusor.<br>Designa las unidades radiodifusoras e los parámetros SI para cada una de ellas. |
| EIT | Tablas de información de eventos.<br>Transporta información sobre los eventos existentes en cada servicio y su orden cronológica de exhibición. |

| | |
|---|---|
| NIT | Tabla de información de red.<br>Transporta información sobre la formación del TS y sobre su medio de transmisión. |
| RST | Tabla de estado de evento.<br>Permite la actualización del estado de eventos, por ejemplo cuando hay alteraciones en el horario de programación. |
| SDT | Tabla de descripción de servicios.<br>Lista e identifica los servicios existentes en un TS. |
| TDT | Tabla de fecha y horario.<br>Informa la hora y la fecha al sistema. |
| TOT | Tabla de cambio de fecha y hora.<br>Informa al receptor la hora, fecha y huso horario. |

## 2.3 GENERACION DEL TRANSPORT STREAM

Para la generación del *Transport Stream* se realiza con las herramientas de OpenCaster 2.4, cuyo procedimiento de instalación se explica en el capítulo anterior.

### 2.3.1 Formatos de Audio y Video

Los formatos de audio y video son los modos en el que se guarda su información con el fin de que luego puedan ser interpretados por el ordenador. Normalmente, un video es una colección de imágenes acompañadas por sonidos, que pueden ser grabadas en pistas diferentes que luego se sincronizan para su ejecución.

Para transformar la información analógica de las imágenes en digital se usan los códec (acrónimo de codificador/decodificador). En muchos casos estas utilidades analizan los fotogramas y emplean algoritmos para comprimir sus datos. La compresión puede ser temporal, en la que se analiza un fotograma y se guarda la

diferencia entre éste y el que le precede, o espacial, en la que se eliminan los datos de los píxeles que no cambian en cada fotograma.

En los formatos de audio tenemos dos tipos, los formatos con pérdida de información como por ejemplo: AAC, MP2, MP3, WMA; y los formatos sin pérdida como AIFF, FLAC, WAV, MIDI.

En los formatos de video existen tres de gran implementación: El AVI, el MOV y el correspondiente al estándar MPEG. El formato QuickTime Movie (MOV), creado por Apple, es multiplataforma y ahora permite interactuar con películas en 3D. El AVI (*Audio Video Interleaved*), permite audio y video intercalado también es un formato multiplataforma. Tanto *.avi como *.mov son contenedores de audio y vídeo con lo que son formatos de archivo, a este archivo habría que especificarle el tipo de video o audio que está conteniendo y que puede ser sin compresión o con la compresión soportada por dicho fichero como pueden ser para los *.avi el divx, Dv-pal, H264, etc. El formato correspondiente al estándar MPEG (Moving Pictures Experts Group) produce una compresión de los datos con una pequeña pérdida de la calidad; desde su creación, se ha definido el MPEG-1, utilizado en CD-ROM y Vídeo CD, el MPEG-2, usado en los DVD de Vídeo y la televisión digital, y el MPEG-4, que se emplea en la compresión de video en el estándar ISDB-Tb y para transmitir vídeo e imágenes en ancho de banda reducido; es un formato adecuado para distribuir multimedia en la Web. El formato MPEG4 es la base de actuales formatos como el divx xvid o el H264 siendo este último (H264) un códec tan potente que soporta vídeos de gran formato y calidad excelente con anchos de banda muy reducidos. [13]

### 2.3.2 Resolución y Relación de Aspecto de Pantalla

La resolución de video o de pantalla, es el número de píxeles que pueden ser mostrados en la pantalla. Viene dada por el producto del ancho por el alto, medidos ambos en pixeles, con la que se obtiene la llamada relación de aspecto.

**Tabla. 3.4 Formatos de video en ISDB-Tb**

| Nombre del Formato | Resolución (pixeles) | Relación de Aspecto |
|---|---|---|
| LDTV (*Low Definition Television*) Televisión en baja definición. | 320x240 | 4:3 |
| SDTV (*Standard Definition Television*) Televisión en definición estándar. | 640x480 | 4:3 |
| EDTV (*Enhanced Definition Television*) Televisión en definición mejorada. | 1280x720 | 16:9 |
| HDTV (*High Definition Television*) Televisión en alta definición. | 1920x1080 | 16:9 |

### 2.3.3 Codificador

En esta etapa partimos de una fuente común de audio y video, pero la parte de codificación la manejamos de forma independiente para lo que es video y audio.

Para poder realizar la parte de codificación utilizando las herramientas de OpenCaster, se debe partir de un archivo .AVI, pero como no todas las fuentes de video se encuentran en este formato, vamos a presentar una herramienta de Linux que nos permite transformar cualquier fuente de video a un archivo .avi.

- ***ffmpeg*:**

Es una biblioteca de codecs, capaz de codificar/decodificar en varios formatos de audio y vídeo, está desarrollada en C por GNU/Linux, puede ser compilado en la mayoría de sistemas operativos, por medio de línea de comandos, como se muestra a continuación.

ffmpeg –i fuente_video.mp4 –b 8000kb –r 25 –s 1920x1080 –aspect 16:9 –f video_final.avi

- Fuente de video (-i): fuente_video.mp4
- Bit rate de video (-b): 8000Kbps
- Frames por Segundo (-r): 25 fps
- Resolución de Video (-s): 1920x1080
- Aspecto de Pantalla (-aspect): 16:9
- Salida de video (-f): video_final.avi

También existen otras herramientas y software para la conversión de formatos de videos, un ejemplo es la herramienta *mencoder* que funciona sobre la plataforma Linux.

Es importante obtener la información con la que fue codificado el video fuente, para mantener los parámetros base como tasas de transmisión, aspecto, y resolución en la generación del Transport Stream. Para ver las propiedades de audio y video del archivo existen dos formas:

Dar clic derecho sobre el archivo, y escoger la opción propiedades, ahí se puede ver todos los parámetros como se indica en la Figura 3.5.

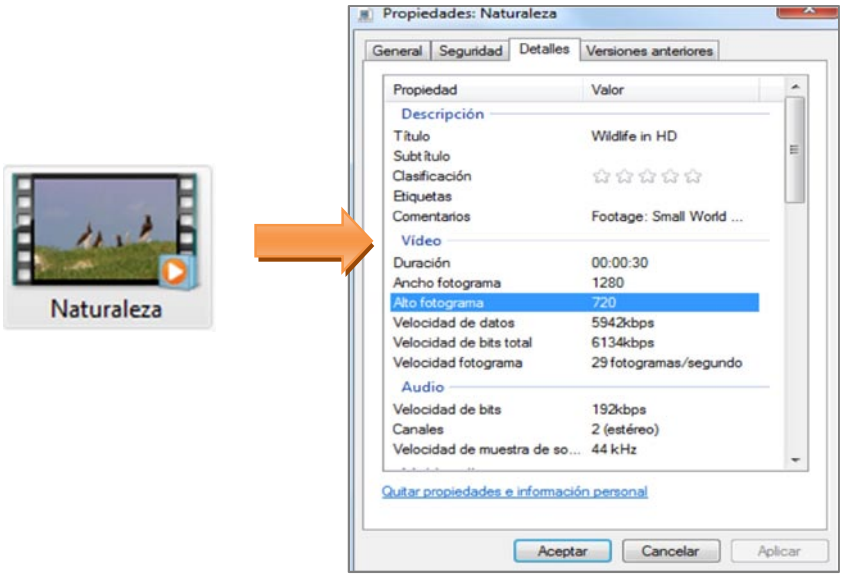


**Figura. 3.6. Propiedades de Video**

Otra manera para ver las propiedades del video fuente es utilizar una herramienta en Linux llamada idvid, para su instalación se escribe en siguiente comando en un terminal.

*$ sudo apt-get install tovid.*

Es necesario ubicarse en la carpeta donde se encuentra el archivo, y se escribe el siguiente comando para su análisis:

*$ idvid nombre_video.avi*

En la siguiente figura se muestra el resultado del análisis de un video, donde se puede observar la resolución, velocidad de transmisión, aspecto, etc.

```
root@diego-HP-Pavilion-dv4-Notebook-PC:~/Escritorio/Videos# idvid don.mp4
Removing unneeded temporary files in / ...
Just kidding! Had this been for real, your hard drive would be
well on its way to oblivion right now, because you're running as root.
I hope you know what you're doing...
------------------------------
idvid
Video identification script
Part of the tovid suite, version 0.31
http://www.tovid.org
------------------------------
Analyzing file: 'don.mp4'. This may take several minutes...
=========================================================
              File: don.mp4
             Width: 1920 pixels
            Height: 1080 pixels
      Aspect ratio: 1.77:1
            Frames: 7009
          Duration: 00:04:52 hours/mins/secs
         Framerate: 24.000 frames per second
      Video format: H264
     Video bitrate: 5949488 bits per second
-------------------------
Audio track 1 (Stream 0.1, AID 0):
-------------------------
             Codec: aac
           Bitrate: 0000 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.
=========================================================
```

**Figura. 3.7 Información del video fuente**

La Figura 3.8 muestra un diagrama del proceso de codificación para transformar el formato del video fuente a formato AVI.

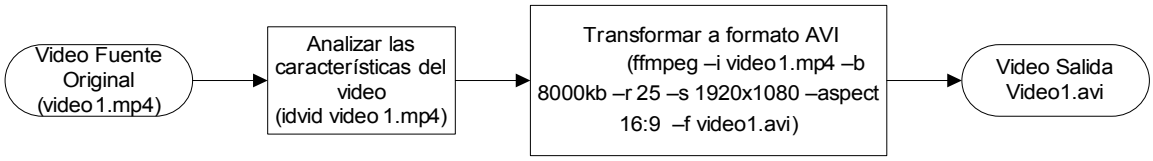


**Figura. 3.8 Transformación a formato .AVI**

### 2.3.4 Transport Stream con un Servicio

El proceso para la generación de un *Transport Stream* con un servicio de televisión digital que contiene interactividad (Aplicación GINGA NCL) se muestra en la Figura. 3.9.

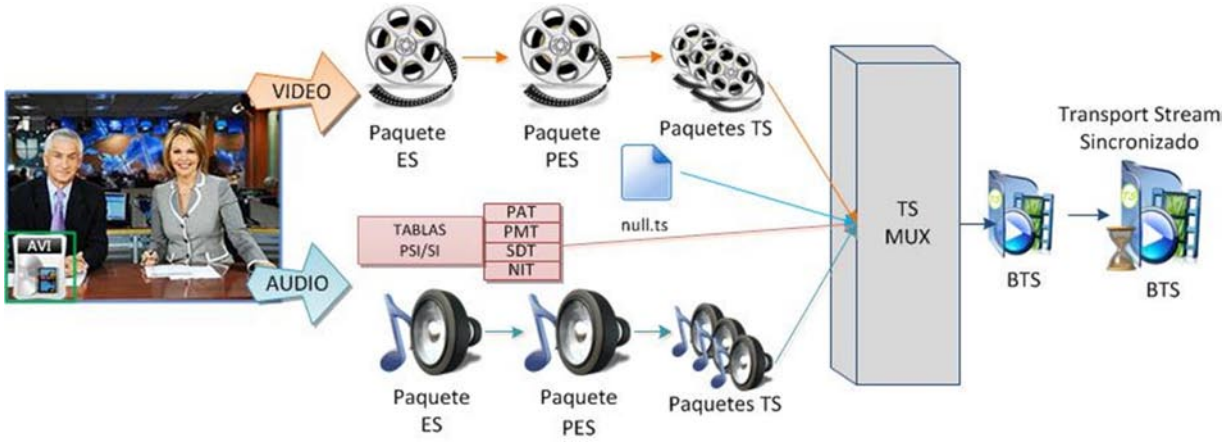


Figura. 3.9 Proceso de generación de Transport Stream con un servicio

En la Figura 3.9, se puede observar que el audio y el video son codificados individualmente, como indica en el estándar MPEG-2.

Para este procedimiento es necesario partir del video con formato AVI, ya que la versión de OpenCaster instalada no soporta otro formato de codificación, la Figura 3.10, muestra en diagrama como se realiza la codificación del audio y video por separado con ayuda de la herramienta *ffmpeg*.

