Desarrollo de un algoritmo de compresión del transport stream enfocado en redes de contribución bajo el estándar ISDB-T

Camalle Vásquez, Jerson Rodrigo

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

Ing. Olmedo Cifuentes, Gonzalo Fernando, Ph.D.

01 de marzo de 2021

Curiginal

Document Information

Analyzed document	JERSON_CAMALLE_ESCRITO.pdf (D100820736)
Submitted	4/7/2021 12:13:00 AM
Submitted by	Olmedo Cifuentes Gonzalo Fernando
Submitter email	gfolmedo@espe.edu.ec
Similarity	6%
Analysis address	gfolmedo.espe@analysis.urkund.com

Sources included in the report

SA	**ErickMunoz.docx** Document ErickMunoz.docx (D62487701)	2
W	URL: https://repositorio.espe.edu.ec/bitstream/21000/11248/1/T-ESPE-049421.pdf Fetched: 10/26/2019 6:21:06 PM	2
W	URL: https://docplayer.es/31985117-Desarrollo-de-un-analizador-de-flujos-unicos-de-paqu ... Fetched: 5/24/2020 4:55:02 PM	3
W	URL: http://docshare04.docshare.tips/files/27989/279897601.pdf Fetched: 12/19/2020 9:28:46 PM	6
W	URL: http://www.telemidia.puc-rio.br/~rafaeldiniz/public_files/normas/SBTVD/es/Codifica ... Fetched: 1/4/2021 8:45:23 PM	1
W	URL: https://docplayer.es/82267282-Departamento-de-electrica-y-electronica.html Fetched: 1/29/2020 6:06:28 PM	2
SA	**Universidad de las Fuerzas Armadas ESPE / tesis escrito.pdf** Document tesis escrito.pdf (D21514115) Submitted by: gfolmedo@espe.edu.ec Receiver: gfolmedo.espe@analysis.urkund.com	4
SA	**Universidad de las Fuerzas Armadas ESPE / Proyecto_Erick_Malusin.pdf** Document Proyecto_Erick_Malusin.pdf (D98928339) Submitted by: gfolmedo@espe.edu.ec Receiver: gfolmedo.espe@analysis.urkund.com	2
SA	**tesis17ucca.pdf** Document tesis17ucca.pdf (D8370918)	1

Firmado electrónicamente por:
GONZALO FERNANDO
OLMEDO CIFUENTES

Ing. Olmedo Cifuentes, Gonzalo Fernando, Ph.D.

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Desarrollo de un algoritmo de compresión del transport stream enfocado en redes de contribución bajo el estándar ISDB-T**" fue realizado por el señor **Camalle Vásquez, Jerson Rodrigo** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 24 de marzo de 2021

Firmado electrónicamente por:
GONZALO FERNANDO
OLMEDO CIFUENTES

.....

Ing. Olmedo Cifuentes, Gonzalo Fernando, Ph.D

C. C: 171169634-2

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

RESPONSABILIDAD DE AUTORÍA

Yo, **Camalle Vásquez, Jerson Rodrigo**, con cédula de ciudadanía n° 172320301-2, declaro que el contenido, ideas y criterios del trabajo de titulación**: Desarrollo de un algoritmo de compresión del transport stream enfocado en redes de contribución bajo el estándar ISDB-T** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 24 de marzo de 2021

.....

Camalle Vásquez, Jerson Rodrigo

C.C.: 172320301-2

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Camalle Vásquez, Jerson Rodrigo,** con cédula de ciudadanía n° 172320301-2, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de un algoritmo de compresión del transport stream enfocado en redes de contribución bajo el estándar ISDB-T** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 24 de marzo de 2021

Camalle Vásquez, Jerson Rodrigo

C.C.: 172320301-2

Dedicatoria

A mis hermanas: Verónica, Mayra y Johana, quienes han sido un ejemplo de lucha y esfuerzo.

A mi novia Marie, quien ha estado siempre a mi lado dándome su apoyo incondicional.

Y por último dedico a Antony mi hermano menor con quien hemos luchado la vida y día a día me inspira para convertirme en un hombre ejemplar.

Jerson Camalle

Agradecimiento

Agradezco a mi universidad y a todos sus tutores, que han sido parte de mi formación académica y profesional.

Agradezco a mi familia y amigos que han sido una fuente constante de inspiración para lograr completar esta meta de mi vida satisfactoriamente.

Mi más sincero agradecimiento a mi novia Marie Rodríguez, quien es la persona que día a día me da aliento para seguir superándome y que al tener objetivos en común todo sacrificio que realizamos es para hacer nuestro sueño realidad.

También agradezco a mi tutor, el ingeniero Gonzalo Olmedo, por todo el apoyo brindado en el desarrollo de este proyecto.

Jerson Camalle

Tabla de contenido

Índice de Tablas

Índice de Figuras

Resumen

El presente proyecto de investigación consiste en el desarrollo de un algoritmo capaz de reducir el tamaño de las tramas Transport Stream (TS) y Broadcast Transport Stream (BTS) basado en las normativas dadas por el estándar ISDB-T, en el lenguaje de programación JAVA. El trabajo realizado consta de dos partes. En la primera, se realiza el reconocimiento automático del archivo de extensión .ts con la finalidad de determinar si la trama que se encuentra a la entrada es un TS o un BTS para luego someter dicha trama al proceso de compresión según el modelo de referencia dado por el estándar ISDB-T. En la segunda parte, se realiza el proceso de descompresión de las tramas TS y BTS, una vez realizado este proceso dichas tramas serán expuestas a procesos de verificación mediante el uso de softwares externos como lo es FFMPEG y así comprobar que al recuperar las tramas TS y BTS poseen sus características originales.

Palabras Clave:

- **TRANSPORT STREAM**
- **BROADCAST TRANSPORT STREAM**
- **COMPRESIÓN**
- **FFMPEG**
- **LENGUAJE DE PROGRAMACIÓN JAVA**
- **ISDB-T**

Abstract

This research project consists of the development of an algorithm capable of reducing the size of the Transport Stream (TS) and Broadcast Transport Stream (BTS) frames based on the regulations given by the ISDB-T standard, in the JAVA programming language. . The work carried out consists of two parts. In the first, the automatic recognition of the .ts file is carried out in order to determine if the frame found at the input is a TS or a BTS and then subject said frame to the compression process according to the given reference model. by the ISDB-T standard. In the second part, the decompression process of the TS and BTS frames is carried out, once this process has been carried out, said frames will be exposed to verification processes through the use of external software such as FFMPEG and thus verify that when recovering the TS frames and BTS possess their original characteristics.

Keywords:

- **TRANSPORT STREAM**
- **BROADCAST TRANSPORT STREAM**
- **COMPRESSION**
- **FFMPEG**
- **JAVA PROGRAMMING LANGUAGE**
- **ISDB-T**

Capítulo I

Generalidades

Antecedentes

En la mayor parte de América Latina, la televisión digital terrestre se ha vuelto cada vez más atractiva, en sí misma puede proporcionar una mayor calidad al procesar información digital lo que ha provocado una caída en el precio de los equipos utilizados para esta tecnología (Sotelo, 2011).

El estándar ISBD-T es el más común en América del Sur. Los países que utilizan este estándar son: Ecuador, Brasil, Chile, Bolivia, Perú, Argentina, Venezuela, Paraguay, Uruguay y Colombia.

En Ecuador el estándar ISDB-T fue analizado, comparado y evaluado ante las necesidades de los telespectadores de Televisión Digital Terrestre (TDT). El 26 de marzo de 2010, el Consejo Nacional de Telecomunicaciones (CONATEL) propició la introducción del estándar ISDB-T, ya que presentó algunas ventajas como mejor calidad de señal, más canales disponibles y funciona mejor en interiores o *indoor* (Alulema, 2012).

Actualmente, la transmisión de audio, video y datos interactivos se logra a través de un protocolo llamado *Broadcast Transport Stream* (BTS). Este proceso abre varios campos de investigación, como el que se presenta en este proyecto de estudio, el cual permite el desarrollo de un algoritmo de compresión propio para BTS y *Transport Stream* (TS) según los protocolos del estándar ISDB-T.

Justificación e Importancia

La relevancia de realizar el siguiente proyecto de investigación se basa en el desarrollo de un algoritmo enfocado en la compresión de las tramas TS y BTS, al ser un proceso fundamental dentro de lo que es la TDT, ya que dichas tramas son el resultado de la multiplexación ya sea audio, video o datos interactivos, los mismos que serán difundidos a las diferentes estaciones.

El proyecto de investigación se enfoca principalmente en la compresión de la trama TS y BTS con el fin de reducir el tamaño del paquete completo entre otras características, lo que significa que ocupara menos espectro radioeléctrico al momento de transmitir dichas tramas bajo el estándar ISDB-T.

Se anhela que en futuros trabajos de investigación se implemente este trabajo con el fin de reducir los costos que implica la importación de equipos que tienen su propio sistema de compresión de TS y BTS, de esta forma, se espera que la industria tecnológica del país se beneficie.

Alcance

Se proyecta obtener un algoritmo capaz de reducir el tamaño de las tramas TS y BTS, manteniendo la información original después de que estas tramas hayan pasado por el proceso de compresión mediante el análisis y la detección de paquetes de transporte. Con esto se conseguirá una trama descomprimida de TS y BTS a nivel de software con la esperanza de que futuras investigaciones se lleven a cabo a nivel de hardware.

Objetivos

Objetivo General

Desarrollo de un algoritmo que permita la compresión del *Transport Stream* enfocado en redes de contribución bajo el estándar ISDB-T.

Objetivos Específicos

- Identificar las características principales de la trama TS y BTS, mediante el estado del arte para la detección de paquetes.
- Identificar las características técnicas del estándar ISDB-T requeridas para la BTS.
- Detectar los paquetes de transporte asociados a los flujos elementales de la señal que podrían entrar a un proceso de predicción.
- Implementar y evaluar un algoritmo de compresión de flujos TS y BTS con el fin de evitar pérdidas de información (audio, video o datos interactivos).
- Evaluar la recuperación de tramas TS y BTS, que posean las características originales en primera instancia.

Capítulo II

Introducción a la televisión digital terrestre

Estándares para TDT

La televisión digital terrestre en comparación con la analógica posee varias características que la hacen más atractiva al momento de interactuar con el televidente, ya que ofrece mayor calidad de audio, video e imagen. Además, uno de los objetivos de migrar a la TDT es aprovechar el espectro radioeléctrico, con lo cual se podría introducir nuevas tecnologías en las bandas de frecuencia que quedarían libres debido a la transmisión en formato digital (TDT, 2020).

La característica que hace de la TDT una mejor opción a la hora de transmitir señales de televisión sobre la señal abierta es que puede ofrecer al espectador servicios interactivos que permiten al distribuidor interactuar con el consumidor.

Estándar DVB-T (*Digital Video Broadcasting - Terrestrial*)

En 1996, la organización europea DVB (*Digital Video Broadcasting*) culmino con la especificación 2k/8k ahora denominada DVB-T. Este estándar permite transmitir audio, video y datos por medio de MPEG-2 y emplea el esquema de modulación también conocido como COFDM (Multiplexación por División de Frecuencia Ortogonal Codificada) (Ladebusch & Liss, 2006).

Este estándar ha sido acogido en varios países además de Europa de los que destacan Colombia, Panamá, Emiratos Árabes Unidos, Catar y Australia entre otros.

Estándar ATSC (*Advanced Television Systems Committee*)

Este estándar fue creado por los Estados Unidos para reemplazar la televisión analógica NTSC. Este estándar se define como una imagen con una resolución de 1920 x 1080 píxeles y una relación de aspecto de 16: 9, lo que significa que representa una resolución más alta en comparación con los estándares antiguos. Algunos países que adoptaron esta regulación digital con un televisor analógico se vieron afectados por el hecho de que no podían comprar un televisor digital. Por tanto, en estos casos era más habitual conectar sus televisores analógicos a un decodificador de señal y así apreciar los diferentes programas de televisión.

Este estándar ha sido acogido en varios países de los que destacan Estados Unidos, Republica dominicana, Corea de Sur, México y Puerto Rico entre otros.

Estándar DTMB (*Digital Terrestrial Multimedia Broadcast*)

Además de los servicios convencionales, este estándar también ofrece servicios para terminales fijos y terminales móviles. Hay dos tipos de recepción por los que se conoce el estándar DTMB: el móvil, que admite la definición estándar SD, y el fijo, que admite alta definición HD.

DTMB es el resultado de la fusión de varias tecnologías como un código de ruido (PN-*Pseudo-random Noise*), una codificación LDPC (*Low-Density Parity-Check*) y una modulación TDS-OFDM (*Time Domain Synchronization - Orthogonal Frequency Division Multiplexing*).

Este estándar ha sido acogido en varios países de los que destacan China, Hong Kong, Cuba y Camboya entre otros.

Estándar ISDB (*Integrated Service Digital Broadcasting*)

La Radiodifusión Digital de Servicios Integrados fue creada por ARIB (*Association of Radio Industries and Businesses*) y se implementó por primera vez en Japón para reemplazar varias tecnologías heredadas como NTSC-J y HDTV. Como resultado, varios países cambiaron a este nuevo estándar no solo en Asia, sino en todo el mundo.

Este estándar ha sido acogido en varios países de Asia como Japón, Filipinas, Maldivas y en América de los que destacan Brasil, Argentina, Uruguay, Chile, Venezuela, Guatemala y Honduras entre otros.

Estándar ISDB-T (*Integrated Service Digital Broadcasting - Terrestrial*)

El estándar ISDB-T fue desarrollado en Japón en 1990, está diseñado para transmitir tanto HDTV (Televisión de Alta Definición) como SDTV (Televisión Digital Estándar). Además, su función está diseñada para ser utilizada en anchos de banda de 6 a 8 MHz. Para el caso especial de Ecuador, el estándar operara con un canal de 6 MHz (Gómez, Lapo, & Oñate, 2019).

Este estándar ha sido acogido en varios países de Asia como Japón, Filipinas, Maldivas, Sri Lanka y en América de los que destacan Brasil, Argentina, Uruguay, Chile, Venezuela, Perú, Costa Rica, Guatemala y El Salvador entre otros.

Características principales

A continuación, se presentan las características más importantes dentro del estándar ISDB-T:

- La transmisión simultánea de canales de HDTV y canales móviles de ancho de banda de 6MHz está destinada a la transmisión de TV analógica.
- Tiene una gran capacidad para resistir la interferencia de trayectos múltiples, lo que puede provocar un efecto "*ghosts*" en la televisión analógica. Este estándar tiene la menor robustez contra la interferencia de canales analógicos adyacentes, en comparación con otros sistemas, está por debajo del estándar ATSC, que es notoriamente robusto.
- Tiene una mayor capacidad para resistir la interferencia de la banda UHF y puede resistir señales transitorias, como motores de automóviles y líneas eléctricas que existen en las ciudades. Estas señales transitorias se agrupan principalmente en la banda VHF y son más fuertes en el rango de baja frecuencia.
- Con el multiplexor de canales SDTV puede seleccionar más de 2 canales de televisión SDTV y no solo uno en HDTV.
- Permite el suministro de EPG (*Electronic Program Guide*).
- Ofrece aplicaciones interactivas mediante la transmisión de datos a través de broadcast o por Internet.

Estándar ISDB-T Internacional

El estándar ISDB-T Internacional nació en 2009 y es promovido por Japón y Brasil. En la actualidad es uno de los estándares más utilizados y recomendados en América Latina y ha sido adoptado en otras partes del mundo. El estándar tiene nuevas tecnologías de codificación de audio, video e interactividad propuestas por Brasil y mantienen la capa física del sistema Japones.

Capítulo III

Televisión Digital Terrestre en Ecuador

Beneficios de la TDT

Existen cuatro estándares globales para la adopción de TDT. Con base en esta información, los países han adoptado los estándares de acuerdo con sus propias necesidades. Como la mayoría de los países sudamericanos, Ecuador ha elegido el estándar internacional ISDB-T (TDT, 2020).

La norma adoptada por Ecuador permite:

- Se puede acceder a varias programaciones en una misma estación a través de una señal compartida como: noticias, deportes, venta, series, etc.
- Apreciar por medio de equipamiento móvil y portátil la calidad de las señales digitales.
- Entregar a la población señales de alerta de emergencia y así convertir a la televisión en una herramienta más dedicada a salvaguardar vidas.
- Evidenciar aplicaciones interactivas como servicios de salud, entretenimiento, entre otros.
- Servicio de guía o programación electrónica.

Estándar ISDB-Tb

En 2006, Brasil desarrolló el estándar ISDB-Tb, también conocido como SBTVD. Sin embargo, esto es el resultado de varias modificaciones al estándar ISDB-T Japonés para lograr una mejor calidad de señal y agregar ciertas características como la compresión de datos por medio de MPEG-4, la codificación de audio por medio MPEG-4 HE-AAC v.1 nivel 2, la codificación de video por medio de H.264 a 30 FPS y un nuevo aplicativo llamado GINGA (Moncayo Flores & Pozo Caicedo, 2014) .

