



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**PROYECTO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO ELECTRÓNICO EN TELECOMUNICACIONES**

AUTOR: ARMIJOS SANTAMARÍA, MARCO ANDRÉS

**TEMA: “DESARROLLO DE UN CODIFICADOR MPEG-4 AAC/TS
UTILIZANDO MATLAB”**

DIRECTOR: DR. OLMEDO, GONZALO

CODIRECTOR: ING. ACOSTA, FREDDY

SANGOLQUÍ, JULIO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICADO

Dr. Olmedo, Gonzalo

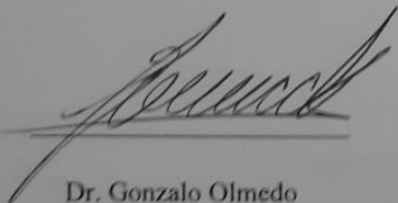
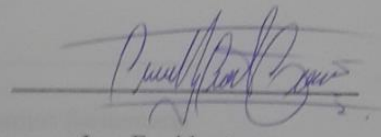
Ing. Acosta, Freddy

CERTIFICAN

Que el trabajo titulado “DESARROLLO DE UN CODIFICADOR MPEG-4 AAC/TS UTILIZANDO MATLAB”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el reglamento de estudiantes de la Universidad de las Fuerzas Armadas.

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

Sangolquí, 10 de julio de 2015.

	
Dr. Gonzalo Olmedo	Ing. Freddy Acosta
DIRECTOR	CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

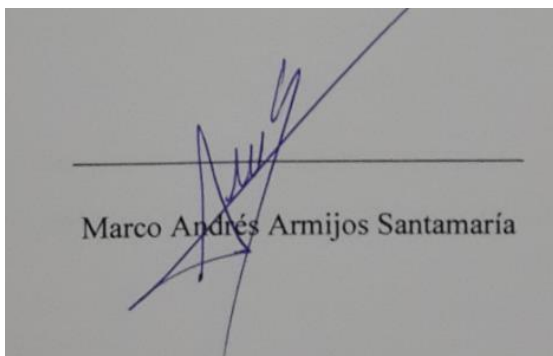
MARCO ANDRÉS ARMIJOS SANTAMARÍA

DECLARO QUE:

El proyecto de grado denominado “DESARROLLO DE UN CODIFICADOR MPEG-4 AAC/TS UTILIZANDO MATLAB”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas que constan en el documento, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi autoría.

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

Sangolquí, 10 de julio de 2015.



Marco Andrés Armijos Santamaría

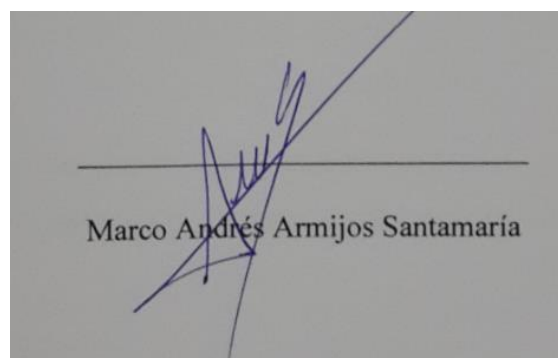
UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

MARCO ANDRÉS ARMIJOS SANTAMARÍA

Autorizo a la Universidad de las Fuerzas Armadas – ESPE, la publicación en la biblioteca virtual de la Institución del trabajo “DESARROLLO DE UN CODIFICADOR MPEG-4 AAC/TS UTILIZANDO MATLAB”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y auditoria.

Sangolquí, 10 de julio de 2015.



Marco Andrés Armijos Santamaría

DEDICATORIA

A mis padres Freddy y Paulina, por el gran esfuerzo que han realizado en el transcurso de estos años para poder cumplir esta importante meta académica; por sus enseñanzas, principios y valores que me han permitido seguir adelante pese a todas las dificultades presentadas y por su gran motivación para alcanzar mis sueños y metas confiando principalmente en Dios.

A mi familia, por su cariño y confianza, para el logro de esta importante meta.

AGRADECIMIENTO

A Dios, por darme la vida y la oportunidad de poder superarme y alcanzar las metas que me he propuesto, por estar incondicionalmente en momentos difíciles y entregarme la fe necesaria para salir adelante tras cada prueba y tropiezo.

A mi padre, por su esfuerzo, confianza y apoyo que me fueron de gran ayuda para iniciar y culminar esta importante meta académica.

A mi madre, por su gran amor y preocupación para poder cumplir con esta importante meta, por sus sabios consejos y por estar conmigo en cada momento importante de mi vida.

A mi familia, especialmente a mis abuelitos Enrique† y María, que desde pequeño me enseñaron el valor del estudio y de la superación, y estuvieron siempre brindándome su apoyo y cariño incondicional.

A mi amiga Dianita, que sin su ayuda, paciencia y palabras de ánimo en momentos difíciles no hubiese sido posible realizar este gran sueño.

A mis amigos y compañeros que forman parte de mi vida, por todos los momentos vividos y su apoyo incondicional.

A mis tutores, que me han entregado su tiempo y la paciencia suficiente para poder guiarme hacia la culminación de este proyecto.

ÍNDICE DE CONTENIDO

CERTIFICADO	i
DECLARACIÓN DE RESPONSABILIDAD.....	ii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDO.....	vi
RESUMEN.....	xv
ABSTRACT.....	xvi
CAPÍTULO I.....	1
PRESENTACIÓN.....	1
1.1. Antecedentes	1
1.2. Justificación.....	2
1.3. Alcance del proyecto	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	4
CAPÍTULO II	5
INTRODUCCIÓN A LA TELEVISIÓN DIGITAL TERRESTRE.....	5
2.2. Introducción.....	5
2.2.1. La Televisión Digital Terrestre (TDT).....	5
2.2.2. La Transmisión Digital de Servicios Integrados (ISDB)	5
2.2.3. La Transmisión Digital Terrestre de Servicios Integrados (ISDB-T)....	7

2.2.3.1.	Características técnicas de la ISDB-T	7
2.2.3.2.	La ISDB-T Internacional	8
2.3.	Sistema brasileño de Transmisión Digital de Servicios Integrados (SBTVD/ ISDB-Tb)	8
2.3.1.	Introducción	8
2.3.2.	Características técnicas	9
2.3.3.	Sistema de difusión de alertas	10
CAPÍTULO III.....		11
LA COMPRESIÓN DIGITAL DE AUDIO		11
3.2.	La Compresión digital	11
3.2.1.	Introducción	11
3.2.2.	Algoritmos de compresión	14
3.2.2.1.	Algoritmos de compresión sin pérdidas	14
3.2.2.2.	Algoritmos de compresión con pérdidas	15
3.3.	La compresión digital de audio	16
3.3.1.	Introducción	16
3.3.2.	Métodos de codificación	17
3.3.2.1.	Método de codificación perceptual.....	18
3.3.2.2.	Método de codificación paramétrica.....	19
3.3.2.3.	Método de codificación de forma de onda.....	20
3.3.2.4.	Método de codificación híbrida	21
3.3.3.	Parámetros de codificación de audio	21
3.4.	Características de los principales codificadores de audio	23
3.5.	El codificador MPEG	24
3.5.1.	Introducción	24

3.5.2.	Principio de funcionamiento básico	25
3.5.3.	El decodificador MPEG	27
3.5.4.	Perspectiva general de procesos de codificación de audio MPEG	28
3.5.4.1.	Códigos de Huffman	28
3.5.4.2.	Compresión de audio	28
3.5.4.3.	Función ventana	33
3.5.4.4.	Transformada Discreta del Coseno Modificada (MDCT)	34
3.6.	Formatos de archivo de sonido digital.....	35
3.6.1.	Formateo del <i>bitstream</i>	36
3.6.2.	Encabezamiento de la trama.....	37
CAPÍTULO IV		38
ADVANCED AUDIO CODING (AAC).....		38
4.2.	Generalidades	38
4.2.1.	Introducción	38
4.2.2.	Estandarización	39
4.2.3.	Ventajas de AAC con respecto a MP3	40
4.3.	Funcionalidad de AAC	42
4.3.1.	Codificación modular	44
4.3.2.	Corrección de errores	45
4.3.3.	Error Resiliente (ER).....	46
4.3.4.	Perfil de bajo retardo	46
4.4.	Licencias y patentes.....	46
4.5.	Extensiones y mejoras	47
4.6.	Formatos contenedores	47

4.7.	Productos que definen el formato AAC	498
4.7.1.	Estándares HDTV	498
4.7.1.1.	ISDB-T Japonés	498
4.7.1.2.	ISDB-Tb Internacional	49
4.7.1.3.	Digital Video Broadcasting (DVB)	49
4.7.2.	Hardware	49
4.7.3.	Software	49
4.8.	MPEG AAC	50
4.8.1.	Introducción	50
4.8.2.	Generalidades	50
4.8.3.	Control de Ganancia.....	56
4.8.4.	Banco de filtros	58
4.8.4.1.	Resolución del banco de filtros y diseño de la función ventana... 59	
4.8.5.	Predicción.....	62
4.8.6.	Cuantificación y codificación	63
4.8.7.	Codificación sin ruido (<i>Noiseless coding</i>)	69
4.8.8.	Multiplexación del <i>bitstream</i>	73
4.8.9.	Audio Data Transport Stream (ADTS)	74
CAPÍTULO V.....		76
DISEÑO Y ESPECIFICACIONES DEL CODIFICADOR.....		76
5.2.	Introducción.....	76
5.2.1.	Propósito	76
5.2.2.	Ámbito del sistema.....	76
5.2.3.	Definiciones, acrónimos y abreviaturas	77

5.3.	Descripción General	78
5.3.1.	Funciones del producto	78
5.3.1.1.	Función de reconocimiento de la señal de audio de entrada.....	78
5.3.1.2.	Función de procesamiento de la señal de entrada.....	78
5.3.1.3.	Función de transformación de dominios.....	78
5.3.1.4.	Función de obtención de los coeficientes MDCT y factores de escala.....	79
5.3.1.5.	Función de codificación.....	79
5.3.1.6.	Función de formación del <i>bitstream</i>	79
5.3.1.7.	Función de transformación de formato.....	79
5.3.2.	Perspectiva del producto	80
5.3.3.	Características de los usuarios	80
5.3.4.	Restricciones	81
5.3.5.	Requisitos futuros	81
5.4.	Requisitos específicos	82
5.4.1.	Interfaces	82
5.4.1.1.	Interfaz de usuario	82
5.4.1.2.	Interfaces de hardware	83
5.4.1.3.	Interfaces de software	84
5.5.	Diseño del codificador.....	84
5.5.1.	Señal de entrada del codificador	84
5.5.2.	Tratamiento de la señal de entrada.....	85
5.5.3.	Transformación de dominios y descomposición espectral.....	86
5.5.4.	Cálculo de coeficientes de la MDCT, factores de escala, bandas del factor de escala y proceso de cuantificación.....	87
5.5.5.	Proceso de codificación.....	87
5.5.6.	Formación del <i>bitstream</i>	88

5.5.7. Conversión de formato de audio	91
5.6. Diagramas.....	92
5.6.1. Diagrama general del codificador	92
5.6.2. Diagrama de flujo del codificador.....	93
CAPÍTULO VI.....	94
PRUEBAS Y RESULTADOS	94
6.2. Introducción.....	94
6.3. Generación de archivo de audio AAC.....	94
6.4. Obtención de parámetros gráficos del proceso de codificación	95
6.5. Análisis del formato contenedor ADTS	97
6.6. Análisis de la trama AAC.....	97
6.7. Generación de archivo de audio TS.....	102
CAPÍTULO VII.....	104
CONCLUSIONES Y RECOMENDACIONES.....	104
7.2. Conclusiones	104
7.3. Recomendaciones	106
BIBLIOGRAFÍA.....	108

ÍNDICE DE FIGURAS

Figura 2.1. Estructura general del sistema de transmisión digital ISDB	6
Figura 3.1. Definición de entropía y redundancia.....	12
Figura 3.2. Diagrama general del proceso de compresión	14
Figura 3.3. Diagrama general del proceso de codificación perceptual	18
Figura 3.4. Diagrama general del proceso de codificación paramétrica.....	20
Figura 3.5. Diagrama de bloques del codificador MPEG	25
Figura 3.6. Diagrama de bloques del decodificador MPEG	27
Figura 3.7. La compensación digital.....	29
Figura 3.8. Codificación en punto flotante	30
Figura 3.9. Ejemplo de un sistema de codificación en punto flotante con mantisa de 8 <i>bits</i> y 3 <i>bits</i> de exponente.....	31
Figura 3.10. Transitorio al final de un bloque de una transformada.....	32
Figura 3.11. Aplicación de funciones matemáticas tipo ventana para la superposición de segmentos.....	33
Figura 4.1. Diagrama del modelo de referencia del codificador MPEG AAC	53
Figura 4.2. Bloque de control de ganancia para el codificador MPEG AAC	57
Figura 4.3. Proceso de conmutación de forma de ventana para el codificador MPEG AAC	60
Figura 4.4. Proceso de conmutación de bloques para MPEG AAC	61
Figura 4.5. Número de coeficientes MDCT en cada SFB en bloques largos con frecuencia de muestreo de 48 kHz.....	65
Figura 4.6. Diagrama de bloques de un lazo de iteración interno para el codificador MPEG AAC	66
Figura 4.7. Diagrama de bloques de un lazo de iteración externo para el codificador MPEG AAC	68
Figura 5.1. Tratamiento de bloques durante el proceso de codificación.....	83
Figura 5.2. Diagrama de flujo para la señal de entrada del codificador.....	85
Figura 5.3. Formación matricial de bloques largos de dimensión $N \times 1024$	86
Figura 5.4. Descomposición espectral de la señal de un bloque de muestra mediante el uso de la función ventana y la MDCT	86

Figura 5.5. Estructura de cabeceras ADTS para las versiones MPEG-4 y MPEG-2 en hexadecimal	89
Figura 5.6. Comando del programa FFMPEG empleado para la conversión de audio del formato AAC a TS.....	91
Figura 5.7. Comando de MATLAB para ejecución de archivos BATCH.....	91
Figura 5.8. Diagrama general del codificador diseñado	92
Figura 5.9. Diagrama de flujo del codificador diseñado.....	93
Figura 6.1. Señal de audio de entrada original y modificada.....	95
Figura 6.2. Muestra de bloque de la señal de entrada en el cálculo de la MDCT.....	96
Figura 6.3. Análisis de coeficientes de la MDCT para un bloque de muestra.....	96
Figura 6.4. Detalles de cabecera ADTS según el analizador P23.....	97
Figura 6.5. Ventana principal del analizador AAC Audio ES Viewer con el análisis general del archivo de audio obtenido.....	98
Figura 6.6. Verificación de parámetros del archivo de audio obtenido según el analizador AAC Audio ES Viewer.....	99
Figura 6.7. Verificación y análisis de la cabecera ADTS del archivo de audio obtenido según el analizador AAC Audio ES Viewer	100
Figura 6.8. Verificación de parámetros del archivo de audio obtenido en representación hexadecimal según el analizador AAC Audio ES Viewer.....	101
Figura 6.9. Representación gráfica de bloques de la trama AAC para el archivo de audio codificado	102
Figura 6.10. Características del audio TS obtenido en la salida del codificador dadas por el programa SMPLAYER.....	103

ÍNDICE DE TABLAS

Tabla 2.1. Características técnicas de la ISDB-Tb.....	9
Tabla 3.1. Principales métodos de compresión con pérdida	16
Tabla 3.2. Principales frecuencias de muestreo de audio.....	22
Tabla 3.3. Características de los principales codificadores de audio.....	24
Tabla 3.4. Principales formatos de sonido digital	35
Tabla 3.5. Archivos de audio comprimido y sus estándares	36
Tabla 4.1. Formatos contenedores de AAC	48
Tabla 4.2. Frecuencias de muestreo y <i>bitrate</i> por canal para AAC	55
Tabla 4.3. Libro de códigos Huffman para MPEG AAC.....	72
Tabla 4.4. Estructura general del <i>bitstream</i> MPEG AAC.....	73
Tabla 4.5. Estructura de bits del formato ADTS.....	75
Tabla 5.1. Características del usuario “Usuario”	80
Tabla 5.2. Características del usuario “Desarrollador de Software”	80
Tabla 5.3. Libro de códigos Huffman del codificador diseñado.....	88
Tabla 5.4. Estructura del <i>bitstream</i> para una trama AAC en formato ADTS	89
Tabla 6.1. Características del audio de entrada.....	94
Tabla 6.2. Características del audio codificado	94
Tabla 6.3. Características del audio de salida TS.....	103

RESUMEN

El presente proyecto consiste en el diseño de un codificador de audio basado en las normativas dadas por el estándar ISO/IEC 13818-7, conocido como AAC (*Advanced Audio Coding*) y salida TS (*Transport Stream*) en el lenguaje de programación de MATLAB. El mismo tiene como base un diseño que se acopla a las características de los estándares de Televisión Digital Terrestre como es el caso de la ISDB-Tb, cuyas especificaciones técnicas admiten al formato AAC como formato de codificación de audio y al formato TS como modelo de paquete para el proceso de transmisión de contenidos. El trabajo realizado consta de dos partes principales. En la primera parte se realiza el proceso de compresión y codificación de audio, donde se emplea el perfil de baja complejidad de AAC (AAC-LC), según el modelo de referencia dado por el estándar. En esta parte se verifican varios procesos comunes de compresión y codificación de audio de los estándares MPEG, como es el caso de la MDCT, la función ventana y el proceso de codificación de Huffman. En la segunda parte, se procede con la conversión del formato de audio obtenido tras el proceso de codificación, mediante el uso de programas externos como FFMPEG que se asocian al lenguaje de programación diseñado, con el fin de obtener el archivo de salida en formato TS.

PALABRAS CLAVE:

- **ADVANCED AUDIO CODING**
- **TRANSPORT STREAM**
- **TRANSFORMADA DISCRETA DEL COSENO MODIFICADA**
- **FFMPEG**
- **CODIFICADOR MPEG**

ABSTRACT

This project involves the design of an audio encoder based in regulations given by the standard ISO / IEC 13818-7, known as Advanced Audio Coding and output Transport Stream in the programming language of MATLAB. The same has a design that engages with features of Digital Terrestrial Television standards such as the case of the ISDB-Tb, whose technical specifications allow to AAC to be audio encoding format and the TS format to be model process for package contents transmission. The project has two main parts. The first part has the process of compression and coding of audio, where the profile of low complexity (AAC-LC) and the reference model of the standard are used. In this part, several common processes of compression and coding of audio MPEG standards are verified, as is the case of MDCT, the window function and Huffman coding. In the second part, the encoded audio is processed using FFMPEG as external program. The same is associated with the programming language designed, with the objective of get the output file in format TS.

KEYWORDS:

- **ADVANCED AUDIO CODING**
- **TRANSPORT STREAM**
- **MODIFIED DISCRETE COSINE TRANSFORM**
- **FFMPEG**
- **MPEG ENCODER**

GLOSARIO

A

- AAC** *Advanced Audio Coding*. Tipo de formato informático de señal digital de audio basado en un algoritmo de compresión con pérdida.
- AAC-LC** *Advanced Audio Coding Low Complexity*. Perfil de baja complejidad de AAC.
- AAC-SSR** *Advanced Audio Coding Sampling Rate Profile*. Tipo de perfil de AAC con facilidad de conmutar los bloques pequeños a la salida del filtro.
- AAC-HE** *Advanced Audio Coding High Efficiency*. Tipo de perfil de AAC que emplea replicación de banda espectral y estéreo paramétrico.
- AC-3** *Dolby Digital AC3*. Tecnología de compresión de audio de laboratorios Dolby.
- ADIF** *Audio Data Interchange Format*. Formato contenedor de AAC para el almacenamiento del audio codificado.
- ADPCM** *Adaptive Differential Pulse Code Modulation*. Codificador de forma de onda basado en DPCM que añade mayores funcionalidades.
- ADTS** *Audio Data Transport Stream*. Formato contenedor de la trama AAC que sirve para el transporte del audio codificado.
- AIF** *Audio Interchange File Format*. Formato contenedor usado en tramas de audio que sirve para el almacenamiento del audio codificado.
- ATSC** *Advanced Television Systems Committee*. Grupo encargado del desarrollo de los estándares de Televisión Digital en Estados Unidos.

ATSC-M/H *Advanced Television Systems Committee - Mobile / Handheld o A/153.* Grupo encargado del desarrollo de los estándares de Televisión Digital para dispositivos móviles y portátiles.

B

BAT Archivo de procesamiento por lotes del DOS de Windows.

BD-ROM *Blu-ray Disk ROM.* Formato de disco óptico de nueva generación para video HD.

BSAC *Bit Sliced Arimethic Codification.* Estándar de MPEG-4 para la codificación escalable de audio.

BST OFDM *Band Segmented Transmission Orthogonal Frequency Division Multiplexing.* Técnica empleada por la ISDB-T que divide la banda de frecuencia de un canal en trece segmentos.

C

CBR *Constant Bit Rate.* Característica del proceso de codificación que implica que la tasa de salida del codificador de los datos es constante.

CELP *Code Excited Linear Prediction.* Algoritmo de codificación de señales de voz.

CQF *Conjugate Quadrature Filters.* Bancos de filtros para el tratamiento espectral de señales.

CRC *Cyclic Redundancy Check.* Código de detección de errores de uso común para detectar cambios accidentales en los datos en bruto.

D

- DAT** *Digital Audio Tape.* Medio de grabación y reproducción de señales de audio.
- DCT** *Discrete Cosine Transform.* Transformada basada en la Transformada Discreta de Fourier que utiliza únicamente números reales.
- DFT** *Discrete Fourier Transform.* Transformada que obtiene una representación en el dominio de la frecuencia de una señal en el dominio del tiempo.
- DPCM** *Differential Pulse Code Modulation.* Codificador de forma de onda que se basa en PCM pero añade funcionalidades basadas en la predicción.
- DTS** *Digital Theater System.* Sistema digital de sonido que permite la existencia de 6 canales de audio en una sola señal comprimida.
- DTTB** *Digital Terrestrial Television Broadcasting.* Conjunto de sistemas para la difusión de Televisión Digital Terrestre.
- DVB** *Digital Video Broadcasting.* Organismo encargado de crear y proponer procedimientos de estandarización para la Televisión Digital compatible.
- DVB-H** *Digital Video Broadcasting Handheld.* Organismo encargado de crear y proponer procedimientos de estandarización para la Televisión Digital en dispositivos portátiles.
- DVD** *Digital Versatile Disc.* Disco óptico para el almacenamiento de datos.

E

ER *Error Resiliente*. Tipo de error obtenido en algunos procesos de codificación de audio y video.

ETSI *European Telecommunications Standards Institute*. Organización de estandarización independiente, sin fines de lucro de la industria de las telecomunicaciones.

F

FAAC *Freeware Advanced Audio Coder*. Conjunto de herramientas de libre distribución para la codificación de audio en AAC.

FAAD *Freeware Advanced Audio Decoder*. Conjunto de herramientas de libre distribución para la decodificación de audio en AAC.

FFMPEG Colección de *software* libre que puede grabar, convertir y hacer streaming de audio y vídeo.

FLAC *Free Lossless Audio Codec*. Codificador de audio que permite que el audio digital sea comprimido sin pérdidas.

H

HCR *Huffman Codeword Reordering*. Técnica que permite dar ciertos niveles de protección a las palabras código en el proceso de codificación.

HD *High Definition*. Sistema de vídeo o audio con una mayor resolución que la definición estándar.

HVXC *Harmonic Vector Excitation Coding*. Algoritmo de codificación para voz.

I

IDE *Integrated Development Environment.* Aplicación que proporciona servicios integrales para facilitarle al programador el desarrollo de *software*.

IEC *International Electrotechnical Commission.* Organización de normalización en los campos eléctricos, electrónicos y tecnologías relacionadas.

IMDCT *Inverse Modified Discrete Cosine Transform.* Función inversa de la Transformada Discreta del Coseno Modificada.

IPTV *IP Television.* Solución para la transmisión de Televisión Digital sobre redes de datos.

ISDB *Integrated Services Digital Broadcasting.* Norma para la difusión de audio y video digital.

ISDB-Tb *Integrated Services Digital Broadcasting Terrestrial (International).* Norma para la difusión de audio y video digital terrestre.

ISO *International Organization for Standardization.* Organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación.

L

LATM *Low Overhead Audio Transport Multiplex.* Tipo de formato contenedor de trama AAC que contiene varios contenedores tipo LOAS.

- LFE** *Low Frequency Effects*. Tipo de técnica usada para la descripción de una pista de audio contenida dentro de los efectos de sonido de una película.
- LOAS** *Low Overhead Audio Stream*. Tipo de formato contenedor de trama AAC que sirve para la formación de LATM.
- LTP** *Long Term Predictor*. Técnica de codificación que mejora la predicción con una menor complejidad computacional.
- M**
- MDCT** *Modified Discrete Cosine Transform*. Transformada lineal ortogonal, basada en la idea de la cancelación del *aliasing* del dominio del tiempo.
- MID** *Musical Instrument Digital*. Protocolo de comunicación serial que permite a los dispositivos comunicarse y compartir información para la generación de sonidos.
- MNR** *Mask to Noise Ratio*. Razón para el enmascaramiento de ruido durante el proceso de compresión.
- MP3** *MPEG-1 Audio Layer III o MPEG-2 Audio Layer III*. Formato de compresión de audio digital que usa un algoritmo de compresión con pérdida.
- MPA** *MPEG-1 Audio Layer I or MPEG-2 Audio Layer I*. Grupo de estándares de codificación de audio y vídeo normalizados por el grupo MPEG.
- MPEG** *Moving Picture Experts Group*. Grupo de trabajo de expertos formado por la ISO e IEC para establecer estándares de audio y video.

O

OFDM *Orthogonal Frequency Division Multiplexing*. Tipo de multiplexación que consiste en enviar un conjunto de ondas portadoras de diferentes frecuencias, donde cada una transporta información.

P

PAC *Perceptual Audio Coder*. Tipo de algoritmo utilizado para comprimir audio digital eliminando la información que no es percibida por el oído humano.

PCM *Pulse Code Modulation*. Procedimiento de modulación utilizado para transformar una señal analógica en una señal digital.

PES *Packetized Elementary Stream*. Especificación que define los flujos elementales, que es normalmente la salida de un codificador de audio o vídeo, en paquetes dentro de un flujo de transporte o programa MPEG.

PNS *Perceptual Noise Shaping*. Técnica para el tratamiento del ruido durante el proceso de compresión.

PQMF *Pseudo Quadrature Mirror Filter*. Banco de filtros para el tratamiento de señales en el que se produce la cancelación del *aliasing* entre bandas adyacentes.

Q

QMF *Quadrature Mirror Filter*. Banco de filtros que divide la señal de entrada en dos bandas que posteriormente suelen ser submuestreadas por un factor de 2.

R

RM *Real Media*. Formato contenedor de audio.

S

SACD *Super Audio CD*. Tipo de disco óptico desarrollado para suministrar audio de alta definición.

SBR *Spectral Band Replication*. Técnica empleada para el desarrollo y mejora de codificadores de audio a bajas tasas de bits.

SMR *Signal to Mask Ratio*. La relación de señal a máscara, es la medida de la potencia de la señal respecto a la del nivel de la máscara utilizada en la codificación sub-banda.

SNR *Signal to Noise Ratio*. Relación existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe.

SQNR *Signal to Quantization Noise Ratio*. Relación existente de la señal transmitida con el ruido de cuantificación.

T

TDT *Terrestrial Digital Television*. Transmisión de imágenes en movimiento y su sonido asociado mediante codificación binaria a través de una red de repetidores terrestres.

TMCC *Transmission and Multiplexing Control Configuration*. Configuración realizada dentro de las normas de la ISDB-T para el proceso de control de transmisión y multiplexación de contenidos.

TNS *Temporal Noise Shaping*. Técnica usada para reformar el ruido de cuantificación en el dominio del tiempo.

- TS** *Transport Stream*. Formato de trama usada para la transmisión de contenidos en sistemas de Televisión Digital Terrestre.
- V**
- VBR** *Variable Bit Rate*. Método de compresión que consigue mayor calidad de sonido o video para un tamaño de archivo determinado, en contraste con CBR.
- VCB11** *Virtual Codebooks*. Libro de códigos virtual.
- VHDL** *Very High Speed Integrated Circuit Hardware Description Language*. Lenguaje de programación para la descripción de circuitos digitales.
- W**
- WAV** *Waveform Audio File Format*. Formato de audio digital normalmente sin compresión de datos.
- WMA** *Windows Media Audio*. Técnica de compresión de audio desarrollada por Microsoft.

CAPÍTULO I

PRESENTACIÓN

1.1. Antecedentes

El sistema brasileño de Televisión Digital Terrestre nace a partir del estándar ISDB-T (estándar japonés) con ciertas modificaciones en cuanto a la adopción de nuevas técnicas para la codificación de audio y video, el cual fue lanzado en una operación comercial el 2 de diciembre de 2007, en São Paulo, Brasil [1].

El 26 de marzo de 2010, la ex Superintendencia de Telecomunicaciones (SUPERTEL), presenta al ex Consejo Nacional de Telecomunicaciones (CONATEL), el informe para la definición e implementación de la Televisión Digital Terrestre en el Ecuador. Tras el análisis efectuado en diferentes aspectos (técnico, socio-económico y de cooperación internacional), el organismo recomienda la adopción del estándar ISDB-Tb/SBTVD (estándar japonés con variaciones brasileñas) [2].

Con la adopción del estándar ISDB-Tb en el Ecuador, han comenzado diversas investigaciones a nivel académico y profesional sobre dicho estándar que será implementado a futuro con el denominado “apagón analógico”.

Con dicho proceso, el Ecuador estará envuelto en un proceso de evolución, por tanto, es importante que las investigaciones en relación al nuevo equipamiento y la capacitación de profesionales en el tema estén involucrados en dicho proceso.

Además, con este proceso surge una oportunidad para la industria nacional la cual debe asumir el tema como una oportunidad para el desenvolvimiento industrial y tecnológico del país. Esa oportunidad es generada por una creciente demanda de varios factores, principalmente de equipamiento.

La industria nacional se puede favorecer en este tema, al ofrecer equipos de calidad con bajos precios, por tanto es necesario empezar con temas investigativos a

favor del desarrollo técnico en equipamiento de Televisión Digital Terrestre que cumplan con los protocolos del estándar ISDB-Tb para el Ecuador.