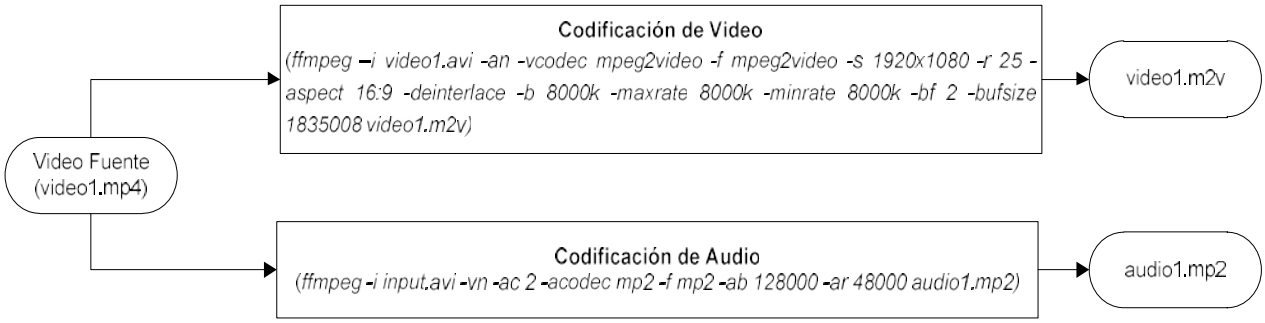


**Figura. 3.10 Codificación de Audio y Video por separado**

- **Codificación del Video:**

*ffmpeg –i video1.avi -an -vcodec mpeg2video -f mpeg2video -s 1920x1080 -r 25 -aspect 16:9 -deinterlace -b 8000k -maxrate 8000k -minrate 8000k -bf 2 -bufsize 1835008 video1.m2v*

**Tabla. 3.5 Parámetros codificación video**

| Parámetro | Descripción |
|---|---|
| -i | Archivo fuente de audio y video en formato .avi |
| -an | No audio, ignora a la señal de audio si es que está presente en la entrada. |
| -vcodec | Se trata del códec de video, en este caso se trata de mpeg2video que es el códec que necesitamos para que sea compatible con nuestra señal de Transport Stream. |
| -f | Formato de salida, en este caso el formato que necesitamos es mpeg2video. |
| -s | Resolución del video en pixeles, ejemplo 1920x1080. |
| -r | Es el número de frames por segundo. El que utilizaremos para nuestro estándar es de 25 fps. |
| -aspect | Nos indica el aspecto de la presentación en la pantalla. |
| -deinterlace | Define una opción para desentrelazar imágenes. |
| -b | Es la tasa de bits de la señal codificada en Kbps. Y para mantenerla |

| | |
|---|---|
| | constante ponemos este mismo parámetro en maxrate y minrate, que son las tasas máximas y mínimas respectivamente. |
| -bf | Numero de cuadros del tipo b para cada GOP |
| -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. |
| video1.m2v | Nombre del archivo generado |

- **Codificación del Audio:**

*ffmpeg -i input.avi -vn -ac 2 -acodec mp2 -f mp2 -ab 128000 -ar 48000 audio1.mp2*

**Tabla. 3.6 Parámetros codificación audio**

| Parámetro | Descripción |
|---|---|
| -i | Archivo fuente de audio y video en formato .avi |
| -vn | No video, ignora a la señal de video si es que está presente en la entrada. |
| -ac | Define el número de canales. Por defecto es igual a 1 |
| -acodec | Se trata del códec de audio, en este caso se trata de mp2 que es el códec que necesitamos para que sea compatible con nuestra señal de Transport Stream. |
| -f | Formato de salida (mp2). |
| -ab | Es la tasa de bits de la señal de audio en bits por segundo. En este ejemplo la tasa de bits es de 128kbps. |
| -ar | Define la frecuencia de muestreo de la señal de audio. La frecuencia por defecto es de 44100 Hz. |
| audio1.mp2 | Nombre del archivo generado |

- **Paquetización del Video:**

Para el proceso de paquetización del video, se parte del archivo ES (*Elementary Stream*) que en este caso es el *video1.m2v* el cual se obtiene después del proceso de codificación.

La Figura. 3.11, muestra el proceso de paquetización del video en dos etapas, en la primera etapa se puede ver la conversión de un paquete ES a paquetes PES, y en la segunda etapa se ve la conversión de los paquetes PES a paquetes TS de tamaño fijo (188 bytes).

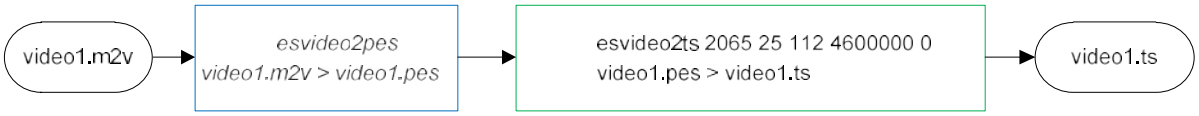


**Figura. 3.11Paquetización del Video**

***esvideo2pes***: es el comando de Opencaster que permite pasar de un ES de video a PES de video, la Tabla 3.7, muestra sus atributos.

*$ esvideo2pes video1.m2v > video1.pes*

**Tabla. 3.7 Parámetros esvideo2pes**

| Parámetro | Descripción |
|---|---|
| video1.m2v | Video ES a ser convertido |
| video1.pes | Video convertido a PES |

***pesvideo2ts***: es el comando de Opencaster que permite pasar de un archivo PES de video a TS de video, la Tabla 3.8 muestra sus atributos.

*$ pesvideo2ts 2065 25 112 9200000 0 video1.pes > video1.ts*

**Tabla. 3.8 Parámetros pesvideo2pes**

| Valor del parámetro | Descripción |
|---|---|
| 2065 | Numero de PID asignado al video |
| 25 | Numero de Frames por Segundo |
| 112 | Es el valor del VBV (*Video Buffer Verifier*), para nuestro caso este valor será fijo ya que así está definido en el estándar MPEG-2. |
| 9200000 | Es el ancho de banda de la señal de Transport Stream. Es recomendado que el ancho de banda sea un 15% más grande que la tasa de bits de la señal de video definida en la función *ffmpeg.*<br>En este caso 8000k+(15% de 8000K)=9200K |
| 0 | Indica que el video no va a estar en un *loop*. |
| video1.pes | Video PES a ser convertido. |
| Video1.ts | Video de salida ya en formato TS |

- **Paquetización del Audio:**

Para el proceso de paquetización del audio, también se parte de un archivo ES (*Elementary Stream*) el cual es resultado del procedimiento de codificación del audio, que en este caso es *audio1.mp2.*

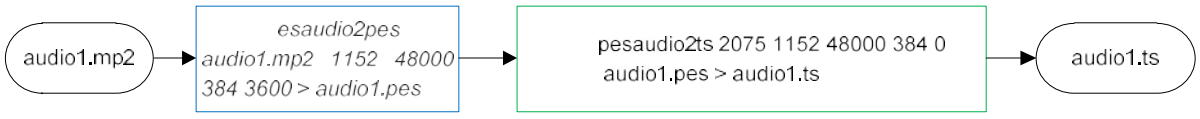


**Figura. 3.12Paquetización del Audio**

La Figura. 3.12, muestra el proceso de paquetización del audio en dos etapas, procedimiento muy similar a la paquetización del video pero con diferentes parámetros. La primera etapa es la conversión de un paquete ES a paquetes PES,

y en la segunda etapa es la conversión de los paquetes PES a paquetes TS de tamaño fijo (188 bytes).

***esaudio2pes***: es el comando de Opencaster que permite pasar de un ES de video a PES de video, la Tabla 3.7, muestra sus atributos.

*$ esaudio2pes audio1.mp2 1152 48000 384 3600 > audio1.pes*

**Tabla. 3.9 Parámetros esaudio2pes**

| Valor del parámetro | Descripción |
|---|---|
| audio1.mp2 | Audio ES a ser convertido |
| audio1.pes | Audio convertido a PES |
| 1152 | Numero de muestras por frame |
| 48000 | Frecuencia de muestreo, debe ser la misma que se definió en el ES. |
| 384 | Tamaño del frame de audio. |
| 3600 | Desplazamiento del PTS (*Presentation Time Stamp*), se define para poder sincronizar con el video. |

***pesaudio2ts***: es el comando de OpenCaster que permite pasar de un archivo PES de audio a TS de audio, la Tabla 3.10, muestra sus atributos.

$ pesaudio2ts 2075 1152 48000 384 0 audio1.pes > audio1.ts

**Tabla. 3.10 Parámetros esaudio2pes**

| Valor del parámetro | Descripción |
|---|---|
| 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. |
| 384 | 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*. |
| audio1.pes | Audio PES a ser convertido. |
| audio1.ts | Audio de salida ya en formato TS |

- **Generación de Tablas**

Para la generación de tablas PSI/SI, se lo hace mediante un script en python, al compilar este script se obtiene las secciones de la tabla en paquetes, las mismas que son necesarias para la multiplexación ya que contienen información básica para que el Transport Stream pueda ser transmitido.

**Tabla. 3.11. Script de la tabla para un servicio de televisión digital**

```
#! /usr/bin/env python
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 *

tvd_ts_id                = 0x073b #ID de red.
tvd_orig_network_id= 0x073b #ID de red original.
ts_freq                  = 671 #Frecuencia de transmisión
ts_remote_control_key    = 0x05 #Tecla de control remoto.
tvd_service_id_sd        = 0xe760 #ID de servicio de TV Digital.
```

```
tvd_pmt_pid_sd              = 1031 #PID de la PMT del servicio.

#Definición Red NIT
nit = network_information_section(
        network_id = tvd_orig_network_id,
        network_descriptor_loop = [
                network_descriptor(network_name = "ESPETV",),
                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 = tvd_ts_id,
                original_network_id = tvd_orig_network_id,
                transport_descriptor_loop = [
                        service_list_descriptor(
                                dvb_service_descriptor_loop = [
                                        service_descriptor_loop_item (
                                                service_ID = tvd_service_id_sd,
                                                service_type = 1,
                                        ),
                                ],
                        ),
                        terrestrial_delivery_system_descriptor(
                                area_code = 1341,
                                guard_interval = 0x01,
                                transmission_mode = 0x02,
```

```
                                frequencies = [
                                        tds_frequency_item( freq=ts_freq )
                                ],
                        ),
                        partial_reception_descriptor (
                                service_ids = []
                        ),
                        transport_stream_information_descriptor (
                                remote_control_key_id = ts_remote_control_key,
                                ts_name = "ESPETV",
                                transmission_type_loop = [
                                        transmission_type_loop_item(
                                                transmission_type_info = 0x0F,
                                                service_id_loop = [
                                                        service_id_loop_item(
                                                                service_id=tvd_service_id_sd
                                                        ),
        ]
                                ),
                                transmission_type_loop_item(
                                        transmission_type_info = 0xAF,
                                        service_id_loop = [],
                                ),
                        ],
                )
        ],
        ),
],
version_number = 0,
section_number = 0,
last_section_number = 0,
```

```
)

#Definición servicios SDT
sdt = service_description_section(
        transport_stream_id = tvd_ts_id,
        original_network_id = tvd_orig_network_id,
        service_loop = [
                service_loop_item(
                        service_ID = tvd_service_id_sd,
                        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 = "ESPE",
                                        service_name = "ESPETV",
                                ),
                        ],
                ),
        ],
        version_number = 0,
        section_number = 0,
        last_section_number = 0,
)

#Definición del mapa de programas PAT
pat = program_association_section(
        transport_stream_id = tvd_ts_id,
        program_loop = [
```

```
                program_loop_item(
                        # Programa especial para la tabla NIT
                        program_number = 0,
                        PID = 16,
                ),
                program_loop_item(
                        program_number = tvd_service_id_sd,
                        PID = tvd_pmt_pid_sd,
                ),
        ],
        version_number = 0,
        section_number = 0,
        last_section_number = 0,
)

#Tabla PMT
pmt_sd = program_map_section(
        program_number = tvd_service_id_sd,
        PCR_PID = 2065,
        program_info_descriptor_loop = [],
        stream_loop = [
                stream_loop_item(
                        stream_type = 2, # mpeg2 video stream type
                        elementary_PID = 2065,
                        element_info_descriptor_loop = [
                        ]
                ),
                stream_loop_item(
                        stream_type = 3, # mpeg2 audio stream type
                        elementary_PID = 2075,
                        element_info_descriptor_loop = []
```

```
                ),

        ],
        version_number = 0,
        section_number = 0,
        last_section_number = 0,
)

#Secciones y paquetes que le transportan
out = open("./nit.sec", "wb")
out.write(nit.pack())
out.close
out = open("./nit.sec", "wb") # python flush bug
out.close
os.system("sec2ts 16 < ./nit.sec > ./nit.ts")

out = open("./pat.sec", "wb")
out.write(pat.pack())
out.close
out = open("./pat.sec", "wb") # python flush bug
out.close
os.system("sec2ts 0 < ./pat.sec > ./pat.ts")

out = open("./sdt.sec", "wb")
out.write(sdt.pack())
out.close
out = open("./sdt.sec", "wb") # python flush bug
out.close
os.system("sec2ts 17 < ./sdt.sec > ./sdt.ts")

out = open("./pmt_sd.sec", "wb")
```

```
out.write(pmt_sd.pack())
out.close
out = open("./pmt_sd.sec", "wb") # python flush bug
out.close
os.system("sec2ts " + str(tvd_pmt_pid_sd) + "< ./pmt_sd.sec > ./pmt_sd.ts")
```

El script que muestra la Tabla 3.10, debe ser guardado en un archivocon el nombre "*tabla_1servicio.py"*, dentro del mismo directorio donde se encuentran el resto de archivos generados anteriormente.

Para ejecutar este archivo "*tabla_1servicio.py"*, se debe abrir un terminal, ingresar al directorio donde se encuentran todos los archivos, y escribir el siguiente código para compilar y ejecutar respectivamente:

```
$ chmod u+x tabla_1servicio.py
$ ./tabla_1servicio.py
```

.

Con esto se generan los siguientes archivos que se muestran en la Tabla 3.11.

**Tabla. 3.12 Secciones y Paquetes TS generados por las tablas**

| Secciones | Paquetes TS |
|---|---|
| nit.sec | nit.ts |
| pat.sec | pat.ts |
| sdt.sec | sdt.ts |
| pmt_sd.sec | pmt_sd.ts |

- **Multiplexación**

Hasta ahora se ha generado los flujos de transporte de las señales de audio y video de manera independiente y las tablas PSI/SI; resta por generar un archivo

vacío el cual guardara los paquetes nulos para completar el ancho de banda que se transmite en los 6Mhz.

Para generar este archivo se debe crear un documento vacío con el nombre "*null.ts*", y guardarlo en el directorio donde se encuentran todos los archivos.

En la Figura. 3.13, se puede observar gráficamente los archivos necesarios para la multiplexación, el comando con el que se realiza la multiplexación y finalmente el archivo *videofinal1.ts* resultante de este proceso el cual es un solo flujo de transporte (BTS).

