

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, REDES Y
COMUNICACIÓN DE DATOS

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERÍA

DESARROLLO DE UN SOFTWARE ANALIZADOR DE
TRANSPORT STREAM PARA ISDB-TB EMPLEANDO
MATLAB Y LENGUAJE DE PROGRAMACIÓN
MULTIPLATAFORMA

EDISON FABRICIO NARANJO ESPÍN

Sangolquí - Ecuador

2012

Certificación

Certificamos que el presente proyecto de grado titulado **DESARROLLO DE UN SOFTWARE ANALIZADOR DE TRANSPORT STREAM PARA ISDB-Tb EMPLEANDO MATLAB Y LENGUAJE DE PROGRAMACIÓN MULTI-PLATAFORMA** bajo nuestra dirección, ha sido desarrollado en su totalidad por el Sr. Edison Fabricio Naranjo Espín con CI:1712439601.

Atentamente

Ing. Gonzalo F. Olmedo C., PhD. DIRECTOR	Ing. Darwin O. Alulema F., MSc. CODIRECTOR

Resumen

El presente proyecto de grado abarca el proceso de desarrollo de una aplicación de Software denominada Analizador de Paquetes Transport Stream para ISDB-Tb.

Inicialmente se recopila información referente a la formación de los Flujos de Transporte (Transport Stream) MPEG-2, la estructura de su cabecera, campo de adaptación y la información específica de Programas y Servicio que éstos pueden contener.

Posteriormente se detallan las Especificaciones de Requisitos de Software con referencia a la norma IEE-830-STD y en base a éstas se elabora la Ingeniería de Software para el Analizador de Paquetes Transport Stream a través del modelado visual empleando UML (Unified Modeling Language) poniendo énfasis en el paradigma de la Programación Orientada a Objetos (P.O.O).

Finalmente se realizan pruebas de funcionamiento del Analizador de Paquetes Transporte Stream tanto en MATLAB como en Java, ya que éstas fueron escogidas como herramientas de desarrollo de Software para este proyecto. Los casos de prueba se enfocan en la búsqueda de paquetes de transporte y generación de reportes. Adicionalmente se realiza una comparación del rendimiento en la generación de reportes empleando distintos Sistemas Operativos tales como Microsoft Windows y Linux.

Dedicatoria

A mi madre, por enseñarme que nunca hay que rendirse sin pelear, por brindarme su apoyo incondicional, su amor, sus desvelos y sembrar en mí los más altos valores de responsabilidad, honradez y humildad. Ella siempre pensó que podía llegar yo muy lejos, confió en mi talento, disfrutó y se sintió muy orgullosa de mis logros alentándome a seguir luchando cada día para ser mejor.

A mi padre, por su tenacidad ya que con su esfuerzo me ayudó a que cambiara la historia deseándome siempre que fuera el mejor.

A mis hermanos, quiero agradecerles sinceramente porque dedicaron parte de su tiempo a ayudarme en el difícil trayecto para continuar mi camino y crecer.

En general, agradezco a mi familia, por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida. Gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.

A los que me olvidaron sin darme derecho de mostrar mi talento apenas mi estrella dejó de brillar, con la consecución de esta carrera les demuestro que a pesar de todo nada pudo vencer mi entereza y ferviente deseo de éxito que desde niño soñé.

A la Escuela Politécnica del Ejército por ser un pilar fundamental en la reconstrucción de mi vida y permitirme volver a empezar en busca de la consecución de un gran objetivo trazado desde hace mucho tiempo.

Ahora que he conseguido este objetivo al que lo puedo llamar nuestro porque siempre están conmigo, no sé que tan lejos o cerca puedo llegar, lo que si sé es que siempre retribuiré su confianza y sus buenos deseos.

Con mucho cariño

Edison F. Naranjo E.

Agradecimientos

En primer lugar deseo agradecer a Dios y a la Virgen Santísima por darme la vida, salud, protección y guía para poder estudiar lo que me propuse.

A mi madre **Margoth Espín**, por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor y su infinito deseo de bienestar para mí.

A mi padre **Luis Fernando Naranjo**, por su ejemplo de perseverancia, constancia y trabajo tesonero que lo caracterizan y que me ha transmitido siempre, por el valor mostrado para salir adelante y por su amor.

A mis hermanos **Katia** y **Fernando Naranjo** por su cariño, ayuda incondicional y porque nunca dejaron de creer en mí y cuan lejos podía llegar pese a todos los momentos difíciles que tuve que atravesar para llegar a estas instancias.

A mis sobrinos **Verito** y **Fernandito** por ver en mí un ejemplo a seguir.

Al glorioso ***Colegio Militar Eloy Alfaro*** por haberme brindado una sólida formación moral y académica.

A mis maestros que con dedicación, paciencia y generosidad supieron transmitirme sus conocimientos para poder llegar a ser un buen profesional. De una manera particular al Ing. René López que me dió toda su confianza y me aseguró que podía brillar con luz propia.

Y finalmente agradezco al Dr. Gonzalo Olmedo y al Ing. Darwin Alulema por su apoyo y confianza para llegar a finalizar con éxito este proyecto de graduación.

Prólogo

El 26 de marzo de 2010, Ecuador optó finalmente por elegir el estándar ISDB-T/SBTVD (desarrollo japonés con variaciones brasileñas) para implantar la televisión digital, según ha confirmado el Consejo Nacional de Telecomunicaciones (CONATEL).

Con esta opción, en detrimento del estándar europeo DVB-T, Ecuador se suma así a Argentina, Chile, Perú, Venezuela y Brasil en adoptar la norma japonesa para la televisión digital terrestre (TDT). Autoridades ecuatorianas estiman que en un plazo de siete años quede totalmente implementada la TDT pudiéndose llevar a cabo entonces el apagado analógico.

Tanto Japón como Brasil, socios en este desarrollo latinoamericano, se han comprometido a brindar asesoramiento técnico y apoyo logístico durante este proceso. Para ello, ya se han firmado memorandos de cooperación entre Ecuador y Brasil y Ecuador y Japón, con la presencia de las máximas autoridades diplomáticas y de Telecomunicaciones de los países implicados.

Según este compromiso, Japón prestará asesoramiento técnico en tecnología digital, mientras que Brasil creará un centro de producción de contenidos digitales, y donará equipos para la televisión estatal ecuatoriana.

Actualmente la Escuela Politécnica del Ejército se constituye como representante de la Red Latinoamericana de Cooperación, Investigación, Desarrollo y Formación de Software para TV Digital Interactiva. Uno de sus principales objetivos trazados es asegurar la calidad de señal transmitida. Por este motivo, es necesario la elaboración de un analizador cuyo propósito debe estar orientado a verificar la estructura de los paquetes Transport Stream, y en base a esta información detectar que tipo de errores se están generando y posteriormente corregirlos.

Índice general

Certificación	2
Resumen	3
Dedicatoria	4
Agradecimientos	6
Prólogo	7
1. Análisis del Paquete Transport Stream	15
1.1. Introducción	15
1.1.1. Televisión Digital Terrestre	15
1.1.2. ISDB-T (Integrated Services Digital Broadcasting)	15
1.1.3. Analizador de Paquetes	16
1.1.4. Estado del Arte	17
1.2. Formación del Transport Stream	17
1.2.1. Generación de los Flujos de Señal MPEG-2	17
1.2.2. Procesado de la Señal	19
1.2.3. Múltiplex tipo "Program Stream"	27
1.2.4. Múltiplex tipo "Transport Stream"	29
1.3. Componentes de la Cabecera del Paquete Transport Stream	32
1.3.1. Transport Packet	32
1.3.2. Transport Packet Header	32
1.3.3. Campo de Adaptación	34
1.4. Información Específica de los Programas (PSI) e Información de Servicio (SI)	35
1.4.1. Información Específica de los Programas	35

1.4.2.	Información de Servicio (SI)	36
1.4.3.	Tablas MPEG-2	37
1.4.4.	Tablas Principales de ISDB-T - SI	39
1.4.5.	Tablas opcionales de ISDB-T - SI	40
1.4.6.	Códigos de PID	41
1.5.	Arquitectura de las Tablas de Información de Servicio	41
1.5.1.	Distribucion de las secciones en los paquetes de transporte	41
1.5.2.	Composicion de las secciones	42
2.	Especificación de los Requisitos de Software	45
2.1.	Introducción	45
2.1.1.	Propósito	45
2.1.2.	Ámbito del Sistema	45
2.1.3.	Definiciones, Acrónimos y Abreviaturas	46
2.1.4.	Referencias	47
2.1.5.	Visión General	48
2.2.	Descripción General	48
2.2.1.	Perspectiva del Producto	48
2.2.2.	Funciones del Producto	48
2.2.3.	Características de los Usuarios	50
2.2.4.	Restricciones	50
2.2.5.	Suposiciones y Dependencias	51
2.2.6.	Requisitos futuros	51
2.3.	Requisitos Específicos	51
2.3.1.	Interfaces	51
2.3.2.	Requisitos Funcionales	54
2.3.3.	Requisitos no funcionales	57
3.	Desarrollo del Analizador de Paquetes Transport Stream	59
3.1.	Modelado con UML	59
3.1.1.	Identificación de los actores	60
3.1.2.	Diagramas de Casos de Uso	60
3.1.3.	Diagramas de Secuencia	67
3.1.4.	Diagrama de Paquetes	71
3.1.5.	Diagramas de Clases	72

3.1.6. Diagrama de Despliegue	74
3.2. Consideraciones para el Diseño del A.P.T.S. con MATLAB	75
3.3. Diagramas de Flujo	77
3.3.1. Diagrama de Flujo para el A.P.T.S. con MATLAB	77
3.3.2. Diagrama de Flujo para el A.P.T.S. con Java	78
3.4. Pseudocódigos	79
3.4.1. Pseudocódigo para búsqueda de paquetes de transporte en MATLAB	79
3.4.2. Pseudocódigo para búsqueda de paquetes de transporte en Java .	80
4. Pruebas y Resultados	81
4.1. Pruebas de Funcionamiento del A.P.T.S. con MATLAB	81
4.1.1. Apertura de Archivos con MATLAB	81
4.1.2. Búsqueda de Paquetes de Transporte con MATLAB	83
4.2. Pruebas de Funcionamiento del A.P.T.S. con Java	85
4.2.1. Apertura de Archivos con Java	85
4.2.2. Búsqueda de Paquetes de Transporte con Java	87
4.2.3. Generación de Reporte PID con Java	89
4.2.4. Generación de Reporte de Cabecera con Java	91
4.3. Rendimiento del A.P.T.S. en la generación de reportes	93
4.3.1. Generación de Reportes con un mismo Sistema Operativo	93
4.3.2. Generación de Reportes con distintos Sistemas Operativos	95
4.4. Transición	97
4.4.1. Entrega	97
5. Conclusiones y Recomendaciones	99
5.1. Conclusiones	99
5.2. Recomendaciones	100
Apendice A	102
Apendice B	104
Apendice C	107
Apendice D	114
Bibliografía	115

Índice de figuras

1.1. Flujos de Programa y Transporte MPEG-2.	18
1.2. Estructura de la trama de audio AES/EBU	21
1.3. Compresión MPEG-2 de la señal de video digital	23
1.4. Conversión de un E.S. en un P.E.S.	24
1.5. Cabecera de un P.E.S.-Packet	25
1.6. Estructura del múltiplex MPEG-2 "Program Stream"	28
1.7. División de un P.E.S. en una serie de Transport Packets	30
1.8. Conformación del múltiplex Transport Stream	31
1.9. Transport Packet	32
1.10. Transport Packet Header	32
1.11. Campo de Adaptación	35
1.12. Representación de una tabla PAT	37
1.13. Ejemplo de una tabla PMT	39
1.14. Primer comienzo de sección dentro de un paquete TS	42
2.1. GUI para MATLAB	51
2.2. GUI para Netbeans IDE	52
3.1. Actores que intervienen en el A.P.T.S.	60
3.2. Caso de Uso contextual del A.P.T.S.	61
3.3. Relación entre los Casos de Uso del A.P.T.S.	61
3.4. Diagrama de Secuencia Abrir Archivo	68
3.5. Diagrama de Secuencia Buscar Paquete de Transporte	69
3.6. Diagrama de Secuencia Generar Reporte PID	69
3.7. Diagrama de Secuencia Generar Reporte de Cabecera	70
3.8. Diagrama de Secuencia Almacenar Datos	71
3.9. Diagrama de Paquetes del A.P.T.S.	72
3.10. Clase Principal del A.P.T.S.	72

3.11. Clases pertenecientes al Paquete Operaciones	73
3.12. Clases que representan las GUI del A.P.T.S.	74
3.13. Diagrama de Despliegue del A.P.T.S.	74
3.14. Estructura del A.P.T.S. con MATLAB	75
3.15. Diagrama de Flujo para el A.P.T.S. con MATLAB	77
3.16. Diagrama de Flujo para el A.P.T.S. con Java	78
3.17. Pseudocódigo del algoritmo de búsqueda en MATLAB	79
3.18. Pseudocódigo del algoritmo de búsqueda en Java	80
4.1. Caso de prueba apertura de archivos con MATLAB - mux1-cp.ts	82
4.2. Caso de prueba apertura de archivos con MATLAB - espetv.ts	82
4.3. Caso de prueba apertura de archivos con MATLAB - espe_sek.ts	83
4.4. Caso de prueba búsqueda de paquetes de transporte con MATLAB - mux1-cp.ts	84
4.5. Caso de prueba búsqueda de paquetes de transporte con MATLAB - es- petv.ts	84
4.6. Caso de prueba búsqueda de paquetes de transporte con MATLAB - espe_sek.ts	85
4.7. Caso de prueba apertura de archivos con Java - mux1-cp.ts	86
4.8. Caso de prueba apertura de archivos con Java - espetv.ts	86
4.9. Caso de prueba apertura de archivos con Java - espe_sek.ts	87
4.10. Caso de prueba búsqueda de paquetes de transporte con Java - mux1-cp.ts	88
4.11. Caso de prueba búsqueda de paquetes de transporte con Java - espetv.ts	88
4.12. Caso de prueba búsqueda de paquetes de transporte con Java - espe_sek.ts	89
4.13. Caso de prueba generación de reporte PID con Java - mux1-cp.ts	90
4.14. Caso de prueba generación de reporte PID con Java - espetv.ts	90
4.15. Caso de prueba generación de reporte PID con Java - espe_sek.ts	91
4.16. Caso de prueba generación de reporte de cabecera con Java - mux1-cp.ts	92
4.17. Caso de prueba generación de reporte de cabecera con Java - espetv.ts .	92
4.18. Caso de prueba generación de reporte de cabecera con Java - espe_sek.ts	93
4.19. Tiempos de ejecución para la generación de Reportes PID - MS Windows 7	94
4.20. Tiempos de ejecución para la generación de Reportes de Cabecera - MS Windows 7	95
4.21. Comparación de resultados para Reportes PID - Windows 7 vs Ubuntu 12.04	96

4.22. Comparación de resultados para Reportes de Cabecera - Windows 7 vs Ubuntu 12.04	97
----------------------------------------------------------------------------------------------------	----

Índice de cuadros

1.1. Estructura de la Cabecera de un PES-Packet	27
1.2. Estructura de la Cabecera de un Paquete TS	33
1.3. Códigos de PID para los distintos tipos de tablas	41
1.4. Valores de "table_id " para las distintas tablas SI	44
2.1. Características del Usuario "Usuario"	50
2.2. Características del Usuario "Desarrollador de Software"	50
2.3. Características de la GUI del A.P.T.S. con MATLAB	52
2.4. Características de la GUI del A.P.T.S. con Netbeans IDE	53
4.1. Datos obtenidos en el caso de prueba apertura de archivos con MATLAB	83
4.2. Datos obtenidos en el caso de prueba búsqueda de paquetes de transporte con MATLAB	85
4.3. Datos obtenidos en el caso de prueba apertura de archivos con Java . . .	87
4.4. Datos obtenidos en el caso de prueba búsqueda de paquetes de transporte con Java	89
4.5. Características de los Ordenadores para los casos de prueba	94
4.6. Detalle de los tiempos ejecución para la generación de Reportes PID - MS Windows 7	94
4.7. Detalle de los tiempos de ejecución para la generación de Reportes de Cabecera - MS Windows 7	95
4.8. Detalle de los tiempos ejecución para la generación de Reportes PID - Ubuntu 12.04	96
4.9. Detalle de los tiempos de ejecución para la generación de Reportes de Cabecera - Ubuntu 12.04	96

Capítulo 1

Análisis del Paquete Transport Stream

1.1. Introducción

1.1.1. Televisión Digital Terrestre

Es la transmisión de imágenes en movimiento y su sonido asociado mediante una señal digital a través de una red de repetidores terrestres. La codificación digital de la información brinda varias ventajas entre las que podemos destacar:

- La posibilidad de comprimir la señal con lo cual se puede hacer un uso más eficiente del espectro radioeléctrico.
- Multiplexación de la señal permitiendo así la transmisión de un mayor número de canales ocupando el mismo espacio de un canal analógico. El número de programas transmitidos en cada canal múltiple dependerá de la proporción de compresión empleada.

La compresión también ha hecho viable la emisión de señales de televisión en alta definición (HDTV), que requieren un ancho de banda mayor que la de definición estándar.

1.1.2. ISDB-T (Integrated Services Digital Broadcasting)

Es un conjunto de normas creadas por Japón para las transmisiones de radio y televisión digital. Este sistema fue diseñado para transmitir una señal móvil y fija simultáneamente

en un ancho de banda de 6 MHz.

Actualmente se constituye en el estándar de Televisión Digital más avanzado. Tiene tres tipos de sistemas:

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

Fueron desarrollados con el propósito de ofrecer flexibilidad, capacidad de expansión y difusión de los servicios de transmisión multimedia, cada uno empleando las redes respectivas y define los esquemas de codificación de canal y de modulación para su respectivo medio de transmisión, pero en todos los casos la codificación de fuente es una adaptación del estándar MPEG-2.

La señal de entrada y de salida especificada para todos los sistemas es la denominada **MPEG-2 Transport Stream (TS)** o **Flujo de Transporte MPEG-2**.

En este capítulo se describirá la estructura del citado flujo de transporte, según está definido en el estándar ISO/IEC 13818-1 [1]. Para su empleo en ISDB-T, dicha estructura debe complementarse con la denominada **Información de Servicio (SI)**.

1.1.3. Analizador de Paquetes

Denominado también *Sniffer* está diseñado para capturar paquetes de datos en una red, decodificarlos y analizarlos. Éste mostrará los resultados del análisis a un administrador de red para tomar acciones futuras, o bien puede ser programado para accionar las reparaciones a las interrupciones de tráfico de una red.

Un paquete de datos representa cada uno de los bloques en que se divide la información a ser enviada. Está generalmente compuesto de tres elementos:

- Una cabecera (*header*) que contiene generalmente la información necesaria para enviar el paquete desde el emisor hasta el receptor.
- El área de datos (*payload*) que contiene los datos.

- La cola (*trailer*), que comúnmente incluye código de detección de errores.

1.1.4. Estado del Arte

El Instituto Nacional de Investigación y Capacitación de Telecomunicaciones del Perú desarrolló anteriormente un Analizador de Cabecera de Transport Stream basado en el lenguaje de Programación Visual Basic orientado para trabajar en plataformas bajo el Sistema Operativo Windows.

El principal limitante de este software es que no es multiplataforma, de ahí surge la necesidad de desarrollar una aplicación con similares características para operar en múltiples plataformas, en especial con herramientas Open Source.

1.2. Formación del Transport Stream

1.2.1. Generación de los Flujos de Señal MPEG-2

En la figura 1.1 se muestra de manera esquemática la secuencia de operaciones que da lugar a los flujos de programa y transporte MPEG-2. Dicha secuencia de operaciones puede dividirse en dos grandes bloques denominados *Capa de Compresión* y *Capa de Sistema*.

En la **Capa de Compresión** se realizan las operaciones propiamente dichas de codificación MPEG, recurriendo a los procedimientos generales de compresión de datos, y aprovechando además para las imágenes su redundancia espacial (áreas uniformes) y temporal (imágenes sucesivas), la correlación entre puntos cercanos y la menor sensibilidad del ojo a los detalles finos de las imágenes fijas.

En cuanto al audio se utilizan modelos psicoacústicos del oído humano, que tienen en cuenta la curva de sensibilidad en frecuencia (máxima entre 1 y 5 kHz), los efectos de enmascaramiento frecuencial (señales simultáneas a diferentes frecuencias) y enmascaramiento temporal (un sonido de elevada amplitud enmascara sonidos más débiles anteriores o posteriores), para reducir la cantidad de datos que hay que transmitir, sin deteriorar de forma perceptible la calidad de la señal de audio.