Existen varios trabajos de investigación que hacen referencia a la implementación de los procesos relacionados a la transmisión de Televisión Digital Terrestre bajo el estándar ISDB-Tb que sirven como guía y marcan un inicio en este tema de investigación.

1.2. Justificación

La importancia de realizar el presente proyecto de investigación radica en la elaboración de un código o algoritmo de programación en base a uno de los más importantes procesos definidos para la Televisión Digital Terrestre según las normas de la ISDB-Tb, como es la obtención del paquete *Transport Stream*. Dicho paquete es capaz de multiplexar video, audio y datos para la difusión de contenidos.

Para este fin se emplea una herramienta de programación en binario como es el caso de MATLAB, siendo una herramienta utilitaria para la implementación del código diseñado a niveles de *hardware*, empleando para aquellos dispositivos reprogramables como son las FPGA (*Field Programmable Gate Array*) [3].

Este proyecto de investigación se rige estrictamente a la obtención del paquete *Transport Stream* para una transmisión de audio de alta calidad empleando codificación AAC, la misma que ofrece avanzadas características de compresión multicanal, además de incluir características de protección de los derechos de autor dentro de su estructura, entre otras importantes características [4].

Una codificación basada en AAC presenta varias ventajas, como es la utilización de replicación de banda espectral, la cual se dedica a hacer una codificación en el dominio frecuencial, aprovechando las características de respuesta en frecuencia del sistema auditivo humano y el estéreo paramétrico que está diseñado para incrementar la eficiencia de codificación a bajas tasas de bits.

Una vez obtenido el algoritmo requerido, se habrá obtenido una parte del proceso del sistema de Televisión Digital Terrestre bajo las normas de la ISDB-Tb, con grandes proyecciones de que el mismo, acompañado por el *Transport Stream* de video, sean implementados a niveles de *hardware*, mediante las facilidades que presenta MATLAB para integrarse en las FPGA, con herramientas destinadas para este fin, como es el caso del *HDL Coder*.

Con esto se espera lograr el diseño de equipamiento de Televisión Digital Terrestre en el país, reduciendo de gran manera los costos que implica la importación de dichos equipos, además del beneficio de la industria tecnológica en nuestro país.

Otra ventaja de realizar el presente trabajo de investigación es poder obtener la trama *Transport Stream* con el objetivo de poder modificarla según diversas necesidades, como es el caso de añadir campos dentro de la trama que permitan enviar señales de alerta temprana a un determinado sector poblacional, entre otros aspectos de interés.

Con esto quedaría abierto un gran campo de investigación, donde se podrían implementar, de forma inicial, varios procesos que involucra la Televisión Digital Terrestre bajo el estándar ISDB-Tb, próximo a implementarse en nuestro país.

1.3. Alcance del proyecto

Obtener un algoritmo de programación en MATLAB el cual permita generar el paquete *Transport Stream* bajo las normas del estándar ISDB-Tb, empleando para aquello una codificación de audio MPEG-2/MPEG 4 AAC, con el objetivo de que dicho algoritmo pueda ser integrado a niveles de *hardware* en futuros proyectos de investigación empleando dispositivos reprogramables, además de poder manipular la trama obtenida según las necesidades requeridas, con fines de investigación y posibles mejoras sobre el estándar.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar un codificador de audio bajo las normas de la ISO/IEC 13818-7 denominado como MPEG-4 AAC (*Advanced Audio Coding*), con salida de audio en formato *Transport Stream*, que cumpla con los requerimientos de la ISDB-Tb.

1.4.2. Objetivos específicos

- Realizar el proceso de compresión y codificación de audio empleando técnicas de la familia de protocolos MPEG.
- Realizar diferentes pruebas sobre el archivo de audio codificado, mediante el uso de programas especializados, con el fin de evaluar que el mismo cumpla con los requerimientos establecidos.
- Analizar e investigar de forma general las características técnicas que posee el estándar ISDB-Tb, así como las diferentes etapas que se presentan durante el proceso de codificación y obtención de la trama final.
- Analizar factores críticos típicos de un proceso de codificación durante la elaboración del algoritmo tales como marcas de tiempo, códigos de redundancia cíclica, diseño de cabecera, verificación de tasas de bits constante y variables, entre otros aspectos.
- Realizar diferentes pruebas sobre la trama final obtenida, para verificar que la misma mantenga, dentro de su estructura, las características del audio original.
- Diseñar el algoritmo en lenguaje de programación m de MATLAB, con el fin que el mismo pueda ser traducido a un lenguaje de programación compatible con dispositivos reprogramables mediante el uso de herramientas propias del programa.

CAPÍTULO II

INTRODUCCIÓN A LA TELEVISIÓN DIGITAL TERRESTRE

2.2. Introducción

2.2.1. La Televisión Digital Terrestre (TDT)

Se define como un conjunto de tecnologías donde se incluye principalmente la transmisión y recepción terrestre de imágenes y sonido en tiempo real de forma digital.

La transmisión terrestre de forma digital presenta varias ventajas, entre las cuales se pueden destacar:

- Se aprovecha de gran manera el espectro radioeléctrico, ya que se ocupan mayores canales en el mismo ancho de banda usado comúnmente para un canal analógico, empleando diversas técnicas de compresión digital.
- Se tiene una mejor calidad de recepción de audio y video que una transmisión analógica, eliminando las interferencias. Además posibilita la transmisión de contenidos en alta definición.

2.2.2. La Transmisión Digital de Servicios Integrados (ISDB)

Se define como un conjunto de tecnologías de transmisión digital emergente creados por Japón para la transmisión de radio y televisión digital, con el objetivo de suministrar flexibilidad, capacidad de expansión y difusión de los servicios de transmisión multimedia.

Actualmente, la ISDB maneja tres tipos de sistemas:

- ISDB-S (Satélite).
- ISDB-T (Terrestre).

- ISDB-C (Cable).

La estructura de la ISDB está compuesta de tres bloques funcionales:

- Bloque de código fuente.
- Bloque multiplexor.
- Bloque de transmisión de código.

Dentro de esta estructura se consideran varios aspectos, como la configuración para el servicio de transmisión que involucra la recepción fija, móvil y portable.

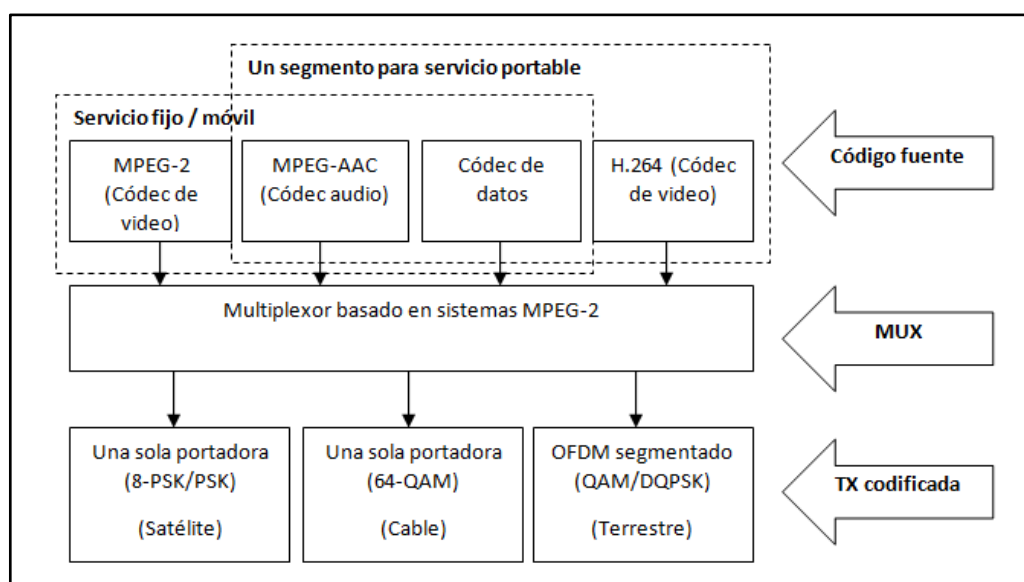


Figura 2.1. Estructura general del sistema de transmisión digital ISDB

En diferentes pruebas realizadas sobre la ISDB, se determinó que la ISDB-T ofrece características de recepción superior, razón por la cual la misma fue adoptada, inicialmente por Japón, como el sistema de transmisión terrestre de televisión y sonido digital.

2.2.3. La Transmisión Digital Terrestre de Servicios Integrados (ISDB-T)

Se define como un conjunto de conceptos de transmisión digital terrestre basado en el conjunto de normas dadas por la ISDB.

La ISDB-T ha sido probada en diferentes aspectos y se logró determinar que posee el mejor desempeño en la transmisión terrestre de Televisión Digital, siendo dichas pruebas realizadas en Japón durante más de 3 años [5].

2.2.3.1. Características técnicas de la ISDB-T

La ISDB-T toma en consideración la conformidad entre la transmisión digital terrestre de video y de sonido, pero además puede suministrar transmisión de datos que consisten en texto, diagramas, imágenes fijas e imágenes de video para dispositivos portátiles, así como imágenes de alta calidad y sonido estéreo.

Entre las características más importantes de la ISDB-T, se tienen las siguientes:

- Provee una variedad de servicios de video, sonido y datos.
- Tiene robustez ante cualquier interferencia multi-trayectoria y pérdida de intensidad encontrada durante la recepción portátil o móvil.
- Tiene receptores separados dedicados al vídeo, sonido y datos, así como receptores completamente integrados.
- Posee flexibilidad para acomodar diferentes configuraciones de servicios y asegurar flexibilidad en el uso de capacidad de transmisión.
- Puede abarcar un área suficientemente amplia para asegurar la satisfacción de requerimientos futuros.
- Puede acomodar redes de frecuencia única.
- Es capaz de usar frecuencias libres de manera efectiva.
- Es compatible con servicios análogos existentes y otros servicios digitales.

Las características definidas para la ISDB-T son posibles debido a una serie de herramientas tales como el sistema de modulación OFDM, asociado a técnicas de

segmentación de bandas, las cuales brindan un sistema flexible y con la posibilidad de una transmisión jerárquica, además de brindar la robustez necesaria contra ruidos impulsivos. Otra de las herramientas empleadas, es el Control de Configuración de Multiplexación y Transmisión (TMCC) que permite un cambio dinámico de los parámetros de transmisión, para poder realizar ajustes del sistema para un rendimiento optimizado, dependiendo del tipo de dispositivo (recepción en alta definición, recepción móvil, etc.).

2.2.3.2. La ISDB-T Internacional

En junio de 2006, Brasil adopta el estándar ISDB-T como sistema de transmisión terrestre. Actualmente, la ISDB-T es un sistema internacional.

La ISDB-Tb no es exactamente igual al sistema original, ya que el sistema brasileño posee una etapa de estandarización, donde adopta nuevas tecnologías, tales como la tecnología H.264 para la codificación de vídeo, así como tecnologías basadas en AAC para la codificación de audio, además de otras tecnologías para *middleware* [5]. Sin embargo, la estructura del sistema es la misma para ambos casos, al igual que el sistema de transmisión.

2.3. Sistema brasileño de Transmisión Digital de Servicios Integrados (SBTVD/ISDB-Tb)

2.3.1. Introducción

La ISDB-Tb es una variación del sistema japonés, la principal diferencia entre estos sistemas es la adopción de nuevas técnicas de compresión, como ejemplo, el sistema brasileño adopta la tecnología MPEG-4 (H.264) como técnica de compresión de video, y la familia de tecnologías AAC para la compresión de audio, entre otras técnicas experimentales.

2.3.2. Características técnicas

La ISDB-Tb es bastante flexible, siendo capaz de operar en diversas configuraciones, ya que hace uso de un código corrector de errores, además posibilita la utilización de entrelazadores (*interleaving*) en el dominio del tiempo y de la frecuencia y utiliza la técnica de modulación OFDM.

Además, el sistema hace uso de segmentación de banda de transmisión (BST OFDM) y posibilita la transmisión jerárquica de hasta tres capas, escogiendo para cada una de ellas los parámetros adecuados de tasa de transmisión y robustez deseada.

En la Tabla 2.1., se indican las principales características técnicas de la ISDB-Tb.

Tabla 2.1. Características técnicas de ISDB-Tb

Ancho de banda de canal	Generalmente de 6 MHz (es posible usar la norma con 13 segmentos en 7 MHz u 8 MHz)	
Acceso condicional	Multi – 2	
Middleware	Ginga-NCL y Ginga-J	
Multiplexación	Sistema MPEG-2 TS	
Codificación de audio	Fijo y móvil	Estéreo: MPEG-4 AAC@L2 o MPEG-4 HE-AAC v1@L2 Audio Multicanal 5.1: MPEG-4 AAC@L4 o MPEG-4 HE-AAC v1@L4
	Portátil	MPEG-4 HE-AAC v2@L2 (Únicamente estéreo)
	Fijo y móvil	MPEG-4 AVC (H.264) HP@L4
Codificación de video	Portátil	MPEG-4 AVC (H.264) BP@L1.3

Codificación de canal	Esquema de modulación	64 QAM-OFDM 16 QAM-OFDM QPSK-OFDM DQPSK-OFDM
	Código de corrección de errores	Codificación interna: Convolución de 7/8, 5/6, 3/4, 2/3, 1/2 Codificación externa: RS(204, 188)
	Intervalo de guarda	1/32, 1/16, 1/8, 1/4
	Intercalado	Por tiempo, frecuencia, bit y byte
	Tipo de modulación	BST-OFDM (OFDM Segmentado)
	Arquitectura de transmisión	- Para receptores fijos y móviles: 13 segmentos (Full HD). Otros arreglos también son posibles. - Para receptores portátiles: Servicio One-Seg

2.3.3. Sistema de difusión de alertas

Este sistema puede ser integrado dentro de las características de la ISDB-T e ISDB-Tb, el cual permite configurar la transmisión de avisos emergentes y enviar alertas a cada dispositivo en una determinada área de recepción, ante posibles eventos de tipo catastrófico.

La señal de alerta se encuentra integrada en uno de los segmentos de la secuencia de datos de la transmisión y enciende todos los receptores apagados, presentando de esta manera la información de alerta.

El *Forum* del sistema brasileño de Televisión Digital Terrestre cerró un acuerdo con el gobierno de Brasil para la adopción del estándar japonés del sistema de alertas en junio de 2011 [6].

CAPÍTULO III

LA COMPRESIÓN DIGITAL DE AUDIO

3.2. La Compresión digital

3.2.1. Introducción

La compresión digital es la reducción del volumen de datos digitales tratables para representar una determinada información, empleando una menor cantidad de espacio [7].

El espacio ocupado por una información codificada sin compresión es el cociente entre la frecuencia de muestreo y la resolución de la información, por lo que si se emplean más bits será mayor el tamaño del paquete.

Sin embargo, la resolución viene impuesta por el sistema digital con que se trabaja y no se puede alterar el número de bits a voluntad, por ello, se utiliza la compresión, para transmitir la misma cantidad de información que ocuparía una gran resolución en un menor número de bits.

Una de las formas de minimizar la cantidad de información en el proceso de compresión es utilizar predicciones estadísticas en las señales. La información contenida en una muestra (entropía) es una función de lo diferente que es ésta del valor predecible.

La entropía se define como el área ocupada por una señal y esta es el área que debe ser transmitida. El área remanente es llamada redundancia dado que ésta no se añade a la información transmitida, esto se puede apreciar en la Figura 3.1.

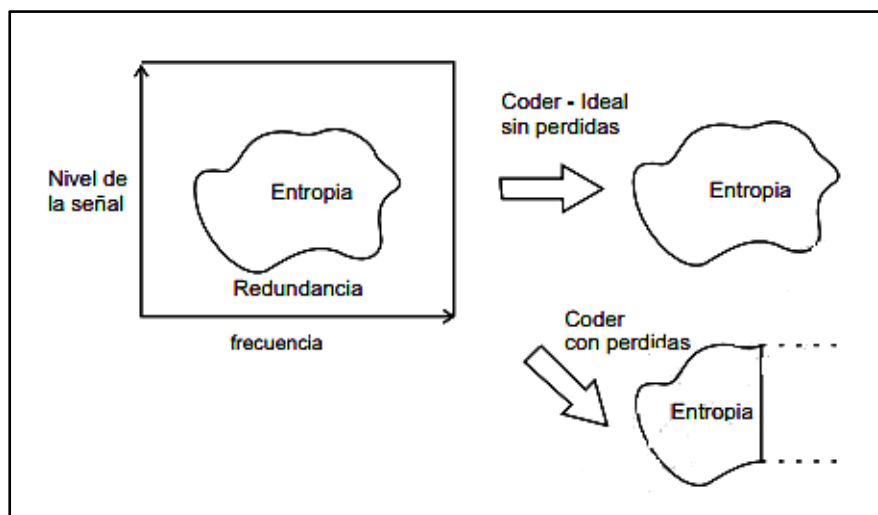


Figura 3.1. Definición de entropía y redundancia [27]

En la Figura 3.1. también se puede apreciar la definición de entropía y redundancia, además de la diferencia entre un compresor ideal y un compresor con pérdidas, los cuales se detallan a continuación:

- **Compresor ideal (sin pérdidas)**

Un compresor ideal extrae exclusivamente la forma de la entropía para ser transmitida, pudiendo obtenerse la información original, por lo que el compresor debería tener un alto grado de complejidad, además que debería poseer un canal que acepte cualquier tipo de entropía y si la capacidad del canal no fuese la suficiente, el compresor trataría parte de la entropía donde se podría eliminar información útil.

Para evitar este problema, el compresor modera el factor de compresión (*coding gain*) para que solamente se remueva la redundancia necesaria, sin originar elementos indeseables, lográndose un sistema sin pérdida (*subjectively lossless*).

- **Compresor con pérdidas**

Un compresor con pérdidas extrae una aproximación de la entropía para ser transmitida, esta aproximación aprovecha las limitaciones del sistema visual y auditivo humano para esconder la distorsión introducida.

Un compresor con pérdidas se emplea cuando la reconstrucción exacta no es indispensable para que la información tenga sentido. La información reconstruida es solo una aproximación de la información original.

La mayor ventaja de este compresor reside en las altas razones de compresión que ofrece en relación a un compresor sin pérdida.

El compresor con pérdidas acepta una pérdida de datos para poder mejorar el factor de compresión. Se aplica generalmente al almacenamiento y transmisión digital de datos analógicos como ficheros de gráficos y de sonidos.

El proceso de compresión permite reducir costos cuando se trata de transmitir señales. En este proceso el costo está relacionado con la velocidad de bit, por lo que al disminuir el flujo de datos se aminora la velocidad de bit (*bit rate*) y con esto los costos.

En la Figura 3.2. se puede observar de manera general el proceso de compresión, donde el flujo de datos es reducido en la fuente por un compresor.

Los datos comprimidos son enviados por un canal de comunicación hasta un expansor o *decoder*, donde los datos vuelven a su forma original.

La relación entre el flujo de datos de la fuente y el flujo de datos del canal es conocida como el factor de compresión. Además, esta relación está definida por el control de ganancia.

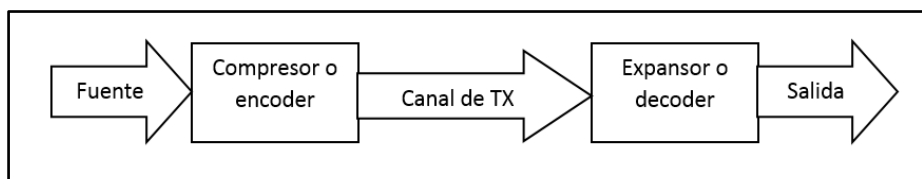


Figura 2.2. Diagrama general del proceso de compresión

3.2.2. Algoritmos de compresión

Existen dos tipos de algoritmos de compresión, sin pérdida y con pérdida.

3.2.2.1. Algoritmos de compresión sin pérdidas

Son aquellos que al reproducir los datos a su forma original mediante la descompresión lo hacen de manera exacta, es decir, se puede obtener una reconstrucción completa de los datos originales. Este proceso se lo realiza bit por bit, siendo el factor de compresión bajo (en el orden de 2 a 1).

Una de las técnicas comunes en un algoritmo de compresión sin pérdidas es la codificación en series, donde las series del mismo valor son comprimidas para transmitir las en un código previamente ordenado en líneas de unos o ceros seguidas por un número que indica la longitud de la línea.

En otra técnica de algoritmo de compresión sin pérdidas se utilizan códigos de longitud variable, aplicando análisis estadísticos en la frecuencia con la que son utilizadas determinadas letras. De esta manera, cuando se obtienen letras de uso frecuente se emplean códigos cortos mientras los códigos largos son utilizados para representar letras de poca frecuencia.

En este principio se basa el código Huffman, donde el mismo estudia la probabilidad con la que un código es transmitido y los códigos más repetidos son organizados para ser transmitidos mediante símbolos de longitud de palabra corta. A los códigos menos frecuentes se les permitirá longitudes de palabra largas.

3.2.2.2. Algoritmos de compresión con pérdidas

Son aquellos que tienen como objetivo representar una cierta cantidad de información utilizando una menor cantidad de la misma, pero sin lograr una reconstrucción exacta de los datos originales.

Los algoritmos de compresión con pérdidas usan técnicas de compresión sin pérdidas cuando es posible, pero su principal característica es la de obviar información. Para lograr esto, la información es procesada y dividida en dos grupos de datos, donde un grupo contiene idealmente toda la información importante y el otro toda la información que es irrelevante y únicamente los datos importantes necesitarán ser mantenidos para su respectiva transmisión.

Este tipo de algoritmo de compresión es muy utilizado en aplicaciones de audio y vídeo ya que permite grandes factores de compresión.

En los codificadores basados en un algoritmo de compresión con pérdida las diferencias son arregladas de tal manera que se hacen no detectables a los sentidos humanos de oído y visión, por tanto este tipo de codificadores se basan en comprender cómo trabaja la percepción psicoacústica y psicovisual del ser humano, por tal motivo se les conoce como codificadores de percepción.

Existen dos técnicas comunes para el proceso de compresión con pérdidas, las cuales se detallan a continuación:

- **Por códigos de transformación**

Donde los datos originales son transformados de tal forma que se simplifican (siendo imposible la reconstrucción original), creando un nuevo conjunto de datos propensos a elevadas razones de compresión sin pérdida.

- **Por códigos predictivos**

Donde los datos originales son analizados para predecir el comportamiento de los mismos. Después se compara esta predicción con la realidad, codificando el error y la información necesaria para la reconstrucción. El error es propenso a elevadas razones de compresión sin pérdida.

En la Tabla 3.1. se puede apreciar los principales métodos de compresión con pérdida.

Tabla 3.1. Principales métodos de compresión con pérdida

Compresión de imagen con pérdida	Compresión de video con pérdida	Compresión de audio con pérdida
Compresión Fractal	Flash	AAC
JPEG	H.261	ADPCM
Compresión Wavelet	H.263	ATRAC
PNG	H.264/MPEG-4 AVC	DTS
GIF	MNG	MP2
	Motion JPEG	MP3
	MPEG-1 / MPEG-2 / MPEG-4	WMA
	OGG	Opus
	MP4	Vorbis

3.3. La compresión digital de audio

3.3.1. Introducción

La compresión digital de audio se define como el proceso en el cual se reduce la tasa de bits de una señal digital de audio, buscando como fin reducir su tamaño para poder ser transmitido o almacenado.

La técnica más simple para reducir el volumen de datos en una señal digital es la reducción de su frecuencia de muestreo o la cantidad de bits de cuantificación, pero se debe considerar que al reducir la frecuencia de muestreo se reduce también al

ancho de banda que es posible procesar, y al reducir los bits de cuantificación se disminuye el SQNR (relación de señal a ruido de cuantificación), por lo que se introduce ruido en la señal.

Existen varios algoritmos de compresión de audio (llamados normalmente *códecs* de audio) cuyo funcionamiento se basan en técnicas de compresión con y sin pérdidas.

Uno de los métodos de compresión sin pérdidas es el DPCM, cuyo funcionamiento se basa en la predicción de las muestras de una señal.

Cabe indicar que las técnicas de compresión sin pérdidas son muy poco eficientes, ya que logran relaciones de compresión máximas de 4:1 aproximadamente.

De igual manera, una de las técnicas de compresión con pérdidas es el ADPCM, que divide la señal análoga en bandas de frecuencia mediante filtros QMF, donde cada banda es tratada de forma independiente mediante las características dadas por DPCM.

En la actualidad, existen numerosos laboratorios y equipos de investigación de todo el mundo para la búsqueda y desarrollo de algoritmos de compresión de audio cada vez más eficientes, con una mayor relación de compresión e independientes de la fuente original del sonido.

3.3.2. Métodos de codificación

En la actualidad, existe una gran variedad de técnicas de compresión de audio, los mismos que cada vez presentan mayor complejidad ya que añaden características adicionales. Se pueden clasificar en cinco grandes grupos en base al método empleado para la reducción de la tasa de bits.

3.3.2.1. Método de codificación perceptual

Este método implica contar con un modelo matemático del sistema de audición humano y es enteramente dependiente de la precisión del mismo, además del manejo preciso de técnicas de filtrado y de análisis del audio.

La técnica consiste básicamente en eliminar la parte de la señal de audio que el oído humano no es capaz de percibir debido a la presencia de otras señales que la enmascaran.

Para este proceso, se tiene como ejemplo una señal en formato PCM donde la misma pasa por un proceso de transformación al dominio frecuencial y se cuantifica y codifica a partir de un conjunto de datos obtenidos en un modelo psicoacústico, siendo este modelo el que determina la calidad final y es el mismo que aprovecha el enmascaramiento temporal o frecuencial, siendo el umbral de enmascaramiento el que controla la cuantificación. Este proceso puede ser apreciado en la Figura 3.3.

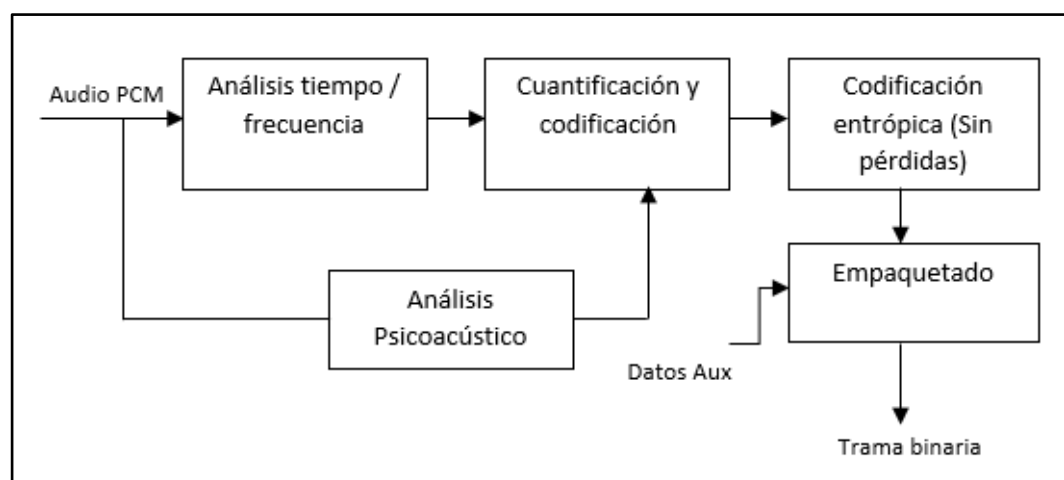


Figura 3.3. Diagrama general del proceso de codificación perceptual

Para la cuantificación de audio se emplean varias técnicas, donde generalmente se utiliza la cuantificación no uniforme para música, cuantificación diferencial o logarítmica para señales de voz y cuantificación vectorial para un procesamiento avanzado de la señal de voz.

El bloque que representa la codificación entrópica tiene el objetivo de realizar una compresión sin pérdidas para reducir el tamaño de la trama binaria. La trama binaria obtenida se empaqueta junto con información adicional que contiene datos imprescindibles para la decodificación como el tamaño de la trama, el número de bits de codificación, el margen dinámico, entre otros.

El método de codificación perceptual está basado en un algoritmo de compresión con pérdidas donde las distintas versiones de MPEG-1, siendo MP3 la más conocida, utilizan este método de codificación.

3.3.2.2. Método de codificación paramétrica

Este método de codificación se basa en que el audio puede ser sintetizado y representado con tonos aislados, patrones armónicos y componentes ruidosas, mismos que son representados en parámetros como la amplitud, la frecuencia fundamental o los componentes espectrales.

Para este método, de una señal analógica se extrae la información aplicando la transformada de Fourier, para luego realizar la estimación de parámetros. Posteriormente, los mismos se codifican basándose en modelos de percepción humana, y se multiplexan para formar la trama binaria. Este proceso puede ser apreciado en la Figura 3.4.

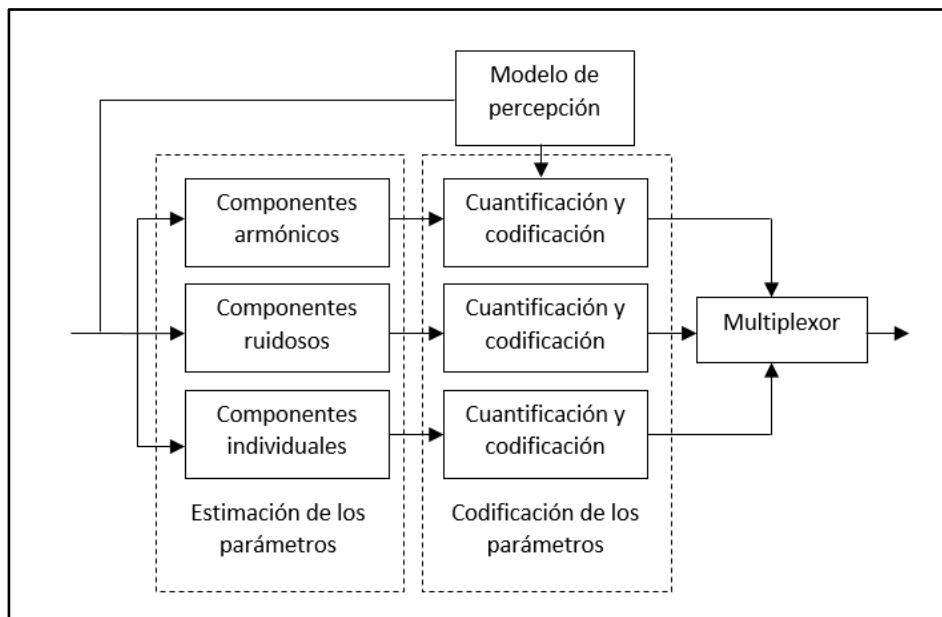


Figura 3.4. Diagrama general del proceso de codificación paramétrica

Un ejemplo de codificación paramétrica es el codificador HVXC.