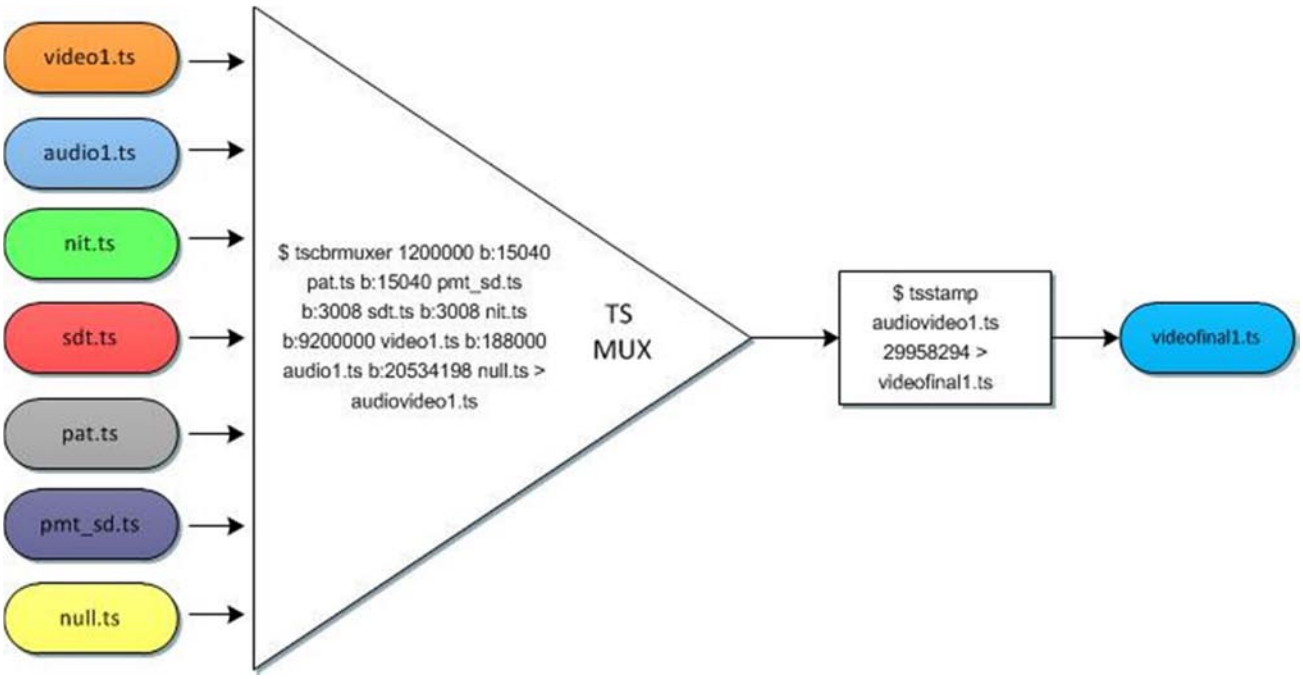


**Figura. 3.13Proceso de Multiplexación**

***tscbrmuxer***: Es el comando de OpenCaster para multiplexar, la Tabla 3.13 muestra sus atributos.

$ tscbrmuxer 1200000 b:15040 pat.ts b:15040 pmt_sd.ts b:3008 sdt.ts b:3008 nit.ts b:9200000video1.ts b:188000 audio1.ts b:20534198 null.ts > audiovideo1.ts

**Tabla. 3.13. Parámetros tscbrmuxer**

| Valor del Parámetro | Descripción |
|---|---|
| 1200000 | Es la cantidad de paquetes a multiplexar. El sistema ISDB-T transmite cerca de 20000 paquetes por segundo. Para este ejemplo la duración del video es de un minuto, con lo cual, si el archivo generado es de cerca de 30 segundos se va a necesitar 600000 paquetes. |
| b: 15040 | Tanto la PAT como la PMT, se quiere enviarla al menos 10 veces por segundo. Sabiendo que cada una de las tablas entra en un solo paquete de 188 bytes, tomando en cuenta que se tiene que enviar 10 paquetes por segundo y cada paquete es de 188 bytes $*$ 8 = 1504 bits. |
| b: 3008 | NIT y SDT, para llegar a este valor se sigue un análisis similar al de la PAT y la PMT, a diferencia que estas tablas son enviadas 2 veces por segundo. |
| b: 9200000 | Es el ancho de banda del video que ya se definió previamente en el proceso de paquetización. |
| b: 188000 | Es el ancho de banda del audio que ya se definió previamente en el proceso de paquetización. |
| b: 20534198 | Es el ancho de banda asignado a los paquetes nulos, el sistema ISDB-T tiene un ancho de banda fijo de 29.958.294 bps. Para determinar este valor se debe realizar el siguiente calculo: 29958294-15040*2-3008*2-9200000-188000=20534198 |
| audiovideo1.ts | Es el nombre del archivo de salida .ts |

- **Sincronización**

En el proceso anterior de multiplexación se cambió la posición de los paquetes de video, por lo tanto la llegada de los mismos será imprecisa en el STB. Para solucionar este problema se usa la herramienta *tsstamp*, para sincronizar el audio y el video.

$ tsstamp audiovideo1.ts 29958294 > videofinal1.ts

### 2.3.5 Transport Stream con un Servicio e Interactividad

Para la generación de un *Transport Stream* que contenga un servicio e interactividad, se debe seguir el proceso explicado anteriormente para la codificación, paquetización, y la generación de las secciones y paquetes TS a partir del *script*, pero ahora este script no es el mismo. La multiplexación ahora tiene nuevos parámetros, ya que se incluye un nuevo flujo de datos.

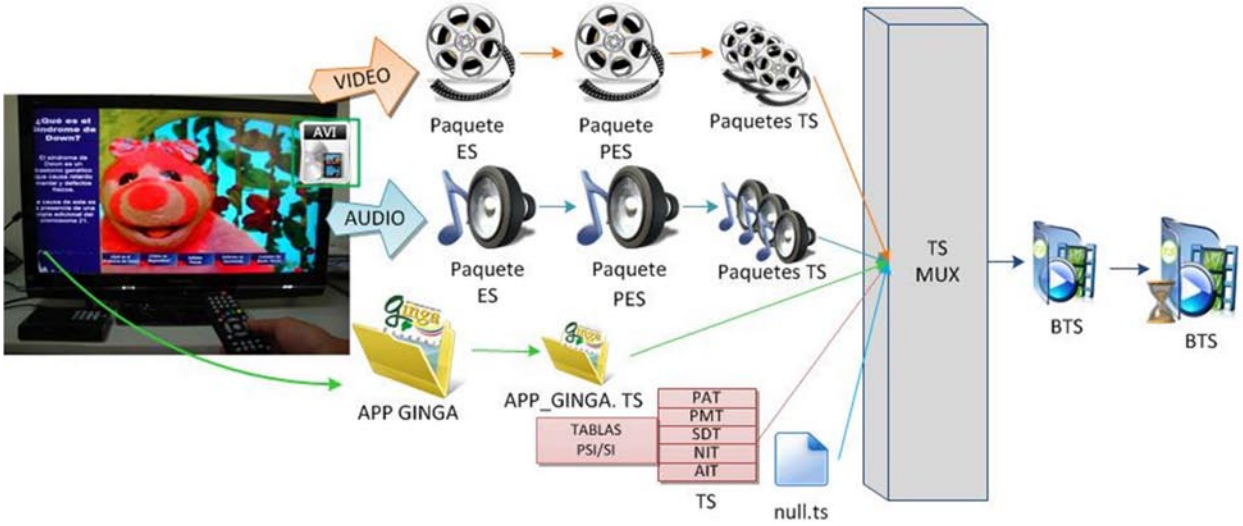


**Figura. 3.14. Generación Transport Stream con un Servicio e Interactividad**

La Figura. 3.14 muestra gráficamente como es el flujo de datos para obtener un *Transport Stream* con un servicio e interactividad. En las Tablas PSI/SI se aumenta la tabla AIT que es la encargada de asociar la interactividad a la tabla PMT del servicio del Transport Stream.

**Tabla. 3.14. Script de la tabla para un servicio de televisión digital con interactividad**

```
#! /usr/bin/env python
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 *

tvd_ts_id               = 0x073b #ID de red.
tvd_orig_network_id= 0x073b #ID de red original.
ts_freq                         = 671   #Frecuencia de transmisión
ts_remote_control_key   = 0x05  #Tecla de control remoto.
tvd_service_id_sd   = 0xe760 #ID de servicio de TV Digital.
tvd_pmt_pid_sd              = 1031 #PID de la PMT del servicio.

#Definición Red NIT

nit = network_information_section(
        network_id = tvd_orig_network_id,
        network_descriptor_loop = [
                network_descriptor(network_name = "ESPETV",),
                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 = tvd_ts_id,
                original_network_id = tvd_orig_network_id,
                transport_descriptor_loop = [
                        service_list_descriptor(
                                dvb_service_descriptor_loop = [
                                        service_descriptor_loop_item (
                                                service_ID = tvd_service_id_sd,
                                                service_type = 1,
                                        ),
                                ],
                        ),
                        terrestrial_delivery_system_descriptor(
                                area_code = 1341,
                                guard_interval = 0x01,
                                transmission_mode = 0x02,
                                frequencies = [
                                        tds_frequency_item( freq=ts_freq )
```

```
                                ],
                        ),
                        partial_reception_descriptor (
                                service_ids = []
                        ),
                        transport_stream_information_descriptor (
                                remote_control_key_id = ts_remote_control_key,
                                ts_name = "ESPETV",
                                transmission_type_loop = [
                                        transmission_type_loop_item(
                                                transmission_type_info = 0x0F,
                                                service_id_loop = [
                                                        service_id_loop_item(
                                                                service_id=tvd_service_id_sd
                                                        ),
                                        ]
                                ),
                                transmission_type_loop_item(
                                        transmission_type_info = 0xAF,
                                        service_id_loop = [],
                                ),
                        ],
                )
        ],
        ),
],
version_number = 0,
section_number = 0,
last_section_number = 0,
)

#Definiciónservicios SDT

sdt = service_description_section(
        transport_stream_id = tvd_ts_id,
        original_network_id = tvd_orig_network_id,
        service_loop = [
                service_loop_item(
                        service_ID = tvd_service_id_sd,
                        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 = "ESPE",
```

```
                                        service_name = "ESPETV",

                                    ),
                            ],
                    ),
            ],
            version_number = 0,
            section_number = 0,
            last_section_number = 0,
)

#Definición del mapa de programas PAT

pat = program_association_section(
            transport_stream_id = tvd_ts_id,
            program_loop = [
                    program_loop_item(
                            # Programa especial para la tabla NIT
                            program_number = 0,
                            PID = 16,
                    ),
                    program_loop_item(
                            program_number = tvd_service_id_sd,
                            PID = tvd_pmt_pid_sd,
                    ),
            ],
            version_number = 0,
            section_number = 0,
            last_section_number = 0,
)


#Tabla PMT

pmt_sd = program_map_section(
            program_number = tvd_service_id_sd,
            PCR_PID = 2065,
            program_info_descriptor_loop = [],
            stream_loop = [
                    stream_loop_item(
                            stream_type = 2, # mpeg2 video stream type
                            elementary_PID = 2065,
                            element_info_descriptor_loop = [
                            ]
                    ),
                    stream_loop_item(
                            stream_type = 3, # mpeg2 audio stream type
```

```
                elementary_PID = 2075,
                element_info_descriptor_loop = []
        ),

        stream_loop_item(
                stream_type = 5, # AIT stream type
                elementary_PID = 2001,
                element_info_descriptor_loop = [
                        data_component_descriptor (
                                data_component_id = 0xA3, # sistema AIT
                                additional_data_component_info =
ait_identifier_info(
                                        application_type =
GINGA_NCL_application_type,
                                        ait_version = 0
                                ).bytes(),
                        ),
                        application_signalling_descriptor(
                                application_type = 9, # 9 GINGA-NCL
                                AIT_version = 1, # current ait version
                        ),
                ]
        ),
        stream_loop_item(
                stream_type = 0x0B, # DSMCC stream type
                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 = "ESPED",
                        ),
                        stream_identifier_descriptor(
                                component_tag = 0x0C,
                        ),
                        carousel_identifier_descriptor(
                                carousel_ID = 2,
                                format_ID = 0,
                                private_data = "ESPED",
                        ),
                        data_component_descriptor (
                                data_component_id = 0xA0, # sistema GINGA
                                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,
)

# Programa especial para la tabla AIT

ait = application_information_section(
        application_type = 0x0009, # GINGA-NCL
        common_descriptor_loop = [],
                application_loop = [
                        application_loop_item(
                        organisation_id = 0x0000000A,
                        application_id = 0x64,
                        application_control_code = 0x01, # AUTOSTART
                        application_descriptors_loop = [
                        transport_protocol_descriptor(
                                protocol_id = 0x0001,
                                transport_protocol_label = 0,
                                remote_connection = 0,
                                component_tag = 0x0C, # association_tag
                        ),
                        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 = "/",
                                class_path_extension = "",
                                initial_class = "main.ncl", # nombre del archivo NCL
                                                        # a ser ejecutado.
                        ),
                        ]
                ),
        ],
        version_number = 0,
        section_number = 0,
        last_section_number = 0,
)

#Secciones y paquetes que le transportan

out = open("./nit.sec", "wb")
out.write(nit.pack())
out.close
out = open("./nit.sec", "wb") # python flush bug
out.close
os.system("sec2ts 16 < ./nit.sec > ./nit.ts")

out = open("./pat.sec", "wb")
out.write(pat.pack())
out.close
out = open("./pat.sec", "wb") # python flush bug
out.close
os.system("sec2ts 0 < ./pat.sec > ./pat.ts")

out = open("./sdt.sec", "wb")
out.write(sdt.pack())
out.close
out = open("./sdt.sec", "wb") # python flush bug
out.close
os.system("sec2ts 17 < ./sdt.sec > ./sdt.ts")

out = open("./pmt_sd.sec", "wb")
out.write(pmt_sd.pack())
out.close
out = open("./pmt_sd.sec", "wb") # python flush bug
out.close
os.system("sec2ts " + str(tvd_pmt_pid_sd) + " < ./pmt_sd.sec > ./pmt_sd.ts")

out = open("./ait.sec", "wb")
out.write(ait.pack())
```

```
out.close
out = open("./ait.sec", "wb") # python flush bug
out.close
os.system("sec2ts " + str(2001) + " < ./ait.sec > ./ait.ts")
```

- **Generando el carrusel de objetos**

Se crea una carpeta que contenga la aplicación ginga en el directorio de trabajo, la carpeta debe llamarse *"app_ginga".*

Cabe mencionar que el nombre del archivo principal de la aplicación ginga debe llamarse main.ncl, porque así está definido en la tabla AIT.