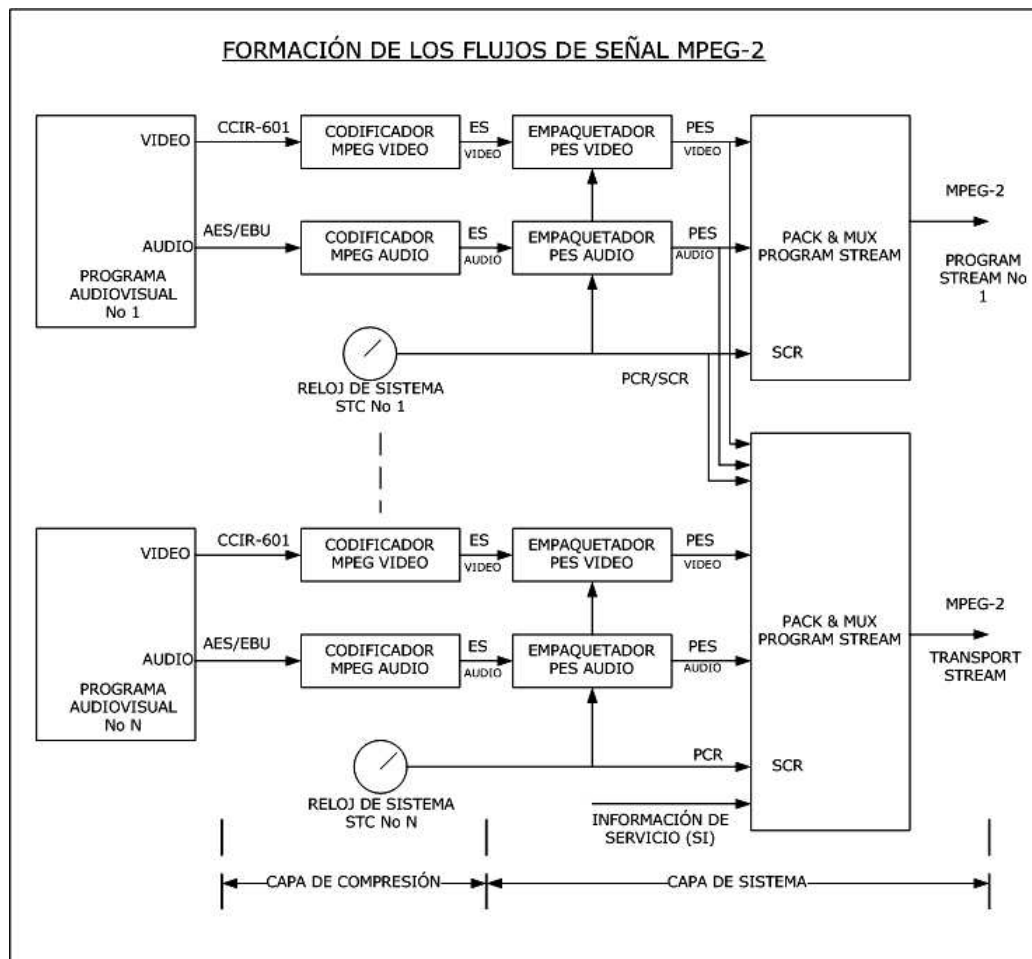


Fig. 1.1: Flujos de Programa y Transporte MPEG-2.

P.A. Sarginson, B.Sc. MPEG-2: Overview of the System Layer. Report No. 1996/2. 1.996.

En la **Capa de Sistema** se realizan las operaciones que conducen a la obtención de los flujos de señal MPEG-2, consistentes en la organización en "paquetes", de los datos comprimidos y el posterior multiplexado de todas las señales asociadas al programa (video, audio, datos, etc.).

En el caso de los Flujos de Programa MPEG-2, se multiplexan todos los componentes del programa (video, audio, datos, etc.) y se incorpora el reloj del sistema, pero solamente se transmite la información correspondiente a un único programa audiovisual.

En cambio, en el caso de los Flujos de Transporte (Transport Stream) MPEG-2, además de la posibilidad de multiplexado de varios programas audiovisuales, se añaden en el múltiplex diversas informaciones relativas al servicio: Tabla de Asociación de Programas

(PAT), Información para acceso condicional (CAT), Mapa de cada Programa (PMT), Tabla de datos de la Red (NIT), etc., según se verá posteriormente.

1.2.2. Procesado de la Señal

Como se ve en el diagrama de la figura 1.1, se parte de un programa digitalizado sin comprimir, que se compone de varias informaciones simultáneas. Cada una de éstas informaciones digitales, todavía sin comprimir está agrupada en bloques elementales denominados **Unidades de Presentación** (*Presentation Units*), que pueden ser de video o audio.

1. Unidades de Presentación de Video

En cuanto a la señal de video digitalizada sin comprimir, se emplea el formato 4:2:2 con cuantificación de 8 bits recogido en CCIR-601 [2], cuya Unidad de Presentación es una **Imágen o Cuadro** (830 kB para sistemas de 625 líneas).

De acuerdo con este estándar se digitalizan las señales Y (luminancia), C_R y C_B (obtenidas a partir de las señales diferencia de color), utilizando la siguientes frecuencias de muestreo:

$$\begin{aligned} \text{Señal Y:} & \quad 13,5 \text{ MHz} \\ \text{Señales } C_R \text{ y } C_B & : 6,75 \text{ MHz} \end{aligned}$$

En estas condiciones, el **flujo bruto resultante**, para la citada cuantificación de 8 bits resulta ser:

$$(13.5 \times 8) + (2(6.75 \times 8)) = \mathbf{216 \text{ Mbps}}$$

A cada línea completa, cuya duración es de $64 \mu\text{s}$ en un sistema de 625 líneas, le corresponderá el siguiente número de muestras:

$$\begin{aligned} \text{Señal Y:} & \quad 13,5(10^6)(64)(10^{-6}) = 864 \text{ muestras/línea completa} \\ \text{Señales } C_R \text{ y } C_B & : 6,75(10^6)(64)(10^{-6}) = 432 \text{ muestras/línea completa} \end{aligned}$$

Sin embargo, el cálculo anterior incluye los sincronismos en la duración de la línea. Si se eliminase la digitalización de los intervalos de supresión cuya duración es de $12 \mu s$ quedarían $52 \mu s$ útiles, con lo que resultaría:

$$\text{Señal Y: } 864(52/64) = 702 \text{ muestras video/línea completa}$$

En realidad se emplean por línea activa digital:

$$\begin{aligned} \text{Señal Y:} & \quad 720 \text{ muestras/línea completa} \\ \text{Señales } C_R \text{ y } C_B : & \quad 360 \text{ muestras/línea completa} \end{aligned}$$

Con todo ello, considerando que del total de 625 líneas, solamente 576 son útiles, el **flujo neto resultante** para la señal digitalizada en formato 4:2:2 con cuantificación de 8 bits es:

$$216 \text{ Mbps } (720/864)(576/625) = 165,888 \equiv \mathbf{166 \text{ Mbps}}$$

En esta situación, teniendo en cuenta que se transmiten 25 imágenes por segundo y que hay 8 bits por cada byte, resultará para cada **Imágen o Cuadro**:

$$166 \text{ Mbps} / (25 \times 8) = \mathbf{830 \text{ kB/cuadro}}$$

2. Unidades de Presentación de Audio

En cuanto al audio, la Unidad de Presentación es una **Trama de Audio AES/EBU** [3]. En este caso se requieren de 16 a 24 bits para proveer el rango dinámico y la relación señal / ruido deseados.

El formato de los datos de audio sin comprimir puede verse en la figura 1.2 para una cuantificación de 20 bits.

Consiste en secuencias de bloques de 192 tramas, cada una de las cuales está compuesta por dos subtramas correspondientes a los dos canales de audio estéreo.

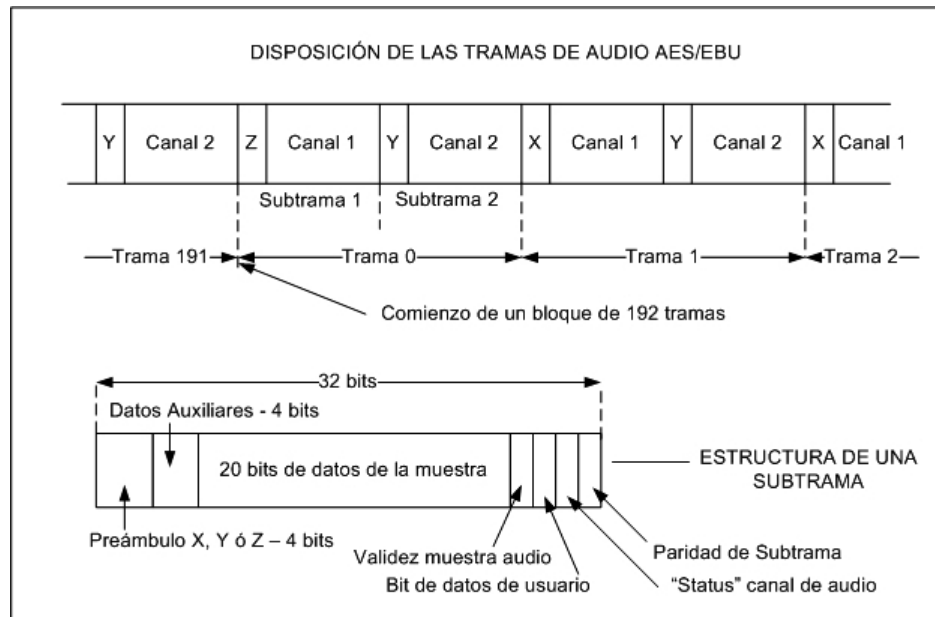


Fig. 1.2: Estructura de la trama de audio AES/EBU

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 6

A su vez, cada subtrama contiene los 20 bits de datos correspondientes a las muestras, precedidos por otros 4 bits de datos auxiliares, que pueden emplearse para extender la cuantificación a 24 bits. La subtrama comienza con un preámbulo que indica a que canal pertenece la muestra y termina con 4 bits que aportan información sobre el canal incluyendo un bit de paridad. En total cada subtrama tiene 32 bits.

El preámbulo de cada subtrama puede ser de tipo X, Y ó Z. Se emplea el tipo Z para indicar el comienzo de cada bloque de 192 tramas y después alternativamente los tipos X e Y para identificar las subtramas de los dos canales de audio: X para el canal 1 e Y para el canal 2.

a) **Codificación MPEG**

El proceso continua en la denominada *Capa de Compresión*. Es aquí donde se reduce la velocidad binaria del flujo de señal mediante la sustitución de las Unidades de Presentación por Unidades de Acceso (*Access Units*). En el caso de la señal de video, las Unidades de Acceso son de 3 tipos, correspondiendo a otros tantos tipos de imágenes MPEG:

- **Imágenes tipo I (Intra).**- Se codifican sin ninguna referencia a otras imágenes, es decir, contiene todos los elementos necesarios para su reconstrucción.
- **Imágenes tipo P (Previstas).**- Se codifican con respecto a la imagen tipo I o de otra P anterior, gracias a las técnicas de predicción con compensación de movimiento. Su tasa de compresión es claramente mayor que de las imágenes I.
- **Imágenes tipo B (Bidireccionales).**- Se codifican por interpolación entre las dos imágenes de tipo I ó P precedente y siguiente que las enmarcan. Ofrecen la tasa de compresión más alta.

En el caso de la señal de audio, las *Unidades de Acceso* típicamente contienen unas pocas decenas de milisegundos de audio comprimido. El resultado de la codificación MPEG de una secuencia de video, es una sucesión de *Unidades de Acceso de Video*. Dicha sucesión de unidades de acceso constituye el denominado ***Video Elementary Stream (V.E.S.)***.

Paralelamente, el resultado de la codificación MPEG para el audio es una sucesión de *Unidades de Acceso de Audio*, que componen el denominado ***Audio Elementary Stream (A.E.S.)***.

En la figura 1.3 se ilustra la operación asociada a la ***Capa de Compresión***.

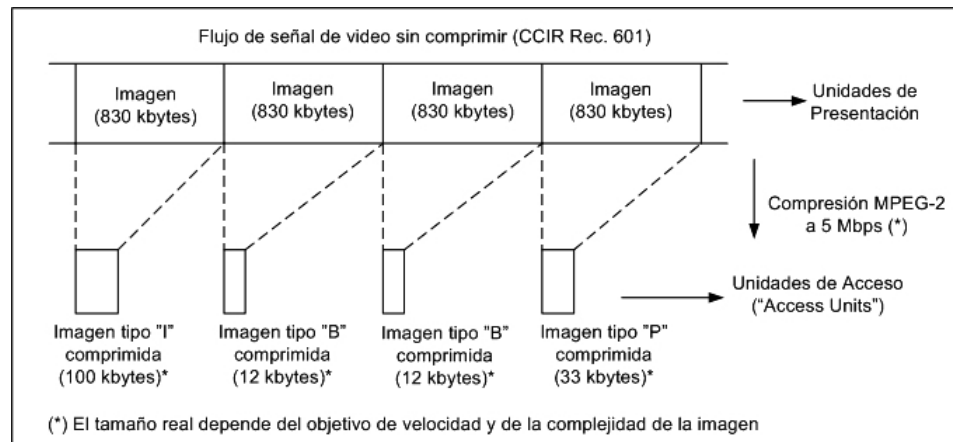


Fig. 1.3: Compresión MPEG-2 de la señal de video digital
Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 7

b) Empaquetado P.E.S.

Ya en la *Capa de Sistema* nos encontramos con los *Empaquetadores P.E.S.*, que constituyen el siguiente paso en la generación tanto del múltiplex de programa como del múltiplex de transporte MPEG-2.

Se trata de convertir cada *Elementary Stream* compuesto exclusivamente por *Unidades de Acceso* en un *Packetised Elementary Stream (P.E.S.)*.

Un P.E.S. está compuesto íntegramente por PES-Packets, como se muestra en la figura 1.4.

Un "P.E.S.-Packet" se compone de una "Cabecera(Header)" y de una "Carga Útil (Payload)".

El "*Payload*" consiste simplemente en bytes de datos tomados secuencialmente desde el "*Elementary Stream (E.S.)*" original. No hay ningún requerimiento para alinear el comienzo de una "Access Unit" y el comienzo de un "PES-Packet Payload". Así, una nueva "Access Unit" puede comenzar en cualquier punto del "Payload" de un "PES-Packet", y también es posible que varias pequeñas "Access Units" estén contenidas en un simple "PES-Packet".

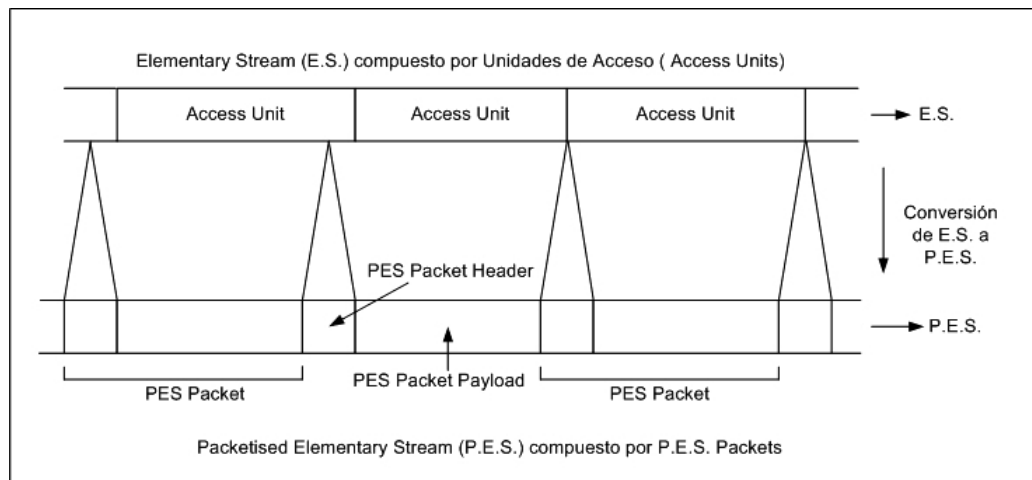


Fig. 1.4: Conversión de un E.S. en un P.E.S.

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 9

Los *"PES-Packets"* pueden ser de longitud variable, limitados normalmente a 64 kBytes, aunque la excepción a esta norma se da precisamente en los *"Video-PES"*, cuando se emplean en un *"Transport Stream"*, donde pueden tener longitud indefinida.

Cabecera del PES-Packet

La figura 1.5 muestra los campos que conforman la Cabecera de un PES-Packet. Dicha cabecera tiene longitud variable, e incorpora a su vez datos para autoinformar de su longitud.

■ PES-Packet Start Code

Los primeros 4 bytes, es decir: el *" PES-Packet start code prefix"* más el *"Stream id"* constituyen el *"PES-Packet start code"*. Debe garantizarse que esta combinación de 32 bits únicamente aparece en el comienzo de la cabecera de un PES-Packet, excepción hecha de los datos de tipo "privado".

El byte *"Stream_id"* permite distinguir los PES-Packets pertenecientes a los diferentes Elementary Stream de un mismo Programa. MPEG especifica los valores permitidos para este campo, disponiéndose de 32 valores para E.S. de audio y otros 16 valores para E.S. de video.

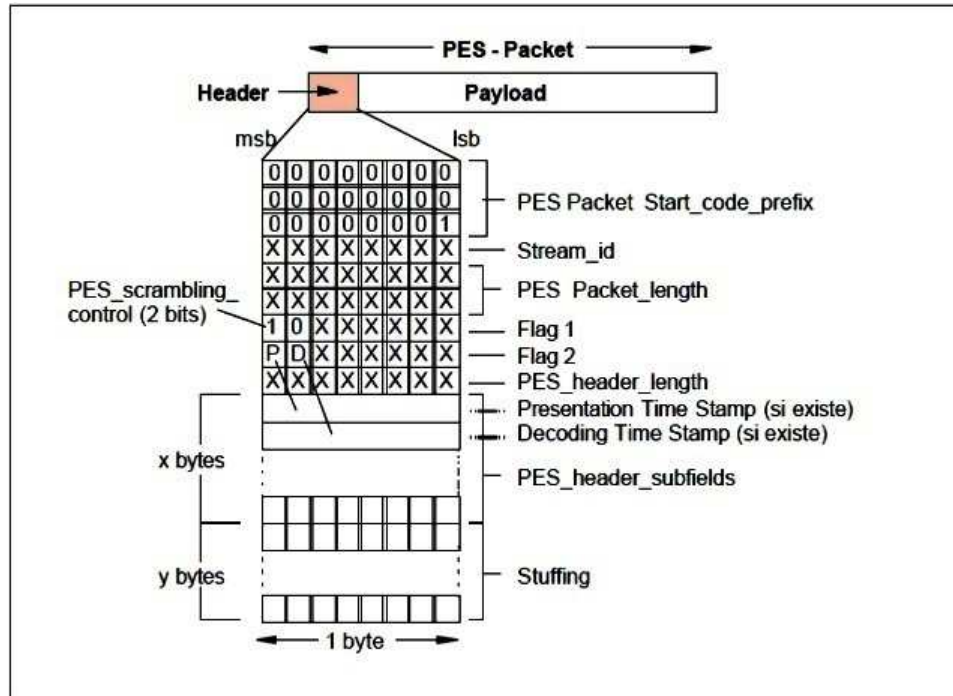


Fig. 1.5: Cabecera de un P.E.S.-Packet

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 10

■ Flags 1 y 2

Los bytes **"Flag 1"** y **"Flag 2"** son indicadores que muestran la presencia o ausencia de varios campos opcionales que pueden estar incluidos en la cabecera de un PES-Packet. Estos campos opcionales llevan información complementaria relativa al P.E.S. tales como: si está cifrado o no, prioridad relativa, datos de *"copyright"*, un campo para identificación de errores en el paquete, etc.

Particularmente importantes son los 2 bits más significativos de la **"Flag 2"**, marcados como P y D en la figura 1.5. Cuando están, estos bits indican respectivamente la presencia de un campo denominado **"Presentation Time Stamp (PTS)"** y de otro campo denominado **"Decoding Time Stamp (DTS)"** dentro de la cabecera del PES-Packet.

■ Time Stamps

Los **"Time Stamps"** o **"marcas de tiempo"** son el mecanismo proporcionado por la Capa de Sistema de MPEG-2 para asegurar una correc-

ta sincronización en el decodificador entre Elementary Streams que están relacionados. Dichos "Time Stamps" son valores binarios de 33 bits expresados en unidades de 90 kHz.

El más importante de los Time Stamps es el "**Presentation Time Stamp**". Especifica el momento en que una Unidad de Acceso debe retirarse del buffer del decodificador, ser decodificada y seguidamente presentada al espectador. MPEG supone que todo esto puede realizarse instantáneamente, aunque como en la práctica lleva algún tiempo, es responsabilidad del diseñador del decodificador resolver este problema.

En muchos tipos de "Elementary Streams" es suficiente con los PTS, pero en el caso de Elementary Streams compuestos por video codificado MPEG, puede necesitarse un segundo tipo de Time Stamp denominado "**Decoding Time Stamp(DTS)**".

Un "DTS" especifica el momento en que una Unidad de Acceso debe retirarse del buffer del decodificador y ser decodificada, pero no presentada al espectador. En lugar de esto último, la imagen decodificada se almacena temporalmente para ser presentada más tarde. Tal tratamiento es necesario únicamente para imágenes de tipo I y P de una secuencia de video codificada MPEG; estas imágenes I y P son las que por interpolación permiten generar las imágenes de tipo B (Ver Codificación MPEG página 21).

Un "DTS" nunca aparece aislado, sino que debe ir acompañado de un "PTS", que es el que informa del momento en que la Unidad de Acceso almacenada temporalmente debe presentarse al espectador. De esta forma, el PTS siempre será mayor que su DTS asociado (es decir, posterior en el tiempo), puesto que la presentación de la imagen al espectador ocurrirá más tarde.

No es necesario ubicar "Time Stamps" para cada unidad de acceso. El decodificador normalmente conoce de antemano el ritmo al que las unidades de acceso tienen que ser decodificadas, siendo suficiente proporcionar ocasionalmente los "Time Stamps" simplemente para asegurar el sincronismo

del proceso de decodificación a largo plazo.