El estándar ISDB-Tb posee una arquitectura básica, la cual consta de 3 partes principales:

- Sistema de producción. – Esta es la primera etapa de presentación de contenido multimedia y datos interactivos.
- Sistema de Transmisión. – Es la segunda etapa, aquí la información proveniente del sistema de producción pasa por varios codificadores y un procesamiento de banda-base.
- Visualizador. – Esta es la tercera etapa, una vez realizado el proceso de multiplexación, codificación y modulación se puede presentar la información en el receptor correspondiente.

A continuación, en la ***Tabla 1*** se presentan algunas de las especificaciones del estándar ISDB-Tb:

Tabla 1

Especificaciones Básicas del sistema ISDB-Tb.

Parámetros	Modo 1	Modo 2	Modo 3
N° de segmentos		13	
Ancho de banda útil	5.575 MHz	5.573 MHz	5.572 MHz
Separación entre portadoras	3.968 kHz	1.984 kHz	0.992 kHz
Total, número de portadoras	1405	2809	4992
Tipo de modulación	QPSK, 16QAM, 64QAM, DQPSK		
Símbolos por cuadro		204	
Tamaño de símbolo	252 us	504 us	1008 us
Intervalo de guarda	1/4, 1/8, 1/16, 1/32	1/4, 1/8, 1/16, 1/32	1/4, 1/8, 1/16, 1/32
Taza de código interno	1/4, 2/3, 3/4, 5/6	1/4, 2/3, 3/4, 5/6	1/4, 2/3, 3/4, 5/6
Velocidad de transferencia útil	3.651 Mbps – 23.234 Mbps		

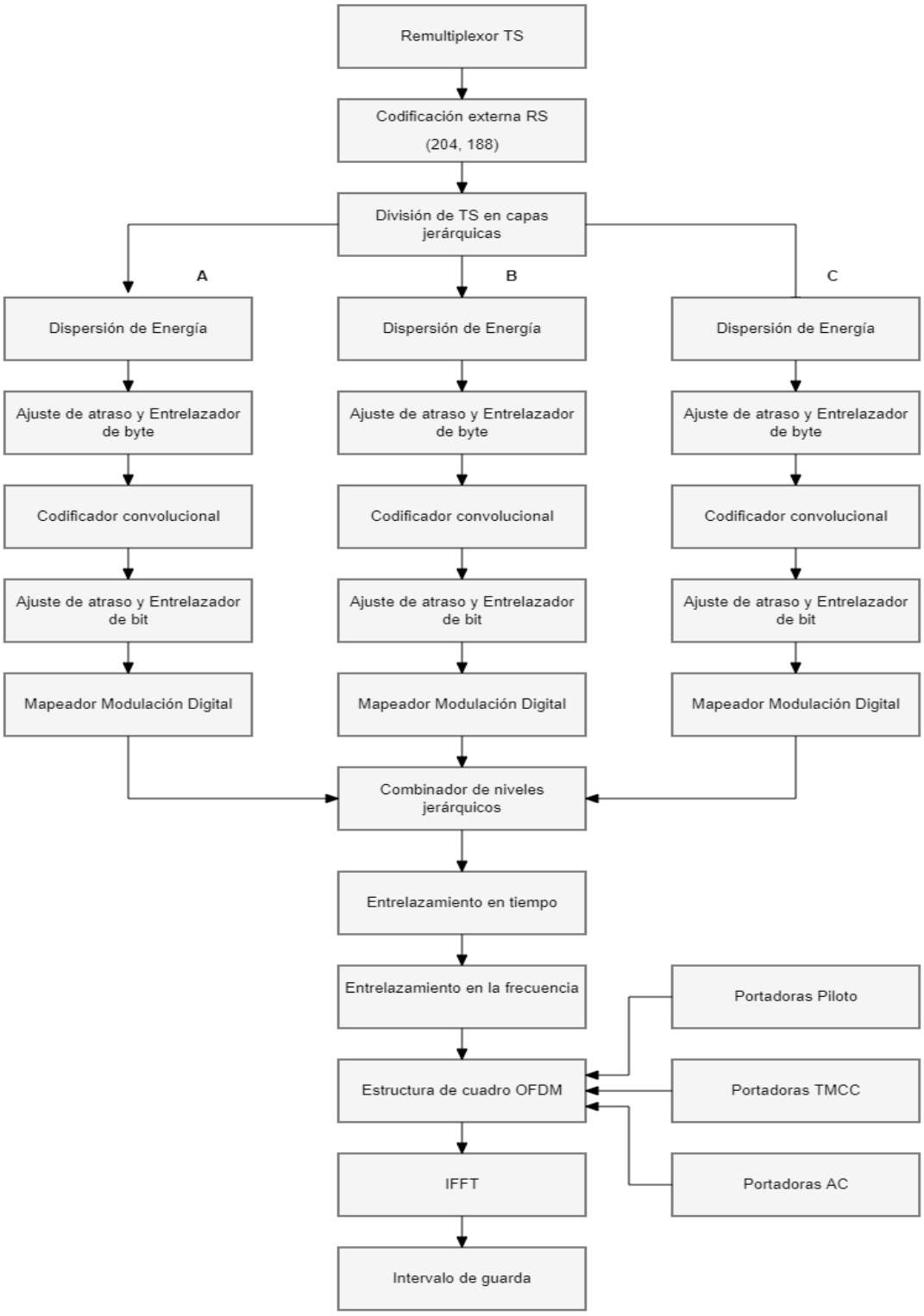
Capa física

En el estándar ISDB-Tb, la capa física juega un papel importante en la transmisión de audio, video y datos interactivos porque ofrece alta calidad. Además, una de las principales funciones de la capa física es que procesa la información confidencial del usuario y puede agregar o extraer programas según la estructura de una programación local. Además, se encuentra dividida en 3 capas jerárquicas A, B y C, las cuales tienen una estructura igual entre sí y de esta forma el modulador posee una configuración independiente.

En la ***Figura 1*** se muestra el diagrama de transmisión jerárquica en ISDB-Tb, para posteriormente describir los bloques que comprenden este proceso.

Figura 1

Diagrama de transmisión jerárquica en ISDB-Tb.



Dispersor de energía

Este bloque de procesamiento se ubica en las tres capas jerárquicas A, B y C para evitar la transmisión continua de una secuencia de ceros o unos, esto se logra mediante una compuerta XOR entre el flujo de datos a la entrada y una determinada secuencia.

Ajuste de retraso

Es común que debido a la modificación de la configuración de la capa jerárquica se genere un retardo en el receptor, la corrección de estos retardos se realiza insertando tanto retardo como sea necesario durante la transmisión, logrando un ajuste significativo.

Entrelazador

Este bloque de procesamiento es el encargado de combinar los bytes o bits a ser transmitidos, con el fin de poder corregir los errores que se generan después de pasar por el canal de transmisión.

Codificación de Canal

Dentro del estándar ISDB-Tb se realizan dos procesos de codificación, uno interno y otro externo. El objetivo de estos procesos es eliminar los errores que ocurren de la transmisión mediante redundancia, agregando o quitando bits según sea el caso.

Modulación

El flujo de datos se somete a un proceso de entrelazado y luego se asigna a una constelación de modulación digital. Para el estándar ISDB-Tb, los esquemas de modulación son: DQPSK, QPSK, 16-QAM y 64-QAM, estos esquemas se utilizan para la

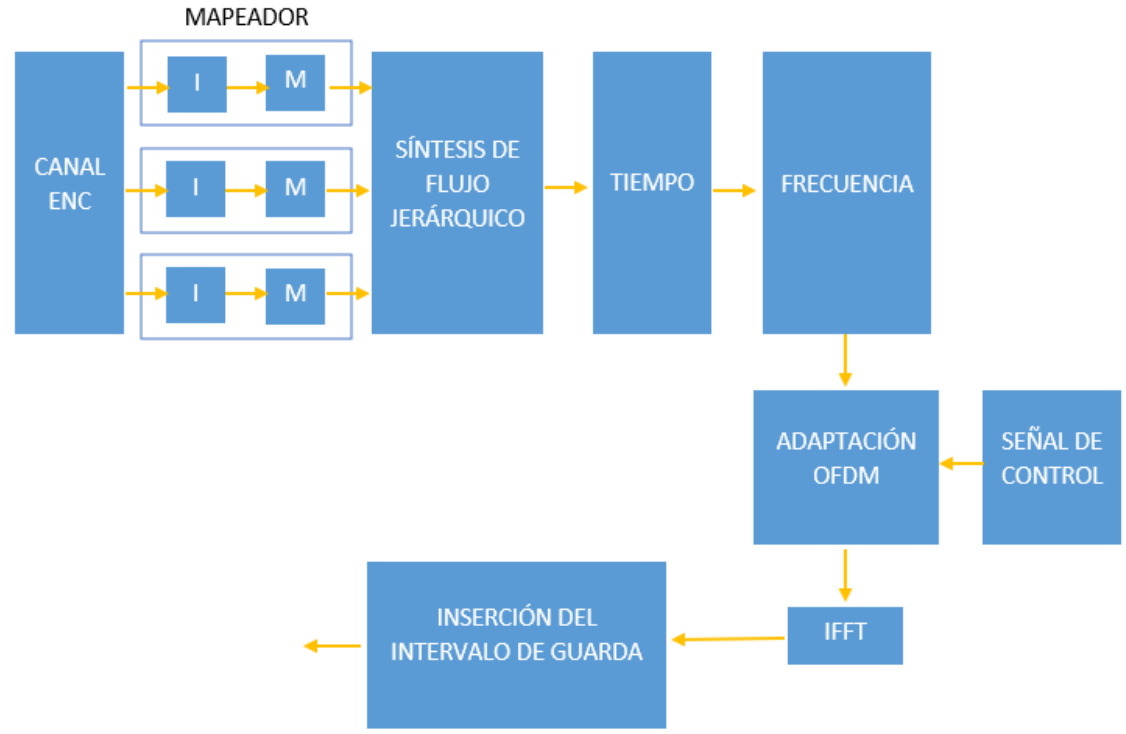
transmisión. Este proceso se realiza antes de aplicar la Transformada Rápida de Fourier Inversa (IFFT).

Para obtener el símbolo OFDM se agregan ciertas propiedades, como el tiempo de guarda y algunas banderas de control.

Una vez que se aplica IFFT, la señal resultante se encontrara en banda base teniendo así múltiples portadoras ortogonales (Jarrín A., Morejon G., & Bernal, 2010), tal como se muestra en la ***Figura 2***:

Figura 2

Proceso de modulación.



Codificación Interna FEC (*Forward Error Correction*)

Se trata de un proceso de corrección de errores con un proceso de codificación muy utilizada en los sistemas de tiempo real porque es la prioridad de analizar los datos transmitidos en el lugar de esperar su retransmisión.

La ventaja del corrector de errores FEC es que puede ahorrar energía durante el proceso de transmisión, y también se caracteriza por discriminar mensajes corruptos durante el proceso de recepción.

Una de las formas más efectivas de corregir errores es la técnica de bits de redundancia. Al trabajar en los procesos de envío y recepción, el decodificador recibe un solo paquete donde los bits de redundancia permiten reconocer la secuencia de datos original.

Codificación Externa (*Reed Solomon*)

El principio del corrector de errores Reed Solomon radica en agregar bytes de redundancia a un bloque de información logrando con esto una codificación de todo el bloque de información, muy utilizado sobre comunicaciones inalámbricas, móviles, entre otros, para el proceso de detección y corrección de errores a nivel de bytes (Sandoval Ruiz & Fedón, 2007).

Multiplexación

Este bloque de procesamiento tiene como objetivo principal ahorrar espectro de radiofrecuencia y transmitir señales digitales. Por tanto, dado que el propio sistema tiene ortogonalidad entre sus subportadoras, la eficiencia y robustez de la modulación, se utiliza

como esquema de transmisión (Cantos Sanchez, Tapuy Rendon, & Ramos Sanchez, 2014).

El sistema de multiplexación ha permitido el desarrollo el desarrollo de tecnologías como DVB-T e ISDB-T. La característica esencial del sistema es que puede trabajar en un canal de 6 MHz, lo que hace que el espectro de la señal sea más estrecho.

Debido a que en el estándar ISDB-Tb existe un bloque de frecuencias fundamentales, también llamado segmentos, este tiene un total de 14 segmentos, de los cuales solo 13 estarán disponibles ya que uno está dedicado a la distribución de bandas. Cabe señalar que los segmentos se distribuyen por el canal de 6 MHz.

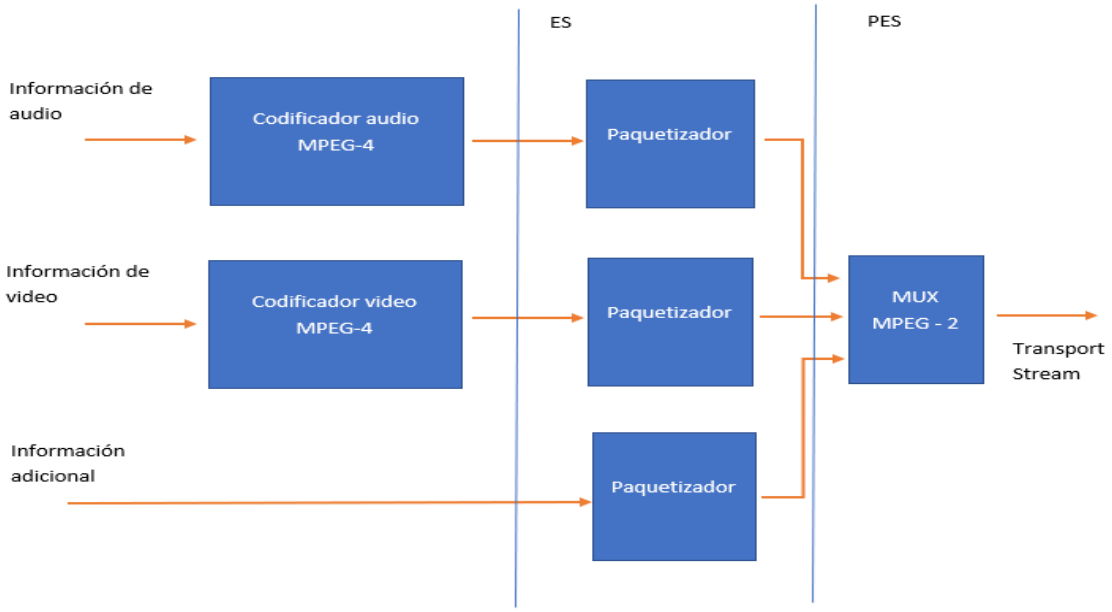
Flujo de Transporte TS

Una vez que se realiza la multiplexación de audio, video y datos interactivos mediante el proceso de compresión MPEG-4, se obtienen y procesan paquetes de 188 bytes denominados TS (Transport Stream) en el transmisor.

Como se muestra en la ***Figura 3***, los 4 bytes son de cabecera y el resto corresponde a la información del paquete TS.

Figura 3

Conformación del TS para una programación.



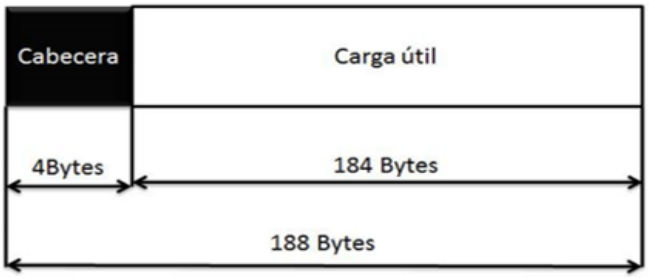
A continuación, se presentará la estructura de los paquetes TS detallando su estructura, cola, cabeceras y ciertos elementos esenciales como los PIDs con la finalidad de reconocer cuales son los campos que definen a un paquete TS.

Encabezado del paquete TS

La cabecera de un paquete TS se representa en 4 bytes como se muestra en la ***Figura 4*** de los cuales 13 bits corresponden al *Packet Identifier* (PID) el mismo que indica que tipo de información posee el cuerpo del paquete.

Figura 4

Cabecera de un paquete.