Después se usa la herramienta oc-update.sh que viene con OpenCaster para generar el carrusel de objetos.

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

Esto va a generar un archivo *app_ginga.ts* que contiene los paquetes que llevan las secciones del carrusel. La Tabla 3.15 describe los parámetros:

**Tabla. 3.15. Parámetros para la generación de interactividad**

| Parámetro | Descripción |
|---|---|
| app_ginga | El directorio donde está la aplicación. También define el nombre del archivo generado. |
| 0x0C | El association_tag del carrusel generado. |
| 1 | El número de versión de los módulos generados. |
| 2004 | El PID en el que se envía el carrusel. |
| 2 | El identificador del carrusel. |

- **Multiplexación**

En la Figura. 3.15. se muestra el diagrama de multiplexación con interactividadpara un Transport Stream que contiene un servicio e interactividad.

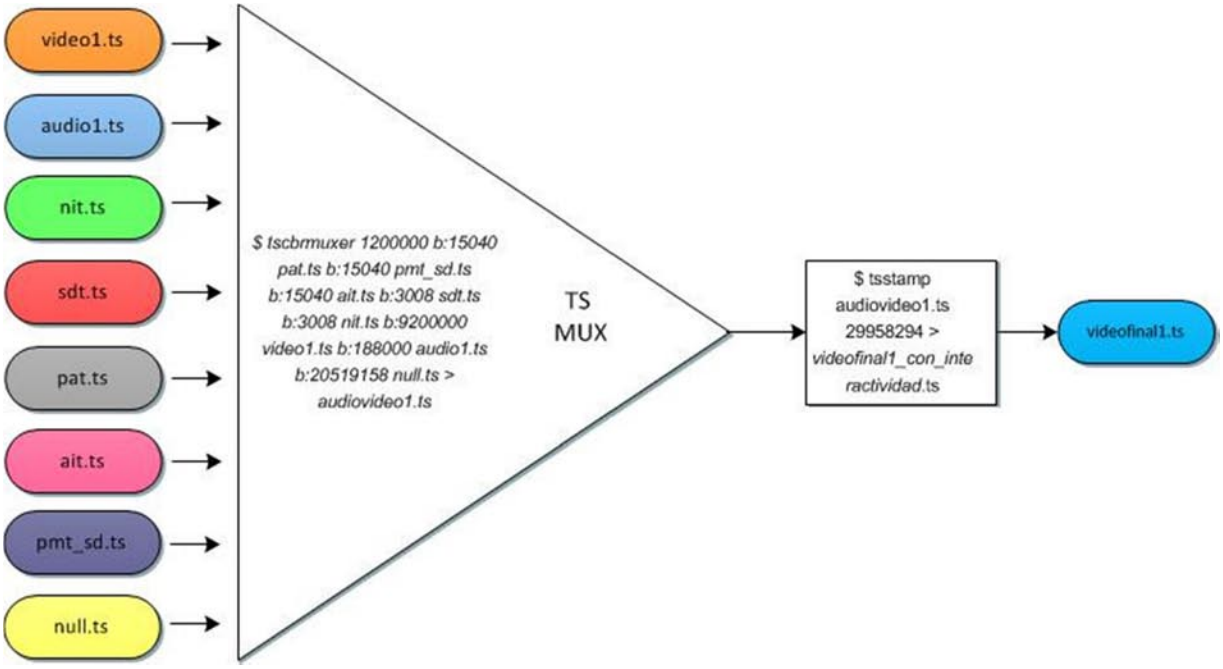


**Figura. 3.15Proceso de Multiplexación con Interactividad**

Para la multiplexación se aumenta en el Transport Stream la tablaAIT por lo que también se realiza un nuevo cálculo:

29958294-15040*3-3008*2-9200000-188000=20519158

*$ tscbrmuxer 1200000 b:15040 pat.ts b:15040 pmt_sd.ts b:15040 ait.ts b:3008 sdt.ts b:3008 nit.ts b:9200000 video1.ts b:188000 audio1.ts b:20519158 null.ts > audiovideo1.ts*

Finalmente también se realiza la sincronización, con la herramienta tsstamp.

*$ tsstamp audiovideo1.ts 29958294 > videofinal1_con_interactividad.ts*

# CAPÍTULO 4

## TRANSMISIÓN Y PRUEBAS DE RECEPCIÓN

### 3.1 TRANSMISIÓN EN EL ESTANDAR ISDB-Tb

El sistema ISDB-T de televisión digital en su capa física o de transmisión es bastante flexible, ya que puede operar en diferentes configuraciones, tiene corrección de errores, utiliza entrelazadores (*interleaving*) en el dominio del tiempo y de la frecuencia, utiliza la modulación COFDM (*Coded Orthogonal Frequency Division Multiplexing*), y posibilita la utilización de tres capas A, B, C, con diferentes segmentos para obtener una taza de transmisión robusta.

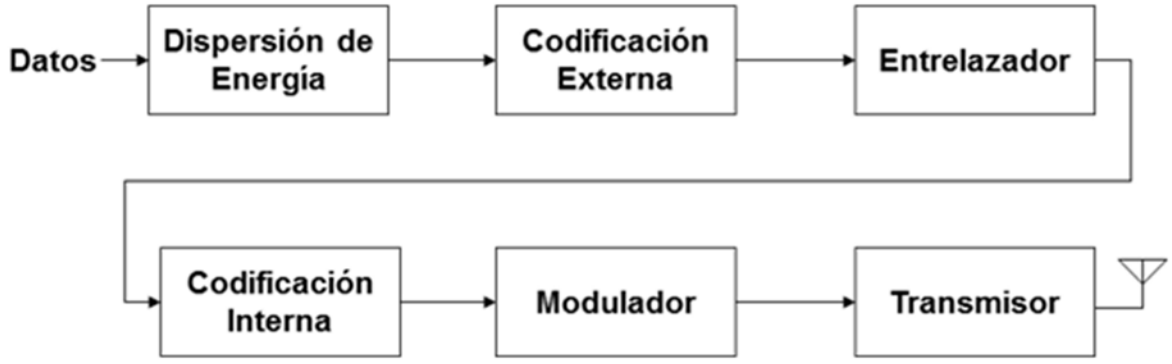


**Figura. 4.1. Capa física de los sistemas de TV Digital.**

En la figura 4.1 se puede ver el diagrama de bloques de la capa física de un transmisor de televisión digital de cualquier estándar, para entender cada bloque a continuación se describe de manera general, sus funciones.

- **Dispersión de Energía.-** Su objetivo es distribuir la energía de la señal de entrada a lo largo de la banda disponible, acontece independientemente en cada una de las capas, y evita que una secuencia sucesiva de ceros o unos sean transmitidos, garantizando así una transmisión binaria adecuada.
- **Codificación Externa.-** Su función es introducir redundancia para permitir la decodificación en el receptor y corregir los errores producidos en el canal de comunicación. Para lograr esto se lo hace con un código externo corrector de errores en bloques llamado *Reed Solomon,* trabaja con paquetes de 188 bytes.
- **Entrelazador.-** Es el encargado de romper largas cadenas de errores de transmisión producidas en cascada por ruidos impulsivos provocados en el canal y que no podrían ser corregidas por los decodificadores, esto lo logra aleatorizando los datos en el emisor y al existir una pérdida de datos en el canal, el receptor no se verá afectado por que los errores se dispersan [18].
- **Codificación Interna.-** Tiene una función similar a la codificación externa, su objetivo es aumentar la robustez del sistema frente a interferencias de canal atreves de la inserción de redundancias, el código externo trabaja con bytes y el código interno opera con bits.
- **Modulador.-** Aquí si existe una gran diferencia con el resto de estándares ya que en ISDB-Tb se emplea un sistema con múltiples portadoras y modulación MQAM, denominada COFDM, por tanto este bloque es el encargado de la modulación de las señales para que pueda ser posible la transmisión de las mismas, el sistema COFDM presenta una alta robustez a los múltiples trayectos del canal, una vez que las señales a ser transmitidas son distribuidas en N sub-canales de banda estrecha.
- **Transmisor.-** Este bloque es el responsable de trasladar la señal de banda base a un canal de frecuencia y también de proporcionar la ganancia de potencia para la cobertura deseada.

Para poder describir más detalladamente los tipos de modulación, las capas y los modos de transmisión se muestra en la Figura 4.2 un diagrama de bloques de un trasmisor ISDB-Tb [20].

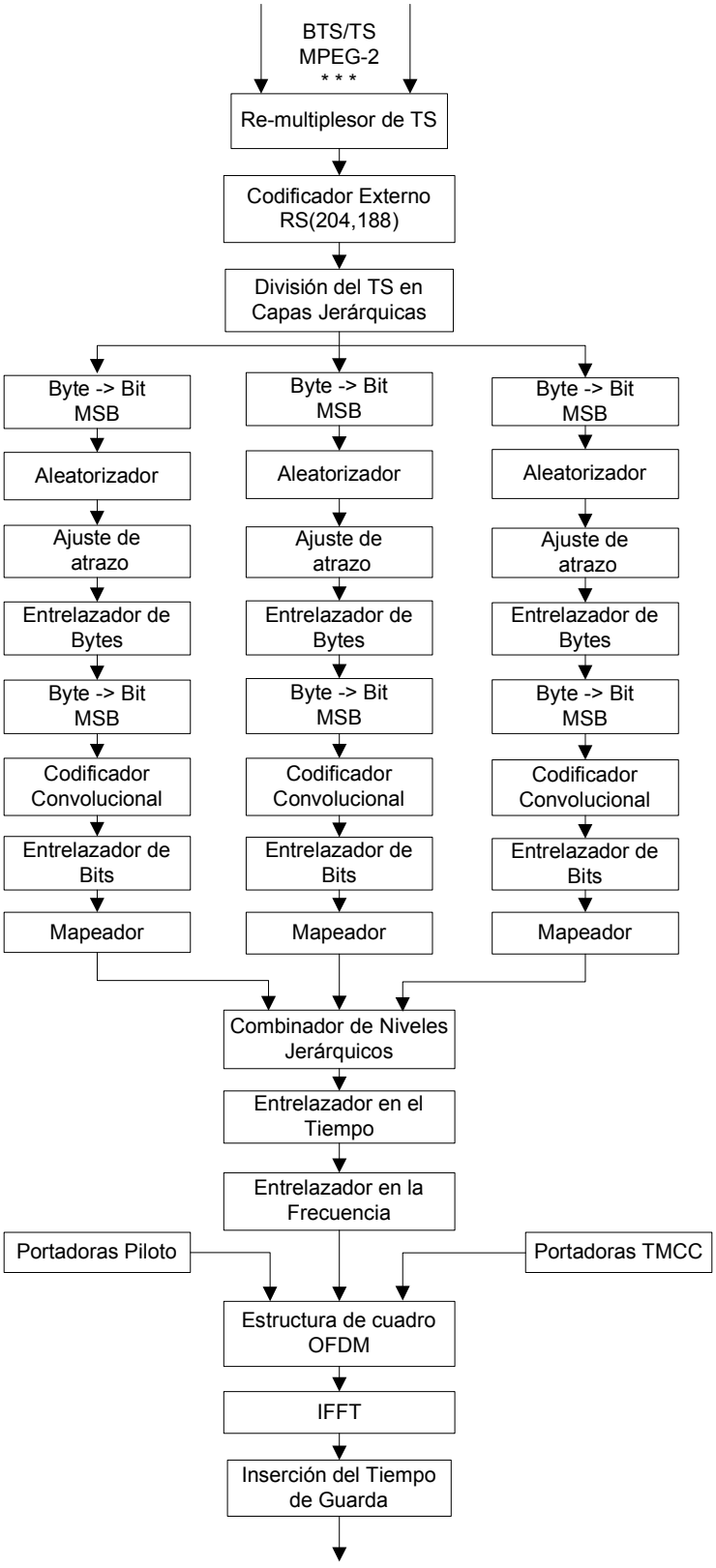

**Figura. 4.2. Diagrama de bloques de un Transmisor ISDB-Tb**

### 3.1.1 Re-Multiplexor

El re-multiplexor es la primera etapa del sistema de transmisión, este es el responsable de recibir el flujo de datos y decidir correctamente en cuál de las tres capas jerárquicas debe ser transmitido un paquete.

El flujo de entrada puede ser un TS MPEG-2 (flujo formado por paquetes de 188 bytes) o por un BTS (flujo formado por paquetes de 204 bytes). Cada paquete antes de ser encaminado a su destino, pasa por un codificador *Reed Solomon*, RS (204,188), que inserta 16 bytes de paridad a cada 188 bytes [14],[15]. En la Figura 4.3 se ilustra el funcionamiento del re-multiplexor descrito, en conjunto con el codificador *Reed Solomon*, clasificado como codificador externo.