El requerimiento especificado en MPEG es que un *"Time Stamp"* debe aparecer al menos cada 0,7 segundos en los PES de video o audio. Por otra parte, si una Unidad de Acceso tiene un *"Time Stamp"* asociado con ella, entonces este tiene que estar codificado en la cabecera del PES-packet en que dicha unidad de acceso comienza.

■ PES header length

El *"PES header data length field"* es el último de los bytes obligatorios de una cabecera de un PES-packet. Su valor indica el número de bytes opcionales presentes en la cabecera, hasta que se alcanza el primer byte del *"PES-packet payload"*.

Hay 25 campos opcionales en una cabecera, que pueden contener por tanto un total de 200 bits de datos adicionales.

En la tabla 1.1 pueden verse algunos detalles complementarios relativos a los campos de la cabecera de un paquete PES

Campo	Definición	Nº de bits
Start_code_prefix	Código de inicio (0X00,0X01)	24
Stream_id	Identificación del P.E.S.	8
Packet_length	Longitud del P.E.S. Packet	16
PES_scrambling_control	Indica si hay cifrado, y su código	2
Flags	Marcadores	14
PES_header_length	Longitud restante de la cabecera (x+y)	8
PES_header_subfields	Campo variable, función de los flags	x bytes
Stuffing	Relleno	y bytes

Tabla 1.1: Estructura de la Cabecera de un PES-Packet

1.2.3. Múltiplex tipo "Program Stream"

El tipo de múltiplex denominado *"Program Stream"* o *"Flujo de Programa"* se crea a partir de uno o varios PES pertenecientes al mismo Programa Audiovisual, que

deben compartir obligatoriamente el mismo reloj de referencia.

Este "Program Stream" está compuesto por "**packs**", que a su vez comprenden: una cabecera del pack o "**pack-header**", opcionalmente una cabecera de sistema o "**system header**", y un número indeterminado de PES-packets, tomados de los Elementary Streams que componen el Programa Audiovisual, en cualquier orden. La figura 1.6 muestra la estructura de este múltiplex.

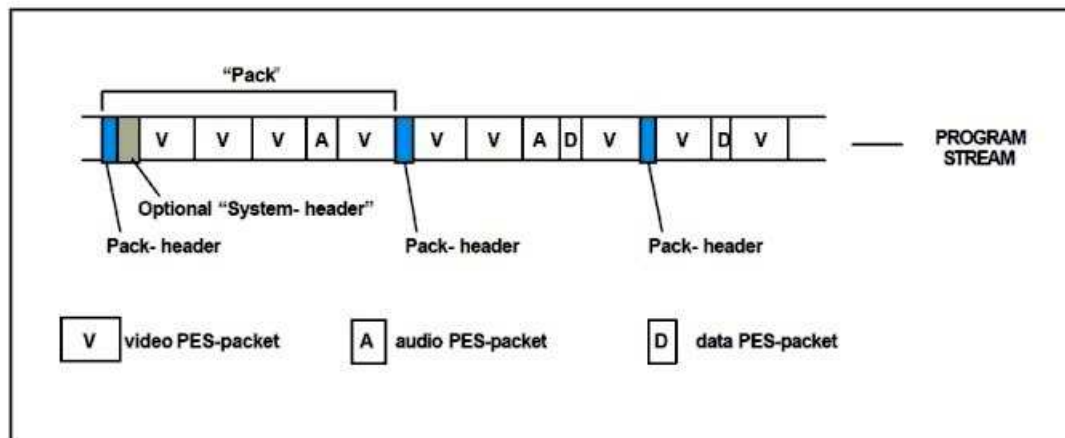


Fig. 1.6: Estructura del múltiplex MPEG-2 "Program Stream"

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 13

No existe una especificación acerca de la longitud que debe tener cada "**pack**" de un "**Program Stream**". Únicamente debe aparecer un "**pack-header**" al menos cada 0,7 segundos, puesto que la citada cabecera contiene importante información para la sincronización: la denominada "**Referencia del Reloj de Sistema**" o "**System Clock Reference (SCR)**".

En efecto: en un "**Program Stream**" que puede contener únicamente un programa audiovisual, el reloj en el multiplexor se llama: "**System Clock**". Todas las Unidades de Acceso de los "**Elementary Streams**" del programa tienen asignados "**Time Stamps**" basados en este System Clock, que están contenidas en la cabecera de los PES-packets.

A su vez, muestras del "**System Clock**", las denominadas "**System Clock References (SCR)**" se encuentran codificadas en los campos opcionales de los "**pack-headers**" del "**Program Stream**". Son números binarios de 42 bits que expresan unidades de 27MHz.

En el campo opcional *"system-header"* de un pack se incluye un resumen de las características del *"Program Stream"* tal como: su velocidad binaria máxima, el número de *"Elementary Streams"* de video y de audio que lo componen, información complementaria de temporización, etc. Así, un decodificador puede usar la información contenida en este *"system header"* para establecer si es capaz de recuperar la información del *"Program Stream"* o no.

El *"Program Stream"* está concebido para su empleo en entornos libres de errores, puesto que es bastante vulnerable a ellos. Hay dos razones para esto: en primer lugar, el *"Program Stream"* comprende una sucesión de paquetes relativamente largos y de longitud variable. Como se ha visto, cada paquete comienza con unas cabeceras que contienen información importante para su recuperación, por lo que cualquier error en dichas cabeceras puede provocar la pérdida entera del paquete. A su vez, como el paquete de un *"Program Stream"* puede contener muchos kilobytes de datos, la pérdida de un solo paquete puede representar la pérdida o corrupción de una trama de video completa.

En segundo lugar, la diferente longitud de los paquetes implica que un decodificador no puede predecir donde termina un paquete y comienza otro, por lo que se ve obligado a leer e interpretar el campo incluido en la cabecera que informa de la longitud del paquete. Si este campo que identifica la longitud del paquete se corrompe por un error, el decodificador perderá el sincronismo con el flujo de datos, resultando como mínimo la pérdida de un paquete.

1.2.4. Múltiplex tipo "Transport Stream"

El múltiplex tipo *"Transport Stream"* está compuesto íntegramente por *"paquetes de transporte"* o *"transport packets"* que tienen siempre una longitud fija de 188 bytes. Cada *"paquete de transporte"* incluye una *"Cabecera"* o *"Header"* (4 bytes) seguida a veces de un *"Campo de Adaptación"* o *"Adaptation Field"* (usado eventualmente para rellenar el exceso de espacio disponible) y en cualquier caso, de una *"Carga Útil"* o *"Payload"*.

Los paquetes de transporte se forman a partir de los *"PES-Packets"* correspondientes a cada *"Flujo Elemental"* de señal (video, audio, datos, etc.), según se muestra en la figura 1.7.

El proceso de formación de los paquetes de transporte está sujeto al cumplimiento de

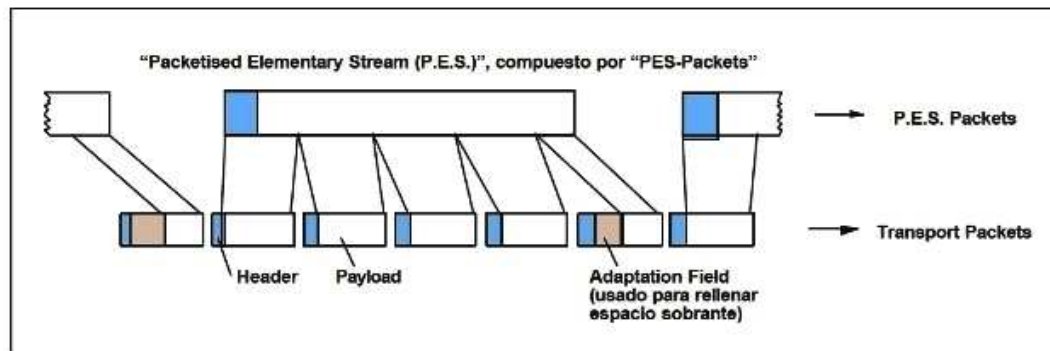


Fig. 1.7: División de un P.E.S. en una serie de Transport Packets
 Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 15

las dos condiciones fundamentales siguientes:

1. El primer byte de cada PES-Packet debe ser el primer byte del *payload* de un transport packet.
2. Un transport packet solamente puede contener datos tomados de un PES-Packet.

Es improbable que un PES-packet rellene las "cargas útiles" de un número entero de paquetes de transporte de forma exacta. Como se muestra en la figura 1.7, con frecuencia se dará el caso de que, para no contravenir las dos condiciones anteriores, el espacio sobrante del último paquete de transporte correspondiente a un PES-packet, se rellene deliberadamente mediante un "campo de adaptación" de longitud apropiada.

Este despilfarro de espacio puede minimizarse mediante una elección cuidadosa de la longitud de los PES-packets. En principio, PES-packets muy largos asegurarían una mayor proporción de paquetes de transporte completamente llenos de datos útiles.

Además de la función de relleno, el Campo de Adaptación se utiliza también para la transmisión del "**Reloj de Referencia del Programa**" o "**Program Clock Reference**" (**PCR**). Estas marcas de tiempo, que permiten sincronizar el reloj del decodificador con el *Program Clock* del Programa al que pertenecen los paquetes de transporte, deben aparecer en el Transport Stream al menos una vez cada 0,1 segundos.

a) **Formación del Transport Stream**

Los paquetes de transporte resultantes del proceso anterior, aplicado a cada uno de los flujos elementales de señal (video, audio, datos, etc.) pertenecientes

a su vez a varios programas audiovisuales, se disponen secuencialmente para configurar el *"Flujo de Transporte MPEG-2"* o *"MPEG-2 Transport Stream (TS)"*.

No existen condiciones en cuanto al orden en que los paquetes de transporte deben aparecer en el múltiplex tipo TS; tan sólo debe respetarse el orden cronológico de los paquetes de transporte pertenecientes a un mismo flujo elemental. Es importante destacar que en el TS, además de los paquetes de transporte asociados a los flujos elementales de señal, es necesario incorporar paquetes de transporte que contienen información sobre el servicio, así como paquetes de transporte *"nulos"* que se emplean para absorber eventuales reservas de capacidad del múltiplex.

La figura 1.8 ilustra de forma simplificada el proceso de conformación del múltiplex "MPEG-2 Transport Stream".

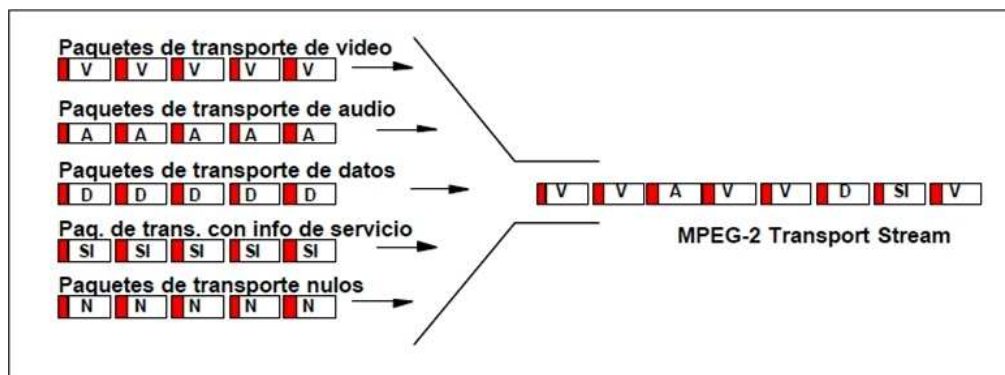


Fig. 1.8: Conformación del múltiplex Transport Stream

1.3. Componentes de la Cabecera del Paquete Transport Stream

1.3.1. Transport Packet

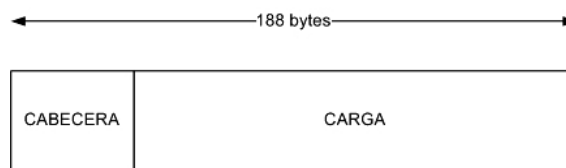


Fig. 1.9: Transport Packet

1.3.2. Transport Packet Header

A diferencia de los "PES-Packet", cuya cabecera tiene longitud variable, la cabecera o "header" de un paquete de transporte tiene siempre una longitud fija de 4 bytes. El formato de esta cabecera puede verse en la figura 1.10 y la descripción de sus componentes la tabla 1.2

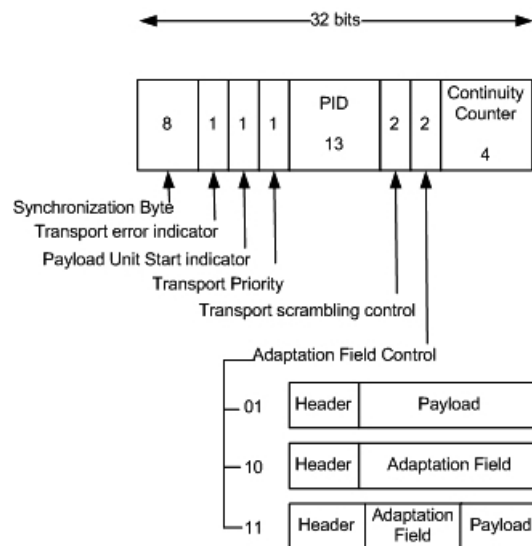


Fig. 1.10: Transport Packet Header

Campo	Definición	Nº de bits
sync_byte	Byte de Sincronización 0100 0111 (0X47)	8
transport_error_indicator	Identifica un error detectado más atrás	1
payload_unit_start_indicator	Inicio de PES en el paquete de transporte	1
transport_priority	Indicador de prioridad	1
Flags	Marcadores	14
PID	Identificador del paquete de transporte	13
transport_scrambling_control	Tipo de cifrado de transporte	2
adaptation_field_control	Control del Campo de Adaptación en el paquete	2
continuity_counter	Contador de continuidad entre paquetes afines	4

Tabla 1.2: Estructura de la Cabecera de un Paquete TS

■ Sync Byte

Es el primer byte de la cabecera y siempre tiene el valor 47 (hexadecimal). Dicho valor no es único dentro de un paquete de transporte y puede ser bastante frecuente que aparezca en otros campos. Sin embargo, el hecho de que un *"sync byte"* aparezca siempre cada 188 bytes dentro de un transport stream hace fácil el proceso de su localización por los decodificadores para identificar el comienzo de cada paquete de transporte.

■ Transport Error Indicator

Si está en "1" indica que hay, al menos, un bit erróneo no corregido dentro del paquete.

■ Payload Unit Start Indicator

Se pone a "1" para indicar que el primer byte del *"Payload"* del paquete de transporte es también el primer byte de un PES-packet.

■ Transport Priority

Si está en "1" indica que el paquete asociado de transporte tiene más prioridad que otros paquetes con el mismo PID.

■ Packet Identifier (PID)

Como se ha explicado, un TS puede contener muchos programas diferentes, cada uno de los cuales contiene a su vez muchos flujos elementales distribuidos en

paquetes de transporte. El campo de 13 bits "***Packet Identifier (PID)***" se emplea para distinguir los paquetes de transporte asociados a un determinado flujo elemental, de entre todos los demás.

De los 2^{13} valores posibles para el PID, 17 están reservados para usos especiales. Esto deja 8.175 posibles valores que pueden asignarse a los diferentes flujos elementales, por lo que esta cifra representa el máximo número de éstos que un TS puede contener. Es responsabilidad del multiplexor asegurar que cada flujo elemental tenga asignado un exclusivo valor de PID de entre todos los posibles. Por lo demás, MPEG no condiciona la asignación de valores concretos a los diferentes flujos elementales.

- **Transport Scrambling Control**

Control de codificación de transporte, debe ser obligatoriamente un campo que identifica el modo de codificación del payload para el paquete TS.

- **Adaptation Field Control**

Indica si la cabecera va seguida por un campo de adaptación y/o carga útil.

- **Continuity Counter**

Se incrementa entre sucesivos paquetes de transporte pertenecientes al mismo flujo elemental. Esto permite al decodificador detectar la pérdida o ganancia de un paquete y así poder ocultar los errores que de otra forma podrían presentarse.

1.3.3. Campo de Adaptación

Los campos más destacables dentro del Campo de Adaptación de una cabecera TS son los siguientes:

- **Longitud del campo de adaptación:** Indica la longitud de la cabecera extra.
- **Indicador de discontinuidad:** Está en el PCR y en el contador de continuidad. Se utiliza para evitar pérdidas de información producidas por un salto en el codificador.
- **PCR (*Program Clock Reference*):** El PCR es una información de sincronización del reloj de 27 MHz del receptor necesaria para la decodificación del video, audio y datos. Se incluye periódicamente en los paquetes de transporte. El receptor necesita esta información a una cadencia de unas 10 veces por segundo para hacer

funcionar el bucle de fase de su oscilador local. Síncrono y en fase con el reloj PCR de 27 MHz se dispone de otro reloj de 90 KHz que se necesita en el sistema para sincronizar otras funciones.

- **Bytes comodines:** Son bytes de relleno para conseguir una trama de 188 bytes en el supuesto de que no hubiera información suficiente para llenar el paquete.
- **Cuenta atrás para corte:** Indicador que permite una conmutación de paquetes limpia entre un TS y otro TS.

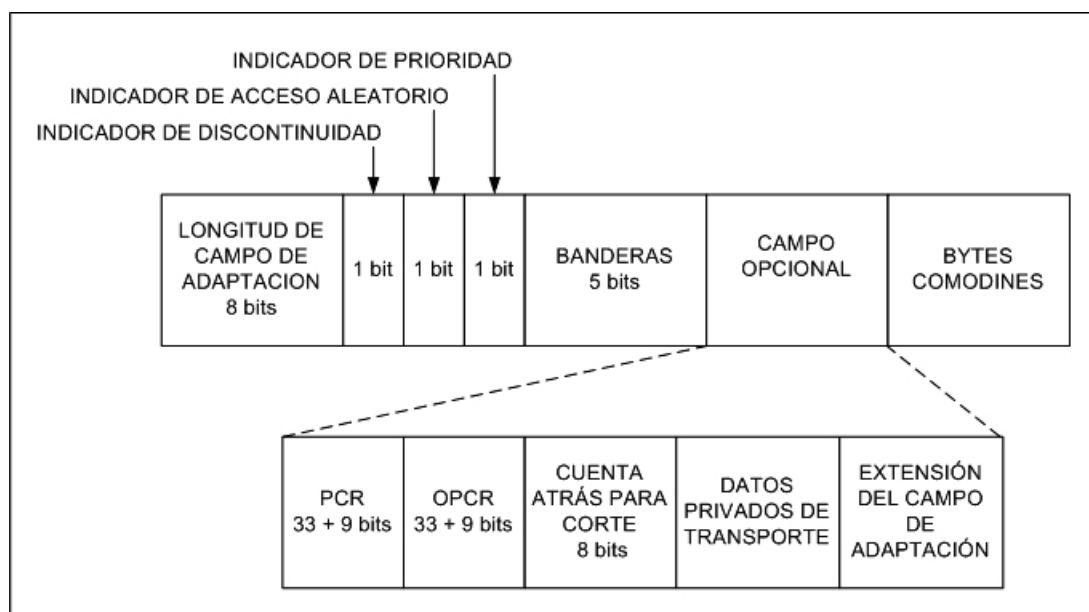


Fig. 1.11: Campo de Adaptación

1.4. Información Específica de los Programas (PSI) e Información de Servicio (SI)

1.4.1. Información Específica de los Programas

Según se ha visto, el Múltiplex MPEG-2 Transport Stream puede contener varios programas audiovisuales, cada uno de los cuales está compuesto por uno o varios flujos elementales PES distribuidos en paquetes de transporte. Estos paquetes a su vez están marcados con un PID que identifica a que flujo elemental pertenecen.

Sin embargo, para que el decodificador pueda recuperar completamente un programa a través de los valores de los PID de los paquetes correspondientes, es necesario incluir información adicional dentro del flujo de transporte que relacione estos PID con los programas a que pertenecen. Tal información se denomina ***”Información Específica de los Programas”*** o ***Program Specific Information (PSI)***. Esta ***”Información Específica de los Programas (PSI)”***, definida por MPEG-2 para la Capa de Sistema (ISO/IEC 13818-1), comprende la inclusión dentro del flujo de transporte, de 4 tipos de tablas:

- Program Association Table (PAT)
- Conditional Access Table (CAT)
- Program Map Table (PMT)
- Private

1.4.2. Información de Servicio (SI)

La ***”Información de Servicio”*** o ***”Service Information” (SI)***, en sistemas ISDB-T incluye, además, otros 4 tipos de tablas de inserción obligatoria dentro del Transport Stream y 4 tipos de tablas opcionales.

Tablas Obligatorias:

- Network Information Table (NIT)
- Service Description Table (SDT)
- Event Information Table (EIT)
- Time Date Table (TDT)

Tablas Opcionales:

- Bouquet Association Table (BAT)
- Running Status Table (RST)
- Time Offset Table (TOT)
- Stuffing Tables (ST)

Cada tabla está constituida, según su importancia, por una o varias secciones (256 como máximo, con una longitud máxima de 1.024 bytes excepto para tablas tipo "Private" y "EIT" que pueden alcanzar los 4.096 bytes). Dichas secciones están distribuidas a lo largo de una serie de paquetes de transporte identificados con un PID común.

1.4.3. Tablas MPEG-2

1. Program Association Table (PAT)

Esta tabla, de inclusión obligatoria, es transportada por los paquetes con PID=0x0000 y contiene una lista completa de todos los programas disponibles en el Transport Stream. Cada programa aparece junto con el valor del PID de los paquetes que a su vez contienen la tabla con los datos que identifican a dicho programa (Program Map Table, PMT).

La PAT debe transmitirse sin cifrar aunque todos los demás programas lo estén.