Otro ejemplo de un codificador paramétrico son los llamados *vocoders*, mismos que analizan la señal de voz correspondiente a un segmento temporal, considerando estacionario para extraer los parámetros del modelo y la excitación, siendo esta información la que se codifica.

3.3.2.3. Método de codificación de forma de onda

Este método se basa en el estudio de la señal de audio, de forma que se intenta reproducir la forma de la señal de entrada, aprovechando la redundancia de la misma y, a partir de la una predicción lineal, se puede codificar la señal auditiva. De esta forma se consiguen tasas de compresión elevadas cuando las señales son redundantes y prácticamente nulas cuando no lo son.

Este proceso puede ser realizado en el dominio del tiempo como de la frecuencia.

Algunos ejemplos de este codificador en el dominio del tiempo son:

- Modulación por codificación de pulsos (PCM).
- Modulación por codificación de pulsos diferenciales (DPCM).
- Modulación por codificación de pulsos diferenciales adaptativos (ADPCM).

En el dominio de la frecuencia, se tiene como ejemplo las siguientes técnicas de codificación:

- Codificación en sub bandas.
- Codificación por transformada.

3.3.2.4. Método de codificación híbrida

Este método, también conocido como método de codificación de análisis por síntesis, combina las técnicas del método de codificación de forma de onda con las técnicas del método de codificación paramétrica, especialmente con los *vocoders*.

El objetivo del método es obtener voz de alta calidad a tasas de bit bajas (inferior a 8 kHz). Su funcionamiento se basa en analizar un conjunto de muestras para obtener los parámetros, a manera de análisis de una sola señal. Al decodificar la trama, se sintetizan los parámetros para conseguir que se parezca a la señal original.

3.3.3. Parámetros de codificación de audio

Existen varios parámetros a tener en consideración durante el proceso de codificación de audio. Los mismos se detallan a continuación:

- **Número de canales**

Este parámetro depende del número de señales simultáneas de audio que contiene el flujo de datos, puede ser 1 canal (mono), 2 canales (estéreo), 6 canales (multicanal 5.1), 8 canales (multicanal 7.1).

- **Frecuencia de muestreo**

Este parámetro determina la calidad del audio, por lo que si la frecuencia de muestreo es más alta, será más alta la fidelidad del sonido obtenido.

La máxima audiofrecuencia perceptible para el oído humano está alrededor de los 20 kHz, por lo se necesitaría una frecuencia de muestreo de 40000 (según el criterio de Nyquist), sin embargo se establece un estándar de 44100 muestras por segundo para poder compensar los filtros utilizados durante la conversión análoga – digital.

Las frecuencias de muestreo de audio más relevantes se aprecian en la Tabla 3.2.

Tabla 1.2. Principales frecuencias de muestreo de audio

Muestras por segundo	Aplicación
8000	Audio para teléfonos, cuyo muestreo es adecuado para la voz humana.
22050	Audio para radios.
32000	Video digital en formato mini DV.
44100	Reproducción de señales con componentes de hasta 20 kHz. Aplicado en audio en formatos MPEG (MP3, AAC).
47250	Formato PCM de Nippon Columbia (Denon). Permite reproducir señales con componentes de hasta 22 kHz.
48000	Audio en televisión digital, DVD, formato de películas, audio profesional y sistemas DAT.
192400	En sistemas HD DVD, audio de alta definición para DVD y BD-ROM.
2822400	Aplicado en sistemas SACD, Direct Stream Digital, desarrollado por Sony y Phillips.

- **Número de bits por muestra**

Este parámetro determina la precisión con la que se reproduce la señal original y el rango dinámico de la misma. Los valores más usados son 8 bits (para un rango dinámico de hasta 45 dB), 16 bits (para un rango dinámico de hasta 90 dB) y 24 bits (para un rango dinámico de hasta 120 dB).

- **Tipo de compresión**

Este parámetro indica si el proceso de compresión es con o sin pérdidas.

- **Tasa de bits**

Este parámetro determina el número de bits de información por unidad de tiempo, mismo que no se puede deducir de los parámetros anteriores ya que se puede aplicar una compresión con o sin pérdidas.

La tasa de bits empleada es variable (VBR) ya que la misma tiene mayor eficiencia en comparación con una tasa de bits constante (CBR) cuando hay silencios o segmentos donde la complejidad es baja y se puede almacenar o transmitir con un menor número de bits.

3.4. Características de los principales codificadores de audio

En la Tabla 3.3. se pueden apreciar las características más relevantes de los principales codificadores de audio.

Tabla 3.3. Características de los principales codificadores de audio [32]

Códec	AAC	FLAC	MP3	WAV	WMA
Tipo compresión	Con pérdidas, híbrido	Sin pérdidas	Con pérdidas	Con pérdidas, sin pérdidas, híbrido	Con pérdidas, sin pérdidas
Frecuencia de muestreo	De 8 kHz a 192 kHz	De 1 kHz a 1.04 MHz	De 8 a 48 kHz	De 1 Hz a 13.77 MHz	De 8 a 96 kHz
Tasa de bits	De 8 a 529 kilobit/s	Variable	De 8 a 320 kilobit/s	Variable (sin pérdidas), mayor a 196 kilobit/s (con pérdidas)	De 4 a 768 kilobit/s (con pérdidas), Variable (sin pérdidas)
Bits por muestra	Variable	4, 8, 16, 24, 32	Variable	Variable (sin pérdidas), mayor a 2.2 (con pérdidas)	16, 24 (sin pérdidas), variable (con pérdidas)
CBR	Si	No	Si	Si	Si
VBR	Si	Si	Si	Si	Si
Multicanal	Hasta 28 canales	Hasta 8 canales	No	Hasta 16 canales	Hasta 8 canales

3.5. El codificador MPEG

3.5.1. Introducción

Un codificador MPEG es un codificador basado en un método de codificación perceptual, es decir, su principio de funcionamiento se basa en determinados fenómenos psicoacústicos que ocurren en el oído humano.

De forma general, este tipo de codificadores operan descomponiendo la señal en bloques, correspondiendo cada uno a un cierto rango de frecuencia y tiempo determinados. Usando esta distribución tiempo-frecuencia, la señal es analizada de acuerdo con principios psicoacústicos, donde este análisis indica los bloques que deben ser codificados con máxima precisión, y así mismo indica los cuales pueden

tolerar cierto nivel de ruido de cuantificación sin degradar la calidad final del sonido percibido.

En base a la información obtenida de cada bloque, los *bits* disponibles (según el *bitrate* elegido) son distribuidos en los mismos. Los coeficientes espectrales en cada bloque son cuantificados utilizando los *bits* asignados.

3.5.2. Principio de funcionamiento básico

El codificador MPEG recibe como entrada las muestras de audio digital PCM y produce el *bitstream* comprimido para el respectivo almacenamiento o transmisión. Un diagrama generalizado del codificador puede ser apreciado en la Figura 3.5.

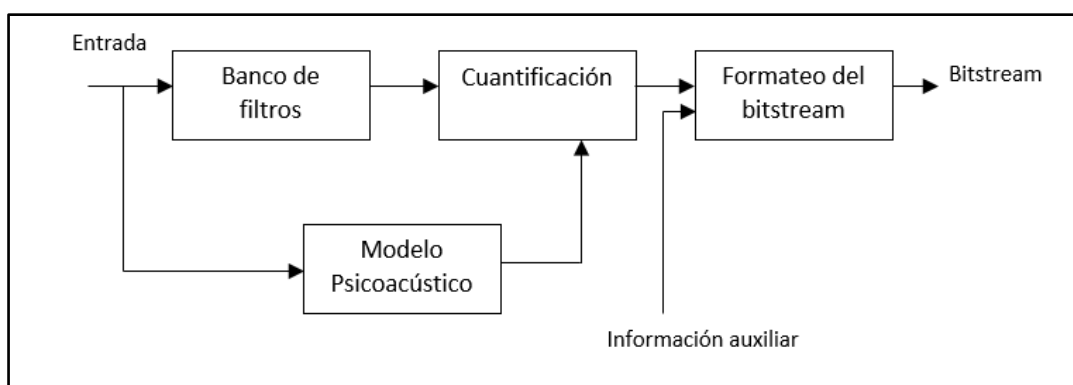


Figura 3.5. Diagrama de bloques del codificador MPEG [28]

- **Banco de filtros**

El banco de filtros está encargado de dividir la señal de audio en bandas de frecuencia con igual ancho de banda, este bloque es fundamental para el codificador MPEG.

De manera conceptual, el banco de filtros puede ser considerado como un conjunto de filtros pasabanda trabajando de forma paralela, donde todos reciben la misma señal de entrada y todos entregan una parte del espectro a la salida.

- **Modelo Psicoacústico**

Se define como un modelo matemático del comportamiento del oído humano basado en principios psicoacústicos de enmascaramiento frecuencial y temporal. Su función se basa en analizar la entrada de audio PCM y la salida de los filtros para determinar las bandas de frecuencia que deben ser mantenidas en la trama de salida y en qué medida deben ser conservadas. Es decir, algunas bandas de frecuencia van a ser enmascaradas por otras bandas vecinas y no será necesario que las mismas sean consideradas a la salida, pero otras bandas que no estarán enmascaradas si necesitarán ser procesadas.

La trama comprimida a la salida debe tener un *bitrate* específico, por este motivo las bandas de frecuencia a ser consideradas no pueden ser cuantificadas con un excesivo número de *bits*, deben ser cuantificadas con el número justo de *bits* de manera que el ruido de cuantificación que se genera en el proceso esté por debajo del umbral de audición humano.

En definitiva, este bloque analiza el audio original en base a principios psicoacústicos y determina el número de *bits* necesarios para cuantificar cada una de las bandas de frecuencia que entrega el banco de filtros, para que de esta manera, el ruido de cuantificación quede enmascarado por otras componentes de la señal de audio y el *bitrate* resultante a la salida no supere un determinado límite.

La salida definitiva del bloque (la cantidad de *bits* de cuantificación por banda) se expresa en términos de una relación denominada MNR (relación de máscara a ruido), mismo que se calcula como se indica en la Ecuación (1):

$$MNR(dB) = SNR(dB) - SMR(dB) \quad (1)$$

Dónde,

- MNR (dB): es la relación máscara a ruido.
- SNR(dB): es la relación señal a ruido.
- SMR(dB): es la relación señal a máscara (del modelo psicoacústico).

Cabe indicar que la norma no define ninguna implementación específica para el modelo psicoacústico, pero ofrece varias posibilidades de diseño de acuerdo a la cantidad de *bits* disponibles para cada una de las bandas.

3.5.3. El decodificador MPEG

El proceso de decodificación para MPEG es mucho más simple que el proceso de codificación, pero aun así ofrece un cierto grado de complicación matemática.

Para el proceso de decodificación, se emplea un modelo perceptual, donde el espectro cuantificado es reconstruido de acuerdo a los *bits* asignados y luego, sintetizado en una señal de audio compleja. El proceso puede ser apreciado en la Figura 3.6.

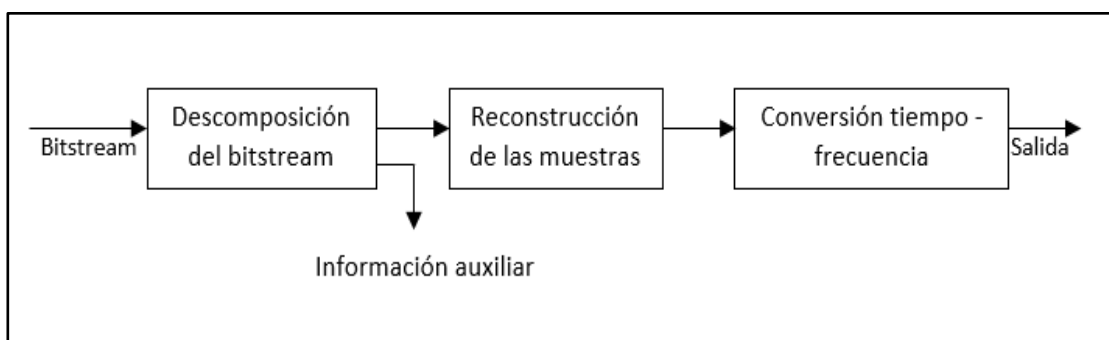


Figura 3.6. Diagrama de bloques del decodificador MPEG [28]

3.5.4. Perspectiva general de los procesos de codificación de audio MPEG

3.5.4.1. Códigos de Huffman

Se define como un código estadístico que se basa en la propiedad del prefijo, según la cual, ninguna secuencia de *bits* que represente a un carácter del código podrá aparecer como subsecuencia inicial de otra secuencia de longitud mayor que represente a otro carácter del código.

La construcción del código se basa en asignar a cada símbolo del alfabeto fuente una secuencia de *bits* cuya longitud está relacionada de forma directa con la probabilidad de aparición del símbolo, por tanto a los símbolos con mayor frecuencia de aparición serán asignados con palabras de código de menor longitud.

En el proceso de construcción del código, primeramente se ordena el conjunto de símbolos del alfabeto fuente en orden decreciente de probabilidades de aparición, luego se juntan los dos símbolos con menor probabilidad de aparición en un solo símbolo cuya probabilidad será la suma de las probabilidades de los símbolos que dieron origen a este nuevo símbolo.

El proceso se repite hasta que sólo se tengan dos símbolos, y posteriormente se realiza el proceso de codificación, donde se asigna un “1” a uno de los símbolos y un “0” al otro. Posteriormente, se recorre la estructura que se ha construido hacia atrás de forma que cuando dos símbolos hayan dado origen a un nuevo símbolo, estos dos símbolos heredarán la codificación asignada a este nuevo símbolo y luego se añade un “1” a la codificación de uno de los símbolos y un “0” a la del otro símbolo.

3.5.4.2. Compresión de audio

Existen varias técnicas para la compresión de audio, cada una con sus respectivas ventajas y desventajas, además considerando el costo y la complejidad de la misma.

El método de codificación más elemental es la compensación. Este método, de manera digital, es similar a los mecanismos de reducción de ruido en los grabadores

analógicos de cintas. La técnica de la compensación se puede apreciar en la Figura 3.7.

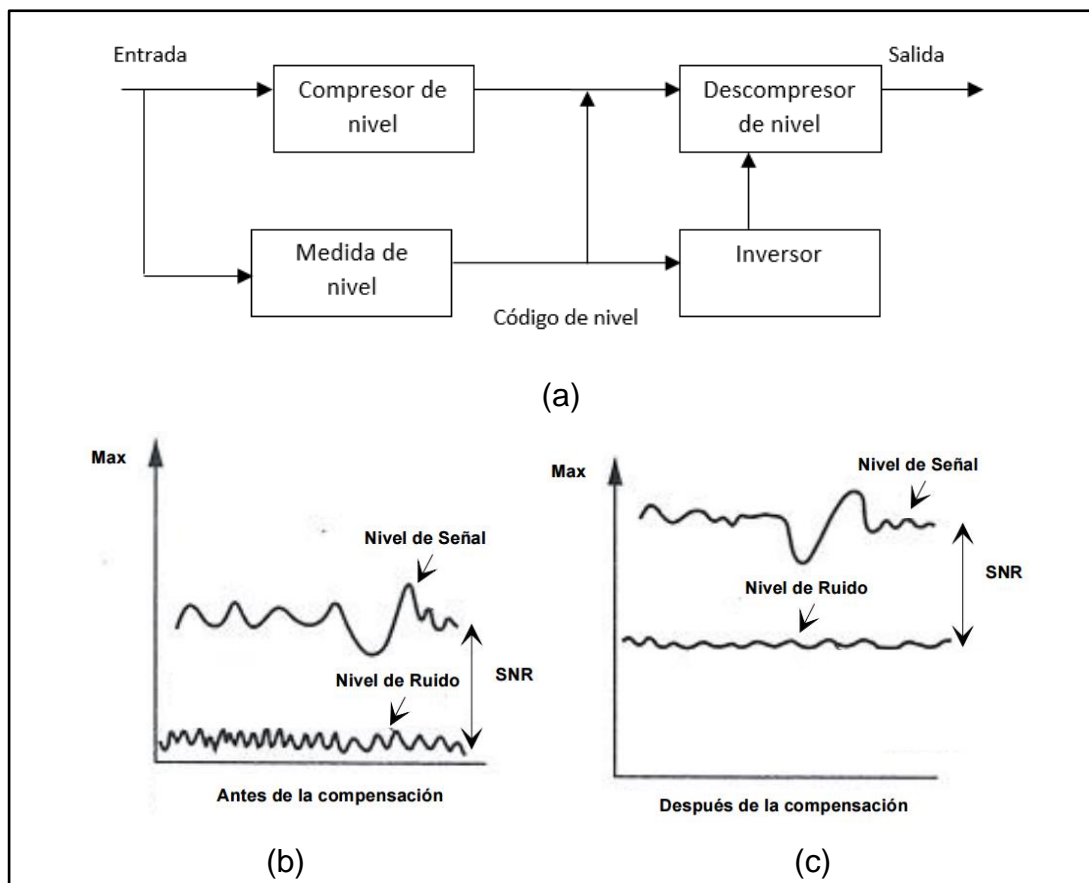


Figura 3.7. La compensación digital [28]

La señal de entrada es monitoreada y cuando su nivel disminuye por debajo de un valor referencial, se amplifica en el codificador. La ganancia aplicada para aumentar la señal es añadida en los datos transmitidos de manera que en el decodificador se puede añadir una atenuación del mismo valor para la compensación.

Una forma habitual de obtener la codificación de la ganancia es reducir el tamaño de palabra de las muestras, para que de esta manera se necesite transmitir un número menor de *bits*. Al realizar esta reducción, la distorsión se incrementa por cada *bit* eliminado.

Una alternativa para la reducción de la longitud de palabra consiste en convertir la señal PCM uniforme en una señal no uniforme, en este caso el tamaño del paso de cuantificación se incrementa con la magnitud de la muestra tal que el nivel de distorsión es mayor cuando se tiene niveles más altos.

La compensación está relacionada con la codificación en punto flotante, dicha técnica puede ser apreciada en la Figura 3.8.

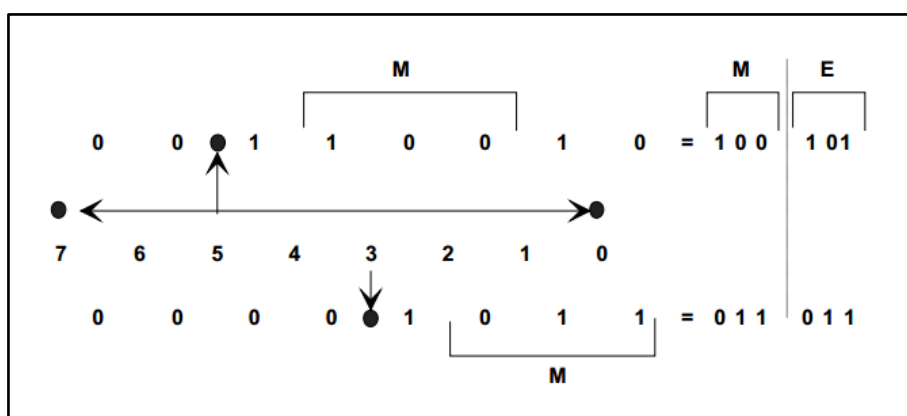


Figura 3.8. Codificación en punto flotante [28]

El valor de la muestra se expresa como una mantisa y un exponente binario que determina el desplazamiento que debe realizarse sobre dicha parte para obtener el valor absoluto correcto sobre una escala PCM. El exponente es el equivalente a la ganancia o al factor de escala en un sistema compensado.

Para este tipo de codificación, la relación señal a ruido viene determinada por el número de *bits* en la mantisa.

En la Figura 3.9. se puede apreciar cómo la relación señal a ruido varía como una función diente de sierra, donde el mejor valor que se obtiene es cuando la mantisa cercana al desbordamiento es reemplazada por el peor valor cuando la mantisa se desborda y el exponente se incrementa.

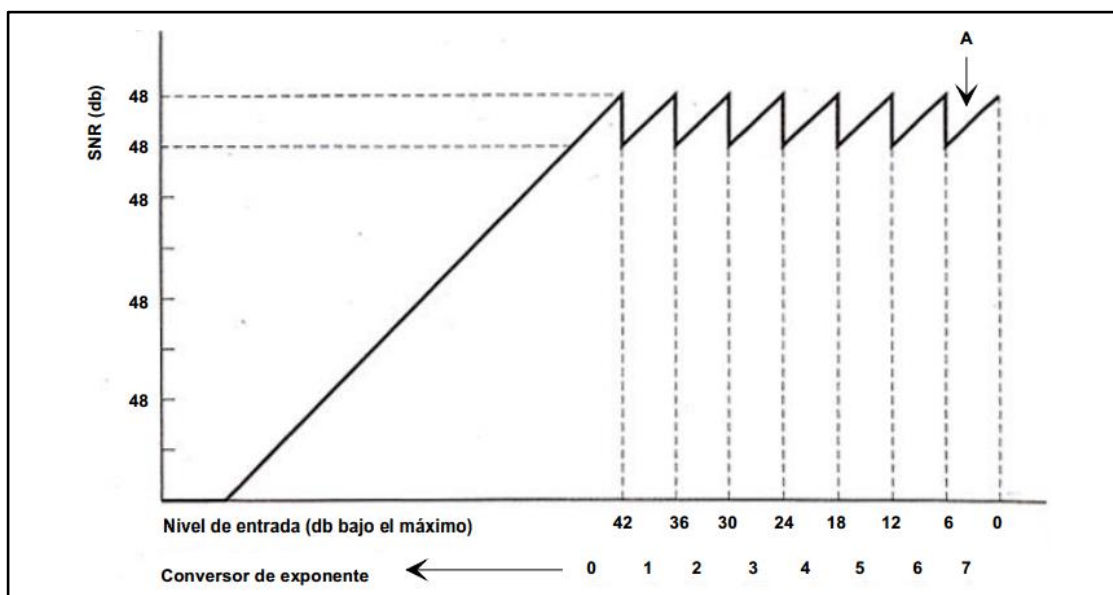


Figura 3.9. Ejemplo de un sistema de codificación en punto flotante con mantisa de 8 bits y 3 bits de exponente [28]

Como se aprecia en la Figura 3.9., la máxima relación SNR es de 48 dB con una entrada máxima de 0 dB. Como el nivel de entrada disminuye y el nivel de ruido permanece igual, entonces la SNR disminuye a 42 dB. Una reducción mayor en el nivel de la señal tendría como consecuencia que el conversor desplace el rango (punto A) aumentando la ganancia de la entrada analógica en 6 dB. La SNR volvería a su valor inicial y el exponente variaría de 7 a 6 para obtener el mismo cambio en el receptor.

Un sistema en punto flotante necesita de un exponente con cada mantisa, siendo esto innecesario y costoso ya que los sistemas de audio reales no cambian rápidamente como para necesitar una información tan precisa. Por tanto, una alternativa más adecuada es la codificación en punto flotante en bloque. En esta codificación, la magnitud de la muestra mayor de un bloque se utiliza para determinar el valor de un exponente que será válido para el bloque completo. Enviar un exponente por bloque requiere una velocidad de datos menor que en la codificación en punto flotante.

En el proceso de codificación por bloques, la re-cuantificación en el codificador aumenta el error de cuantificación, pero lo hace a lo largo de la duración del bloque. En la Figura 3.10. se puede apreciar que si se presenta un transitorio hacia el final del bloque, el decodificador reproducirá la forma de onda correctamente, sin embargo el ruido de cuantificación comenzará desde el principio del bloque y puede producir un aumento en la distorsión (también llamado pre-eco) que es audible antes del transitorio. Para hacerlo inaudible se puede utilizar un enmascaramiento temporal.

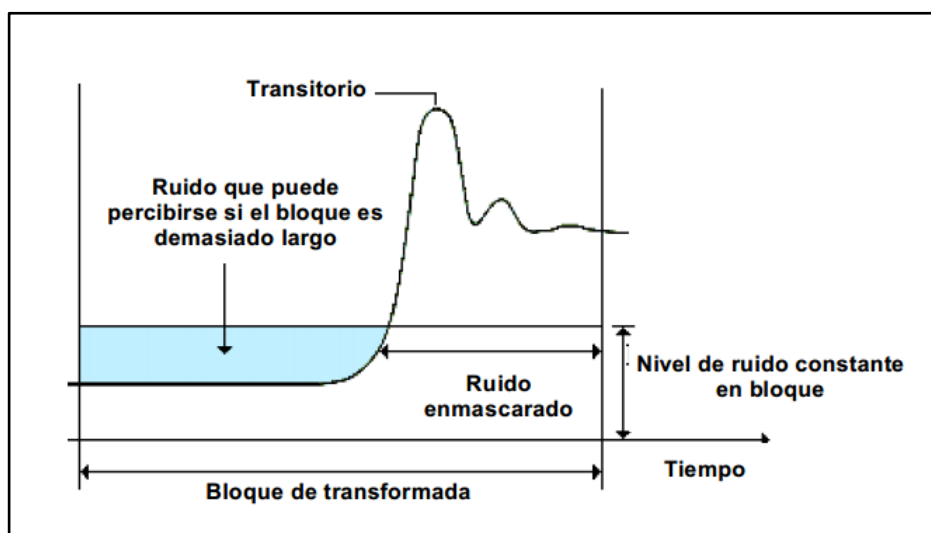


Figura 3.10. Transitorio al final de un bloque de una transformada [28]

La codificación sub-banda separa el espectro de audio en diferentes bandas de frecuencia, con lo que cada banda puede ser procesada de forma individual.

Este tipo de codificación permite también que el nivel de los productos de distorsión aumente selectivamente de manera que la distorsión sólo se crea a frecuencias en las que el enmascaramiento espectral sea eficaz.

La señal de audio es considerada como una forma de onda en el dominio del tiempo, sin embargo, el análisis espectral indica que cualquier forma de onda se puede representar por un conjunto de elementos armónicamente relacionados de amplitud y fases, por tanto es posible descomponer una forma de onda periódica en sus frecuencias y fases.

Los sonidos en la vida real no son periódicos, contienen transitorios para los que la transformación no puede realizarse con precisión, para aquello se puede separar la forma de onda en dos segmentos cortos y transformar cada uno de ellos de forma individual. De esta manera se reduce el retardo pero existe la posibilidad de que aparezcan efectos no deseados debido a que la forma de onda ha sido truncada en ventanas temporales rectangulares. Una solución para esto, es el uso de funciones de ventana, y superponer los segmentos como se aprecia en la Figura 3.11.

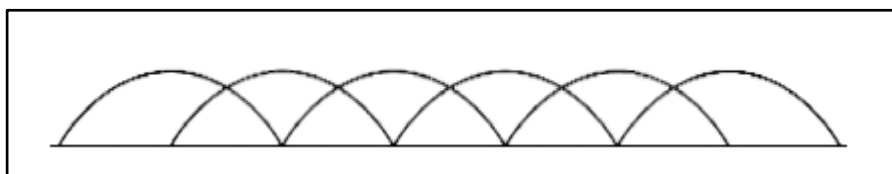


Figura 3.11. Aplicación de funciones matemáticas tipo ventana para la superposición de segmentos [28]

La DFT no es capaz de producir espectro continuo sino coeficientes a frecuencias discretas. La resolución en frecuencia (número de coeficientes a diferentes frecuencias) es igual al número de muestras en la ventana.

El uso de la DFT necesita de cálculos intensivos y además es necesario el empleo de números complejos para determinar la fase y la amplitud, por tanto se tiene como alternativa la DCT o la MDCT, que tiene la capacidad de eliminar la sobrecarga que se produce por la superposición de las ventanas y volver al dominio críticamente muestreado, dicho término es usado para indicar que el número de coeficientes no exceda el número que se obtendría sin superposición de ventanas.

3.5.4.3. Función ventana

Una función ventana se define como una función matemática, usada frecuentemente dentro del procesamiento de señales para evitar las discontinuidades al inicio y final de los bloques analizados.

La función es usada de forma general para limitar una señal considerada como infinita, para que la misma pueda ser analizada en un tiempo finito.

La utilización de una ventana modifica el espectro en frecuencia de una señal. Existen distintos tipos de ventana que permiten obtener distintos resultados en el dominio de las frecuencias.

3.5.4.4. Transformada Discreta del Coseno Modificada (MDCT)

Se define como una transformada lineal ortogonal, la cual se basa en la teoría de cancelación del *aliasing* en el dominio del tiempo. La MDCT está diseñada para aplicarse sobre bloques consecutivos con gran cantidad de datos, donde los bloques de solapan de manera que la mitad de un bloque coincide con la primera mitad del siguiente bloque. Este solapamiento, además de las cualidades de energía de compactación de la DCT, hace que la MDCT sea ampliamente usada para aplicaciones de compresión de señal. Como resultado de estas ventajas, la MDCT es usada en formatos de compresión de audio como MP3, AC-3, WMA y AAC.

La definición usada de la MDCT se puede apreciar en la Ecuación (2):

$$X(m) = \sum_{k=0}^{n-1} f(k)x(k) \cos\left(\frac{\pi}{2n}\left(2k + 1 + \frac{n}{2}\right)(2m + 1)\right) \text{ para } m = 0 \dots \frac{n}{2} - 1 \quad (2)$$

En aplicaciones comunes para compresión de señales, las propiedades de la transformada mejoran aún más mediante el uso de una función de ventana Wn ($n = 0, \dots, 2N - 1$) que es multiplicada con Xn y Yn en las fórmulas de la MDCT e IMDCT, con el fin de evitar discontinuidades en los límites $n = 0$ y $2N$, al hacer que la función tienda a ser cero en estos puntos.

3.6. Formatos de archivo de sonido digital

Existe una gran variedad de formatos de sonido digital, entre los principales se pueden destacar los de la Tabla 3.4.