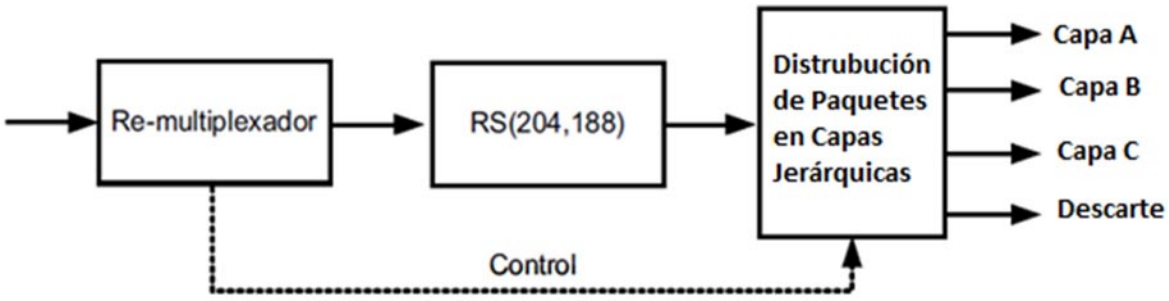


**Figura. 4.3. Re-multiplexor y Codificador RS**

### 3.1.2 Segmentación del Canal y Capas

El estándar ISDB-T transmite hasta en tres capas independientes empleando la estructura de segmentación de banda, esta estructura divide un canal de 6 MHz en 14 segmentos de los cuales se ocupa 13 segmentos de 429Khz cada uno [19], los mismos que pueden ser configurados de acuerdo a las necesidades del radiodifusor, esto permite una gran flexibilidad en la elección de los servicios ofrecidos, un ejemplo de esto se muestra en la Figura 4.4.

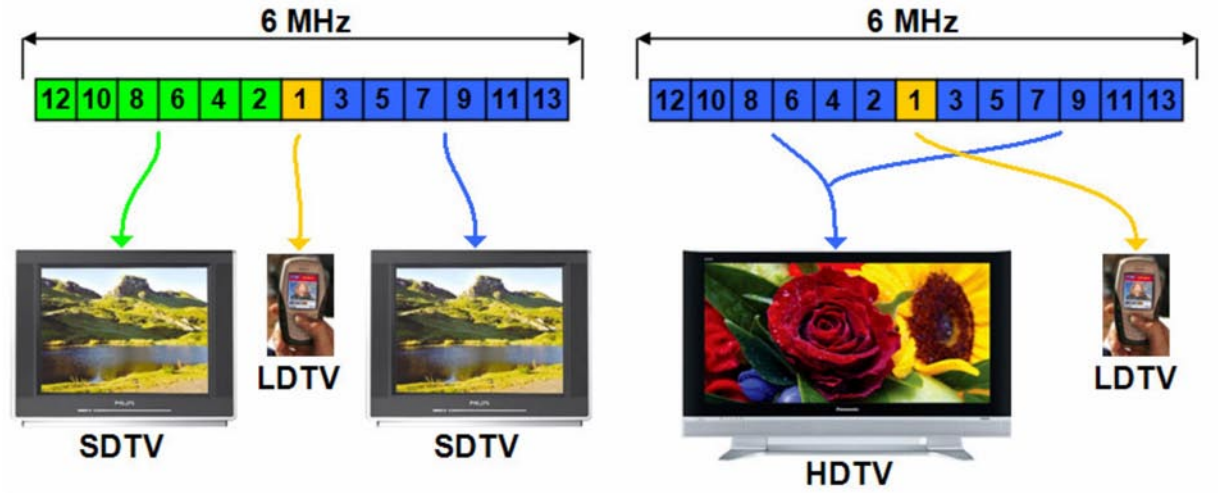


**Figura. 4.4. Ejemplo de segmentación de canal**

Las capas A, B y C de acuerdo al número de segmentos se las puede apreciar en la Figura 4.5.

**Figura. 4.5. Transmisión Jerárquica Tres Capas Activas**

La configuración recomendada de las capas para la transmisión se muestra en la Tabla 4.1 [20].

**Tabla. 4.1. Ejemplo de configuración recomendada**

| Capa | Número de segmentos | Tasa de codificación | Modo de Operación | Modulación | Tasa de video transmitida |
|---|---|---|---|---|---|
| A | 1 | 2/3 | Modo 1 | QPSK | LD (One-seg) 400 Kbps |
| B | 8 | 3/4 o 5/6 | Modo 3 | 64QAM | HD 15 Mbps |
| C | 4 | 3/4 o 5/6 | Modo 2 | 16QAM | SD 6 Mbps |

### 3.1.3 Modos de Operación

La técnica de transmisión empleando múltiples portadoras ortogonales codificadas (COFDM) aumenta la robustez de sistema frente a los múltiples trayectos cuando el número de subportadoras es suficientemente elevado. En el caso del estándar ISDB-T, existen tres modos de operación distintos, la Tabla 4.2 muestra el número de portadoras empleadas, de portadoras útiles y el tiempo efectivo de símbolo, en cada uno de los modos de operación.

**Tabla. 4.2. Modos de Operación**

| | **MODO 1** | **MODO 2** | **MODO 3** |
|---|---|---|---|
| Número de portadores útiles | 1405 | 2809 | 5617 |
| Número total de portadoras | 2048 | 4096 | 8192 |
| Tiempo efectivo de símbolo | 252us | 504us | 1008us |

### 3.1.4 Aleatorizador o Dispersión de Energía

La aleatorización se realiza mediante una operación OR-Exclusiva bit a bit entre los datos que son entregados y una secuencia pseudo-aleatoria. Esta operación garantiza una dispersión de energía de la señal en el dominio de la frecuencia, ya que convierte los datos en una salida con características aleatorias [16],[17].

Cuando más concentrada en pequeñas bandas este la potencia de la señal, más susceptible será a las interferencias de banda estrecha o al desvanecimiento selectivo causado por los multitrayectos, la aleatorización ayuda a reducir la influencia de estos fenómenos en la señal.

A continuación en la Figura 4.6 se muestra un circuito generador de secuencias pseudo-aleatorias.

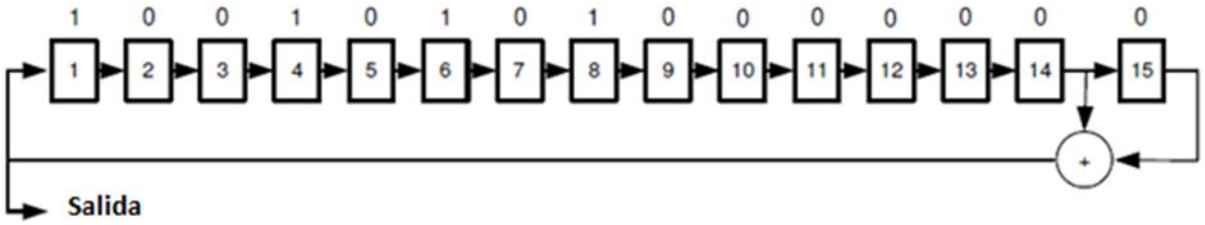


**Figura. 4.6. Generador de secuencias pseudo-aleatorias.**

### 3.1.5 Ajuste de Retraso

Esta etapa tiene como objetivo hacer que el atraso insertado por el Entrelazador sea idéntico entre las diferentes capas del sistema. Existe dos tipos de ajuste de retraso una para bytes y otra para bits. De acuerdo con la configuración de cada capa, un determinado atraso debe ser insertado, para ejemplo se muestra en la Tabla 4.3 un ajuste de retaso en Bits, donde *Ns* es el número de segmentos [14],[15].

**Tabla. 4.3. Ajuste de atraso en Bits**

| Mapeamiento | Modo 1 | Modo 2 | Modo 3 |
|---|---|---|---|
| **QPSK** | (384$xN_s$)-240 | (760$xN_s$) -240 | (1536$xN_s$) -240 |
| **16QAM** | (768$xN_s$)-480 | (1536$xN_s$) -480 | (3072$xN_s$) -480 |
| **64QAM** | (1152$xN_s$)-720 | (2304$xN_s$) -720 | (4608$xN_s$) -720 |

### 3.1.6 Entrelazador Externo

El Entrelazador o *Interleaving* usado en el sistema ISDB-T es de tipo convolucional y es compuesto por B ramas; cada rama posee un determinado número de retrasos y el número de retrasos por rama sigue un orden determinado M. La primera rama tiene 0 retrasos, la segunda tiene M retrasos, la tercera rama tiene 2xM atrasos, y así sucesivamente, hasta llegar a la última rama que tiene (B-1) x M retrasos.

El Entrelazador del estándar ISDB-T se muestra en la Figura 4.7, y posee retrasos M=17 y ramas B=12.

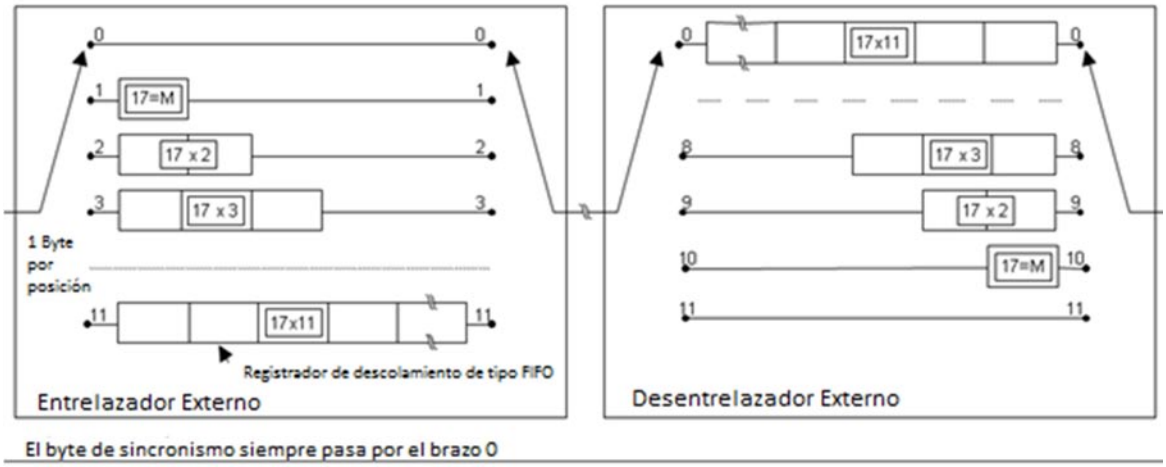


**Figura. 4.7. Estructura del Entrelazador de bytes.**

### 3.1.7 Código Interno

En el estándar ISDB-T, el sistema puede seleccionar una banda de códigos convolucionales de punteo, basados en un código convolucional madre de tasa 1/2 con 64 estados. Los polinomios generadores para el código madre están dados por:

- $G1(X)=1+X+X^2+X^3+X^6$
- $G2(Y)=1+Y^2+Y^3+Y^5+Y^6$

La Figura 4.8 ilustra un diagrama de bloques de un codificador de tasa ½, cuya profundidad de código es igual a 7, En este codificador, los bits de datos serán codificados, siguiendo una secuencia de registradores. El codificador posee dos salidas, X & Y, definidas por las operaciones realizadas entre la salida de algunos registradores, en función de G1 y G2 [21].

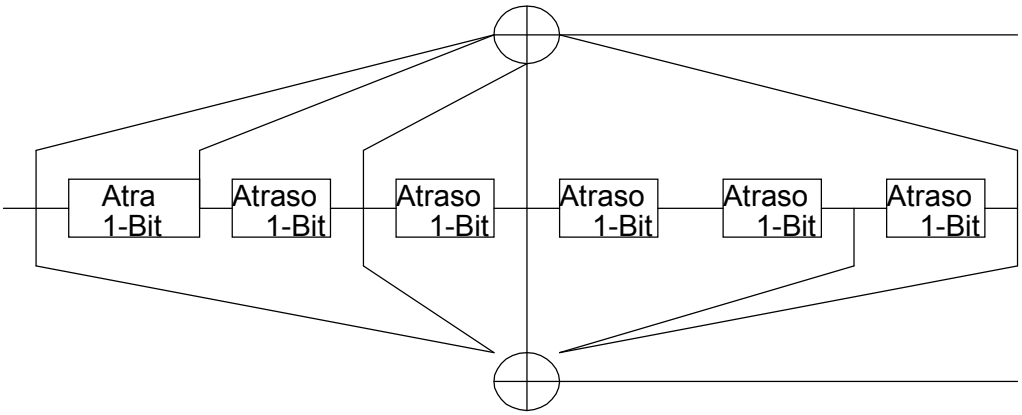


**Figura. 4.8. Codificador convolucional de tasa 1/2**

Es posible configurar una tasa de codificación con diferentes valores, aumentado la tasa de código y disminuyendo el número de bits enviados reduciendo la protección contra errores. Las diferentes tasas de código son obtenidas a partir del uso de una técnica denominada punteo, consiste en excluir algunos bits de la salida del codificador. Las reglas de punteo se muestran en la tabla 4.4.

**Tabla. 4.4. Punteo Codificador Convolucional**

| Tasa de código R | Patrón de Punteo | Secuencia de Transmisión |
|---|---|---|
| ½ | X: 1<br>Y: 1 | $X_1 Y_1$ |
| 2/3 | X: 1 0<br>Y: 1 1 | $X_1 Y_1 Y_2$ |
| ¾ | X: 1 0 1<br>Y: 1 1 0 | $X_1 Y_1 Y_2 X_3$ |
| 5/6 | X: 1 0 1 0 1<br>Y: 1 1 0 1 0 | $X_1 Y_1 Y_2 X_3 Y_4 X_5$ |
| 7/8 | X: 1 0 0 0 1 0 1<br>Y: 1 1 1 1 0 1 0 | $X_1 Y_1 Y_2 Y_3 Y_4 X_5 Y_6 X_7$ |

### 3.1.8 Entrelazador de Bits

Los bits son agrupados al entrelazador de acuerdo con el mapeamiento seleccionado y cada uno de estos entrelazadores tienen una estructura que se puede ver en la Figura 4.9, el bit más significativo es encaminado a la rama sin atraso.