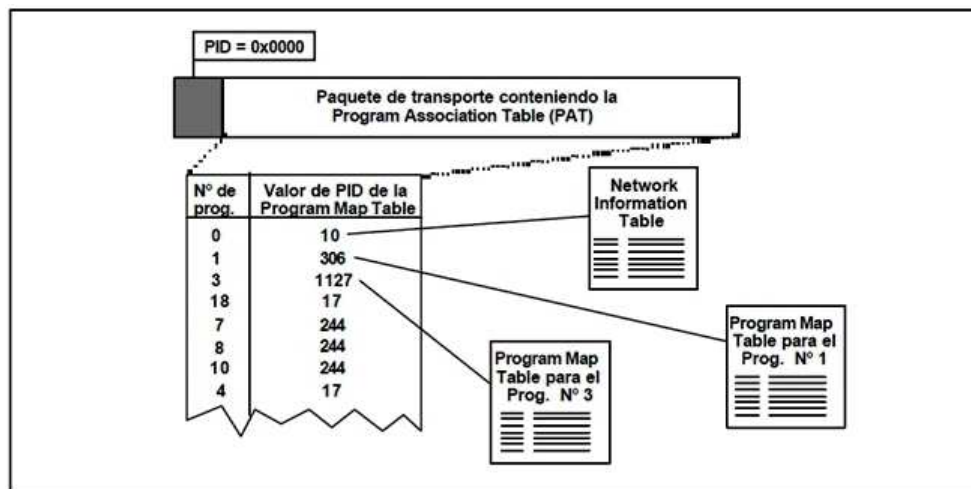


Fig. 1.12: Representación de una tabla PAT

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 20

Como se muestra a modo de ejemplo en la figura 1.12, una sola Program Map Table (la nº 244) puede contener los detalles de varios programas (los numerados como 7, 8 y 10).

2. Conditional Access Table (CAT)

Esta tabla debe estar presente si al menos un programa del múltiplex es de acceso condicional. Se transporta por los paquetes con PID=0x0001, y proporciona de-

talles de los sistemas de cifrado empleados, así como los valores de los PID de los paquetes de transporte que contienen la información del control de acceso condicional.

Los datos para el acceso condicional se envían en forma de "**Entitlement Management Messages (EMM)**". En estos "EMM" se especifican los niveles de autorización o los servicios a que pueden acceder determinados decodificadores, y pueden ir dirigidos a decodificadores individuales o a grupos de ellos. El formato de esta información no está especificado en MPEG-2, puesto que depende del tipo de sistema de cifrado empleado.

3. Program Map Table (PMT)

Cada programa audiovisual incluido en un Transport Stream tiene una tabla PMT asociada con él. Dicha tabla proporciona detalles acerca del programa y de los flujos elementales que comprende. Según MPEG-2, las tablas PMT pueden ser transportadas por paquetes con valores de PID arbitrarios, exceptuando los valores 0x0000, reservado para PAT, y 0x0001 reservado para CAT. Sin embargo, las especificaciones ISDB-T -SI también restringen el uso de los valores de PID que van desde 0x0002 hasta 0x001F (ver tabla nº 1.3).

En la figura 1.13 se muestra un ejemplo de tabla PMT. Mediante ella, un decodificador puede determinar que el flujo elemental codificado de video correspondiente al programa nº3 se encuentra en los paquetes de transporte identificados mediante el PID=726 y que el flujo elemental cuyos paquetes están etiquetados con PID=57 lleva el audio correspondiente en inglés.

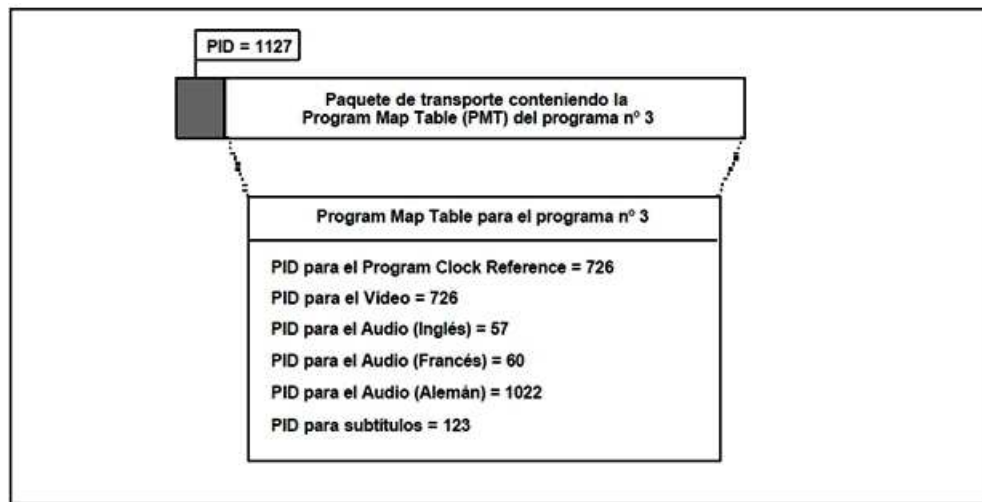


Fig. 1.13: Ejemplo de una tabla PMT

Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 21

1.4.4. Tablas Principales de ISDB-T - SI

1. Network Information Table (NIT)

Esta tabla, si está presente, constituye por definición el Programa N° 0 del múltiplex y está considerada como de datos privados, es decir, definidos por el radiodifusor (proveedor de servicios) y no por MPEG. Se transporta por los paquetes identificados con PID=0x0010.

Esta tabla proporciona información acerca de la red física usada para transmitir el "Transport Stream", como por ejemplo: frecuencias del canal, detalles del transpondedor del satélite, características de modulación, detalles de redes alternativas disponibles, etc.

2. Service Description Table (SDT)

Contiene datos que describen los servicios en el sistema, como por ejemplo: nombres de los servicios, nombre del proveedor y otros parámetros asociados a cada servicio de un mismo múltiplex. Se transporta por paquetes identificados con PID=0x0011.

3. Event Information Table (EIT)

Se utiliza para transmitir información relativa a los acontecimientos en curso o futuros en el múltiplex MPEG recibido en la actualidad, y eventualmente sobre

otros multiplex MPEG, tal como: denominación, hora de comienzo, duración, etc. Se transporta por los paquetes identificados con PID=0x0012

4. Time & Date Table (TDT)

Esta tabla proporciona información relativa a la hora y fecha del momento, y se utiliza para poner en hora el reloj interno del receptor. La citada información se incluye en una tabla específica debido a las frecuentes actualizaciones de la misma. Se transporta por paquetes identificados mediante PID=0x0014.

1.4.5. Tablas opcionales de ISDB-T - SI

1. Bouquet Association Table (BAT)

El término "bouquet" se usa para referirse a una "Colección de servicios comercializados como entidad única". Las tablas BAT proporcionan información relativa a los "bouquets", además de informar del nombre del bouquet aportan la lista de los servicios disponibles en cada bouquet. Se transportan por paquetes identificados con PID=0x0011.

2. Running Status Table (RST)

Las tablas RST actualizan de forma rápida la información relativa a la situación de un acontecimiento (que está o no sucediendo). Se transmiten una sola vez y no de forma repetitiva por los paquetes identificados mediante PID=0x0013.

3. Time Offset Table (TOT)

Proporciona información relativa a la fecha y hora real así como a la diferencia horaria local ("*local time offset*"). Se actualiza frecuentemente, siendo transmitida por paquetes identificados con PID=0x0014 (como las TDT).

4. Stuffing Tables (ST)

Estas tablas de "relleno" se emplean para invalidar tablas que ya no sirven. Por ello usan paquetes que comparten valores de PID con otros tipos de tablas: 0x0010; 0x0011; 0x0012 0x0013 y 0x0014.

1.4.6. Códigos de PID

Tabla	Valor de PID
PAT	0x0000
CAT	0x0001
TSDT	0x0002
Reservados	0x0003 a 0x000F
NIT,ST	0x0010
SDT,BAT,ST	0x0011
EIT,ST	0x0012
RST,ST	0x0013
TDT,TOT,ST	0x0014
Sincronización de Red	0x0015
Paquetes Nulos	0x1FFF
PMT	0xFFFF

Tabla 1.3: Códigos de PID para los distintos tipos de tablas

1.5. Arquitectura de las Tablas de Información de Servicio

Las tablas PSI definidas por MPEG-2 (documento ISO/IEC 13818-1) y las tablas SI están constituidas por una o más secciones que se distribuyen a lo largo de una serie de paquetes de transporte del multiplex. Una sección es la estructura sintáctica que debe emplearse para insertar las citadas tablas en los paquetes de transporte. La estructura sintáctica usada en ISDB-T para las tablas SI, coincide con la especificada por MPEG-2 para las secciones de las tablas PSI de tipo "Private".

1.5.1. Distribución de las secciones en los paquetes de transporte

Al contrario que la distribución de los PES entre los Transport Packets (ver figura 1.7), las secciones de las tablas PSI y SI no tienen que empezar ni acabar forzosamente con un paquete de transporte. Sin embargo, si un paquete TS contiene el comienzo de al menos una sección, su payload unit start indicator (ver figura 1.10) debe ponerse a "1". La posición de comienzo de una nueva sección en un paquete TS está indicada por el

”Pointer field”, que debe ser el primer byte de la carga útil del citado paquete y proporciona el desplazamiento (offset), en bytes, desde el final del pointer field hasta el primer byte de dicha sección.

Así, un valor de 0x00 indica que la sección comienza inmediatamente después del pointer field. Si ninguna sección tiene su comienzo dentro del paquete TS, no habrá pointer field y el ”payload unit start indicator” se pondrá a 0.

Solamente hay un ”pointer field” en un paquete TS para marcar la posición de la primera sección que tenga su comienzo dentro de él, puesto que el comienzo de cualquier otra sección de una tabla se puede identificar fácilmente contando la longitud de las secciones precedentes, al no permitir la sintaxis que haya huecos entre ellas.

La figura 1.14 ilustra esta situación en el caso general.

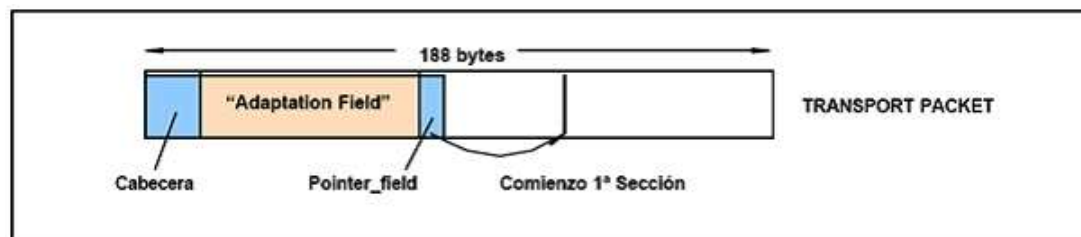


Fig. 1.14: Primer comienzo de sección dentro de un paquete TS
Delgado, Alejandro. Flujos de Programa y Transporte MPEG-2. pág. 25

Si una sección termina antes del final de un paquete de transporte, pero no es conveniente abrir otra dentro del mismo paquete, se debe emplear un mecanismo de relleno que emplee para cada byte sobrante el valor 0xFF. Consecuentemente, este valor no debe usarse para el byte que identifica el tipo de tabla: ”table id” (ver siguiente apartado). No obstante, también puede usarse como método de relleno de los paquetes de transporte el mecanismo ”Adaptation field”.

1.5.2. Composición de las secciones

Las secciones pueden tener longitud variable. Dentro de cada tabla, la longitud de las secciones está limitada a 1K (1.024 bytes), excepto en el caso de las secciones de las tablas ”Private” y ”EIT” cuya longitud está limitada a 4K (4.096 bytes).

Cada sección se identifica por la combinación, entre otros, de los siguientes elementos:

- **table_id**

El campo "table_id" identifica a qué tipo de tabla pertenece la sección. Algunos valores de table_id han sido definidos por ISO y otros por ETSI. También el usuario puede asignar otros valores para uso privado. La tabla 1.4 muestra la lista de valores de table_id.

- **table_id_extension**

Se usa para identificar una subtabla.

- **version_number**

Cuando las características descritas del flujo de transporte (TS) cambian; por ejemplo: comienzo de nuevos acontecimientos, diferente composición de los flujos elementales para un servicio dado, etc, deben transmitirse los nuevos datos de la Información del Servicio (SI) conteniendo la información actualizada. Esto se hace enviando una subtabla con los mismos identificadores que en las subtablas previas que contienen los datos relevantes, pero incrementando en 1 el valor previo de "version number".

- **current_next_indicator**

Cada sección debe numerarse como válida "ahora" ("current") o válida en un inmediato futuro ("next"). Esto permite la transmisión de una versión futura de la misma antes del cambio, dando al decodificador la oportunidad de prepararse para ello. No hay, sin embargo, ningún requerimiento para transmitir por adelantado la siguiente versión de una sección, pero si se hace, deberá ser la versión correcta inmediata de dicha sección.

- **section_number**

Este campo (que se incrementa en 1 con cada sección adicional), permite que las diferentes secciones de una sub-tabla particular sean reensambladas en su orden original por el decodificador. No obstante, se recomienda que las secciones se transmitan en orden numérico, a menos que se quieran transmitir algunas secciones de la sub-tabla con más frecuencia que otras (por ejemplo debido a consideraciones de acceso aleatorio).

table_id	Descripción
0X00	program_association_section
0X01	conditional_access_section
0X02	program_map_section
0X03	transport_stream_description_section
0X04 a 0X3F	Reservado
0X40	network_information_section- actual_network
0X41	network_information_section- other_network
0X42	service_description_section-actual_transport_stream
0X43 a 0X45	Reservado para usos futuros
0X46	service_description_section-other_transport_stream
0X47 a 0X49	Reservado para usos futuros
0X4A	bouquet_association_section
0X4B a 0X4D	Reservado para usos futuros
0X4E	event_information_section-actual transport_stream, present/following
0X4F	event_information_section-other transport_stream, present/following
0X50 a 0X5F	event_information_section-actual transport_stream, schedule
0X60 a 0X6F	event_information_section-other transport_stream, schedule
0X70	time_date_section
0X71	running_status_section
0X72	stuffing_section
0X73	time_offset_section
0X74 a 0X7D	Reservado para usos futuros
0X7E	discontinuity_information_section
0X7F	selection_information_section
0X80 a 0XFE	Definido por el usuario
0XFF	Reservado para relleno (stuffing)

Tabla 1.4: Valores de "table_id" para las distintas tablas SI

Capítulo 2

Especificación de los Requisitos de Software

2.1. Introducción

2.1.1. Propósito

El propósito de este capítulo es presentar una descripción general de las funcionalidades del Analizador de Paquetes Transport Stream para ISDB-Tb y está dirigido a los responsables del proyecto y desarrolladores del software.

2.1.2. Ámbito del Sistema

El nombre de la aplicación a desarrollar es **Analizador de Paquetes Transport Stream para ISDB-Tb** y en adelante lo conoceremos como **A.P.T.S.**

Tendrá 2 versiones:

- La primera versión (V.1) será elaborada en el entorno de desarrollo integrado (IDE) de MATLAB empleando el lenguaje de programación M.
- La segunda versión (V.2) será desarrollada en el IDE Netbeans utilizando el lenguaje de Programación Java

Sus funcionalidades serán las siguientes:

1. Apertura exclusiva de archivos de extensión .ts.

2. Visualización de datos referentes al archivo de extensión .ts tales como ubicación, tamaño en kB, cantidad total de paquetes de transporte de 188 Bytes contenidos en el mismo y el número del paquete de Transporte que está siendo analizado.
3. Decodificación de cada paquete de transporte de 188 Bytes del archivo de extensión .ts en formato hexadecimal y su respectivo despliegue en una tabla de datos matricial.
4. Búsqueda de paquetes de transporte ingresando su respectivo número. Si el dato ingresado está fuera del rango de paquetes contenidos en el archivo .ts o es incorrecto no se procesará.
5. Visualización de los elementos de la cabecera de cada paquete de transporte así como también la tabla a la cual pertenece y la condición del Campo de Adaptación.
6. Búsqueda de los PID contenidos en todo el archivo .ts y generación de su respectivo reporte. ¹ (*)
7. Elaboración de un reporte de los elementos de la cabecera para un intervalo máximo de 4000 paquetes. (*)
8. Almacenamiento de los reportes generados en archivos de extensión .csv. (*)

2.1.3. Definiciones, Acrónimos y Abreviaturas

- a) **Transport Stream (TS).**- Protocolo de comunicación para audio, video y datos especificado en los estándares de MPEG-2.
- b) **Transport Packet.**- Paquete de Transporte que constituye la unidad básica de composición del TS y tiene una longitud fija de 188 bytes.
- c) **TS Header.**- Cabecera del paquete de transporte que tiene una longitud fija de 4 bytes.
- d) **PID.**- Identificador del Paquete de Transporte.
- e) **MPEG.**- Grupo de expertos en imágenes en movimiento.
- f) **MPEG-2.**- Designación para un grupo de estándares de codificación de audio y video acordado por MPEG y publicados como estándar ISO 13818.

¹(*) Sólo para la versión V.2

- g) **IDE.-** Entorno de Desarrollo Integrado que constituye un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse exclusivamente a un solo lenguaje de programación o utilizarse para varios.
- h) **MATLAB.-** Software matemático que ofrece un IDE con un lenguaje de programación propio (lenguaje M).
- i) **Java.-** Lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995.
- j) **CSV.-** Tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma y las filas por saltos de línea.
- k) **A.P.T.S. .-** Analizador de Paquetes Transport Stream.
- l) **Usuario.-** Persona encargada de utilizar el A.P.T.S.

2.1.4. Referencias

- IEEE-830-STD-1998 ²
- Desarrollo de Software Analizador de Transport Stream para ISDB-T ³
- MPEG-2 Transport Stream Packet Analyzer ⁴

²<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

³<http://gingaperu.wordpress.com>

⁴<http://www.pjdaniel.org.uk/mpeg/>

2.1.5. Visión General

En la sección 2.2 se encuentra la descripción general del Software a desarrollar, sus funcionalidades, tipo de usuario y restricciones en el diseño del mismo. En la Sección 2.3 se indican los requisitos específicos de la aplicación tanto funcionales como no funcionales.

2.2. Descripción General

2.2.1. Perspectiva del Producto

La aplicación A.P.T.S. funciona como un sistema independiente y estará disponible en todo momento. No interactuará con Sistemas de Gestión de Bases de Datos y podrá ser ejecutado en un ordenador (PC, Servidor, Laptop) que cumpla con los requisitos que se detallarán más adelante.

2.2.2. Funciones del Producto

Las funciones que debe realizar el A.P.T.S. se pueden dividir en varios bloques:

a) Función de Apertura de Archivos

- Apertura exclusiva de archivos de extensión .ts.
- Visualización de un mensaje de error cuando el archivo seleccionado no sea de extensión .ts.

b) Función de Decodificación y Procesamiento de los Paquetes de Transporte de los archivos de extensión .ts

- Visualización de datos correspondientes a cada archivo .ts tales como ubicación (*path*), tamaño en kbytes, cantidad total de paquetes de 188 bytes y el número del paquete que se está analizando.
- Despliegue de la información del Paquete de Transporte analizado en formato hexadecimal a través de una tabla de datos matricial, así como también de cada uno de los elementos de su cabecera, la tabla a la cual pertenece y su condición de Campo de Adaptación. La aplicación contará con una barra de progreso que indicará la posición del paquete en el archivo y será empleada en otros procesos.

c) Función de Búsqueda de Paquetes de Transporte

- Al abrir el archivo de extensión .ts el Paquete de Transporte procesado que aparecerá por defecto será el N° 1.
- Avance y retroceso tanto al paquete siguiente como al paquete anterior del Paquete de Transporte que está siendo analizado.
- Avance y retroceso tanto al primero como al último Paquete de Transporte del archivo de extensión .ts.
- Búsqueda de un Paquete de Transporte específico ingresando su número siempre y cuando se encuentre en el rango de paquetes contenidos en el archivo de extensión .ts. Si los datos ingresados son incorrectos no se procesará la información y se mostrará un mensaje de error.

d) Función de Generación de Reportes de PID (Sólo V.2)

- Generación de un reporte sintetizado en donde consten los PID encontrados en el archivo de extensión .ts, su cantidad, porcentaje respecto a la cantidad total de PID y a que tabla pertenecen.
- Los datos obtenidos podrán ser guardados en un archivo de extensión .csv y pueden ser abiertos posteriormente en MS Excel con la opción de Importar datos externos.

e) Función de Generación de Reportes de Cabeceras de los Paquetes de Transporte (Sólo V.2)

- Generación de un Reporte de Cabeceras de los Paquetes de Transporte. El usuario deberá ingresar tanto el paquete inicial como el paquete final, estos valores deben estar en el rango de paquetes contenidos en el archivo de extensión .ts y su diferencia debe ser menor o igual a 4000 paquetes. Si los datos ingresados son incorrectos se desplegará un mensaje de error y no se procesará la información.
- Los datos proporcionados por el reporte serán: la ubicación del archivo, el paquete inicial, el paquete final y la cantidad de paquetes procesados. Adicionalmente se podrá filtrar la información del reporte en base a los Códigos PID (ver pág. 41).
- Los datos obtenidos podrán ser guardados en un archivo de extensión .csv y pueden ser abiertos posteriormente en MS Excel con la opción de Importar datos externos.

2.2.3. Características de los Usuarios

Tipo de Usuario	Usuario
Formación	Formación Superior Técnica. Conocimientos de TV Digital.
Habilidades	Manejo del Protocolo Transport Stream, MATLAB y Java
Actividades	Uso del A.P.T.S.

