
# DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

## CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

## TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES

## TEMA: ENCRIPTACIÓN DE CONTENIDO DIGITAL DENTRO DEL FLUJO DE TRANSPORTE DE TELEVISIÓN DIGITAL TERRESTRE

**AUTORA: SILVA BRAVO, MARÍA EVELINA**

**DIRECTOR: OLMEDO CIFUENTES, GONZALO FERNANDO**

**SANGOLQUÍ**

**2019**

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

## CERTIFICACIÓN

Certifico que el trabajo de titulación, "**ENCRIPTACIÓN DE CONTENIDO DIGITAL DENTRO DEL FLUJO DE TRANSPORTE DE TELEVISIÓN DIGITAL TERRESTRE**" fue realizado por la Srta. MARÍA EVELINA SILVA BRAVO, ha sido revisado en su totalidad y analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 16 de septiembre de 2019

**Ing. Gonzalo Fernando Olmedo Olfuentes, PhD.**

**C.C.: 1711696342**

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

## AUTORÍA DE RESPONSABILIDAD

Yo, SILVA BRAVO MARÍA EVELINA, declaro que el contenido, ideas y criterios del trabajo de titulación **"ENCRIPTACIÓN DE CONTENIDO DIGITAL DENTRO DEL FLUJO DE TRANSPORTE DE TELEVISIÓN DIGITAL TERRESTRE"** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 16 de septiembre de 2019

Evelina Silva B

**María Evelina Silva Bravo**

**C.C.: 1104936180**

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

## AUTORIZACIÓN

Yo, SILVA BRAVO MARÍA EVELINA, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación "**ENCRIPTACIÓN DE CONTENIDO DIGITAL DENTRO DEL FLUJO DE TRANSPORTE DE TELEVISIÓN DIGITAL TERRESTRE**", cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 10 de septiembre de 2019

Evelina Silva B.

**María Evelina Silva Bravo**

**C.C.: 1104936180**

## DEDICATORIA

*Dedico este trabajo a mi familia: Héctor, Soledad, Sofía, María Luz y Pillín, quienes son mi mayor alegría, el centro de mi vida y hacen que todo sacrificio valga la pena.*

*María Evelina Silva Bravo*

## AGRADECIMIENTO

*Agradezco a mis padres, Héctor y Soledad por toda la confianza y el apoyo que me dieron durante mi carrera profesional, y a pesar de la distancia que nos separaba, nunca permitieron que me sienta sola.*

*Gracias papá por los largos viajes que hacías para estar unas pocas horas a mi lado, por tu paciencia e infinito amor y dedicación por tu familia.*

*Gracias a mamá por cada consejo y palabra de aliento y por recibirme con abrazos y mucho amor cuando volvía a casa.*

*Gracias a mi hermana Sofía, mi mejor amiga y mi más fiel compañera de vida. Porque nada sería igual sin ella, sin sus palabras, sin sus ocurrencias.*

*Agradezco a mis tutores, Dr. Gonzalo Olmedo e Ing. Freddy Acosta, profesionales que son un ejemplo para sus alumnos, y que siempre tuvieron la paciencia y dedicación para ayudarme a solventar mis dudas durante este proyecto.*

*Gracias también a todos los maestros que me formaron profesionalmente y gracias a la Universidad de las Fuerzas Armadas por darme la oportunidad de cumplir mi sueño de ser Ingeniera.*

*Y finalmente, agradezco a mis amigos, quienes se convirtieron en mi familia en esta ciudad lejos de casa. Juntos fuimos creciendo y aprendiendo a superar obstáculos, cosechando maravillosos recuerdos que nos durarán toda la vida.*

*María Evelina Silva Bravo*

# ÍNDICE DE CONTENIDO

## Contenido

## ÍNDICE DE TABLAS



## ÍNDICE DE FIGURAS

# RESUMEN

En este trabajo de titulación fueron desarrollaron dos algoritmos de encriptación y desencriptación de contenido digital en los espacios de relleno de las tablas PSI/SI del flujo de transporte TS. Los algoritmos se diseñaron con base al estándar MPEG-2 TS de tal forma que no se altera el audio y video original, ni se modifica la estructura de las tablas que forman la programación. El documento elaborado consta de la información teórica en el capítulo 2, seguido del diseño y la lógica de los algoritmos y las funciones diseñadas en el capítulo 3. En el capítulo 4, se muestran los resultados obtenidos para dos aplicaciones diferentes, una imagen digital y un stream de datos, mediante procesos de transmisión y recepción en tiempo real ejecutados en el laboratorio de Televisión Digital de la Universidad de las Fuerzas Armadas ESPE dentro de dos escenarios propuestos. En el capítulo final se presentan las conclusiones, recomendaciones y trabajos futuros.

**PALABRAS CLAVE:**

- **TDT**
- **TRANSPORT STREAM**
- **PAT**
- **PMT**
- **BYTES DE RELLENO**

## ABSTRACT

In this project two algorithm of encryption and decryption of digital content in the nules bytes of the transport Steam PSI/SI tables were developed. The algorithms were designed based on the MPEG-TS standard so the original audio and video are not altered, nor is the structure of the tables that forms the programming. The elaborated document consists of the corresponding theory in chapter 2, followed by the design and logic of the algorithms in chapter 3. In chapter 4, the results obtained for two different applications are shown, for an image and a data stream, through the transmission and reception processes in real time in a Digital Television Laboratory for two proposed scenarios. The final chapter presents conclusions, recommendations and future work of the project.

**KEYWORDS:**

- **TDT**
- **TRANSPORT STREAM**
- **PAT**
- **PMT**
- **NULL BYTES**

# CAPÍTULO 1

# INTRODUCCIÓN

## 1.1. Antecedentes

La Televisión Digital Terrestre (TDT), trajo consigo un sin número de nuevas posibilidades debido a las características de su tecnología, entre ellas, disponer de televisión móvil, el manejo de señales de emergencia, variedad en la calidad de imagen, y la transmisión de información multimedia de forma bidireccional, conocida como interactividad (Zuffo, 2008).

Al igual que varios países latinoamericanos, en Ecuador se adoptó el estándar de TDT japonés brasileño ISDB-T internacional, el cual, ha mostrado un mejor desempeño en áreas altamente pobladas y una señal fuerte y con bajo nivel de interferencia, en comparación a otros estándares como el europeo y el americano. Desde el año 2012 se inició el proceso de migración hacia la TDT que espera culminar en el 2021 con el apagón analógico. (MINTEL, 2018)

En ciudades como Quito, Guayaquil, Cuenca, entre otras, las señales de TDT ya se encuentran disponibles. Según el Plan Maestro de Transición a la TDT (MINTEL,

2018), existen 536 estaciones de televisión abierta analógicas operando a nivel nacional, de las cuales, 31 estaciones transmiten en formato digital a través de autorizaciones temporales, logrando una cobertura de 54% de la población. el 5,47% de las estaciones de televisión abierta en nuestro país se encuentran transmitiendo en formato digital.

La TDT encapsula datos dentro de un flujo de transporte (TS – de sus siglas en inglés *Transport Stream*) que se caracteriza por ser desarrollado en lenguaje de alto nivel, sin embargo, responde a estructuras binarias agrupadas en paquetes, con longitud fija de 188 bytes, de los cuales 4 son de cabecera y 184 de carga útil. Dentro de un TS, se pueden transportar datos de audio, video y datos digitales que pueden ser utilizados para contenido interactivo, guía de programación, entre otros. Adicionalmente, se transmiten tablas de especificación de programa PSI/SI. Dichas tablas son establecidas para guiar y simplificar los procesos de multiplexación, demultiplexación y presentación de programas en el decodificador. Entre ellas están la de asociación de programas (PAT), la de mapa de programa (PMT), la de acceso condicional (CAT), etc. Una de las formas más organizadas de transmisión de un contenido de datos es utilizando un carrusel de datos y objetos, en base a la especificación Digital Store Media Command and Control (DSM-CC), y transmitido de forma cíclica y ordenada.

## 1.2. Justificación e importancia del proyecto

La importancia de esta investigación radica en el aprovechamiento del ancho de banda de un TS. Los bytes de relleno representan un gran porcentaje dentro de las tablas PSI, y como su nombre lo indica, tienen la única función de completar la longitud fija de un paquete TS, por lo que en ellos no se transmite información útil ni de configuración.

El proyecto se orienta en ampliar las posibilidades del usuario interactivo de TDT, puesto que se enviará contenido adicional que puede ser aprovechado por ejemplo con la trancmisión de información a través de DSM-CC (del *inglés Digital Storage Media Command and Control*), sistemas de emergencia, envío de metadatos, encriptación de información, sistemas de envío de información segura, etc.

Existen varios trabajos orientados al análisis de TS, por ejemplo, el trabajo de (Benavides, 2015) donde se presenta un software que permite explorar el contenido de los paquetes de TS y que facilita la detección y corrección de errores o problemas en el flujo. También, el trabajo de (Yánez, 2015), que consiste en guías de trabajo para la manipulación y recomposición del flujo TS mediante los software MATLAB y FFMPEG. Respecto a DSM-CC, existe el trabajo de (Núñez, 2016) donde se diseñó un algoritmo capaz de recuperar el contenido multimedia de aplicaciones interactivas que contiene un TS, sin la necesidad de un decodificador especializado.

También, al hablar de seguridad de la información, el trabajo puede ser orientado a resguardar y proteger el contenido extra que se envía, manteniendo la confidencialidad, e integridad de los datos. Es así como la información estará al alcance solamente de los

usuarios que dispongan del algoritmo de recepción respectivo, permitiendo a la TDT convertirse en un sistema de transporte de información segura.

## 1.3. Alcance del proyecto

El trabajo de titulación tiene como objetivo la encriptación de contenido dentro del flujo de transporte de la TDT, es decir, además del audio, video y datos que se transmiten por una señal de televisión digital, se envía contenido adicional que puede ser de cualquier tipo, por lo que la aplicación del trabajo puede darse en varios campos, por ejemplo, para la seguridad de la información, puesto que garantiza tres aspectos principales: confidencialidad, disponibilidad e integridad.

Los algoritmos diseñados serán capaces de separar la carga nula de las tablas del flujo de transporte para convertirlos en paquetes de contenido digital sin alterar la información útil de las misma.

Se comprobará la validez de los algoritmos dentro de dos escenarios, cableado y RF, en un proceso de transmisión y recepción en tiempo real dentro del Laboratorio de la Universidad de las Fuerzas Armadas ESPE, y para dos tipos de contenido digital, los cuales se elegirán de acuerdo con los criterios planteados en el trabajo.

## 1.4. Objetivos

### 1.4.1. General

- Encriptar contenido digital dentro del flujo de transporte de Televisión Digital Terrestre.

### 1.4.2. Específicos

- Desarrollar un algoritmo que permita la encriptación de archivos digitales dentro de los bytes de relleno de las tablas PAT y PMT de un flujo de transporte TS.
- Proponer una ecuación para el cálculo de tamaño de contenido digital a ser encriptado dentro de los bytes nulos de las tablas PAT y PMT.
- Desarrollar un algoritmo de recepción de un TS, mediante el cual se pueda recuperar información existente en los bytes de relleno de las tablas PSI/SI.
- Verificar que el sistema propuesto de encriptación no afecta la transmisión del audio, video y datos de un TS.
- Implementar escenarios reales de laboratorio y realizar pruebas que permitan validar la funcionalidad del sistema.

## 1.5. Resumen de contenidos

El documento se encuentra organizado en los siguientes capítulos: el segundo contiene el marco teórico que hace referencia a la televisión digital, la estructura del flujo de transporte, tablas y equipos del Laboratorio de TDT de la Universidad de las Fuerzas

Armadas Espe. El tercer capítulo es la explicación sobre el diseño e implementación de los dos algoritmos, así como de las funciones utilizadas para cada parte. En el Capítulo 4 se presentan los resultados obtenidos para dos aplicaciones en el laboratorio de la Institución, para finalmente en el Capítulo 5 presentar las conclusiones, recomendaciones y trabajos futuros del trabajo de titulación.

# CAPÍTULO 2

# MARCO TEÓRICO

## 2. Televisión digital terrestre

Un sistema de televisión consiste en la transmisión de señales de audio y video a través de ondas electromagnéticas, y es el medio por el cuál se accede a un mundo de información y entretenimiento. Su desarrollo inició en el año 1920 con la televisión a blanco y negro, para después evolucionar en 1967 a la era de la televisión analógica a color, que permitía a los usuarios una mejor experiencia visual. Sin embargo, la demanda de mejores servicios como audio y video de alta calidad llevaron a la invención de la TDT (Song, Yang, & Jun, 2015).

La diferencia entre esta nueva tecnología y las anteriores, es el aprovechamiento de los recursos del espectro radioeléctrico, que a su vez permite aumentar la calidad del contenido transmitido en función de sonido e imagen, así como la emisión de un mayor número de canales de programación a través de un solo canal de transmisión mediante un proceso conocido como multiplexación. Para el funcionamiento de la TDT se sigue un proceso de cuatro fases: (1) los radiodifusores transmiten información en formato digital,

(2) las estaciones televisivas retransmiten las señales, (3) las antenas convencionales de TV captan la información y (4) el decodificador transforma las señales digitales en analógicas para que puedan ser procesadas por el televisor. (Organización de Telecomunicaciones de Iberoamérica, 2017)

Según (MINTEL, s.f.), los servicios de TDT se encuentran en gran parte de Europa como España, Italia, Portugal, entre otros; en países asiáticos como Japón, Corea y Taiwán; y respecto a América del Norte en México, EEUU y Canadá. Los países latinoamericanos a excepción de Colombia y Las Guyanas, han adoptado ya un estándar de TDT, pero aun no se ha producido el cese de transmisiones analógicas para dar paso a la tecnología digital.

## 2.1. Estándares de televisión digital terrestre

En diferentes partes del mundo se crearon estándares de TDT para definir la transmisión e interpretación de señales de televisión. Por ejemplo, el estándar ATSC (*Advanced Television Systems Comittee*) desarrollado en EEUU, el DVB-T (*Digital Video Broadcasting-Terrestrial*) por la Organización de Radiodifusión Digital de la Unión Europea, ISDB-T (*Integrated Service Digital Broadcasting-Terrestrial*) por Japón, y DTMB (*Digital Terrestrial Television Multimedia Broadcasting*) desarrollado en China. Todos estos fueron aprobados por la Organización Internacional de Telecomunicaciones y comercializados en varios países.

En Ecuador, el gobierno ecuatoriano adoptó el estándar ISDB-Tb japonés con variación brasileña en abril de 2009. Para el año 2010, en Quito se inció las pruebas con

el canal del estado Ecuador TV y en el año 2013 en la ciudad de Guayaquil, se inauguró el sistema de TDT con la señal al aire del canal TC Televisión (Conocimiento, 2013).

## 2.2. Estándar ISDB-tb

Fue desarrollado en 1998 por la Asociación de Radio Industrias y negocios (ARIB), y actualmente proporciona e integra servicios como audio, video y multimedia, los cuales incluyen TV de alta definición (HDTV), TV multi programa de definición estándar (SDTV), y TV de baja definición para recepción portátil y datos o también llamada *1Seg*. En Japón se usa desde el año 2003 con los servicios de HDTV y SDTV, y en 2006 el gobierno de Brasil adopta este estándar con ciertos cambios como la utilización de MPEG-4 para la compresión de audio y video, así como la posibilidad de interactividad a través de un middleware llamado Ginga. De este desarrollo nace SBTVD conocido como ISDB-Tb o ISDB-T Internacional, el cual fue adoptado en la mayoría de los países sudamericanos.