A continuación, se presenta las características principales de la cabecera de un paquete TS:

- Byte sincro: Este byte está destinado a mantener la sincronización de la información de entrada con el decodificador. Tiene el valor 0x47 en orden hexadecimal y es al mismo tiempo una característica única de los paquetes TS.
- Indicador de error de transporte: Este bit, también conocido como error de transporte, se activa tan pronto como el sistema registra un error en la etapa de transmisión.
- Indicador de arranque: Este bit, se activa tan pronto como el sistema registra que en la cabecera existe un paquete PES.
- PID (Packet Identifier): Es un arreglo de 13 bits que permite identificar que paquetes traen información útil y a su vez determinar los PIDs de flujos elementales ya estandarizados.
- Control de cifrado: Es un arreglo de 2 bits, también conocido como control de

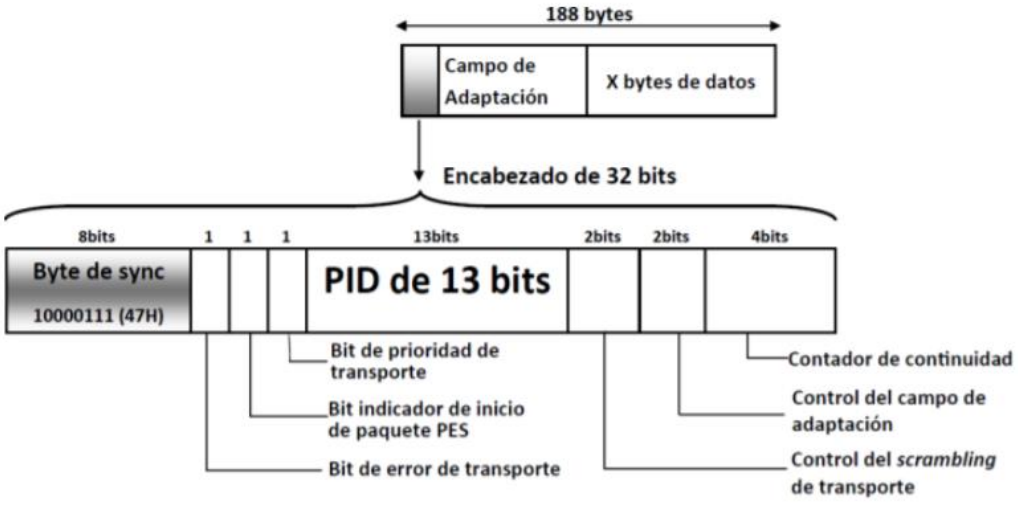
scrambling de transporte, permite identificar si existe cifrado de datos dentro de la carga útil de información.

- Control campo de adaptación: Es un arreglo de 2 bits, permite identificar si existe la presencia del campo de adaptación en la cabecera.
- Control de carga: Es un arreglo de 2 bits, que permite identificar si hay datos de carga útil.
- Contador de continuidad: Se incrementa en 1 cuando se cumple la condición de que la fuente envíe el mismo paquete. Mediante este proceso se intenta evitar errores.

En la ***Figura 5*** se muestra gráficamente la distribución de los bits correspondientes a la cabecera de un paquete TS.

Figura 5

Estructura de un paquete TS.



Nota: Obtenido de (Moncayo Flores & Pozo Caicedo, 2014).

En la ***Tabla 2*** se especifican los bits que posee la cabecera de una trama TS.

Tabla 2

Lista de campos existentes en la cabecera de un paquete TS.

Número de bits	Campos
8	Byte de sincronismo
1	Bit de error de transporte
1	Bit indicador de inicio de paquetes Pes
1	Bit de prioridad de transporte
13	PID
2	Control de *scrambling* de transporte
2	Control del campo de adaptación
4	Contador de continuidad

Campo de adaptación

Este campo se utiliza como relleno en el paquete de datos TS y su propósito es mantener constante la trama de 188 bytes, por lo que no tiene una longitud fija. La característica importante de este campo es que no se encuentra en todos los paquetes TS, pero cuando existe, tiene información importante para la sincronización a través de *Program Clock Reference* (PCR).

A continuación, se presenta las características principales del campo de adaptación:

- Adaptation field length. - Es un arreglo de 8 bits en donde si el valor de la matriz es 0, significa que solo hay 1 byte de relleno.
- Discontinuity indicator (indicador de discontinuidad). - Este bit indica que hay una discontinuidad en el campo de adaptación, cuando se activa indica que puede haber dos tipos de discontinuidades, la primera afecta al sistema de

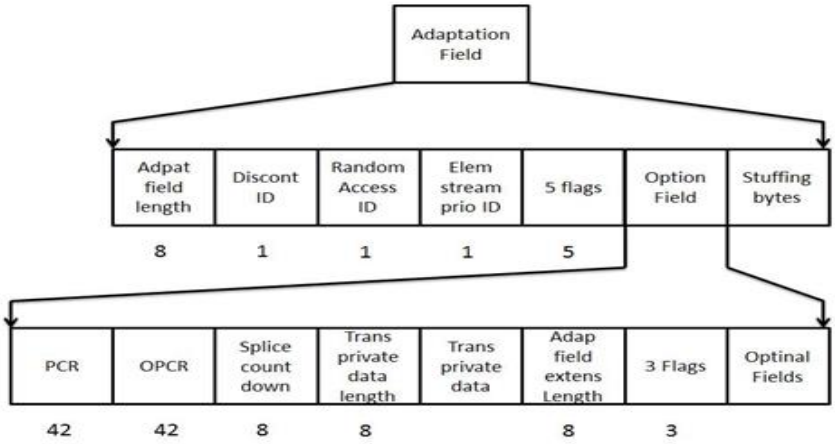
base de tiempos y la segunda al contador de continuidad.

- Random access indicator (indicador de acceso aleatorio). - Este bit se activará cuando el paquete de datos actual y el siguiente paquete de datos de la trama TS tengan el mismo PID.
- Elementary stream priority indicator. - Este bit se usa para identificar qué carga útil tiene prioridad sobre las demás.
- Flags. - Es un arreglo de 5 bits, que tiene como objetivo brindar redundancia al receptor ya que se estará ejecutando periódicamente.
- Stuffing bytes (bytes de relleno). - Es un arreglo de 8 bits, y su finalidad es mantener el equilibrio en el paquete utilizado como relleno, completando así los 188 bytes de la trama TS, lo que no afecta a la recepción, ya que el decodificador suele descartarlos sin ningún problema.
- PCR (Program Clock Reference). - Es un arreglo de 42 bits, está compuesto por 2 partes: la primera es un arreglo de 33 bits que porta el *program clock reference base* y la segunda es un arreglo de 9 bits que porta el program *clock reference extension*.
- OPCR (Original Program Clock Reference): Es un arreglo de 42 bits, y sirve de respaldo en la sincronización si existe algún fallo en PCR.

En la ***Figura 6*** se muestra gráficamente la distribución de los bits correspondientes al campo de adaptación.

Figura 6

Campos de adaptación.



Nota: Obtenido de *(Gutiérrez Tapia & Cochancela Alvear, 2013)*.

En la ***Tabla 3*** se especifican los bits que posee el campo de adaptación de una trama TS.

Tabla 3

Lista de campos existentes en el campo de adaptación de un paquete TS.

Número de bits	Campos
8	Adaptation Field Length
1	Discontinuity indicator
1	Random Access Indicator
1	Elementary Stream Priority Indicator
5	5 Flags
-	Optional Fields
8	Stuffing bytes

PID (Identificador de paquete)

Este campo permite la identificación de diferentes tipos de paquetes de datos transportados por la trama TS. La característica de este campo es que puede tomar 8192

valores según la información transportada en su carga útil. Cabe señalar que ya existe una asignación estándar, como el valor 8191 que representa los paquetes nulos en la trama (Villamarín, Olmedo, Lara, & Illescas, 2012).

En la ***Tabla 4*** se puede apreciar los valores que puede tomar el PID.

Tabla 4

Valores de PID.

Valor Hex	Valor Decimal	Asignación
0000	0	Tabla PAT
0001	1	Tabla CAT
0010	16	Tabla NIT
0011	17	Tabla BAT
0011	17	Tabla SDT
0012 **0026** **0027**	18 38 39	Tabla EIT
0014	20	Tabla TDT
0014	20	Tabla TOT
0000	0	Tabla ST
1FFF	8191	Paquetes Nulos

En la ***Tabla 5*** se muestran los códigos de PID permitidos usar para identificar paquetes de audio y video.

Tabla 5

PID permitidos para identificar la carga útil.

Código	Descripción
0 0000 0000 0000	Tabla de asignación de programas
0 0000 0000 0001	Tabla de acceso condicional
0 0000 0000 0010 **0 0000 0000 1111**	Reservado
0 0000 0001 0000 **1 1111 1111 1110**	Se pueden asignar como PID de la tabla de información de la red, la tabla de asignación de programas, para la identificación de audio o video.
1 1111 1111 1111	Identificador de paquetes.

Secciones

Las secciones constan de un encabezado, una carga útil y un código de redundancia o también se denominan detectores de errores (Granja Toledo, 2011).

A continuación, se presenta las características principales de la estructura de las secciones:

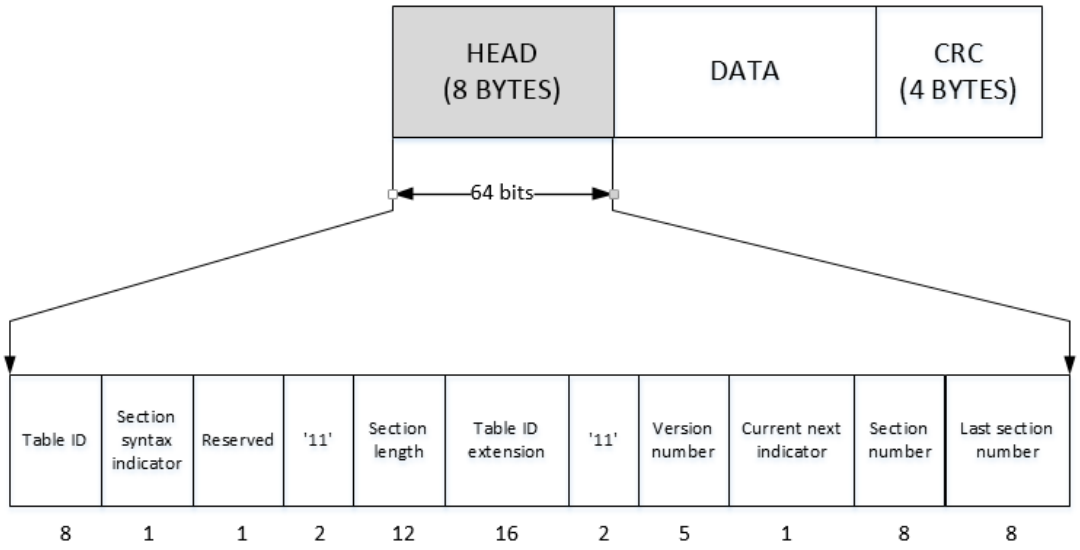
- Table ID. - Permite identificar a que tabla pertenece.
- Section syntax indicator - Si es extendido o normal, este bit permite identificar qué modo de funcionamiento se debe utilizar.
- Section length. - Es un arreglo de 12 bits se utiliza para determinar el tamaño de la sección en bytes.
- Table id extension. - Este apartado sirve como una opción de extensión del *Table_ID*.
- Version number. - Es un arreglo de 5 bits que se puede utilizar para identificar el número de sección.
- Current next indicator. - Este bit se activa cuando se determina que la tabla es válida. De lo contrario, se desactivará y la tabla se descartará.
- Section number. - Es un arreglo de 8 bits indica el numero de la sección en la tabla.
- Last section number. - Es un arreglo de 8 bits especifica el numero de la última sección de la tabla.

- CRC 32. – Es un arreglo de 32 bits que permite estar acorde a la normativa ISO/IEC 13818-1.

En la ***Figura 7*** se muestra gráficamente la estructura de las secciones.

Figura 7

Formato extendido de las secciones.



Nota: Obtenido de (Mancheno Arcos, 2015).

En la ***Tabla 6*** se especifican los bits que posee la estructura de secciones en una trama TS.

Tabla 6

Lista de campos existentes en la estructura de secciones.

Número de bits	Campos
8	Table ID
1	Section syntax indicator
1	Reserved
2	‘11’
12	Section lenght

16	Table ID extension
2	‘11’
5	Version number
1	Current next indicator
8	Section number
8	Last section number

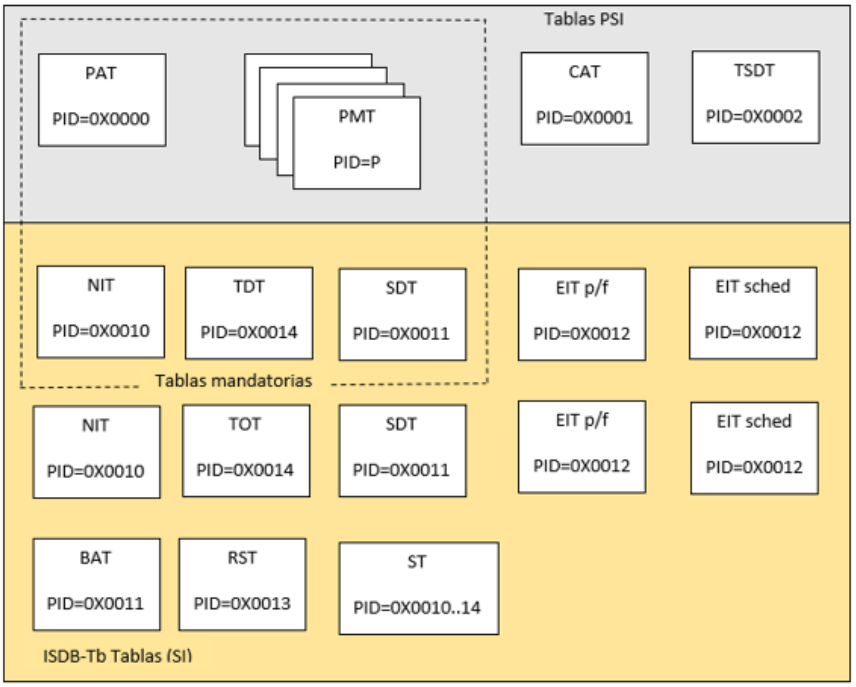
Tablas PSI/SI

Las tablas PSI / SI (información específica del programa / información de servicio) son estructuras que permiten la transmisión de información y se dividen en dos grupos: tablas privadas o tablas llamadas SI y tablas no privadas llamadas PSI.

En la ***Figura 8*** se muestra la estructura de las tablas PSI/SI.

Figura 8

Estructura de las Tablas PSI/SI.



Nota: Obtenido de *(468, 2019)*.

Una característica de las tablas PSI es que proporciona información sobre programas que se han multiplexado con MPEG-2. Hay algunas tablas que describen el contenido de los programas, p. Ej. B. PAT, PMT, CAT (Muñoz Vera, 2015).

En la ***Tabla 7*** se muestra los PID de algunas de las tablas PSI más destacadas.

Tabla 7

Tablas PSI.

PID	Nombre de la Tabla	Table ID
$\mathbf{0000_H}$	*PAT (Program Association)*	00_H
Asignado por la PAT	*PMT (Program Map)*	02_H
$\mathbf{0010_H}$	*NIT (Network Information)*	$40_H\ y\ 41_H$

En la ***Tabla 8*** se muestra los PID de algunas de las tablas SI más destacadas.

Tabla 8

Tablas SI

PID	Nombre de la Tabla	Table ID
$\mathbf{0011_H}$	BAT *(Bouquet Association)*	$4A_H$
$\mathbf{0010_H}$	NIT *(Network Association)*	$40_H, 41_H$
$\mathbf{0011_H}$	SDT *(Service Descriptor)*	$42_H, 46_H$
$\mathbf{0012_H, 0026_H, 0027_H}$	EIT *(Event Information)*	$4E_H, 4F_H, 50_H$
$\mathbf{0014_H}$	TDT *(Time Date)*	70_H
$\mathbf{0014_H}$	TOT *(Time Offset)*	73_H
$\mathbf{000_H}$	ST *(Stuffing)*	72_H

Tabla PAT (*Program Association Table*)

La función de la tabla PAT se centra en asignar un PID a cada programa con el fin de registrar y consolidar los paquetes que componen el PMT. Cabe destacar que el PID asignado para la tabla PAT es 0x00.

También se conoce como Tabla de asignación de programas y una de sus características es que debe transmitirse cada 100ms ya que se puede dividir en hasta 255 secciones (Enríquez Chicaiza & Rivadeneira Obregón, 2018).