Tabla 3.4. Principales formatos de sonido digital [28]

Formato	Aplicación	Tamaño de archivo
WAV	Datos PCM, sin compresión	Muy grande
MID	Patrones activos en tarjetas de sonido, usado en partituras	Muy pequeño
MUS	Similar al MID	Muy pequeño
AIF	Formato estándar de plataformas MAC	Muy grande
MP3	Formato principal para compresión de audio	Pequeño
RM	Usado por Internet para <i>streaming</i>	Muy pequeño
AAC	Formato usado para compresión de audio, con mejor desempeño que MP3 para frecuencias bajas	Pequeño

De manera general, son dos los tipos fundamentales de archivos digitales de audio:

- Archivos basados en PCM, en donde los datos contenidos representan exactamente una muestra de la forma de onda. En este formato, se realiza digitalización pero no compresión. Entre los ejemplos de estos archivos tenemos a WAV y AIF.
- Archivos comprimidos, donde los datos son comprimidos mediante el uso de modelos psicoacústicos, ya sea para almacenamiento o transmisión. Entre los ejemplos, tenemos a MP3, Ogg Vorbis, WMA y AAC.

En la Tabla 3.5. se puede apreciar los principales archivos comprimidos con sus extensiones y los estándares a los que pertenecen.

Tabla 3.5. Archivos de audio comprimido y sus estándares [28]

Formato	Extensión del fichero	Estándar
Digital Theater Systems	DTS	ETSI TS 102 114
Dolby Digital	AC-3	ATSC
MPEG-1 Layer I	MPA	ISO/IEC-11172-3
MPEG-1 Layer III	MP3	ISO/IEC-11172-3
MPEG-2 Layer I	MPA	ISO/IEC-13818-3
MPEG-2 Layer III	MP3	ISO/IEC-13818-3
MPEG-2 Layer VII	AAC	ISO/IEC 13818-7
MPEG-4 Layer III subparte IV	AAC	ISO/IEC 14496-3

3.6.1. Formateo del *bitstream*

El *bitstream* que se obtiene a la salida del codificador está normalizado. El *bitstream* resultante está dividido en pequeños paquetes de datos denominados tramas (*frames*), donde cada trama es totalmente independiente de la otra.

Una trama MPEG está dividida en cuatro partes:

- **Encabezamiento (*header*):** Contiene la información de sincronización y detalles sobre las características de la trama.
- **Control de error (CRC):** Se usa para detectar posibles errores en el *bitstream*.
- **Datos de audio:** Son las muestras del audio comprimido que contiene el *bitstream*.
- **Datos auxiliares (*Ancillary Data*):** Son los datos auxiliares que puede o no contener el *bitstream*.

3.6.2. Encabezamiento de la trama

Los primeros bytes de las tramas contienen información importante sobre los datos contenidos en el paquete. Dicha información contenida permite a la trama ser autosuficiente y completamente independiente del resto de las tramas en un archivo de audio completo.

Un ejemplo de encabezamiento de la trama se indica en el siguiente esquema, donde cada letra representa un *bit*.

AAAAAAAA AABBCCD EEEEEFFGH IJJKLMM

El esquema de encabezamiento depende exclusivamente del formato de audio a utilizar, por ejemplo el formato AAC emplea como esquema los formatos ADTS y ADIF.

CAPÍTULO IV

ADVANCED AUDIO CODING (AAC)

4.2. Generalidades

4.2.1. Introducción

AAC se define como un estándar para la codificación de audio basado en un algoritmo de compresión con pérdida, que al igual que todos los codificadores de audio, tiene como objetivo reducir el volumen de datos, pero obteniendo a la salida un archivo que suene lo más parecido posible al original.

AAC fue diseñado para ser el sucesor del formato MP3, debido a que puede lograr una mejor calidad de sonido a similares tasas de *bits* [8].

El formato AAC corresponde al estándar internacional ISO/IEC 13818-7 como una extensión de MPEG-2, aunque debido a su gran rendimiento fue definido dentro de la familia de MPEG-4. Además, AAC ha sido adoptado en varios estándares de televisión móvil tales como DVB-H y ATSC-M/H, además en estándares de televisión digital como es el caso de la ISDB-Tb.

Dicho formato ha sido elegido por varias compañías, como su formato principal, ya que una de sus características es brindar protección de derechos de autor a un determinado archivo de audio dentro del proceso de codificación.

AAC fue desarrollado con la cooperación y contribuciones de compañías como AT&T, Laboratorios Dolby, Sony y Nokia. AAC fue declarado como estándar internacional por la MPEG en abril de 1997, el mismo fue especificado como parte 7 en el estándar MPEG-2 y como parte 3 y sub parte 4 en el estándar MPEG-4.

4.2.2. Estandarización

En 1997, AAC fue introducido como la parte 7 del estándar MPEG-2, conocido también como ISO/IEC 13818-7:1997.

MPEG-2 en su parte 7, define tres tipos de perfiles:

- Perfil de baja complejidad (AAC-LC).
- Perfil principal (AAC-Main).
- Perfil de frecuencia de muestreo escalable (AAC-SSR).

El perfil AAC-LC consiste de un formato base muy similar al formato de Codificación Perceptual (PAC) usado por AT&T [9], con la adición de TNS (*Temporal Noise Shapping*) [10], la ventana doble de Káiser, un cuantificador no uniforme, y una reelaboración del formato del *bitstream*, para manejar hasta 16 canales estéreo, 16 canales mono, 16 canales de efectos de baja frecuencia y 16 canales de comentario en un solo *bitstream*.

El perfil principal de AAC añade un conjunto de predictores recursivos que son calculados sobre cada salida del banco de filtros. El perfil AAC-SSR hace uso de un banco de filtros PQMF de 4 bandas, con cuatro bancos de filtros pequeños siguientes, con el fin de permitir velocidades de muestreo escalables.

En 1999, MPEG-2 en su parte 7 fue actualizada e incluida en la familia de los estándares MPEG-4, llegando a ser conocido como MPEG-4 parte 3, audio MPEG-4 o ISO/IEC 14496-3:1999, en esta actualización se incluyen ciertas mejoras, las cuales se detallan a continuación:

- Se añaden los Tipos de Objetos de Audio (*Audio Object Types*), los cuales son usados para permitir la inter operatividad con un rango de divergencia de otros formatos de audio como Twin VQ, CELP, HVXC, Text to Speech y audio estructurado MPEG-4.

- Se añade el bloque de PNS (*Perceptual Noise Substitution*) a los tres perfiles definidos de manera inicial para AAC y son definidos en los estándares MPEG-4.

El *software* de referencia para la parte 3 de MPEG-4 está especificado en la parte 5 de MPEG-4 y la formación del *bitstream* está especificada en la parte 4 de MPEG-4. La versión 2 para el audio MPEG-4 (ISO/IEC 14496-3:1999/Amd 1:2000) define nuevos Tipos de Objetos de Audio para AAC, entre los cuales se tiene al perfil AAC de bajo retraso (AAC-LD), la codificación aritmética en BSAC y las versiones de codificación de audio paramétrica usando Error Resiliente (ER) [11].

El perfil HE-AAC (definido como perfil de AAC-LC con SBR) y el perfil AAC-LC están estandarizados bajo la ISO/IEC 14496-3:2001/Amd 1:2003 [12]. El perfil HE-AAC en su versión 2 (definido como perfil de AAC-LC con SBR y estéreo paramétrico) fue especificado bajo la ISO/IEC 14496-3:2005/Amd 2:2006 [13].

La versión actual de los estándares AAC están definidos bajo la ISO/IEC 14496-3:2009 [14].

AAC+ en su versión 2 es también estandarizada por la ETSI como TS 102005 [15].

La parte 3 de MPEG-4 también contiene otros métodos para la compresión de sonido, el mismo incluye formatos de compresión sin pérdidas, formatos de compresión de audio sintético y de baja tasa de *bits*.

4.2.3. Ventajas de AAC con respecto a MP3

AAC está diseñado para ser el sucesor del formato definido como la parte 3 de MPEG-1, conocido también como formato MP3, el cual está especificado por la ISO/IEC 11172-3 (audio MPEG-1) y 13818-3 (audio MPEG-2).

Varias pruebas realizadas sobre el formato AAC indicaron que el mismo posee mejor calidad de sonido y transparencia que el formato MP3 para archivos codificados a la misma tasa de *bits* [8], pero a partir de esta comparación, se han determinado varios aspectos que indican que cada formato de audio posee sus propias ventajas y que la calidad para ambos es muy similar.

Como una aproximación, cuando se usan ambos codificadores, la ventaja de AAC sobre MP3 tiende a ser evidente sobre tasas de *bits* mayores a 100 kilobit/s, pero ciertamente los codificadores AAC no son tan buenos como MP3 para la optimización de las herramientas de codificación.

Las principales mejoras que posee AAC con respecto a MP3, se detallan a continuación:

- Se tienen mayores tasas de frecuencia de muestreo (de 8 a 96 kHz para AAC y de 16 a 48 kHz para MP3).
- Soporta hasta 48 canales (MP3 soporta hasta 2 canales en modo MPEG-1).
- Tasa de *bits* arbitrarias y longitud de trama variable.
- Estandarizada velocidad de *bits* constante con *bit* de reserva.
- Alta eficiencia y empleo de banco de filtros simple (MP3 emplea codificación híbrida, AAC emplea MDCT en su totalidad).
- Alta eficiencia de codificación para señales estacionarias (AAC usa bloques de tamaño de 1024 o 960 muestras, de manera que permite mayor eficiencia de codificación que los bloques de 576 muestras de MP3).
- Alta precisión de codificación para señales transitorias (AAC usa bloques de tamaño de 120 o 128 muestras, permitiendo mayor precisión de codificación que los bloques de 192 muestras de MP3).
- Se emplea la función ventana derivada de Káiser-Bessel para eliminar la fuga espectral.
- Mejor manejo de las frecuencias de audio por encima de los 16 kHz.
- Manejo de audio estéreo de manera más flexible (diferentes métodos pueden ser usados en diferentes rangos de frecuencia).

- Añade módulos en el proceso de codificación, para incrementar la eficiencia de compresión, tales como TNS, PNS, entre otros. Estos módulos pueden ser combinados para crear los diferentes perfiles de codificación.

En general, el formato AAC permite mayor flexibilidad para el diseño de codificadores en comparación con MP3, y corrige mucho de los diseños establecidos en las especificaciones del audio MPEG-1. El incremento de flexibilidad conduce a estrategias de codificación más concurrentes y, como resultado, una compresión más eficiente. Sin embargo, las ventajas de AAC no son del todo decisivas, y a pesar de todas las ventajas indicadas, MP3 ha demostrado ser sorprendentemente robusto a pesar de varios defectos considerables.

Los perfiles AAC y HE-AAC son mejores que los perfiles de MP3 a velocidades bajas de *bits* (típicamente menores de 128 kilobit/s), esto debido al uso de parámetros de codificación como es la MDCT y un mejor tamaño de la función ventana, que dejan a MP3 sin poder competir en cuanto al proceso de codificación.

Mientras el formato MP3 posea *hardware* casi universal y soporte de *software*, será el principal formato debido a que el mismo fue usado en primera instancia como formato generalizado de archivos musicales, sin embargo AAC se presenta como un fuerte contendiente y el apoyo hacia este formato ha crecido en los últimos años.

4.3. Funcionalidad de AAC

AAC es un algoritmo de codificación de audio en banda ancha que explota dos estrategias de codificación primaria, para reducir dramáticamente la cantidad de datos necesarios, y representar un audio digital de alta calidad.

- Los componentes de la señal son perceptualmente irrelevantes y descartados.
- Las redundancias en la señal de audio codificada son eliminadas.

El actual proceso de codificación consiste de los siguientes pasos:

- La señal es convertida del dominio del tiempo al dominio de la frecuencia usando para aquello la MDCT. Esto se hace mediante el uso de banco de filtros que toman un apropiado número de muestras en el dominio del tiempo y convierten dichas muestras al dominio de la frecuencia.
- La señal en el dominio de la frecuencia es cuantificada basada en un modelo y codificación psicoacústica.
- Los códigos de corrección de errores internos son añadidos.
- La señal es almacenada o transmitida.
- Con el fin de evitar las muestras corruptas, de manera opcional, se puede aplicar en cada trama una moderna implementación del algoritmo Luhnmod N [16].

El estándar de audio MPEG-4 no define un simple conjunto de esquemas de compresión de alta eficiencia sino más bien una compleja caja de herramientas para realizar una amplia gama de operaciones desde códigos de audio de bajas tasas de *bits* hasta códigos de audio de alta calidad.

- La familia de algoritmos de codificación de audio MPEG-4 extiende el rango desde códigos de audio baja tasa de *bits* (debajo de los 2 kilobit/s) hasta códigos de audio de alta calidad (hasta 64 kilobit/s por canal).
- AAC ofrece frecuencias de muestreo entre 8 y 96 kHz y canales entre 1 y 48.
- En contraste con el banco de filtros híbrido de MP3, AAC usa la MDCT junto con una longitud superior de ventana de 1024 o 960 puntos.
- Los codificadores AAC pueden conmutar dinámicamente entre un bloque simple MDCT de 1024 puntos u 8 bloques de 128 puntos (o entre 960 puntos y 120 puntos, respectivamente).
- Si se produce un cambio de señal o un transitorio, 8 ventanas cortas de 128 o 120 puntos son elegidas para una mejor resolución temporal.

4.3.1. Codificación modular

AAC posee un enfoque modular para la codificación. Dependiendo de la complejidad del *bitstream* para ser codificado, el rendimiento deseado y una aceptable salida, los diseñadores pueden crear perfiles para definir qué conjunto específico de herramientas se quiere usar para alguna aplicación en particular.

La parte 7 del estándar MPEG-2 (AAC) fue primeramente publicada en 1997 y ofrece tres perfiles:

- **Baja complejidad:** El perfil más simple y mayormente usado y soportado.
- **Principal:** Similar al perfil anterior, con el uso de predicciones.
- **Frecuencia de muestreo escalable:** Definido como MPEG-4 AAC-SSR, que emplea y prioriza el uso del control de ganancia.

La parte 3 del estándar MPEG-4 define varias nuevas herramientas de compresión (conocidas como Tipos de Objetos de Audio) y su uso en nuevos perfiles.

AAC no es usado en algunos de los perfiles de audio MPEG-4. La parte 7 de MPEG-2 en su perfil de baja complejidad (LC), el perfil principal y el perfil de frecuencia de muestreo escalable (SSR) son combinados con otros tipos de objeto en los perfiles de audio de MPEG-4 [17], algunos de ellos se indican a continuación:

- **Main Audio Profile:** Definido en 1999, hace uso de la mayoría de tipos de objetos de audio (AAC Main, AAC-LC, AAC-SSR, AAC-LTP, AAC Scalable, Twin VQ, CELP, HVXC, TTSL).
- **Scalable Audio Profile:** Definido en 1999, hace uso de AAC-LC, AAC-LTP, AAC Scalable, Twin VQ, CELP, HVXC, TTSL.
- **Speech Audio Profile:** Definido en 1999, hace uso de CELP, HVCX y TTSL.

- ***Synthetic Audio Profile:*** Definido en 1999, hace uso de TTSI, y Main Synthesis.
- ***High Quality Audio Profile:*** Definido en 2000, hace uso de AAC-LC, AAC-LTP, AAC Scalable, CELP, ER-AAC-LTP, ER-AAC-LC, ER-AAC Scalable, ER-CELP.
- ***Low Delay Audio Profile:*** Definido en 2000, hace uso de CELP, HVXC, TTSI, ER-AAC-LD, ER-CELP, ER-HVCX.
- ***Mobile Audio Internetworking Profile:*** Definido en 2000, usa ER-AAC, ER-AAC Scalable, ER-Twin VQ, ER-BSAC, ER-AAC-LD.
- ***AAC Profile:*** Definido en 2003, usa AAC-LC.
- ***High Efficiency AAC Profile:*** Definido en 2003, usa AAC-LC, SBR.
- ***High Efficiency AAC v2 Profile:*** Definido en 2006, usa AAC-LC, SBR, PS.

4.3.2. Corrección de errores

Cuando se aplica la protección de errores en el formato AAC, el mismo permite la corrección de errores hasta un cierto punto. El código de corrección de errores es usualmente aplicado por igual para todo el *payload*. Sin embargo, ya que las diferentes partes de un *payload* en AAC muestran una diferente sensibilidad a los errores de transmisión, el corrector de errores no resulta muy eficiente.

Para aquello, el *payload* de AAC puede ser subdividido en partes con diferente sensibilidad de errores.

- Los códigos de corrección de errores pueden ser aplicados a cualquiera de estas partes, utilizando la herramienta de protección de errores (EP), la misma que se encuentra definida en el estándar MPEG-4.
- El conjunto de herramientas provee la capacidad para la corrección de errores de las partes más sensibles del *payload*.

4.3.3. Error Resiliente (ER)

Las técnicas para el error resiliente pueden ser usadas para realizar esquemas de codificación más robustas contra los errores.

Para AAC, existen tres métodos diseñados, los cuales están definidos en el estándar MPEG-4:

- HCR (*Huffman Codeword Reordering*), para evitar la propagación del error dentro de los datos espectrales.
- VCB11 (*Virtual Codebooks*), para detectar errores graves dentro de los datos espectrales.
- RVLC (*Reversible Variable Length Code*), para reducir la propagación del error dentro de los factores de escala.

4.3.4. Perfil de bajo retardo

El codificador de audio AAC con bajo retardo (AAC-LD) está diseñado para combinar las ventajas de la codificación del audio perceptual con el bajo retardo necesario para una comunicación de dos vías. El mismo está estrechamente derivado del formato MPEG-2 AAC.

4.4. Licencias y patentes

No se requiere licencias para generar o distribuir contenido en formato AAC [18]. Esta única razón puede hacer de AAC un formato más atractivo para distribuir contenido que su predecesor MP3, particularmente para contenido del tipo *streaming*, como es el caso de radio internet.

Sin embargo, una patente es requerida para todos los desarrolladores de códigos AAC [19]. Por esta razón, las implementaciones de *software* libre y de código abierto, como FFMPEG y FAAC, pueden ser distribuidas en código fuente únicamente, de esta manera se evita infringir la patente.

4.5. Extensiones y mejoras

Algunas extensiones han sido añadidas al formato AAC, desde la aparición del primer estándar en 1997, las mismas se detallan a continuación:

- PNS (*Perceptual Noise Substitution*), añadido en 1999 en el estándar MPEG-4. Permite la codificación de ruido como datos pseudoaleatorios.
- LTP (*Long Term Predictor*), añadido en 1999 en el estándar MPEG-4, consiste de un predictor con una menor complejidad computacional.
- ER (*Error Resilience*), añadido en la versión 2 de MPEG-4 en el año 2000, usado para transportar el error sobre canales propensos.
- AAC-LD (*AAC Low Delay*), definido en el año 2000, es usado para aplicaciones en tiempo real que necesitan de bajo retardo.
- HE-AAC (*High Efficiency AAC*), es la combinación de SBR y AAC-LC, es usado para bajas tasas de *bits*.
- HE-AAC v2 (*High Efficiency AAC v2*), es la combinación de PS (*Parametric Stereo*) y HE-AAC; usado para tasas de *bits* más bajas. Fue definido en el 2004 y 2006.
- SLS (*MPEG-4 Scalable To Lossless*), definido en 2006, puede complementar a una trama AAC para proveer la opción de decodificación sin pérdidas.

4.6. Formatos contenedores

Los datos de audio AAC pueden ser empaquetados en formas basadas en archivos MPEG-2, uno de estos es el formato básico llamado ADIF (*Audio Data Interchange Format*) [20].

ADIF consiste de una sola cabecera seguida de los bloques de datos de AAC. Sin embargo, si los datos son para ser transmitidos dentro de un flujo de transporte MPEG-2, se usa un formato de auto sincronización denominado como ADTS (*Audio Data Transport Stream*). El mismo consiste de una serie de tramas, donde cada trama tiene una cabecera seguida de los datos de audio AAC [20]. Este formato está

definido en la parte 7 de MPEG-2, pero sólo es considerado como informativo para MPEG-4, entonces un decodificador MPEG-4 no necesariamente debe soportar este formato. Estos contenedores soportan el archivo de extensión *.aac.

La parte 3 de MPEG-4 también define su propio formato de auto sincronización denominado como LOAS (*Low Overhead Audio Stream*), que no solo encapsula AAC. Este formato fue definido para usarse en las tramas de transporte para DVB cuando los codificadores usan SBR o estéreo paramétrico como extensiones de AAC. Sin embargo, esto está limitado para una sola trama AAC no multiplexada. Este formato es también referido como LATM (*Low Overhead Audio Transport Multiplex*), el cual solo es una versión de múltiples tramas intercaladas de un LOAS.

En la Tabla 4.1. se indican los formatos contenedores según el tipo de aplicación.

Tabla 4.1. Formatos contenedores de AAC [33]

Aplicación	Estándar	Descripción
Multiplexación	ISO/IEC 14496-1	Esquema de multiplexación MPEG-4 (M4Mux)
Multiplexación	ISO/IEC 14496-3	LATM
Almacenamiento	ISO/IEC 14496-3 (Informativo)	ADIF
Almacenamiento	ISO/IEC 14496-12	Formato MPEG-4 (MP4)
Transmisión	ISO/IEC 14496-3 (Informativo)	ADTS (sólo para AAC)
Transmisión	ISO/IEC 14496-3	LOAS (basado en LATM)

4.7. Productos que definen el formato AAC

4.7.1. Estándares HDTV

4.7.1.1. ISDB-T Japonés

En diciembre de 2003, Japón inició la transmisión terrestre bajo el estándar ISDB-T que implementa vídeo MPEG-2 y audio MPEG-2 AAC. En abril de 2006, Japón inicia la transmisión de programación para dispositivos móviles con el estándar ISDB-T, denominado como *One-Seg*, que usó la primera implementación de video H.264/AVC con audio HE-AAC.

4.7.1.2. ISDB-Tb Internacional

En diciembre de 2007, Brasil inició la transmisión terrestre bajo el estándar ISDB-Tb que implementa video H.264/AVC con audio AAC-LC y video H.264/AVC con audio HE-AAC v2 para la programación en dispositivos móviles.

4.7.1.3. Digital Video Broadcasting (DVB)

Soporta AAC, HE-AAC y HE-AAC v2 como codificador de audio para sus aplicaciones desde el 2004. Las transmisiones DVB que usan codificación de video H.264, normalmente usan la codificación HE-AAC para audio.

4.7.2. Hardware

A nivel de *hardware*, existen varias compañías que han adoptado varios perfiles basados en AAC para la codificación de sus archivos de audio en equipos portables como reproductores, teléfonos celulares y consolas de vídeo.

4.7.3. Software

La mayoría de reproductores de contenido multimedia incluyen decodificadores para el proceso de obtención de sonido en AAC, o incluso emplean librerías externas para el proceso de decodificación.

- **Nero Digital:** En mayo de 2006, la empresa Nero lanza una herramienta gratuita denominada Nero Digital Audio, la cual es capaz de codificar sonido AAC bajo los perfiles AAC-LC, HE-AAC y HE-AAC v2 [21].
- **FAAC y FAAD2 (*Freeware Advanced Audio Coder and Decoder 2*):** FAAC soporta archivos AAC bajo los perfiles LC, Main y LTP [22]. FAAD2 soporta los perfiles LC, Main, LTP, SBR y PS [23]. Cabe indicar que FAAD2 es *software* libre, pero FAAC es licenciado.
- **Fraunhofer FDK AAC:** Se trata de un codificador y decodificador AAC de código abierto incluido en *Android* que actualmente ha sido portado hacia otras plataformas.
- **FFMPEG:** Las librerías de FFMPEG contienen códigos de *software* libre para la codificación o decodificación AAC (la decodificación se realiza de manera experimental bajo el uso de librerías externas).

4.8. MPEG AAC

4.8.1. Introducción

MPEG AAC empezó a estructurarse desde el año 1994 por un comité de audio MPEG-2, cuyo objetivo fue la definición de un estándar multicanal con gran calidad de audio que supere a las características dadas por MPEG-1.

Dicha estructuración culminó en el año 1997 con el nombre de MPEG AAC bajo la estandarización ISO/IEC 13818-7.

MPEG AAC hace uso de varias técnicas para la codificación de audio, las cuales otorgan un codificador con características avanzadas para brindar alta calidad de audio y soporte multicanal.

4.8.2. Generalidades

El objetivo para el desarrollo de MPEG AAC fue la de generar un codificador capaz de obtener una indistinguible calidad de audio bajo las especificaciones de la

ITU-R TG10-2 [ITU-R TG10-2/3] a una tasa de *bits* de 384 kilobit/s (o menor) para 5 señales de audio con canales de gran ancho de banda.

En diferentes pruebas realizadas, se determinó que MPEG AAC satisface los requerimientos de calidad de la ITU-R a 320 kilobit/s con 5 canales de banda ancha (o con menor cantidad de canales).

Las especificaciones de MPEG AAC son resultado de un esfuerzo colaborativo de compañías alrededor del mundo que ha contribuido con el desarrollo de tecnologías avanzadas para la codificación de audio.

MPEG AAC combina la eficiencia de codificación de un banco de filtros de alta resolución, técnicas de predicción, y codificación de Huffman, para lograr audio de alta calidad a bajas tasas de *bits*. Las especificaciones de AAC han sido sometidas a varias revisiones desde su primera aparición.

Con el fin de definir el sistema AAC, el comité de audio seleccionó un enfoque modular en el cual el sistema general se divide en varios bloques, donde cada uno de ellos define una parte del proceso de codificación.

El modelo de referencia de AAC describe las características de cada uno de los bloques y como los mismos operan entre sí, cada aspecto del modelo de referencia ha sido evaluado mediante varios experimentos bajo la ITU-R BS.1116.

Entre los principales bloques del modelo de referencia, se tienen los siguientes:

- Control de ganancia.
- Banco de filtros.
- Predictor.
- Cuantificador.
- Codificador sin ruido.
- Multiplexor del *bitstream*.

- TNS.
- Codificador de estéreo M/S.
- Codificador de intensidad de estéreo.

Con el fin de permitir un equilibrio entre la calidad del audio y la complejidad del codificador, el sistema AAC ofrece tres tipos de perfiles:

- Perfil Principal (AAC Main).
- Perfil de Baja Complejidad (AAC-LC).
- Perfil de Frecuencia de Muestreo Escalable (AAC-SSR).

En la configuración bajo el perfil principal de AAC, el sistema genera la mejor calidad de audio a cualquier tasa de *bits*. Con excepción del bloque de pre-procesamiento, todos los bloques del modelo de referencia son usados para este perfil, sin embargo el codificador requiere de un alto uso de memoria y procesamiento, en comparación con el perfil de AAC-LC. Además, cabe indicar que el perfil principal de AAC es capaz de decodificar un *bitstream* de un perfil AAC-LC, debido a las prestaciones que posee.

En la Figura 4.1. se puede apreciar el diagrama del modelo de referencia para un codificador MPEG AAC.

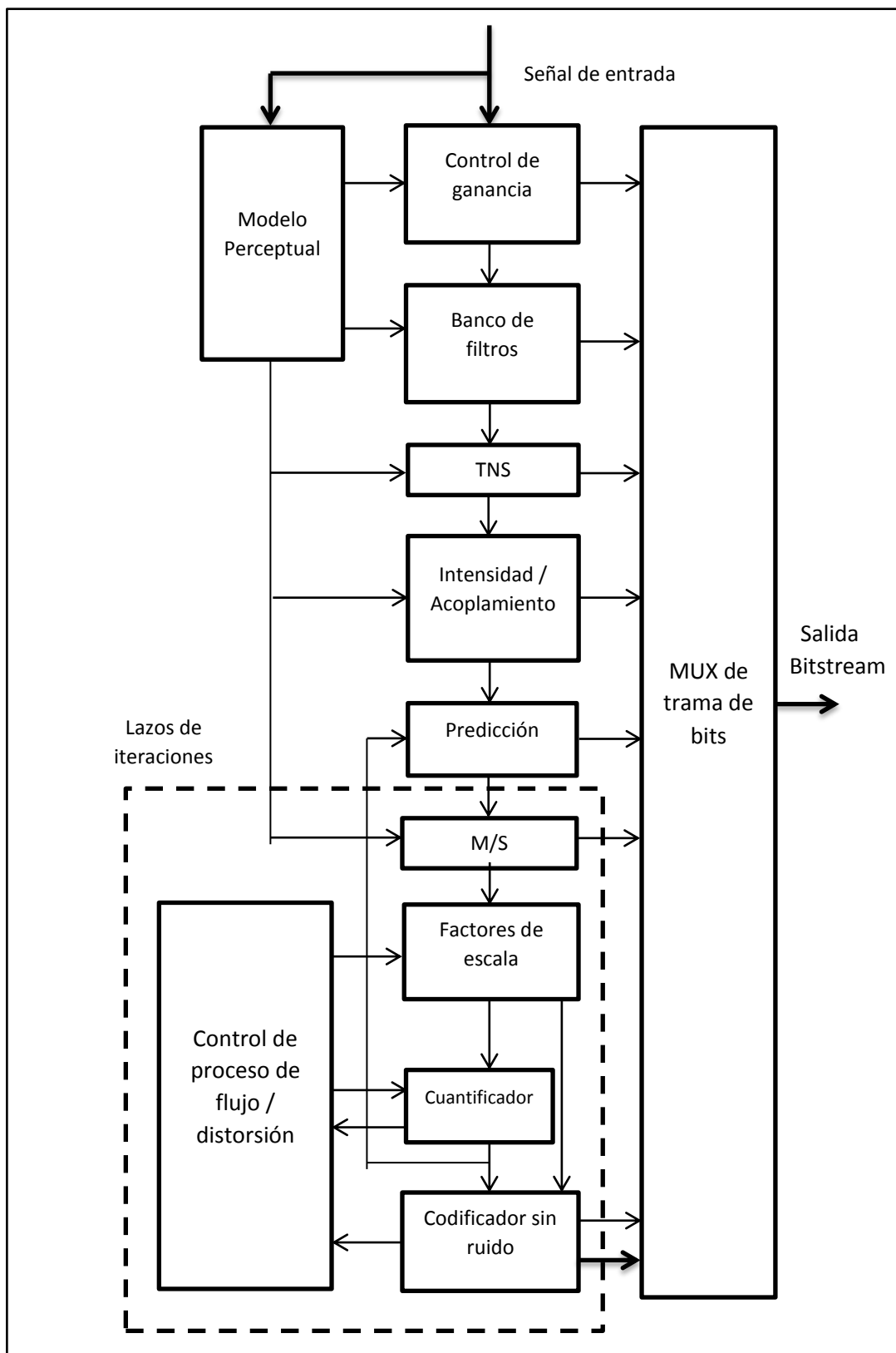


Figura 4.1. Diagrama del modelo de referencia del codificador MPEG AAC [30]

En el perfil AAC-LC, la predicción y las herramientas para el pre procesamiento no son utilizadas, y el bloque de TNS puede o no ser usado o en su caso puede ser limitado. Aunque la calidad del audio para este perfil sigue siendo alta, aunque un tanto reducida en cuanto al perfil principal de AAC, la complejidad del codificador se reduce considerablemente, reduciendo el uso de memoria y procesamiento.