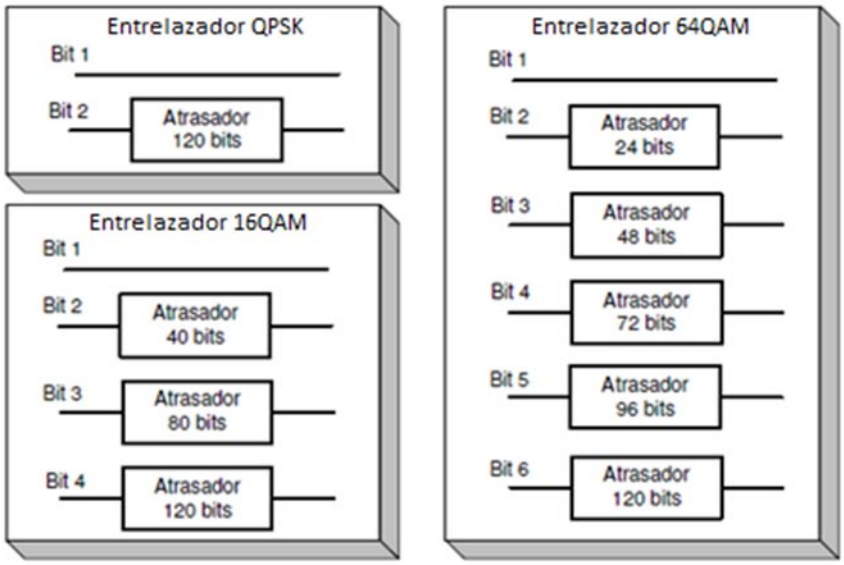


**Figura. 4.9. Estructura de los entrelazadores de bits**

### 3.1.9 Mapeamiento

La modulación convierte la información entregada por el codificador en ondas electromagnéticas. El mapeamiento se lo realiza de acuerdo al tipo de modulación escogido, el estándar ISDB-T emplea 4 tipos diferentes de modulaciones digitales:

a) **DQPSK:** (*Differential Quadrature Phase Shift Keying*). Tiene la característica de transmitir su información en la diferencia de fase entre el símbolo actual y el símbolo anterior. Este esquema utiliza 8 posibles símbolos para cargar la información de 2 bits. Su diagrama de constelación se muestra en la Figura 4.10.

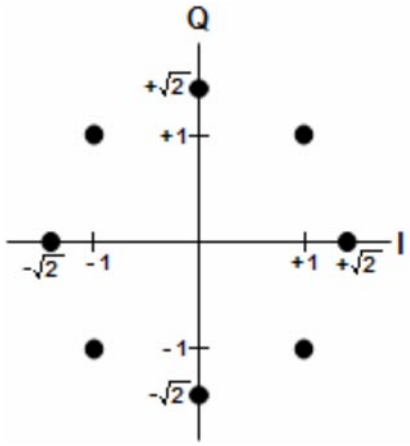


**Figura. 4.10. Diagrama de constelación DQPSK**

b) **QPSK:** (*Quadrature Phase Shift Keying*). La secuencia de bits en serie procedentes del codificador interno se divide en dos canales paralelos llamados canales I y Q de canal, la fase de la portadora transmite dos bits por símbolo.Su diagrama de constelación se muestra en la Figura 4.11.

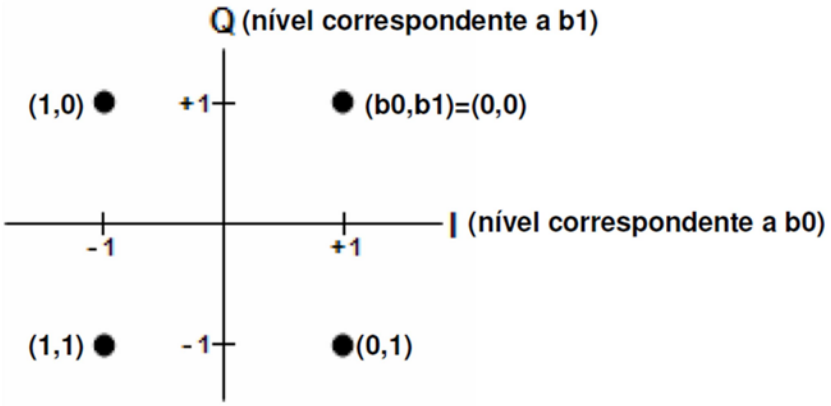


**Figura. 4.11. Diagrama de constelación QPSK**

c) **16QAM:** (*Quadrature Amplitude Modulation*). Está formado por 16 símbolos y utiliza tanto la fase como la amplitud de la portadora para transmitir 4 bits por símbolo. Su diagrama de constelación se muestra en la Figura 4.12.

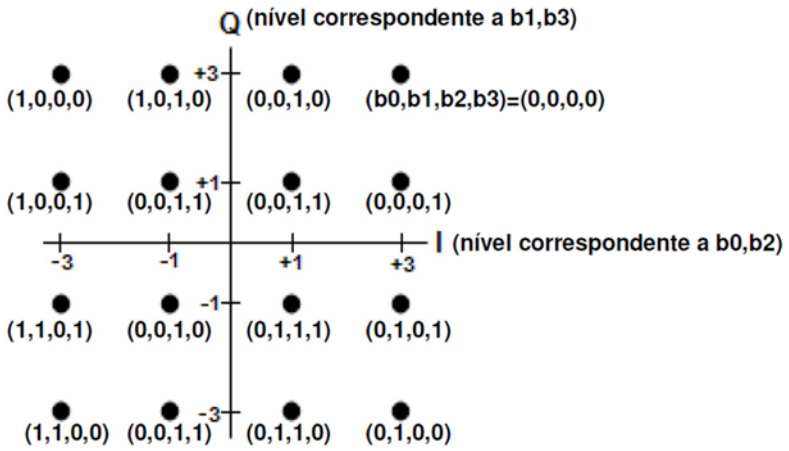


**Figura. 4.12. Diagrama de constelación 16QAM**

d) **64QAM:** (*Quadrature Amplitude Modulation*). Al igual que la modulación 16QAM, guarda información tanto en fase como en amplitud de su portadora, para transmitir 6 bits por símbolo.Su diagrama de constelación se muestra en la Figura 4.13.

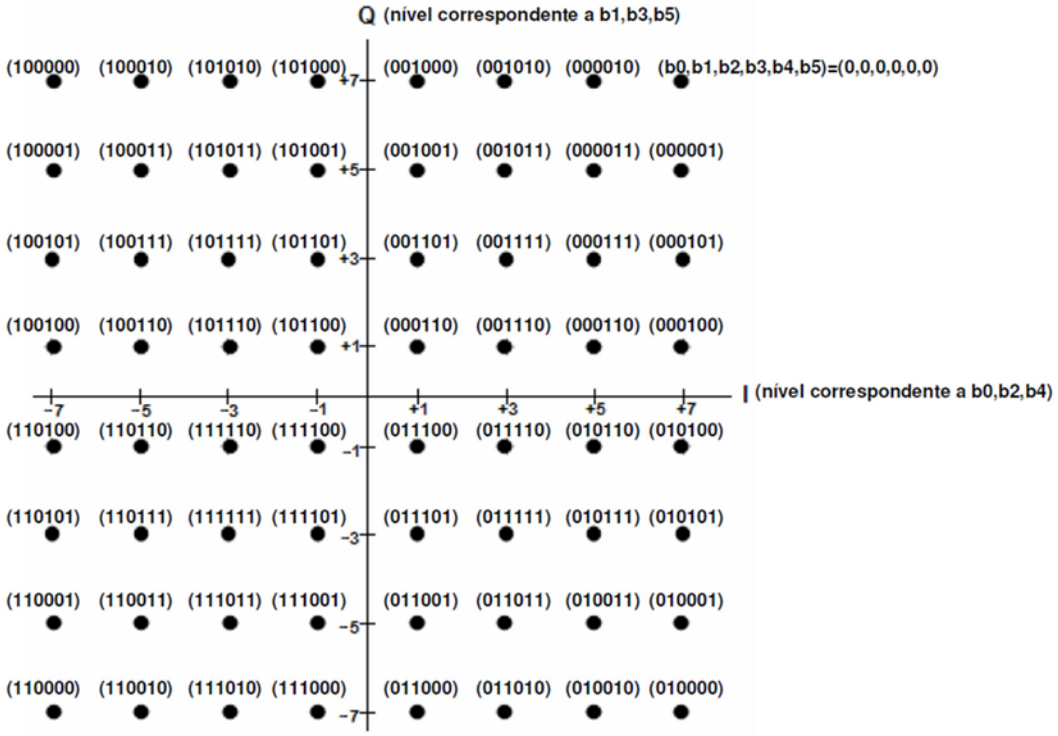


**Figura. 4.13. Diagrama de constelación 64QAM**

### 3.1.10 Combinador De Capas Jerárquicas

La señal que recibe el combinador está formada por una sucesión de símbolos OFDM, y cada uno de estos símbolos, está formado por un conjunto de portadoras de datos y portadoras de sistema. Cada símbolo OFDM es formado por subconjuntos de portadoras, llamados segmentos.

Los datos son organizados para formar los segmentos, los símbolos complejos resultantes del proceso de mapeamiento de las diferentes capas, son combinados para entregar los datos en el orden necesario y generar los símbolos OFDM.

Cada una de las capas del transmisor es responsable por un número configurable de segmentos, esa configuración es válida siempre que el número de segmentos resultantes de la suma de las capas A, B y C sea igual a 13. La Capa A es la responsable de los primeros segmentos de los símbolos OFDM, seguidos por los segmentos de la capa B, y finalmente por los de la capa C. En la Figura 4.14 se muestra la organización de los datos que realiza el combinador [20].

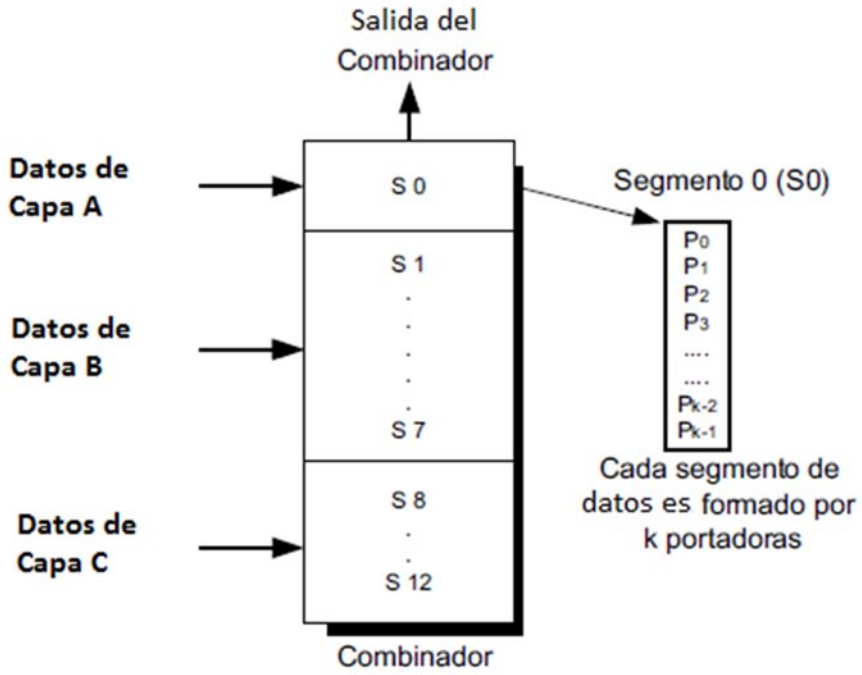


**Figura. 4.14. Ejemplo de organización de datos en el combinador**

### 3.1.11 Entrelazador de Tiempo

El entrelazador temporal mezcla los símbolos mapeados a través de la inserción de diferentes atrasos en cada símbolo, de forma que una secuencia de datos no se transmita al mismo tiempo, aunque estén en portadoras diferentes. Esta acción es tomada para evitar que desvanecimientos en cascada como el ruido impulsivo dañen la señal. En la Figura 4.15 se muestra un ejemplo del entrelazador de tiempo.

**Figura. 4.15. Ejemplo del entrelazamiento temporal sobre un ruido impulsivo**

El valor asumido de atraso es definido al inicio de la transmisión, a través de la variable “I”, que asume algunos valores predeterminados, dependiendo del modo de trasmisión adoptado e del grado de protección escogido. Cuando mayor es el valor de “I”, más protegida estará la señal a ser trasmitida, pero más atraso tendrá tal como se muestra en la Tabla 4.5.

**Tabla. 4.5. Atrasos en el entrelazamiento temporal**

| **Modo 1** | | | **Modo 2** | | | **Modo 3** | | |
|---|---|---|---|---|---|---|---|---|
| Tamaño (I) | Número de símbolos OFDM atrasados | Atraso (ms) | Tamaño (I) | Número de símbolos OFDM atrasados | Atraso (ms) | Tamaño (I) | Número de símbolos OFDM atrasados | Atraso (ms) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | 380 | 95,7 | 2 | 190 | 95,7 | 1 | 95 | 95.7 |
| 8 | 760 | 191,5 | 4 | 380 | 191,5 | 2 | 190 | 191,5 |
| 16 | 1504 | 379,0 | 8 | 760 | 379 | 4 | 380 | 379,0 |

### 3.1.12 Entrelazador de Frecuencia

El entrelazador de frecuencia mezcla las portadoras de los segmentos, con el fin de dar un aspecto aleatorio al espectro de frecuencias, esto se hace para reducir los efectos destructivos del canal en la señal transmitida. Cuando ocurre un fading de multitrayecto este alcanza a portadoras de diferentes segmentos y no afecta a todas las portadoras de un mismo segmento como se muestra en la Figura 4.16.