A continuación, se presentan las características principales de la tabla PAT:

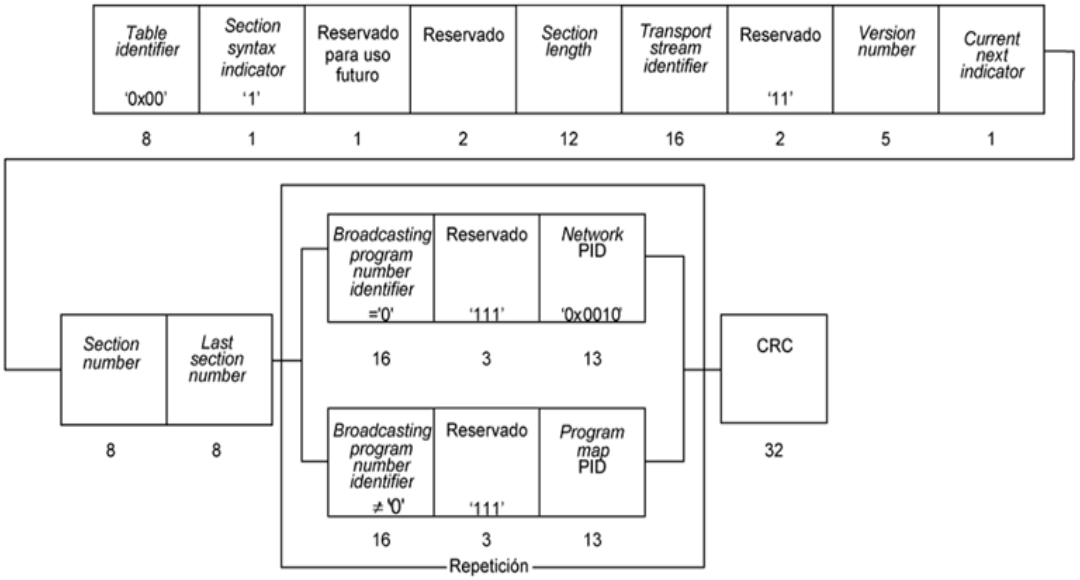
- Table identifier. - Es un arreglo de 8 bits, su valor predeterminado es 0x00.
- Section syntax indicator. - Este bit siempre permanece activo, porque en este modo permite el uso del formato extendido de la sección.
- Section length. - Es un arreglo de 12 bits, donde los 2 primeros bits representan al indicador de tabla y los 10 bits sobrantes indican el número de bytes.
- Transport stream ID. - Es un arreglo de 16 bits y su función es proporcionar un código de identificación a TS.
- Version number. - Es un arreglo de 5 bits que indica la versión de la tabla PAT.
- Current next indicator. - Este bit indica si la tabla PAT es válida, si la tabla no es válida, el sistema no avanzará.
- Program number. - Es un arreglo de 16 bits que indica a qué PID se aplicará el número de programa de la siguiente tabla PMT.
- Network PID. - Es un arreglo de 13 bits destinado a identificar la red y está presente cuando el valor del *program_number* es 0x00.
- Program map PID. - Es un arreglo de 13 bits indica el PID de las tablas PMT.

- CRC 32. - Es un arreglo de 32 bits que sirve como corrector de errores ya que agrega redundancia.

En la ***Figura 9*** se muestra gráficamente la distribución de los bits correspondientes a la tabla PAT.

Figura 9

Tabla PAT.



Nota: Obtenido de *(15603-1, 2007)*.

En la ***Tabla 9*** se especifican los bits que posee la tabla PAT.

Tabla 9

Lista de campos existentes en la tabla PAT.

Número de bits	***Campos***
8	Table Identifier '0x00'
1	Section syntax indicator '1'
1	Reservado para uso futuro

2	Reservado
12	Section length
16	Transport stream identifier
2	Reservado ‘11’
5	Version number
1	Current next indicator
8	Section number
8	Last section number
16	Broadcasting program number identifier
3	Reservado ‘111’
13	*Network PID / Program map PID*
32	CRC

Tabla PMT (*Program Map Table*)

La tabla PMT contiene información sobre todos los programas que se transmitieron a través del flujo TS. Este último puede transportar el mismo número de tablas PMT que los programas. Hay dos intervalos de tiempo en los que se debe transmitir la tabla PMT. La duración de la transmisión es de 100 ms en el servicio fijo y 200 ms en el servicio móvil (Enríquez Chicaiza & Rivadeneira Obregón, 2018).

A continuación, se presenta las características principales de la tabla PAT:

- Table identifier. - Es un arreglo de 8 bits al que se le ha asignado el valor 0x02 y que también especifica el número de la tabla.
- Broadcating program number identifier. - Es un arreglo de 16 bits que indica el número de programa al que se aplica la tabla PMT.
- Section number. - Es un arreglo de 8 bits y, dado que solo se debe transmitir una parte, se asigna el valor 0x00.
- PCR PID. – Es un arreglo de 13 bits que indica el PID de los paquetes TS

que deben contener el campo PCR válido

- Stream type. - Es un arreglo de 8 bits que indica el tipo de programa del elemento contenido en los paquetes como se muestra en la ***Tabla 10***.

- Elementary PID. - Es un arreglo de 13 bits que representa y distingue al TS, desglosando cuál de ellos transporta audio, video o datos.

Tabla 10

Tipos de flujos elementales.

Valor	***Descripción***
0x00	Reservado
0x01	ISO/IEC 11172 Video
0x02	Video conforme la recomendación de la ITU H.262
0x03	ISO/IEC 11172 Audio
0x04	Audio conforme a ISO/IEC 13818-3
0x05	*ITU-T Rec. H.222.0* | *ISO/IEC 13818-1.* Sección privada
0x06	Paquete PES que contiene información privada
0x15-0x7F	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reservado
0x80-0xFF	Uso privado

En la ***Figura 10*** se muestra gráficamente la distribución de los bits correspondientes a la tabla PMT.

En la ***Tabla 11*** se especifican los bits que posee la tabla PMT.

Tabla 11

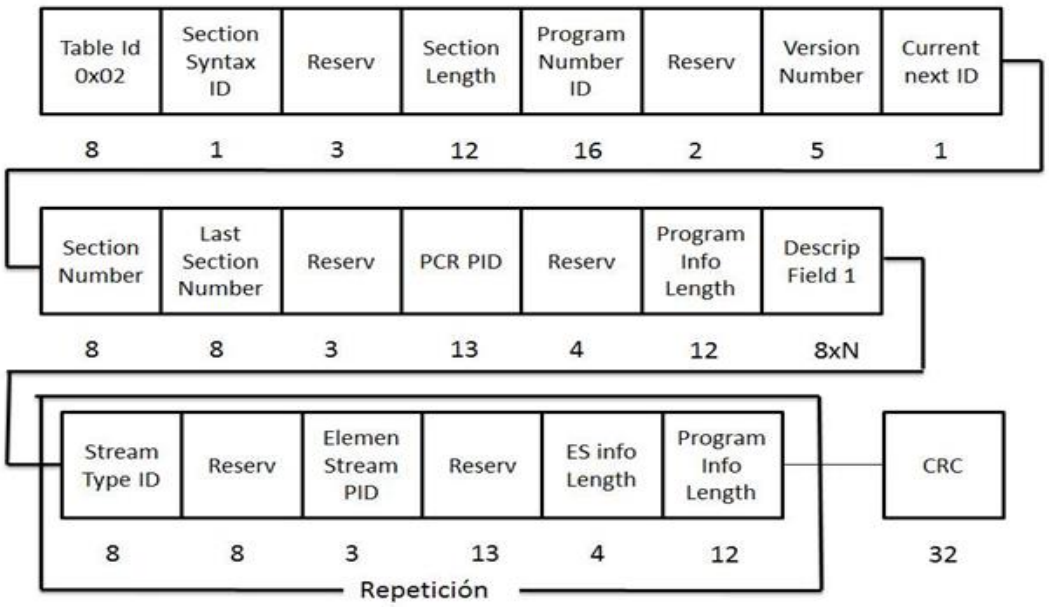
Lista de campos existentes en la tabla PMT.

Número de bits	***Campos***
8	Table Identifier ‘0x02’
1	Section syntax ID
3	Reservado
12	Section length
16	Program Number ID
2	Reservado
5	Version number

1	Current next ID
8	Section number
8	Last section number
3	Reservado
13	PCR PID
4	Reservado
12	Program Info Length
8xN	Descrip Field 1
8	Stream Type ID
8	Reservado
3	Elemen Stream PID
13	Reservado
4	ES Info Length
12	Program Info Length
32	CRC

Figura 10

Tabla PMT.



Nota: Obtenido de *(15603-1, 2007)*.

Tabla NIT (*Network Information Table*)

La tabla NIT contiene los datos y propiedades de la organización física de los multiplexores, que se combinan en una red común, también porque se requiere sintonización de canales, razón por la cual la tabla NIT se transmite cada 10 segundos (Enríquez Chicaiza & Rivadeneira Obregón, 2018).

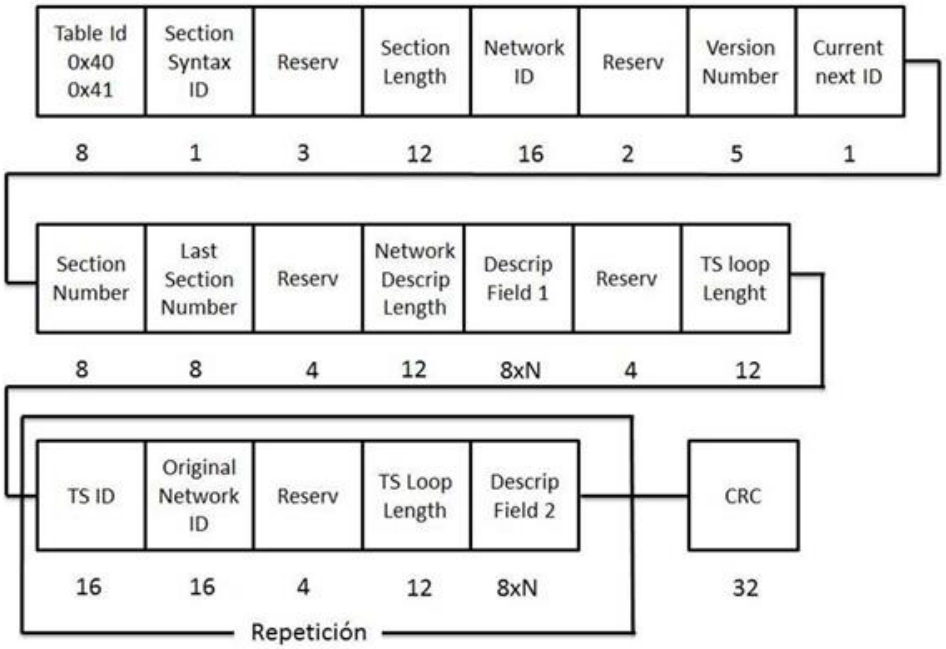
A continuación, se presenta las características principales de la tabla NIT:

- Table identifier. - Es un arreglo de 8 bits al que se le ha asignado el valor 0x40 y que también especifica el número de la tabla.
- Network descriptor length. - Es un arreglo de 12 bits que se utiliza para definir el tamaño (bytes) del descriptor que se ejecutará a continuación.
- Transport stream loop length. - Es un arreglo de 12 bits que se utiliza para definir el tamaño (bytes) del campo cuando se ejecuta desde la primera trama TS hasta el primer byte del CRC.
- Transport stream ID. - Es un arreglo de 16 bits que permite distinguir el flujo TS de los otros flujos del sistema de transmisión.
- Original network ID. - Es un arreglo de 16 bits el cual indica la red de origen del transport stream.
- Transport descriptors length. - Es un arreglo de 12 bits que se utiliza para definir el tamaño (bytes) del descriptor de transport stream.

En la ***Figura 11*** se muestra gráficamente la distribución de los bits correspondientes a la tabla NIT.

Figura 11

Tabla NIT.



Nota: Obtenido de *(15603-1, 2007)*.

En la ***Tabla 12*** se especifican los bits que posee la tabla NIT.

Tabla 12

Lista de campos existentes en la tabla PMT.

Número de bits	***Campos***
8	Table Identifier ‘0x40’
1	Section syntax ID
3	Reservado
12	Section length
16	Network ID
2	Reservado
5	Version number
1	Current next ID
8	Section number
8	Last section number

4	Reservado
12	Network Descrip Length
8xN	Descrip Field 1
4	Reservado
12	TS loop Length
16	TS ID
16	Original Network ID
4	Reservado
12	TS loop Length
8xN	Descrip Field 2
32	CRC

Re-multiplexor

La re-multiplexación se lleva a cabo mediante un proceso adaptativo, en el que TS tiene las funciones de transmisión por capas y recepción parcial (Pisciotta, 2010).

El proceso de entrelazado del TS de entrada y la obtención de un único flujo binario (también llamado BTS) se denomina re-multiplexación. Entre sus principales características destacan los siguientes puntos:

- Se aplica el método de redundancia añadiendo 16 bytes nulos en la cola de los paquetes TS.
- Los paquetes de BTS tienen un tamaño total de 204 bytes.
- Posee una velocidad binaria constante mediante la inserción de nulos los cuales no afectan a las capas jerárquicas del paquete (Cantos Sanchez, Tapuy Rendon, & Ramos Sanchez, 2014).

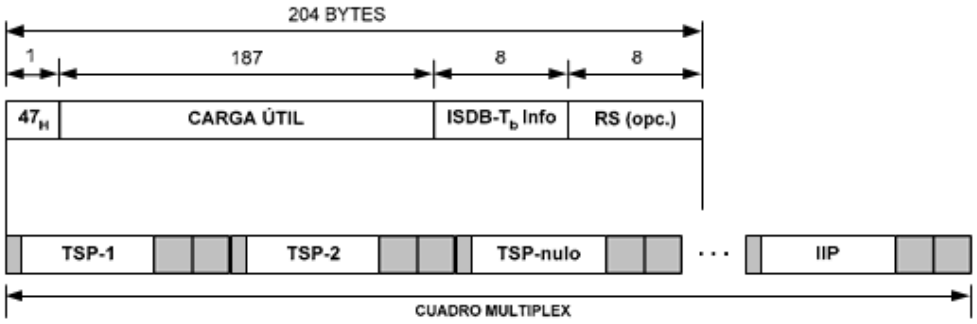
Flujo de transporte para broadcast BTS

El flujo BTS contiene la información binaria total a transmitir y ser transmitida, todo esto a la salida del multiplexor, que en ciertos casos agrega redundancia a los TSP para que la transmisión de datos sea constante.

En la ***Figura 12*** se muestra gráficamente la distribución de paquetes del flujo BTS.

Figura 12

Estructura de los paquetes del flujo BTS.



Nota: Obtenido de *(Pisciotta, 2010)*.

Un paquete TSP está conformado por 204 bytes los mismos que se encuentran distribuidos de la siguiente forma:

- 1 byte destinado para la sincronización.
- 187 bytes conformados por carga útil.
- 8 bytes de información del estándar.
- 8 bytes opcionales.

Existe la presencia de un canal denominado *Transmission Multiplexing Configuration Control* (TMCC) este canal transporta información adicional, la misma que contiene sus propias características.

En la ***Tabla 13*** se muestra los diferentes modos que puede adoptar un paquete TSP.

Tabla 13

Número de TSP.

Modo	Intervalo de guarda			
	1/4	***1/8***	***1/16***	***1/32***
2k	1280	1152	1088	1056
4k	2560	2301	2176	2112
8k	5120	4608	4352	4224

Capítulo IV

Desarrollo del Software

Introducción

En este capítulo se detalla el desarrollo del software denominado Compresor de TS, el cual permite comprimir un archivo de extensión .ts, el mismo que contiene toda la información a transmitir. El algoritmo desarrollado también permite realizar la expansión tanto de una trama TS como una trama BTS mediante el análisis de la cabecera que portan los diferentes paquetes.

Se utilizó NetBeans IDE como entorno de desarrollo libre para la construcción del software y lenguaje de programación Java el mismo que se ejecuta sobre este entorno, además de que brinda más ventajas y es compatible con la mayoría de sistemas operativos.

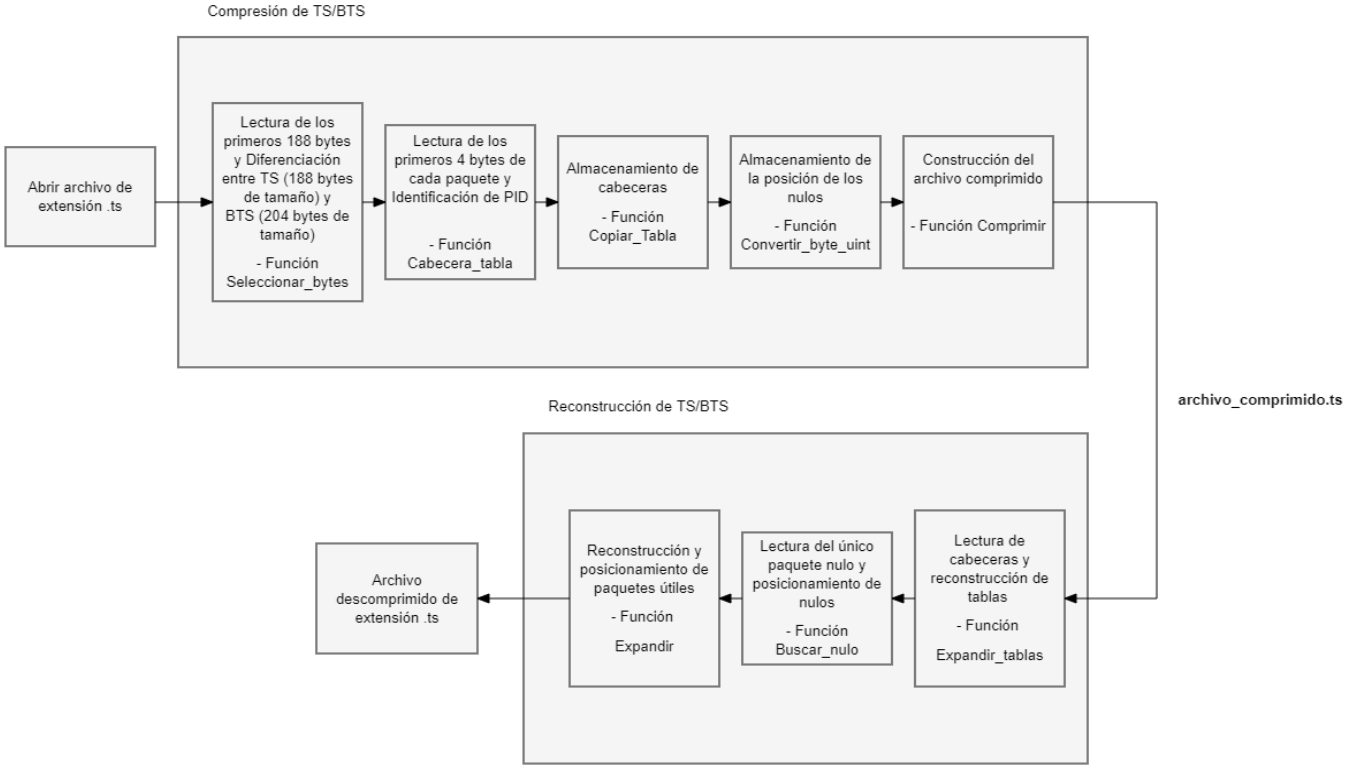
El algoritmo de compresión se ejecuta de manera automática identificando por su cuenta si el archivo .ts es una trama transport stream o una trama broadcast transport stream lo que lo hace amigable con el usuario y a la vez autónomo al momento de ejecución.