### 2.2.1. Características técnicas

Entre las características de ISDB-T se encuentran:

- Su técnica de transmisión es OFDM (*Orthogonal Frequency Division Multiplexing*), por lo que no existe interferencia entre símbolos y una mejora de la relación señal a ruido. Las técnicas de codificación son QPSK, 16QAM, 64QAM, DQPSK, que permiten resolver problemas de propagación. Trabaja con tres modos, que se entienden como la separación entre rangos para frecuencia de portadoras:

- Modo 1: 3.968 ≈ 4 kHz
- Modo 2: 1.984 ≈ 2 kHz
- Modo 3: 0.992 ≈ 1 kHz

- Utiliza GINGA como Middleware o capa de software intermedio, que permite el desarrollo de contenido interactivo. Además, existe un canal de retorno destinada a aplicaciones que requieran uso de internet.
- Utiliza códecs del estándar MPEG-4 para video H.264 y audio MPEG-4 AAC, y el formato MPEG-2 TS (ISO/IEC 1318-1), el cual encapsula un flujo de datos multiplexados que transportan varias programaciones y servicios.

- Respecto a su ancho de banda ocupa 13 segmentos más uno de guarda, en un total de 6MHz, es decir, cada uno de 428,57 kHz, y son organizados de tal forma que los segmentos pares estén a la derecha del segmento central y los impares a su lado izquierdo, ***Figura 1***.


***Figura 1*.** Organización de segmentos.

- La transmisión es de tipo jerárquica, priorizando la modulación y el FEC (*Forward Error Correction*) en el servicio. Define el máximo de tres capas conocidas como A, B y C para la transmisión de diferentes servicios como Tv Móvil (*"one seg"*), HDTV y SDTV, en el mismo canal, es decir de forma simultánea.

- Tiene transmisión en banda angosta en el cual usa un solo segmento denominado “*one seg*” y que se ubica en la parte central de la banda. Se usa para la recepción en dispositivos móviles y portátiles.

### 2.2.2. TDT en ecuador

Según (MINTEL, s.f.), en el país existe 577 estaciones de televisión y un total de 30 de ellas cuentan con una concesión temporal para transmitir señales digitales de TDT bajo el estándar ISDB-T, en ciudades como Quito, Guayaquil, Cuenca, Santo Domingo, Manta, Latacunga y Ambato. De acuerdo al cronograma presentado en el Plan Maestro de TDT del Ministerio de Telecomunicaciones (Metro Ecuador, 2018), el apagón analógico, es decir, el cese de emisiones analógicas de televisión iniciará en el Ecuador desde el año 2020 y continuará progresivamente hasta el 2023.

## 2.3. Sistema de compresión MPEG-2

El estándar MPEG-2 (*Movie Picture Experts Group*) define el formato de los componentes de un programa multimedia, como audio, video, datos de contro y/o datos de usuario; y la forma en que éstos se condensan en un solo flujo de bits transmitidos de forma síncrona a través de un proceso denominado multiplexación (Fairhust, 2001). Lo componentes del flujo son:

- **Elementary Stream (ES):** Es el componente más básico en un flujo de datos en MPEG, y contiene una señal comprimida de audio video o datos, independiente para cada uno.

- **Packetized Elementary Stream:** Cada ES ingresa a un procesador MPEG-2, por ejemplo, un compresor de video o audio que acumula los datos en un flujo de paquetes de PES (*Packetized Elementary Stream*). Estos paquetes pueden ser de tamaño fijo o variable, y su longitud llega hasta 65536 bytes donde se incluye una cabecera y el contenido codificado. (Fairhust, 2001)

### 2.3.1. Multiplexación MPEG-2

Cada paquete PES conforma otros paquetes de transporte de tamaño fijo denominado flujo de transporte (TS -*Transport Stream*). Con este método se asegura la transmisión en ambientes de ruido que implican pérdida o corrupción de paquetes, así como también hace posible la transmisión de más de un programa a la vez (Fairhust, 2001). El TS se forma cuando al multiplexor ingresan los PES como se ilustra en la Figura 2, donde son añadidas tablas PSI (*Program Specific Information*), las cuales, como su nombre lo indica, añaden información específica de cada programa, y facilitan la tarea de organizar y presentar los programas en el decodificador. Las principales tablas PSI son las siguientes:

a. PAT (*Program Association Table*): tabla de asociación de programa.
b. PMT (*Program Map Table*): tabla de mapa de programa.
c. CAT (*Conditional Access Table*): tabla de acceso condicional.

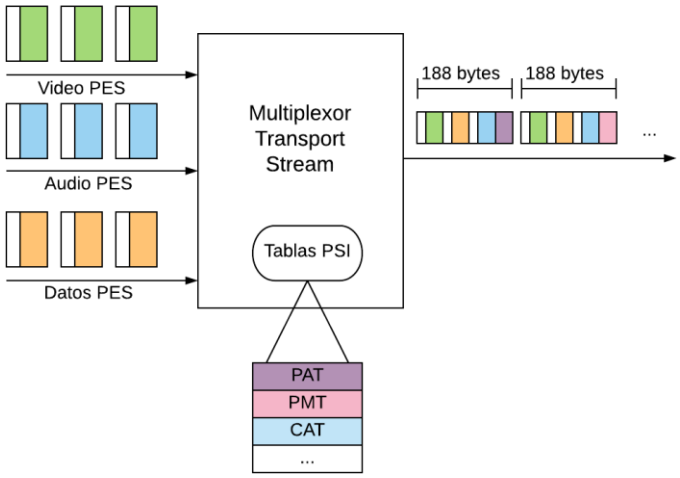


***Figura 2.*** Multiplexor en MPEG-2

### 2.3.2. Paquetes TS

Cada paquete que se forma a la salida del multiplexor tiene un tamaño fijo de 188 bytes, de los cuales 4 bytes corresponden a la cabecera y el resto se considera como la carga útil que se compone de los paquetes PES, un código CRC-32, o la información de cada servicio, Figura 3.

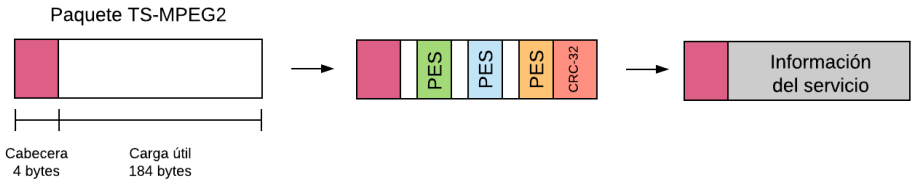


***Figura 3.*** Estructura de un paquete TS-MPEG2

#### 2.3.2.1. Cabecera de un paquete

La cabecera que se encuentra al inicio de cada paquete sirve para: (1) indicar el inicio de un nuevo paquete y (2) identificar el tipo de contenido presente en la carga útil. De sus cuatro bytes, el primero siempre corresponde al valor de 47H, mientras que el segundo y tercer byte almacenan el PID que consiste en un identificador de contenido que varía dependiendo de cada tabla. La estructura en bits de la cabecera se presenta en la Figura 4.

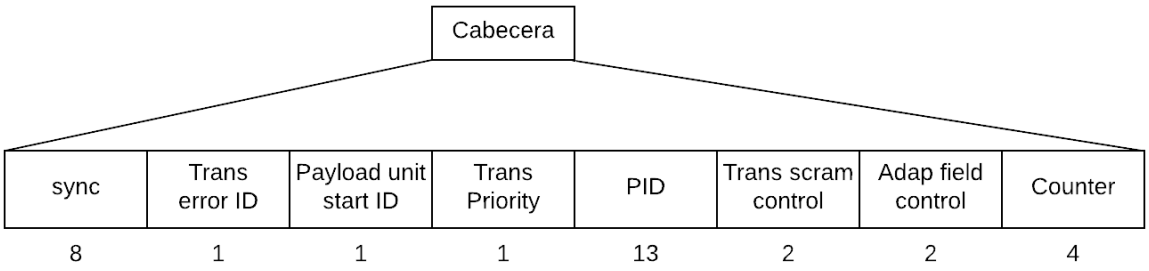


Figura 4 Cabecera de un paquete TS MPEG2

- *sync:* Campo de 8 bits de valor (47H) utilizado para indicar el inicio de un paquete TS y para la sincronización.
- *Trans error ID:* Campo de 1 bit que cuando su valor es 1 indica que existe error en el paquete y 0 cuando no hay error.
- *Payload unit start ID:* Campo de 1 bit que se activa cuando el paquete trasporta tablas PES.
- *PID:* Campo de 13 bits contenido entre el segundo y tercer byte. De acuerdo a su valor indica el tipo de contenido que transporta el paquete TS. Debido al número de bits que lo conforman, puede identificar hasta 8192 valores. Cuando su valor

es 1FFFH el paquete es nulo, mientras que, algunos valores están reservados específicamente para tablas de acuerdo a la norma, y otros valores se pueden asignar mediante broadcast.

- *Trans scram control:* Campo de 2 bits que indica la codificación del contenido de paquete.
- *Adapt field control:* Campo de dos bits que muestra si el TS tiene campo de adaptación y carga útil.
- *Counter:* Campo de 4 bits cuyo valor varía entre 0000 hasta 1111 y se traduce como un contador progresivo que identifica un paquete con el mismo PID. Si llega a su máximo valor se reinicia.

#### 2.3.2.2. Carga útil

Los 184 bytes de cada paquete corresponden a la carga útil, de los cuales: (1) el primer byte sirve como campo de adaptación y permite al decodificador reconocer el inicio de los datos útiles y (2) los bytes restantes donde están los paquetes PES (audio, video o datos comprimidos), o las tablas PSI/SI donde se añade información específica de cada programa, como se representa en la Figura 5. Para mantener la longitud fija de cada paquete, en caso de no utilizar todos los bytes disponibles se rellenan con bytes de valor 0XFF, los cuales son claves para el desarrollo del trabajo.

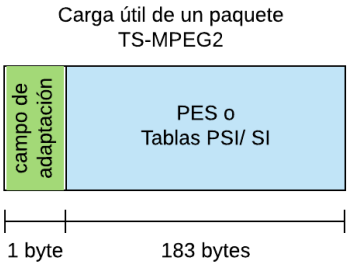


***Figura 5.*** Carga útil de un paquete TS -MPEG2

### 2.3.3. Tablas PSI/SI

Las tablas PSI transportan información específica de cada programa, para que el receptor pueda identificar fácilmente las propiedades del flujo y realizar la demultiplexación y decodificación de la señal. Se envía periódicamente y están protegidas por un código CRC que permite verificar la integridad de cada tabla en el receptor.

La estructura básica de las tablas PSI/SI se presenta en la Figura 6. Las principales secciones se describen a continuación (Yánez, 2015):

- *Table ID:* Campo de 1 byte para identificar el tipo de tabla: PAT, PMT, CAT, por lo que su valor es único para cada una. En la tabla Tabla 1, se presentan los valores correspondientes.
- *Sección de Longitud:* Campo de 12 bits representado en un solo byte, y su valor indica la longitud que va desde el byte continuo hasta el último del código CRC-32. En el trabajo permitirá identificar el inicio de los bytes de relleno.

- *CRC-32:* Campo de 4 bytes conocido como código de redundancia cíclica para la corrección de errores.

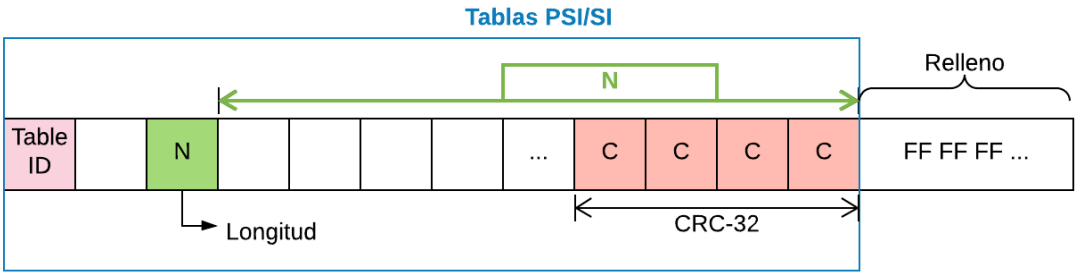


***Figura 6.*** Estructura de una tabla PSI/SI

Algunas de las tablas están asociadas con PID predefinidos, los cuales se presentan en la **Tabla 1.** En la siguiente sección se detallará la estructura de las tablas PAT y PMT que son usadas en este trabajo para transportar el contenido digital encriptado.

**Tabla 1**
*Valores de las tablas PSI*

| **Nombre** | **Significado** | **PID** | **Table ID** | **Table ID** |
|---|---|---|---|---|
| PAT | *Program Association Table* | 0000H | 00H | 00H |
| CAT | *Conditional Access Table* | 0001H | 01H | 01H |
| PMT | *Program Map Table* | Asignado por PAT | 02H | 02H |

### 2.3.3.1. Tabla de asociación de programas: PAT

La tabla PAT se caracteriza porque el valor tanto de su PID como el de table ID es 00H. Esta tabla se encarga de asignar un valor único de PID para cada programa encontrado en el flujo de transporte, es decir, define el PID de las tablas PMT, el cual por lo general se encuentra antes del código CRC-32 como detalla la Figura 7.

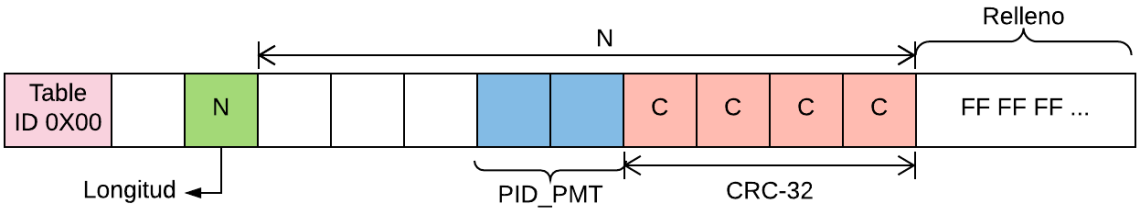


***Figura 7.*** Estructura de la tabla PAT – parte 1

La estructura en bits de la tabla PAT, Figura 8, y el significado de cada elemento que la conforman se detallan de acuerdo a la Norma Brasileña (ABNT, 2007):

- *Table ID:* Campo de 8 bits cuyo valor por defecto es 0x00
- *Section Syntax ID:* Campo de 1 bit que cuando es 1 indica que el paquete no tiene errores y cuando es 0 sí.
- *Reserv:* Campo de 3 bits que se guardan para otros usos.
- *Section Length:* Campo de 12 bits que indica el rango útil de la tabla expresado en bytes.
- *TS ID:* Campo de 16 bits configurable por el usuario para identificar un TS de cualquier multiplexor presente en la red de transporte.