En el perfil AAC-SSR, el bloque de control de ganancia es requerido. El pre-procesamiento realizado por la herramienta del control de ganancia consiste de un banco de filtros CQF, detectores y modificadores de ganancia. El bloque de predicción no es usado en este perfil, y el bloque de TNS es limitado. El perfil de AAC-SSR posee más baja complejidad que los perfiles de AAC-Main y AAC-LC y este puede proveer una señal escalable en frecuencia.

El proceso de codificación de AAC puede ser descrito de la siguiente manera:

- Primeramente, se usa un banco de filtros basado en la MDCT para descomponer la señal de entrada en componentes espectrales sub-muestreadas (dominio tiempo-frecuencia).
- Basado en la señal de entrada, se puede calcular una estimación de los umbrales de tiempo de enmascaramiento dependientes. Un modelo perceptual similar al modelo psicoacústico MPEG es usado para el sistema AAC. Los valores SMR son usados en la etapa de cuantificación para minimizar la distorsión audible de la señal cuantificada a cualquier tasa de *bits*.
- Después del análisis del banco de filtros, el bloque TNS realiza una operación de filtrado sobre los valores espectrales, donde se reemplazan los coeficientes espectrales con las predicciones residuales. La técnica TNS permite al codificador ejercer un control sobre la estructura temporal del ruido de cuantificación.
- Para señales multicanal, se hace uso de la codificación de intensidad estéreo, para este proceso, sólo se transmite la energía.
- El bloque de predicción en el dominio del tiempo puede ser usado con el fin de tomar ventaja de correlaciones entre los componentes espectrales sub-

muestreados de las tramas sub-siguientes que resultan de un aumento de la reducción de la redundancia para señales estacionarias.

- El bloque M/S (*Mid/Side*) es usado para codificadores AAC multicanal a bajas tasas de *bits*, el mismo reemplaza a la transmisión de señales de izquierda y derecha.
- Los componentes espectrales son cuantificados y codificados con el objetivo de mantener el ruido de cuantificación por debajo del umbral de enmascaramiento. Este paso se realiza empleando una etapa de análisis por síntesis y usando herramientas de compresión sin ruido. Un mecanismo denominado como “*bit* reservado” permite velocidad de datos localmente variable con el fin de satisfacer las demandas de la señal en una base trama por trama.
- Finalmente, se hace uso de un multiplexor de *bitstream* para formar la trama, la misma está formada por los coeficientes espectrales cuantificados y codificados y los parámetros de control.

El sistema MPEG AAC soporta hasta 48 canales de audio. Las configuraciones por defecto incluyen canales monofónicos y configuración de 2 y 5 canales LFE plus. Las tasas de muestreo soportadas por el sistema AAC varían desde 8 kHz hasta 96 kHz, esto se puede apreciar en la Tabla 4.2.

Tabla 4.2. Frecuencias de muestreo y *bitrate* por canal para AAC [30]

Frecuencia de muestreo (Hz)	Máxima tasa de <i>bits</i> por canal (kilobit/s)
96000	576
88200	329.2
64000	384
48000	288
44100	264.6
32000	192
24000	144
22050	132.3
16000	96
12000	72
11025	66.25
8000	48

4.8.3. Control de Ganancia

Para el perfil de AAC-SSR, el bloque de control de ganancia es implementado en la etapa de entrada del codificador. El bloque consiste de un banco de filtros PQMF, detectores y modificadores de ganancia. El banco de filtros PQMF divide cada uno de los canales de audio en cuatro bandas de frecuencia de igual ancho de banda, las cuales son muestreadas críticamente. Cada salida del banco de filtros tiene modificación en la ganancia y está es procesada por la herramienta de la MDCT para producir 256 coeficientes espectrales, para un total de 1024 coeficientes. El control de ganancia puede ser aplicado para cada una de las 4 bandas independientemente.

El control de ganancia SSR en el decodificador tiene los mismos componentes que el codificador, pero se encuentran inversamente arreglados. La parte distintiva del perfil AAC-SSR es que las señales de salida de bajo ancho de banda, y las señales de salida de baja frecuencia de muestreo, pueden ser obtenidas por descuidar la señal de las bandas superiores del PQMF. Esto conduce a la salida de anchos de banda de 18 kHz, 12 kHz y 6 kHz cuando se descartan una, dos o tres salidas de PQMF, respectivamente.

La ventaja de esta señal escalable es que la complejidad del decodificador puede ser reducida. El módulo de control de ganancia en el codificador recibe como entrada señales en el dominio del tiempo y produce como salidas datos de control de ganancia y una señal de ganancia modificada, cuya longitud es igual a la longitud de la ventana MDCT. El diagrama de bloques para el control de ganancia se puede apreciar en la Figura 4.2.

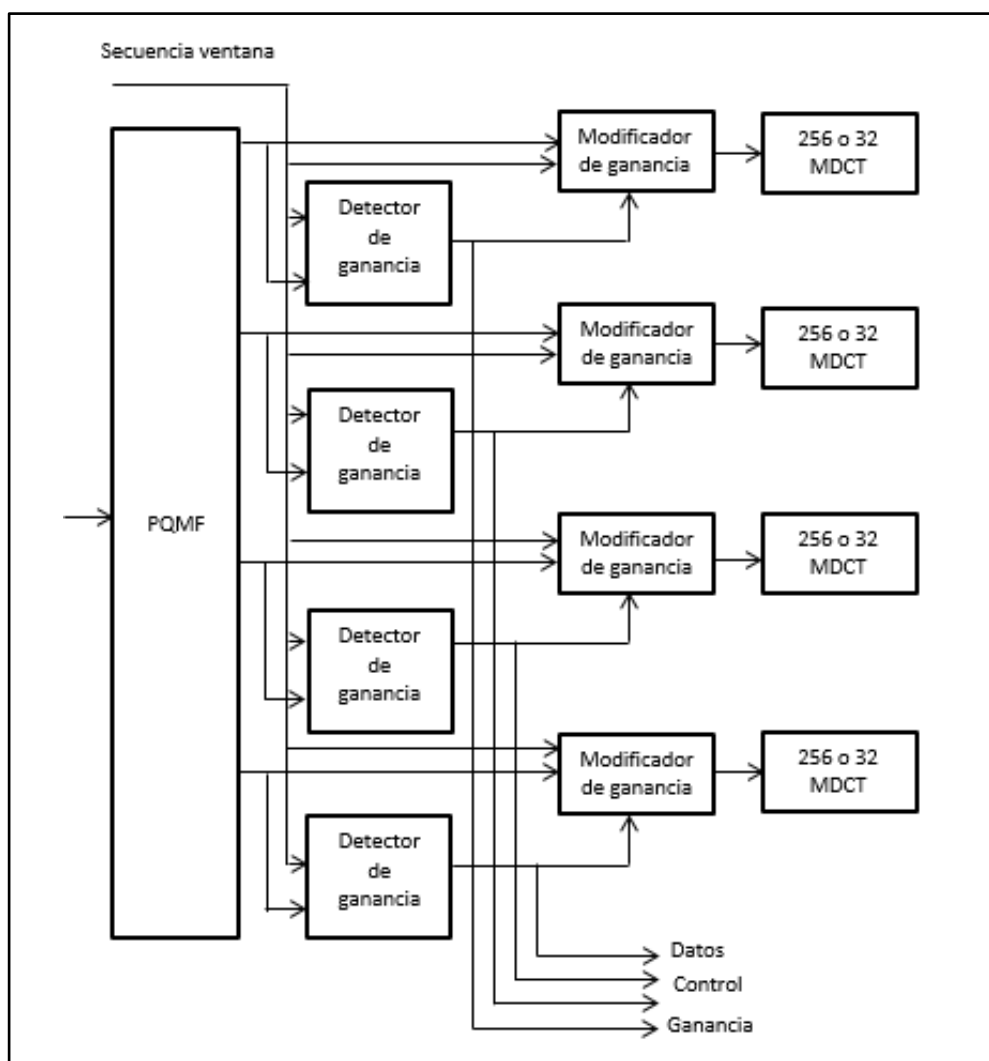


Figura 3.2. Bloque de control de ganancia para el codificador MPEG AAC [30]

En el decodificador, el bloque de control de ganancia se encuentra al final del proceso de decodificación para el perfil de AAC-SSR. El proceso en el decodificador consiste en aplicar compensación de ganancia a las secuencias producidas en cada una de las cuatro etapas de la IMDCT, la superposición, la agregación de secuencias

sucesivas con un apropiado tiempo de alineamiento y se requiere combinar estas secuencias en el proceso inverso de PQMF. La compensación de ganancia en el decodificador requiere los siguientes tres pasos para cada una de las bandas de PQMF:

- Decodificación de los datos del control de ganancia.
- Calculo de la función del control de ganancia.
- Superposición.

En la decodificación de los datos del control de ganancia, los elementos de modificación de ganancia son extraídos desde los elementos del *bitstream*. Desde este punto, la función del control de ganancia es calculada, y es usada para multiplicar la salida de la IMDCT. Consecutivamente, las secuencias son superpuestas y agregadas con tiempos de alineación apropiados. Finalmente, el IPQMF combina las cuatro bandas frecuenciales independientes para sintetizar las señal de salida.

4.8.4. Banco de filtros

Un componente fundamental del sistema MPEG AAC es la conversión de las señales de entrada en el dominio del tiempo a una representación de las mismas en el tiempo y la frecuencia, y el proceso inverso en cuanto a la decodificación. Esta conversión puede ser realizada aplicando la MDCT y la IMDCT.

La transformación de bloque de longitud N puede contener 2048 o 256 muestras de tiempo. Debido a que la función ventana tiene un efecto significativo sobre la respuesta en frecuencia del banco de filtros, el banco de filtros MPEG AAC ha sido diseñado para permitir un cambio en la forma de la ventana para una mejor adaptación de las condiciones de la señal de entrada. La forma de la ventana es determinada en el proceso de codificación y transmitida para la decodificación.

El uso de 2048 muestras en el dominio del tiempo permite una alta eficiencia de codificación para señales con espectro complejo, pero esto genera problemas para señales transitorias. Cabe indicar que los errores de cuantificación que se extienden más de unos pocos milisegundos antes de un evento transitorio no se enmascaran efectivamente. Esto conduce a un fenómeno conocido como pre-eco en el cual el error de cuantificación se extiende y se hace audible. El sistema MPEG AAC se encarga de corregir este problema permitiendo que la longitud del bloque de la transformada pueda variar en función de las condiciones de la señal, mediante un mecanismo de conmutación de bloques. Las señales que son cuasi-estacionarias son acomodadas por una transformada larga, mientras las señales transitorias son generalmente reproducidas más exactamente por las transformadas cortas. Durante la transición entre las transformadas largas y cortas, el *aliasing* es completamente cancelado.

4.8.4.1. Resolución del banco de filtros y diseño de la función ventana

La resolución de frecuencia de un banco de filtros MDCT depende de la función ventana. Una elección natural que satisface los requerimientos para la perfecta reconstrucción MDCT es la ventana de la función seno. Esta ventana produce un banco de filtros con buena resolución para los componentes de la señal espectral, mejorando la eficiencia en la codificación de señales con contenido de armónicos densos.

Sin embargo para otro tipo de señales, una ventana con mejor rechazo final puede proveer una mejor eficiencia de codificación, la ventana KBD puede satisfacer este requerimiento. El sistema AAC permite una conmutación sin pérdidas entre KBD y la ventana de la función seno mientras se conserva un muestreo crítico y una perfecta reconstrucción de la señal, como se aprecia en la Figura 4.3. Un simple *bit* por trama es transmitido dentro del *bitstream* para indicar la forma de la ventana. La forma de la ventana es variable únicamente para bloques con longitud de 2048 muestras. Las decisiones de la forma de ventana para el codificador son aplicables únicamente para la segunda parte de la función ventana, ya que la primera parte está limitada por la forma de la ventana de la trama anterior.

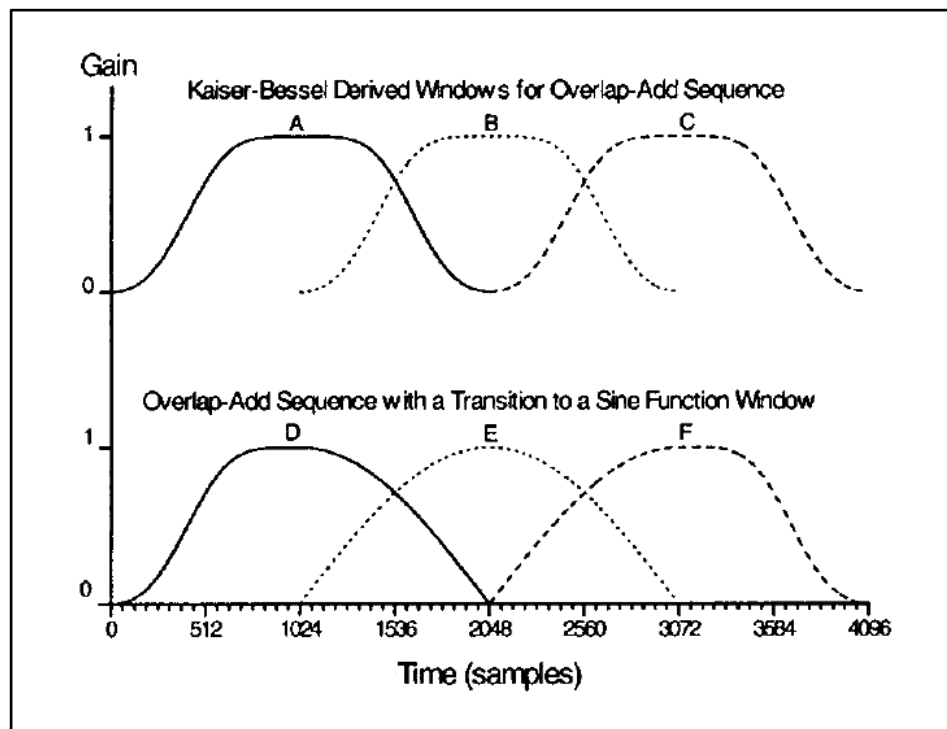


Figura 4.3. Proceso de conmutación de forma de ventana para el codificador MPEG AAC [30]

La adaptación de la resolución tiempo-frecuencia del banco de filtros para las características de la señal de entrada se realiza por desplazamiento entre transformadas cuya longitud de entrada son entre 2048 o 256 muestras. La longitud de 256 muestras para codificar señales transitorias fue seleccionada como la mejor relación entre la resolución de frecuencia y la eliminación del efecto del pre-eco para velocidades de datos alrededor de los 64 kilobit/s por canal.

La conmutación de bloques de transformación es una efectiva herramienta para adaptar la resolución de tiempo-frecuencia del banco de filtros pero potencialmente genera un problema de sincronización de bloque entre los diferentes canales que se están codificando. Si uno de los canales usa una longitud de 2048 y durante el mismo intervalo de tiempo otro canal usa una longitud de 256, los bloques largos siguientes presentarán un problema de alineamiento. Esta falta de alineamiento entre canales es indeseable ya que esto crea problemas en la combinación de canales durante la codificación y la generación del *bitstream* de salida. Este problema de alineamiento de bloque entre cada canal ha sido solucionado por el sistema MPEG AAC.

Durante las transiciones entre las transformadas largas y cortas se utiliza una ventana puente de inicio y fin que conserva las propiedades de cancelación del *aliasing* en el dominio del tiempo de las transformadas MDCT e IMDCT y mantiene el alineamiento de los bloques. Estas transformadas de tipo puente designan secuencias de inicio y fin, respectivamente. La transformada convencional larga con longitud de 2048 muestras es denominada como una secuencia larga, mientras que las transformadas cortas ocurren en grupos llamadas secuencias cortas. La secuencia corta está compuesta de un arreglo de 8 bloques cortos, que permiten solaparse hasta un 50% y tiene la mitad de la transformada en los límites de secuencia para solaparse con las formas de ventana de inicio y fin. Este solapamiento de secuencias y la agrupación de bloques de transformación de inicio, fin, secuencias largas y cortas, se pueden apreciar en la Figura 4.4.

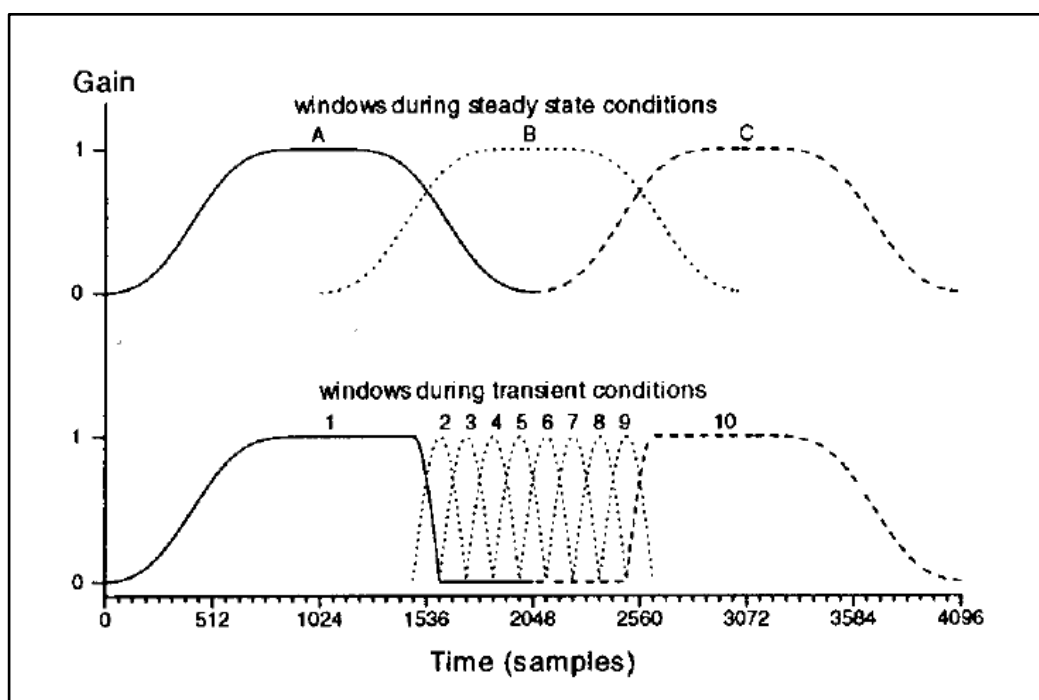


Figura 4.4. Proceso de conmutación de bloques para MPEG AAC [30]

En la Figura 4.4. se puede apreciar el proceso de solapamiento de ventana apropiado para las condiciones tanto del estado estacionario como para el estado transitorio. Las curvas A, B y C representan este proceso cuando no se usa la

conmutación de bloque y todas las transformadas tienen una secuencia de 2048 muestras y son compuestas de secuencias largas únicamente.

La parte inferior de la Figura 4.4. indica el uso de la conmutación de bloques para una transición sin problemas desde y hacia la transformación de la muestra más corta que está presente en la región entre el número de muestras desde 1600 hasta 2496. La Figura 4.4. además muestra que las transformadas de longitud corta (#2 - #9) son agrupadas dentro de una secuencia de 8 transformadas con solapamiento del 50% de longitud de 256 muestras y empleando una ventana de función seno con una apropiada longitud. La secuencia de inicio (#1) y la secuencia de fin (#10) permiten una transición sin errores entre transformadas cortas y largas.

4.8.5. Predicción

La predicción puede ser usada en el esquema de codificación AAC para mejorar la reducción de redundancia. La predicción es especialmente efectiva en el caso de señales que presentan fuertes componentes estacionarias y alta demanda de velocidad de datos.

Para cada canal, la predicción es aplicada a los componentes espectrales que resultan de la descomposición espectral, a la salida del banco de filtros. Para cada componente espectral sobre los 16 kHz, existe un respectivo predictor, teniendo como resultado un banco de predictores, donde cada predictor maneja la autocorrelación entre los valores de los componentes espectrales de las tramas consecutivas. Si se activa la predicción, el cuantificador se alimentará con el error de predicción en lugar del componente espectral original, resultando una alta eficiencia de codificación.

El predictor es implementado usando una estructura reticular en donde los elementos se encuentran conectados en cascada. Los parámetros del predictor son adaptados a las estadísticas de la señal actual sobre una base de trama por trama, usando para aquello el algoritmo de adaptación LMS-based (*Least Mean Square*).

Para poder garantizar que la predicción sea únicamente usada para el incremento de la ganancia de codificación, es necesario el uso de un control de predicción apropiado y una pequeña cantidad de información de control que debe ser transmitida al decodificador. Para el control de predicción, los predictores son agrupados dentro de las bandas de los factores de escala. La información del control de predicción para cada trama es determinada en dos pasos. Primeramente, para cada banda del factor de escala se determina si la predicción brinda o no una ganancia de codificación y como consecuencia del mismo, todos los predictores que pertenecen a una banda del factor de escala se activan o desactivan. A continuación, se determina si la predicción en la trama actual genera suficiente ganancia de codificación adicional para justificar los *bits* adicionales necesarios para la información del predictor. De lo contrario, si la predicción no se utiliza en la trama actual y sólo se transmite un *bit* de información para comunicar que no se está usando la misma.

Con el fin de aumentar la estabilidad de los predictores, se aplica un mecanismo de reinicio cíclico en el codificador y el decodificador, en el cual todos los predictores son inicializados nuevamente durante un cierto tiempo. Todo el conjunto de predictores está subdividido en treinta grupos de reinicio, los cuales son periódicamente inicializados. El mecanismo de reinicio está controlado por un *bit* de encendido y apagado, el cual siempre debe ser transmitido tan pronto como la predicción esté habilitada y un conjunto de 5 *bits* condicionales que especifican el grupo de predictores que deben ser reiniciado.

4.8.6. Cuantificación y codificación

El principal objetivo de las etapas de cuantificación y codificación es cuantificar los datos espectrales de tal manera que el ruido de cuantificación satisfaga las demandas del modelo psicoacústico. Al mismo tiempo, el número de *bits* necesarios para la codificación de la señal cuantificada desde estar por debajo de un cierto límite, normalmente el promedio de *bits* disponibles para un bloque de datos de audio. Este valor depende de la frecuencia de muestreo y de la velocidad de datos deseada. En AAC, el mecanismo del “*bit* reservado” entrega la posibilidad de influir en la distribución de *bits* entre bloques de audio consecutivos para un régimen de

tiempo corto. Estas dos restricciones, el cumplimiento de las demandas del modelo psicoacústico por una parte y mantener el número de *bits* asignados por debajo de un cierto número por otro lado, están vinculados a los principales retos del proceso de cuantificación.

No existe una estrategia estandarizada para un óptimo proceso de cuantificación, el único requerimiento es que el *bitstream* generado cumpla con los requerimientos de AAC.

A continuación, se indican las principales características del proceso de cuantificación de AAC:

- Cuantificación no uniforme.
- Codificación Huffman de los valores espectrales usando diferentes tablas.
- La conformación del ruido por la amplificación de grupos de valores espectrales (también llamados bandas de factores de escala). La información acerca de la amplificación se almacena en los valores de los factores de escala.
- Codificación de Huffman de los diferentes factores de escala.

La principal ventaja de un cuantificador no uniforme es la distribución del ruido dependiendo de la amplitud de los coeficientes. La relación señal a ruido en un cuantificador no uniforme se mantiene constante con un amplio rango de valores de energía de la señal. El rango de valores cuantificados está limitado a ± 8191 . Los coeficientes cuantificados son codificados usando codificación Huffman. Un método de codificación altamente flexible permite el uso de varias tablas Huffman para un determinado conjunto de datos espectrales, para lo cual están disponibles tablas de dos y cuatro dimensiones (con o sin signo). Para calcular el número de *bits* necesarios para codificar los datos cuantificados, se debe realizar el proceso de codificación y se debe calcular el número de *bits* necesarios para los datos espectrales.

El uso de un cuantificador no uniforme no es suficiente para cumplir las demandas psicoacústicas. Con el fin de cumplir estos requerimientos de la mejor manera, se debe modelar el ruido de cuantificación en unidades, de forma similar a las bandas críticas de sistema auditivo humano. Debido a que el sistema AAC ofrece una frecuencia de resolución relativamente alta para bloques largos de 23.43 Hz/línea a una frecuencia de muestreo de 48 kHz, es posible la creación de grupos de valores espectrales que reflejan muy de cerca el ancho de banda de las bandas críticas. En la Figura 4.5. se puede apreciar el ancho de las bandas de los factores de escala para bloques largos (el ancho de las bandas de los factores de escala está limitado a 32 coeficientes excepto para la última banda). El total del número de bandas de factores de escala para bloques largos a una frecuencia de muestreo de 48 kHz es de 49.

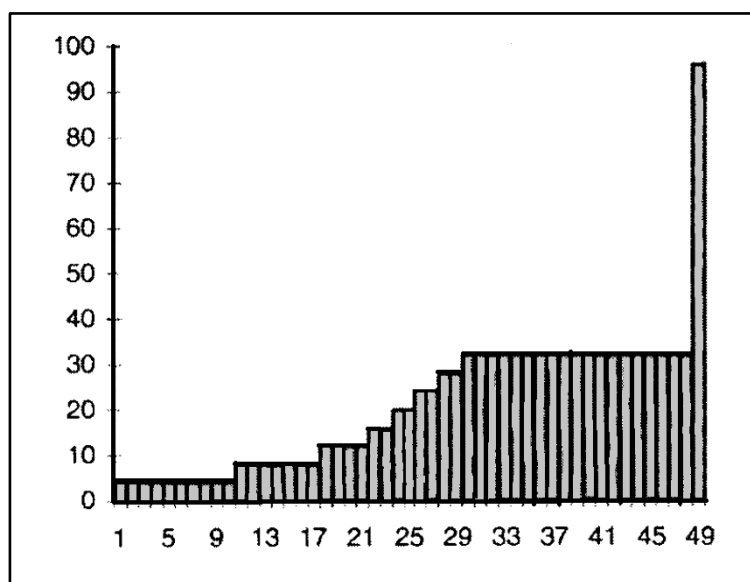


Figura 4.5. Número de coeficientes MDCT en cada SFB en bloques largos con frecuencia de muestreo de 48 kHz [30]

La amplificación del factor de escala inverso es aplicada en el decodificador. Por esta razón, la información de amplificación almacenada en los factores de escala en pasos de 1.5 dB es transmitida al decodificador. El primer factor de escala representa el tamaño del cuantificador global y el mismo es codificado en un valor PCM denominado como ganancia global (*global gain*). Los siguientes factores de escala son diferencialmente codificados usando una codificación especial de Huffman.

El método de iteraciones es uno de los más utilizados para realizar el tratamiento de ruido dentro del codificador, generando una buena de calidad de audio a la salida del codificador. El mismo consta de dos lazos anidados, un lazo interno y uno externo que son usados para determinar una óptima cuantificación. La tarea de repetición del lazo interno consiste en cambiar el tamaño de paso del cuantificador hasta que los datos espectrales puedan ser codificados con el número de *bits* disponibles. Para este propósito, se elige un tamaño de paso inicial de cuantificación, los datos espectrales son cuantificados y se cuenta el número de *bits* necesarios para codificar los datos cuantificados. Si este número es más alto que el número de *bits* disponibles, se incrementa el tamaño de paso del cuantificador y todo el proceso se repite. El lazo de iteración interno se puede apreciar en la Figura 4.6.

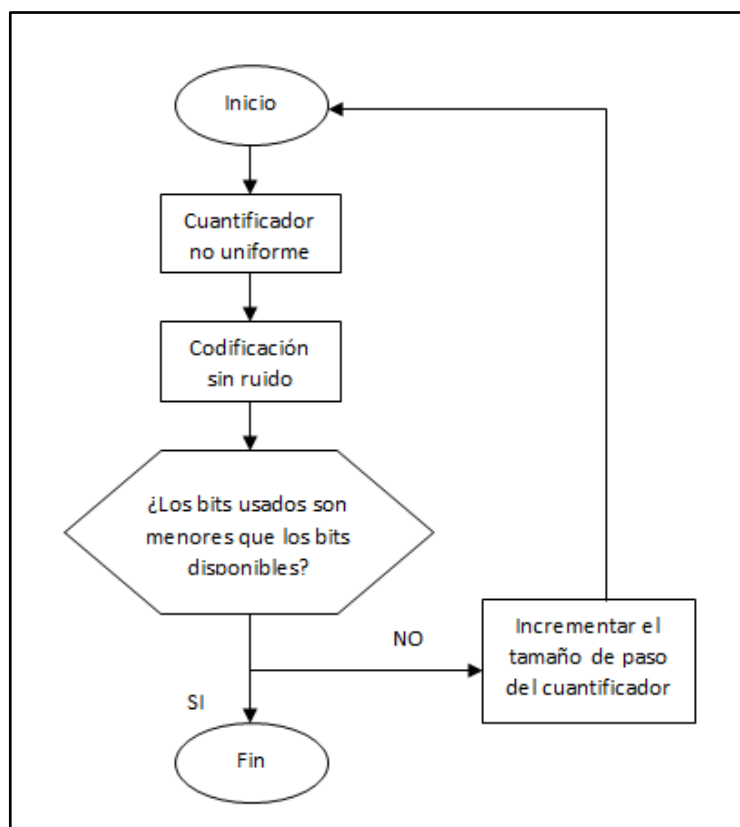


Figura 4.6. Diagrama de bloques de un lazo de iteración interno para el codificador MPEG AAC [30]

La tarea de repetición del lazo externo consiste en amplificar las bandas del factor de escala (*Scale Factor Bands*) de tal manera que se cumplan, de la mejor manera posible, las demandas del modelo psicoacústico.