El propósito de utilizar el lenguaje de programación Java para el desarrollo del algoritmo es que al tener información útil distribuida en varios paquetes de la trama TS y BTS, se los puede manipular, almacenar sin problemas y de esta forma analizar de manera completa a los flujos de transporte.

Diagrama de bloques del Software

Figura 13

Diagrama de bloques del algoritmo.



En la ***Figura 13*** se presenta el diagrama de bloques del algoritmo de compresión y expansión detallando las funciones que se ejecutaran durante todo el proceso.

A continuación, se describe la función de cada bloque, de izquierda a derecha:

Compresión de TS/BTS

Se genera un archivo ejecutable llamado ALTS_COMPRIMIR.jar, para que el programa sea más fácil de usar.

Función Seleccionar_bytes

En esta función se hace la lectura de los primeros 188 bytes que posee un paquete correspondiente a una trama TS, con esto la función busca identificar el valor 0x47 en la posición 188, si se detecta este valor en dicha posición se concluye que el archivo de extensión .ts es un transport stream de lo contrario se asume que estamos trabajando con un broadcast transport stream.

Una vez identificado que tipo de trama se está analizando se procede a la lectura de las cabeceras de los diferentes paquetes, es así que se especifica un rango de 4 bytes a leer, los cuales corresponden a la cabecera de un paquete.

La información contenida en los 4 bytes antes mencionados se la convierte a bits, dentro del algoritmo se representa mediante un *string* de nombre *bits* que se muestra en la ***Tabla 14***, el cual representa un valor binario.

Tabla 14

Métodos de la función Seleccionar_bytes.

Método	***Función***
String bits ( )	Es una cadena de caracteres.

Función Cabecera_tabla

Al tener en cuenta que la cabecera de un paquete inicia en el byte 1 y termina en el byte 4, esta función se encarga de analizar la información contenida en dichos bytes para consecuentemente identificar el PID de la cabecera.

Esta función utiliza arreglos del tipo ArrayList, que en particular no tienen un tamaño definido, es decir, se pueden agregar valores del tipo *string* o *integer*. Esto permite definir un arreglo de nombre *ts_bts*, en donde se añade información específica mediante el método *.add*.

Una vez identificado el PID se almacena y el valor que representa en binario se lo convierte a entero.

En la ***Tabla 15*** se muestra algunos de los métodos utilizados dentro del algoritmo.

Tabla 15

Métodos de la función Cabecera_tabla.

Método	**Función**
ArrayList ts_bts < >	Es un arreglo de tamaño indefinido.
. add	Agrega información en los ArrayList.

Función Copiar_Tabla

El objetivo de esta función es identificar específicamente los PID correspondientes a las tablas PMT, PAT, NIT, TOT y SDT, ya que el algoritmo está condicionado a lo siguiente: si se detecta una tabla conocida se almacena su cabecera y se borra el cuerpo de la tabla, esto quiere decir que de 188 bytes se eliminaran 184 para el caso de una trama TS, en el caso de una trama BTS se eliminaran 200 bytes, este proceso se realiza con el fin de optimizar la compresión y reducir aún más el tamaño del archivo comprimido. Al final se almacena una tabla completa de cada una de las tablas conocidas.

Función Convertir_byte_uint

En esta función se convierten a enteros sin signo las posiciones de nulos que se encuentran en bytes, para esto se hace uso de 2 vectores los cuales se muestran en la ***Tabla 16***, además se agrega un contador el cual verifica si existe un cambio de signo.

El objetivo de esta función es ir comparando el valor de la posición actual con la anterior y así evidenciar si existe un cambio de signo un ejemplo: si una posición es 255 y la siguiente es 0, si existe ese cambio de signo el contador *tmpcont* aumenta en 1 caso contrario se mantiene.

La siguiente ecuación nos permite obtener las posiciones de cada nulo:

$$y = x + (255 \times pos) + pos$$

Tabla 16

Variables de la clase Convertir_byte_uint.

Variables	Función
Nulos_sin_sig [ ]	Entero sin signo.

Nulos_sin_sig_conv [ ]	Entero sin signo convertido.
Tmpcont [ ]	Contador
y	Posición convertida a entero sin signo.
x	Valor en bytes.
pos	Valor de las posiciones.

Función Comprimir

En esta función se hace la lectura y búsqueda del PID 8191 para poder ir creando el nuevo archivo de extensión .ts, si el PID no corresponde a un nulo se almacena todo el paquete ya que significa que porta información útil caso contrario se almacena la posición del nulo detectado.

Se define un arreglo del tipo *ByteArrayOutputStream*, en donde se ira almacenando bytes, utilizando los métodos .*seek* que permite mover el puntero dentro de un archivo*, .read* que indica cuantos bytes se van a leer del archivo y *.write* que permite escribir dentro de un archivo*.*

Después del proceso antes mencionado se crea un nuevo archivo comprimido que estará distribuido de la siguiente forma: paquetes útiles, cabeceras de tablas, un paquete nulo, la posición de los nulos y los cuerpos de las tablas.

En la ***Tabla 17*** se muestra algunos de los métodos utilizados dentro del algoritmo.

Tabla 17

Métodos de la clase Comprimir.

Método	Función
ByteArrayOutputStream [ ]	Arreglo de solo Bytes
. seek	Permite mover el puntero dentro de un archivo.
. read	Indica cuantos bytes se van a leer del archivo original.
. write	Permite escribir dentro de un archivo.

A continuación, se detalla como estará distribuida la información dentro del archivo comprimido .ts:

- Cabeceras (4 bytes).
- Flujos elementales (188 bytes).
- Un paquete nulo.
- Posición de los nulos.
- Tablas *PMT, PAT, SDT, NIT, TOT.*

Reconstrucción de TS/BTS

Se genera un archivo ejecutable llamado ALTS_EXPANDIR.jar, para que el programa sea más fácil de usar.

Función Expandir_tablas

El proceso que se lleva a cabo en esta función es la lectura constante de 4 bytes con el fin de encontrar las distintas cabeceras que contiene el archivo.

Si la función detecta un PID de una tabla, por ejemplo: se detecta la cabecera de la tabla PAT el algoritmo se va al final del archivo que está haciendo lectura identifica la tabla y copia su cuerpo en el lugar correspondiente. Este proceso culmina cuando el algoritmo ya no detecta más PID de tablas conocidas.

Función Buscar_nulo

Esta función se dedica a la búsqueda del único paquete nulo que posee el archivo comprimido de extensión .ts mediante una lectura completa de la trama.

Una vez detectada la posición del paquete nulo, la función retorna un *long* de nombre *position_null* indicando dicha información.

En la ***Tabla 18*** se muestra el tipo de arreglos utilizados en esta función.

Tabla 18

Métodos de lá clase Buscar_nulo.

Método	Función
Long position_null ( )	Es un entero de mayor tamaño.

Función Expandir

Esta función comienza a leer la trama completa e identifica archivos útiles como audio, video y datos interactivos, pero también las posiciones de los nulos. La función desplaza los paquetes útiles al mismo tiempo que coloca los nulos en el lugar que les corresponde, de esta forma se logra descomprimir toda la información que posee el archivo comprimido.

Diagramas de flujo

Todas las funciones y clases descritas en el capítulo, para que puedan operar de manera efectiva, deben seguir un proceso que oriente su comportamiento. A continuación, se presentan los diagramas de flujo que indican la secuencia de ejecución de los algoritmos de compresión y expansión.

- Compresión. – En la ***Figura 14*** se muestra el diagrama de flujo que permite identificar si el archivo de extensión .ts es una trama TS o BTS, almacenar la cabecera de las tablas y después las posiciones de los nulos.
- Expansión de tablas. – En la ***Figura 15*** se muestra el diagrama de flujo que permite reconstruir las tablas a partir de la lectura de sus cabeceras.
- Expansión. – En la ***Figura 16*** se muestra el diagrama de flujo que permite recuperar todos los valores originales de la trama TS o BTS.

Figura 14

Diagrama de flujo del Compresor.

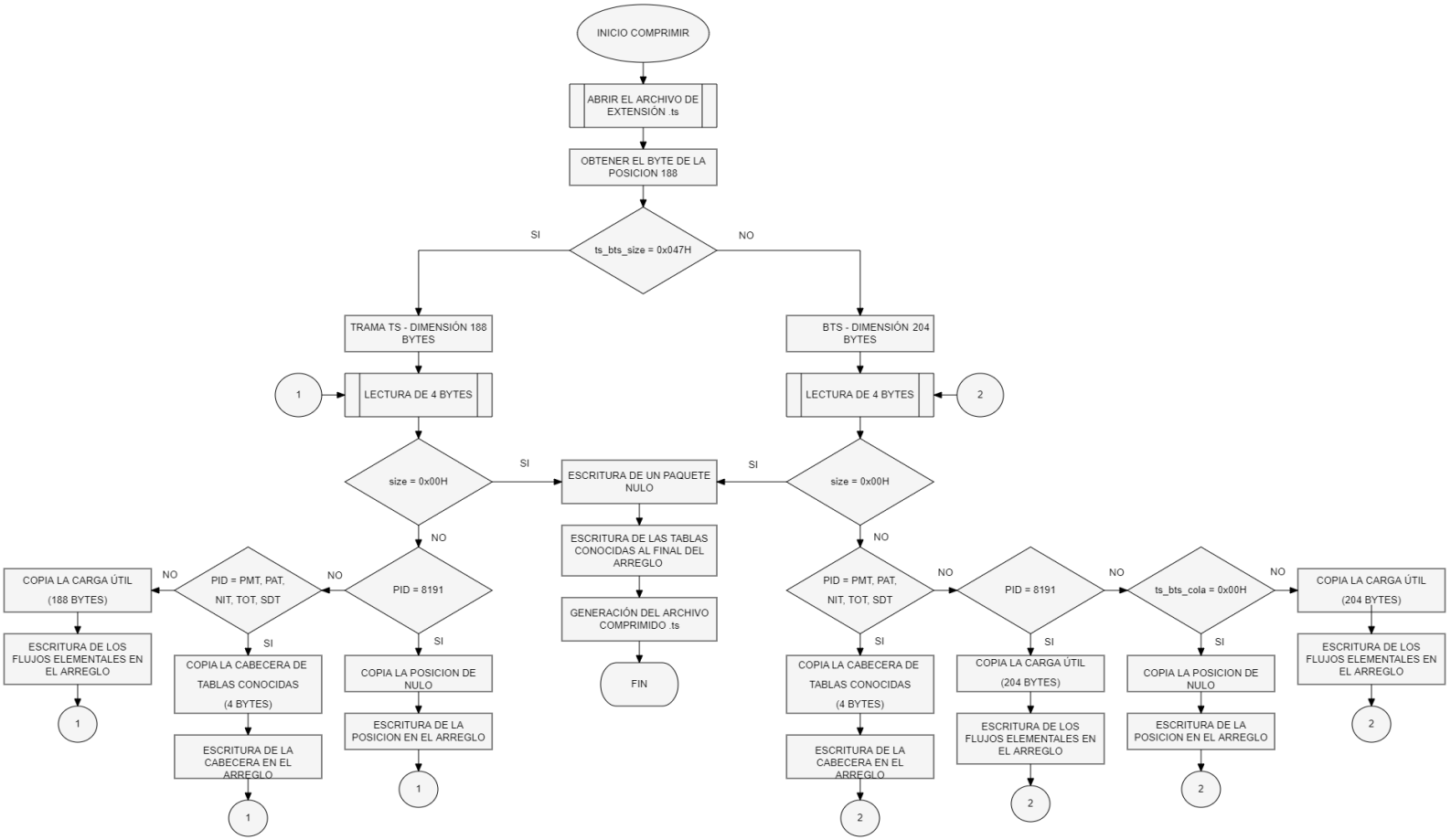


Figura 15

Diagrama de flujo de Expansor_tablas.

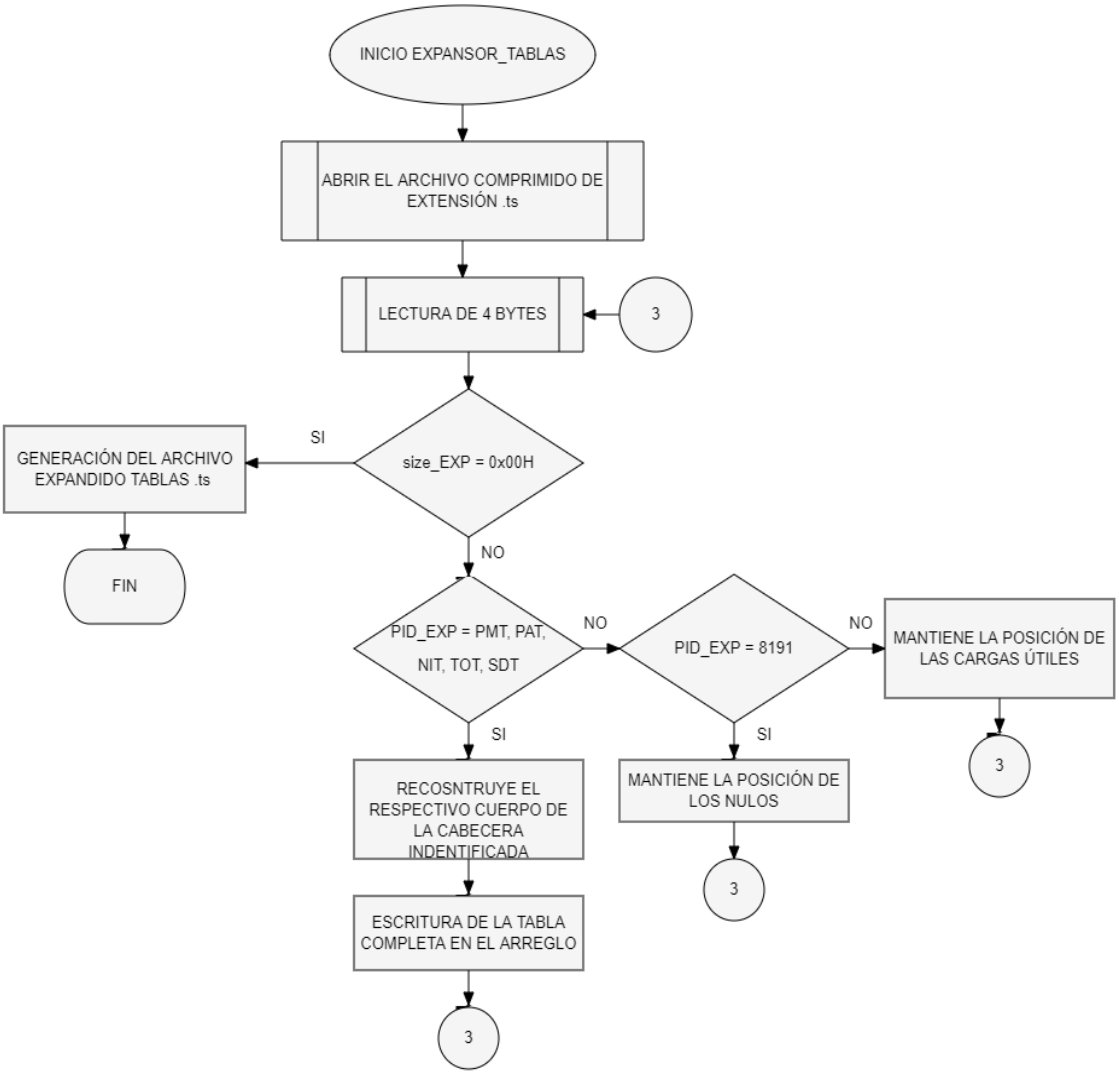
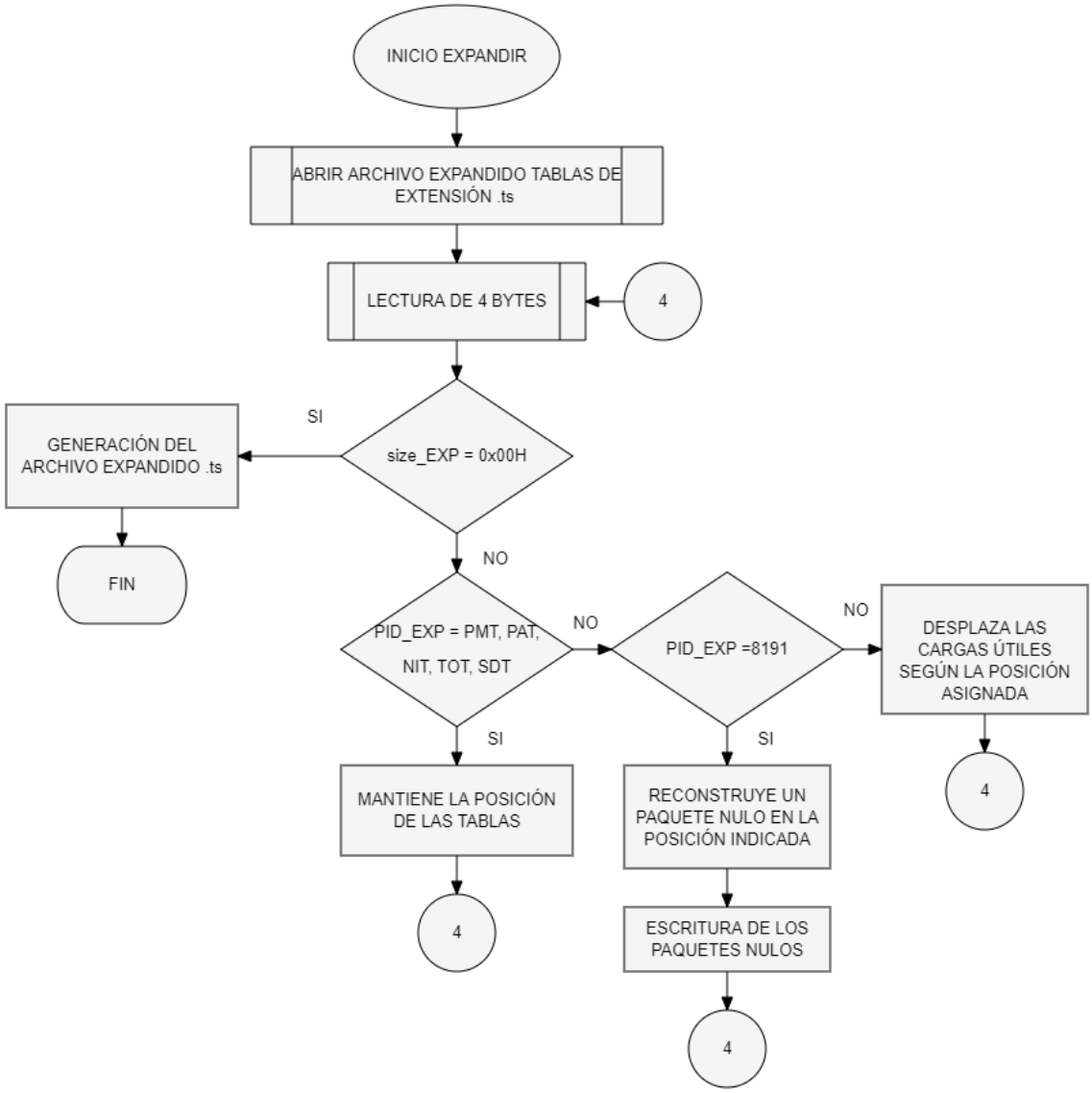


Figura 16

Diagrama de flujo del Expansor.



Capítulo V

Pruebas y Resultados

Generalidades

Una vez finalizado el desarrollo del algoritmo de compresión, se realizó la fase de prueba, para lo cual se evaluó el software frente a diferentes archivos .ts con audio, video y datos interactivos para probar la efectividad del software. Después de procesar los resultados, se utilizó el software gratuito FFMPEG para realizar una comparativa de las características originales de las tramas TS y BTS versus las características de las tramas TS y BTS expandidas.

FFMPEG

Es un software libre que contiene una variedad de funciones desde modificar la calidad de audio y video hasta la opción de *streaming*, una de sus particularidades es que se ejecuta mediante la consola de comandos y es compatible con varios sistemas operativos.

FFMPEG posee varios componentes como:

- ffmpeg
- ffserver
- ffplay
- ffprove

Y las librerías más comunes son:

- libavcodec
- libavformat
- libavutil
- libpostproc
- libswscale

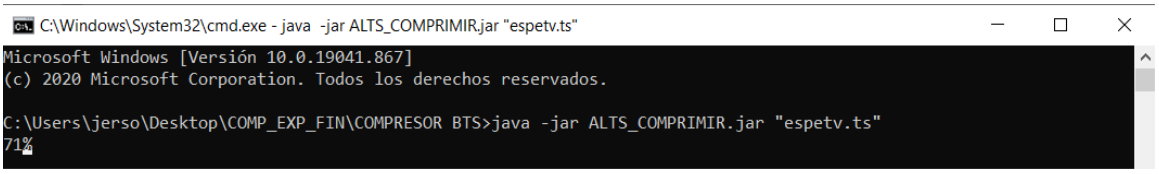
Transport Stream

Compresión

En primera instancia se define el archivo .ts a comprimir y se lo traslada a la carpeta junto al ejecutable ALTS_COMPRIMIR.jar para su debido funcionamiento. Se procede abrir la ventana de comandos CMD en la cual se visualizará el porcentaje de ejecución del programa como se muestra en la ***Figura 17.***

Figura 17

Cmd de la compresión de una trama TS.

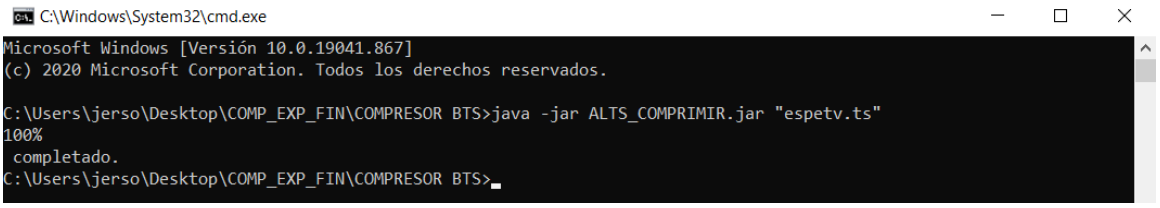


Al culminar el proceso de compresión con éxito se mostrará en la ventana de comando CMD la palabra completado, caso contrario marcará un error ya sea por falta de paquetes o por la carencia de un archivo .ts a comprimir.

En la ***Figura 18*** se puede visualizar el mensaje de completado una vez culminada la compresión.

Figura 18

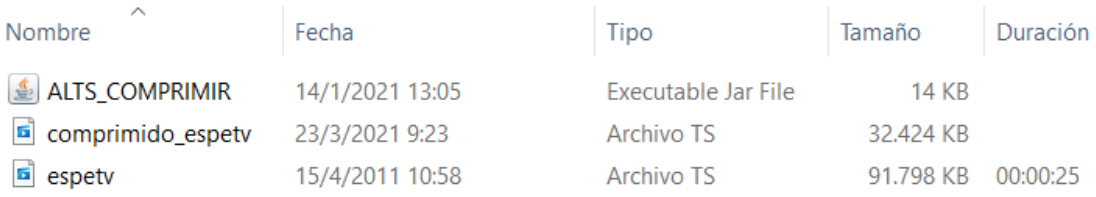
Cmd de la compresión TS completado.



A continuación, se muestra el tamaño del archivo espetv.ts original y el archivo comprimido_espetv.ts, en donde claramente se puede evidenciar que el tamaño se ha reducido a más de la mitad de su tamaño original.

Figura 19

Detalles del archivo TS original y comprimido.



En la ***Tabla 19*** se muestra el tamaño de los archivos después de la compresión.

Tabla 19

Tamaño de los archivos utilizados en la compresión.

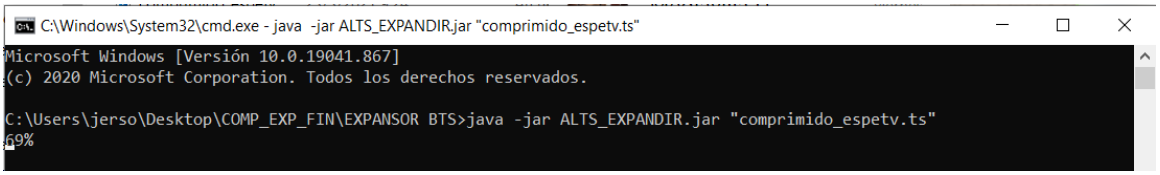
Nombre del archivo	Tamaño
Espetv.ts	91.798 KB
Comprimido_espetv.ts	32.424 KB

Expansor tablas

Una vez se obtenga el archivo .ts comprimido se lo traslada a la carpeta junto al ejecutable ALTS_EXPANDIR.jar para su debido funcionamiento. Se procede abrir la ventana de comandos CMD en la cual se visualizará el porcentaje de expansión de tablas como se muestra en la ***Figura 20.***

Figura 20

Cmd de la expansión de una trama TS.



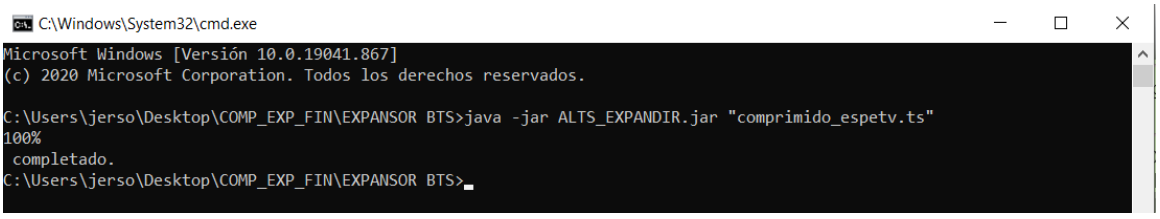
Expansor

Al culminar el proceso de expansión con éxito se mostrará en la ventana de comando CMD la palabra completado, caso contrario marcará un error ya sea por falta de paquetes o por la carencia de un archivo .ts a expandir.

En la ***Figura 21*** se puede visualizar el mensaje de completado una vez culminada la expansión.

Figura 21

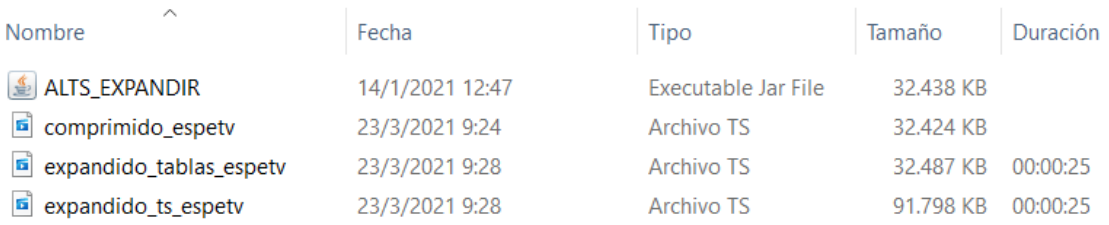
Cmd de la expansión TS completado.



En la ***Figura 22*** se muestra el tamaño del archivo comprimido_espetv.ts, expandido_tablas_espetv.ts y el archivo expandido_espetv.ts, en donde claramente se puede evidenciar que el tamaño ha aumentado al mismo valor que la trama TS original.

Figura 22

Detalles del archivo TS original, expandido_tablas y expandido.



En la ***Tabla 20*** se muestra el tamaño de los archivos después de la expansión.

Tabla 20

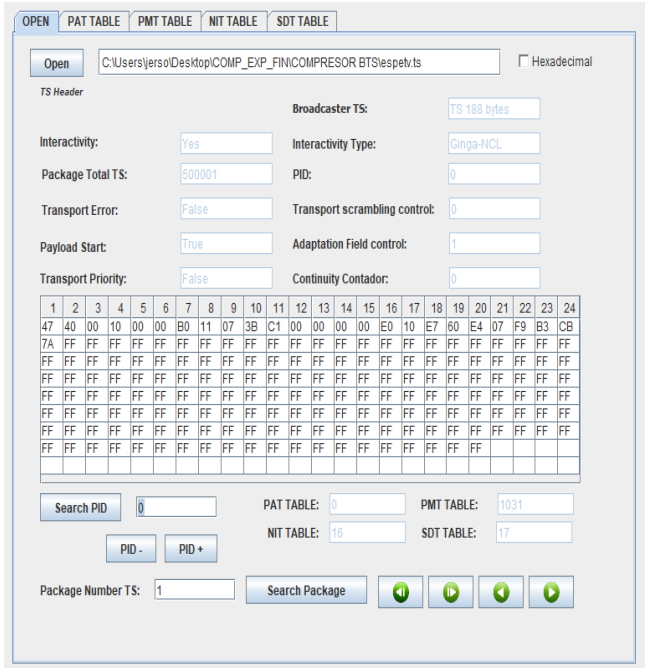
Tamaño de los archivos utilizados en la expansión.

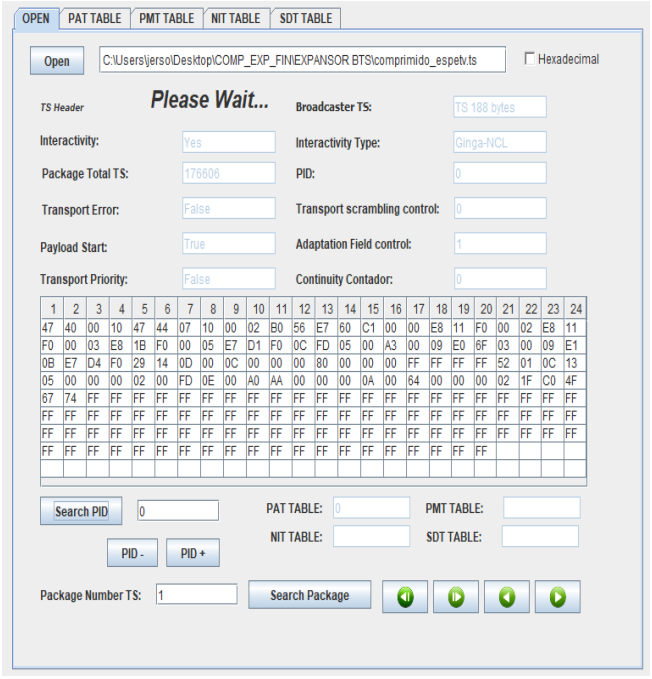
Nombre del archivo	Tamaño
Comprimido_espetv.ts	32.424 KB
Expandido_tablas_espetv.ts	32.487 KB
Expandido_ts_espetv.ts	91.798 KB

A continuación, se tiene una comparativa del TS original vs el TS comprimido dándonos a ver que el tamaño de los paquetes antes y después son de 188 bytes, el número total de paquetes del archivo comprimido ha disminuido debido a que ahora se posee únicamente las cabeceras de las tablas y las posiciones de los nulos además de las cargas útiles.

Figura 23

Comparativa entre el TS original vs el TS comprimido resultante.

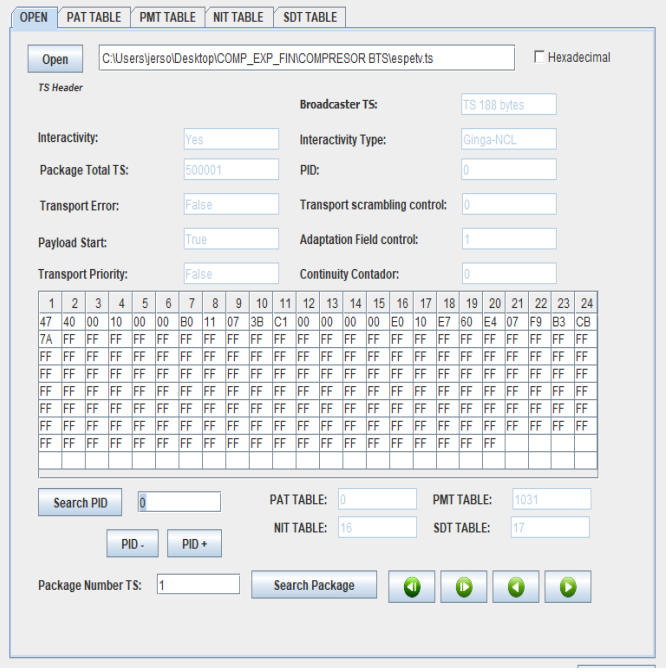


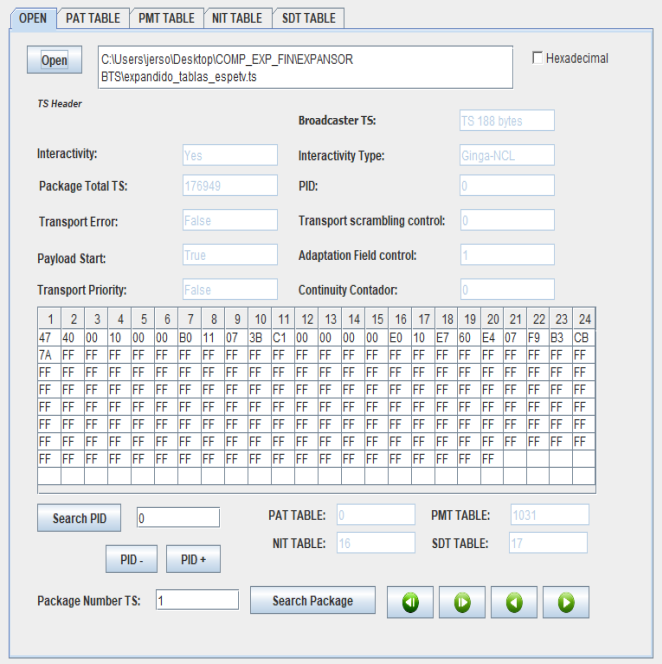


Se presenta una comparativa del TS original vs el TS expandido_tablas dándonos a ver que el tamaño de los paquetes antes y después son de 188 bytes, el número total de paquetes del archivo expandido_tablas sigue siendo menor al total de paquetes del archivo original, pero con la variación de que las tablas ya están reconstruidas en su totalidad.

Figura 24

Comparativa entre el TS original vs el TS expandido_tablas resultante.





C:\Users\jerso\Desktop\COMP_EXP_FIN\EXPANSOR BTS\expandido_tablas_espetv.ts

☐ Hexadecimal

TS Header

Broadcaster TS: TS 188 bytes

Interactivity: Yes

Interactivity Type: Ginga-NCL

Package Total TS: 176949

PID: 0

Transport Error: False

Transport scrambling control: 0

Payload Start: True

Adaptation Field control: 1

Transport Priority: False

Continuity Contador: 0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
47	40	00	10	00	00	B0	11	07	3B	C1	00	00	00	00	E0	10	E7	60	E4	07	F9	B3	CB
7A	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF				

Search PID: 0

PAT TABLE: 0

PMT TABLE: 1031

NIT TABLE: 16

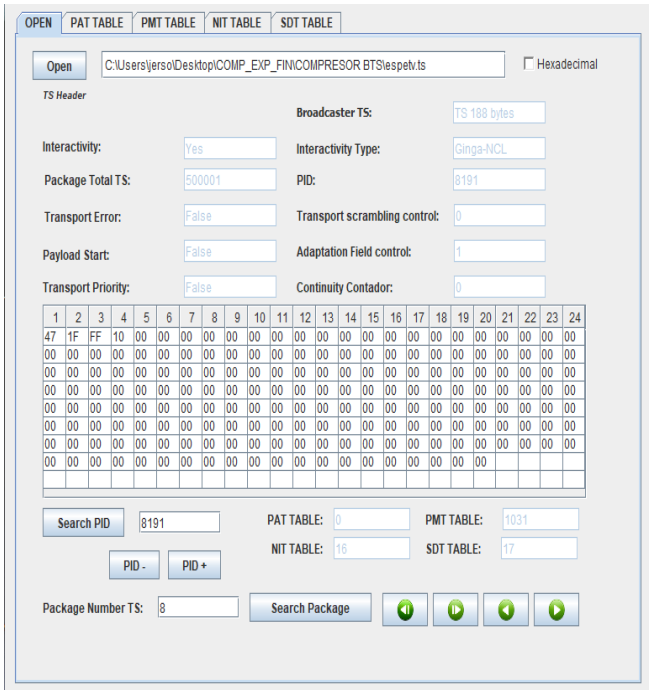
SDT TABLE: 17

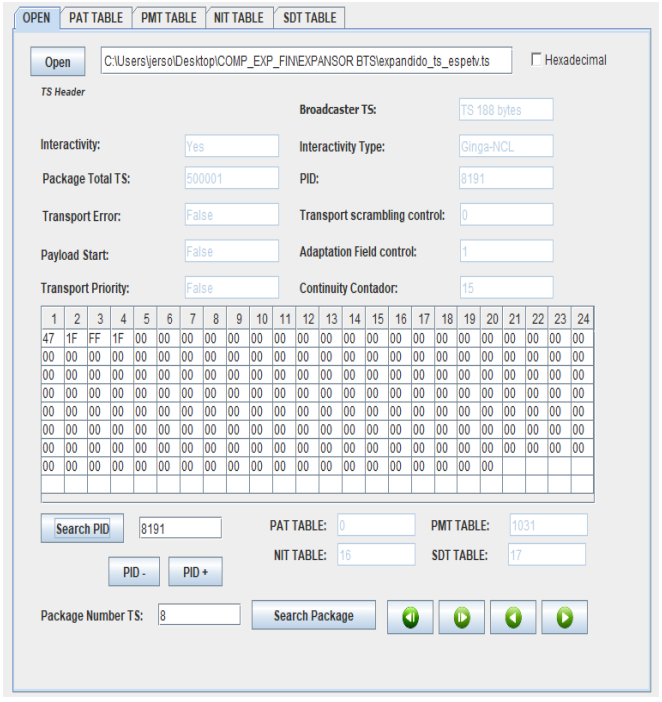
Package Number TS: 1

En la ***Figura 25*** se tiene una comparativa del TS original vs el TS expandido dándonos a ver que el tamaño de los paquetes antes y después son de 188 bytes, el número total de paquetes no ha cambiado, y el contenido del paquete correspondiente al PID 8191 en ambos casos los datos no cambian.

Figura 25

Comparativa entre el TS original vs el TS expandido resultante.

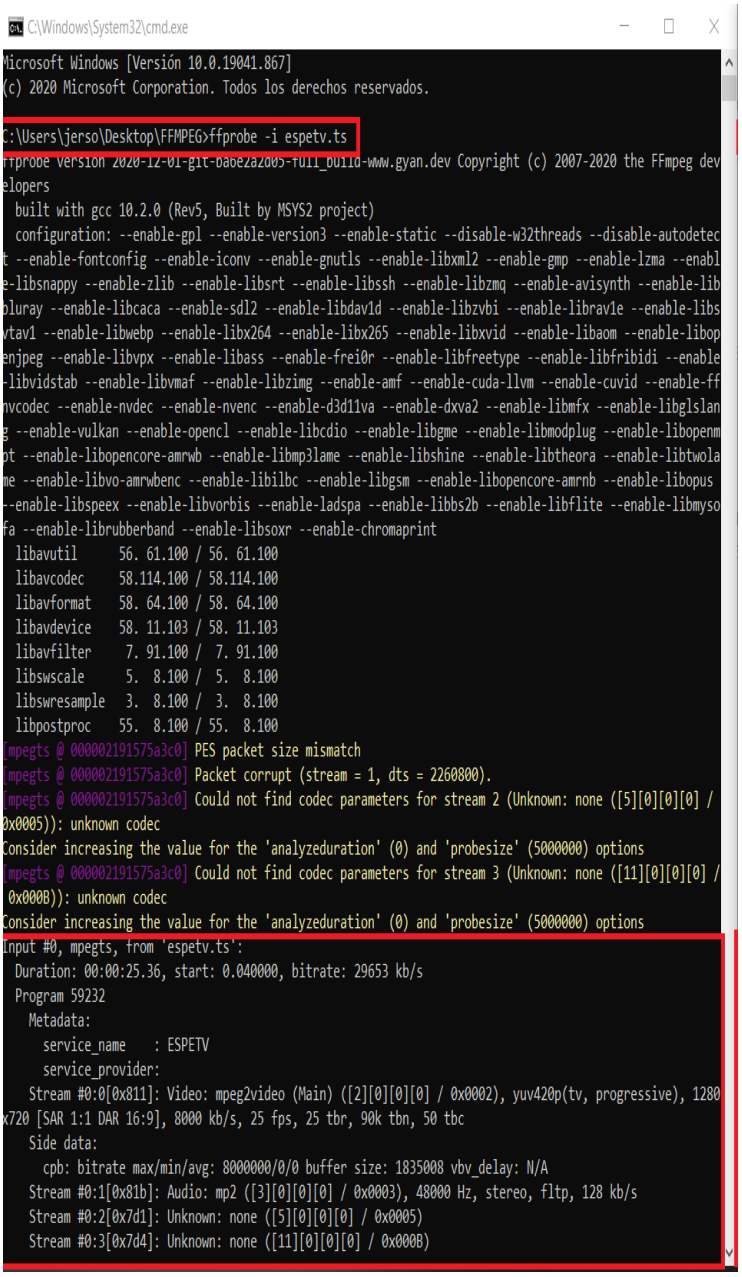


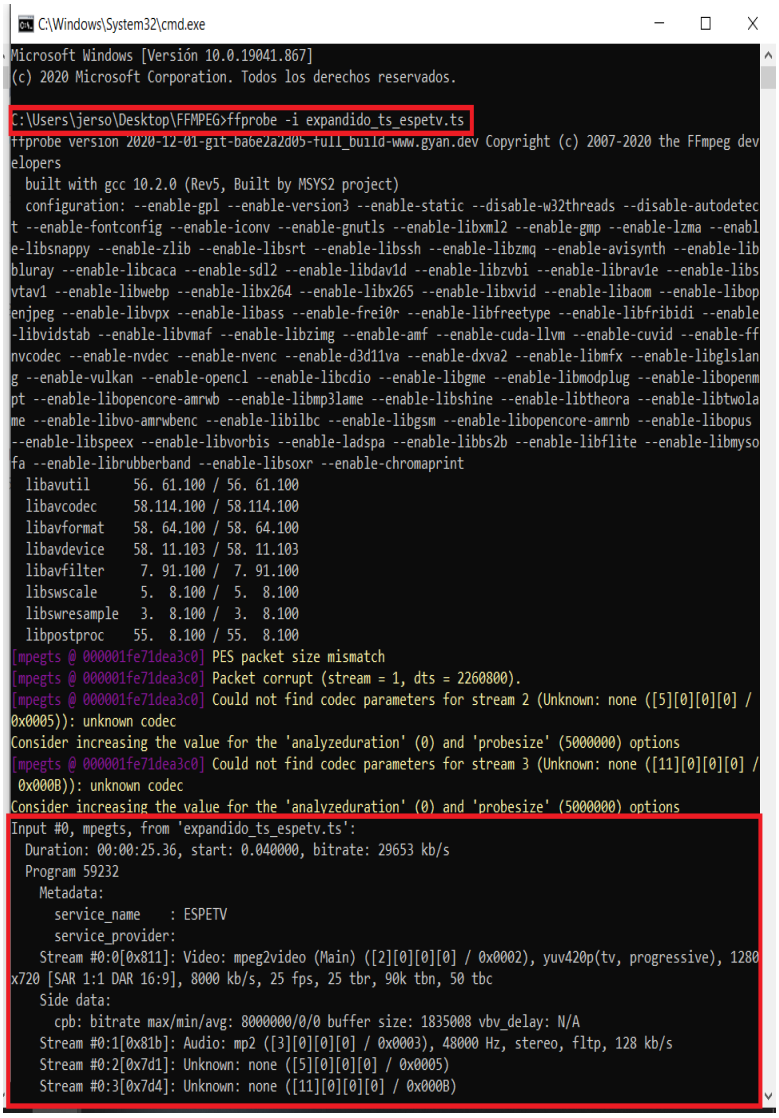


En la ***Figura 26*** se presenta el análisis del TS original vs el TS expandido mediante el software FFMPEG el mismo que nos muestra que todos los parámetros han sido recuperados como la diferente codificación de audio, video, datos interactivos, tasa de transmisión y duración entre otros.

Figura 26

Comparativa entre el TS original vs el TS expandido resultante en FFMPEG.





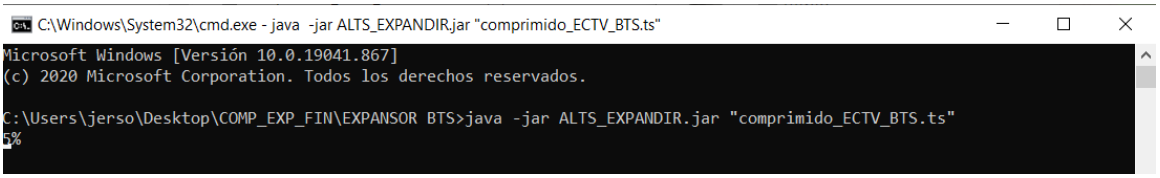
Broadcast transport stream

Expansor

Una vez se obtenga el archivo .ts comprimido se lo traslada a la carpeta junto al ejecutable ALTS_EXPANDIR.jar para su debido funcionamiento. Se procede abrir la ventana de comandos CMD en la cual se visualizará el porcentaje de expansión de como se muestra en la ***Figura 27.***

Figura 27

Cmd de la expansión de una trama BTS.

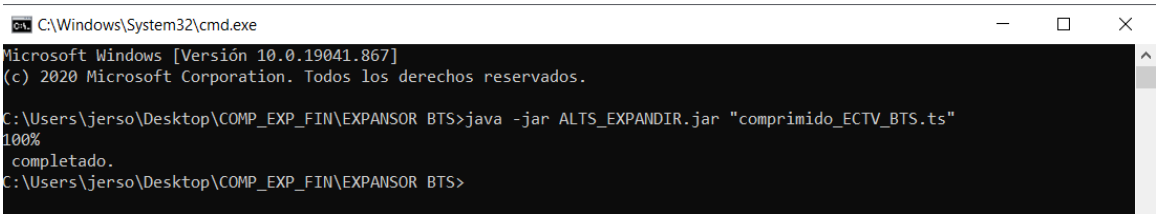


Al culminar el proceso de expansión con éxito se mostrará en la ventana de comando CMD la palabra completado, caso contrario marcará un error ya sea por falta de paquetes o por la carencia de un archivo .ts a expandir.

En la ***Figura 28*** se puede visualizar el mensaje de completado una vez culminada la expansión.

Figura 28

Cmd de la expansión BTS completado.

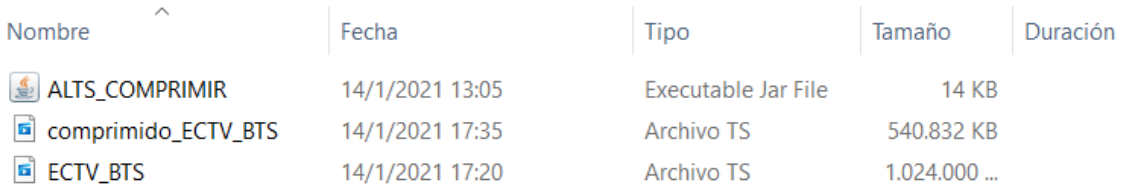


En la

Figura 29 se muestra el tamaño del archivo ECTV_BTS.ts original y el archivo comprimido_ECTV_BTS.ts, en donde claramente se puede evidenciar que el tamaño se ha reducido a la mitad de su tamaño original.

Figura 29

Detalles del archivo BTS original y comprimido.



En la ***Tabla 21*** se muestra el tamaño de los archivos después de la compresión.

Tabla 21

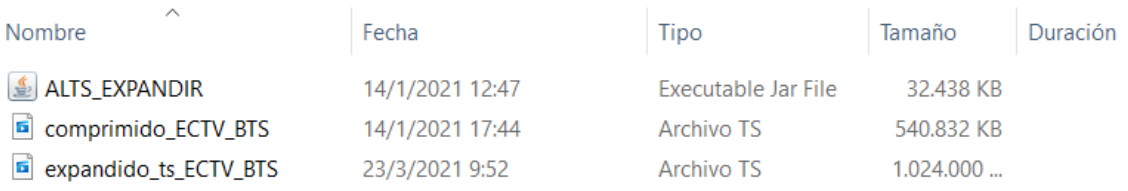
Tamaño de los archivos utilizados en la compresión.

Nombre del archivo	Tamaño
ECTV_BTS.ts	1.024.000 KB
Comprimido_ ECTV_BTS.ts	540.832 KB

A continuación, se muestra el tamaño del archivo comprimido_ECTV_BTS.ts y expandido_ts_ECTV_BTS.ts, en donde claramente se puede evidenciar que el tamaño ha aumentado al mismo valor que la trama BTS original.

Figura 30

Detalles del archivo BTS original y expandido.



En la ***Figura 22*** se muestra el tamaño de los archivos después de la expansión.

Tabla 22

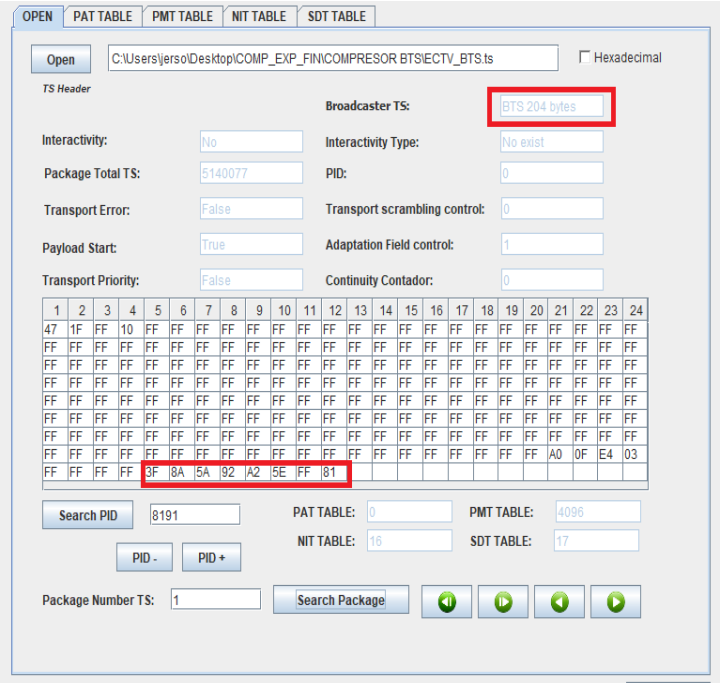
Tamaño de los archivos utilizados en la expansión.

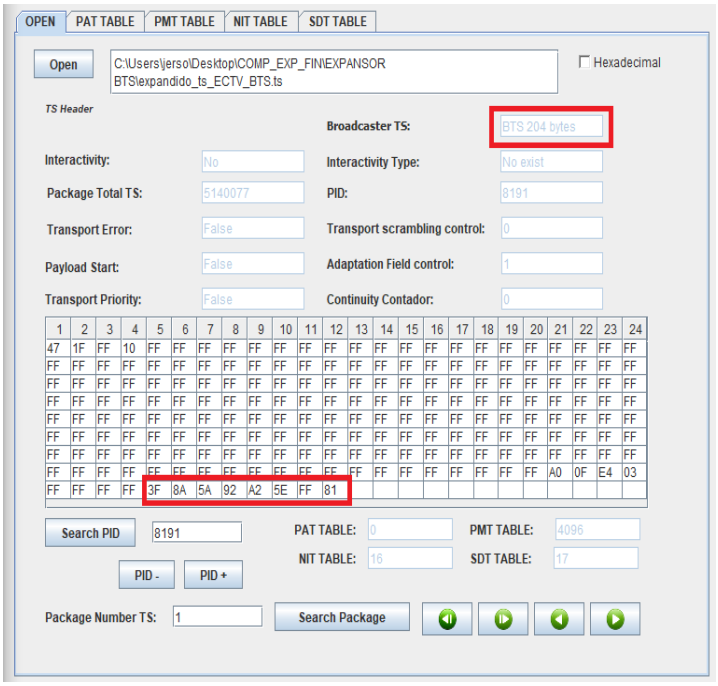
Nombre del archivo	Tamaño
Comprimido_ECTV_BTS.ts	540.832 KB
Expandido_ts_ECTV_BTS.ts	1.024.000 KB

Se tiene una comparativa del BTS original vs el BTS expandido dándonos a ver que el tamaño de los paquetes antes y después son de 204 bytes, el número total de paquetes no ha cambiado, y el contenido del paquete correspondiente al PID 8191 en ambos casos los datos no cambian.

Figura 31

Comparativa entre el BTS original vs el BTS expandido resultante.

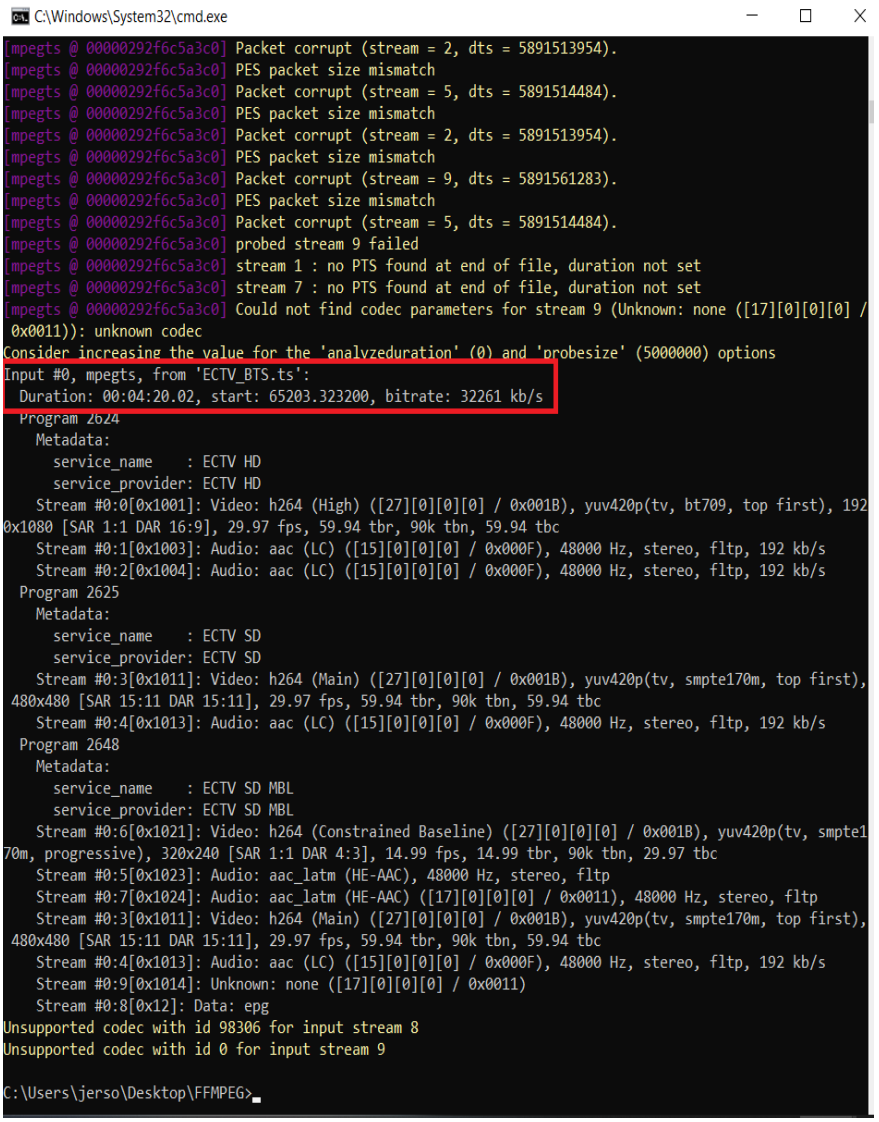




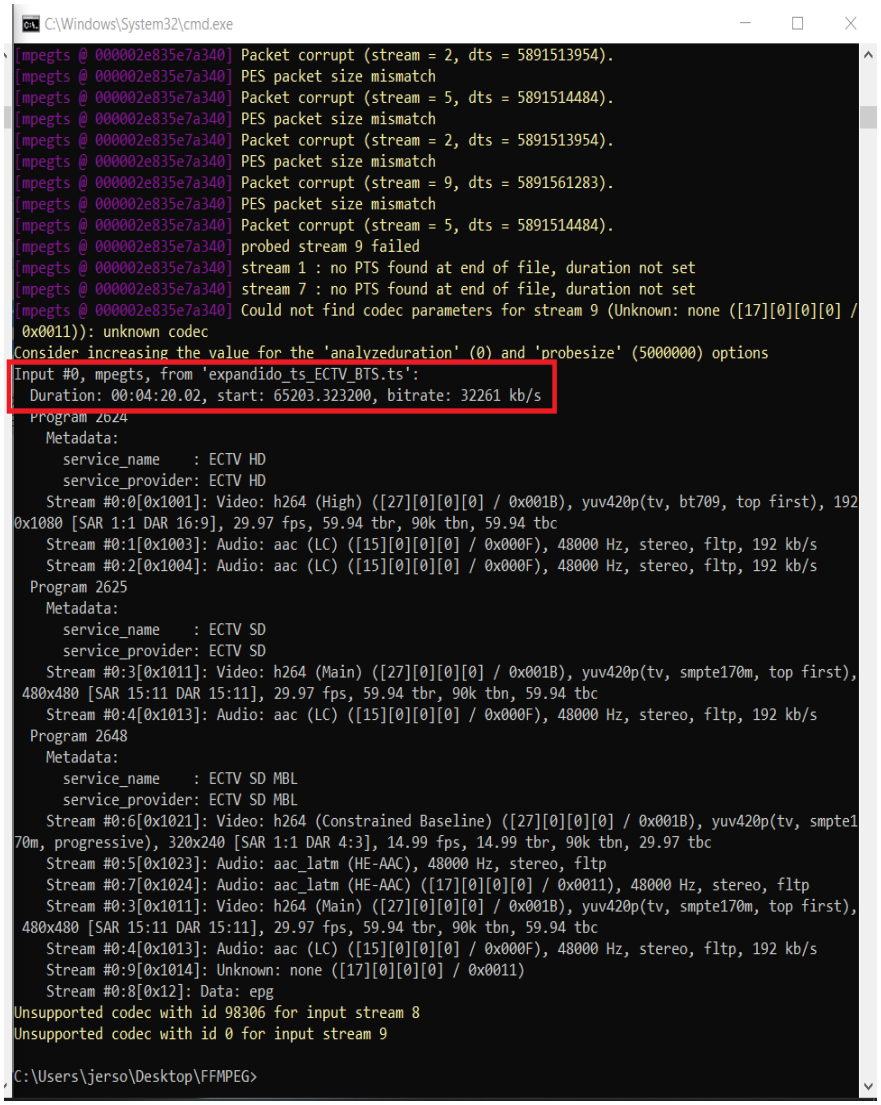
Se presenta el análisis del BTS original vs el BTS expandido mediante el software FFMPEG el mismo que nos muestra que todos los parámetros han sido recuperados como la diferente codificación de audio, video, datos interactivos, tasa de transmisión y duración entre otros.

Figura 32

Comparativa entre el BTS original vs el BTS expandido resultante en FFMPEG.



Cabe recalcar que la reproducción de los archivos expandidos tiene la misma calidad y velocidad que su archivo original.



En vista que se logra recuperar la trama TS y BTS sin ninguna alteración, queda comprobado que los programas tanto de compresión como de expansión son funcionales para cualquier tipo de archivo de extensión .ts.

Capítulo VI

Conclusiones y Recomendaciones

Conclusiones

- Se desarrolló un algoritmo de compresión y expansión para la trama TS y BTS mediante el uso del lenguaje de programación Java, bajo el estándar ISDB-T de televisión digital terrestre.
- Se desarrolló un algoritmo autónomo ya que puede identificar qué tipo de trama se tiene a la entrada todo esto antes de realizar cualquier acción.
- Se obtuvo un analizador de 4 bytes para la identificación de los diferentes PID asociados a los flujos elementales como lo son audio, video o datos interactivos, tablas, nulos y cabeceras.
- Se obtuvo el valor binario de cada posición de los nulos existentes dentro del *transport stream*, de esta manera se facilita el posicionamiento al momento de expandir el archivo previamente comprimido, en caso de tener un *broadcast transport stream* no solo se almacena la cabecera sino también la cola del paquete ya que en los últimos 16 bytes de un paquete nulo en algunos casos trae información útil la cual no se puede omitir.
- Se obtuvo el valor binario de cada uno de los PID correspondientes a flujos elementales, tablas y posiciones de los nulos es de esta forma que no interesa que tan alejada sea la posición de un paquete nulo, con este método se identifica la posición que le corresponde dentro del TS o BTS.

- Se generó un nuevo archivo de extensión .ts de menor tamaño, en caso de tramas TS el tamaño se reduce a más de la mitad y con tramas BTS el tamaño se reduce a la mitad, esto se da debido a que los 16 bytes que se agregan en los BTS por lo general traen información útil.
- Se realizaron pruebas del funcionamiento del algoritmo de compresión y expansión con diferentes TS y BTS, verificando que al momento de expandir una trama que ha sido expuesta al algoritmo antes mencionado se obtiene una trama con los mismos valores y características que la trama original.
- El análisis se llevó a cabo mediante el software desarrollado y FFMPEG, obteniendo resultados iguales.

Recomendaciones

- Se recomienda que, al momento de hacer las conversiones de los valores enteros a un valor binario, antes de hacer la conversión los enteros no deben tener signo ya que las posiciones más grandes se verán afectadas y no se podrá lograr un posicionamiento correcto.
- Una observación importante al momento de ejecutar el algoritmo de compresión o el de expansión, es que no se debe detener la ejecución del programa ya que se vería afectada la trama comprimida o expandida resultante.
- Se recomienda en el proceso de expansión esperar a que se muestre en pantalla que se ha completado el proceso ya que en primer lugar se expanden las tablas y luego las demás cargas útiles de la trama.
- Durante el proceso de expansión se recomienda esperar a que en la pantalla de

comandos se muestre que el proceso se ha completado, porque primero se expanden las tablas y luego se expanden las otras cargas útiles de la trama.

Referencias

15603-1, A. N. (2007). *Televisión digital terrestre — Multiplexación y servicios de información (SI) Parte 1: SI del sistema de radiodifusión .* Asociación Brasileira de Normas Técnicas.

468, E. E. (2019). *Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems.* Niza: European Broadcasting Union 2019.

Alulema, D. (2012). La Televisión Digital Terrestre en el Ecuador es interactiva. *EÍDOS*, 12-19.

Cantos Sanchez, L. R., Tapuy Rendon, S. J., & Ramos Sanchez, B. (2014). *Simulación del estandar de televisión digital isdb-tb basado en un esquema de modulación-demodulación ofdm implementado en matlab-simulink.*

Enríquez Chicaiza, P. E., & Rivadeneira Obregón, E. D. (2018). *Análisis comparativo para el servicio de televisión digital terrestre en Quito.* Quito.

Gómez, J., Lapo, H., & Oñate, L. (2019). Análisis y comparación de ISDB-T. *INGENIUS*, 71-77.

Granja Toledo, N. R. (2011). *Análisis del transport stream para el estándar de televisión digital ISDB-Tb.* Quito.

Gutiérrez Tapia, A. G., & Cochancela Alvear, M. Á. (2013). *DISEÑO DE UN LABORATORIO DE TELEVISIÓN DIGITAL PARA LA TRANSMISIÓN DE SEÑALES CON MULTIPROGRAMACIÓN, CONTENIDOS INTERACTIVOS Y*

GUÍA ELECTRÓNICA DE PROGRAMACIÓN (EPG). Cuenca: Universidad de Cuenca.

Jarrín A., R., Morejon G., S., & Bernal, I. (2010). Diseño de una red de frecuencia única para un canal de televisión. *Revista Politécnica*, 11-25.

Ladebusch, U., & Liss, C. A. (2006). Terrestrial DVB (DVB-T): una tecnología de transmisión para uso portátil y móvil estacionario. (págs. 183-193). IEEE.

Mancheno Arcos, C. R. (2015). *Desarrollo de una plataforma de generación y análisis de flujos de transporte para BTS.* 30 - 35.

Moncayo Flores, T. J., & Pozo Caicedo, M. E. (2014). *Generación del flujo único de paquetes de transporte TS de acuerdo a la norma ISDB-TB y desarrollo de una aplicación para su análisis.* Quito: Escuela Politecnica Nacional.

Muñoz Vera, J. A. (2015). *Estudio de factibilidad para la implementación de un laboratorio de televisión digital terrestre (TDT) para el laboratorio de networking en la Facultad de Ingeniería Industrial de la Universidad de Guayaquil.* Guayaquil: Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Ingeniería en Teleinformática.

Pisciotta, N. O. (2010). Sistema ISDB-Tb. En N. O. Pisciotta, *Serie de Materiales de Investigación* (págs. 35-41). Universidad Blas Pascal.

Sandoval Ruiz, C. E., & Fedón, A. (2007). Codificador y decodificador digital Reed-Solomon programados para hardware reconfigurable. *Dialnet Ingeniería y universidad*, 17-32.

Sotelo, R. (2011). Sistema de transmisión ISDB-T. *Memoria Investigaciones en Ingeniería*, 67-77.

TDT. (1 de 12 de 2020). *Televisión Digital Terrestre ECUADOR*. Obtenido de https://tdtecuador.mintel.gob.ec/que-es-la-tdt/

Villamarín, D., Olmedo, G., Lara, R., & Illescas, M. A. (2012). Generación de Transport Stream con Audio, Video y Datos de Interactividad para el Sistema de Televisión Digital Terrestre ISDB-Tb. *MASKAY*, 49 - 55.