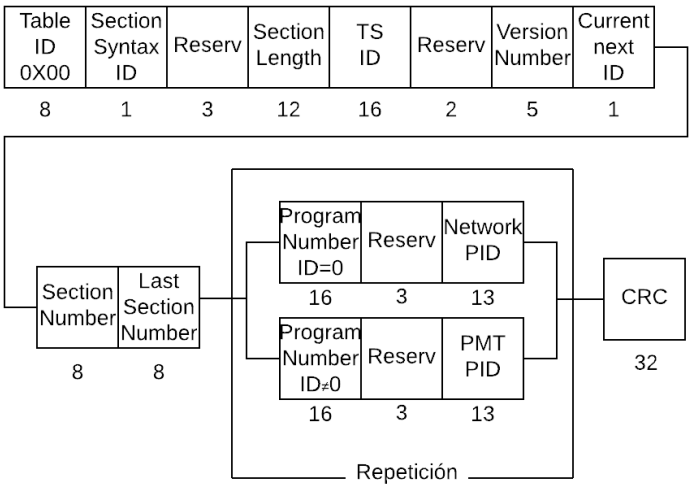


***Figura 8***. Estructura de la tabla PAT parte 2, adaptado de (ABNT, 2007)

- *Version Number:* Campo de 5 bits que indica el número de versión e incrementa su valor en 1 si encuentra una alteración en la tabla.
- *Current Next ID:* Campo de 1 bit que indica la validez y aplicabilidad de la tabla cuando su valor es 1, y cuando es 0 indica que la PAT no es válida.
- *Section Number:* Campo de 8 bits que indica el cambio de la sección. Su valor inicial es 00H y aumenta cuando se detecta una nueva sección.
- *Last Section Number:* Campo de 8 bits que indica el valor previo de *Section Number* conforme éste va en aumento.
- *Program Number ID:* Campo de 16 bits que cuando es 0000 el siguiente valor corresponde al valor al PID de la red.
- *Network PID:* Campo de 13 bits que indica el PID de la red.
- *PMT PID:* Campo de 13 bits que asigna un valor único de PID a las PMT que se encuentran en flujo de transporte.

- *CRC:* Campo de 32 bits que generan una salida igual a cero en los registros del codificador.

### 2.3.3.2. Tabla de mapeo de programas: PMT

Las funciones de la PMT consisten en (1) identificar el número de programas dentro del flujo de transporte y (2) la localización de PCR (*Program Clock Reference*), que permite la sincronización del reloj de 27 MHz para que cada programa pueda codificarse a velocidad diferente. Además, cada programa tiene asociada una tabla PMT, por lo que el valor de PID para cada una es independiente, y es asignado por la tabla PAT. Para el sistema ISDB-T el PID puede tomar valores de 0x030 a 0x1FE (Yánez, 2015).

La estructura de la tabla PMT se muestra en la Figura 9,donde cada término corresponde a:

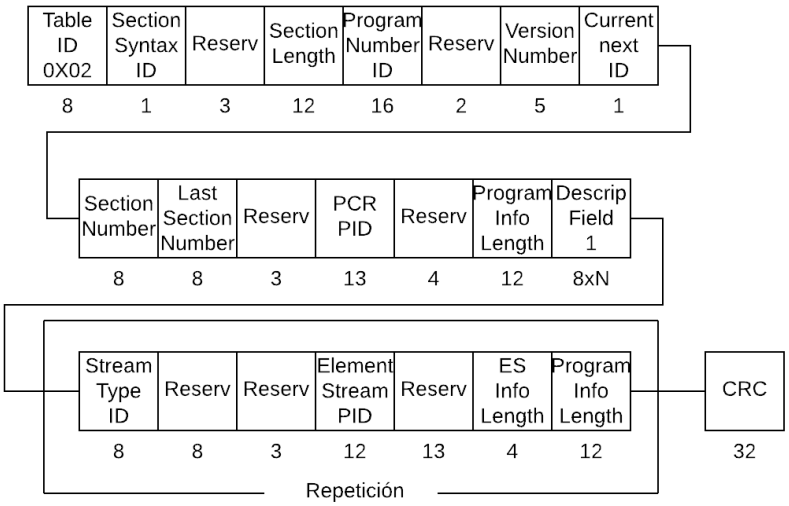


***Figura 9.*** Estructura tabla PMT, adaptado de (ABNT, 2007)

- *Table ID:* Campo de 8 bits de valor 02H.
- *Section Syntax ID:* Campo de 1 bit que al ser 0 indica error en el paquete.
- *Reserv:* Campo de 3 bitss que son reservados para usar posteriormente.
- *Section Lenght:* Campo de 12 bits que indica hasta donde existe carga útil y donde empiezan los bytes de relleno.
- *Program Number ID:* Campo de 16 bits que corresponde al número de programa asociado a esa PMT.
- *Version Number:* Campo de 5 bits que indica la versión de la PMT e incrementa su valor si existe un cambio en la tabla.
- *Current next ID:* Campo de 1 bit que asegura la validez de la tabla cuando toma el valor de 1.
- *Section Number:* Campo de 8 bits que aumenta de acuerdo al número de secciones en la PMT.
- *Last Section Number:* Campo de 8 bits que indica el valor anterior del campo *Section Number,*
- *PCR PID:* Campo de 13 bits que indica el PID del PCR.
- *Stream Type:* Campo de 8 bits que indica el tipo de stream que el programa asociado a la PMT está utlizando.
- *Element Stream PID:* Campo de 13 bits que indica el PID de cada *Element Stream* para su identificación en el paquete.

- *ES Info Length:* Campo de 12 bits donde está el tamaño del descriptor de cada stream.
- *CRC:* Campo de 32 bits que genera una salida igual a cero en el decodificador.

## 2.4. Laboratorio de televisión digital

El laboratorio de la Universidad de las Fuerzas Armadas ESPE, Figura 10. Equipos del Laboratorio de TDT de la Universidad de las Fuerzas Armadas ESPE ,cuenta con los equipos detallados en esta sección que se utilizaron para realizar la transmisión y ejecución en tiempo real de un archivo TS, y comprobar los resultados del trabajo de investigación.

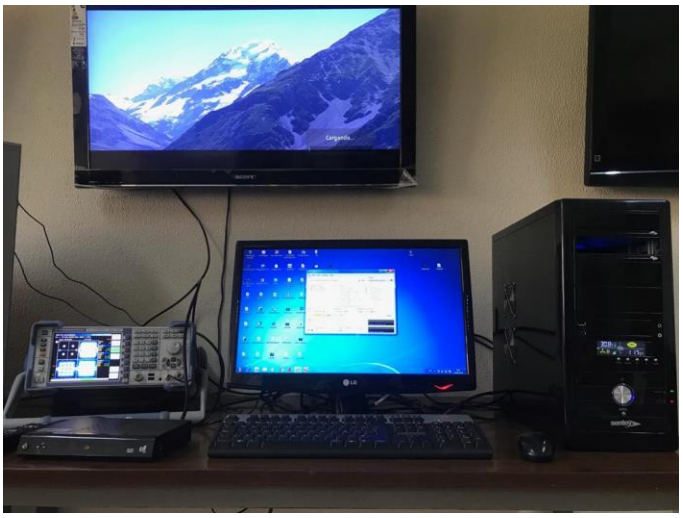

***Figura 10.*** Equipos del Laboratorio de TDT de la Universidad de las Fuerzas Armadas ESPE

### 2.4.1. Transmisión

#### 2.4.1.1. Tarjeta DTA-115

La transmisión del TS se realizó a través de una tarjeta moduladora DTA-115, Figura 11 , la cual está instalada en la computadora del laboratorio. Entre sus características están su convertidor de frecuencia VHF/UHF y el soporte a los estándares de modulación QAM, OFDM y VSB. Se usa como un modulador de prueba multiusos de propósito general para desarrollar y calificar equipos DTV, o para experimentar con nuevos esquemas de modulación de RF. Cuenta con una salida RF y un puerto DVB-ASI bidireccional. (Dektec, s.f.).

***Figura 11.*** Tarjeta moduladora DTA-115. Fuente: Dektec.com

#### 2.4.1.2. Computadora y Stream Xpress TS Player

La tarjeta mencionada en la sección anterior debe estar instalada en una computadora que cumpla con un mínimo Core 2@2GHz para ISDB-T, 512Mb en RAM y un puerto PCI. Además, se necesita el software de Windows de reproducción, en este

caso, Stream Xpress TS Player de Dektec. A través del programa se genera el flujo de transporte y se logra reproducir en tiempo real transmisiones compatibles con el estándar MPEG-2. En la Figura 12 se muestra la interfaz gráfica y en la pestaña *Adapter* se puede constatar que está reconocida la tarjeta DTA-115.

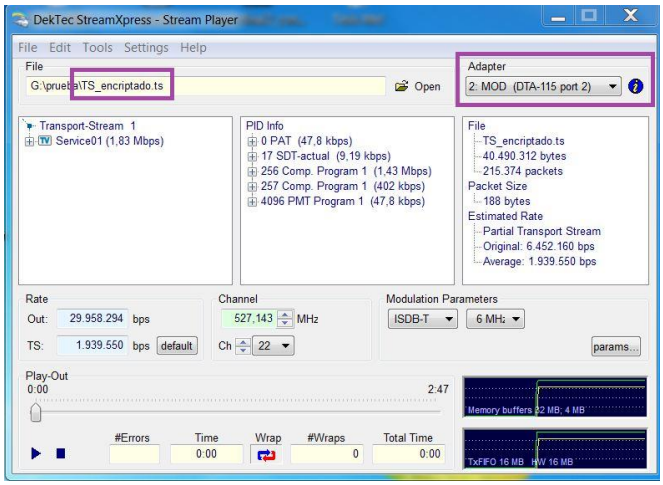


***Figura 12***. Interfaz gráfica del software Stream Xpress y reconocimiento de tarjeta DTA-115

En el software debe cargarse el archivo TS que se va a transmitir,y se configuran los parámetros de modulación y el canal, Figura 13,. Se da click en *Play* y la señal empieza a ser transmitida.

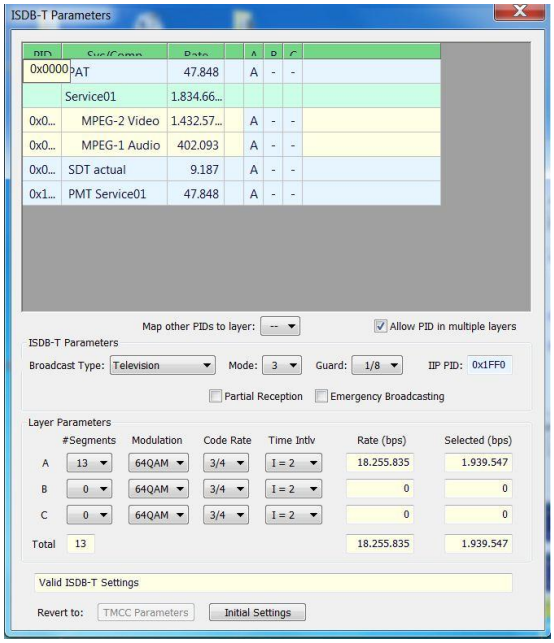


***Figura 13*.** Menú de configuración de parámetros en el software Stream Xpress Player

### 2.4.2. Recepción

Para la recepción se utilizó el Analizador de TV y Satélite Ranger Neo 2 PROMAX, Figura 14. Este dispositivo cuenta con las funciones de medidor de TV y analizador de espectro de banda terrestre y satélite. Además, permite gestionar los datos que se generan en cada instalación que facilita el control de la información generada o guardarla dentro de un USB para su posterior análisis. (Promax, 2019)


***Figura 14.*** Analizador Ranger Neo 2 (Promax, 2019)

Otra de las funciones es la grabación de Transport Stream, que permite grabar en tiempo real el TS que se recibe. El equipo debe configurarse de la siguiente forma:

1. Encender el analizador.
2. Conectar la señal de entrada al equipo.
3. Acceder al menú de Ajustes y seleccionar en la opción “Fuente de la Señal” el tipo de señal.
4. Acceder al menú de Ajustes y seleccionar la entrada del transport stream: Demoduladores RF/IPTV o Entrada ASI.
5. Pulsar la tecla Utilidades y seleccionar “Grabación TS”.
6. Aparece la pantalla para grabar/reproducir el TS, Figura 15. En el menú Avanzado seleccionar el destino de grabación, en este caso pendrive USB.
7. Pulsar la tecla de grabación RECORD para iniciar el proceso.
8. Al terminar, pulsar la tecla STOP.


***Figura 15.*** Grabación de TS con Ranger NEO 2

#### 2.4.2.1. Entrada de señal RF

En el equipo de recepción es posible configurar una señal RF como fuente de entrada, puesto que cuenta con un demodulador interno. Además, utiliza un sistema de autoidentificación de forma automática que identifica el tipo de señal y sus parámetros característicos. Dentro de las opciones disponibles para este tipo de señal se puede elegir entre sintonizar por canal o por frecuencia, la modificación de la frecuencia central, del nivel de referencia o del span (Promax, 2019).

#### 2.4.2.2. Entrada TS-ASI

La opción de entrada TS-ASI permite monitorizar y analizar los TS que provengan de receptores de señal satélite, reproductores de TS, multiplexadores, etc. El equipo detecta automáticamente si el TS se compone de 188 o 204 bytes (Promax, 2019).

# CAPÍTULO 3

# DISEÑO E IMPLEMENTACIÓN

## 3.1. Herramientas de desarrollo

El software usado para desarrollar el trabajo es Matlab (acrónimo de MATrix LABoratory), considerado como una herramienta que permite realizar cálculos científicos y tecnológicos complejos a partir de una representación de valores basada en matrices. (Mathworks, s.f.)

A diferencia de otros lenguajes de propósito general como Basic, C o C++, este software posibilita la programación de algoritmos al incluir una amplia gama de funciones y librerías de forma rápida mediante el uso del lenguaje específico que proporciona. Se encuentra disponible para Windows, Apple, UNIX, así como para versiones móviles como IOS Android, por lo que es considerado como una multiplataforma de propósito general y en particular en las relaciones con los estudios relativos a la ingeniería. (Reinoso, Jiménez, Payá, Aparicio, & Peidró, 2018)

El presente trabajo tiene el objetivo de encriptar contenido digital dentro del flujo de transporte (TS) de Televisión Digital. Se divide en dos partes: (1) el algoritmo de transmisión, donde se identifican las tablas PAT y PMT presentes en el flujo, a través de funciones diseñadas para identificar su PID y para reemplazar sus bytes nulos por paquetes de contenido digital, y (2) el algoritmo de recepción que consiste en una reconstrucción del contenido digital que se encuentra dentro de las tablas.

Los diagramas de bloques de cada parte se presentan en la Figura 16. En el algoritmo para transmisión se necesita el archivo TS y el contenido digital, y su salida corresponde al TS modificado, mientras que en el receptor se analiza el archivo modificado y su salida es el contenido digital que fue encriptado.

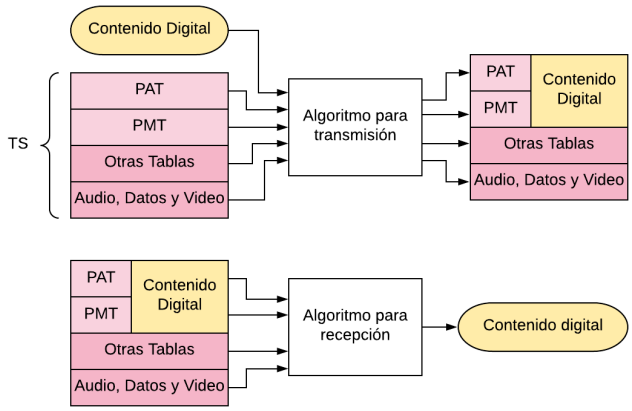


***Figura 16.*** Diagrama de bloques de los algoritmos

## 3.2. Estructura del algoritmo de transmisión

El algoritmo inicia al abrir el archivo (*TS.ts*) en modo lectura con la finalidad de acceder a sus características, tales como nombre, ubicación, fecha de modificación, y tamaño en bytes, el cual, al ser divido para el tamaño de cada paquete, permite conocer el número total de paquetes presentes el TS. En las siguientes secciones se explican las funciones usadas dentro del algoritmo principal que son TS_PID, PID_PMT y CABECERA, que permiten crear un proceso iterativo donde se analiza cada paquete del TS para seleccionar las tablas PAT y PMT y realizar el proceso principal de encriptación de contenido.