- a. Al inicio, no se amplifica ninguna banda del factor de escala.
- b. Se llama al lazo interno para su operación.
- c. Para cada banda del factor de escala, se calcula la distorsión causada por la cuantificación.
- d. La distorsión actual es comparada con la distorsión permitida que se calcula con el modelo psicoacústico.
- e. Si el resultado es positivo, se almacena. Esto es importante, ya que el proceso de iteración no converge necesariamente.
- f. Se amplifican las bandas del factor de escala con una distorsión mayor a la permitida. En este punto, se pueden aplicar diferentes métodos para determinar las bandas del factor de escala que son amplificadas.
- g. Si todas las bandas del factor de escala son amplificadas, el proceso de iteraciones se detiene. El mejor resultado es almacenado.
- h. Si no existe ninguna banda de factor de escala con una distorsión real por encima de la distorsión permitida, el proceso de iteración se detendrá.
- i. De lo contrario el proceso se repetirá con nuevos valores de amplificación.

El lazo de iteración externo se puede apreciar en la Figura 4.7. El proceso de cuantificación y codificación para bloques cortos es similar que para bloques largos, pero se debe tomar en cuenta la agrupación y el intercalado.

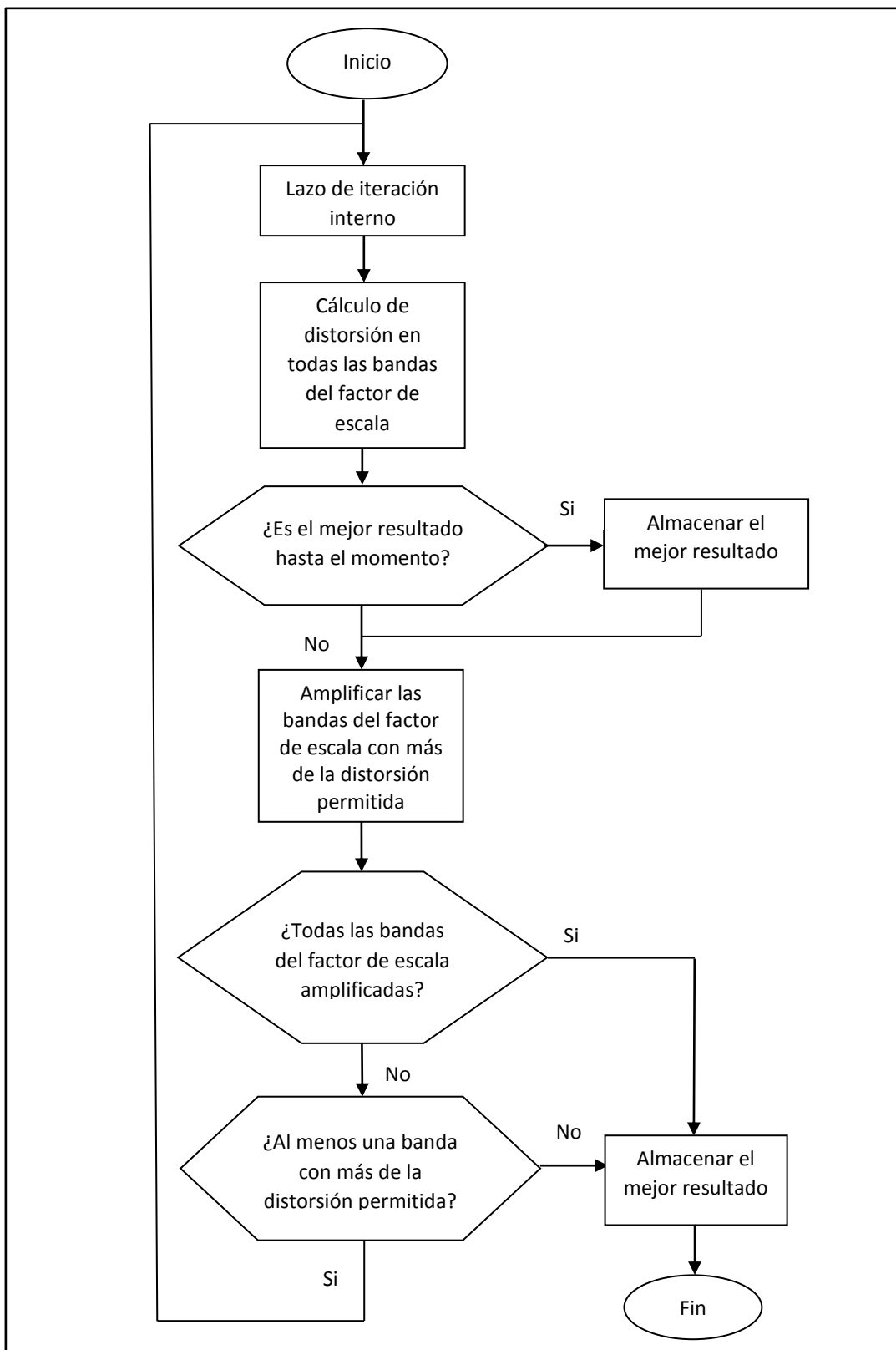


Figura 4.7. Diagrama de bloques de un lazo de iteración externo para el codificador MPEG AAC [30]

4.8.7. Codificación sin ruido (*Noiseless coding*)

La entrada al módulo de codificación sin ruido es un conjunto de 1024 coeficientes espectrales cuantificados. Como primer paso, se puede aplicar al espectro un método de compresión de rango dinámico sin ruido. Hasta cuatro coeficientes pueden ser codificados de forma separada con un valor de ± 1 en un arreglo de coeficientes cuantificados para transportar la señal. Los coeficientes son codificados como magnitudes enteras y se tiene un desplazamiento desde la base del arreglo de coeficientes para marcar la localización. Debido a que la información lateral para transportar los coeficientes hace uso de algunos *bits*, esta compresión sin ruido es aplicada sólo si la misma representa un ahorro neto de *bits*.

Los 1024 coeficientes espectrales cuantificados forman secciones para la codificación sin ruido, de tal manera que los mismos puedan ser codificados mediante un solo libro de códigos Huffman. Debido a razones en cuanto a la eficiencia de codificación, los límites de la sección sólo pueden ser los mismos límites de la banda del factor de escala, de modo que para cada sección del espectro se debe transmitir la longitud de la sección, en términos del número de las bandas del factor de escala que abarca, y el número de libro de códigos Huffman utilizada para la sección.

El seccionamiento es dinámico y por lo general varía de un bloque a otro, por lo tanto el número de *bits* necesarios para representar todo el conjunto de coeficientes espectrales cuantificados es minimizado. Esto se puede realizar usando un algoritmo de combinación comenzando con el número máximo posible de secciones, cada una de las cuales utiliza el libro de códigos de Huffman con el menor índice posible.

Las secciones eventualmente contienen sólo coeficientes cuyo valor es cero. Por ejemplo, si la entrada de audio se limita a la banda de 20 kHz o menor, entonces la mayoría de coeficientes son cero. Dichas secciones son codificadas con el libro de códigos de Huffman cero, que es un mecanismo de escape que indica que todos los coeficientes son cero y estos no requieren que ninguna palabra de código de Huffman sea enviada para esta sección.

Si la secuencia de la ventana consiste de 8 ventanas cortas, entonces el conjunto de 1024 coeficientes se trataría de una matriz de 8 por 128 coeficientes frecuenciales que representan la evolución en tiempo-frecuencia de la señal sobre la duración de las 8 ventanas cortas. Aunque el mecanismo de seccionamiento es suficientemente flexible para representar eficientemente las 8 secciones de cero, el agrupamiento y el intercalado proporciona una mayor eficiencia de codificación. Como se explicó anteriormente, los coeficientes asociados con ventanas cortas contiguas pueden ser agrupados para que compartan factores de escala entre todas las bandas del factor de escala dentro del grupo. Además, los coeficientes dentro de un grupo son intercalados intercambiando el orden de las bandas del factor de escala y las ventanas. De manera más específica, se asume que antes del intercalado el conjunto de 1024 coeficientes “C” se indexa como:

$$C [G] [W] [B] [K]$$

Dónde,

- G: es el índice de grupos.
- W: es el índice en las ventanas dentro de un grupo.
- B: es el índice en las bandas del factor de escala dentro de una ventana.
- K: es el índice en los coeficientes dentro de una banda del factor de escala.

Después del intercalado, los coeficientes son indexados como:

$$C [G] [B] [W] [K]$$

Esto tiene la ventaja de combinar todas las secciones cero debido a la limitación de banda dentro de cada grupo.

El espectro codificado utiliza un cuantificador por cada banda del factor de escala. El tamaño de paso de cada cuantificador está especificado como un conjunto de factores de escala y una ganancia global que normaliza estos factores de escala. Con el fin de aumentar la compresión, los factores de escala asociados con las bandas

del factor de escala que sólo tienen coeficientes de valor cero no son transmitidos. La ganancia global y los factores de escala son cuantificados en pasos de 1.5 dB. La ganancia global es codificada como un entero sin signo de 8 *bits* y los factores de escala se codifican diferencialmente con relación al valor anterior de la banda del factor de escala y luego codificadas mediante Huffman. El rango dinámico de la ganancia global es suficiente para representar todos los valores finales desde una fuente de audio de 24 *bit* PCM.

La codificación Huffman es usada para representar *n-tuples* de coeficientes cuantificados, con el código extraído de uno de 12 libros de códigos. Los coeficientes espectrales con *n-tuples* son ordenados en orden ascendente y el tamaño *n-tuple* es de dos o cuatro coeficientes. El máximo valor absoluto de los coeficientes cuantificados que puede ser representado por cada libro de códigos Huffman y el número de coeficientes en cada *n-tuple* para cada libro de código se muestra en la Tabla 4.3.

Existen dos libros de código para cada valor absoluto máximo, con cada uno representando una función de distribución de probabilidad distinta. Con el fin de ahorrar en almacenamiento del libro de códigos (una consideración importante en un decodificador producido en masa), la mayoría de libros de códigos representan valores sin signo. Para estos libros de código la magnitud de los coeficientes es codificación Huffman y el *bit* de signo de cada coeficiente no nulo se añade a la palabra de código.

Tabla 4.3. Libro de códigos Huffman para MPEG AAC [30]

Índice del Codebook	Tamaño n -Tuple	Máximo Valor Absoluto	Valores con Signo
0	4	0	SI
1	4	1	SI
2	4	1	NO
3	4	2	NO
4	2	2	SI
5	2	4	SI
6	2	4	NO
7	2	7	NO
8	2	7	NO
9	2	12	NO
10	2	12	NO
11	2	16 (ESC)	NO

El libro de código “0” indica que todos los coeficientes dentro de una sección son cero. El libro de código “11” puede representar los coeficientes cuantificados que tienen un valor absoluto mayor o igual a 16. Si la magnitud de uno o ambos coeficientes es mayor o igual que 16, se usa un mecanismo especial de codificación de salida para representar estos valores. La magnitud de los coeficientes está limitada a no más de 16 y los correspondientes 2 -tuple son codificadas por Huffman. Los *bits* de signo, según sea necesario, se añaden a la palabra de código. Para cada coeficiente de magnitud mayor o igual a 16, también se añade un código de escape, de la siguiente manera:

$$\text{Código de escape} = \langle \text{prefijo de escape} \rangle \langle \text{separador de escape} \rangle \langle \text{palabra de escape} \rangle$$

Dónde,

- $\langle \text{prefijo de escape} \rangle$: es una secuencia de N binarios de “1”.
- $\langle \text{separador de escape} \rangle$: es un “0” binario.
- $\langle \text{palabra de escape} \rangle$: es un *bit* entero sin signo $N+4$, primero MSB (*Most Significant Bit*) y N debe ser lo suficientemente grande para que la magnitud del coeficiente cuantificado sea igual a:

$$2^{(N+4)} + \langle \text{palabra de escape} \rangle$$

4.8.8. Multiplexación del *bitstream*

El sistema MPEG AAC tiene una sintaxis de *bitstream* muy flexible. Se definen dos capas: la primera capa especifica la información de audio “*raw*” mientras que la otra capa especifica el mecanismo específico de transporte de audio. La capa de datos está diseñada para que pueda realizar el procesamiento por sí misma, y de hecho es del todo suficiente para aplicaciones tales como la compresión para dispositivos de almacenamiento. La composición del *bitstream* se muestra en la Tabla 4.4.

Tabla 4.4. Estructura general del *bitstream* MPEG AAC [30]

<stream>	{<transport>}<block>{<transport>}<block>...
<block>	[<prog_config_ele>]<audio_ele>[<audio_ele>][<coupling_ele>] [<data_ele>][<fill_ele>]<term_ele>

Los *tokens* en el *bitstream* se indican entre paréntesis angulares (< >). El *bitstream* se indica por el *token* <stream> y se tiene una serie de *tokens* <block> que contiene toda la información necesaria para decodificar 1024 muestras de frecuencia de audio. Además, cada *token* <block> comienza en un límite de *byte* con respecto al comienzo del primer <block> en el *bitstream*. Entre los *tokens* <block> puede existir información de transporte, indicado por el <transport>, siendo necesario para la sincronización o para el control de errores.

Debido a que el sistema AAC posee un *buffer* de datos que permite variar la velocidad de datos como sea requerido por la señal de entrada, la longitud de cada <block> no es constante. Para aquello, el *bitstream* AAC utiliza cabeceras de velocidad variable. Estas cabeceras están alineadas por bytes para permitir la edición del *bitstream* desde cualquier punto.

4.8.9. Audio Data Transport Stream (ADTS)

Existen una variedad de métodos para el empaquetamiento del *bitstream* con los datos de audio codificados para fines de almacenamiento o transporte. Para el caso de transporte, el más usado es el formato ADTS.

El formato ADTS se define como un formato de audio de empaquetamiento diseñado para el transporte de la información, puede ser usado por los formatos MPEG-TS o AAC.

El formato ADTS se estructura de la siguiente manera:

```

AAAAAAAA AAAABCCD EEEFFFGH HHIJKLMM MMMMMMMM
MMMOOOOO OOOOOOPP (QQQQQQQQ QQQQQQQQ)

```

Donde cada letra representa un *bit*, la descripción de cada uno de los *bits* se puede apreciar en la Tabla 4.5.

Tabla 4.5. Estructura de bits del formato ADTS

Letra	Longitud (bits)	Descripción
A	12	Palabra de sincronización, todos los <i>bits</i> siempre deben ser "1".
B	1	Versión MPEG: "0" para MPEG-4, "1" para MPEG-2.
C	2	Capa: siempre "0".
D	1	Ausencia de protección: "1" si la trama no tiene CRC o "0" si la trama tiene CRC.
E	2	Tipo de perfil de AAC.
F	4	Índice de frecuencia de muestreo (valor de 15 no está permitido).
G	1	<i>Bit</i> privado, se establece en "0" para la codificación.
H	3	Configuración de canales.
I	1	Autenticidad, se establece en "0" para la codificación.
J	1	Punto inicial, se establece en "0" para la codificación.
K	1	ID de derechos de autor, se establece en 0 para la codificación.
L	1	ID de inicio de autor, se establece en 0 para la codificación.
M	13	Longitud de trama, este valor debe incluir 7 o 9 bytes de longitud de la cabecera: $Frame\ Length = (Protection\ Absent == 1 : 7 ? 9) + \text{tamaño}\ (AAC\ Frame)$.
O	11	Tamaño del buffer
P	2	Número de tramas AAC (RDBs) en la trama ADTS menos 1, para una máxima compatibilidad se debe usar siempre una trama AAC por trama ADTS.
Q	16	CRC, si es que existe.

CAPÍTULO V

DISEÑO Y ESPECIFICACIONES DEL CODIFICADOR

5.2. Introducción

5.2.1. Propósito

El propósito de este capítulo es presentar una descripción del diseño y las funcionalidades del codificador desarrollado, el cual está dirigido a los responsables del proyecto y desarrolladores de *software*.

5.2.2. Ámbito del sistema

El *software* desarrollado se define como un codificador de audio MPEG-4 AAC con salida TS.

El aplicativo consta de una única versión realizada en el Entorno de Desarrollo Integrado (IDE) de MATLAB, empleando para aquello lenguaje de programación m.

Entre las principales funcionalidades del aplicativo, se pueden indicar las siguientes:

- Apertura exclusiva de archivos de audio de extensión *.wav, el mismo que es ampliamente usado como formato de fuente original, sin codificación.
- Obtención de una señal de audio de extensión *.aac, tras el proceso de codificación de audio (según la versión del sistema operativo Windows, el formato puede ser reconocido como ADTS).
- Obtención de una señal de audio de extensión *.ts, tras aplicar la conversión de formatos mediante el procedimiento establecido por FFMPEG, mediante la ejecución de librerías externas.
- Visualización de gráficos del tratamiento de la señal durante el proceso de compresión.

- Visualización del número de bloques que serán procesados durante el proceso de codificación.
- Visualización del valor de varios parámetros de codificación establecidos por AAC, como son los factores de escala, coeficientes MDCT, función ventana, entre otros.
- Integración del proceso de transformación de formato de audio (*.aac→*.ts) en MATLAB, mediante la asociación de un archivo *.bat al lenguaje de programación m.

5.2.3. Definiciones, acrónimos y abreviaturas

- MATLAB:** *Software* matemático que emplea IDE y un lenguaje de programación propio, conocido como lenguaje de programación m.
- AAC:** Formato de señal digital de audio basado en un algoritmo de compresión con pérdida, bajo la norma ISO/IEC 13818-7.
- ADTS:** Formato contenedor usado para el empaquetamiento de AAC, con fines de transporte.
- WAV:** Formato de audio digital normalmente sin compresión de datos.
- Factor de escala:** Parámetro del proceso de codificación, son básicamente los coeficientes MDCT con ciertos procesos matemáticos realizados para la cuantificación de valores y reducción de la entropía.
- Ventana:** Función matemática usada en el análisis y el procesamiento de señales para evitar discontinuidades en los bloques analizados.
- MDCT:** Transformada lineal ortogonal, basada en la idea de la cancelación del *aliasing* del dominio del tiempo.
- AAC-LC:** Perfil de codificación de AAC de baja complejidad.
- SFB:** Bandas del factor de escala, se definen como una agrupación de coeficientes MDCT, donde la señal es amplificada en ciertas regiones espectrales.
- TS:** Formato estándar de contenedores para la transmisión y el almacenamiento de audio, vídeo y datos. Se utiliza en sistemas de difusión de Televisión Digital.

- k) **FFMPEG:** Programa que contiene una colección de *software* libre para grabar, convertir (transcodificar) y realizar *streaming* de audio y vídeo.

5.3. Descripción General

5.3.1. Funciones del producto

El funcionamiento del aplicativo se puede dividir en varios bloques, los cuales se detallan a continuación:

5.3.1.1. Función de reconocimiento de la señal de audio de entrada

- Apertura exclusiva de archivos de audio de extensión *.wav como señal de entrada.
- Visualización de un mensaje de error cuando el formato de la señal de entrada no cumpla con los requisitos establecidos.
- Lectura de datos de la señal de audio de entrada.

5.3.1.2. Función de procesamiento de la señal de entrada

- Tratamiento de los datos obtenidos de la señal de entrada como preparación previa para el proceso de codificación.
- Formación matricial de N bloques con M cantidad de muestras.
- Despliegue de gráficos correspondientes a la señal de entrada.

5.3.1.3. Función de transformación de dominios

- Multiplicación de los bloques establecidos en el arreglo matricial con la función ventana para evitar discontinuidades al principio y al final de los mismos.
- Aplicación de la MDCT con el fin de transformar la señales del dominio temporal al dominio espectral (descomposición espectral).

- Despliegue de gráficos correspondientes al proceso.

5.3.1.4. Función de obtención de los coeficientes MDCT y factores de escala

- Obtención de los coeficientes de la MDCT tras el proceso de transformación establecido.
- Obtención de los factores de escala y formación de las bandas de factor de escala en base a los coeficientes de la MDCT.
- Cuantificación de los coeficientes de la MDCT.

5.3.1.5. Función de codificación

- Codificación de los coeficientes de la MDCT mediante codificación sin ruido (*Noiseless Coding*).
- Codificación de datos mediante libros de códigos de Huffman.
- Asignación directa de los códigos de Huffman en base a un libro de códigos establecido.

5.3.1.6. Función de formación del *bitstream*

- Formación del *bitstream* en base a los parámetros de codificación calculados en las funciones y procesos anteriores.
- Aplicación del formato contenedor ADTS para la formación del *bitstream* con fines de transporte.
- Obtención de un archivo de audio codificado con extensión *.aac.

5.3.1.7. Función de transformación de formato

- Transformación de formato *.aac a *.ts mediante el uso de la herramienta FFMPEG.
- Configuración de librerías externas en FFMPEG para el tratamiento de archivos de audio basados en el formato AAC.

- Asociación de archivos *.bat al lenguaje de programación m usado por MATLAB.
- Obtención de archivo de audio con extensión *.ts.

5.3.2. Perspectiva del producto

El aplicativo desarrollado funciona como un sistema independiente, con ciertos requisitos de funcionamiento y abierto a mejoras en cuanto al proceso de codificación establecido para AAC. El mismo tiene interacción con programas externos, como es el caso de FFMPEG para la transformación del formato de audio, y puede ser ejecutado desde cualquier ordenador que cumplan los requisitos que se detallarán durante el desarrollo del capítulo.

5.3.3. Características de los usuarios

Tabla 5.1. Características del usuario “Usuario”

Tipo de Usuario	Usuario.
Formación	Formación superior técnica con conocimientos de formatos de codificación de audio y TV Digital.
Habilidades	Manejo del protocolo <i>Transport Stream</i> y MATLAB.
Actividades	Uso del codificador.

Tabla 5.2. Características del usuario “Desarrollador de Software”

Tipo de Usuario	Desarrollador de Software.
Formación	Formación superior técnica con ciertos conocimientos de formatos de codificación MPEG-2 y MPEG-4 y los diferentes perfiles dados por AAC. Conocimientos de TV Digital y uso de herramientas de transformación de formatos como FFMPEG.
Habilidades	Manejo del protocolo <i>Transport Stream</i> , MATLAB y FFMPEG.
Actividades	Desarrollo del codificador.

5.3.4. Restricciones

- El aplicativo recibe únicamente como señal de entrada archivos de audio con extensión *.wav, bajo los siguientes criterios: frecuencia de muestreo de 44100 Hz, 16 *bits* por cada muestra y un solo canal (mono).
- El aplicativo está diseñado bajo los criterios de codificación del perfil de baja complejidad (AAC-LC), cuya restricción se basa en que la calidad del audio de salida podría verse reducida en comparación con el perfil principal de AAC.
- El aplicativo únicamente procesa bloques largos, no se consideran bloques conmutados (bloques cortos), para el cual se requiere un procesamiento diferente con funciones ventanas especiales.
- No se considera el bloque de modelo psicoacústico (presente en el modelo de referencia).
- No se considera el bloque TNS (presente en el modelo de referencia).
- No se considera el bloque de codificación M/S, ya que se trabaja con un solo canal (mono).
- No se hace uso de una tasa de control de *bits*.
- Se hace uso del formato contenedor ADTS, mismo que es ampliamente usado para el transporte de tramas bajo la norma AAC.
- El aplicativo emplea algunos valores forzados para la codificación, debido a que únicamente se trabaja con frecuencias de muestreo de 44100 Hz y un solo canal, además de no tener un modelo psicoacústico de referencia.
- La eficiencia de procesamiento de los bloques de audio, dependerá de las capacidades del ordenador donde se ejecute el programa.

5.3.5. Requisitos futuros

- El aplicativo está abierto a mejoras en cuanto a la agregación de una mayor cantidad de bloques según el diagrama general del proceso de codificación dado por AAC, que mejoren el proceso de codificación y generen un archivo de audio de mayor calidad.

- El aplicativo debe ser igualado o superado a los perfiles de mayor eficiencia dados por AAC, los cuales hacen uso de la PNS que mejora el tratamiento de ruido durante el proceso de codificación.
- Se debe extender la funcionalidad del aplicativo a niveles de *hardware*, con el objetivo principal de diseñar equipamiento propio, en especial para los estándares de televisión digital como ISDB-Tb, que hace uso de AAC como técnica de codificación de audio.
- Según las necesidades del desarrollador, se podría continuar con la implementación a nivel de *software* para la transformación de formatos en el mismo lenguaje de programación dado por MATLAB.

5.4. Requisitos específicos

5.4.1. Interfaces

5.4.1.1. Interfaz de usuario

La interfaz de usuario para el aplicativo será el *Command Window* de MATLAB, donde el programa deberá ser ejecutado una vez configurado el archivo de sonido con extensión *.wav desde el archivo M.

Durante la ejecución del mismo, se podrá observar el tratamiento de los bloques generados para el proceso de codificación, como se indica en la Figura 5.1.

```

*****
***** CODIFICADOR DE AUDIO AAC -> TS *****
*****

*****
***** CODIFICACIÓN *****
*****

...Bloque # 1...
...Bloque # 2...
...Bloque # 3...
...Bloque # 4...
...Bloque # 5...
...Bloque # 6...
...Bloque # 7...
...Bloque # 8...
...Bloque # 9...
...Bloque # 10...
...Bloque # 11...
...Bloque # 12...
...Bloque # 13...
...Bloque # 14...
...Bloque # 15...
...Bloque # 16...
...Bloque # 17...
...Bloque # 18...
...Bloque # 19...
...Bloque # 20...
...Bloque # 21...
...Bloque # 22...
...Bloque # 23

```

Figura 5.1. Tratamiento de bloques durante el proceso de codificación

Al finalizar el tratamiento de los bloques, se procederá con la conversión del formato de archivo de audio mediante el uso de la herramienta FFMPEG, para el cual el programa indicará los detalles de la transformación.

El archivo de audio con formato *.ts, se encontrará en la carpeta de ubicación del programa de MATLAB ejecutado.

5.4.1.2. Interfaces de hardware

El ordenador donde se ejecute el aplicativo deberá tener las siguientes características de *hardware*:

- Procesador Intel CORE i3, similar o superior.
- Memoria RAM mínima de 4 GB para un nivel aceptable de procesamiento.

5.4.1.3. Interfaces de software

El ordenador donde se ejecute el aplicativo deberá tener las siguientes características de *software*:

- MATLAB R2009a o versiones superiores (S.O. Windows, Linux o MAC OS).
- FFMPEG versión N-69422 o superior.
- FAAD v2.7 o superior (para el proceso de decodificación si es requerido por el usuario y/o desarrollador).
- Librerías externas de FFMPEG como es el caso de “*libfdk_aac*”, “*libfaac*” y “*libfaad*”, en sus versiones más recientes, las mismas que son usadas para el proceso de conversión y en varias pruebas de reconocimiento del formato AAC en sus diferentes perfiles.

5.5. Diseño del codificador

El codificador de audio fue diseñado bajo la estructura básica de un codificador MPEG bajo las normativas dadas por AAC en su perfil AAC-LC, siguiendo para aquello el diagrama general del mismo. El aplicativo hace uso de una asociación externa con el programa de conversión FFMPEG, con el objetivo de obtener en su salida un archivo de audio TS.

5.5.1. Señal de entrada del codificador

La entrada del codificador fue diseñada para recibir una señal de audio en formato *.wav con frecuencia de muestreo de 44100 Hz, 16 *bits* por cada muestra y un solo canal (mono). En la entrada del codificador se tendrá un previo análisis para que la señal cumpla con los requisitos establecidos. El diagrama de flujo de este proceso puede ser apreciado en la Figura 5.2.

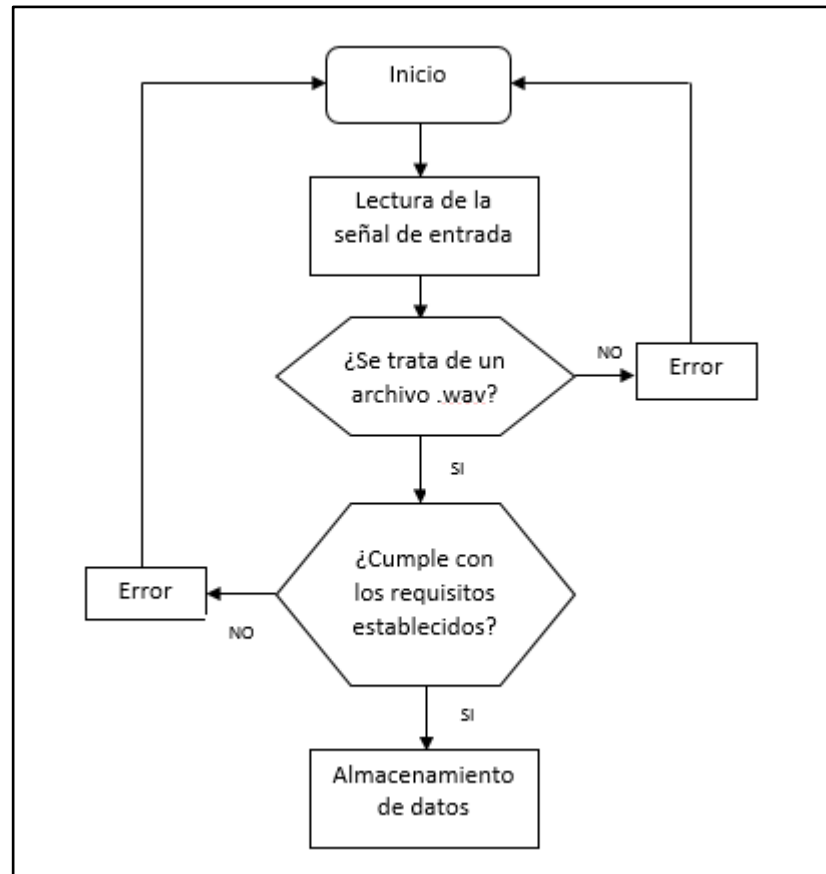


Figura 5.2. Diagrama de flujo para la señal de entrada del codificador

5.5.2. Tratamiento de la señal de entrada

Una vez obtenidos los datos de la señal de audio de entrada, los mismos son tratados para formar los bloques que serán procesados durante la codificación. En este caso, se generan N bloques largos, cada uno con 1024 muestras, como se indica en la Figura 5.3.