Tabla 2.1: Características del Usuario "Usuario"

Tipo de Usuario	Desarrollador de Software
Formación	Formación Superior Técnica. Conocimientos de TV Digital.
Habilidades	Manejo del Protocolo Transport Stream, Programación en MATLAB y Java
Actividades	Desarrollo del A.P.T.S.

Tabla 2.2: Características del Usuario "Desarrollador de Software"

2.2.4. Restricciones

- El A.P.T.S. no interactuará con Sistemas de Gestión de Bases de Datos (S.G.B.D.).
- El A.P.T.S. procesará información solamente de archivos de extensión .ts.
- La eficiencia en la Generación de Reportes de PID dependerá de la cantidad de Paquetes de Transporte contenidos en el archivo de extensión .ts y de la capacidad de procesamiento del ordenador en donde se ejecute el A.P.T.S.
- Para la generación de Reportes de Cabeceras se limitó la cantidad de 4000 los paquetes de transporte a ser analizados con el fin de evitar desbordamientos de la memoria de la maquina virtual de java.
- \LaTeX será utilizado para todos los documentos relacionados con el A.P.T.S.
- La Ingeniería de Software será desarrollada empleando notación UML (Unified Modeling Language).
- El IDE a ser usado para la programación con Java será Netbeans.
- Para la implementación del A.P.T.S. se emplearán Interfaces Gráficas de Usuario (GUI), Programación Orientada a Objetos con Java y funciones .m con MATLAB.

2.2.5. Suposiciones y Dependencias

- Debe tomarse en cuenta las versiones tanto de los Sistemas Operativos como de las Herramientas de Desarrollo que aparezcan posterior a la elaboración del A.P.T.S., ya que éstos pueden presentar modificaciones en su estructura que pueden alterar el funcionamiento y desempeño del mismo.

2.2.6. Requisitos futuros

- Se debe extender la funcionalidad del A.P.T.S. para que interactúe con un módulo de Hardware mediante una interfaz de comunicación.

2.3. Requisitos Específicos

2.3.1. Interfaces

Interfaces de de Usuario

1. GUI en MATLAB

En la figura 2.1 se muestre esquemáticamente la pantalla principal de la GUI del A.P.T.S. con todos sus componentes y en la tabla 2.3 se indican sus características.

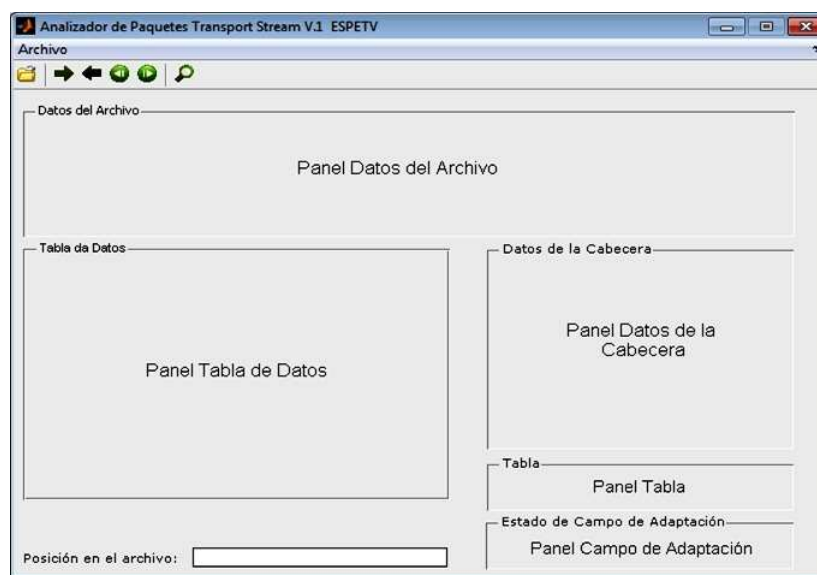


Fig. 2.1: GUI para MATLAB

Objeto	Descripción	Tamaño [pixels]
GUI MATLAB	Pantalla Principal del A.P.T.S.	Ancho = 667 Alto = 439
Panel Datos del Archivo	Contiene los datos del archivo .ts	Ancho = 642 Alto = 123
Panel Tabla de Datos	Despliega los datos del paquete de Transporte en formato hexadecimal	Ancho = 355 Alto = 238
Panel Datos de la Cabecera	Indica los datos de la Cabecera del Paquete de Transporte	Ancho = 255 Alto = 191
Panel Tabla	Muestra la Tabla a la que pertenece el Paquete de Transporte	Ancho = 255 Alto = 50
Panel Campo de Adaptación	Indica el estado de Campo de Adaptación	Ancho = 255 Alto = 50

Tabla 2.3: Características de la GUI del A.P.T.S. con MATLAB

2. GUI en Netbeans IDE

En la figura 2.2 se encuentra de manera detallada la GUI del A.P.T.S. con Netbeans IDE con todos sus componentes. Como se puede apreciar es similar a la GUI de MATLAB, lo que cambia son las dimensiones, las mismas se indican en la tabla 2.4

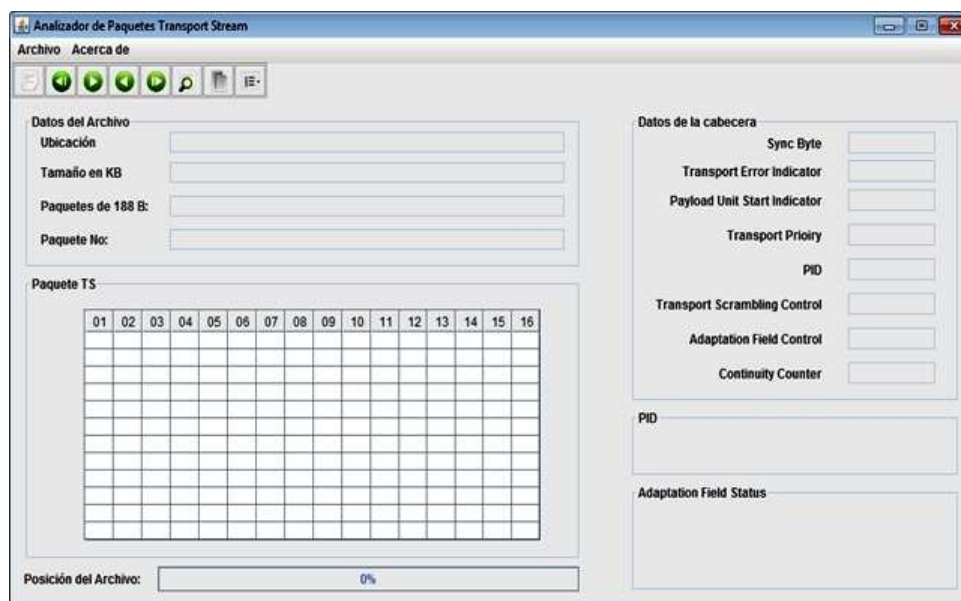


Fig. 2.2: GUI para Netbeans IDE

Objeto	Tamaño [pixels]
GUI Netbeans IDE	Ancho = 925 Alto = 524
Panel Datos del Archivo	Ancho = 548 Alto = 149
Panel Tabla de Datos	Ancho = 521 Alto = 263
Panel Datos de la Cabecera	Ancho = 307 Alto = 263
Panel Tabla	Ancho = 307 Alto = 78
Panel Campo de Adaptación	Ancho = 255 Alto = 78

Tabla 2.4: Características de la GUI del A.P.T.S. con Netbeans IDE

Interfaces de Hardware

El ordenador donde se ejecute el A.P.T.S. debe tener las siguientes características:⁵

- Procesador: Intel CORE i3 o superiores
- Memoria RAM: 4 GB

Interfaces de Software

1. Herramientas de Desarrollo de Software

- MATLAB R2009a o versiones superiores.
- Netbeans IDE 7.2 para plataformas Windows y Linux
- jdk 7 para plataformas Windows y Linux.

2. Sistemas Operativos

- Microsoft Windows 7 / Microsoft Windows XP.
- Ubuntu 12.04 / CentOS 6.

3. Software para modelado UML

- Star UML
- Netbeans IDE 7.2

⁵ Estas características son para un rendimiento óptimo del A.P.T.S. Puede usarse computadores de menor capacidad pero la velocidad del procesamiento en la Generación de Reportes puede ser menor.

2.3.2. Requisitos Funcionales

1. Apertura de Archivos

- **Nombre:** Abrir Archivo.
- **Objetivo:** Abrir un archivo de extensión .ts para analizar su estructura.
- **Entrada:** Archivo.
- **Proceso:** Se validará la extensión del archivo seleccionado. Si éste no es de extensión .ts se mostrará un mensaje de error y el usuario deberá escoger otro archivo, caso contrario el A.P.T.S. procederá a procesar la información del mismo.
- **Salida:** En la Pantalla Principal del A.P.T.S. se mostrarán los siguientes datos del archivo .ts y del Paquete de Transporte N° 1:
 - 1) Ubicación del archivo .ts en el ordenador, su tamaño en kbytes, cantidad total de paquetes de transporte contenidos en el mismo y el número del paquete de transporte analizado que en este caso será el 1.
 - 2) Información de los 188 bytes del Paquete de Transporte N° 1 en formato hexadecimal desplegada en una tabla de datos matricial.
 - 3) Valores de los elementos de la Cabecera del Paquete de Transporte N° 1.
 - 4) Información de la Tabla a la que pertenece el Paquete de Transporte N° 1 y su estado de Campo de Adaptación.

2. Búsqueda de Paquetes de Transporte

- **Nombre:** Buscar Paquete.
- **Objetivo:** Buscar un paquete de transporte específico en el archivo de extensión .ts.
- **Entrada:** Archivo de extensión .ts , número del paquete de transporte.
- **Proceso:** Se pueden presentar los siguientes casos:
 - 1) Si el usuario desea buscar el siguiente paquete de transporte, el número del paquete incrementará su valor en 1 y el A.P.T.S. procesará la información. Si el nuevo número del paquete de transporte es mayor a la cantidad total de estos contenidos en el archivo .ts se mostrará un mensaje de error.

- 2) Si el usuario desea buscar el paquete de transporte anterior, el número del paquete decrementará su valor en 1 y el A.P.T.S. procesará la información. Si el nuevo número del paquete de transporte es menor a 1 se mostrará un mensaje de error.
 - 3) Si el usuario desea buscar el primer paquete de transporte, el número del paquete fijará su valor en **1** y el A.P.T.S. procesará la información.
 - 4) Si el usuario desea buscar el último paquete de transporte, el número del paquete fijará su valor en la **cantidad total** de estos contenidos en el archivo .ts y el A.P.T.S. procesará la información.
 - 5) Si el usuario desea buscar un paquete de transporte específico debe ingresar el número del mismo por teclado a través de una interfaz, éste fijará su valor de acuerdo al dato ingresado y el A.P.T.S. procesará la información. Si el dato es incorrecto, es decir no es un valor numérico o está fuera del rango de paquetes de transporte contenidos en el archivo.ts se mostrará un mensaje de error.
- **Salida:** La información obtenida luego del proceso corresponderá al paquete de transporte seleccionado por el usuario de acuerdo a los casos mencionados anteriormente y esta se mostrará de manera similar a la salida en el ítem apertura de archivos. En una barra de progreso se indicará la posición del paquete en el archivo .ts.

3. Generación de Reporte de PID

- **Nombre:** Generar Reporte de PID.
- **Objetivo:** Obtener información sintetizada referente a los PID existentes en el archivo de extensión .ts.
- **Entrada:** Archivo de extensión .ts , cantidad total de paquetes de transporte.
- **Proceso:** El A.P.T.S. obtendrá los PID de todos los paquetes de transporte contenidos en el archivos .ts, el avance de este proceso se visualizará a través de una barra de progreso.
- **Salida:** Una vez finalizado el proceso, el A.P.T.S. mostrará una pantalla con un reporte sintetizado de los PID contenidos en el archivo .ts. en donde consten los PID encontrados, su cantidad, porcentaje respecto a la cantidad total de PID y a que tabla pertenecen.

4. Generación de Reporte de Cabeceras de los Paquetes de Transporte

- **Nombre:** Generar Reporte de Cabeceras de los Paquetes de Transporte .
- **Objetivo:** Obtener información referente a las cabeceras de los paquetes de transporte del archivo de extensión .ts.
- **Entrada:** Archivo de extensión .ts , paquete de transporte inicial , paquete de transporte final.
- **Proceso:** El usuario ingresará por teclado a través de una interfaz los valores correspondientes al paquete de transporte inicial y final, éstos deben estar en el rango de paquetes existentes en el archivo .ts y su diferencia debe ser menor o igual a 4000. Si los datos ingresados son incorrectos aparecerá un mensaje de error y se debe ingresar nuevamente los valores. Posteriormente el A.P.T.S. procesará cada uno de los paquetes del intervalo seleccionado y obtendrá los valores de los elementos de sus cabeceras.El avance de este proceso se visualizará a través de una barra de progreso.
- **Salida:** Una vez finalizado el proceso, el A.P.T.S. mostrará una pantalla con el reporte de las cabeceras en donde conste la ubicación del archivo .ts, el paquete inicial, el paquete final, la cantidad de paquetes procesados, el número del paquete y los datos de los elementos de las cabeceras. Adicionalmente el usuario podrá filtrar la información del reporte en base a los Códigos PID (ver pág. 41) y se indicará la cantidad de elementos encontrados luego del filtrado.

5. Almacenamiento de información en archivos de extensión .csv

- **Nombre:** Almacenar información en archivos de extensión .csv .
- **Objetivo:** Guardar información de reportes tanto de PID como de cabeceras en formato .csv.
- **Entrada:** Reportes de PID o Cabecera.
- **Proceso:** El usuario escogerá la opción de almacenamiento de datos en las respectivas pantallas de los reportes y el A.P.T.S. procesará esta información.
- **Salida:** Archivo de extensión .csv ya sea del reporte de PID o de cabeceras. Éste podrá ser utilizado posteriormente con MS Excel u otros programas que soporten este tipo de archivos.

2.3.3. Requisitos no funcionales

Rendimiento

El tiempo de respuesta del A.P.T.S. en la apertura de archivos y búsqueda de paquetes será inmediata. Para la Generación de Reportes el rendimiento de la aplicación dependerá de la cantidad de paquetes de transporte contenidos en el archivo .ts y de la capacidad de procesamiento del ordenador.

Seguridad

- El A.P.T.S. no procesará datos de información reservada tales como login, password o claves secretas.
- Debe existir validación de ingreso de datos tanto para la apertura de archivos como para la búsqueda de paquetes y generación de reportes.

Fiabilidad

- Se espera que el A.P.T.S sea una aplicación muy fiable para el usuario.
- Los desarrolladores del A.P.T.S. deberán documentar que medidas debe tomar el usuario en caso de que se presente desbordamiento de memoria.

Disponibilidad

El A.P.T.S. deberá estar disponible cuando el usuario requiera hacer el análisis de un archivo de extensión .ts.

Portabilidad

- La segunda versión del A.P.T.S. (V.2 en Java) será desarrollada con la concepción de multiplataforma, es decir que pueda funcionar tanto en Windows como en Linux.
- El A.P.T.S. deberá ser portable a versiones futuras de las Herramientas con las que será desarrollado.

Mantenibilidad

- El A.P.T.S no requerirá de una instalación compleja.

- Se proporcionará a los usuarios los archivos ejecutables para que puedan probar su funcionamiento en cualquier ordenador que cumpla con las características tanto de hardware como de software mencionadas anteriormente.

Capítulo 3

Desarrollo del Analizador de Paquetes Transport Stream

3.1. Modelado con UML

UML - Unified Modeling Language es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del mismo. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual.

El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar. UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

En este capítulo se irán detallando los diferentes diagramas utilizados para desarrollar un modelo conceptual del A.P.T.S. basándonos en la Especificación de Requerimientos de Software (E.R.S.).

3.1.1. Identificación de los actores

En terminología UML, un actor especifica un papel desempeñado por un usuario o cualquier otra entidad que interactúa con el sistema.

Para el desarrollo del A.P.T.S. se han identificado los siguientes actores:

- **Usuario.-** Constituye un actor *primario* ya que éste inicializará el sistema y dirige la secuencia lógica del mismo.
- **Ordenador.-** Constituye un actor *secundario* y representa a un ente externo que interactúa con el A.P.T.S.

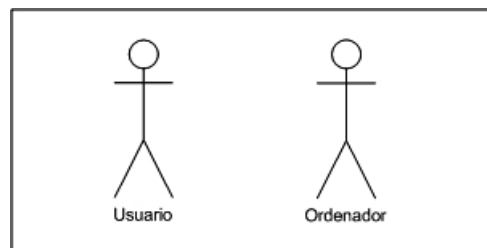


Fig. 3.1: Actores que intervienen en el A.P.T.S.

3.1.2. Diagramas de Casos de Uso

Un **Caso de Uso** se compone de una secuencia de eventos iniciada por el usuario del A.P.T.S. Los casos de uso son ideas simples y prácticas que no requieren muchas habilidades tecnológicas para ser utilizadas (a diferencia de las demás actividades de desarrollo). Por el contrario, si se volvieran muy complejas se perdería su utilidad.

A continuación se indican los casos de uso identificados para el desarrollo del A.P.T.S.

1. Caso de Uso Contextual del A.P.T.S.

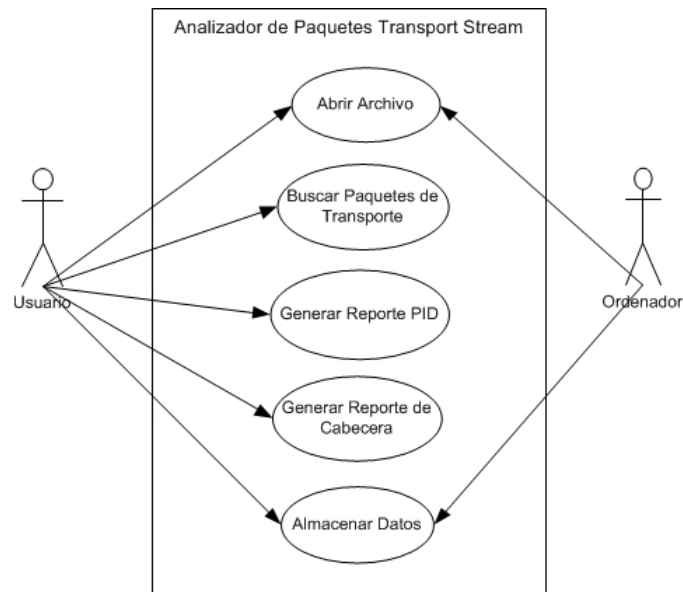


Fig. 3.2: Caso de Uso contextual del A.P.T.S.

En la figura 3.2 se indican los casos de uso que se ejecutarán en el A.P.T.S. y la relación de cada uno de éstos con los actores.

2. Relación entre Casos de Uso

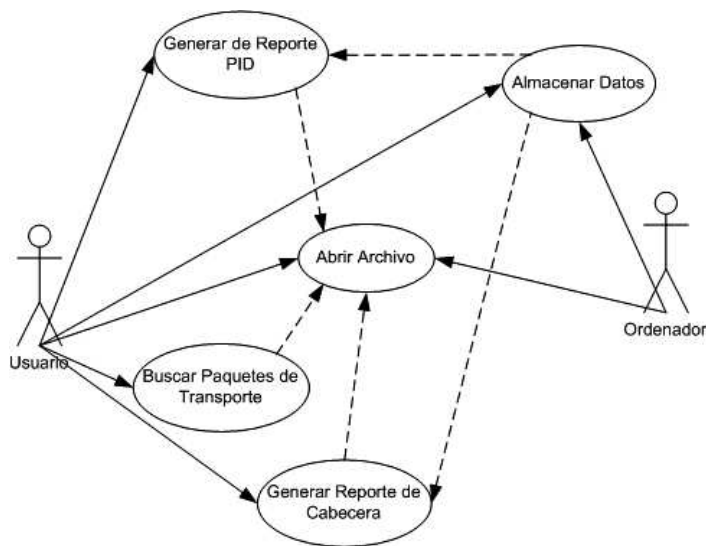


Fig. 3.3: Relación entre los Casos de Uso del A.P.T.S.

En la figura 3.3 se muestra las relaciones entre los casos de uso del A.P.T.S. El caso de uso Abrir Archivo es de tipo Básico (Principal) y tiene una relación de inclusión con los casos de uso Buscar Paquetes de Transporte, Generar Reporte PID y Generar Reporte de Cabecera, esto significa que requieren de la realización previa del caso de uso Abrir Archivo para después poder ser ejecutados.

El caso de uso Almacenar Datos tiene una relación de extensión con los casos de uso Generar Reporte PID y Generar Reporte de Cabecera, esto quiere decir que extiende la funcionalidad de los mismos pero su ejecución es opcional.

3. Documentación de Actores y Casos de uso

a) Actores

Actor	Usuario
Casos de Uso	Abrir Archivo, Buscar Paquetes de Transporte Generar Reporte de PID, Generar Reporte de Cabecera Almacenar Datos
Tipo	Primario
Descripción	Es el actor principal y representa a cualquier persona que desee utilizar el A.P.T.S.