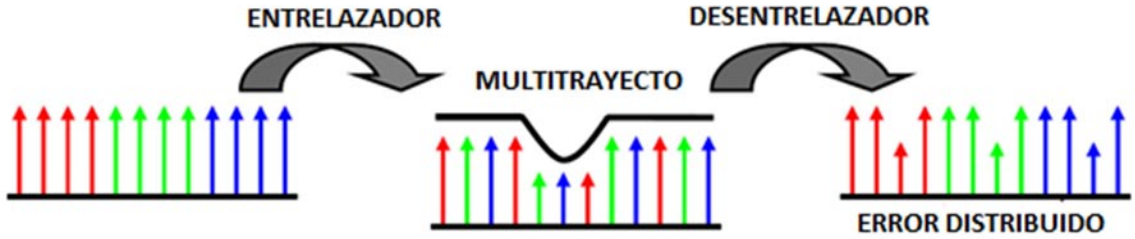


**Figura. 4.16. Ejemplo del entrelazamiento en frecuencia**

### 3.1.13 Señal OFDM

El sistema de transmisión utiliza segmentación de banda con la técnica OFDM. Y OFDM hace uso de múltiples portadoras ortogonales enviadas simultáneamente en el canal de transmisión, el estándar ISDB-T permite configurar la cantidad de portadoras que componen los símbolos OFDM transmitidos tal como se menciono en la sección 4.1.3.

En esta sección se describe que existen portadoras que no transportan datos útiles e información del sistema, estas portadoras que trasportan información nula son necesarias para completar el total de las portadoras de la IFFT.

La Transformada Rápida de Fourier Inversa (IFFT) es utilizada en el transmisor para generar los símbolos y la Transformada Rápida de Fourier (FFT) es utilizada para detectar los mismos en el receptor.

**Tiempo de Guarda**

El tiempo de guarda aumenta la robustez frente a los multitrayectos y a la interferencia intersimbólica (ISI); el intervalo de tiempo consiste en hacer una copia de la parte final de un símbolo OFDM, al inicio del mismo símbolo; como se muestra en la Figura 4.17. El tiempo efectivo del símbolo ($Tu$) es obtenido a la salida del bloque de la IFFT. Parte de este símbolo es copiado para la generación del tiempo de guarda ($\Delta$); entonces la duración del símbolo será ($Tu + \Delta$) [21].

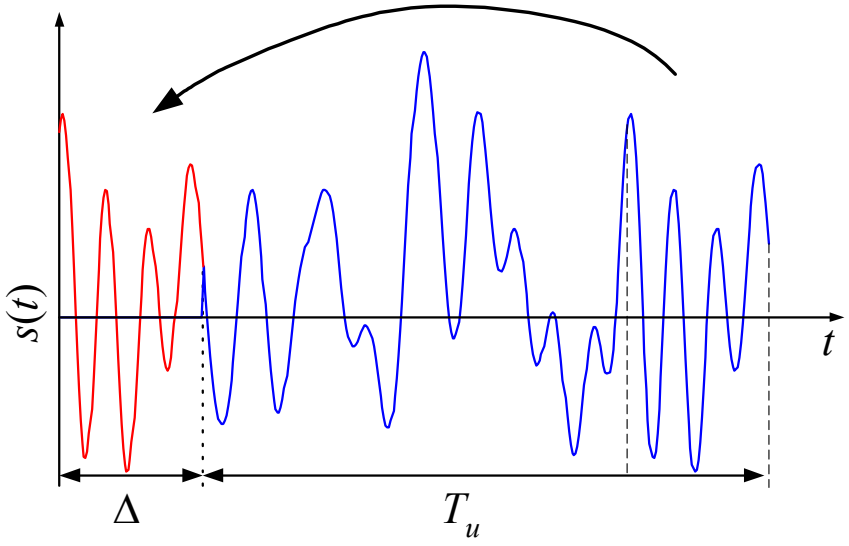


**Figura. 4.17. Tiempo de guarda**

A su vez un cuadro OFDM, es formado por 204 símbolos OFDM por lo que tendrá una duración 204 veces mayor como se muestra en la Tabla 4.6.

Se puede configurar la duración del intervalo de guarda seleccionando una fracción de tiempo de símbolo OFDM efectivo, las opciones disponibles del sistema son $1/4$, $1/8$,$1/16$ y $1/32$.

El intervalo de guarda reduce el flujo del sistema, cuando mayor sea el tiempo de guarda, mayor es el tiempo de dispersión introducido por el canal, por ende menor será el flujo del sistema [20].

**Tabla. 4.6. Duración de símbolo y de cuadro OFDM**

| **MODO** | **1** | | | |
|---|---|---|---|---|
| Tu – Tiempo efectivo de símbolo (us) | 252 | | | |
| Intervalo de guarda | 1/4 | 1/8 | 1/16 | 1/32 |
| Tg – Duración tiempo de Guarda (us) | 63 | 31,5 | 15,76 | 7,32 |
| Duración de símbolo OFDM (us) | 315 | 283,5 | 267,75 | 259,875 |
| Duración de cuadro OFDM (ms) | 64,26 | 57,83 | 54,621 | 53,0145 |
| **MODO** | **2** | | | |
| Tu – Tiempo efectivo de símbolo (us) | 504 | | | |
| Intervalo de guarda | 1/4 | 1/8 | 1/16 | 1/32 |
| Tg – Duración tiempo de Guarda (us) | 126 | 63 | 31,5 | 15,75 |
| Duración de símbolo OFDM (us) | 630 | 567 | 533,5 | 519,75 |
| Duración de cuadro OFDM (ms) | 128,52 | 115,668 | 109,242 | 106,029 |
| **MODO** | **3** | | | |
| Tu – Tiempo efectivo de símbolo (us) | 1008 | | | |
| Intervalo de guarda | 1/4 | 1/8 | 1/16 | 1/32 |
| Tg – Duración tiempo de Guarda (us) | 252 | 126 | 63 | 31,5 |
| Duración de símbolo OFDM (ms) | 1,26 | 1,134 | 1,071 | 1,0395 |
| Duración de cuadro OFDM (ms) | 257,04 | 231,336 | 218,484 | 212,058 |

**Formación de símbolo y cuadro OFDM**

El símbolo OFDM es formado por portadoras que transportan los datos a ser trasmitidos, más portadoras especiales que trasportan información adicional del sistema. Las portadoras adicionales son clasificadas en SP's (*Scattered Pilot*, Piloto dispersas), CP's (C*ontinual Pilot*, Pilotos Continuas), AC's (*Auxiliary Channel*, Canal Auxiliar) y las TMCC's (*Transmission and Multiplexing Configuration Control,*

Portadoras de de parámetros de transmisión, multiplexación, control y configuración). Una señal para ser trasmitida debe ser organizada en cuadros, cada cuadro consiste en 204 símbolos OFDM.

**Portadoras TMCC**

Las portadoras TMCC's poseen un papel importante, ya que a traves de ellas son transmitidas los parámetros de transmisión del sistema como el numero de segmentos de las capas, el esquema de modulación, la tasa de códigos y la profundidad del entrelazador temporal, además de eso las portadoras TMCC son utilizadas en la sincronía de los cuadros OFDM.

Los parámetros de configuración son generados en el multiplexor con los paquetes IIP (*ISDB-Tb Information Packets*) que son los responsables de la configuración del trasmisor.

Las portadoras TMCC pertenecientes a un mismo símbolo OFDM transportan la misma información. Portando 204 símbolos o bits por cuadro OFDM, como se muestra en la figura 4.18, independientemente del modo y del número de TMCC's transmitidas por segmento.

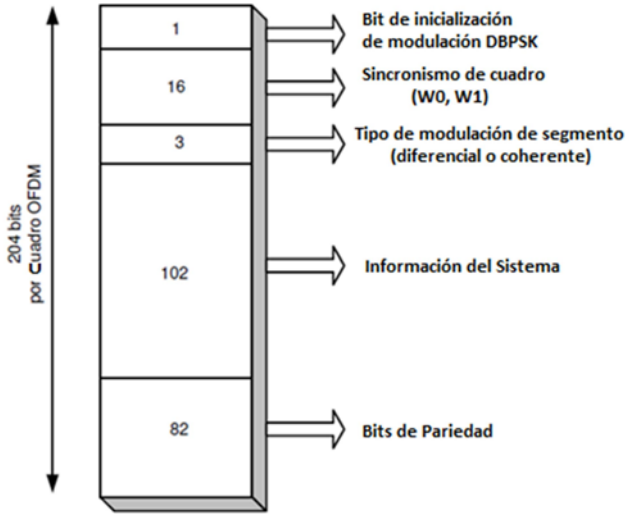


**Figura. 4.18. Formación en bits del cuadro TMCC**

Los 102 bits de información de sistema transportan las configuraciones que serán utilizados en la recepción para que el receptor pueda configurarse de la misma manera que el trasmisor y así decodificar correctamente la señal recibida. En la Tabla 4.7 se muestra una descripción de los 102 bits de información de sistema [20].

**Tabla. 4.7. TMCC 102 bits de información del sistema**

| Bits de TMCC | DESCRIPCIÓN |
|---|---|
| 20 al 21 | Identificación del Sistema |
| 22 al 25 | Indicación de cambio de parámetros |
| 26 | Identificación de alarma de emergencia |
| **Configuraciones actuales** | |
| 27 | Indicación de recepción parcial |
| 28 al 40 | Configuración de capa A |
| 41 al 53 | Configuración de capa B |
| 54 al 66 | Configuración de capa C |
| **Próximas configuraciones** | |
| 67 | Indicación de recepción parcial |
| 68 al 80 | Configuración de capa A |
| 81 al 93 | Configuración de capa B |
| 94 al 106 | Configuración de capa C |
| **Reservados** | |
| 107 al 109 | Reservados (Para radio digital) |
| 110 al 121 | Reservados |

### 3.1.14 Capacidad de Transmisión del Sistema

La capacidad de transmisión del sistema varía de acuerdo con las configuraciones seleccionadas tomando en cuenta el número de segmentos que vaya a utilizar, la Tabla 4.8 muestra la taza útil de datos en Mbps considerando los 13 segmentos [20].

**Tabla. 4.8. Tasa útil de datos en 13 segmentos**

| Modulación | Código Convolucional | Número de Portadoras (MODO 1/2/3) | Tasa de información (Mbps) | | | |
|---|---|---|---|---|---|---|
| | | | Tiempo de Guarda (1/4) | Tiempo de Guarda (1/8) | Tiempo de Guarda (1/16) | Tiempo de Guarda (1/32) |
| DQPSK QPSK | 1/2 | 156/312/624 | 3,651 | 4,056 | 4,295 | 4,425 |
| | 2/3 | 208/216/832 | 4,868 | 5,409 | 5,727 | 5,900 |
| | 3/4 | 234/468/936 | 4,476 | 6,085 | 6,443 | 6,638 |
| | 5/6 | 260/520/1040 | 6,085 | 6,761 | 7,159 | 7,376 |
| | 7/8 | 273/546/1092 | 6,389 | 7,099 | 7,517 | 7,744 |
| 16QAM | 1/2 | 312/624/1248 | 7,302 | 8,113 | 8,590 | 8,851 |
| | 2/3 | 416/832/1664 | 9,736 | 10,818 | 11,454 | 11,801 |
| | 3/4 | 468/936/1872 | 10,953 | 12,170 | 12,886 | 13,276 |
| | 5/6 | 520/1040/2080 | 12,170 | 13,522 | 14,318 | 14,752 |
| | 7/8 | 546/1092/2184 | 12,779 | 14,198 | 15,034 | 15,489 |
| 64QAM | 1/2 | 468/936/1872 | 10,953 | 12,170 | 12,886 | 13,276 |
| | 2/3 | 624/1248/2496 | 14,604 | 16,277 | 17,181 | 17,702 |
| | 3/4 | 702/1404/2808 | 16,430 | 18,255 | 19,329 | 19,915 |
| | 5/6 | 780/1560/3120 | 18,255 | 20,284 | 21,477 | 22,128 |
| | 7/8 | 819/1638/3276 | 19,168 | 21,298 | 22,551 | 23,234 |

Todos los parámetros que se mencionaron anteriormente son muy importantes conocerlos ya que se los aplicaran al momento de configurar por software al trasmisor, para esto en la Tabla 4.9 se muestra un resumen de los parámetros que hasta ahora se han visto [21].