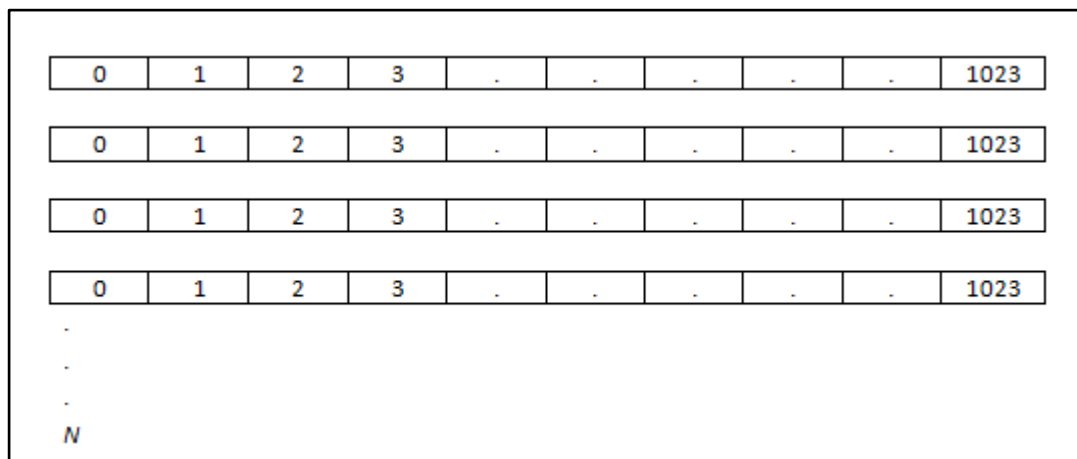


Figura 5.3. Formación matricial de bloques largos de dimensión $N \times 1024$

5.5.3. Transformación de dominios y descomposición espectral

Cada uno de los N bloques obtenidos en el proceso de tratamiento previo, serán procesados de forma independiente, donde cada uno será multiplicado por la función ventana, con el fin de evitar las discontinuidades al inicio y final de cada uno de los bloques. A la señal obtenida tras el proceso realizado con la función ventana, se le aplicará la MDCT, con lo cual se obtendrá la señal en el dominio espectral. El proceso descrito puede resumirse en la Figura 5.4.

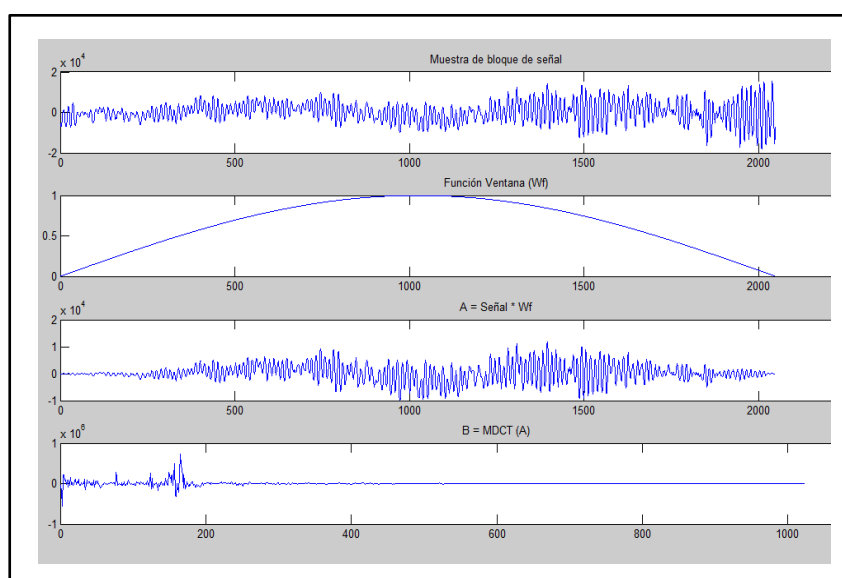


Figura 5.4. Descomposición espectral de la señal de un bloque de muestra mediante el uso de la función ventana y la MDCT

5.5.4. Cálculo de coeficientes de la MDCT, factores de escala, bandas del factor de escala y proceso de cuantificación

Tras aplicar el proceso de la MDCT, se proceden a obtener los coeficientes de la transformada para cada uno de los bloques analizados, los mismos básicamente son determinados como los factores de escala. Esta elección viene determinada debido a que el sistema no posee un modelo psicoacústico de referencia, mismo que interviene en el cálculo de los valores de los factores de escala para el tratamiento de ruido durante el proceso de codificación.

Además, se procede con la asignación de un valor máximo permitido de bandas de factor de escala que agruparán a los coeficientes de la MDCT, con el fin que los mismos puedan ser amplificados en ciertas regiones espectrales para aumentar la relación señal a ruido y se pueda lograr una codificación eficiente.

Los factores de escala serán los encargados de realizar el control de la cantidad de error permitida en cada una de las bandas del espectro.

Posteriormente, se procede a calcular el valor promedio de los coeficientes de la MDCT en base al número de bandas del factor de escala, ya que según el criterio de diseño del codificador, se usará el valor promedio para el cálculo real de las bandas del factor de escala y para el proceso de cuantificación de los coeficientes.

5.5.5. Proceso de codificación

Para el proceso de codificación se emplea la técnica de Huffman, o también conocida como codificación de entropía.

Los libros de códigos de Huffman para el proceso de codificación, serán aplicados de forma directa, sin calcular el número de *bits*, debido a que el sistema no posee un control para el *bitrate*.

En la Tabla 5.3. se puede apreciar el libro de códigos de Huffman usado para el proceso de codificación en el aplicativo diseñado.

Tabla 5.3. Libro de códigos Huffman del codificador diseñado

Índice del <i>Codebook</i>	Tamaño <i>n-Tuple</i>	Máximo Valor Absoluto	Valores con Signo
0	4	0	-
1	4	1	SI
2	4	1	SI
3	4	2	NO
4	4	2	NO
5	2	4	SI
6	2	4	SI
7	2	7	NO
8	2	7	NO
9	2	12	NO
10	2	12	NO
11	2	16 (ESC)	NO

Según lo observado en la Tabla 5.3., el índice del libro de códigos Huffman será generado para representar *n-tuples* de coeficientes cuantificados, donde los coeficientes espectrales con *n-tuples* son ordenados de forma ascendente con tamaños de dos o cuatro.

Cabe informar que cualquier valor de coeficiente cuantificado mayor a 12, será automáticamente asignado al valor de 12, y lo mismo para los valores menores a -12, que de igual manera serán asignados con el valor de -12. Esto debido a que el codificador diseñado no posee un mecanismo especial de codificación de salida para valores iguales o superiores a 16.

5.5.6. Formación del *bitstream*

Una vez se ha realizado el proceso de codificación, se procederá con la formación del *bitstream* en base al formato contenedor ADTS. El formato ADTS contiene la información de codificación AAC, el cual es usado para poder transportar

la trama por un medio de transmisión o también se define como un paquete elemental (PES) para el formato TS.

Para la formación de la cabecera ADTS, se toma como referencia la estructura de los *bits* indicados en la Tabla 4.5. como ejemplo, se pueden citar las estructuras de cabecera ADTS en la Figura 5.5. para las versiones MPEG-4 y MPEG-2 respectivamente.

0xFF 0xF1 0x50 0x40 0x16 0xDF 0xFC
0xFF 0xF9 0x50 0x40 0x16 0xDF 0xFC

Figura 5.5. Estructura de cabeceras ADTS para las versiones MPEG-4 y MPEG-2 en hexadecimal

La estructura del *bitstream* generado se puede apreciar en la Tabla 5.4.

Tabla 5.4. Estructura del bitstream para una trama AAC en formato ADTS

Cabecera ADTS fija		
Campo	# bits	Detalles
Palabra de sincronismo	12	Todos los <i>bits</i> siempre deben ser "1".
Versión MPEG	1	"0" para MPEG-4, "1" para MPEG-2.
Capa	2	Siempre "0".
Presencia de protección	1	"1" si la trama no tiene CRC o "0" si la trama tiene CRC.
Perfil de AAC	2	Tipo de perfil usado.
Frecuencia de muestreo	4	Valor de 15 no está permitido.
Bit de reserva	1	Se establece en "0" para la codificación.
Configuración de canales	3	Número de canales empleados.
Autenticidad	1	Se establece en "0" para la codificación.
Punto inicial	1	Se establece en "0" para la codificación.

Cabecera ADTS variable		
Campo	# bits	Detalles
ID derechos autor	1	Se establece en "0" para la codificación.
ID de inicio de autor	1	Se establece en "0" para la codificación.
Longitud de trama	13	Debe incluir 7 o 9 bytes de longitud de la cabecera.
Tamaño del buffer	11	Tamaño del <i>buffer</i> .
Número de tramas AAC	2	Se debe usar siempre una trama AAC por trama ADTS.
Otros valores		
SCE (LEN_SE_ID)	3	Se considera como "0" si se emplea un solo canal.
SCE (LEN_TAG)	4	Se considera como "0" si se emplea un solo canal.
Ganancia global	8	Se puede o no tener ganancia global.
Reserva ICS	1	Se relaciona a la ganancia global.
Tipo de ventana	2	Información del tipo de ventana usado.
Forma de ventana	1	Información de la forma de la ventana usada.
Número de bandas de factor de escala	6	Información del número de bandas de factor de escala usado para la codificación.
Bandera de predicción global	1	Se define nulo si no se emplea algún tipo de predicción.
Libros de códigos usado para todas las sub bandas	N/D	Valor de <i>bits</i> variable, depende del uso de <i>bits</i> usado por cada banda para la codificación.
Datos de pulsos	1	Valor nulo si no se tiene.
Datos TNS	1	Valor nulo si no se tiene.
Datos de control de ganancia	1	Valor nulo si no se tiene.
Datos espectrales	N/D	Valor de <i>bits</i> variable.
ID de finalización	3	Identificativo de finalización.
Alineamiento de byte	N/D	Valor de <i>bits</i> variable, depende de la longitud del <i>bitstream</i> presente al momento.
CRC ADTS		
CRC	16	Puede o no estar presente y depende del valor de un <i>bit</i> de la cabecera.

5.5.7. Conversión de formato de audio

Tras la obtención del *bitstream* en formato AAC con encapsulamiento ADTS, se procede con la conversión del formato hacia TS, que representará el formato de salida del codificador de audio diseñado.

Para el proceso de conversión, se hace uso del programa FFMPEG, el cual contiene un conjunto de librerías de codificación y decodificación para diferentes formatos de audio, entre ellos para el formato AAC.

Tras el proceso de instalación de FFMPEG, se procede con la ejecución del comando que se indica en la Figura 5.6., el mismo que será capaz de transformar el formato de audio AAC obtenido al formato TS requerido.

```
ffmpeg -i audio_codificado.aac salida.ts
```

Figura 5.6. Comando del programa FFMPEG empleado para la conversión de audio del formato AAC a TS

Debido a que la ejecución de los comandos de FFMPEG se lo realiza desde el Símbolo del Sistema (CMD) para Windows o desde un terminal para Linux, se procede a asociar, para el caso de Windows, un archivo ejecutable por lotes (*.bat) al lenguaje de programación de MATLAB. Por tanto, el archivo de extensión *.bat, contendrá el comando de conversión de audio, indicado en la Figura 5.6., y el mismo será asociado al lenguaje de programación de MATLAB, mediante el comando indicado en la Figura 5.7.

```
dos('archivo.bat')
```

Figura 5.7. Comando de MATLAB para ejecución de archivos BATCH

De esta manera, se ha procedido a generar una asociación entre MATLAB y FFMPEG para el proceso de conversión de formatos de audio, con el fin de cumplir los requisitos del codificador diseñado.

5.6. Diagramas

5.6.1. Diagrama general del codificador

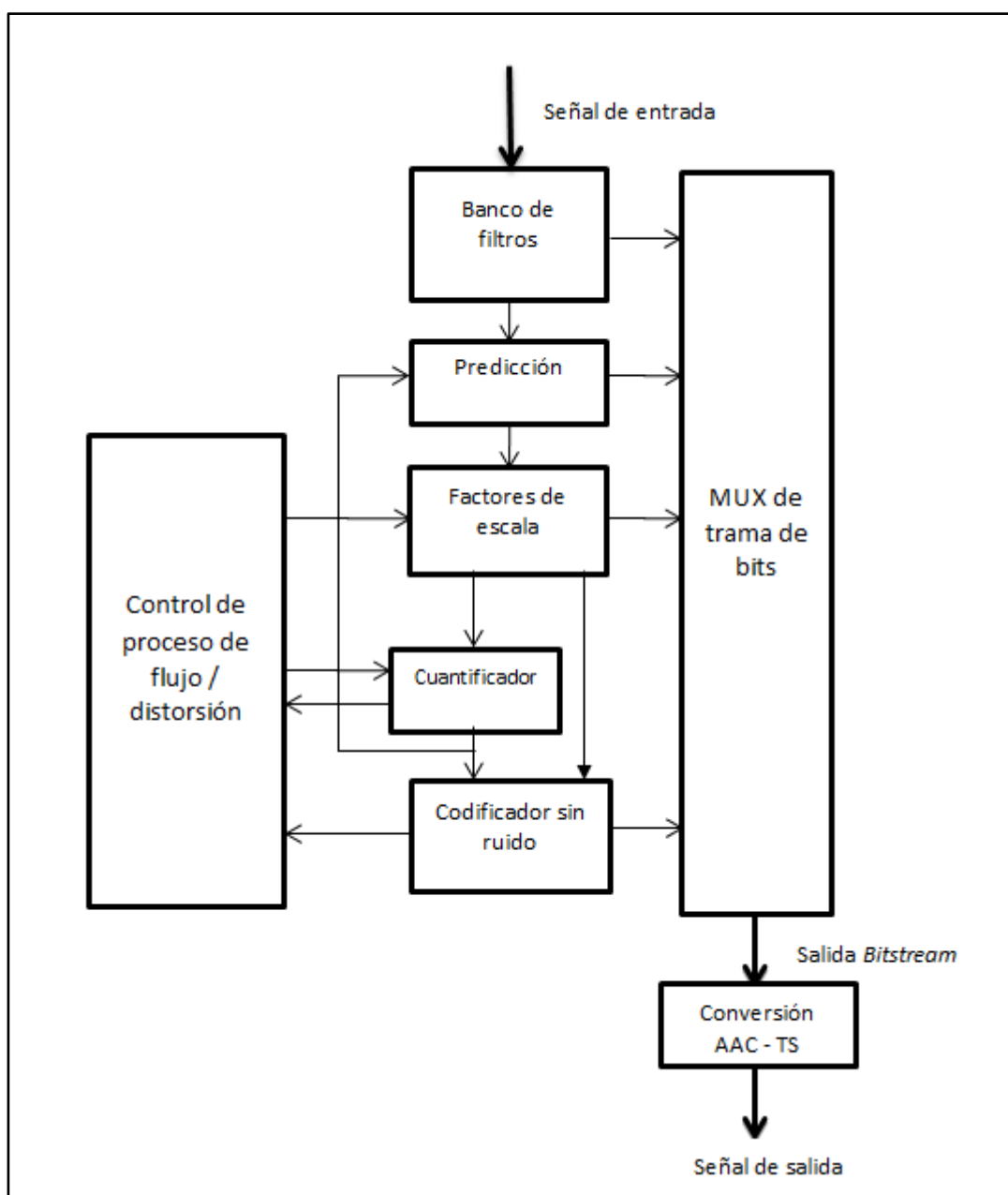


Figura 5.8. Diagrama general del codificador diseñado

5.6.2. Diagrama de flujo del codificador

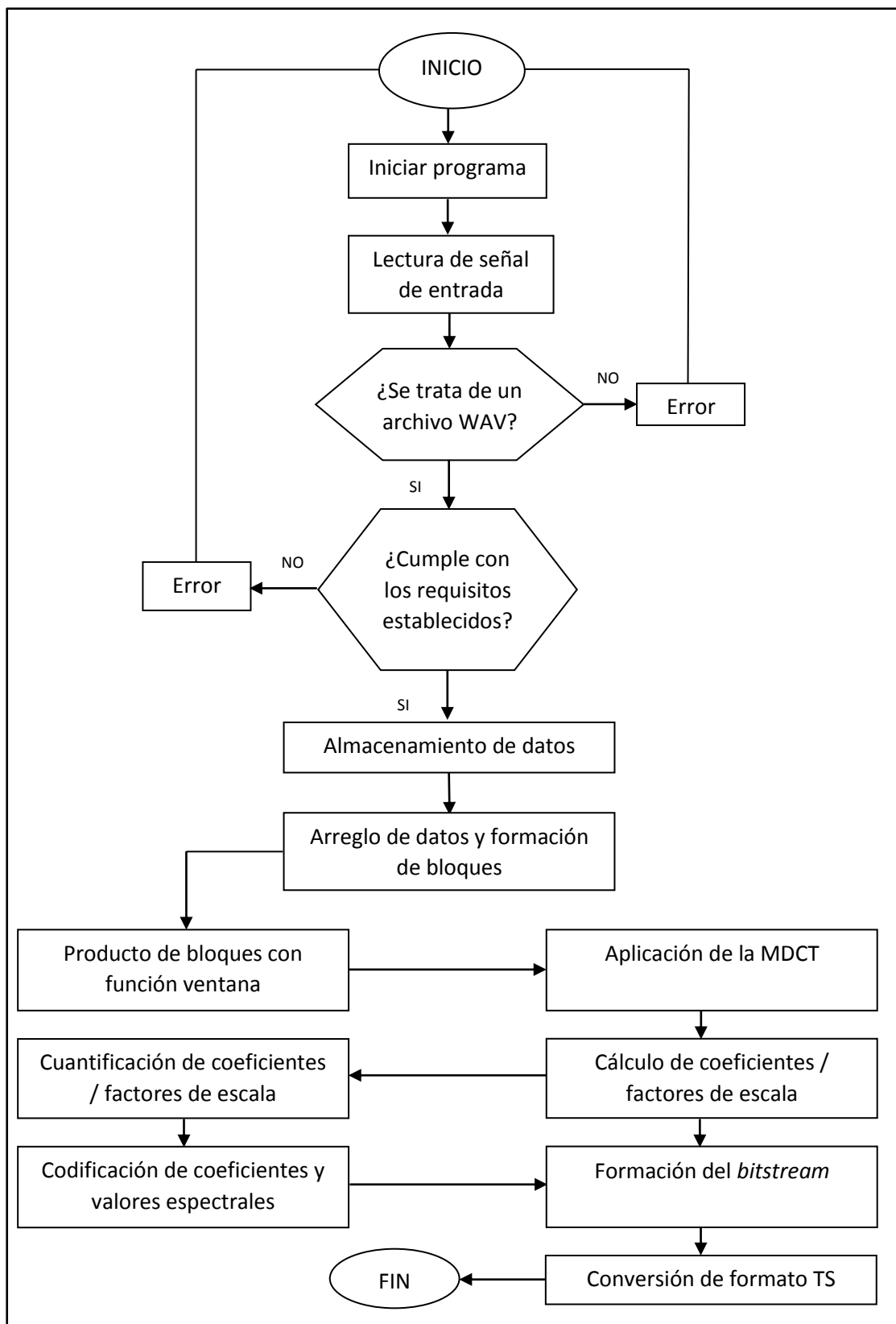


Figura 5.9. Diagrama de flujo del codificador diseñado

CAPÍTULO VI

PRUEBAS Y RESULTADOS

6.2. Introducción

En este capítulo se detallan las diferentes pruebas realizadas sobre el codificador diseñado, además de los resultados obtenidos.

6.3. Generación de archivo de audio AAC

En primera instancia, se procederá con la obtención del archivo de audio codificado con extensión AAC y formato contenedor ADTS. Para aquello se utilizará como prueba un archivo de audio WAV, el mismo se detalla en la Tabla 6.1.

Tabla 6.1. Características del audio de entrada

Archivo de entrada	Tamaño de archivo	Frecuencia de muestreo	Número de canales	Número de bits
audio_entrada.wav	1649 kilobytes	44100 Hz	1	16

Tras el proceso de codificación establecido, se procede con la obtención del archivo de audio requerido, el mismo que posee las características dadas en la Tabla 6.2.

Tabla 6.2. Características del audio codificado

Archivo de salida	Tamaño de archivo	Frecuencia de muestreo	Número de canales	Perfil
audio_codificado.aac	279 kilobytes	44100 Hz	1	LC (<i>Low Complexity</i>)

Tomando como referencia el tamaño de archivo del audio de entrada de prueba y el audio codificado, se puede determinar que se tiene una eficiencia de codificación del 83.08%.

6.4. Obtención de parámetros gráficos del proceso de codificación

Considerando el archivo de entrada de prueba dado en la Tabla 6.1., se obtienen los siguientes parámetros gráficos para el codificador diseñado.

- Señal de audio de entrada original y modificada

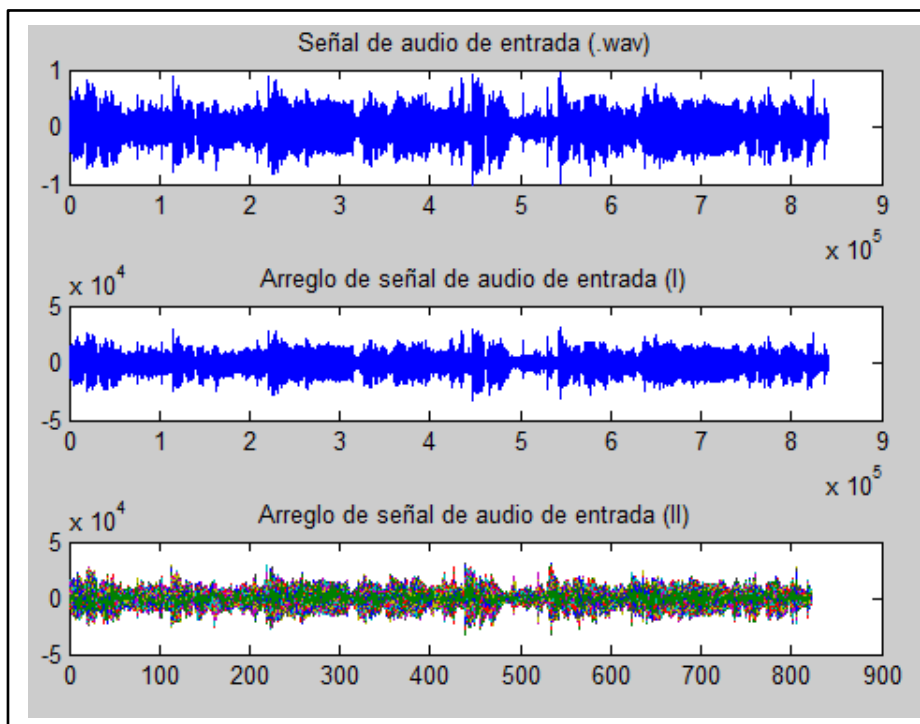


Figura 6.1. Señal de audio de entrada original y modificada

- **Análisis de un bloque de muestra con función ventana y la MDCT**

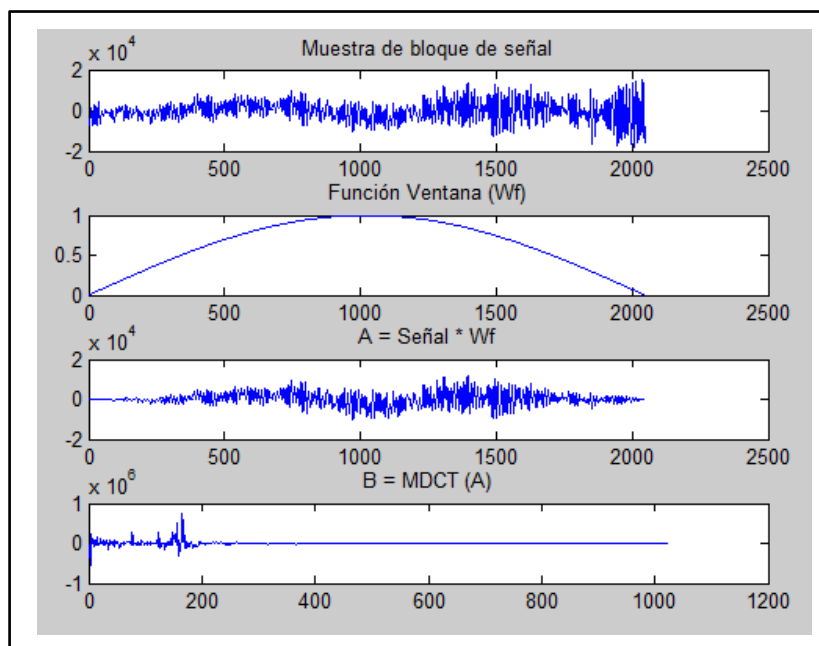


Figura 6.2. Muestra de bloque de la señal de entrada en el cálculo de la MDCT

- **Análisis de coeficientes MDCT para un bloque de muestra**

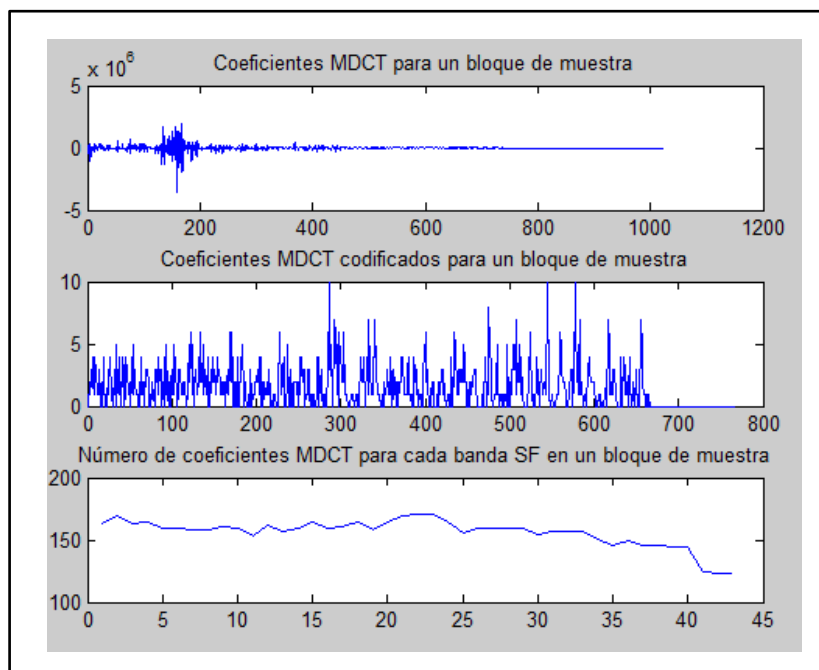


Figura 6.3. Análisis de coeficientes de la MDCT para un bloque de muestra

6.5. Análisis del formato contenedor ADTS

Al obtener el archivo de audio codificado en formato AAC, es posible verificar la estructura de la cabecera ADTS mediante diferentes programas especializados para dicho análisis. Para este caso, se hará uso de un programa de libre distribución denominado P23 [24], el mismo que permite el ingreso de los *bits* en hexadecimal que conforman la cabecera e indicará los parámetros configurados en la misma.

En la Figura 6.4. se pueden apreciar los detalles de la cabecera ADTS para el archivo de audio codificado obtenido en base al archivo de audio de pruebas.

Correct AAC frame header	
Bits	11111111111100010101000001000000000101101101111111111100
ID	MPEG-4
MPEG Layer	0
CRC checksum absent	1
Profile	Low Complexity profile (AAC LC)
Sampling frequency	44100
Private bit	0
Channel configuration	1
Original/copy	0
Home	0
Copyright identification bit	0
Copyright identification start	0
AAC frame length	182
ADTS buffer fullness	VBR
No raw data blocks in frame	0

Figura 6.4. Detalles de cabecera ADTS según el analizador P23

6.6. Análisis de la trama AAC

Para el análisis completo de la trama AAC, se utilizará el programa AAC Audio ES Viewer [25], mismo que proporciona una versión limitada de pruebas para el análisis mediante una representación visual de las tramas elementales AAC bajo el estándar ISO/IEC 13818-7.

Tomando como referencia el archivo de prueba de audio codificado con las características dadas en la Tabla 6.2, se puede apreciar la ventana principal del analizador en la Figura 6.5.