Actor	Ordenador
Casos de Uso	Abrir Archivo, Almacenar Datos
Tipo	Secundario
Descripción	Es un actor secundario y representa al dispositivo donde se albergará el A.P.T.S.

b) Casos de Uso

Caso de Uso Abrir Archivo

Caso de Uso	Abrir Archivo
Tipo	Básico
Precondiciones	Ninguna
Descripción	Este caso de uso es inicializado por el usuario. Se escogerá un archivo de extensión .ts a través de una GUI para realizar el respectivo análisis de sus paquetes de transporte.
Secuencia	<ul style="list-style-type: none"> ■ El usuario selecciona un archivo .ts ■ El A.P.T.S desplegará la siguiente información: <ol style="list-style-type: none"> 1) Ubicación en el ordenador del archivo, tamaño en kbytes, cantidad total de paquetes de transporte contenidos en el mismo y el número del paquete que en este caso sera el 1. 2) Información de los 188 bytes del Paquete de Transporte en formato hexadecimal desplegada en una tabla de datos matricial. 3) Valores de los elementos de la Cabecera. 4) Información de la Tabla a la que pertenece y su estado de Campo de Adaptación. ■ Se visualizará mediante una barra de progreso la posición del paquete con respecto al archivo .ts
Excepciones	<ul style="list-style-type: none"> ■ Si el archivo seleccionado no es de extensión .ts saltará un mensaje de error y no se procesará. ■ Todas las opciones del A.P.T.S. deben encontrarse validadas en caso de no existir información
Resultado	Datos del archivo .ts en el ordenador y del paquete de transporte # 1.

Caso de Uso Buscar Paquetes de Transporte

Caso de Uso	Buscar Paquetes de Transporte
Tipo	Inclusión
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Abrir Archivo
Descripción	Este caso de uso permitirá ver la información de un paquete de transporte específico referente a sus 188 bytes decodificados en formato hexadecimal mediante una tabla de datos matricial, valor de los elementos de su cabecera, tabla a la que pertenece y su estado de campo de adaptación.
Secuencia	<ul style="list-style-type: none">■ El usuario tendrá las siguientes opciones:<ol style="list-style-type: none">1) Búsqueda del primer ó último paquete de transporte.2) Búsqueda del paquete de transporte siguiente ó anterior.3) Búsqueda de un paquete de transporte específico.■ De acuerdo a la opción ejecutada por el usuario, la información será procesada.■ Se visualizará mediante una barra de progreso la posición del paquete con respecto al archivo .ts
Excepciones	<ul style="list-style-type: none">■ Si la información ingresada por el usuario es incorrecta aparecerá un mensaje de error y la información no se procesará.
Resultado	Información del paquete de transporte indicada en el item Descripción de este caso de uso

Caso de Uso Generar Reporte PID

Caso de Uso	Generar Reporte PID
Tipo	Inclusión
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Abrir Archivo
Descripción	Este caso de uso permitirá generar un reporte sintetizado de los PID existentes en el archivo .ts .
Secuencia	<ul style="list-style-type: none">■ El usuario seleccionará la opción Generar Reporte PID.■ El A.P.T.S. procesará la información de la cabecera de cada uno de los paquetes de transporte contenidos en el archivo.ts .■ El avance del proceso se podrá visualizar mediante una barra de progreso .■ Finalmente aparecerá una GUI con una tabla de datos en donde consten los PID encontrados, su cantidad, porcentaje respecto a la cantidad total de PID, la tabla a la que tabla pertenecen, ubicación del archivo, cantidad total de PID encontrados y la cantidad de paquetes de transporte procesados.
Excepciones	Ninguna
Resultado	Reporte PID.

Caso de Uso Generar Reporte de Cabecera

Caso de Uso	Generar Reporte de Cabecera
Tipo	Inclusión
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Abrir Archivo
Descripción	Este caso de uso permitirá generar un reporte con información referente a las cabeceras de los paquetes de transporte en un intervalo máximo de 4000 de éstos contenidos en el archivo .ts .
Secuencia	<ul style="list-style-type: none"> ■ El usuario seleccionará la opción <i>Generar Reporte de Cabecera</i>. ■ Aparecerá una GUI donde el usuario deberá ingresar por teclado un valor numérico inicial y final. La diferencia entre estos dos debe ser menor o igual a 4000. ■ El A.P.T.S. procesará la información del intervalo de paquetes de transporte ingresado por teclado. ■ El avance del proceso se podrá visualizar mediante una barra de progreso . ■ Finalmente aparecerá una GUI con una tabla de datos en donde conste información tal como número de paquete inicial y final, cantidad de paquetes de transporte procesados, número de cada paquete y junto a éste cada uno de los elementos de su cabecera, ubicación del archivo. Además se podrá filtrar los datos obtenidos en base a los Códigos PID (ver pág. 41).
Excepciones	Si la información ingresada por el usuario es incorrecta aparecerá un mensaje de error y la información no se procesará.
Resultado	Reporte de Cabecera.

Caso de Uso Almacenar Datos

Caso de Uso	Almacenar Datos
Tipo	Extensión
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Generar Reporte PID ó Generar Reporte de Cabecera
Descripción	Este caso de uso permitirá guardar la información obtenida en el Reporte PID ó Reporte de Cabecera en un archivo de extensión .csv
Secuencia	<ul style="list-style-type: none"> ■ El usuario seleccionará la opción <i>Guardar como .csv</i>. ■ Aparecerá una GUI donde el usuario deberá ingresar por teclado el nombre con el que desea guardar el archivo. ■ El A.P.T.S. procesará la información y guardará el archivo en el ordenador.
Excepciones	Ninguna.
Resultado	Archivo .csv.

3.1.3. Diagramas de Secuencia

Un *diagrama de secuencia* muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para *cada caso de uso*. Contiene detalles incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

En esta sección se indicarán los diagramas de secuencia para cada uno de los casos de uso del A.P.T.S.

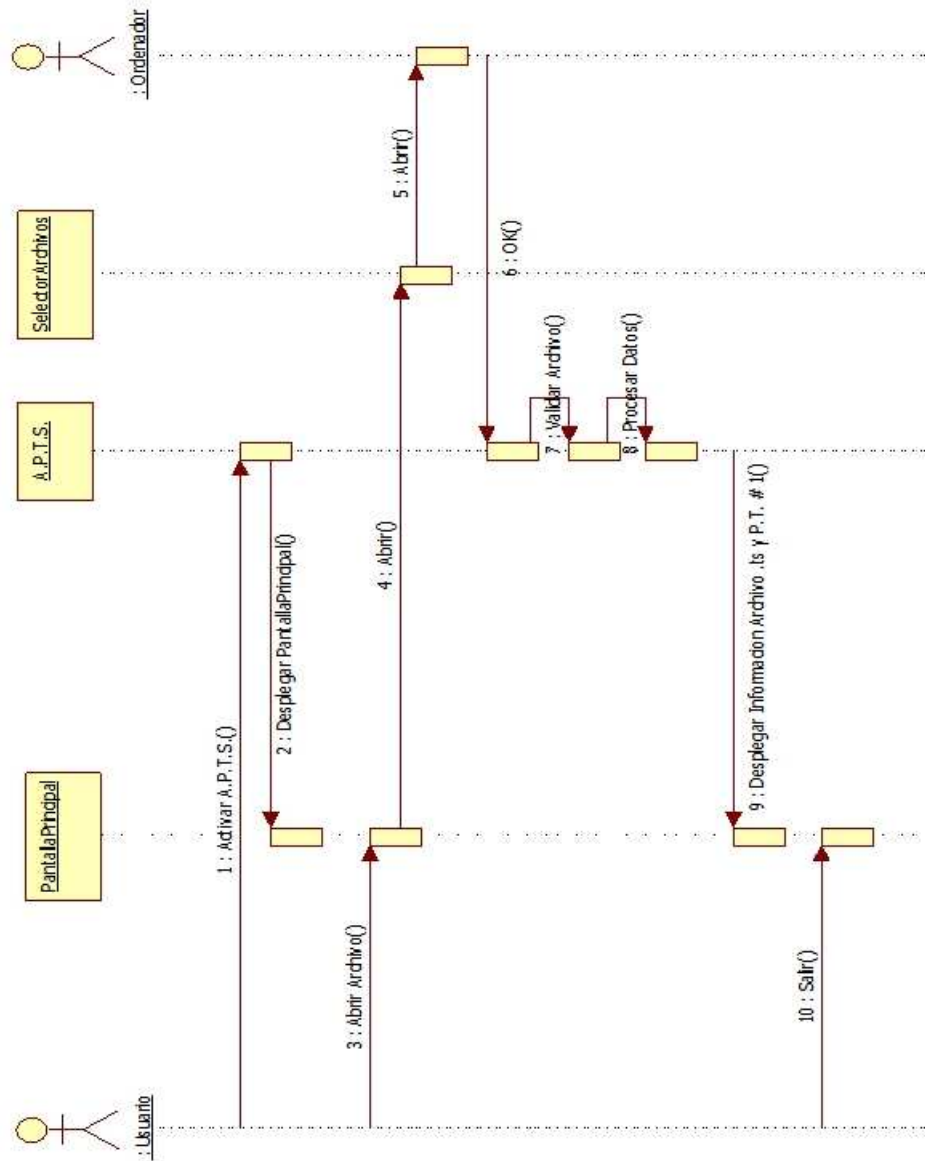


Fig. 3.4: Diagrama de Secuencia Abrir Archivo

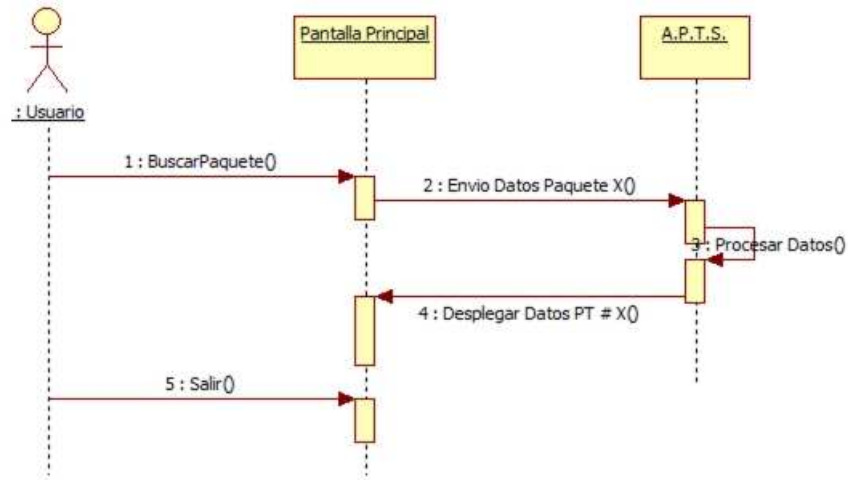


Fig. 3.5: Diagrama de Secuencia Buscar Paquete de Transporte

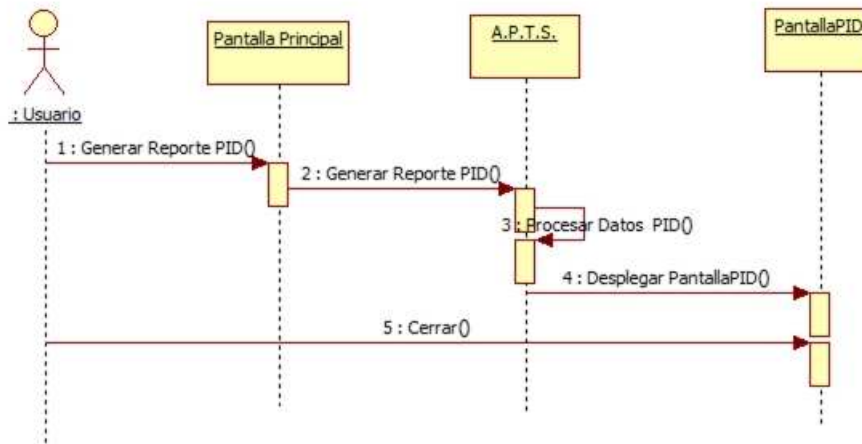


Fig. 3.6: Diagrama de Secuencia Generar Reporte PID

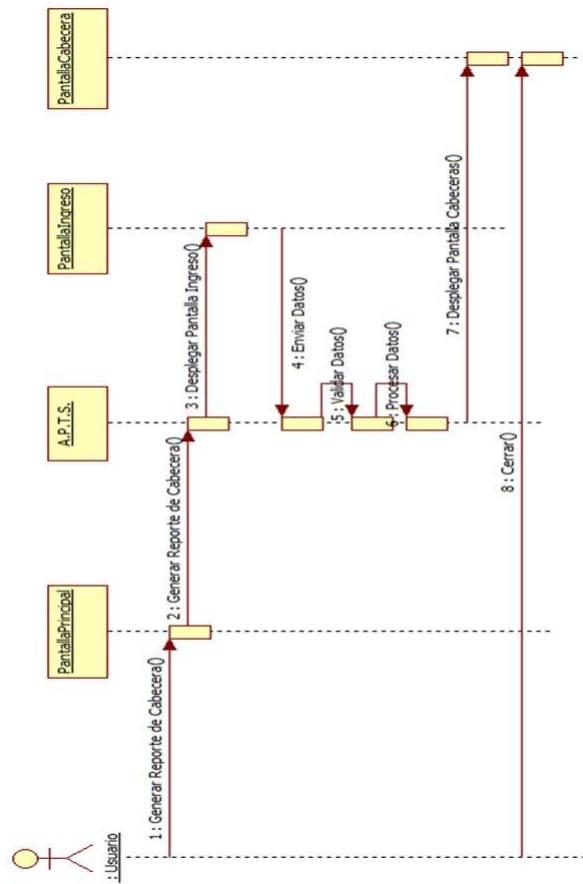


Fig. 3.7: Diagrama de Secuencia Generar Reporte de Cabecera

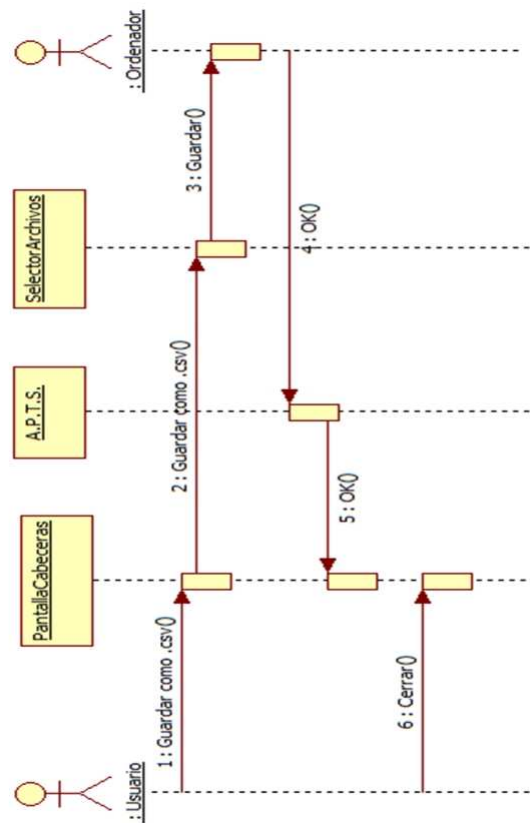


Fig. 3.8: Diagrama de Secuencia Almacenar Datos

3.1.4. Diagrama de Paquetes

En UML, un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas indicando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Para el desarrollo del A.P.T.S. se crearon 3 paquetes:

- Ejecutable.- Este paquete contendrá la clase Principal que permitirá la ejecución del A.P.T.S.
- Operaciones.- Este paquete albergará las clases cuyos métodos representan las funcionalidades del A.P.T.S.
- Formularios.- En este paquete constarán todas las Interfaces gráficas de Usuario del A.P.T.S.

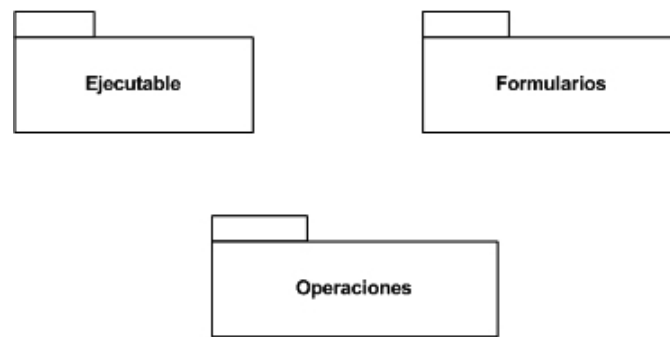


Fig. 3.9: Diagrama de Paquetes del A.P.T.S.

3.1.5. Diagramas de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el modelo conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

A continuación se mostrará las clases que componen el A.P.T.S.

La figura 3.10 representa la Clase Principal del A.P.T.S. Está contenida en el paquete Ejecutable y posee el método main.

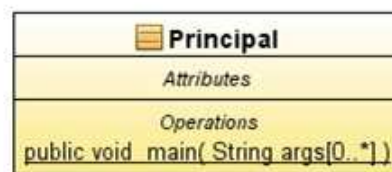


Fig. 3.10: Clase Principal del A.P.T.S.

Las clases mostradas en la figura 3.11 pertenecen al Paquete Operaciones. Sus métodos permitirán ejecutar cada una de las funcionalidades del A.P.T.S.

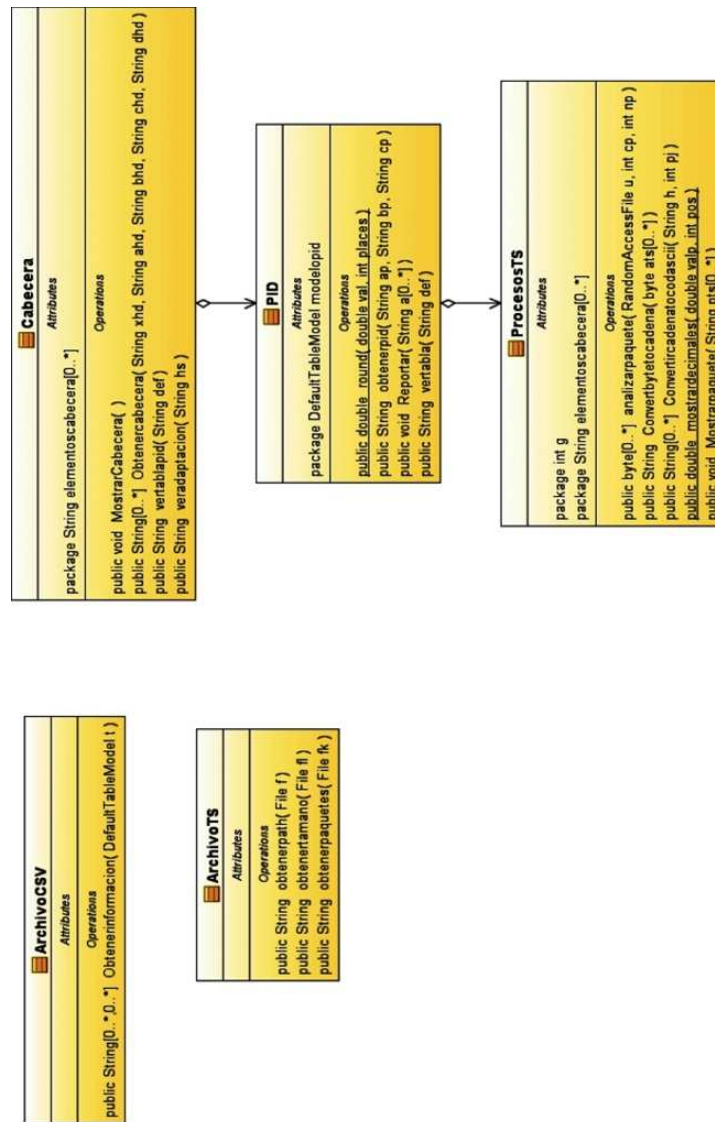


Fig. 3.11: Clases pertenecientes al Paquete Operaciones

En la figura 3.12 encontramos las clases que representan a las GUI del A.P.T.S. Pertenecen al paquete Formularios.

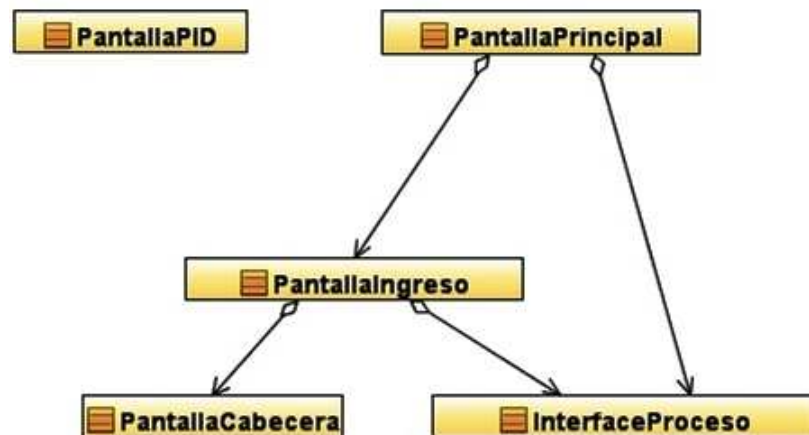


Fig. 3.12: Clases que representan las GUI del A.P.T.S.

3.1.6. Diagrama de Despliegue

El Diagrama de Despliegue es un tipo de diagrama UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

La figura 3.13 muestra el diagrama de despliegue del A.P.T.S.

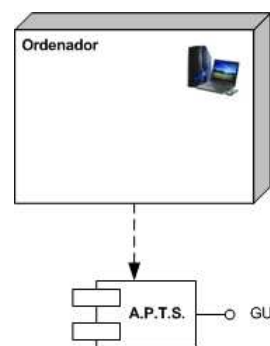


Fig. 3.13: Diagrama de Despliegue del A.P.T.S.

3.2. Consideraciones para el Diseño del A.P.T.S. con MATLAB

Para el desarrollo del A.P.T.S. con MATLAB se tomará en cuenta las siguientes consideraciones:

- Debido a las funcionalidades descritas en las Especificaciones de Requisitos de Software, sólo se utilizarán los casos de uso Abrir Archivo y Buscar Paquetes de Transporte para el desarrollo del A.P.T.S. con MATLAB.
- No se empleará el enfoque de Programación Orientada a Objetos con MATLAB.
- Los métodos para apertura de archivos, validación de datos, decodificación de paquetes de transporte y búsqueda de los mismos se instanciarán a través de funciones `.m`.
- La estructura del A.P.T.S. con MATLAB se indica en la figura 3.14:

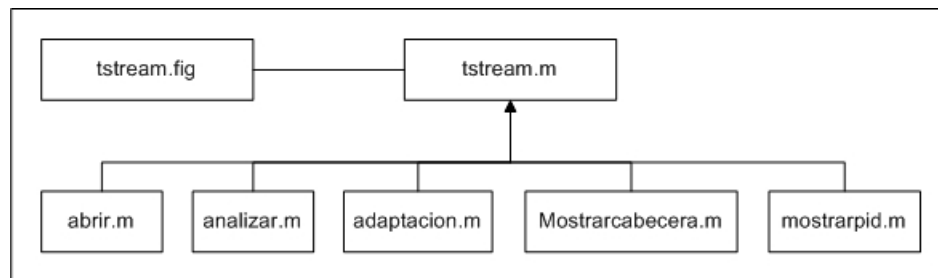


Fig. 3.14: Estructura del A.P.T.S. con MATLAB

tstream.fig corresponde a la Interfaz Gráfica de Usuario para MATLAB descrita en el capítulo 2 (ver pág. 51). Al crearla se generará automáticamente el archivo *tstream.m*.

A continuación se indican las funciones `.m` asociadas a este último con una breve descripción de lo que realiza cada una.

-
- ***abrir.m.***- Realiza la validación de datos al escoger el archivo de extensión .ts.
 - ***analizar.m.***- Ejecuta la decodificación de los paquetes de transporte en formato hexadecimal.
 - ***Mostracabecera.m.***- Obtiene los elementos de la cabecera del paquete de transporte
 - ***mostrarpid.m.***- Muestra la tabla a la que pertenece el paquete de transporte de acuerdo a su PID.
 - ***adaptacion.m.***- Indica el estado de Campo de Adaptación.

3.3. Diagramas de Flujo

3.3.1. Diagrama de Flujo para el A.P.T.S. con MATLAB

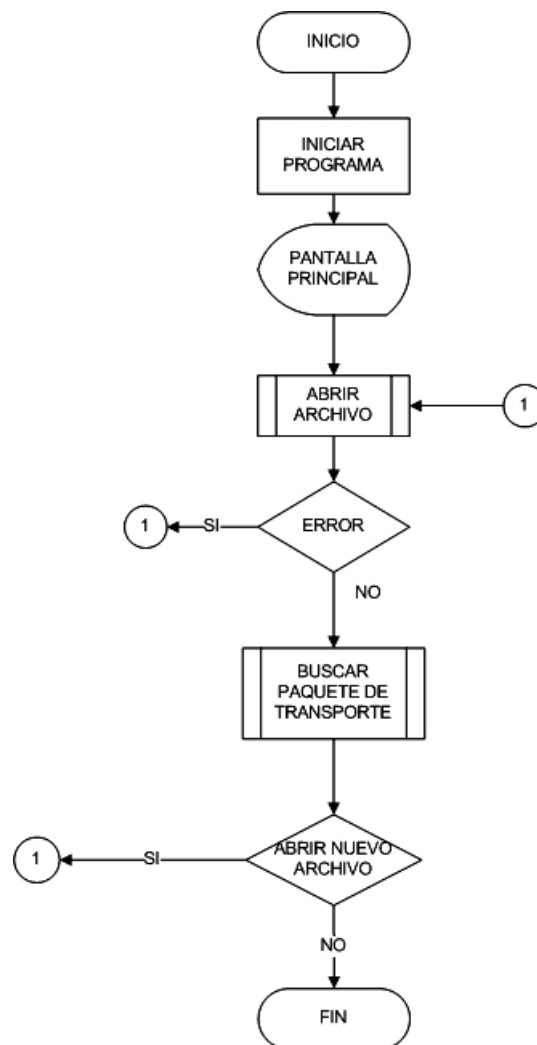


Fig. 3.15: Diagrama de Flujo para el A.P.T.S. con MATLAB

3.3.2. Diagrama de Flujo para el A.P.T.S. con Java

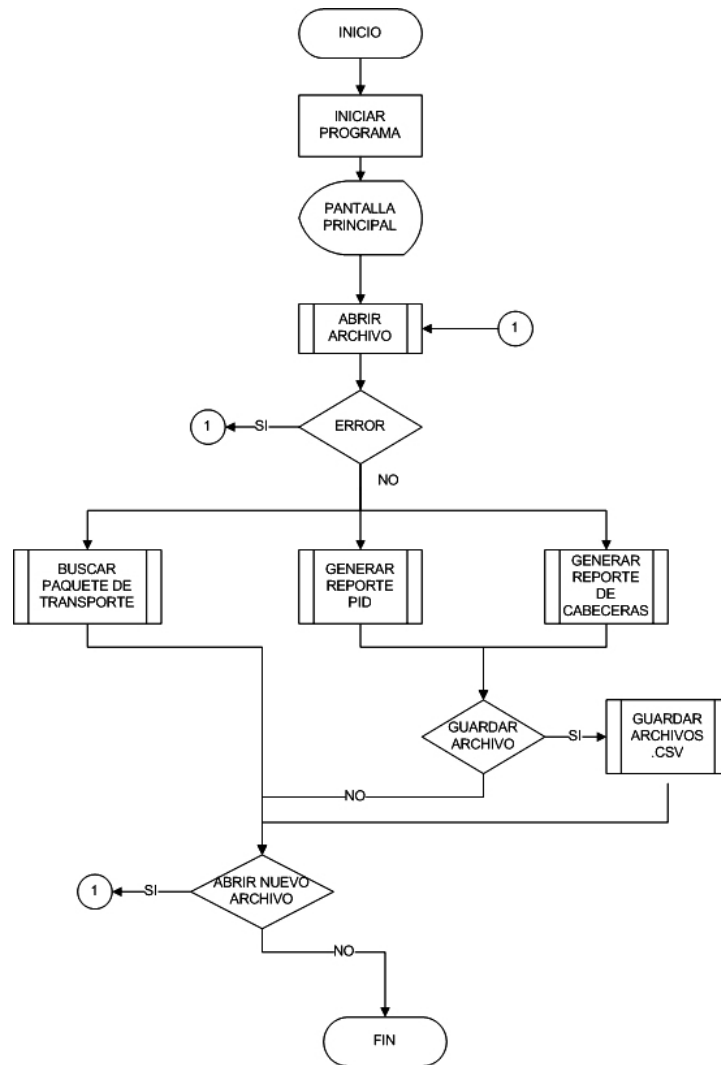


Fig. 3.16: Diagrama de Flujo para el A.P.T.S. con Java

3.4. Pseudocódigos

El conjunto de operaciones destinadas a la decodificación de paquetes de transporte son las siguientes:

Inicio

- a) Creación de un archivo.
- b) Ubicación en una posición x dentro del archivo.
- c) Lectura de 188 bytes.
- d) Conversión de los 188 bytes leídos a formato hexadecimal.
- e) Despliegue de información.

Fin

El aporte de este proyecto de grado constituyen los algoritmos de búsqueda tanto en MATLAB como en Java, los cuales indicaremos a continuación

3.4.1. Pseudocódigo para búsqueda de paquetes de transporte en MATLAB

```
Este algoritmo apunta a una posición  $k$  dentro de un archivo y desde dicha posición lee 188 bytes  
Entrada: archivo,  $m$  ( $m$  entero  $\geq 1$ )  
Salida: cadena  
Analizar(archivo,  $m$ )  
{  
     $k \leftarrow m-1$   
    posicionar(archivo,  $k * 188$ ) //la posición  $k*188$  indica el inicio de cada paquete de transporte  
    cadena  $\leftarrow$  leer(archivo, 188, caracteres) //lectura de 188 bytes  
    retornar cadena  
}
```

Fig. 3.17: Pseudocódigo del algoritmo de búsqueda en MATLAB

La función posicionar nos indica que la variable archivo será considerada como un archivo de acceso aleatorio, es decir que permitirá acceder a cualquier posición dentro del mismo como si fueran arrays en memoria. La variable m se constituye en el número del paquete de transporte.

3.4.2. Pseudocódigo para búsqueda de paquetes de transporte en Java

```
Este algoritmo apunta a una posición k dentro de un archivo , desde dicha posición lee m bytes y retorna un array de bytes de dimensión m

Entrada: archivo, m, n ( m entero >0, n entero >=1)

Salida: byte

Analizar(archivo, m, n)
{
    Dimension byte[m]
    k ← n-1
    posicionar(archivo, k * 188) // la posición k*188 indica el inicio de cada paquete de transporte
    i ← leer(archivo, byte) // lectura de m bytes
    retornar byte
}
```

Fig. 3.18: Pseudocódigo del algoritmo de búsqueda en Java

Este pseudocódigo tiene una estructura muy similar al de MATLAB. La diferencia radica en que la variable m representa la dimensión del array de bytes que para la búsqueda de paquetes de transporte sería 188 y n el número del paquete de transporte.

Además de la búsqueda de paquetes de transporte es empleado en la generación de reportes mediante estructuras repetitivas. En ese caso la variable m tomaría el valor de 4, ya que constituye la cantidad de bytes que contiene la cabecera del paquete de transporte.

Capítulo 4

Pruebas y Resultados

En este capítulo se ejecutarán pruebas de funcionamiento del Analizador de Paquetes Transport Stream. Se han seleccionado tres archivos de prueba:

- espetv.ts
- espe_sek.ts
- mux1-cp.ts ⁶

4.1. Pruebas de Funcionamiento del A.P.T.S. con MATLAB

Para comprobar el funcionamiento del A.P.T.S. con MATLAB se realizarán casos de prueba para Apertura de Archivos y Búsqueda de Paquetes de Transporte.

4.1.1. Apertura de Archivos con MATLAB

⁶Descargado de <http://www.pjdaniel.org.uk/mpeg/>

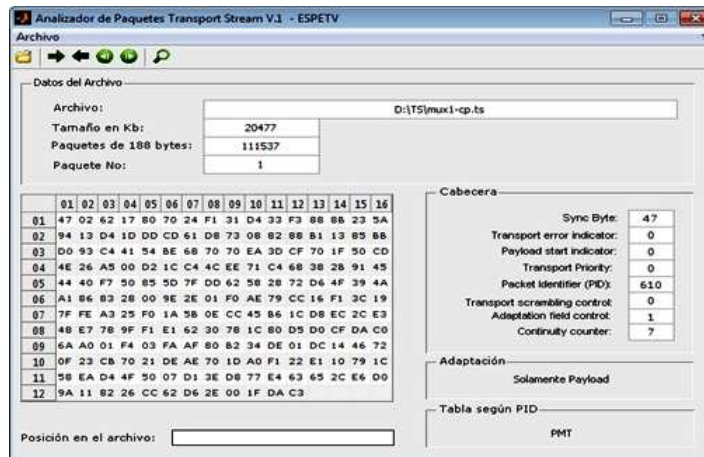


Fig. 4.1: Caso de prueba apertura de archivos con MATLAB - mux1-cp.ts

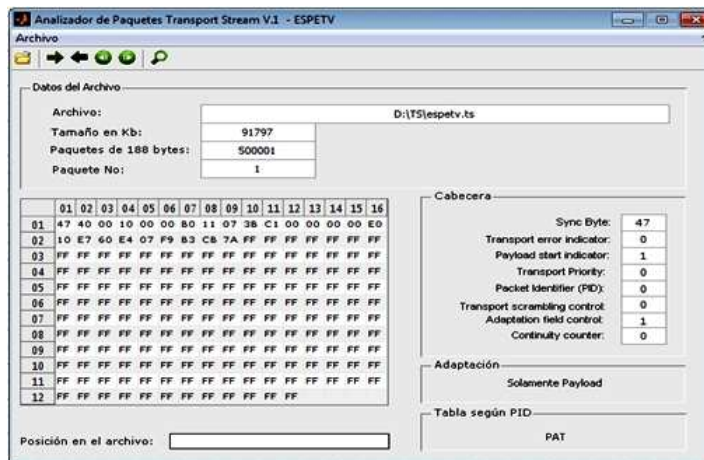


Fig. 4.2: Caso de prueba apertura de archivos con MATLAB - espetv.ts

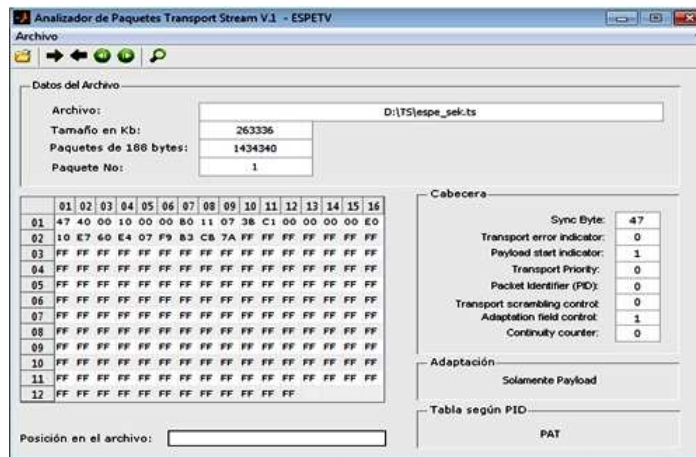


Fig. 4.3: Caso de prueba apertura de archivos con MATLAB - espe_sek.ts

En la tabla 4.1 se indican los datos obtenidos de cada archivo

Archivo	Tamaño en kbytes	Paquetes de 188 bytes	Paquete N°
mux1-cp.ts	20477	111537	1
espetv.ts	91797	500001	1
espe_sek.ts	263336	1434340	1

Tabla 4.1: Datos obtenidos en el caso de prueba apertura de archivos con MATLAB

4.1.2. Búsqueda de Paquetes de Transporte con MATLAB

En este caso de prueba se tomarán las siguientes consideraciones:

- Para el archivo mux1-cp.ts se buscará el paquete de transporte N° 100000.
- Para el archivo espetv.ts se buscará el paquete de transporte N° 479837.
- Para el archivo espe_sek.ts se buscará el último paquete de transporte.

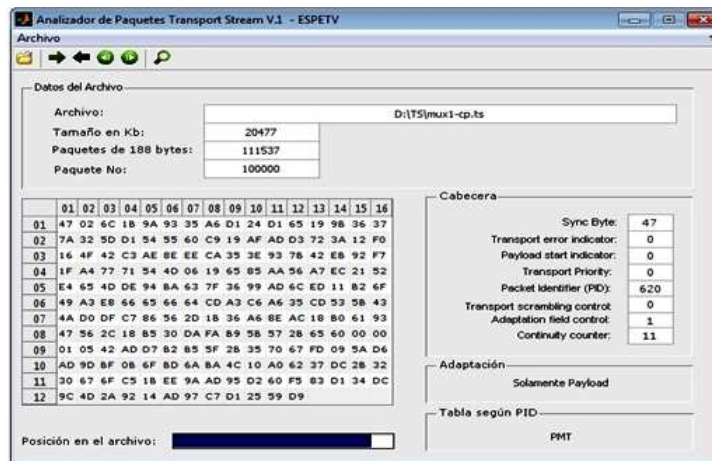


Fig. 4.4: Caso de prueba búsqueda de paquetes de transporte con MATLAB - mux1-cp.ts

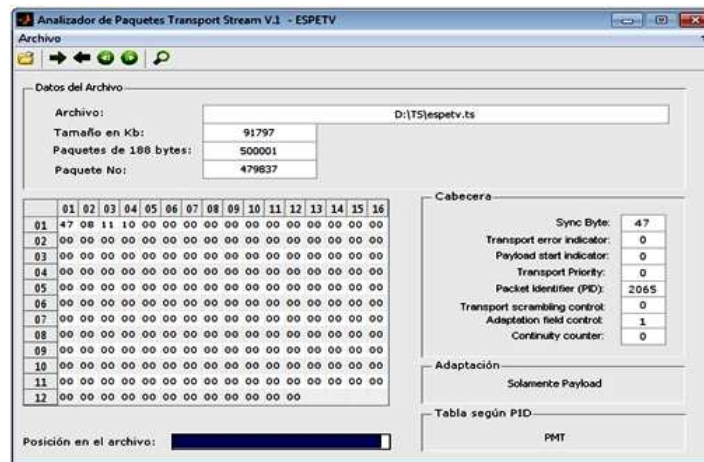


Fig. 4.5: Caso de prueba búsqueda de paquetes de transporte con MATLAB - espetv.ts

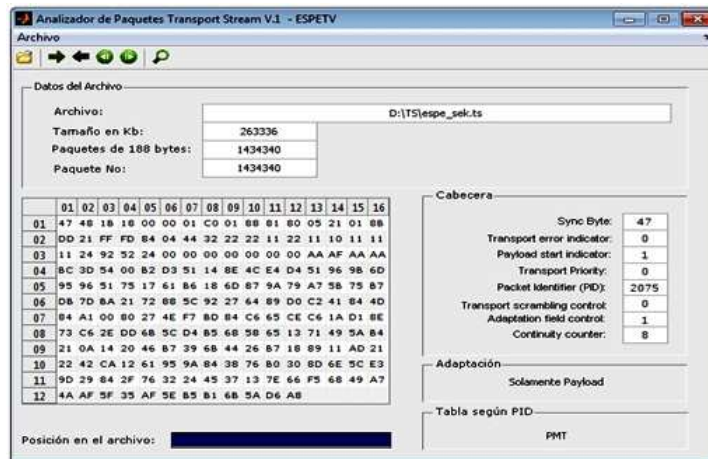


Fig. 4.6: Caso de prueba búsqueda de paquetes de transporte con MATLAB - espe_sek.ts

En la tabla 4.2 se indican los datos obtenidos de cada archivo

Archivo	Paquete N°
mux1-cp.ts	100000
espetv.ts	479837
espe_sek.ts	1434340

Tabla 4.2: Datos obtenidos en el caso de prueba búsqueda de paquetes de transporte con MATLAB

4.2. Pruebas de Funcionamiento del A.P.T.S. con Java

Para comprobar el funcionamiento del A.P.T.S. con Java se realizarán casos de prueba para Apertura de Archivos, Búsqueda de Paquetes de Transporte, Generación de Reporte PID y Generación de Reporte de Cabecera.

4.2.1. Apertura de Archivos con Java

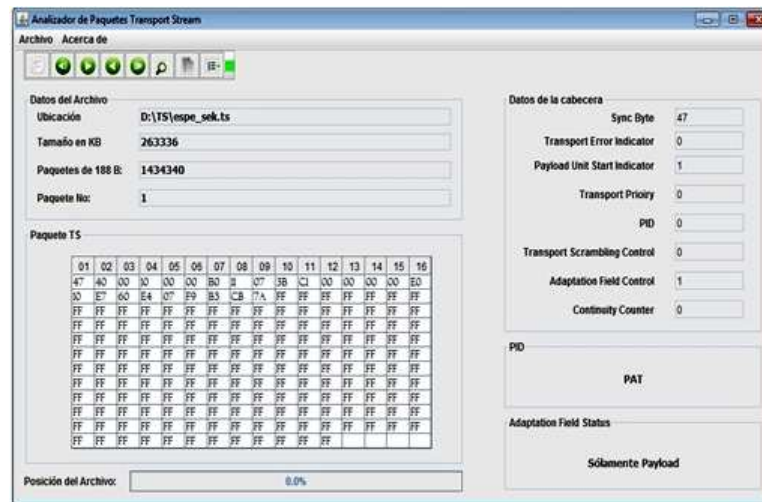


Fig. 4.9: Caso de prueba apertura de archivos con Java - espe_sek.ts

En la tabla 4.3 se indican los datos obtenidos de cada archivo

Archivo	Tamaño en kbytes	Paquetes de 188 bytes	Paquete N°
mux1-cp.ts	20477	111537	1
espetv.ts	91797	500001	1
espe_sek.ts	263336	1434340	1

Tabla 4.3: Datos obtenidos en el caso de prueba apertura de archivos con Java

4.2.2. Búsqueda de Paquetes de Transporte con Java

En este caso de prueba se tomarán las siguientes consideraciones:

- Para el archivo mux1-cp.ts se buscará el paquete de transporte N° 65000.
- Para el archivo espetv.ts se buscará el último paquete de transporte.
- Para el archivo espe_sek.ts se buscará el paquete de transporte N° 1357000.

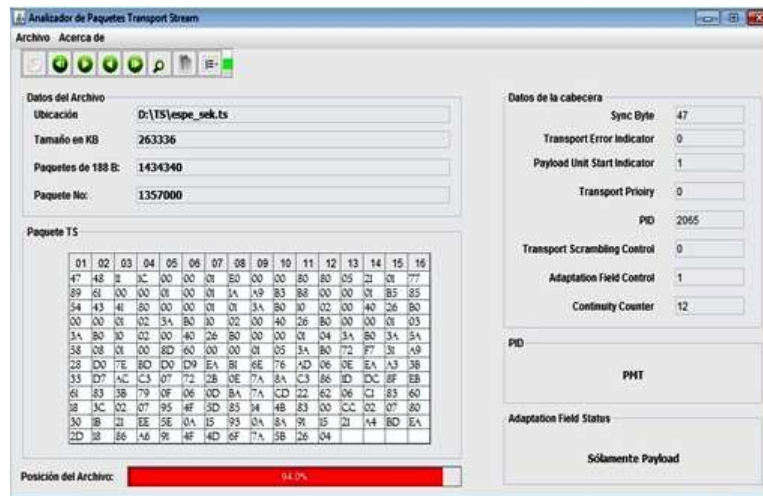


Fig. 4.12: Caso de prueba búsqueda de paquetes de transporte con Java - espe_sek.ts

En la tabla 4.4 se indican los datos obtenidos de cada archivo

Archivo	Paquete N°
mux1-cp.ts	65000
espetv.ts	500001
espe_sek.ts	1357000

Tabla 4.4: Datos obtenidos en el caso de prueba búsqueda de paquetes de transporte con Java

4.2.3. Generación de Reporte PID con Java

A continuación se indicará de manera gráfica los reportes PID generados para los tres archivos .ts de prueba.

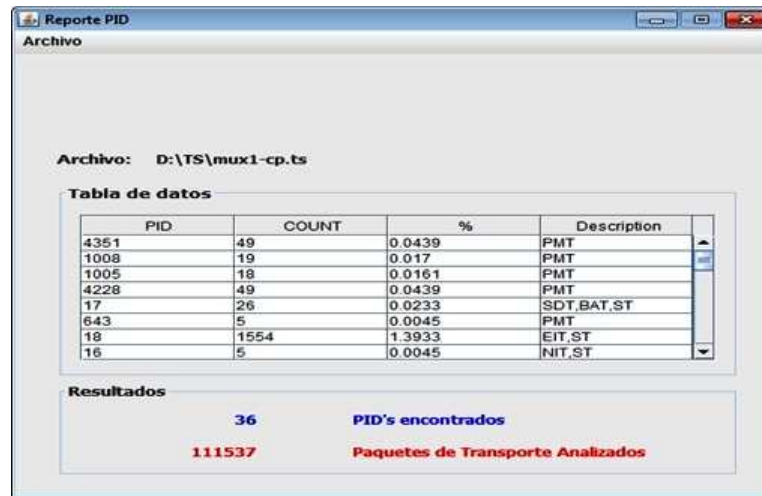


Fig. 4.13: Caso de prueba generación de reporte PID con Java - mux1-cp.ts



Fig. 4.14: Caso de prueba generación de reporte PID con Java - espetv.ts



The screenshot shows a window titled "Reporte PID" with a menu bar containing "Archivo". Below the menu bar, it displays "Archivo: D:\TS\espe_sek.ts". A table titled "Tabla de datos" contains the following information:

PID	COUNT	%	Description
8191	1176340	82.0126	Paquete Nulo
2065	199244	13.891	PMT
1031	721	0.0503	PMT
0	721	0.0503	PAT
2004	47878	3.338	PMT
17	145	0.0101	SDT_BAT_ST
16	145	0.0101	NIT_ST
2075	9001	0.6275	PMT

Below the table, the "Resultados" section shows:

- 9 PID's encontrados
- 1434340 Paquetes de Transporte Analizados

Fig. 4.15: Caso de prueba generación de reporte PID con Java - espe_sek.ts

4.2.4. Generación de Reporte de Cabecera con Java

En este caso de prueba se tomarán las siguientes consideraciones:

- Para el archivo mux1-cp.ts se generará un reporte de cabecera del paquete de transporte 1 al 1000.
- Para el archivo espetv.ts se generará un reporte de cabecera del paquete de transporte 1 al 2500.
- Para el archivo espe_sek.ts se generará un reporte de cabecera del paquete de transporte 1 al 4000 .

En las siguientes figuras se puede apreciar gráficamente los resultados obtenidos en este caso de prueba.

Reporte de Cabecera

Archivo

Resultados

Archivo: D:\T5mux1-cp.ts

Paquete Inicial: 1 Paquetes procesados: 1000 Elemento(s) encontrados: 1000

Paquete Final: 1000

Paq	Sync Byte	T.E.I	P.U.S.I	TR.P	PID	T.S.C	A.F.C	C.C	Descripción
1	47	0	0	0	610	0	1	7	PMT
2	47	0	0	0	620	0	1	5	PMT
3	47	0	0	0	640	0	1	7	PMT
4	47	0	0	0	600	0	1	8	PMT
5	47	0	1	0	650	0	1	5	PMT
6	47	0	0	0	610	0	1	8	PMT
7	47	0	0	0	600	0	1	9	PMT
8	47	0	0	0	620	0	1	6	PMT

Filtro

PAT
 TSDT
 NIT,ST
 EIT,ST
 TDT,TOT,ST
 Paquete Nulo
 Total
 CAT
 Reservados
 SDT,BAT,ST
 RST,ST
 Sincronización de Red
 PMT

Fig. 4.16: Caso de prueba generación de reporte de cabecera con Java - mux1-cp.ts

Reporte de Cabecera

Archivo

Resultados

Archivo: D:\T5esptv.ts

Paquete Inicial: 1 Paquetes procesados: 2500 Elemento(s) encontrados: 2500

Paquete Final: 2500

Paq	Sync Byte	T.E.I	P.U.S.I	TR.P	PID	T.S.C	A.F.C	C.C	Descripción
1	47	0	1	0	0	0	1	0	PAT
2	47	0	1	0	1031	0	1	0	PAT
3	47	0	1	0	2001	0	1	0	PAT
4	47	0	1	0	2004	0	1	0	PAT
5	47	0	1	0	17	0	1	0	SDT,BAT,ST
6	47	0	1	0	16	0	1	0	NIT,ST
7	47	0	0	0	2065	0	2	0	PAT
8	47	0	0	0	1811	0	1	0	Paquete Nulo

Filtro

PAT
 TSDT
 NIT,ST
 EIT,ST
 TDT,TOT,ST
 Paquete Nulo
 Total
 CAT
 Reservados
 SDT,BAT,ST
 RST,ST
 Sincronización de Red
 PMT

Fig. 4.17: Caso de prueba generación de reporte de cabecera con Java - esptv.ts

Reporte de Cabecera

Archivo

Resultados

Archivo: D:\T\Sespe_sek.ts

Paquete Inicial: 1 Paquetes procesados: 4000 Elemento(s) encontrados: 4000

Paquete Final: 4000

Pag	SyncByte	T.EI	P.U.S.I	TR.P	PID	T.S.C	A.F.C.	C.C.	Descripción
1	47	0	1	0	0	0	1	0	PAT
2	47	0	1	0	1031	0	1	0	PMT
3	47	0	1	0	17	0	1	0	SDT.BAT.ST
4	47	0	1	0	16	0	1	0	NIT.ST
5	47	0	1	0	2001	0	1	0	PMT
6	47	0	1	0	2004	0	1	0	PMT
7	47	0	0	0	2055	0	2	0	PMT
8	47	in	in	in	8191	0	1	0	Paquete Nulo

Filtro

PAT
 TSOT
 NIT.ST
 EIT.ST
 TOT.TOT.ST
 Paquete Nulo
 Total

CAT
 Reservados
 SDT.BAT.ST
 RST.ST
 Sincronización de Red
 PMT

Fig. 4.18: Caso de prueba generación de reporte de cabecera con Java - espe_sek.ts

4.3. Rendimiento del A.P.T.S. en la generación de reportes

El propósito de esta sección es comparar el tiempo de ejecución de la generación de reportes PID y de Cabecera en 2 distintos escenarios:

- a) Con un mismo Sistema Operativo empleando ordenadores con distintas características.
- b) Con distintos Sistemas Operativos.

4.3.1. Generación de Reportes con un mismo Sistema Operativo

Para este caso de prueba el Sistema Operativo a emplearse será Microsoft Windows 7 y los ordenadores tendrán las siguientes características:

Ordenador	Procesador	Memoria RAM
PC1	Pentium(R) Dual Core 2,50 GHz	4 GB
PC2	Intel(R) Atom(TM) 1,67 GHz	2 GB
PC3	Intel(R) Core(TM) i3 2.13 GHz	4 GB

Tabla 4.5: Características de los Ordenadores para los casos de prueba

Los resultados de los tiempos de ejecución del Reporte PID se detallan en la tabla 4.6 y en la figura 4.19.

Archivo	t.e. PC1 [s]	t.e. PC2 [s]	t.e. PC3 [s]
mux1-cp.ts	3,5	11	3
espetv.ts	8,6	60	5,30
espe_sek.ts	17	176	15

Tabla 4.6: Detalle de los tiempos ejecución para la generación de Reportes PID - MS Windows 7

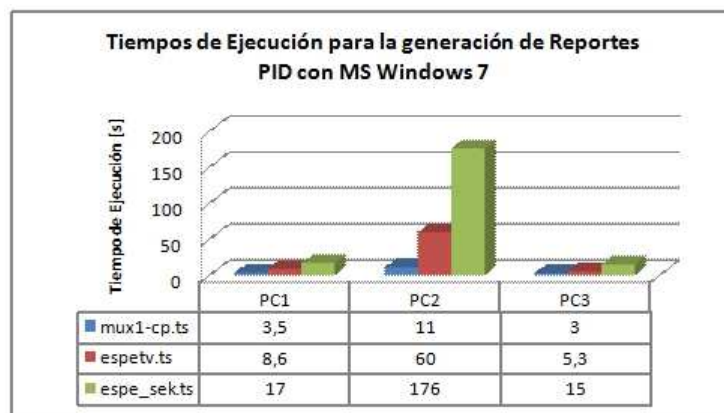


Fig. 4.19: Tiempos de ejecución para la generación de Reportes PID - MS Windows 7

Los tiempos de generación de reportes de cabecera fueron obtenidos en base al caso de prueba de la subsección 4.2.4.

Los resultados de los tiempos de ejecución del Reporte de Cabecera se detallan en la tabla 4.7 y en la figura 4.22.

Reporte de Cabecera	t.e. PC1 [s]	t.e. PC2 [s]	t.e. PC3 [s]
Reporte - mux1-cp.ts	29,63	101	18
Reporte - espetv.ts	101	225	90
Reporte espe_sek.ts	213	350	200

Tabla 4.7: Detalle de los tiempos de ejecución para la generación de Reportes de Cabecera - MS Windows 7

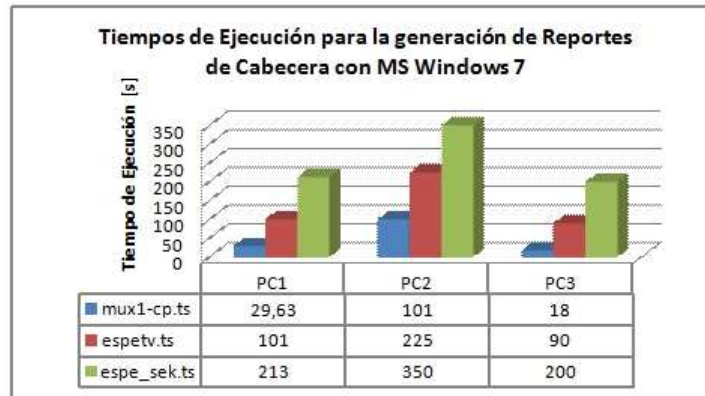


Fig. 4.20: Tiempos de ejecución para la generación de Reportes de Cabecera - MS Windows 7

4.3.2. Generación de Reportes con distintos Sistemas Operativos

Para este caso de prueba la plataforma Open Source empleada es Ubuntu 12.04 y el ordenador utilizado corresponde a PC3, mismo cuyas características se indican en la tabla 4.5.

En las tablas 4.8 y 4.9 se detallan los tiempos de ejecución en la generación tanto de los reportes PID como los de cabecera con Ubuntu 12.04. Posteriormente se muestra gráficamente una comparación de los resultados obtenidos con MS Windows 7 y Ubuntu 12.04.

Archivo	t.e. PC3 Ubuntu 12.04 [s]
mux1-cp.ts	2,08
espetv.ts	6,81
espe_sek.ts	14,23

Tabla 4.8: Detalle de los tiempos ejecución para la generación de Reportes PID - Ubuntu 12.04

Reporte de Cabecera	t.e. PC3 Ubuntu 12.04 [s]
Reporte - mux1-cp.ts	11,33
Reporte - espetv.ts	18,09
Reporte espe_sek.ts	31,6

Tabla 4.9: Detalle de los tiempos de ejecución para la generación de Reportes de Cabecera - Ubuntu 12.04

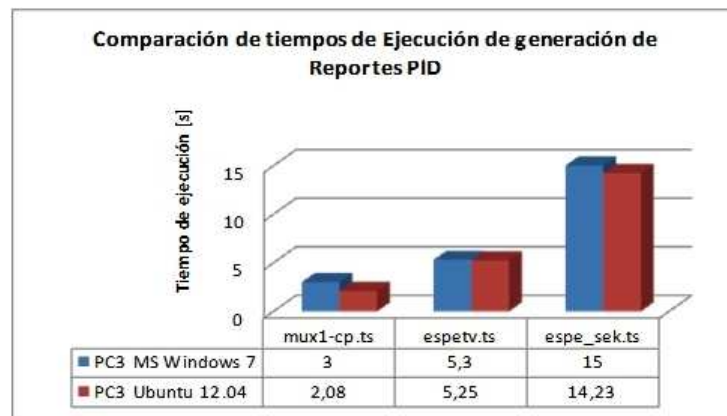


Fig. 4.21: Comparación de resultados para Reportes PID - Windows 7 vs Ubuntu 12.04

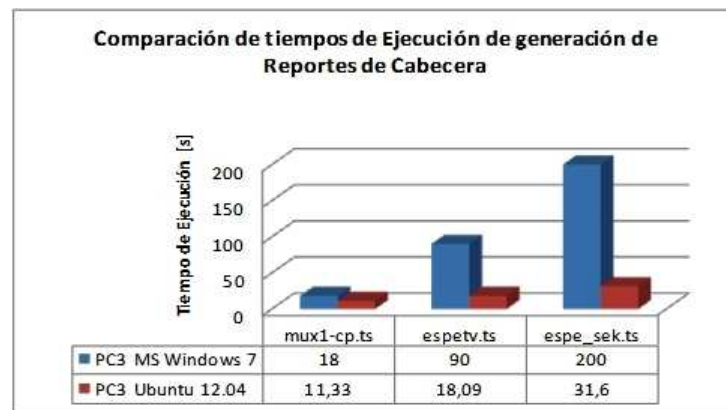


Fig. 4.22: Comparación de resultados para Reportes de Cabecera - Windows 7 vs Ubuntu 12.04

4.4. Transición

4.4.1. Entrega

Adjunto a esta documentación se entregará el respectivo Manual de Usuario así como un CD en el que se encuentra la aplicación y todo el software necesario para que el Analizador de Paquetes Transport Stream pueda funcionar.

El CD contiene los siguiente:

- **Aplicación**

- a) Archivo ejecutable del A.P.T.S en formato .jar

- **Código Fuente**

- a) Una carpeta denominada analizadorts_matlab que contendrá todo el código fuente y los recursos necesarios para el funcionamiento del A.P.T.S. con MATLAB.

- b) Una carpeta denominada analizadorts_java que contendrá todo el código fuente y los recursos necesarios para el funcionamiento del A.P.T.S. con Java.

- **Software necesario para ejecutar la aplicación**

- a) jdk7 para plataformas Windows y Linux.

- b) Netbeans IDE 7.2 para plataformas Windows y Linux.

- **Archivos de prueba**

- a) mux1-cp.ts
- b) espetv.ts
- c) espe_sek.ts

- **Documentación Adicional**

- a) Ejecución de un archivo .jar en Ubuntu 12.04 que corresponde Apéndice A.
- b) Indicaciones para ampliar la memoria virtual de Java que corresponde al Apéndice B.
- c) Manual de Usuario que corresponde al Apéndice C.
- d) MPEG-2 TS and Sections que corresponde al Apéndice D.

Capítulo 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- El Analizador de Paquetes Transport Stream se desarrolló e implementó con herramientas Open Source tales como Netbeans 7.2, jdk7 , las cuales se ejecutan bajo el Sistema Operativo Linux, brindando así la ventaja de reducir costos en licencias, tanto para el desarrollo como para el uso de las aplicaciones.
- El uso de la norma IEEE-830-1998 permitió recopilar la información que interviene en los procesos del Analizador de Paquetes Transport Stream, teniendo como resultado los requerimientos funcionales necesarios para una correcta ejecución del mismo.
- *Unified Modeling Language* (UML) es un marco de referencia que sirve de guía para el desarrollo de un software y debido a su flexibilidad permitió adaptar sus modelos y diagramas para la construcción del Analizador de Paquetes Transport Stream.
- El alcance de este proyecto se limitó a obtener y desplegar información de datos pertenecientes a las cabeceras de los paquetes de transporte contenidos en un archivo de extensión .ts.
- De los casos de prueba efectuados para comprobar el funcionamiento del A.P.T.S se puede concluir lo siguiente:
 - a) La aplicación garantiza al usuario que sólo trabajará con archivos de extensión .ts.

- b) La apertura de archivos .ts y la búsqueda de paquetes de transporte presenta una respuesta de ejecución instantánea tanto en MATLAB como en Netbeans.
 - c) Al comparar el rendimiento del A.P.T.S en la generación de reportes bajo el Sistema Operativo Windows 7 con tres ordenadores distintos, PC3 (ver cuadro 4.5) fue el que menores tiempos de ejecución presentó. Esto se debe a que este dispositivo tiene mejores prestaciones y capacidad de procesamiento que los otros dos ordenadores de prueba.
 - d) Al generar los reportes PID con Ubuntu 12.04, los tiempos de ejecución son muy similares a los obtenidos con Windows 7.
 - e) Al generar los reportes de cabecera con Ubuntu 12.04, los tiempos de ejecución fueron mucho menores a los obtenidos con Windows 7. Al realizar una comparación de resultados se puede deducir que en el reporte de 4000 paquetes, con Windows 7 se procesaron 20 paquetes/segundo mientras que con Ubuntu 12.04 se procesaron 127 paquetes/segundo, por tanto la velocidad de ejecución con este último fue 6 veces mayor.
- La elaboración de las funcionalidades del A.P.T.S fue posible gracias a las conversiones numéricas implementadas tanto en MATLAB como en Java.

5.2. Recomendaciones

- Es necesario adoptar buenas prácticas de documentación de ingeniería de software en el Analizador de Paquetes Transport Stream para que se pueda realizar mantenimiento y actualizaciones de manera rápida y adecuada.
- Se debe extender la funcionalidad del A.P.T.S. para que puedan trabajar con archivos de otras extensiones que manejen el estándar MPEG-2.
- En base a los datos contenidos en los 188 bytes de cada paquete de transporte se puede obtener una mayor cantidad de información aparte de los datos de la cabecera y campo de adaptación tales como estructura de las secciones, estructura de las tablas del archivo .ts e identificación de scripts tanto de audio como de video.
- En una versión futura del A.P.T.S. se debe crear una funcionalidad adicional que le permita interactuar con un módulo de Hardware a través de una interfaz

de comunicación para realizar el análisis de los paquetes de transporte en tiempo real.

- Al querer implementar una nueva funcionalidad en el A.P.T.S., tenga siempre en cuenta la cantidad de información que va a procesar para no disminuir el desempeño requerido del mismo.
- Se recomienda utilizar un ordenador de altas prestaciones y plataformas Open Source para manejar el A.P.T.S., ya que ésto permitirá tener gran eficiencia en la generación de reportes.

Bibliografía

- [1] **ISO/IEC 13818-1:2007**
<http://mumudvb.braice.net/mumudrupal/sites/default/files/iso13818-1.pdf>
- [2] **Norma CCIR 601 4:2:2**
<http://cursotecnicoav.wikispaces.com/file/view/norma+CCIR+601+422.pdf>
- [3] **AES/EBU**
http://en.wikiaudio.org/AES_EBU
- [4] **Netbeans,**
<http://www.netbeans.org/>
- [5] Morales, Herón. **MATLAB 7 para Ciencias e Ingeniería.** Editorial Megabyte. Primera Edición. Marzo 2005.
- [6] Noriega, Adrián. **Programando en Java 2.** Editorial Megabyte. Primera Edición. Febrero 2007.
- [7] **StarUML,**
<http://staruml.sourceforge.net/en/>
- [8] **Conversiones Numéricas en Java,**
<http://gervy.wordpress.com/2009/02/05/conversiones-numericas-en-java/>
- [9] Weitzenfeld, Alfredo. **Ingeniería de Software Orientada a Objetos con UML, Java e Internet.** Editorial Thomson. Segunda Edición. Septiembre 2004.
- [10] **MATLAB,**
<http://www.mathworks.com/products/matlab/>
- [11] **Archivos de Acceso Aleatorio con Java,**
<http://es.slideshare.net/jent46/archivos-aleatorios-en-java-presentation>

-
- [12] **Número de ocurrencias en un array en Java**,
<http://lineadecodigo.com/java/numero-de-ocurrencias-en-un-array/>
- [13] **IEEE 830**,
IEEE Std.830-1998,
<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- [14] **Formato de tesis para L^AT_EX**,
<http://navarroj.com/latex/ezthesis.html>
- [15] **MPEG-2 Transport Stream packet analyser**,
<http://www.pjdaniel.org.uk/mpeg/>
- [16] Delgado, Alejandro. (2010). Flujos de Programa y Transporte MPEG-2.
<http://www94.etc.upm.es/tsmpeg2d.pdf>