### 3.2.1. Función TS_PID

La primera función diseñada es *TS_PID,* la cual determina el PID de un paquete y sigue la lógica de la Figura 17, donde se recibe la variable *trama,* que almacena una cadena de 188 bytes, es decir, lo correspondiente a un paquete que tiene por defecto formato hexadecimal.

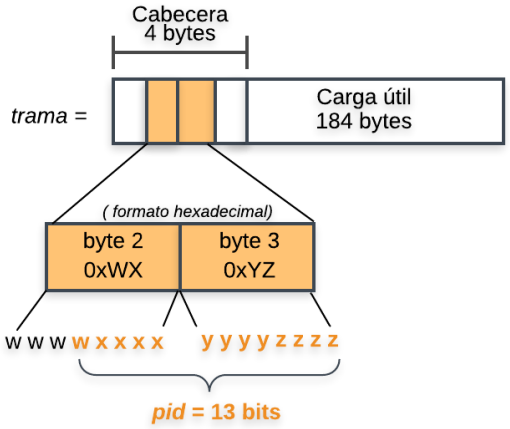


***Figura 17.*** Identificación de PID de un paquete

El segundo y tercer byte de esta cadena que pertenecen a su cabecera, se ubican en un nuevo vector denominado *pid*. El siguiente paso consiste en transformar *pid* a su forma binaria, es decir, en un vector de 16 bits, de los cuales, los trece que van desde la posición cuatro a la dieciséis, corresponden al PID del paquete, y finalmente, se convierte a decimal, Figura *18.* Diagrama de flujo de la función TS_PIDFigura 18. A través de esta función se crea el proceso iterativo principal del algoritmo, donde se identifica el PID para discriminar las tablas PAT y PMT.

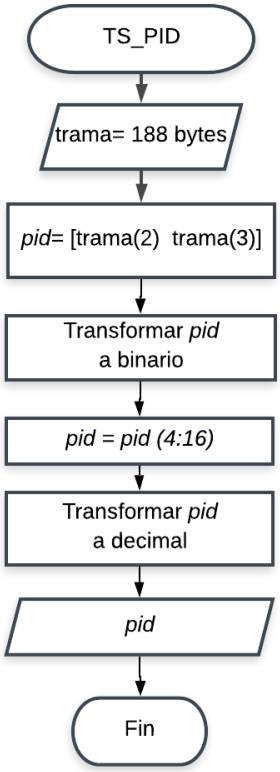


***Figura 18*.** Diagrama de flujo de la función TS_PID

### 3.2.2. Función PID_PMT

El PID varía de acuerdo a la naturaleza de la tabla. El de la PAT, por ejemplo, es un valor fijo de 0x00, y dentro de esta tabla se asigna aleatoriamente un valor único para el identificador de la PMT, por lo que, la siguiente función diseñada consiste en acceder a una tabla PAT para conocer el valor asignado de PID a la PMT, tomando como referencia que se encuentra en la carga útil, y que corresponde a los dos últimos bytes antes del código CRC-32, como se muestra en la Figura 7 del Capítulo II.

En la Figura 19 se ilustra el diagrama de la función, donde se analiza una tabla PAT, es decir, una tabla con PID igual a 0x00. La longitud de la tabla (*N*), indica hasta qué byte existe carga útil, y con esta referencia en un vector *pidpmt* se almacenan los bytes de las posiciones *N-5* y *N-6* que contienen el valor de PID de la PMT. Al igual que el identificador que se encuentra en la cabecera de cualquier tabla, el valor corresponde solamente a 13 bits y se debe transformar a su correspondiente número decimal.

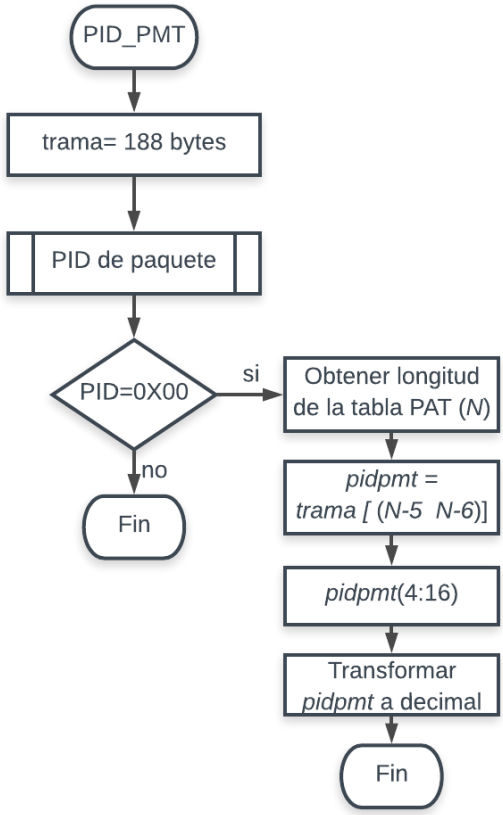


***Figura 19.*** Diagrama de flujo para identificación de PID de PMT

### 3.2.3. Características del archivo digital

Como se mencionó en el Capítulo 2 sección 2.5.2, las tablas no usan toda la longitud disponible, es decir que, de los 188 bytes algunos se ocupan como carga útil, mientras que los restantes se denominan bytes de relleno o bytes nulos, cuyo valor es de 0XFF. Estos bytes se reemplazarán por el contenido de un archivo digital (*digital.**), el cual puede ser de cualquier naturaleza, ya sea imagen, audio, documento, etc.

El archivo debe tener cierto tamaño, delimitado principalmente por la cantidad de bytes de relleno existentes entre las tablas PAT y PMT. Además, se recomienda que, para asegurar la recepción, se envíe como mínimo tres veces dentro del TS. Esto se debe a que al sintonizar un canal de TDT no siempre se ejecuta desde el primer instante, lo que resultaría en paquetes perdidos, sin embargo, si el contenido digital se encuentra más de una vez en el flujo, los paquetes que no fueron capturados se encontrarán repetidos en los segundos posteriores, y de esta forma se asegura su reconstrucción en el receptor.

Con estos datos, se plantea la Ecuación No.1 para el cálculo del tamaño de archivo digital, en función del tiempo de duración del TS y de los bytes de relleno disponibles entre las dos tablas:

$$n = \frac{x\ t_{TS}}{3\ t_t} \tag{1}$$

donde:

- $n$ es el tamaño del archivo a encriptar expresado en bytes.
- $t_{TS}$ es el tiempo de duración del TS.
- $x$ es el número de bytes de relleno disponibles sumado entre las tablas PAT y PMT.
- $t_t$ , correspondiente a $100ms$, definido en la norma como el tiempo en que se transmite las tablas PAT y PMT consecutivas,

Dentro del algoritmo principal se abre *digital* en modo lectura, de forma que se pueda acceder a la cadena de bytes que lo conforman y almacenarla en dos vectores denominados *vectordig* y *vorginal*. En cada iteración del algoritmo, del *vectordig* se irán recortando paquetes como muestra la Figura 20, para posteriormente encriptarlos en las tablas PAT y PMT, mientras que *voriginal* mantendrá el contenido original.

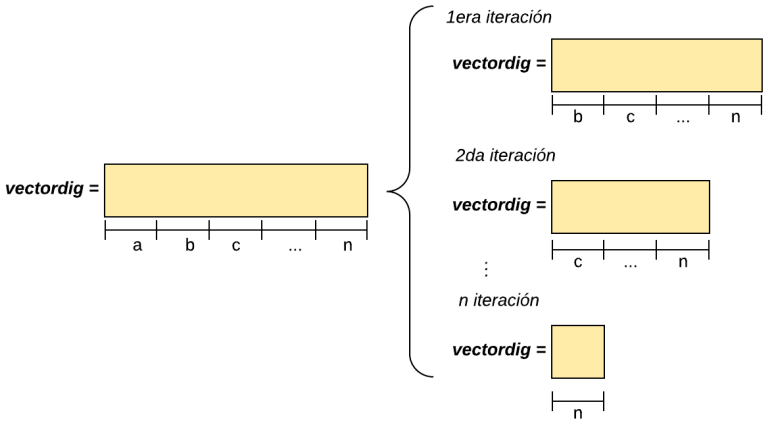


***Figura 20*.** Proceso de recorte para *vectordig*

Además, también se debe crear el nuevo TS denominado *TS_encriptado.ts* y abrirlo en modo escritura. Este archivo almacena las tablas PAT y PMT modificadas, además de los paquetes de audio, video, datos y demás componentes del archivo TS original.

Cuando los tres archivos están disponibles (*TS.ts, digital.** y *TSecnriptado.ts*) el algoritmo se encarga de analizar cada *trama* del flujo original. Si el PID de *trama* corresponde al de PAT o PMT, se procede a almacenar las siguientes variables, Figura 21:

1. La cantidad de espacio de relleno menos cinco bytes (*bnul*).
2. Un contador que aumenta cada que se registra una tabla PAT o PMT (*cont*).
3. La diferencia (*dif*) entre la longitud del vector *vectordig* y *bnul.*

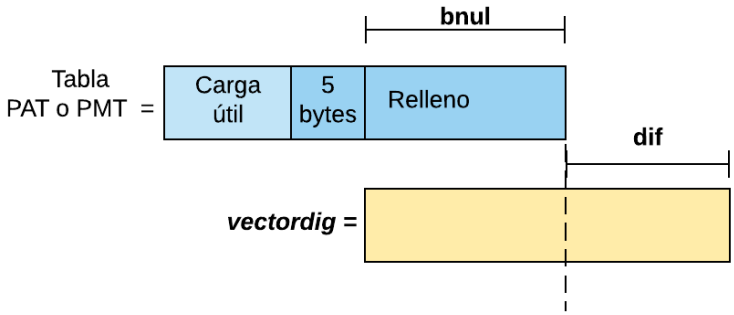


***Figura 21.*** Variables almacenadas de cada tabla PAT o PMT

Una vez creadas estas tres variables se procede a usar la función *cabecera* que se explica en la siguiente sección.

### 3.2.4. Función *CABECERA*

A cada paquete, además del contenido digital, se adiciona una cabecera de cinco bytes, Figura 22, de los cuales:

- 3 bytes corresponden a la numeración del paquete.
- 2 bytes, donde 14 bits se usan para indicar la longitud de paquete, y 2 bits se usan como un indicador del inicio, transmisión o finalización del proceso de encriptación, y se configura como indica la Tabla 2.

**Tabla 2**
*Valores del indicador*

| Valor | Indicador |
|---|---|
| **10** | Inicio |
| **00** | Transmisión |
| **01** | Fin |

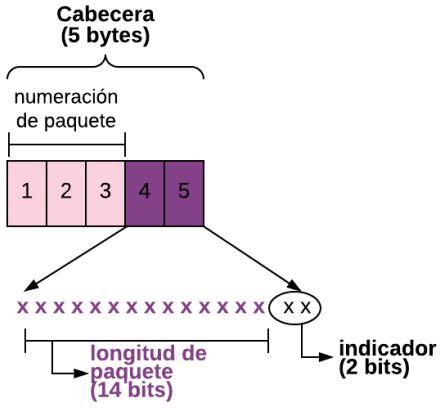


***Figura 22*.** Estructura de la cabecera de paquete de contenido digital

La cabecera permite que el receptor ejecute el proceso de desencriptación hasta que todos los paquetes hayan sido recuperados, es decir, hasta que el contenido digital haya sido reconstruido. La función, Figura 23, tiene como entradas a las características mencionadas de la tabla (*bnul, cont y dif*), y devuelve la cabecera correspondiente en forma de un vector de 5 bytes denominado *cab*. También proporciona una variable llamada *bandera,* que por defecto es 0, y solamente cambia su valor a 1 para indicar que en la tabla se debe encriptar el último paquete de contenido digital.

El primer procedimiento que realiza la función es convertir el número *cont* a tres bytes, que indicarán el número de paquete que se adjunta a la tabla y se ubican al inicio de la cabecera.

El siguiente proceso es comprobar el valor de *dif,* puesto que si este es mayor o igual a cero indica que la longitud del paquete ocupará todos los bytes nulos de la tabla y el valor de *bandera* es 0; y, si *dif* es menor que cero indica que el paquete no ocupará todos los nulos, y, por lo tanto, el tamaño del paquete será igual a la longitud del *vectordig,* y el valor de *bandera* cambia a 1.

La longitud del paquete correspondiente se convierte a una cadena de 14 bits y se concatena con el indicador de 2 bits, Tabla 2, para después pasar a formar los dos bytes finales de la cabecera.

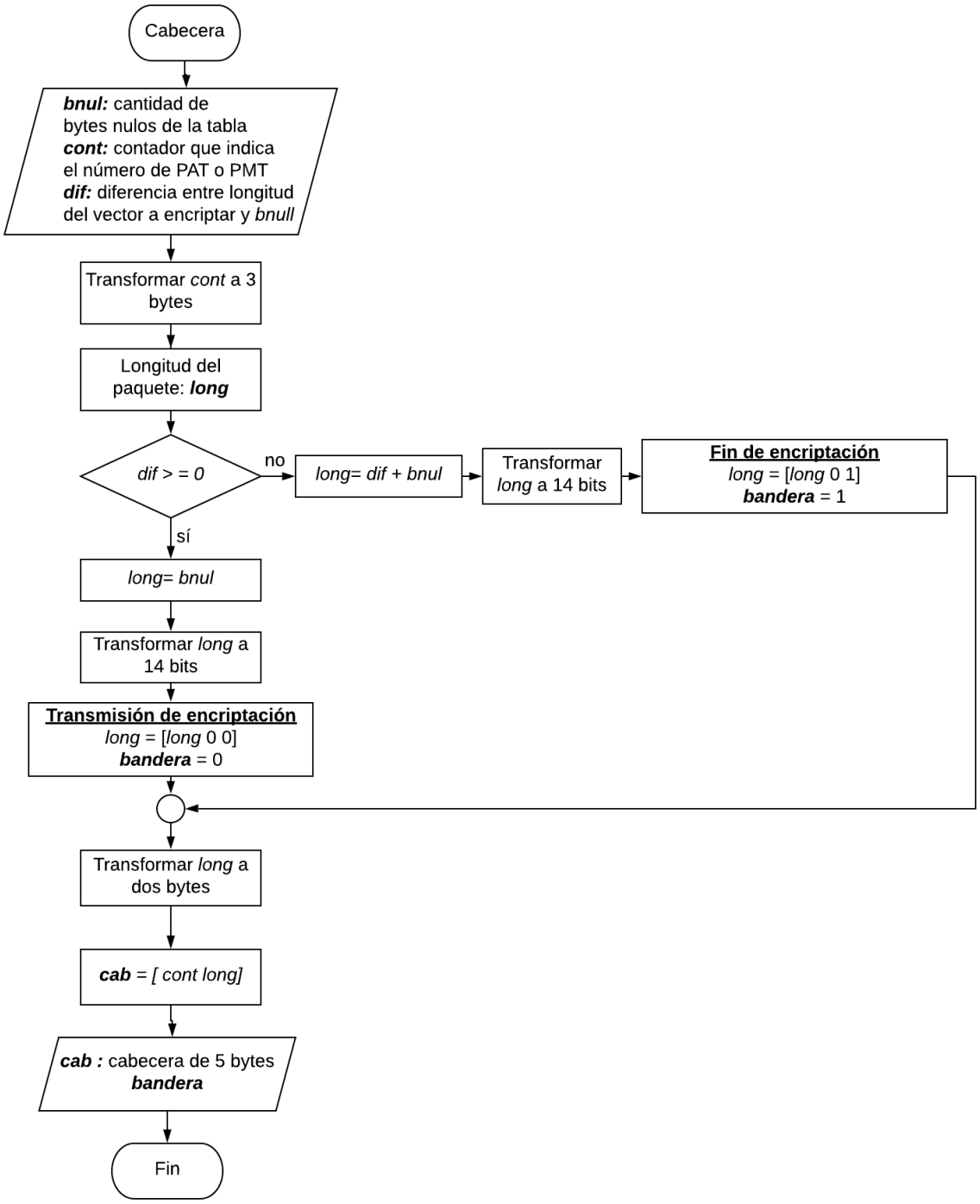


***Figura 23***. Diagrama de flujo de la función Cabecera

### 3.2.4.1. Ejemplo de la función cabecera

En la Figura 24, se muestra un ejemplo de la función cabecera. Para este caso, como indica la variable ***cont,*** se trata de la 10ma tabla PAT o PMT del flujo. La cantidad de bytes de relleno de la tabla es de 160 (***bnul***), y la diferencia entre el vector de contenido digital y los bytes nulos del paquete es 50 (***dif***), lo que significa que aún quedan por encriptar 50 bytes de contenido. Los primeros tres bytes de la cabecera son la numeración, en este caso ***cont*** = 10 en hexadecimal (A):

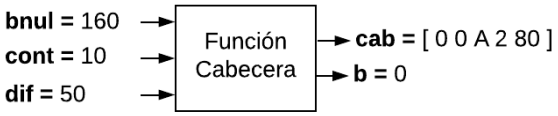


Figura 24 Ejemplo de la función cabecera

***cont*** = [ 0 0 A]

Como la variable ***dif*** es mayor que cero, ***bandera*** es 0 y la longitud del paquete en 14 bits está determinada por los bytes nulos, es decir:

***long = bnul*** = (00000010100000)

*El valor del indicador acorde a la* **Tabla 2**
Valores del indicadorTabla 2, en este caso corresponde al de transmisión por lo que es [0 0] y adjuntado a long y en formato hexadecimal:

***long*** = (00000010 10000**00**) = [ 2 80]

Y, finalmente, se concatena los bytes en un solo vector:

***cab*** = [ ***cont long***] = [ 0 0 A 2 80]

### 3.2.5. Encriptación de paquetes

En esta parte se comprueba el valor de *bandera* proporcionado por la función *cabecera.* En caso de que sea 0, en la variable *trama*, que corresponde a una tabla PAT o PMT, desde el primero hasta el último byte de relleno se escribe: (1) la cabecera "*cab"* seguida de los bytes de *vectordig* contenidos desde su primera posición hasta *bnul;* y se procede a recortar de *vectordig* la parte que ya fue encriptada. Si *bandera* es 1, es el turno de encriptar el último paquete de contenido digital y por tanto el paquete no ocupa todos los bytes nulos. En la variable *trama,* éste último paquete ocupará un espacio igual a la longitud sobrante del *vectordig*, y, para reiniciar el proceso de encriptación, en *vectordig* se almacena el contenido original almacenado en *voriginal*, y el contador se reinicia a 1. De esta forma el proceso se repetirá para todas las tablas PAT y PMT presentes en el flujo de forma ordenada y sucesiva. Finalmente se escribe *trama* en *TSencriptado,* y el proceso termina cuando todas las tablas y paquetes de audio, video y datos del archivo original son copiados en el nuevo archivo TS.

A través del algoritmo de transmisión se logra una nueva estructura de las tablas PAT y PMT, que se ilustra en la Figura 25, donde los bytes de relleno fueron reemplazados por paquetes de contenido digital, los cuales están precedidos por una cabecera que permitirá la reconstrucción del contenido digital en el receptor. El diagrama de flujo de todo el algoritmo de transmisión se presenta en la Figura 26.

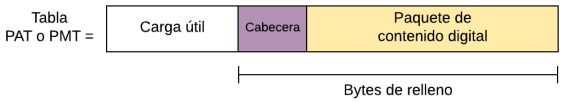


***Figura 25.*** Nueva estructura de las tablas PAT o PMT

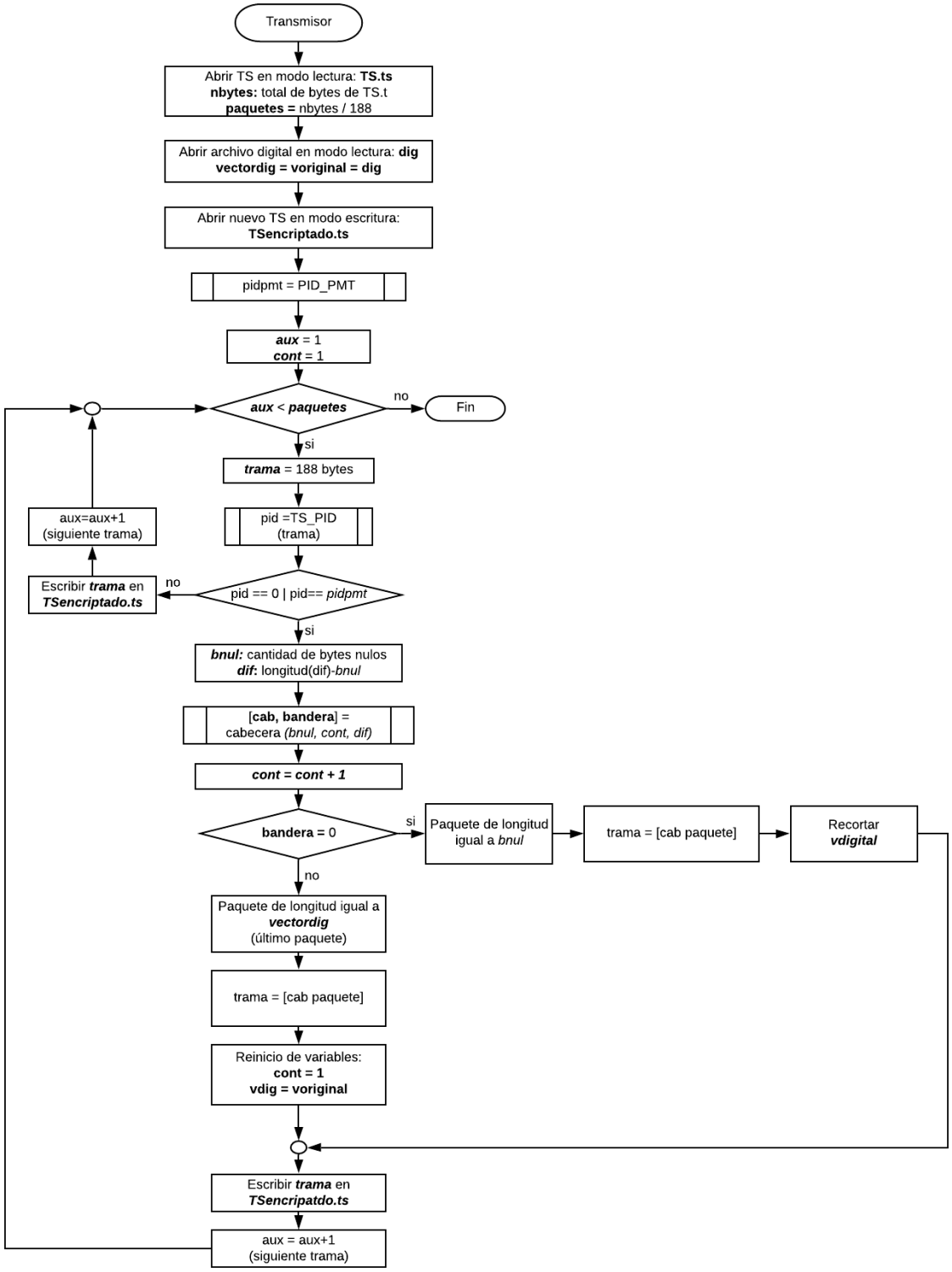


***Figura 26.*** Diagrama de flujo del algoritmo de recepción

## 3.3. Estructura del algoritmo de recepción

El algoritmo de recepción tiene como objetivo reconstruir el contenido digital encriptado en los bytes nulos de las tablas PAT y PMT, el cual está tres o más veces dentro del TS. Cada paquete que lo forma tiene una cabecera que permite reconstruir el contenido digital en forma de un vector para luego ser escrito en un nuevo archivo digital.

Se comienza al abrir el archivo TS (*TSencriptado.ts*) para tener información sobre el número de bytes y por ende el número de paquetes que lo conforman.

### 3.3.1. Funciones TS_PID y PID_PMT

Este algoritmo también utiliza el proceso de identificación de PID y obtención del PID de la PMT (ver Sección 2.1 y 2.2). El proceso consiste en buscar en el TS una tabla PAT, es decir, la tabla cuyo PID sea igual a 0X00, y posteriormente se sigue el proceso para la obtención del PID de PMT, que está ilustrado en la Figura 19 de la Sección 2.2. Posteriormente para cada paquete del TS se analiza su PID, ya que si éste corresponde al PID de una tabla PAT o PMT se debe ejecutar el proceso para recuperar el paquete de contenido digital.

### 3.3.2. Función ANÁLISIS

Otra de las funciones diseñadas en el algoritmo es la que permite analizar la cabecera que se adjunta al inicio de cada paquete, la cual corresponde a un vector de 5 bytes, y devuelve tres valores que corresponden a la numeración (*num*), su longitud (*long*) y una bandera (*bandera*) que indica si dicho paquete es el final. Como el vector tiene formato hexadecimal el primer paso es transformar el número de paquete a decimal, el

cual corresponde a los primeros tres bytes del vector. Seguidamente los dos bytes restantes se convierten a binario, y, del bit de la posición 1 a la 14 corresponde a la longitud y los dos últimos bits son el indicador del proceso de encriptación. (ver *Tabla 2* de la Sección 2.3). En caso de que el indicador sea igual a [0 1], es decir, que se trate del último paquete de contenido digital, la bandera que devuelve la función es de valor 1, caso contrario su valor es 0 por defecto. El diagrama de flujo se presenta en la Figura 27.

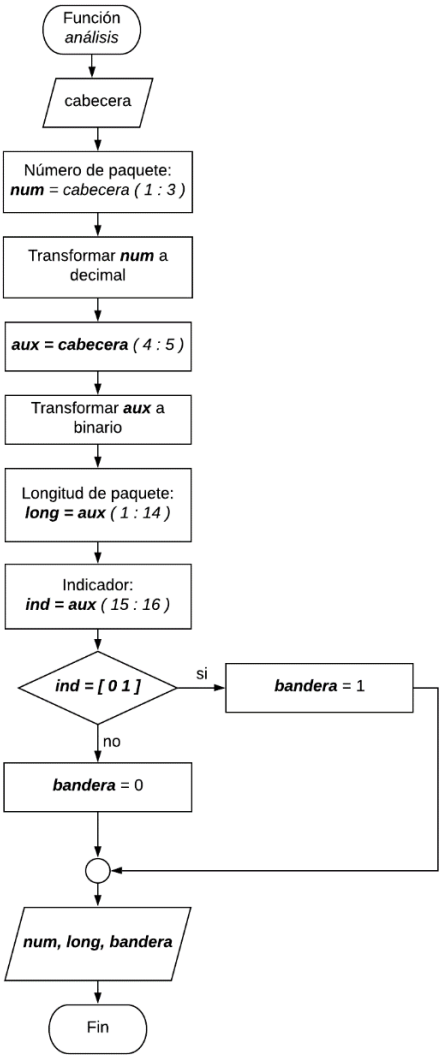


***Figura 27.*** Diagrama de flujo de la función análisis

### 3.3.3. Reconstrucción del contenido digital

En el proceso se pueden dar dos casos, Figura 28Figura *28.* Casos en la recepción :

1. La recepción se ejecutó desde el primer instante de la transmisión, es decir los paquetes de contenido digital se encuentran en orden dentro del flujo.
2. La recepción se ejecutó desde un instante desconocido por lo que el paquete final se encuentra antes del primero.

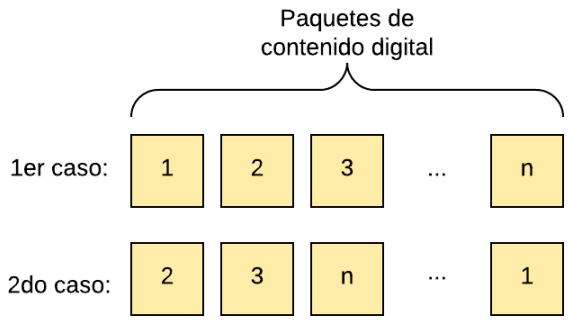


***Figura 28.*** Casos en la recepción

Por lo que, es necesario contar con las siguientes variables:

- Un contador que irá aumentando cada que se registre una tabla PAT o PMT.
- Un auxiliar cuyo valor inicial es 0 y que, cuando se registre el último paquete tomará el valor de su numeración proporcionado por la variable *num* de la función *análisis,* en otras palabras, almacenará el número total de paquetes de contenido digital.
- Dos vectores: *vectora* y *vectorb,* para almacenamiento.

Si el paquete que se está analizando corresponde a una tabla PAT o PMT, se guarda dentro de una variable denominada *trama,* Figura 29, y se extraen sus características como longitud de carga útil (*N*) y cantidad de bytes nulos. En la variable *cabecera* se toman los cinco primeros bytes de la carga de relleno que van de la posición *N*+9 a la *N*+12, y se utiliza la función *análisis* descrita en la anterior sección.

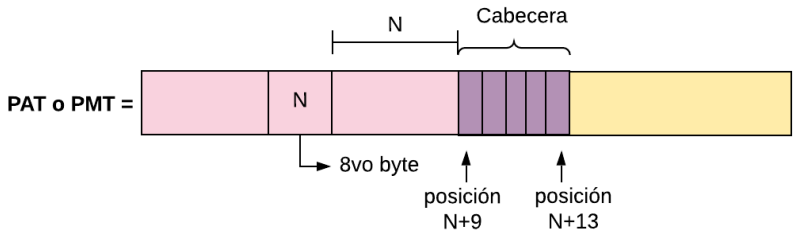


***Figura 29.*** Estructura de la tabla PAT o PMT

El *vectora* será el encargado de ir almacenando los paquetes de contenido digital, y, cuando la función devuelva una bandera igual a 1, significa que el algoritmo encontró el paquete final y se realiza lo siguiente: (1) el auxiliar toma el valor de la numeración de paquete y (2) el *vectora* se llena de los bytes correspondientes de *trama*. Seguidamente se comprueba:

- Si el contador es menor que el auxiliar (segundo caso), quiere decir que aún no se han recibido todos los paquetes, y se pasa el contenido recolectado hasta el momento del *vectora* al *vectorb*, y finalmente se procede a vaciar el *vectora,* Figura 30.

- Si el contador es igual al auxiliar corresponde el primer caso, y por tanto debe acabarse el proceso iterativo ya que se completó la recepción del contenido digital.

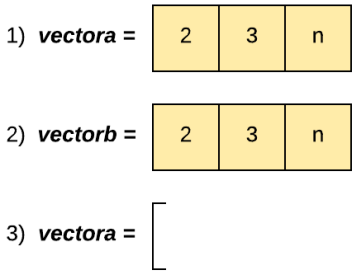


***Figura 30***. Segundo caso con bandera = 1

Si la bandera es cero, se debe comprobar:

- Si el contador es igual al auxiliar se trata del segundo caso. El *vectora* se llena de los bytes correspondientes, y se concatena con el *vectorb,* como se ilustra en la Figura 31. De esta forma, el contenido digital se ordena y el proceso iterativo se termina.

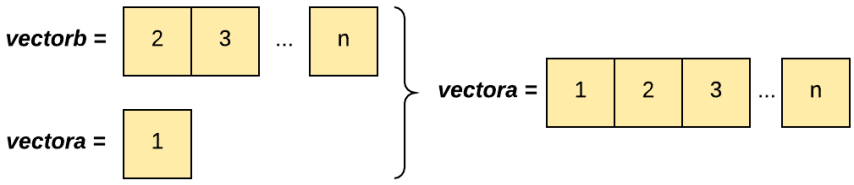


***Figura 31.*** Segundo caso con bandera = 0

- Si el contador es diferente al auxiliar es el primer caso y por lo tanto corresponde almacenar en el *vectora* el paquete correspondiente.

De esta forma, cuando se termine el proceso iterativo el *vectora* tiene almacenado el contenido digital de forma ordenada y el paso final es escribirlo dentro de un archivo que se abre en modo escritura con la extensión correspondiente. El diagrama de flujo corresponde a la Figura 32.

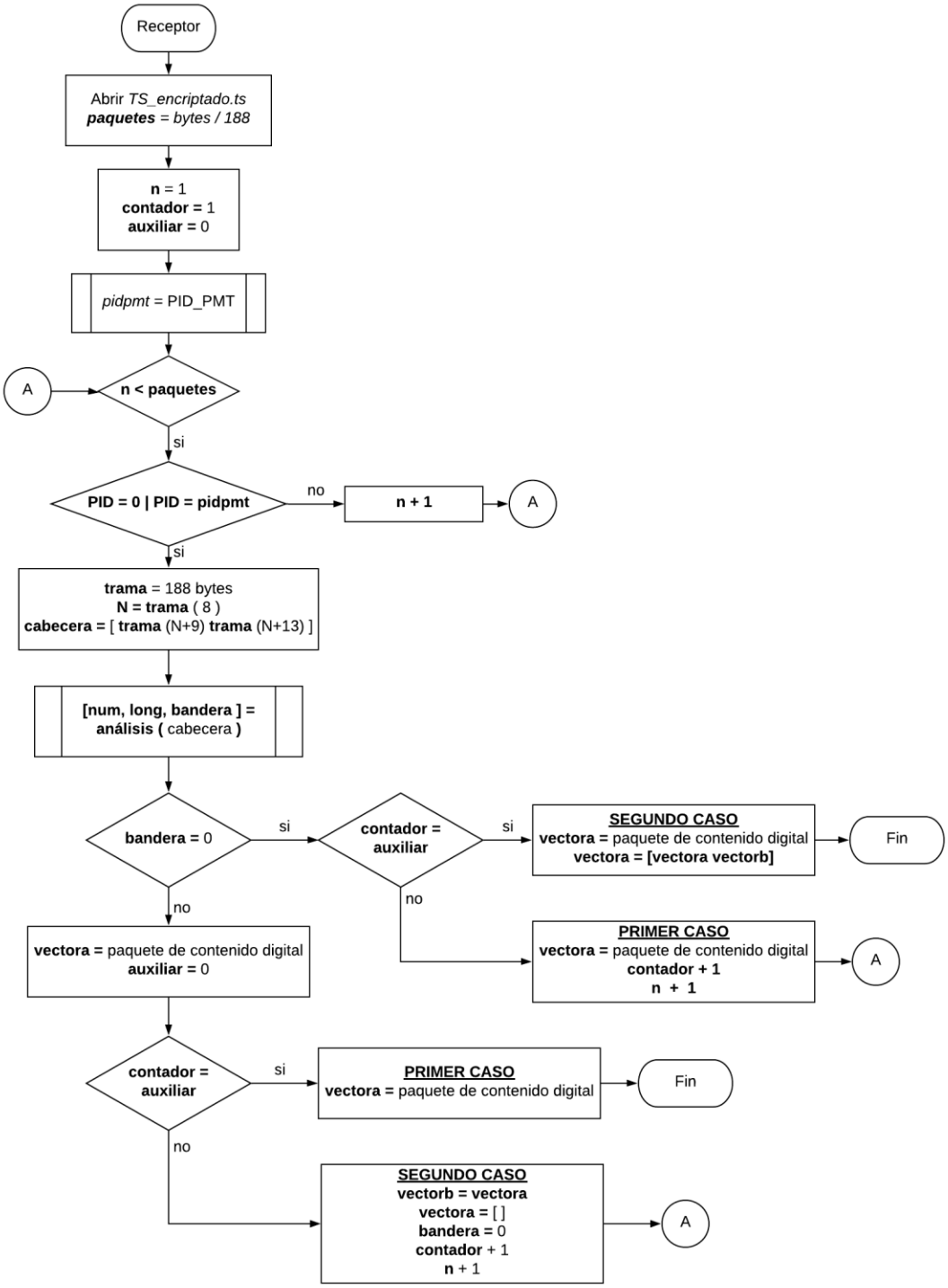


***Figura 32.*** Diagrama de flujo del algoritmo de recepción

# CAPÍTULO 4

# RESULTADOS

Será transmitido un archivo digital en un video TS, a través del algoritmo descrito en la Sección 2.2 del Capítulo 2 en un escenario real de laboratorio. La recepción de la señal será grabada para su posterior desencriptación mediante el algoritmo de recepción propuesto en la Sección 2.3 del Capítulo 3.

## 4.1. Ejecución del algoritmo de transmisión

Para ejecutar el algoritmo de transmisión es necesario contar con dos archivos, el primero corresponde al archivo TS, y el segundo al archivo es el de contenido digital que se va a encriptar en las tablas.

### 4.1.1. Características de archivo TS

En el presente ejemplo, el archivo elegido para ejecutar el algoritmo corresponde a un video de extensión TS, de duración igual a dos minutos y un tamaño de 40 490 312 bytes. A través del algoritmo se obtuvieron las características presentadas en la Tabla 3.

**Tabla 3**

*Características de TS obtenidas mediante el algoritmo de transmisión*

| **Característica** | **Valor** |
|---|---|
| PID de PMT | 4096 |
| Bytes medios de relleno de PAT | 167 |
| Bytes medios de relleno de PMT | 151 |

A continuación, muestra una captura del software TS & BTS ESPE – Analyzer, programa que permite visualizar en una interfaz gráfica cada parte del contenido de un TS (Benavides, 2015). A través de esta herramienta también es posible acceder a la información de las tablas, entre ellos, el PID de PMT, cuyo valor se comprueba con el obtenido a través del algoritmo, Figura 33.

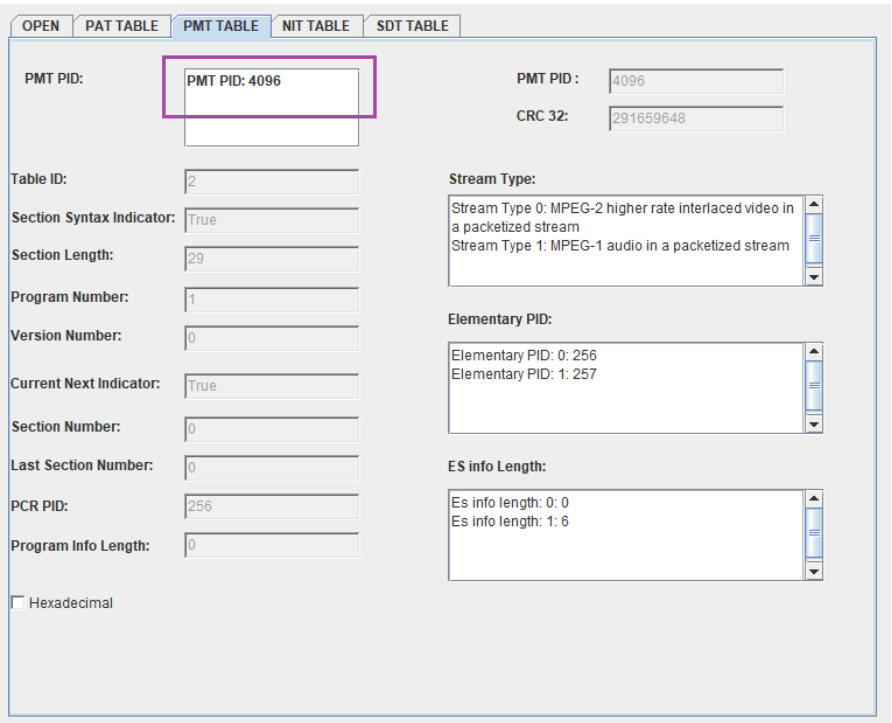


***Figura 33.*** Valor de PID de PMT con el software TS & BTS ESPE-Analyzer

En la siguiente captura, Figura 34, se visualiza una tabla PAT del TS, cuyo PID es 0 y la carga útil corresponde a 21 bytes, mientras que los de relleno son los 167 bytes restantes y se presentan con el valor de 0XFF.

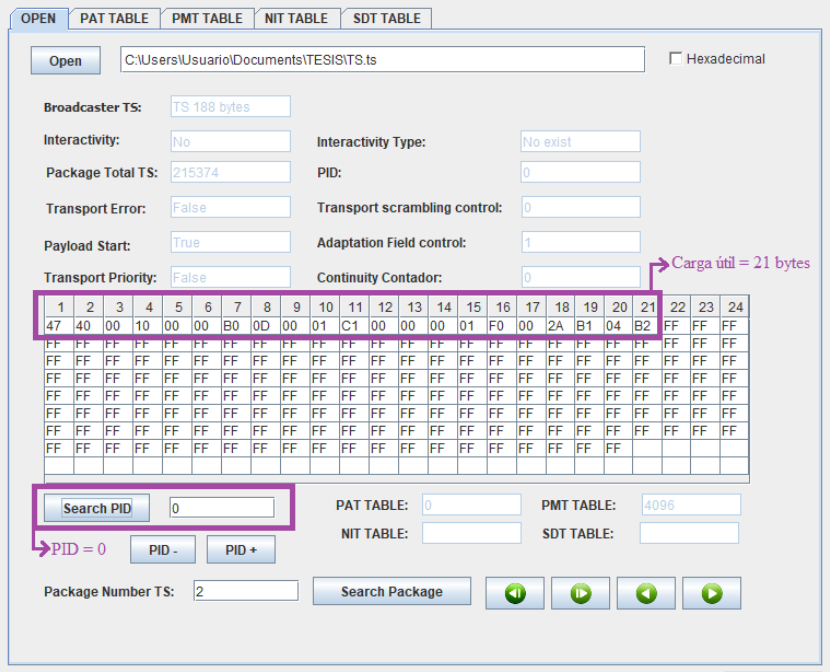


***Figura 34.*** Análisis de PAT con el software TS & BTS ESPE-Analyzer

Así mismo, en la Figura 35, se muestra una tabla PMT, con su carga útil de 37 bytes, y por lo tanto la carga de relleno corresponde a los 151 bytes restantes de valor 0XFF.

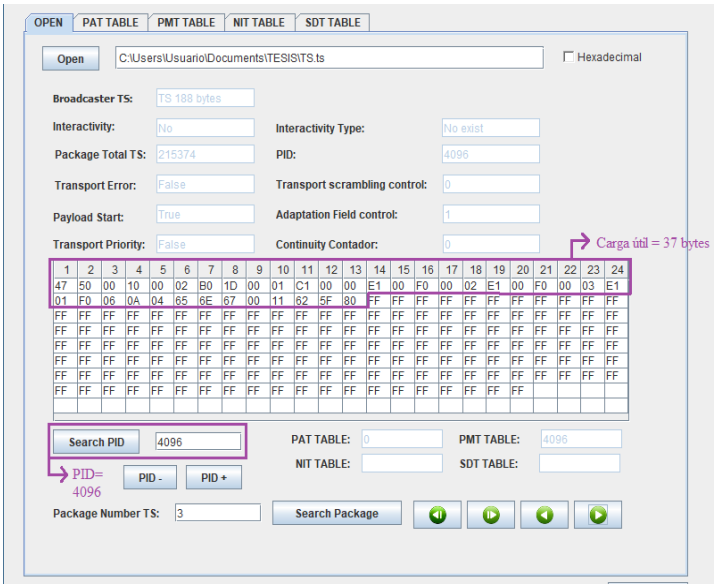


***Figura 35.*** Análisis de PMT con el software TS & BTS ESPE-Analyzer

### 4.1.2. Características de contenido digital

Las pruebas para el algoritmo se realizaron con dos tipos de contenido digital. El primero consiste en una imagen y la segunda un vector cuya aplicación se explicación en la sección 1.2.2. Para constatar que los dos archivos puedan encriptarse dentro del flujo tres o más veces, se utiliza la Ecuación 1, sección 2.4 Capítulo III, que permite conocer el tamaño máximo permitido (*n*) conociendo las características del TS,

$$n = \frac{(167 + 151)[\text{bytes}] \times 120[\text{s}]}{3 \times 100 \times 10^{-3}[\text{s}]} = 127\ 200\ [\ \text{bytes}\ ]$$

#### 4.1.2.1. Imagen

La imagen utilizada es de tipo JPG, Figura 36, su tamaño es 85 601 bytes, por lo que cumple la condición, y tiene una dimensión es de 1024 x 768 pixeles.

***Figura 36*.** Imagen JPG encriptada dentro del flujo

#### 4.1.2.2. Stream de datos

El trabajo realizado en (Acosta, 2018) presenta un sistema donde se obtiene una imagen mosaico (varias imágenes que forman una) a través de reordenar los bloques de una imagen secreta (*secret image*), disfrazada de otra (*target image*), Figura 37. Para recuperar la imagen secreta en el receptor, se envía adicionalmente un stream de datos denominado $E_T$, Figura 38,el cual fue seleccionado para encriptarse dentro de las tablas del flujo. El stream tiene un tamaño de 77 456 bytes y su formato es MAT.

(a) (b) (c)

***Figura 37*.** Imágenes del sistema. (a) *Secret image*; (b) *Target image*; (c) *Mosaic image*

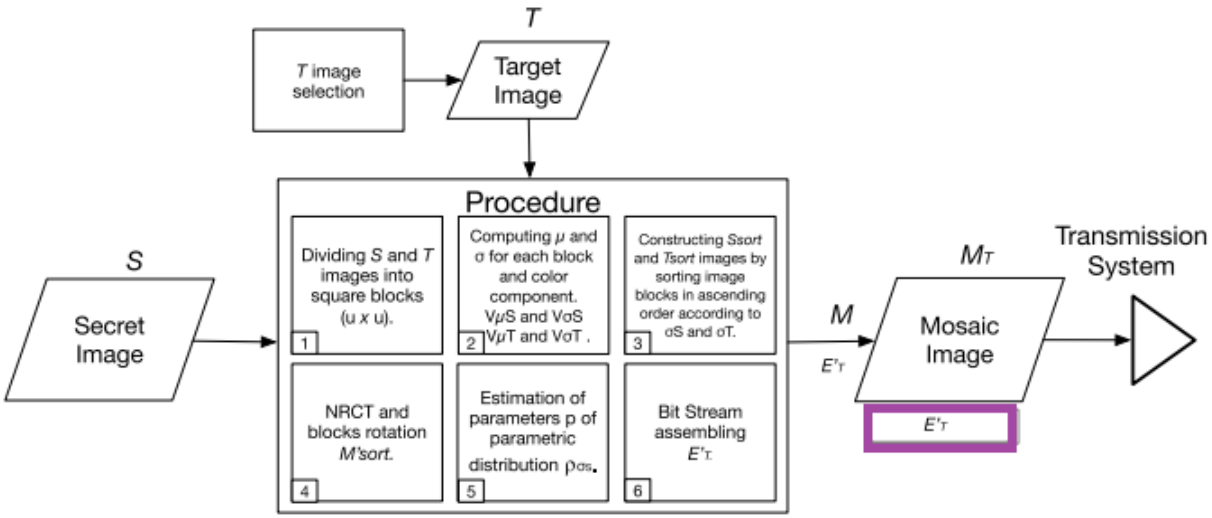


***Figura 38*.** Diagrama de bloques del sistema de transmisión, (Acosta, 2018)

Al abrir el archivo en MatLab, se identificó que el stream contenía bits dentro de una cadena de caracteres, por lo que para encriptarlo dentro del TS, se cambió su estructura a bytes de formato decimal y se almacenó en un vector.

## 4.2. Resultados del algoritmo de transmisión

Al ejecutar el algoritmo de transmisión diseñado, se obtiene un nuevo archivo TS denominado *TSencriptado.ts,* que resulta en una copia del archivo original, es decir, si se abre con un reproductor de TS, se observa que el contenido de video y audio no están alterados. La diferencia radica en sus tablas PAT y PMT que llevan paquetes del contenido digital.

### 4.2.1. Resultado de algoritmo con encriptación de imagen

Para constatar que la imagen se encuentra en forma de paquetes dentro de las tablas se utilizó nuevamente el analizador ESPE-Analyzer. En la Figura 39, se muestra una tabla PAT, cuyos bytes de relleno cambiaron su valor de 0XFF por los bytes que corresponden a los de la imagen. Al inicio de cada paquete se observa la cabecera correspondiente a 5 bytes y el resto a la carga del contenido digital. También se presenta una tabla PMT, Figura 40, cuya carga de relleno almacena el paquete correspondiente, que en este caso es el segundo, indicado por su cabecera.

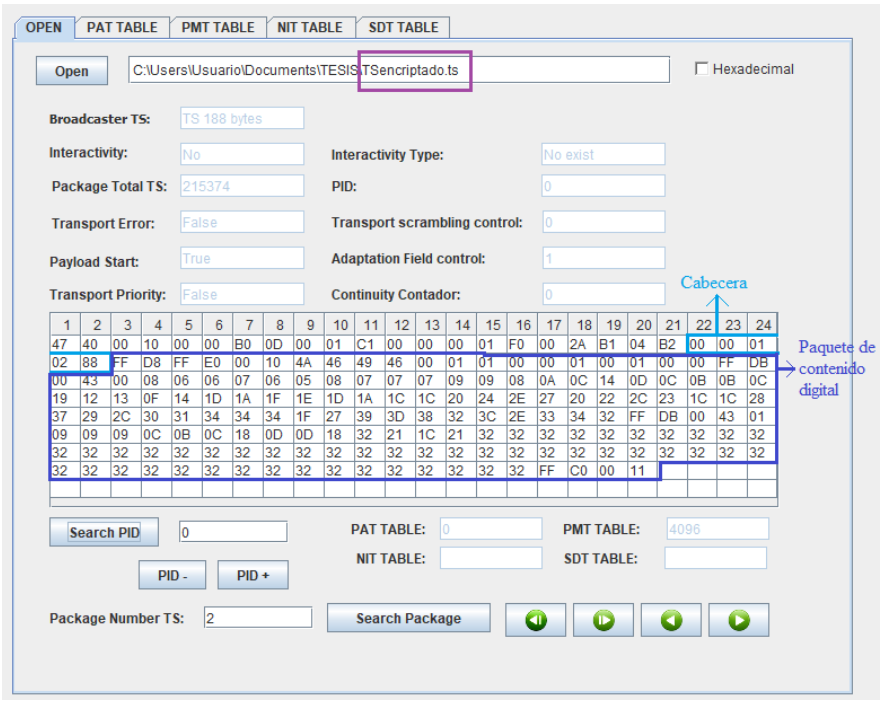


***Figura 39***. Tabla PAT con paquete correspondiente a la imagen encriptada

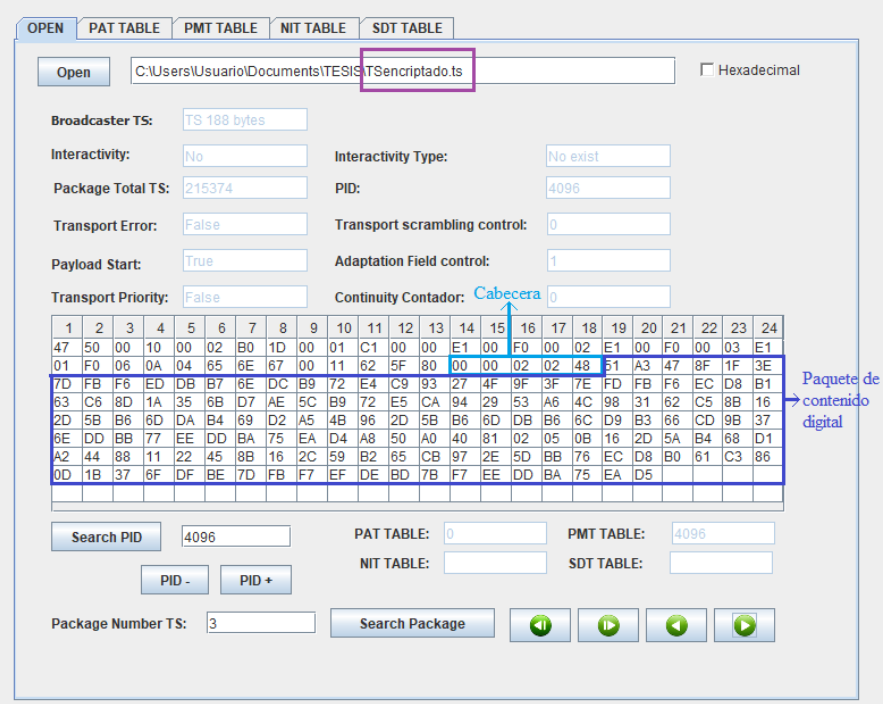


***Figura 40*.** Tabla PMT con paquete correspondiente a la imagen encriptada

### 4.2.2. Resultado de algoritmo con encriptación de stream de datos

De igual forma que en la sección anterior, mediante ESPE-Analyzer se comprobó que los bytes del stream se encuentran en la carga de relleno de las tablas, Figura 41 y Figura 42. Los paquetes cuentan con la cabecera de cinco bytes que permiten identificar que se trata del primer y segundo paquete dentro del flujo.

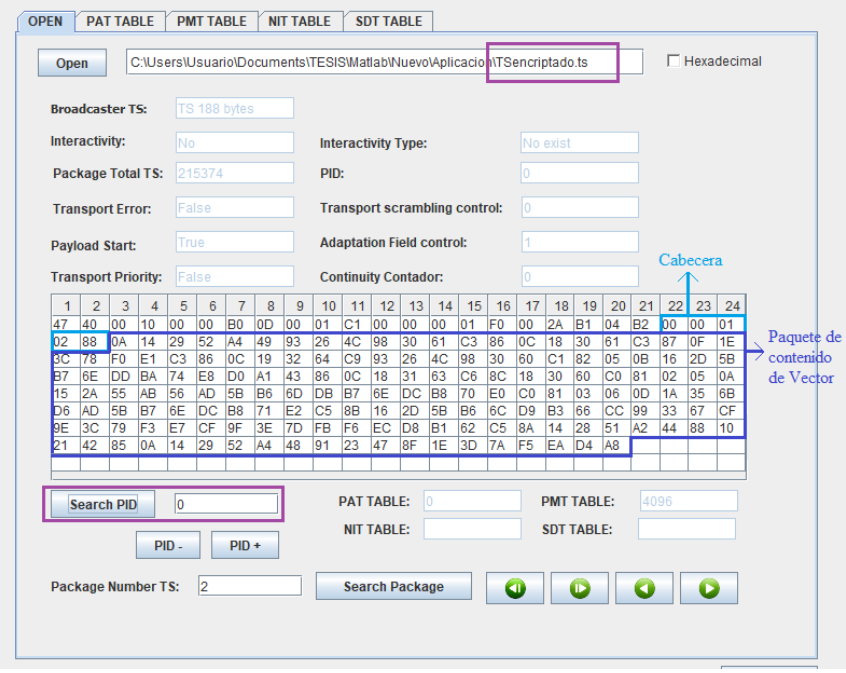


***Figura 41***. Tabla PAT con paquete de contenido de Vector

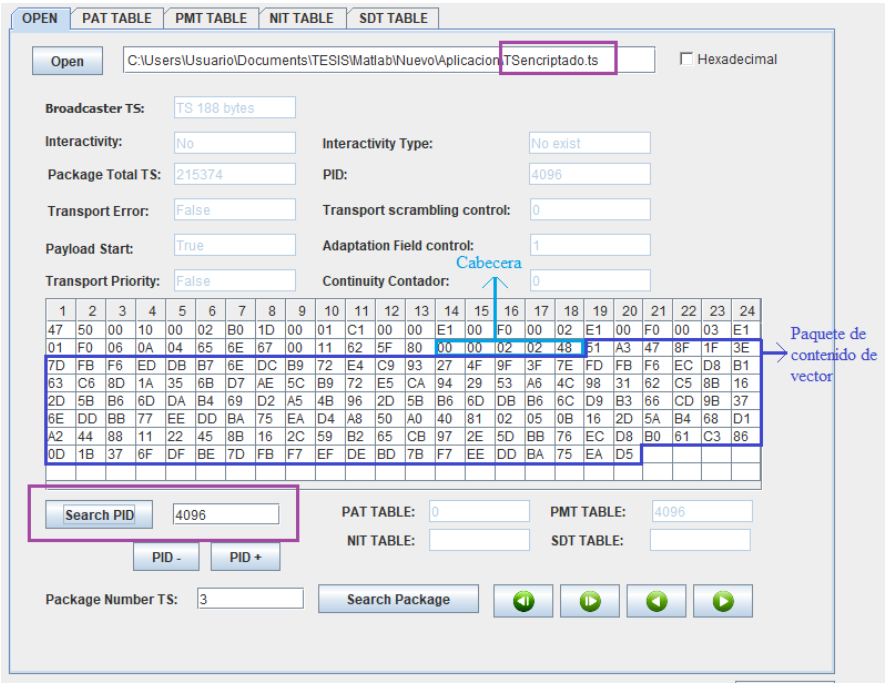


***Figura 42.*** Tabla PMT con paquete de contenido de Vector

## 4.3. Transmisión y recepción del TS en tiempo real

Una vez obtenidos los archivos de video con el contenido encriptado de la imagen y el vector, se procedió a realizar la transmisión y recepción en tiempo real en el laboratorio de Televisión Digital de las Fuerzas Armadas ESPE, con los equipos detallados en el Capítulo 2 sección 2.6.

### 4.3.1. Escenarios

Las pruebas se realizaron en dos escenarios. El primero, Figura 43, es mediante una conexión directa entre la tarjeta moduladora y el analizador, a través de un cable coaxial conectados a las entradas/salidas ASI de los dispositivos.

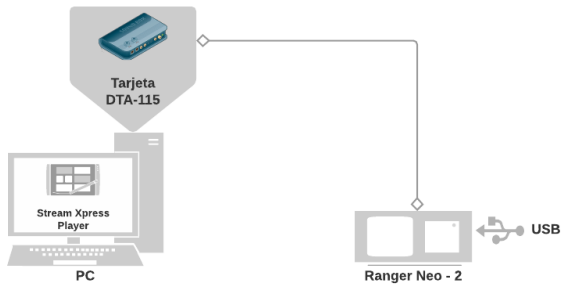


***Figura 43***. Escenario de conexión cableada

El segundo escenario, Figura 44, corresponde a una conexión RF. Tanto en la tarjeta como en el analizador se conectaron antenas a las entradas/salidas RF correspondientes. Además, en los dos escenarios se cuenta con un pendrive USB conectado al analizador para el almacenamiento de la información.

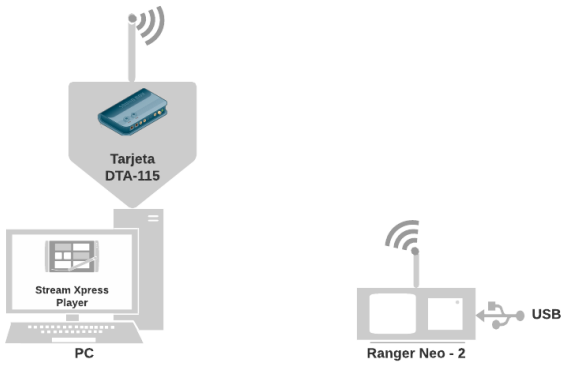


***Figura 44.*** Escenario de conexión RF

Las características de los archivos TS recibidos, Tabla 4, muestran el tiempo de duración, tamaño en bytes, y total de tablas PAT y PMT presentes en el flujo recibido para cada una de las aplicaciones y en los escenarios mencionados.

**Tabla 4**

*Características de los archivos TS recibidos*

| **Archivo encriptado** | **Características** | **Conexión RF** | **Conexión cableada** |
|---|---|---|---|
| Imagen | Tiempo de duración | 1 minuto y 3 segundos | 33 segundos |
| | Tamaño del TS | 268 025 020 bytes | 218 405 052 bytes |
| | Total de tablas PAT | 2 269 | 1 825 |
| | Total de tablas PMT | 2 268 | 1 825 |
| Stream de datos | Tiempo de duración | 40 segundos | 30 segundos |
| | Tamaño del TS | 212 822 204 bytes | 197 974 716 bytes |
| | Total de tablas PAT | 1 786 | 1 652 |
| | Total de tablas PMT | 1 786 | 1 652 |

## 4.4. Resultados del algoritmo de recepción

### 4.4.1. Resultado del algoritmo en escenario con conexión cableada

#### 4.4.1.1. Imagen

En la Figura 45 se observa una captura del analizador correspondiente a una tabla PAT del archivo TS recibido. En este caso, la tabla está almacenando el paquete número 235 de la imagen, indicado por el valor de su cabecera (0xEB). El algoritmo devuelve la imagen, Figura 46, la cual mantuvo las características de la imagen original, es decir, tamaño igual a 85 601 bytes y dimensión de 1024 x 768 pixeles.

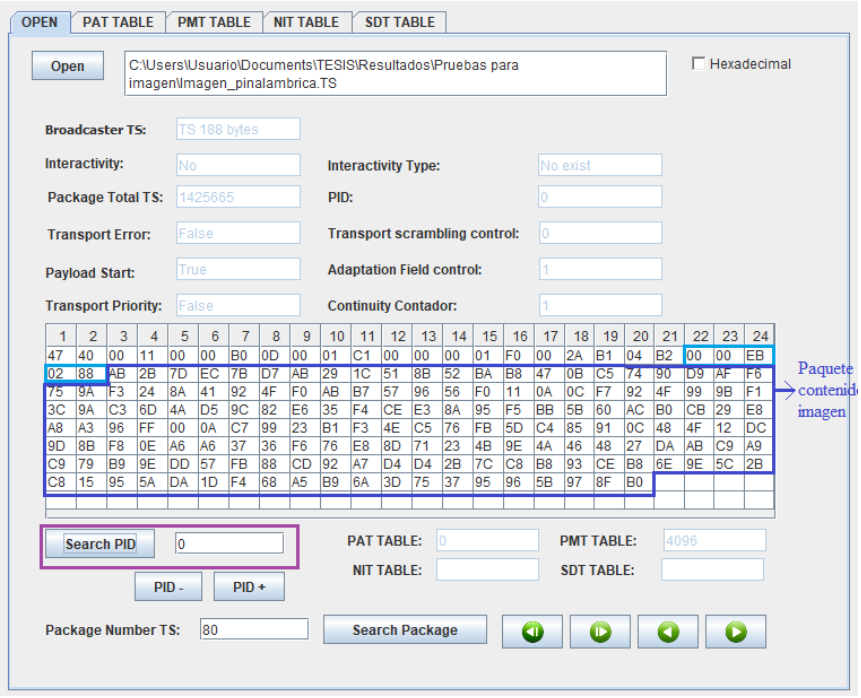


***Figura 45.*** Tabla PAT del TS recibido en un escenario con conexión cableada con contenido digital de imagen

***Figura 46.*** Imagen reconstruida con el algoritmo de recepción en un escenario con conexión cableada

#### 4.4.1.2. Stream de datos

Con el analizador de TS se abre el video recibido y se comprueba que la carga de relleno de las tablas tiene el contenido del stream. En la primera tabla PAT, Figura 47, el paquete corresponde al número 107 del vector, y el algoritmo de recepción se encarga de reconstruir y ordenarlos para recuperarlo.

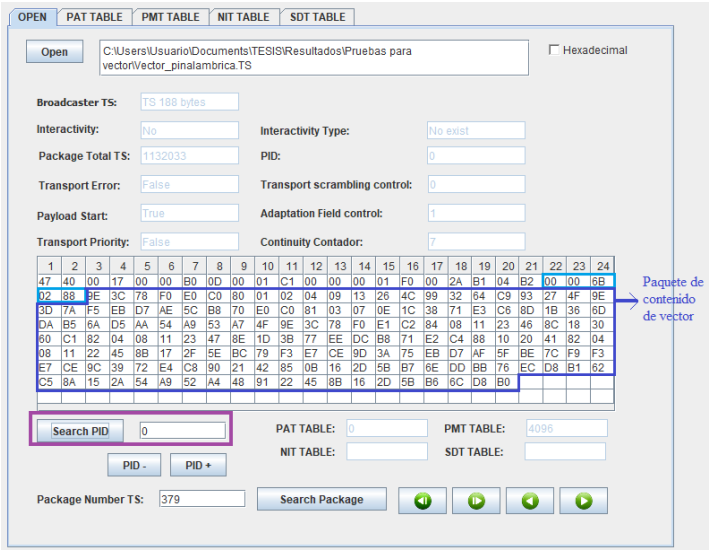


***Figura 47.*** Tabla PAT del TS recibido en un escenario con conexión cableada con contenido digital de vector

Debido a que se cambió el formato del stream original para su encriptación, para comprobar la validez del algoritmo fue necesario regresarlo a la cadena de caracteres. Para esto, se transformó el vector de hexadecimal a binario y luego al respectivo formato. Con la función *isequal* de MatLab se comprobó que son iguales, Figura 48 .

```
>> isequal(srecibido,soriginal)

ans =

     1
```

***Figura 48.*** Función *isequal* de MatLab para comprobar el stream recibido en conexión cableada

Una vez recuperado el stream, $E_T$, se logró recuperar la imagen secreta (*recovered secret image*), Figura 49 , dentro del sistema de recepción propuesto en (Acosta, 2018), Figura 50.

***Figura 49.*** *Recovered secret image,* (Acosta, 2018)

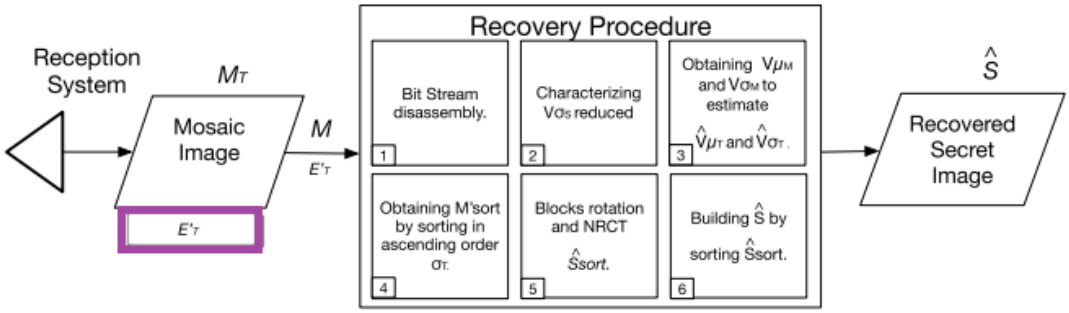


***Figura 50.*** Diagrama de bloques del sistema de recepción, (Acosta, 2018)

### 4.4.2. Resultado del algoritmo en escenario con conexión RF

#### 4.4.2.1. Imagen

El algoritmo de recepción se encarga de reconstruir la imagen, Figura 51, la cual conservó las características de la original. Se abre el archivo TS recibido con el

analizador, y se identifica una tabla PAT, Figura 52. En este caso, el primer paquete de contenido recibido, en realidad es el número 91 (0x5B) de la imagen.

***Figura 51.*** Imagen reconstruida con el algoritmo de recepción en un escenario con conexión RF

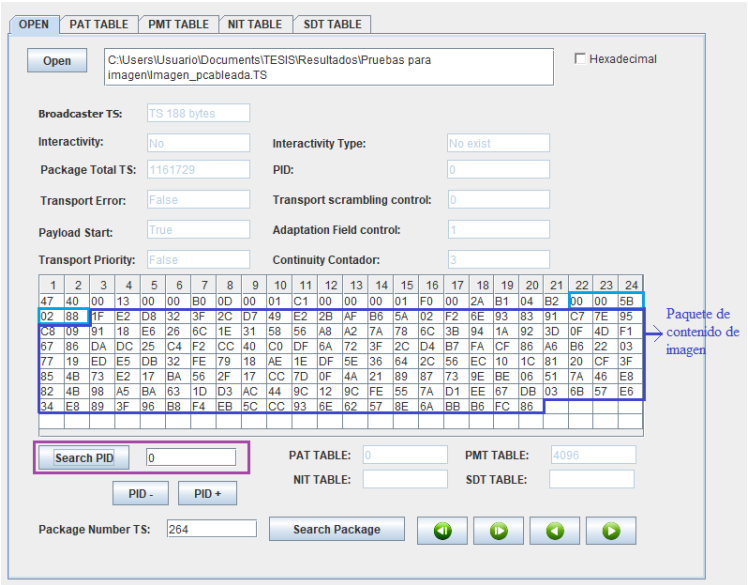


***Figura 52.*** Tabla PAT del TS recibido en un escenario con conexión RF con contenido digital de imagen

### 4.4.2.2. Stream de datos

Con el archivo TS que se recibió se ejecutó el algoritmo de recepción, y a través del analizador se muestra la primera tabla PAT presente en el flujo, Figura 53, que corresponde al paquete número 67 del vector. Haciendo uso de la función *isequal* del software se comprueba que el stream original y el recibido son iguales, Figura 54.

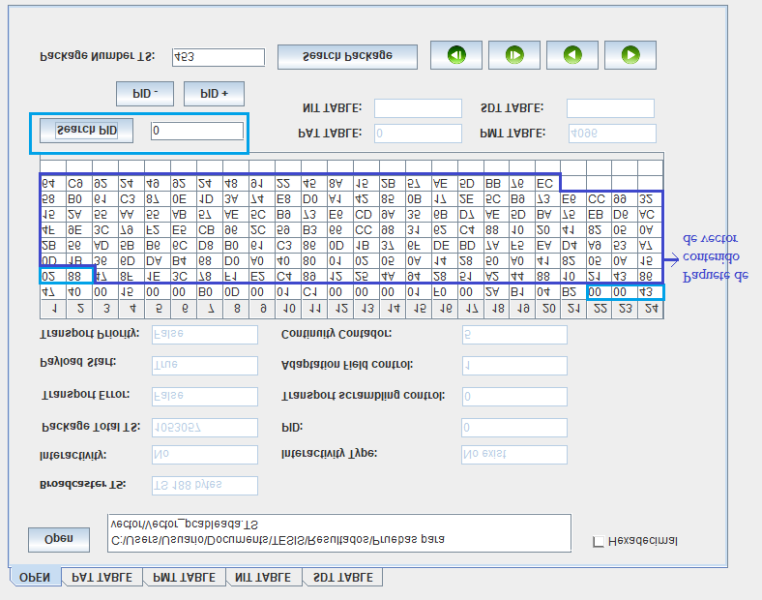

***Figura 53***. Tabla PAT del TS recibido en un escenario con conexión RF con contenido digital del vector

```
>> isequal(srecibido,soriginal)

ans =

     1
```

***Figura 54.*** Función *isequal* de MatLab para comprobar el stream recibido en conexión RF

Al recuperar el stream, dentro del sistema del receptor, (Acosta, 2018), se logró recuperar la imagen secreta, Figura 55.

***Figura 55.*** *Recovered secret image,* (Acosta, 2018)

Para finalizar con el capítulo, en la Figura 56 se muestra un código QR con el link a un video que muestra el funcionamiento de los dos algoritmos dentro de un escenario real de Televisión Digital Terrestre.


***Figura 56****.* Código QR del video de funcionamiento de los algoritmos

# CAPÍTULO 5

# CONCLUSIONES Y RECOMENDACIONES

## 5.1. Conclusiones

- Se diseñó un algoritmo capaz de encriptar contenido digital dentro de los bytes de relleno de las tablas PAT y PMT del flujo de transporte de TDT.
- Se diseñaron funciones dentro de los algoritmos para la identificación de PID de las tablas PAT y PMT de un TS.
- Se planteó una ecuación para determinar el tamaño de archivo de contenido digital que puede encriptarse dentro de un TS, considerando su tiempo de duración y los bytes de relleno de cada tabla, para que de esta forma se asegure su reconstrucción en el receptor.
- Se implementó una función dentro del algoritmo de transmisión para asignar una cabecera que indica el número de paquete de contenido digital, así como su longitud, y una bandera que indica el estado del procedimiento; características que permiten reconstruir los paquetes en el receptor.
- Se comprobó mediante el software TS-ESPE Analyzer el funcionamiento del algoritmo de transmisión, al observar el cambio de valor hexadecimal de los bytes de relleno de las tablas.
- Se diseñó un algoritmo de recepción para ordenar y reconstruir el archivo digital contenido en las tablas PAT y PMT del TS.

- Se implementó una función dentro del algoritmo de recepción que distingue la información de cabecera de cada paquete, para de esta forma ordenar en un solo vector todos los paquetes que corresponde al contenido digital encriptado.
- Se comprobó dentro de un escenario con conexión cableada, así como dentro de un escenario con conexión RF, que los bytes de relleno de las tablas PAT y PMT del flujo no se ven alterados por la tarjeta al momento de la transmisión.
- Se comprobó dentro de un escenario con conexión cableada, así como dentro de un escenario con conexión RF, que el audio, video y datos de un TS no se ven alterados después de haber encriptado contenido dentro de los bytes de relleno de sus tablas.
- Se transmitieron y receptaron los TS con contenido encriptado a través de la tarjeta DTA-115, el programa Stream Xpress TS Player y el Analizador Ranger Neo2, en dos escenarios, cableado y RF, en el laboratorio de TV Digital de la Universidad de las Fuerzas Armadas ESPE.
- Se comprobó que al encriptar el contenido digital tres o más veces dentro del TS se asegura la reconstrucción en el receptor.
- Se recuperó el contenido digital de la imagen a través del algoritmo de recepción, y se constató que mantuvo las características de la original.
- Se recuperó el stream de datos en el receptor, el cual, mantuvo sus características dentro del proceso de transmisión y recepción.

## 5.2. Recomendaciones

- La encriptación de contenido se debe realizar solamente dentro de los bytes de relleno de las tablas PSI/SI, puesto que se comprobó que los paquetes nulos del TS se autoconfiguran al momento de la transmisión en escenarios reales, por lo que se pierde la información que fue encriptada.
- Se debe comprobar que la única información que fue modificada dentro del TS fueron los bytes de relleno de las tablas, puesto que cualquier alteración de información útil impide la reproducción del archivo TS.
- Es importante comprobar que el contenido digital se encuentra en orden dentro del flujo a transmitir mediante un analizador de TS, para evitar errores al momento de la recepción.

## 5.3. Trabajos futuros

Como trabajo futuro se plantea la adecuación de los algoritmos para trabajar con más de una tabla PMT, es decir, con varios servicios, puesto que los algoritmos del presente trabajo están diseñados para un solo servicio.

Además, se propone también la inclusión de otras tablas PSI/SI dentro del proceso, lo cual significaría una mayor capacidad en cuanto al tamaño de archivo que se va a encriptar.

Se plantea también como trabajo futuro la utilización de un algoritmo de cifrado para transformar el contenido digital en una secuencia ininteligible que solo sea accesible

para aquellos que dispongan la clave para descifrarlo, de esta forma, se protegería aún más el contenido que está siendo transmitido, asegurando así su confidencialidad.

**REFERENCIAS BIBLIOGRÁFICAS**

ABNT. (3 de 11 de 2007). ABNT NBR 15603-1. *Televisión digital terrestre - Multiplexación y servicios de información (SI). Parte 1: SI del sistema de radiodifusión*. Río de Janeiro.

Acosta, F. (2018). Data Amount Reduction in Mosaic Image Transmission Techniques for Digital Interactive Television Applications. *IEEE Access*, 7283-70297.

Benavides, N. (2015). *Desfragmentador del flujo de transporte (TS) y analizador de tablas para el sistema de Televisión Digital Terrestre ISDB-T.* Sangolquí.

Conocimiento, M. C. (6 de Mayo de 2013). *Ecuador inauguró Televisión Digital Terrestre*. Obtenido de Noticias: https://web.archive.org/web/20130706155033/http://www.conocimiento.gob.ec/ecuador-inauguro-television-digital-terrestre/

Fairhust, G. (Enero de 2001). *Electronics Research Group.* Obtenido de MPEG-2 Transmission: https://erg.abdn.ac.uk/future-net/digital-video/mpeg2-trans.html

Mathworks. (s.f.). *MATLAB- El lenguaje del cálculo técnico*. Recuperado el 15 de 04 de 2019, de https://la.mathworks.com/products/matlab.html

Metro Ecuador. (22 de Septiembre de 2018). Apagón analógico: Hasta el 2023 deberás tener un televisor digital o un decodificador en tu hogar. *Metro Ecuador*.

MINTEL. (2018). *Plan maestro de transición a la televisión digital terrestre.* Quito.

Mintel. (s.f.). *Ciudades con cobertura tdt*. Obtenido de tdt- televisión digital terrestre: https://tdtecuador.mintel.gob.ec/normativas-para-concesionarios-de-senal-abierta/

Mintel. (s.f.). *Contexto internacional*. (tdt - televisión digital terrestre) recuperado el 13 de julio de 2019, de https://tdtecuador.mintel.gob.ec/contexto-internacional-2/

Núñez, a. (2016). *Estudio, análsis e implementación de un software para construir un extractor y constructor de carrusel de datos incluido en el fluo de transporte mpeg-2 ts de televisión digital.* Sangolquí.

Organización de telecomunicaciones de iberoamérica. (mayo de 2017). *Otitelecom.* Recuperado el 2019 de 07 de 13, de https://www.otitelecom.org/wp-content/uploads/2017/05/oti-mejores-practicas-en-la-transicion-a-la-television-digital-terrestre-tdt.pdf

Reinoso, ó., jiménez, l., payá, l., aparicio, a., & peidró, a. (2018). *Matlab: conceptos básicos y descripción gráfica.* Universitas miguel hernández.

Song, j., yang, z., & jun, w. (2015). *Digital terrestrial television broadcasting: technology and system.* Canadá: wiley.

Yánez, d. (26 de noviembre de 2015). Diseño y desarrollo de guías de laboratorio para generar y manejar el flujo de transporte de televisión digital. Sangolquí, pichincha, ecuador.

Yánez, d. (2015). *Diseño y desarrollo de guías de laboratorio para generar y manejar el flujo de transporte de televisión digital.* Sangolquí.

Zuffo, m. K. (2008). *Tv digital aberta no brasil.* Obtenido de políticas estruturais para um modelo nacional: http://blog.joaomattar.com/2007/07/08/gestao-estrategica-de-negocios-editoriais.