Offset	Length	ID	Layer	Sampling Rate	Profile	Channel Configuration
0x0000000000000000	182	0	0	44100 Hz	1	1-CF
0x0000000000000150	182	0	0	44100 Hz	1	1-CF
0x0000000000000281	182	0	0	44100 Hz	1	1-CF
0x00000000000003FE	182	0	0	44100 Hz	1	1-CF
0x0000000000000554	182	0	0	44100 Hz	1	1-CF
0x00000000000006AA	182	0	0	44100 Hz	1	1-CF
0x0000000000000805	182	0	0	44100 Hz	1	1-CF
0x000000000000095A	182	0	0	44100 Hz	1	1-CF
0x0000000000000AC3	182	0	0	44100 Hz	1	1-CF
0x0000000000000C2C	182	0	0	44100 Hz	1	1-CF
0x0000000000000D9A	182	0	0	44100 Hz	1	1-CF
0x0000000000000F03	182	0	0	44100 Hz	1	1-CF
0x000000000000106F	182	0	0	44100 Hz	1	1-CF
0x00000000000011D0	182	0	0	44100 Hz	1	1-CF
0x000000000000133C	182	0	0	44100 Hz	1	1-CF
0x0000000000001498	182	0	0	44100 Hz	1	1-CF
0x00000000000015F8	182	0	0	44100 Hz	1	1-CF
0x0000000000001758	182	0	0	44100 Hz	1	1-CF
0x00000000000018B1	182	0	0	44100 Hz	1	1-CF
0x0000000000001A18	182	0	0	44100 Hz	1	1-CF
0x0000000000001B89	182	0	0	44100 Hz	1	1-CF
0x0000000000001CF2	182	0	0	44100 Hz	1	1-CF
0x0000000000001E4F	182	0	0	44100 Hz	1	1-CF
0x0000000000001FAF	182	0	0	44100 Hz	1	1-CF
0x00000000000022FD	182	0	0	44100 Hz	1	1-CF
0x0000000000002527	182	0	0	44100 Hz	1	1-CF
0x00000000000027A7	182	0	0	44100 Hz	1	1-CF
0x0000000000002917	182	0	0	44100 Hz	1	1-CF
0x0000000000002B74	182	0	0	44100 Hz	1	1-CF
0x0000000000002D7E	182	0	0	44100 Hz	1	1-CF
0x0000000000002F3E	182	0	0	44100 Hz	1	1-CF
0x00000000000032A6	182	0	0	44100 Hz	1	1-CF
0x00000000000035D1	182	0	0	44100 Hz	1	1-CF
0x0000000000003832	182	0	0	44100 Hz	1	1-CF
0x0000000000003B30	182	0	0	44100 Hz	1	1-CF
0x0000000000003E00	182	0	0	44100 Hz	1	1-CF
0x0000000000004166	182	0	0	44100 Hz	1	1-CF
0x000000000000452E	182	0	0	44100 Hz	1	1-CF

```

Hex View
0x0000000000000000 FF F1 50 40 16 DF FC 01 4A 15 98 28 26 12 0D 82 yAFB.SU.J..(t...
0x0000000000000010 81 61 C0 58 30 16 0C 0D 85 05 90 B0 60 CC 18 0B .a&K0.....*i.
0x0000000000000020 0C 86 EB 9F 3F 7F D3 BE 3E BF 29 C4 5D BC 78 F3 ..e.?@>.)&|w&
0x0000000000000030 DE 7E 26 A7 09 A9 75 01 B4 1D 9F 5F CA 76 DE 9C p~&$.@u.'..Evp.
0x0000000000000040 BB 31 FE E6 A0 D2 1B FA BB F8 9E CE 53 61 F1 96 >lp& 0..@&..ISa&.
0x0000000000000050 06 6F F8 C6 20 F6 1B BF DE A3 C8 CC 2D 0D F9 4F .o&E 0..p&E&I-.&0
0x0000000000000060 CA BE 67 0D F1 62 39 CF EC 2D 01 71 B8 20 C3 77 E&g.&B9Ii-.q..&w
0x0000000000000070 7D E2 17 77 D3 A7 F6 62 EC AE FD 3C EA C2 DE CB }&.w&Soni&y<&Ab&E
0x0000000000000080 7D CC 36 ED 9B 44 43 1E E1 19 EE 1E 92 D7 SE 01 }I&l.DC.&.i..*".
0x0000000000000090 99 D1 0F 27 3E 25 5B 9F EE 21 40 7D 86 66 1D DE .N..>&{.i!&}.f.&p
  
```

Figura 6.5. Ventana principal del analizador AAC Audio ES Viewer con el análisis general del archivo de audio obtenido

En la Figura 6.5. se pueden apreciar tres ventanas que componen la ventana principal del analizador, dichas ventanas se detallan a continuación con la información que proporciona cada una de ellas para el archivo de audio codificado.

- **Ventana de análisis de trama**

Se puede observar varios parámetros de la trama cargada al analizador, entre ellos el tamaño de los bloques que conforman la trama con su respectivo ID y capa a la que pertenecen, además de la frecuencia de muestreo usada en cada uno de ellos y la configuración de canal. Dichos parámetros pueden ser apreciados en la Figura 6.6.

Offset	Length	ID	Layer	Sampling Rate	Profile	Channel Configuration
0x0000000000000000	182	0	0	44100 Hz	1	1 - CF
0x0000000000000160	182	0	0	44100 Hz	1	1 - CF
0x00000000000002B1	182	0	0	44100 Hz	1	1 - CF
0x00000000000003FE	182	0	0	44100 Hz	1	1 - CF
0x0000000000000554	182	0	0	44100 Hz	1	1 - CF
0x00000000000006AA	182	0	0	44100 Hz	1	1 - CF
0x0000000000000805	182	0	0	44100 Hz	1	1 - CF
0x000000000000095A	182	0	0	44100 Hz	1	1 - CF
0x0000000000000AC3	182	0	0	44100 Hz	1	1 - CF
0x0000000000000C2C	182	0	0	44100 Hz	1	1 - CF
0x0000000000000D9A	182	0	0	44100 Hz	1	1 - CF
0x0000000000000F03	182	0	0	44100 Hz	1	1 - CF
0x000000000000106F	182	0	0	44100 Hz	1	1 - CF
0x00000000000011DD	182	0	0	44100 Hz	1	1 - CF
0x000000000000133C	182	0	0	44100 Hz	1	1 - CF
0x000000000000149B	182	0	0	44100 Hz	1	1 - CF
0x00000000000015F8	182	0	0	44100 Hz	1	1 - CF
0x0000000000001758	182	0	0	44100 Hz	1	1 - CF
0x00000000000018B1	182	0	0	44100 Hz	1	1 - CF
0x0000000000001A1B	182	0	0	44100 Hz	1	1 - CF
0x0000000000001B89	182	0	0	44100 Hz	1	1 - CF
0x0000000000001CF2	182	0	0	44100 Hz	1	1 - CF
0x0000000000001E4F	182	0	0	44100 Hz	1	1 - CF
0x0000000000001FAF	182	0	0	44100 Hz	1	1 - CF
0x00000000000020FD	182	0	0	44100 Hz	1	1 - CF
0x0000000000002257	182	0	0	44100 Hz	1	1 - CF
0x00000000000023AE	182	0	0	44100 Hz	1	1 - CF
0x0000000000002517	182	0	0	44100 Hz	1	1 - CF
0x0000000000002674	182	0	0	44100 Hz	1	1 - CF
0x00000000000027DE	182	0	0	44100 Hz	1	1 - CF
0x000000000000293E	182	0	0	44100 Hz	1	1 - CF
0x0000000000002A86	182	0	0	44100 Hz	1	1 - CF
0x0000000000002BD1	182	0	0	44100 Hz	1	1 - CF
0x0000000000002D32	182	0	0	44100 Hz	1	1 - CF
0x0000000000002E9D	182	0	0	44100 Hz	1	1 - CF
0x0000000000003000	182	0	0	44100 Hz	1	1 - CF
0x0000000000003166	182	0	0	44100 Hz	1	1 - CF
0x00000000000032BE	182	0	0	44100 Hz	1	1 - CF

Figura 6.6. Verificación de parámetros del archivo de audio obtenido según el analizador AAC Audio ES Viewer

De la información obtenida en esta ventana para el archivo de audio codificado de pruebas, se puede determinar que el mismo está conformado de varios bloques de tamaño 182, y cada uno de ellos se encuentra asociado al mismo archivo de audio debido a los parámetros de ID = 0 y Capa = 0, además de indicar una frecuencia de muestreo de 44100 Hz y la configuración de un único canal de audio (mono).

- **Ventana de análisis de cabecera ADTS**

Se pueden observar los parámetros configurados que contiene la cabecera ADTS para el archivo de audio codificado, los mismos pueden ser apreciados en la Figura 6.7.

```

└─ adts_frame()
  └─ adts_fixed_header()
    ├── syncword           : 0x0FFF (12 bits)
    ├── ID                 : 0 (1 bit)
    ├── layer              : 0 (2 bits)
    ├── protection_absent  : 1 (1 bit)
    ├── profile            : 1 [Low Complexity profile (LC)] (2 bits)
    ├── sampling_frequency_index : 4 [44100 Hz] (4 bits)
    ├── private_bit       : 0 (1 bit)
    ├── channel_configuration : 1 [1 - CF] (3 bits)
    ├── original/copy     : 0 (1 bit)
    └── home              : 0 (1 bit)
  └─ adts_variable_header()
    ├── adts_error_check()
    └── raw_data_block()
  
```

Figura 6.7. Verificación y análisis de la cabecera ADTS del archivo de audio obtenido según el analizador AAC Audio ES Viewer

Como se puede observar en la figura anterior, la cabecera ADTS contiene la información correcta que fue previamente configurada durante el proceso de formación del *bitstream*, cuyos parámetros pueden ser comparados con los de la Figura 6.4., donde se hace uso de otro analizador de cabecera.

- **Ventana de análisis de trama en formato hexadecimal**

Se puede observar el análisis de la trama AAC cargada al analizador en código hexadecimal, la estructura de la trama indicada puede ser apreciada en la Figura 6.8.

```

Hex View
0x0000000000000000 FF F1 50 40 16 DF FC 01 4A 15 98 28 26 12 0D 82 yñP@.Bú.J..(¿...
0x0000000000000010 81 61 C0 58 30 16 0C 0D 85 05 90 B0 60 CC 18 0B .aAX0.....°í..
0x0000000000000020 0C 86 EB 9F 3F 7F D3 BE 3E BF 29 C4 5D BC 78 F3 ..é.?.Ó%>¿)Ä]kxó
0x0000000000000030 DE 7E 26 A7 09 A9 75 01 B4 1D 9F 5F CA 76 DE 9C B~¿$.@u.'... ÈvB.
0x0000000000000040 BB 31 FE E6 A0 D2 1B FA BB F8 9E CE 53 61 F1 96 »1pæ Ò.ú»ø.ÏSañ.
0x0000000000000050 06 6F F8 C6 20 F6 1B BF DE A3 C8 CC 2D 0D F9 4F .oæE ö.¿PæÈÏ-.ùO
0x0000000000000060 CA BE 67 0D F1 62 39 CF EC 2D 01 71 B8 20 C3 77 È%g.ñb9Ïì-.q, Åw
0x0000000000000070 7D E2 17 77 D3 A7 F6 62 EC AE FD 3C EA C2 DE CB }á.wÓsóbì@ý<éÅBÈ
0x0000000000000080 7D CC 36 ED 9B 44 43 1E E1 19 EE 1E 92 D7 5E 01 }íéi.DC.á.í...x^
0x0000000000000090 99 D1 0F 27 3E 25 5B 9F EE 21 40 7D 86 66 1D DE .Ñ.'>*[.i!@).r.È]

Hex View
0x00000000000045A30 5D AC DE 40 4E E5 84 CA 51 B1 E6 64 6F D7 7B D1 ]-B@Ná.ÈQ±ædo×{Ñ
0x00000000000045A40 01 B6 F7 C2 37 1A 1B B4 4D F1 3F 0B 6A CF F3 FC .q÷Á7..Mñ?.jíoú
0x00000000000045A50 CC FF C3 09 F1 64 B8 67 C1 E4 D0 7D C8 A2 E5 C0 ÌÿÄ.ñd,qÁaD)ÈcáÄ
0x00000000000045A60 0A A9 27 59 8D 5D 87 D8 4A 4D 39 BA 6E 3D 27 17 .@'Y.].@JM9°n='
0x00000000000045A70 57 A2 E9 45 CF 7D A1 F8 9F 5F 95 23 A1 30 16 73 WeéEÏ};ø._.##;0.s
0x00000000000045A80 77 DC C6 AE AB CD 9B AD B0 03 9D 0F 22 DC C6 B8 wÜE@«í.-°..."ÜE,
0x00000000000045A90 96 AA 80 15 E4 7C EE 2A 18 63 F2 CF 5A E3 E8 2F .*.ä|í*.côÏZàè/
0x00000000000045AA0 93 DD B7 7F A2 E8 B9 5E C3 AE 9E 7F 8C E9 38 9C .Ý..cè¹^Ã@...és.
0x00000000000045AB0 43 3A 72 D1 FD 3F 68 24 4B 03 9D BC 6C 61 3F 60 C:rÑÿ?hSK...|a?`
0x00000000000045AC0 F4 07 8C 75 7F 8E 60 9C 73 B9 CE E0 ó..u...s'îä

```

Figura 6.8. Verificación de parámetros del archivo de audio obtenido en representación hexadecimal según el analizador AAC Audio ES Viewer

Con la información dada en la figura anterior, se puede analizar de mejor manera el *bitstream* del archivo codificado en su totalidad al hacer uso de numeración hexadecimal, además de poder verificar errores en la estructura y realizar modificaciones de manera más eficiente, con las opciones dadas por el analizador.

Adicionalmente, el analizador posee una opción para representar gráficamente todos los bloques que conforman la trama. Los resultados gráficos obtenidos para el archivo de pruebas de audio codificado se pueden apreciar en la Figura 6.9.

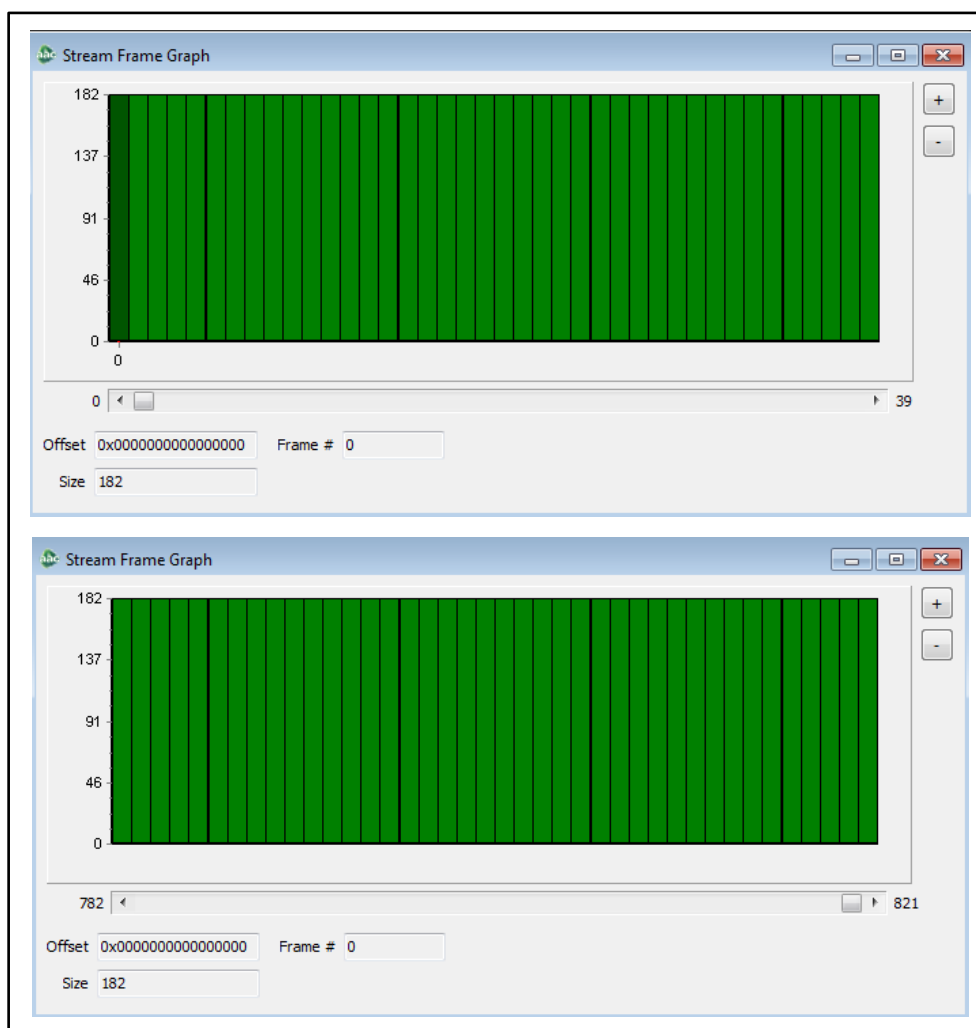


Figura 6.9. Representación gráfica de bloques de la trama AAC para el archivo de audio codificado

En la figura anterior se puede apreciar el tamaño de los bloques que conforman la trama AAC para el archivo de audio codificado, se puede observar que todos los bloques poseen el mismo tamaño y además de poder verificarse el número total de bloques que conforman la trama, que para el caso son 822 (Inicio: 0 / Fin: 821).

6.7. Generación de archivo de audio TS

Las características del archivo de audio TS obtenido a la salida del codificador, en base al archivo codificado AAC, se puede apreciar en la Tabla 6.3.

Tabla 6.3. Características del audio de salida TS

Archivo de salida	Tamaño de archivo	Frecuencia de muestreo	Número de canales	Tasa de bits
salida.ts	990 kilobytes	44100 Hz	1	384 kilobit/s

Dichas características dadas en la Tabla 6.3. fueron obtenidas del reproductor y analizador de formatos de audio denominado SMPLAYER [26], como se puede apreciar en la Figura 6.10.

The screenshot shows the SMPlayer interface with a yellow music note icon and the filename 'salida.ts'. The 'General' section lists: Archivo: C:/Users/user/Dropbox/Tesis/Programas Tesis/Codificador AAC_TS_v3_1/salida.ts, Tamaño: 989 KB (0 MB), Duración: 00:00:00, Demuxer: mpegts. The 'Pista de audio inicial' section lists: Formato: 80, Tasa de bits: 384 kbps, Tasa: 44100 Hz, Canales: 1, Código seleccionado: ffmpeg2float. The 'Pistas de audio' section shows a table with one track: n° 0, Idioma <vacío>, Nombre <vacío>, ID 256.

Figura 6.10. Características del audio TS obtenido en la salida del codificador dadas por el programa SMPLAYER

De lo observado en la figura anterior, se puede determinar que las propiedades del archivo de audio en formato TS no han sido modificadas por el conversor FFMPEG, ya que se ha mantenido los parámetros principales como son la frecuencia de muestreo y el número de canales respecto al archivo de audio de entrada. Además se puede determinar una tasa de *bits* de 384 kilobit/s y únicamente se determinan componentes de audio.

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

7.2. Conclusiones

- Se ha diseñado un codificador de audio basado en un algoritmo de compresión con pérdidas bajo las normativas dadas por el estándar ISO/IEC 13818-7 (AAC), capaz de entregar a su salida una señal de audio en formato *Transport Stream* mediante un proceso de conversión y asociación externa al programa FFMPEG.
- El diseño fue realizado mediante el uso del lenguaje de programación m de MATLAB, e incluso durante la asociación externa al programa FFMPEG, donde se relacionó un archivo BAT de Windows al código de programación. El diseño en lenguaje de programación m es de mucha utilidad para realizar diversas modificaciones y puede ser transformado a lenguaje de programación VHDL para que el mismo pueda ser implementado a niveles de hardware en equipamiento de transmisores de Televisión Digital Terrestre bajo las normas de la ISDB-Tb.
- Existen varias ventajas de realizar el proceso de codificación usando AAC entre las mismas se pueden destacar el uso de la MDCT y las funciones de ventana para la transformación de la señal, ya que brinda un mejor análisis y tratamiento de la misma para el proceso de codificación. Adicionalmente, durante el desarrollo del presente proyecto, se ha indicado las considerables mejoras que presenta el formato AAC con respecto al formato MP3.
- El proceso de codificación en AAC permite un alto índice de compresión de los datos de la señal de audio de entrada, por lo cual el mismo puede ser utilizado en varios sistemas de transmisión actuales.

- En el presente proyecto se omiten varios bloques de procesamiento de la señal durante el proceso de codificación dado en el diagrama general de AAC, con lo que se puede determinar que los mismos no son necesarios pero si importantes para obtener una mejor calidad de audio en la salida y un mayor índice de compresión. De manera general se puede indicar la importancia de un modelo psicoacústico durante el proceso descrito.
- Mediante el estudio y generación de la trama en formato AAC, se ha determinado la importancia del uso de cabeceras especiales, como es el caso de ADTS, cuyo funcionamiento se relaciona directamente con la de una trama AAC. La importancia radica en que la misma contiene toda la información de la trama que se está transmitiendo, dicha información es de gran importancia para el decodificador o también para un programa externo que necesite realizar alguna modificación de la trama según la información almacenada en dicha cabecera. Para el caso, la trama AAC con la cabecera ADTS puede ser considerada como un paquete elemental (PES) para la formación del paquete *Transport Stream* durante el proceso de obtención de la trama en formato TS.
- De los casos de prueba efectuados para comprobar el funcionamiento del codificador diseñado, se puede concluir lo siguiente:
 - a) El codificador únicamente garantiza la apertura de archivos en formato WAV, mismo que debe poseer ciertas características.
 - b) El tiempo de análisis y procesamiento de la señal, hasta obtener el archivo de audio codificado, dependerá del tamaño del archivo de audio de entrada y los bloques que requieran ser analizados.
 - c) La obtención de la señal de audio en la salida del codificador, dependerá exclusivamente de la asociación de MATLAB al programa FFMPEG, cualquier cambio deberá ser realizado desde el archivo con extensión *.bat que se encuentre asociado al lenguaje de programación.

- d) Mediante diversas herramientas gráficas dadas por MATLAB, se puede observar de forma gráfica el proceso de codificación realizado, dando mayor detalle del proceso de lectura y formación de bloques de análisis, además del tratamiento de cada uno de los bloques generados con la función ventana y la MDCT.
 - e) Con la obtención del archivo de audio codificado se puede realizar una comparación de los tamaños de los mismos, donde se puede determinar el éxito del proceso de codificación, así como la calidad de audio del archivo obtenido.
- La elaboración de las funcionalidades del codificador fue posible gracias a las diversas funciones para el tratamiento digital de señales dadas por MATLAB.

7.3. Recomendaciones

- Se debe extender la funcionalidad del codificador diseñado a niveles de *hardware*, mediante diferentes códigos de programación como es el caso de VHDL. Esto con el fin de que la funcionalidad pueda ser implementada en equipos de Televisión Digital Terrestre con las normativas de la ISDB-Tb que cumplan los requerimientos de AAC como estándar para la codificación de audio. Dicha extensión podría a futuro generar equipos realizados en nuestro país.
- Con fines de mejorar la calidad del codificador diseñado, se debería realizar la integración de una mayor cantidad de bloques funcionales dados por el esquema general del codificador AAC. Con esta mejora, se podría obtener una mejor calidad de audio y un mayor porcentaje de compresión con un mejor tratamiento de ruido y tratamiento multicanal, requerimientos que cada día son mayormente necesarios por los sistemas de transmisión.

- Se debería explotar mayormente en temas de investigación y desarrollo al codificador de audio AAC, mismo que como se ha mencionado anteriormente, presenta muchas ventajas en comparación al formato MP3, y actualmente muchas compañías de audio lo han adoptado como su estándar principal.
- Se debe tener en cuenta que el codificador diseñado está abierto a mejoras, los cuales pueden ser fácilmente integradas al código de programación de MATLAB. Adicionalmente, se debe considerar que las funciones a integrarse deben ser revisadas para el trabajo conjunto con las funciones actualmente integradas para que de esta manera no se afecte el rendimiento del codificador.

BIBLIOGRAFÍA

- [1] A. Vorona. (2012, Febrero 16). Directorio de Artículos Gratis - docsetools.com. [En línea]. Disponible en: http://docsetools.com/articulos-enciclopedicos/article_96634.html
- [2] A. Ayala (2010, Junio 25). Programación Java, Software Libre, Televisión Digital Terrestres, Música Libre, apuntes, opinión y más. [En línea]. Disponible en: <http://alejandroayala.solmedia.ec/?p=502>
- [3] The Mathworks Inc. (1989, Octubre 03). MathWorks - MATLAB and Simulink for Technical Computing. [En línea]. Disponible en: <http://www.mathworks.com/fpga-design/>
- [4] WEEBLY, INC. (2006, Marzo 29). Weebly: Create a Free Website, Online Store, or Blog. [En línea]. Disponible en: <http://losdeherramientastelematicas1.weebly.com/aac.html>
- [5] DIBEG.ORG. (1999, Febrero 16). DiBEG | ISDB-T Official Web Site. [En línea]. Disponible en: http://www.dibeg.org/techp/feature/isdb-t_Spanish.pdf
- [6] DATAxis. (2008, Julio 22). Business Media, Research & Events on Next Generation TV in Latin America. [En línea]. Disponible en: <http://nextvlatam.com/brazil-analyzes-the-isdb-t-alert-system/?lang=es>
- [7] Investigacionit.com.ar. (2012, Octubre 24). Investigación IT. [En línea]. Disponible en: <http://investigacionit.com.ar/compression-en-teradata/>
- [8] València, U. P. (2009, Octubre 10). UPV Universitat Politècnica de València. [En línea]. Disponible en: http://www.grc.upv.es/docencia/tra/referencias/AudioCoding/Brandenburg_mp3_aac.pdf

- [9] NAROD. (1999, Diciembre 09). Конструктор сайтов Яндекс.Народ переехал на сервис uCoz. [En línea]. Disponible en: <http://dsp-book.narod.ru/DSPMW/42.PDF>
- [10] Audio Engineering Society. (1995, Octubre 19). Audio Engineering Society (AES). [En línea]. Disponible en: <http://www.aes.org/e-lib/browse.cfm?elib=7395>
- [11] Glaeser, T. (2009, Octubre 05). Leibniz Universität Hannover - Startseite. [En línea]. Disponible en: ftp://ftp.tnt.uni-hannover.de/pub/MPEG/audio/mpeg4/documents/w2803/w2803_n.pdf
- [12] Gang, L. (2003, Noviembre 20). www.pudn.com. [En línea]. Disponible en: http://read.pudn.com/downloads98/doc/comm/401153/14496/ISO_IEC_14496-3%20Part%203%20Audio/C036083E_SUB1.PDF
- [13] Information Processing Society of Japan. (2008, Marzo 30). 情報処理学会. [En línea]. Disponible en: <https://www.itscj.ipsj.or.jp/sc29/open/29view/29n6475t.doc>
- [14] Information Processing Society of Japan. (2008, Marzo 30). 情報処理学会. [En línea]. Disponible en: kikaku.itscj.ipsj.or.jp/sc29/open/29n13085t.doc
- [15] XIN NET TECHNOLOGY CORPORATION. (2003, Noviembre 20). www.pudn.com - 程序员联合开发网 源代码数目238万. [En línea]. Disponible en: http://read.pudn.com/downloads164/doc/747903/CT_aacPlus_whitepaper.pdf
- [16] PERFECT PRIVACY, LLC. (2004, Mayo 24). FPO IP Research & Communities. [En línea]. Disponible en: <http://www.freepatentsonline.com/20070297624.pdf>

- [17] Mp3-tech.org. (2000, Abril 03). MP3'Tech - www.mp3-tech.org. [En línea].
Disponible en: <http://www.mp3-tech.org/programmer/docs/audio.pdf>
- [18] Dolby Laboratories, Inc. (2002, Agosto 15). ViaLicensing. [En línea].
Disponible en: <http://www.vialicensing.com/licensing/aac-faq.aspx>
- [19] Dolby Laboratories, Inc. (2002, Agosto 15). ViaLicensingFree. [En línea].
Disponible en: <http://www.vialicensing.com/licensing/aac-fees.aspx>
- [20] Pennsylvania State University. (1986, Julio 14). Penn State-A Public Research University Serving Pennsylvania and the Global Community. [En línea].
Disponible en:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.4563&rep=rep1&type=pdf>
- [21] Nero AG. (1997, Noviembre 04). Nero 2015 Platinum: Established quality - now even better > Download > Buy > Upgrade. [En línea]. Disponible en:
<http://www.nero.com/esl/company/about-nero/nero-aac-codec.php>
- [22] EASYSPACE LTD. (2000, Junio 04). AudioCoding.com. [En línea]. Disponible en: <http://www.audiocoding.com/faac.html>
- [23] EASYSPACE LTD. (2000, Junio 04). AudioCoding.com. [En línea]. Disponible en: <http://www.audiocoding.com/faad2.html>
- [24] P23. (2007, Febrero 06). Projects. [En línea]. Disponible en: <http://www.p23.nl/projects/aac-header/>
- [25] JONGBEL. (2008, Septiembre 05). HOME. [En línea]. Disponible en: <http://www.jongbel.com/aac-audio-es-viewer-product24.html>

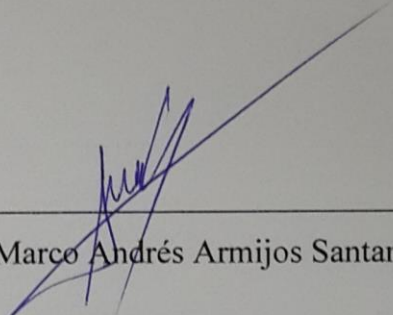
- [26] Dice Holdings, Inc. (1999, Agosto 08). SourceForge - Download, Develop and Publish Free Open Source Software. [En línea]. Disponible en: <http://smplayer.sourceforge.net/>
- [27] ABCRADIOTEL. (2004, Octubre 26). AbcRadioTel. [En línea]. Disponible en: http://www.abcradiotel.com/wp-content/uploads/2012/08/Capitulo_4LFSa.pdf
- [28] Distancia, U. N. (2009, Octubre 13). UNED - Universidad Nacional de Educación a Distancia. [En línea]. Disponible en: http://ocw.innova.uned.es/mmm3/audio_digital/contenidos/pdf/Compresion_de_Audio.pdf
- [29] Granada, U. d. (2009, Octubre 10). Universidad de Granada > Inicio. [En línea]. Disponible en: <http://ceres.ugr.es/~alumnos/luis/codif.html>
- [30] M. Bosi y R. Goldberg, "MPEG-2 AAC", Introduction to Digital Audio Coding and Standards, Kluwer Academic Publishers, Holanda, 2003, pp. 333-367
- [31] Automattic, Inc. (2000, Marzo 03). *WordPress.com: Create a free website or blog.* [En línea]. Disponible en: <https://realidadtrascendental.wordpress.com/psicofonias-2/metodologia/>
- [32] WEBCENTRIC. (2007, Junio 28). CALAMEO. [En línea]. Disponible en: <http://es.calameo.com/read/002697860d4af95980808>
- [33] Nan Hai. (2005, Agosto 29). *南海区质量技术监督综合信息网.* [En línea]. Disponible en: <http://www.nhzjj.com/asp/admin/editor/newsfile/2010318163752818.pdf>

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES
ACTA DE ENTREGA

El presente proyecto fue entregado en el Departamento de Eléctrica y Electrónica, y reposa en los archivos desde:

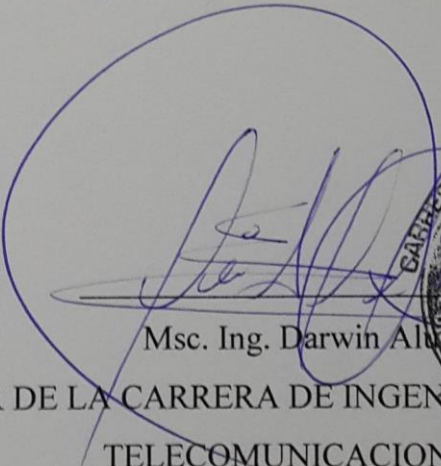
Sangolquí, 11 / 08 / 2015

Elaborado por:



Marco Andrés Armijos Santamaría

Autoridad:



Msc. Ing. Darwin Alujama

DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