**Tabla. 4.9. Resumen parámetros del trasmisor ISDB-T**

| MODO | MODO 1 | MODO 2 | MODO 3 |
|---|---|---|---|
| Número de segmentos $Ns$ | 13 | | |

| Ancho de banda $B = Ns.Bs + \Delta f$ | | 5,575 MHz | 5,573 MHz | 5,572 MHz |
|---|---|---|---|---|
| Número de portadoras | Datos $Npd$ | $96Ns = 1248$ | $192Ns = 2496$ | $384Ns = 4992$ |
| | Piloto $Npp$ | $9Ns$ | $18Ns$ | $36Ns$ |
| | Control $Npc$ | $Ns$ | $2Ns$ | $4Ns$ |
| | Auxiliar $Npa$ | $2Ns$ | $4Ns$ | $8Ns$ |
| | Total $Np$ | $108Ns + 1 = 1405$ | $216Ns + 1 = 2809$ | $432Ns + 1 = 5617$ |
| Esquema de modulación | | QPSK | 16QAM | 64QAM |
| Símbolos por cuadro $Nq$ | | 204 | | |
| Intervalo de símbolo $Ts$ | | 252 $us$ | 504 $us$ | 1008 $us$ |
| Intervalo de guarda $Tg$ | | 63 $us$ (1/4)<br>31,5 $us$ (1/8)<br>15,75 $us$ (1/16)<br>7,875 $us$ (1/32) | 126 $us$ (1/4)<br>63 $us$ (1/8)<br>31,5 $us$ (1/16)<br>15,75 $us$ (1/32) | 252 $us$ (1/4)<br>126 $us$ (1/8)<br>63 $us$ (1/16)<br>31,5 $us$ (1/32) |
| Intervalo de cuadro $Tq = Nq(Te + Tg)$ | | 64,26 $ms$ (1/4)<br>57,384 $ms$ (1/8)<br>54,621 $ms$ (1/16)<br>7,875 $ms$ (1/32) | 128,52 $ms$ (1/4)<br>115,668 $ms$ (1/8)<br>109,242 $ms$ (1/16)<br>106,029 $ms$ (1/32) | 257,04 $ms$ (1/4)<br>231,336 $ms$ (1/8)<br>218,484 $ms$ (1/16)<br>212, 058 $ms$ (1/32) |
| Código interno | | Convolucional (n=9792; r=1/2, 2/3,3/4, 5/6, 7/8) | | |
| Código externo | | Reed Solomon (n=204, k=188) | | |

### 3.1.15 Recepción de un Servicio de Tv Digital con Interactividad

Para la recepción de TS con interactividad, es necesario configurar el STB para qué descargue automáticamentela aplicación interactiva, en la parte inferior derecha aparecerá una barra con un proceso donde indica que se está cargando la aplicación como muestra en la Figura 4.29.

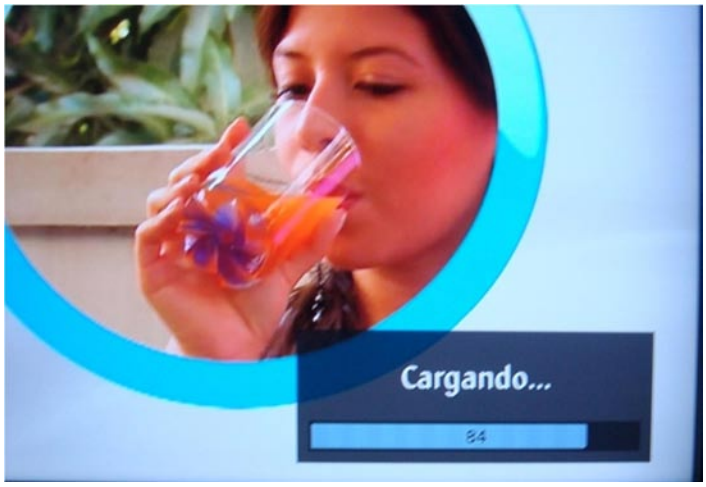


**Figura. 4.29. Proceso de carga de una aplicación interactiva**

En el caso de no tener activa esta opción de carga automática en el STB mediante el control remoto se puede aplastar Menú, luego seleccionar Ginga y se nos desplegara la lista de las aplicaciones como se puede ver en la Figura 4.30, en este caso en el STB del laboratorio se tiene cargado tres tipos de aplicaciones las que están, con la letra E son aplicaciones embebidas en el decodificador es decir que están cargadas en su memoria interna; las que están con la letra N, son las aplicaciones que se encuentran en un servidor FTP y que necesitan estar conectadas a una red para acceder a las mismas, y por último las aplicaciones que se denotan con la letra A son las que el STB está recibiendo por el aire, y se observa que en nombre de la aplicación es “APP_GINGA” como se la llamo al momento de generar la tabla.

**Figura. 4.30. Lista de Aplicaciones Interactivas disponibles**

En la Figura 4.31, se puede observar la aplicación interactiva que se cargó por el aire, esta trata sobre una Guía Hospitalaria de Quito, en la cual el televidente puede acceder a información como dirección, teléfonos y especialidades de los hospitales más importantes de la ciudad.

**Figura. 4.31. Aplicación Interactiva Guía Hospitalaria**

### 3.1.16 Recepción de dos Servicios de Tv Digital con Interactividad

Para la recepción de dos servicios de Tv digital, es necesario volver a sintonizar los canales, para actualizar la lista de canales, como se observa en la Figura 4.32, en este caso se tiene 4 canales en la lista, 2 de ESPETV y 2 de Ecuador TV.

**Figura. 4.32. Lista de Canales Disponibles**

Ahora se tiene disponible las dos programaciones diferentes ESPETV1 y ESPETV2 respectivamente, en el canal ESPETV2 se puede observar que posee una aplicación interactiva de nombre “APP_GINGA2” como se observa en la Figura 4.33.

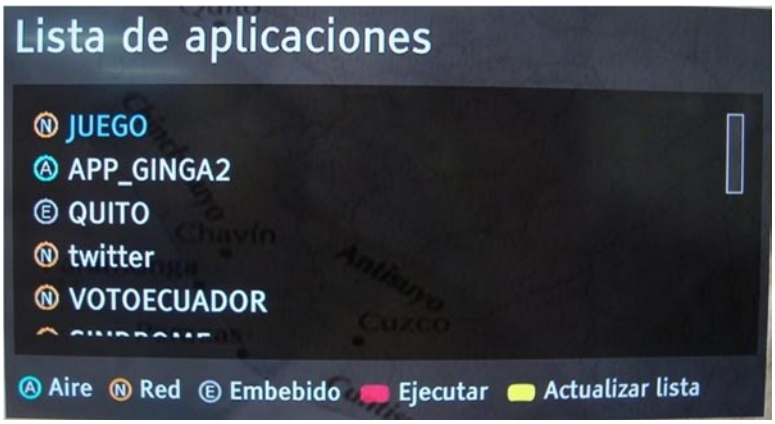


**Figura. 4.33. Aplicación APP_GINGA2 disponible en el canal ESPETV2**

Finalmente en la Figura 4.34, se observa la aplicación que se cargó en el canal ESPETV2, esta trata de una guía turística de la ciudad de Ibarra, donde el televidente puede tener información sobre dónde hospedarse, lugares de diversión, como llegar y lugares de alimentación.

**Figura. 4.34. Guía Turística de la Ciudad de Ibarra**

# CAPITULO 5

## CONCLUCIONES Y RECOMENDACIONES

### 4.1 CONCLUCIONES

- En este proyecto se propuso un modelo para la implementación de un laboratorio de televisión digital del sistema ISDB-Tb, teniendo como objetivo generar nuestra propia programación con interactividad, y poderla probar con un receptor que admita el estándar con aplicaciones interactivas.

- En este trabajo se presentaron conceptos e informaciones necesarias para la generación de flujos de datos (Transport Stream) que puedan ser emitidos con el estándar ISDB-Tb, volviéndose una referencia importante para aquellas personas que quieran iniciar sus investigaciones en esta área, ya que la solución que se presenta para la generación de TS se lo hace en software libre que está al alcance de todos.

- Las informaciones presentadas de forma sintetizada y organizada en este trabajo facilitan el estudio y compresión de diversos conceptos que forman la base principal para la implementación y posterior configuración de los parámetros de transmisión.

- Para poder configurar los parámetros de transmisión fue necesario conocer etapa por etapa el sistema de transmisión del estándar ISDB-T, para con esto

lograr adecuar la señal de transmisión de acuerdo a los servicios y requerimientos, además de poder manejar las tasas de transmisión, modulaciones, retrasos, y la robustez de la señal.

- Se demostró que la etapa de multiplexación de datos necesita de conocimientos acerca de la formación de paquetes TS y BTS, además de conocer sobre la estructura, organización de datos, la generación de tablas y referencias de tiempo que pasan a ser esenciales para la decodificación y exhibición de un programa, esta etapa es de suma importancia ya que pasa a ser una mediadora entre la etapa de producción o generación de contenidos con la etapa de transmisión.

- Se mostró una solución económicamente viable para poder empezar a probar las características de esta tecnología, además de esto abre nuevas puertas para que los estudiantes o investigadores que estén trabajando en esta área, ya que todavía existen muchas características que se puede ofrecer con televisión digital y que todavía están por desarrollarse como EPG, *Close Caption*, Alarmas de emergencias, etc.

- Como resultado del análisis de potencia se concluye que existe demasiada potencia perdida en los cables debido a su atenuación, por lo que se recomienda utilizar cables de corta distancia en el transmisor, y adquirir cables de mejor calidad con menor atenuación para no tener pérdidas tan grandes de potencia.

## 4.2 RECOMENDACIONES

- Para futuras investigaciones en el tema, específicamente en la parte de generación del TS, se recomienda realizar pruebas con un PlayOut comercial, con el fin de comparar los resultados obtenidos.

- Se tiene que adquirir un receptor móvil o un receptor portátil USB para comenzar con las pruebas de transmisión en one-seg, y si el receptor móvil tiene un sistema operativo androide o a su vez ya viene embebido ginga, se podría incluso realizar pruebas de recepción de interactividad en dispositivos móviles.

- Hasta ahora lo que se ha logrado es generación de TS partiendo de una fuente de audio y video previamente grabado pero pensando en el canal de televisión digital de la ESPE, se recomienda desde ya iniciar con la implementación de un trasmisor con los equipos necesarios para la emisión de audio y video en tiempo real.

- Se recomienda además adquirir un amplificador de potencia que logre cubrir un área considerable para poder trabajar en la parte de pruebas de campo y llegar a determinar los niveles de coberturas variando los parámetros de transmisión.

- La calidad de servicio QoS y la calidad de experiencia QoE, se debe incluir en futuros estudios con el fin de colaborar en las pruebas de calidad del estándar.

## REFERENCIAS BIBLIOGRÁFICAS

[1] Superintendencia de Telecomunicaciones Ecuador, “Adopción del estándar para TDT”, http://www.supertel.gob.ec/tdt-ecuador/, 2010, Fecha de consulta 10-01-2011.

[2] Televisión Digital Argentina, “Ventajas Televisión Digital”, http://www.minplan.gov.ar/html/tv_digital/a.php, 2010, Fecha de consulta 10-01-2011.

[3] Radio Victoria, “Televisión Digital”, http://www.radiovictoria.com/mas Novedades.php? img_nove=2&nove=57&idioma=1, 2010, Fecha de consulta 10-01-2011.

[4] Wikimedia Foundation, Inc., “Estandar ISDB-T”, http://es.wikipedia.org/wiki/ISDB-T, 2010, Fecha de consulta 12-01-2011.

[5] ISDB-T Perú, ”ISDB-T Internacional”, http://www.isdbt.org.pe/isdbt.php, 2010, Fecha de consulta 12-01-2011.

[6] Ciencia y Tecnología, “Televisión Digital Terrestre”, http://foros.cantv.net/ printer_friendly_posts.asp?TID=71681, 2010, Fecha de consulta 13-01-2011.

[7] Revista Lideres Ecuador, “Índice de penetración de Televisión en Ecuador”, http://www.revistalideres.ec/2009-03-30/Tecnologia/Noticia-Principal/LD090330P26 MICROCHIP.aspx, 2010, Fecha de consulta 13-01-2011.

[11] ISO/IEC 13818-1,”MPEG-2” http://webstore.iec.ch/preview/info_isoiec13818-1%7Bed3.0%7Den.pdf, Tercera Edicion 2007-10-15, Fecha de consulta 28-05-2011.

[12] ABNT NBR 15603-2, "Televisão Digital Terrestre - Multiplexação e serviços de informação (SI) Sintaxes e definições da informação básica de SI", primeira edição, Dezembro 2.007, Fecha de consulta 28-05-2011.

[13] Wikimedia Foundation, Inc., “Formatos de Audio y Video”, http://es.wikipedia.org/wiki/Formato_de_archivo_inform%C3%A1tico, Fecha de consulta 28-05-2011.

[14] ABNT NBR 15601, "Televisão Digital Terrestre - Sistema de transmissão", primera edición Noviembre 2007, Fecha de consulta 28-05-2011.

[15] ARIB Standard STD-B31, "Transmission system for digital terrestrial televisión broadcasting", ver.1.6, 2007, Fecha de consulta 28-05-2011.

[16] B. Sklar, “ *Digital Communication - Fundamentals and Applications”*, Prentice Hall, 2000, Fecha de consulta 29-05-2011.

[17] S. Haykin, “*Communication Systems*, John Wiley”, 2000, Fecha de consulta 29-05-2011.

[18] F. C. C. Castro, M. C. F. Castro, D. S.Arantes e Y. Iano, "Enfocando a Codificação de canal em transmissão 8-VSB para ATSC", *Telecomunicações*, Vol. 03, 2000, Fecha de consulta 29-05-2011.

[19] M. Takata and M saito, "Transmission System for ISDB-T", *Proceedings of the*IEE, Vol 94, 2006, Fecha de consulta 30-05-2011.

[20] Juliano Silveira Ferreira, “*Uma proposta de implementação do Multiplexador do sistema ISDB-TB”,* 2009, Fecha de consulta 15-06-2011.

[21] Prof. Dr. Luciano Leonel Mendes, “TV Digital – Fundamentos e o Padrão ISDB-$T_B$”, INATEL, 2011, Fecha de consulta 15-06-2011.

**FECHA DE ENTREGA**

El presente proyecto de grado fue entregado en la fecha.

Sangolquí, _____ 2011

Realizado por:

_____

María Augusta Illescas Robalino

_____

Diego Fernando Villamarín Zapata

_____

Ing. Gonzalo Olmedo

COORDINADOR INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES