



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

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

**TEMA: “ANÁLISIS DE CALIDAD DE IMÁGENES PRE-  
PROCESADAS MEDIANTE TRANSFORMACIONES DE COLOR  
REVERSIBLES EN UN ESCENARIO REAL TVI”**

**AUTOR: ARMENDÁRIZ ALDÁS, KATHERINE ALEJANDRA**

**DIRECTOR: ACOSTA BUENAÑO, FREDDY ROBERTO**

**SANGOLQUI**

**2018**



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, *"ANÁLISIS DE CALIDAD DE IMÁGENES PRE-PROCESADAS MEDIANTE TRANSFORMACIONES DE COLOR REVERSIBLES EN UN ESCENARIO REAL TVT"*, realizado por la señorita Katherine Alejandra Armendáriz Aldás, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo que cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas – ESPE, por lo tanto me permito acreditarla y autorizar a la señorita Katherine Alejandra Armendáriz Aldás, para que lo sustente públicamente.

Sangoquí, febrero 2018

Alientamente:



Ing. Acosta Buenafío Freddy Roberto

DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, Armendáriz Aldás Katherine Alejandra, con cédula de identidad N° 1725910010, declaro que este trabajo de titulación "*ANÁLISIS DE CALIDAD DE IMÁGENES PRE-PROCESADAS MEDIANTE TRANSFORMACIONES DE COLOR REVERSIBLES EN UN ESCENARIO REAL TVI*" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangoquí, febrero 2018



---

Katherine Alejandra Armendáriz Aldás

C.C. 172591001-0



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

**AUTORIZACIÓN**

Yo, Armendáriz Aldás Katherine Alejandra, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "*ANÁLISIS DE CALIDAD DE IMÁGENES PRE-PROCESADAS MEDIANTE TRANSFORMACIONES DE COLOR REVERSIBLES EN UN ESCENARIO REAL TVT*" cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, febrero 2018

Katherine Alejandra Armendáriz Aldás

C.C. 1725910010

## DEDICATORIA

A Dios, por haberme brindado la salud para poder culminar con éxito una etapa más de mi vida y por las respectivas bendiciones día a día.

A mis padres, Juan y Norma quienes han sido mi fortaleza y mi apoyo siempre, en todas las decisiones a lo largo de mi vida.

A mi madre Norma por brindarme siempre los mejores consejos que me han sabido guiar como una persona de bien en cada etapa de mi vida.

A mi padre Juan que a través del esfuerzo diario nos ha sabido sacar adelante y nos ha hecho fortalecer siempre los lazos familiares a través de los años, tanto a mi como a mis hermanos.

A mi hermana Karina que siempre ha sabido brindarme una sonrisa o un consejo cuando más lo he necesitado.

A mi hermano Juan Carlos por haber sembrado en mi la curiosidad de seguir esta hermosa carrera, es por el que ahora cumplo esta meta trazada años atrás.

Por todo lo que me han sabido brindar a lo largo de todos estos años, este logro va por y para ustedes mi Familia

## AGRADECIMIENTO

Como principal y siempre a la cabeza de todo, quiero agradecer a Dios porque sin el ninguna de las cosas planteadas como metas serían realizables.

De manera especial, agradezco a mis padres por nunca haber dejado de confiar en mí y en mis decisiones, pase lo que pase siempre estuvieron a mí, como mis ángeles guardianes.

A mis hermanos ya que, como siempre se los he dicho, detrás de ellos yo aprendí a caminar, siempre han sido mi ejemplo a seguir.

A mi cuñada y mi cuñado que igual de una u otra manera me han sabido brindar su apoyo y sus consejos cuando más lo he necesitado.

Con alegría en mi corazón porque son lo más hermoso que Dios me ha dado, a mis sobrin@s que con una sonrisa siempre han sabido alegrar los días más difíciles.

A mi persona favorita, Alexis Vizuite, quien ha sabido apoyarme siempre y ha estado dispuesto a ayudarme cuando lo he necesitado, en las buenas y en las malas.

A mis maestros no me van a alcanzar las palabras para poder explicar lo infinitamente agradecida que estoy con ellos ya que fueron la fuente de sabiduría para ahora llegar a este momento.

A mi querida Universidad de las Fuerzas Armadas ESPE, de la cual me llevo gratos recuerdos y experiencias vividas guardadas ahora en mi corazón y en mi memoria

## ÍNDICE DE CONTENIDOS

<b>CERTIFICACIÓN.....</b>	<b>ii</b>
<b>AUTORÍA DE RESPONSABILIDAD.....</b>	<b>iii</b>
<b>AUTORIZACIÓN.....</b>	<b>iv</b>
<b>DEDICATORIA.....</b>	<b>v</b>
<b>AGRADECIMIENTO.....</b>	<b>vi</b>
<b>ÍNDICE DE CONTENIDOS.....</b>	<b>vii</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>x</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>xii</b>
<b>RESUMEN.....</b>	<b>xv</b>
<b>ABSTRACT.....</b>	<b>xvi</b>
<b>CAPÍTULO I.....</b>	<b>1</b>
<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1.    Antecedentes.....	1
1.2.    Justificación e Importancia.....	4
1.3.    Alcance del Proyecto.....	5
1.4.    Objetivos.....	7
1.4.1.    Objetivo General.....	7
1.4.2.    Objetivos Específicos.....	7
<b>CAPÍTULO II.....</b>	<b>9</b>
<b>2. MARCO TEÓRICO.....</b>	<b>9</b>
2.1.    Televisión Digital Terrestre (TDT).....	9
2.2.    Sistema de televisión digital (ISDB-Tb).....	11
2.3.    Televisión digital interactiva (TVDi).....	13
2.3.1.    Servicios interactivos.....	14
2.3.2.    Aplicaciones interactivas según los servicios.....	15
2.3.3.    Tipos de interactividad.....	16
2.3.4.    Taxonomía de aplicaciones interactivas.....	17
2.4.    Middleware GINGA NCL.....	18
2.5.    Procesamiento de imágenes digitales.....	19
2.5.1.    Etapas del procesamiento de imágenes.....	20

2.5.2.	Procesamiento en el dominio espacial.....	21
2.5.2.1.	Entorno 1 x 1: Operaciones pixel a pixel.....	21
2.5.2.2.	Entorno n x n, n > 1: Operaciones de contorno (filtros).....	27
2.5.3.	Procesamiento en el dominio de la frecuencia.....	29
2.5.4.	Segmentación de imágenes.....	32
2.5.5.	Áreas de aplicación.....	33
2.6.	Tipos de imágenes.....	34
2.7.	Calidad en imágenes.....	40
2.8.	Métodos para análisis de calidad.....	43
2.8.1.	Métodos subjetivos.....	43
2.8.2.	Métodos objetivos.....	45
2.9.	Infraestructura de la TDT.....	48
2.9.1.	Servidor de aplicación y contenidos.....	48
2.9.2.	Servidor de playout.....	48
2.9.3.	Set top box.....	49
<b>CAPÍTULO III.....</b>		<b>50</b>
<b>3. DISEÑO E IMPLEMENTACIÓN.....</b>		<b>50</b>
3.1.	Técnica mejorada de transmisión de imágenes segura a través de imágenes en mosaico mediante transformaciones de color casi reversibles.....	50
3.2.	Diseño de la aplicación en Ginga.....	57
3.3.	Generación del Transport Stream.....	72
3.3.1.	Transport Stream (TS).....	72
3.3.2.	Estructura del Transport Stream.....	72
3.3.3.	Generación del TS.....	76
3.4.	Transmisión en Tiempo Real.....	92
<b>CAPÍTULO IV.....</b>		<b>102</b>
<b>4. ANÁLISIS DE RESULTADOS.....</b>		<b>102</b>
4.1.	Análisis MOS.....	102
4.2.	Análisis RMSE.....	113
4.3.	Análisis MSSIM.....	117
<b>CAPÍTULO V.....</b>		<b>122</b>
<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>122</b>



5.1.	Conclusiones .....	122
5.2.	Recomendaciones .....	124
<b>6.</b>	<b>BIBLIOGRAFÍA.....</b>	<b>126</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Características técnicas principales del sistema ISDB-Tb</i> .....	13
<b>Tabla 2</b> <i>Áreas de aplicación del procesamiento de imágenes</i> .....	33
<b>Tabla 3</b> <i>Aspectos positivos y negativos del formato JPEG</i> .....	41
<b>Tabla 4</b> <i>Aspectos positivos y negativos del formato GIF</i> .....	42
<b>Tabla 5</b> <i>Aspectos positivos y negativos del formato PNG</i> .....	43
<b>Tabla 6</b> <i>Valores de calidad de ACR</i> .....	44
<b>Tabla 7</b> <i>Valores de calidad de DCR</i> .....	44
<b>Tabla 8</b> <i>Valores del campo de control de adaptación</i> .....	74
<b>Tabla 9</b> <i>PIDs usados en la transmisión de Tablas del Sistema</i> .....	75
<b>Tabla 10</b> <i>Tablas PSI principales para el sistema ISDB-Tb</i> .....	76
<b>Tabla 11</b> <i>Tablas SI para el sistema ISDB-Tb</i> .....	76
<b>Tabla 12</b> <i>Parámetros utilizados para mencoder</i> .....	80
<b>Tabla 13</b> <i>Parámetros para la codificación de video</i> .....	82
<b>Tabla 14</b> <i>Parámetros para la codificación de audio</i> .....	83
<b>Tabla 15</b> <i>Parámetros utilizados en la paquetización de video</i> .....	84
<b>Tabla 16</b> <i>Parámetros utilizados en la paquetización de video</i> .....	85
<b>Tabla 17</b> <i>Parámetros para la paquetización de audio</i> .....	86
<b>Tabla 18</b> <i>Parámetros para la paquetización de audio</i> .....	87
<b>Tabla 19</b> <i>Parámetros utilizados para la generación del carrusel de objetos</i> .....	90
<b>Tabla 20</b> <i>Parámetros para la multiplexación de archivos</i> .....	91
<b>Tabla 21</b> <i>Elementos de recepción</i> .....	94
<b>Tabla 22</b> <i>Opciones presentadas al usuario para el análisis MOS</i> .....	102
<b>Tabla 23</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=4</i> .....	105
<b>Tabla 24</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=8</i> .....	106
<b>Tabla 25</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=16</i> .....	106
<b>Tabla 26</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=32</i> .....	107
<b>Tabla 27</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=4</i> .....	108
<b>Tabla 28</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=8</i> .....	108
<b>Tabla 29</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=16</i> .....	109
<b>Tabla 30</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=32</i> .....	109
<b>Tabla 31</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=4</i> .....	110
<b>Tabla 32</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=8</i> .....	110

<b>Tabla 33</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=16</i> .....	111
<b>Tabla 34</b> <i>Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=32</i> .....	112
<b>Tabla 35</b> <i>Resultados obtenidos en el método RMSE con N=4</i> .....	115
<b>Tabla 36</b> <i>Resultados obtenidos en el método RMSE con N=8</i> .....	115
<b>Tabla 37</b> <i>Resultados obtenidos en el método RMSE con N=16</i> .....	116
<b>Tabla 38</b> <i>Resultados obtenidos en el método RMSE con N=32</i> .....	116
<b>Tabla 39</b> <i>Resultados obtenidos en el método MSSIM con N=4</i> .....	119
<b>Tabla 40</b> <i>Resultados obtenidos en el método MSSIM con N=8</i> .....	119
<b>Tabla 41</b> <i>Resultados obtenidos en el método MSSIM con N=16</i> .....	120
<b>Tabla 42</b> <i>Resultados obtenidos en el método MSSIM con N=32</i> .....	120

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Adición: a) Imagen original, b) Ruido, c) Imagen ruido aditivo.....	22
<b>Figura 2</b> Sustracción (diferencia entre imágenes): a) Imagen Rayos X, b) Imagen con contraste radiológico c) Diferencia entre imágenes .....	22
<b>Figura 3</b> Sustracción (detección de movimiento): a) Imagen uno, b) Imagen dos c) Diferencia entre imágenes y detección del movimiento realizado entre imagen uno e imagen dos. ....	22
<b>Figura 4</b> División: a) Imagen formada por una función perfecta y una función de sombreado, b) Función de sombreado conocida, c) Función formada por la función perfecta y la inversa de la función de sombreado conocida.....	23
<b>Figura 5</b> Producto: a) Radiografía dental, b) Máscara de regiones de interés, c) Producto. ....	23
<b>Figura 6</b> Transformaciones básicas de intensidad.....	24
<b>Figura 7</b> Transformación lineal (Imagen negativa).....	24
<b>Figura 8</b> Transformación logarítmica. ....	25
<b>Figura 9</b> Transformación inversa de log. ....	25
<b>Figura 10</b> Transformación de potencia-raíz: a) Imagen original, b) $n=1/2$ , c) $n=1/3$ , d) $n=1/4$ , e) $n=2$ . ....	25
<b>Figura 11</b> Transformación definida a trozos. ....	26
<b>Figura 12</b> Representación del histograma de una imagen.....	26
<b>Figura 13</b> Filtro estadístico ordenado mínimo y máximo. ....	28
<b>Figura 14</b> Filtro estadístico ordenado mediana: a) Imagen original, b) Imagen con ruido, c) Imagen aplicada filtro de mediana 3 x 3.....	29
<b>Figura 15</b> Traslación: a) Imagen simple, b) Espectro de Fourier sin desplazar, c) Espectro de Fourier desplazado hacia el centro.....	31
<b>Figura 16</b> Rotación .....	31
<b>Figura 17</b> Representación del logaritmo del espectro: a) Imagen original, b) Espectro de Fourier, c) Logaritmo del espectro a 8 bits, escala de grises de 256. ....	32
<b>Figura 18</b> Funcionalidad de un servidor playout.....	49
<b>Figura 19</b> Set top box.....	49
<b>Figura 20</b> Imágenes a ser procesadas.....	53
<b>Figura 21</b> Imagen portadora .....	54
<b>Figura 22</b> Ventana de resultado al correr el programa principal del algoritmo empleado .....	55
<b>Figura 23</b> Creación del archivo Dc1532.....	55
<b>Figura 24</b> Ventana de resultado al correr el programa de recuperación de imágenes del algoritmo empleado.....	56
<b>Figura 25</b> Creación del archivo .jpg .....	57
<b>Figura 26</b> Pantalla principal de Eclipse Mars.....	59
<b>Figura 27</b> Creación de un nuevo proyecto NCL.....	59
<b>Figura 28</b> Selección de nuevo proyecto NCL .....	60

<b>Figura 29</b> Configuración del nombre del proyecto.....	60
<b>Figura 30</b> Visualización de la correcta creación del nuevo proyecto NCL .....	61
<b>Figura 31</b> Creación de un documento nuevo dentro del proyecto creado.....	61
<b>Figura 32</b> Pantalla donde se escoge el tipo de documento a crear .....	62
<b>Figura 33</b> Configuración del nombre del documento .....	62
<b>Figura 34</b> Generación del código por default al crear el documento NCL.....	63
<b>Figura 35</b> Declaración de regiones a utilizarse .....	64
<b>Figura 36</b> Región r8 programada para la primera imagen en el carrusel de imágenes .....	64
<b>Figura 37</b> Declaración de descriptores a utilizarse.....	65
<b>Figura 38</b> Importación del conector base.....	65
<b>Figura 39</b> Declaración de los elementos multimedia.....	66
<b>Figura 40</b> Multimedia mverde programado para el botón verde.....	67
<b>Figura 41</b> Conector para mostrar el MENU principal .....	68
<b>Figura 42</b> Redimensionamiento de la pantalla al presionar un botón.....	68
<b>Figura 43</b> Programación para cada uno de los botones dependiendo de lo que se debe o no se debe visualizar en cada caso .....	69
<b>Figura 44</b> Programación para regresar al MENU principal .....	70
<b>Figura 45</b> Pantalla del MENU principal de la aplicación interactiva .....	71
<b>Figura 46</b> Pantalla visualizada al presionar el botón azul.....	71
<b>Figura 47</b> Formación de flujos de datos.....	72
<b>Figura 48</b> Cabecera TS.....	73
<b>Figura 49</b> Finalización del proceso de transformación.....	79
<b>Figura 50</b> Creación del archivo en formato AVI.....	80
<b>Figura 51</b> Propiedades del video en formato MP4 .....	81
<b>Figura 52</b> Finalización del proceso de codificación de video .....	81
<b>Figura 53</b> Creación del archivo ginga.m2v.....	82
<b>Figura 54</b> Finalización del proceso de codificación de audio .....	83
<b>Figura 55</b> Creación del archivo ginga.mp2.....	83
<b>Figura 56</b> Finalización del proceso de paquetización de video.....	84
<b>Figura 57</b> Creación del archivo gingavideo.pes .....	84
<b>Figura 58</b> Creación del archivo gingavideo.ts.....	85
<b>Figura 59</b> Creación del archivo gingaaudio.pes .....	86
<b>Figura 60</b> Creación del archivo gingaaudio.ts.....	87
<b>Figura 61</b> Creación del script gtable.py para las tablas PSI/SI .....	88
<b>Figura 62</b> Creación de los archivos de cada tabla en formato TS .....	88
<b>Figura 63</b> Creación del archivo para paquetes nulos .....	89
<b>Figura 64</b> Carpeta que contiene todo lo relacionado a la aplicación en Ginga.....	89
<b>Figura 65</b> Creación del archivo TS de la aplicación interactiva .....	90
<b>Figura 66</b> Cambio en el programa principal para la correcta generación del TS de la aplicación en Ginga.....	90
<b>Figura 67</b> Finalización del proceso de multiplexación .....	91
<b>Figura 68</b> Creación del archivo pruebaGINGA.ts.....	91
<b>Figura 69</b> Creación del archivo gingaFINAL.fixed.ts .....	92
<b>Figura 70</b> Tarjeta moduladora DTA-115 .....	93

<b>Figura 71</b> Instalación de la tarjeta.....	93
<b>Figura 72</b> Interfaz gráfica StreamXpress y reconocimiento de tarjeta moduladora DTA-115.....	95
<b>Figura 73</b> Búsqueda en el Explorador.....	96
<b>Figura 74</b> Elección del TS.....	96
<b>Figura 75</b> Configuración de los parámetros de transmisión .....	97
<b>Figura 76</b> Transmisión del TS .....	97
<b>Figura 77</b> Contenido de la aplicación interactiva.....	98
<b>Figura 78</b> Menú del set top box y selección de la opción Canales.....	98
<b>Figura 79</b> Submenú al seleccionar la opción Canales .....	99
<b>Figura 80</b> Confirmación necesitada para iniciar la sintonización automática.....	99
<b>Figura 81</b> Contraseña para iniciar el proceso .....	100
<b>Figura 82</b> Búsqueda de canales, uno encontrado.....	100
<b>Figura 83</b> Sintonización exitosa del canal en ambas TVs.....	101
<b>Figura 84</b> Recepción del servicio de TV Digital y la aplicación interactiva .....	101
<b>Figura 85</b> Implementación del MOS con el grupo de estudiantes .....	105
<b>Figura 86</b> Comparación del análisis MOS dependiendo del tamaño del bloque y el tipo de receptor.....	112
<b>Figura 87</b> Comparación del análisis RMSE dependiendo del tamaño del bloque ..	117
<b>Figura 88</b> Comparación del análisis MSSIM dependiendo del tamaño del bloque	121

## **RESUMEN**

El presente proyecto de titulación comprende el análisis de calidad de imágenes pre-procesadas mediante transformaciones de color reversibles en un escenario real TVi en el que se realizará la transmisión a partir de un transmisor y un receptor adecuados para obtener la óptima visualización de la imagen por el usuario, para esto se realizará una aplicación interactiva a través de GINGA NCL, por medio de la misma se observará las imágenes en diferentes tamaños y regiones para que posteriormente se llegue a obtener el TS que se va a transmitir. El análisis de calidad se lo realizará por medio de tres métodos, el primero es el método subjetivo MOS que nos permite ponderar resultados para obtener una puntuación media, el segundo es el RMSE que consiste en la desviación estándar de los errores de predicción, y por último el análisis a través de MSSIM. SSIM es una métrica que se basa en que las estructuras de los objetos de las escenas son independientes de la iluminación, el promedio de los valores SSIM se conoce como MSSIM. Se tomará en cuenta varios tipos de receptores, es decir, diferentes televisores (Led, LCD, Rayos Catódicos) ya que de esta manera se puede también comparar dependiendo de las características de la pantalla en la que se proyecta las imágenes obteniendo así un criterio óptimo por parte del usuario para realizar un análisis más concreto en torno al proceso por el cual se obtuvo la imagen secreta a través de la imagen mosaico.

### **Palabras Clave:**

- **TELEVISIÓN INTERACTIVA**
- **GINGA**
- **CALIDAD EN IMÁGENES PRE-PROCESADAS**
- **ECLIPSE**

## **ABSTRACT**

The present degree Project includes the quality analysis of pre-processed images through reversible color transformations in a real TVi scenario in which the transmission will be made from a suitable transmitter and receiver to obtain the optimum visualization of the image by the user, for this an interactive application will be made through GINGA NCL, through it the images in different sizes and regions will be observed so that the TS to be transmitted can be obtained later. The quality analysis will be carried out by means of three methods, the first is the subjective MOS method that allows us to weight results to obtain an average score, the second is the RMSE that consists of the standard deviation of prediction errors, and Last analysis through MSSIM. SSIM is a metric that is based on the fact that the structures of the objects of the scenes are independent of the illumination, the average of the SSIM values is known as MSSIM. Several types of receivers will be taken into account, that is, different televisions (Led, LCD, Cathode Rays) since in this way it can also be compared depending on the characteristics of the screen in which the images are projected obtaining an optimal criterion by the user to perform a more concrete analysis around the process by which the secret image was obtained through the mosaic image.

### **Keywords:**

- **INTERACTIVE TELEVISION**
- **GINGA**
- **QUALITY IN PRE-PROCESSED IMAGES**
- **ECLIPSE**



# CAPÍTULO I

## 1. INTRODUCCIÓN

### 1.1. Antecedentes

Los diferentes estándares para la transmisión de televisión digital terrestre se empezaron a estudiar en abril del 2009, entre los cuales el adoptado por el gobierno ecuatoriano debido a las características que posee, fue el ISDB-Tb japonés con variación brasileña. Ecuador firma los convenios de cooperación técnica y de capacitación con los gobiernos de Japón y Brasil el 26 de marzo de 2010, confirmando la introducción de este sistema, permitiendo a la población el cambio de señal analógica a digital en un lapso máximo de 7 años (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2017).

El “apagón analógico”, como se le conoce a este tipo de cambio de señal se aplazó recientemente en el país debido a que el 90% de la población no estaba preparada ya que los aparatos convencionales iban a quedar discontinuados al no poseer un sintonizador con el estándar mencionado anteriormente, así como también debido a la falta de densidad de señal para las transmisiones. Por estos motivos y teniendo en cuenta que la transición será progresiva, se lo anunció para junio del 2018 (Freire, 2017).

Si bien es cierto, aunque el cambio de señal no esté dado completamente, en el país ya se puede hablar de TDT (Televisión Digital Terrestre), que desde el 3 de mayo de 2013 se inauguró con la primera señal al aire por medio de la estación televisora TC Televisión en Guayaquil, así como también el 9 de mayo del 2013 salió al aire la señal digital en alta definición de Ecuavisa tanto en Quito como en Guayaquil (Diario el Comercio, 2013); esto ha ayudado a tener mejor calidad de video, imagen y sonido al igual que ha contribuido en el

aprovechamiento del espectro radioeléctrico (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2017).

Entre las ventajas del uso del estándar adoptado por Ecuador se permite el acceso a varios programas dentro de la señal de una misma estación, es decir, posee multiprogramación: noticiero, clases en vivo, información del clima, películas, etc.; de la misma manera la señal de alta calidad puede ser vista en distintos equipos móviles como smartphones o laptops; además de un hecho muy importante que es el poder recibir señales o alertas de emergencia permitiendo a la población salvar vidas sin dejar de lado el acceso a los contenidos interactivos en varios tópicos como salud, entretenimiento, turismo, etc. (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2017).

La interactividad mencionada anteriormente hace referencia a un nuevo concepto de ver la televisión ya que el telespectador, el cual únicamente era catalogado como un elemento pasivo, pasa a ser un elemento activo ya que puede decidir lo que en realidad le interesa para de esta manera beneficiarse de nuevos servicios que ofrece el televisor; convirtiéndose así en un medio de comunicación en el que las personas necesitan mayor protagonismo que simplemente ver y/o escuchar.

Cabe mencionar que la interactividad no solo viene dada a partir de la TDT ya que en décadas pasadas existían precedentes interactivos, un ejemplo importante es el del programa “Winky Dink and You” emitido desde 1953 a 1957, en donde se contaba las hazañas de un personaje de dibujos animados llamado Winky Dink, en donde, la interactividad se lograba a través de una pieza de plástico de vinilo que se pegaba a la pantalla del televisor mediante electricidad estática y el uso de crayones ayudando de esta manera al personaje en diversas actividades para concluir su misión (Greene, 2013). Otro tipo de ejemplo es el llamado teletexto incorporado por el canal BBC en 1972, el cual

consistía en un servicio de información en forma de texto emitido junto a la señal de televisión intercalando texto y bloques de colores (Morales Manuel & Koch Tommaso, 2016)

Hoy en día las personas se ven atraídas a esta nueva manera de obtener información mediante un medio de comunicación que tiempo atrás no permitía tener como principal al telespectador y a su toma de decisiones, por ende la interactividad que se necesita para poder brindar al usuario una correcta visualización de lo que se está informando requiere de una mayor cantidad de información, como por ejemplo imágenes, las cuales se envían a través de la señal de televisión dependiendo del ancho de banda que se tiene definido para cada canal.

Para poder transmitir imágenes en un menor ancho de banda se ha utilizado la transmisión de imagen mosaico en fragmento secreto mediante Transformaciones de Color Reversibles (TCR), en donde se obtiene los valores de desviación estándar de una imagen secreta y los parámetros estadísticos y estructurales de la imagen objetivo para de esta manera disminuir la cantidad de bits necesarios para recuperar la imagen y, por ende, se disminuye el ancho de banda ocupado en la transmisión de la misma. Se utiliza como camuflaje para la imagen secreta, dividiendo la imagen secreta en bloques y transformando sus características de color mediante TCR para que sean similares a las de la imagen objetivo correspondiente (Acosta, 2017). Para un informe comparativo de las imágenes recuperadas después de este proceso, se presenta este proyecto de investigación el cual tiene como alcance realizar un análisis de calidad de las imágenes mencionadas anteriormente en un escenario real de Televisión Digital Interactiva (TVi)

## 1.2. Justificación e Importancia

Actualmente la transmisión de imágenes por medio de sistemas de comunicación tiene diferentes aplicaciones, por ende, las nuevas tecnologías crean un sin número de plataformas o servicios que permitan el intercambio de información.

Se han propuesto varios métodos para la transmisión de imágenes, entre los cuales están el cifrado de imágenes y el ocultamiento de datos. La encriptación de imágenes nos da como resultado una imagen ruidosa que puede despertar la atención del atacante durante la transmisión (Fridrich, 1998; Chenet et al., 2004; Zhang et al., 2013; Kwok y Tang, 2007; Behnia et al., 2008; Xiao et al., 2009; Pati-dar et al., 2011; Suresh y Ajai, 2016; Senthil Ku-mar y Anandhi, 2016; Sarwate y Dakhore, 2015); por otro lado, la ocultación de datos posee la dificultad de que se necesita manejar una gran cantidad de datos para una sola imagen sin mencionar de que la imagen secreta debe estar muy comprimida por adelantado (Lee y Tsai, 2014).

Debido a lo mencionado anteriormente, se ha utilizado el enfoque estadístico sin tomar en cuenta la desviación estándar de la imagen secreta y los parámetros estadísticos y estructurales de la imagen objetivo, disminuyendo notablemente la cantidad de datos requeridos y por ende obteniendo una reducción en el ancho de banda necesario para la transmisión de las imágenes. (Acosta, 2017)

Este trabajo de investigación se enfoca en las imágenes obtenidas a través del proceso desarrollado en (Acosta, 2017), en donde el objetivo de este algoritmo es reducir en la medida de lo posible la información que posee la imagen mosaico, siempre y cuando el receptor sea eficiente para recuperar la imagen secreta por sí solo, todo esto se logra mediante dos fases principales, la creación de imágenes en mosaico con el ensamblaje de flujo de datos y la

recuperación de imágenes secretas. En la primera fase, se obtiene una imagen en mosaico como un contenedor para incluir el flujo de datos a recibir en recepción y en la segunda fase, el flujo de datos se utiliza para volver a cubrir una versión bastante buena de la imagen secreta

La importancia de este trabajo de investigación radica en el hecho de poder realizar un informe comparativo de calidad referente a las imágenes pre-procesadas mediante TCR en comparación con las imágenes originales, todo esto debido a que actualmente no se ha podido comprobar a ciencia cierta si el método utilizado para reducir el ancho de banda es el más óptimo para obtener una buena calidad al momento en que las imágenes recuperadas sean visualizadas por el espectador por medio de diversos tipos de receptores en un escenario real TVi.

La aplicación interactiva desarrollada a través de GINGA nos permitirá una correcta visualización de las imágenes pre-procesadas en diferentes tamaños y regiones para un óptimo análisis de calidad evaluado mediante tres métodos: el método MOS (Mean Opinion Score), el método RMSE y MSSIM; adicionalmente cabe recalcar que existirán tres escenarios diferentes para este análisis que constan de diferentes tipos de receptores, es decir, diferentes tipos de televisores, dentro de los cuales esta LCD, LED y de Rayos Catódicos.

### **1.3. Alcance del Proyecto**

El presente proyecto, se plantea como el informe comparativo de imágenes pre-procesadas mediante TCR, recuperadas a través del proceso empleado en (Acosta, 2017), en un escenario real TVi, en relación a imágenes originales, siendo evaluadas mediante los métodos objetivos RMSE, MSSIM y el método subjetivo MOS, con el fin de conocer que tan buena es la calidad de imagen al utilizar este tipo de proceso que reduce significativamente el ancho de banda para contribuir de esta manera a transmitir mayor cantidad de imágenes por el mismo canal.

La primera etapa del proyecto consiste en realizar el procesamiento de imágenes mediante transformaciones de color reversibles TCR, para lo cual, se obtendrá información necesaria que permita entender esta técnica empleada para la transmisión de imágenes. La imagen portadora que se utilizará será la imagen que aparece cuando un canal finaliza su transmisión diaria, es decir, una imagen de barras de colores y las imágenes secretas serán aleatoriamente escogidas.

Se realizará la correcta recuperación de las imágenes secretas por medio de la cantidad de datos optimizados que se obtuvieron al momento de procesarlas, de esta manera se podrán enviar una mayor cantidad de imágenes a través del canal ya que se optimizará el ancho de banda en la transmisión de cada una.

Una vez obtenidas las imágenes recuperadas, se desarrollará una aplicación en GINGA NCL, que nos permitirá visualizar dichas imágenes en diferentes tamaños y regiones para obtener una mejor percepción del usuario y así obtener un correcto análisis de calidad, el cual se llevará a cabo posteriormente.

Teniendo de esta manera todos los recursos necesarios se procederá a la transmisión de la información, en este caso de la aplicación interactiva, en un escenario real TVi, para lo cual se necesitará una correcta generación del TS (Transport Stream) que contenga audio, video y app en uno solo. La visualización se dará a través de diferentes receptores, es decir, se utilizará tres tipos de televisores: LCD, LED y de Rayos Catódicos para poder tener una comparación en diferentes escenarios.

Como parte final se realizará el análisis de calidad a través del método subjetivo MOS que consiste en evaluar la calidad de distintas muestras y ponderar los resultados para obtener una puntuación media, para esto dispone de

cinco puntos: Mala, Pobre, Regular, Buena y Excelente, entendiendo como Mala una distorsión muy molesta y como Excelente sin distorsión alguna en la imagen presentada. Adicional a esto se realizará un análisis de Error Cuadrático Medio (RMSE) que consiste en medir el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima; además se incluirá otro tipo de análisis que es el MSSIM, el cual es una técnica que nos permite obtener un valor de degradación entre una y otra imagen visualizada por el espectador.

#### **1.4. Objetivos**

##### **1.4.1. Objetivo General**

- Analizar la calidad de imágenes pre-procesadas mediante transformaciones de color reversibles en un escenario real TVi

##### **1.4.2. Objetivos Específicos**

- Desarrollar una aplicación interactiva mediante el uso de GINGA que permita visualizar las imágenes pre-procesadas en diferentes tamaños y regiones
- Transmitir las diferentes imágenes por medio del Transport Stream correspondiente para visualizarlas en los diferentes receptores
- Realizar un análisis de calidad a través del método subjetivo MOS para obtener una puntuación media de distintas muestras tomadas
- Realizar un análisis de calidad por medio del RMSE con el fin de conocer los puntos alrededor de la línea de mejor ajuste

- Realizar un análisis de calidad a través del método MSSIM para obtener un valor de degradación entre una imagen y otra dependiendo de la visualización del espectador



## **CAPÍTULO II**

### **2. MARCO TEÓRICO**

#### **2.1. Televisión Digital Terrestre (TDT)**

La Televisión Digital Terrestre (TDT) es una nueva manera de transmitir las señales existentes en Televisión Abierta con mayores prestaciones o ventajas como la calidad de video, audio e imagen. El espectro radioeléctrico se podrá aprovechar de mejor manera ya que las bandas de frecuencias serán liberadas para tener acceso a nuevas y futuras tecnologías (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2017).

Además de las ventajas mencionadas anteriormente de manera general, la TDT trae consigo también:

- Un mayor aprovechamiento del ancho de banda ya que un solo canal UHF puede transmitir varios programas con excelente calidad. Esto se lo puede lograr mediante la multiplexación de los canales deseados (cuatro o cinco para evitar la distorsión y la disminución en la calidad).
- Mayor límite de calidad de imagen y sonido, esto tiene que ver con la ventaja mencionada anteriormente ya que al aprovechar el ancho de banda las emisiones de TDT pueden tener mejor calidad de manera audiovisual debido a que los receptores de señal digital son capaces de corregir, hasta cierto límite, las distorsiones que son causadas por diversas interferencias. La imagen, sonido y datos que se emplean en la TDT se logra codificar mediante el estándar MPEG-2 que si bien es cierto es una compresión con pérdidas, la calidad de acuerdo a la relación señal/ruido aumenta. Todo esto nos garantiza una correcta recepción de la señal recibida.
- Mayor flexibilidad de las emisiones y servicios adicionales, si bien es cierto, aunque diversos operadores de televisión compartan un único canal multiplexado, cada uno puede adquirir el ancho de banda que necesite para brindar a sus clientes o telespectadores los contenidos que se desee. Toda

esta información es aprovechada por el usuario mediante el receptor TDT y sus diversas aplicaciones que posee.

El uso de la TDT y su constante desarrollo ha ayudado a que aparezcan una serie de nuevas tecnologías que se relacionan de alguna manera con la misma (Ministerio de Energía, Turismo y Agenda Digital, 2017):

- **Alta definición:** Este tipo de televisión (TVAD) representa una recepción más avanzada en lo que se refiere a calidad de imagen, obteniendo de esta manera más nitidez y colores más enriquecedores a la vista del telespectador. Ofrece entre cuatro y cinco veces la resolución de un sistema de televisión de definición estándar ya que se amplía el ángulo de visión de 10° hasta los 32° ayudando al realismo y al impacto que el usuario va a poseer al tener una pantalla de gran formato.
- **Interactividad:** Nos permite ofrecer contenidos adicionales a los programas de televisión, permitiendo al usuario obtener información asociada a lo que está observando en la transmisión cotidiana, como, por ejemplo, participar en concursos, votar, adquirir un bien o un servicio determinado o incluso llegar a participar en el programa que se está visualizando; de esta manera al usuario se le da un nuevo protagonismo, siendo este el encargado de tomar decisiones de acuerdo a sus intereses. Por medio de la interactividad se permite complementar de alguna u otra manera los contenidos de televisión por medio de servicios públicos, comerciales o de entretenimiento que solo eran ofrecidos por medio de smartphones u ordenadores.
- **Tecnología 3D:** Consiste en mostrar a cada ojo del telespectador, dentro de la región estereoscópica del campo de visión, las imágenes adecuadas para de esta manera tener una sensación de profundidad en cada escena. Para poder separar las imágenes dirigidas a cada ojo existen cuatro métodos, entre ellos, la multiplexación espacial (requiere gafas polarizadas), la multiplexación temporal (requiere gafas LCD), la autoestereoscopia (no requiere el uso de gafas) y los anaglifos (requiere gafas rojas y azules)

Existen cuatro estándares vigentes a nivel mundial para poder adoptar la TDT, cada país se ha apegado a uno de ellos dependiendo las necesidades que se requieran. En Ecuador, por ejemplo, se ha adoptado el estándar ISDB-Tb como en la mayoría de los países de Sudamérica.

## **2.2. Sistema de televisión digital (ISDB-Tb)**

ISDB (Integrated Services Digital Broadcasting) o Radiodifusión Digital de Servicios Integrados son un conjunto de normas creado por Japón para las transmisiones de radio y televisión digital.

ISDB está formado por una familia de componentes entre los cuales está el de televisión digital terrestre ISDB-Tb, el cual permite tener una recepción de teléfonos móviles en bandas de TV como, por ejemplo, un servicio integrado es 1seg para recepción en telefonía móvil, computadores portátiles y vehículos.

ISDB-Tb (ISDB-T Built-in) también llamado ISDB-T Internacional es un estándar de televisión digital basado en ISDB-T.

Su principal diferencia de la norma japonesa convencional es el uso del códec MPEG-4 para compresión de vídeo, en cuanto a la compresión de audio se lo hace mediante HE-AAC, la modulación es BST-OFDM-TI, presentando 30 fps hasta en dispositivos portátiles y la interacción usando el middleware desarrollado en Brasil denominado Ginga, el cual está compuesto por sus módulos Ginga-NCL (documentos en lenguaje NCL) y Ginga-J (documentos en lenguaje Java).

Fue desarrollado por un grupo de estudio del Ministerio de Comunicaciones y liderado por la Agencia Brasileña de Telecomunicaciones con

apoyo del Centro de Investigación y Desarrollo (CPqD). El objetivo de este grupo era diseñar e implementar un estándar en Brasil teniendo inclusión digital para las personas que están al margen de la sociedad de la información.

En enero del 2009 se publicó un documento para adherir la ISDB-Tb, la cual ha sido aceptada en varios países como Argentina, Chile, Bolivia, Costa Rica, Filipinas, Ecuador, Nicaragua, Paraguay, Perú, Uruguay, Venezuela y El Salvador a partir de mayo del 2017.

En abril del 2009 fue certificado oficialmente por la Unión Internacional de Telecomunicaciones (UIT) sin dejar atrás el módulo de Ginga-NCL y el lenguaje NCL/Lua como primera recomendación internacional para entornos multimedia interactivos para TV Digital y IPTV. Su importancia radica en la normalización de middleware para la interactividad en dispositivos y cajas de instalación de IPTV y TV Digital.

Este tipo de sistema presenta también algunas adaptaciones, siendo las principales citadas a continuación:

- Multiplexación y estructuras y señalización de datos adaptadas a las normas occidentales de escritura, considerando caracteres de lenguas derivadas del latín
- Máscaras de emisión de los transmisores adaptadas para cumplir con cualquier caso de interferencia con otras estaciones
- Implementación de Recepción Abierta
- Presentación de 30 fps incluso en receptores portátiles como se mencionó anteriormente de manera general.

En la Tabla 1 se muestra un resumen de las características técnicas principales del sistema ISDB-Tb

**Tabla 1**  
*Características técnicas principales del sistema ISDB-Tb*

<b>Codificación del canal de Transmisión</b>	Esquema de Modulación	<ul style="list-style-type: none"> <li>• 64QAM-OFDM,</li> <li>• 16QAM-OFDM,</li> <li>• QPSK-OFDM,</li> <li>• DQPSK-OFDM (Transmisión Jerárquica)</li> </ul>
	Código de corrección de errores	<ul style="list-style-type: none"> <li>• Codificación Interna: Convolución 7/8, 3/4, 2/3, 1/2</li> <li>• Codificación Externa: RS(204,188)</li> </ul>
	Intervalo de Guarda	1/16, 1/8, 1/4
	Intercalado	Por Tiempo, Frecuencia, bit y byte
	Tipo de Modulación	BST-OFDM (OFDM Segmentado - 13 segmentos)
<b>Acceso Condicional</b>		Multi-2
<b>Middleware</b>		Ginga-NCL y Ginga-J
<b>Información de Servicio</b>		Norma ARIB STD B-10
<b>Multiplexado</b>		Sistema MPEG-2
<b>Codificación de Audio</b>	Fijo/Móvil	<ul style="list-style-type: none"> <li>• Estéreo: MPEG-4 AAC@L2 o MPEG-4 HE-AAC v1@L2</li> <li>• Audio Multicanal 5.1: MPEG-4 AAC@L4 o MPEG-4 HE-AAC v1@L4</li> </ul>
	Portátil	Solo Estéreo: MPEG-4 HE-AAC v2@L2
<b>Codificación de Video</b>	Fijo/Móvil	MPEG-4 AVC (H.264) HP@L4
	Portátil	MPEG-4 AVC (H.264) BP@L1.3

### 2.3. Televisión digital interactiva (TVDi)

La televisión digital interactiva hace referencia a un nuevo concepto de ver la televisión, en este caso, el telespectador que antes era considerado como un elemento pasivo, pasa a ser un elemento activo que puede elegir lo que sinceramente le interesa para de esta manera adquirir nuevos servicios mediante este tipo de medio de comunicación, la televisión.

De esta manera, la TVDi ayuda a obtener una nueva adaptación de la televisión a un medio en el cual el telespectador solicita tener mayor protagonismo cuando así se lo requiera.

Dentro de algunos de los conceptos de Televisión Digital Interactiva podemos citar los siguientes con sus respectivos autores:

- “Una televisión en la que el telespectador puede hacer algo más que simplemente verla y/o oírla, pasando a ser una especie de teleparticipante, ya que tiene la posibilidad de realizar acciones que pueden cambiar el contenido mostrado por su televisor” (Melechón Javier, URL)
- “La televisión interactiva se puede definir como cualquier cosa que permita al espectador o espectadores de televisión y a las personas que hacen que el canal de televisión, programa o servicio entable un diálogo” (Gawlinski Mark, BBC)
- “Más específicamente, se puede definir como un diálogo que lleva a los espectadores más allá de la experiencia pasiva de ver y les permite tomar decisiones y tomar medidas, incluso si la acción es tan simple como completar una postal y abrirla por correo, o dibujar una imagen en la pantalla del televisor” (Gawlinski Mark, BBC)

### **2.3.1. Servicios interactivos**

Se puede definir tres categorías de servicios interactivos:

- Servicios de información: Son aquellos servicios que brindan una información que no tiene que ver con la programación audiovisual que se está brindando en ese momento, es decir, es independiente.
- Servicios ligados a la programación: Son aquellos servicios que de una u otra manera complementan con algún tipo de información adicional la programación audiovisual que se está emitiendo en ese instante.

- **Servicios transaccionales:** Son aquellos servicios que brindan la posibilidad de enviar y recibir algún tipo de información de forma personalizada y exclusiva para el usuario que en ese momento se encuentra recibiendo la programación habitual.

### **2.3.2. Aplicaciones interactivas según los servicios**

Una aplicación interactiva es en sí un programa adicional al contenido que se muestra habitualmente en la televisión al que se puede acceder y ejecutar con un decodificador interactivo. Tenemos tres diferentes tipos de aplicaciones interactivas según los servicios que ofrecen, las cuales a su vez se subdividen en las indicadas a continuación:

- **Servicios públicos:**
  - Servicios de información como por ejemplo el estado del tiempo de acuerdo a la hora del día, el tráfico que existe en un lugar determinado, las farmacias que se encuentran de guardia para alguna emergencia, los teléfonos de interés público, algún tipo de información que se requiera de los aeropuertos, etc.
  - Servicios avanzados como por ejemplo las citas que puede tener en algún tipo de servicio sanitario, los impuestos que debe pagar de acuerdo a la información que se entregue, información o servicio de tipo municipal, etc.
- **Servicios comerciales o de ocio:**
  - Concursos que pueden existir en ese momento, votaciones de una u otra entidad, publicidad interactiva que ayude al usuario en temas de interés, venta por impulso, compra de eventos como puede ser partidos de fútbol, entradas al cine o a diferentes conciertos, etc.
  - Informaciones sobre concursantes que se encuentran participando en ese momento en la programación audiovisual, información sobre actores de la película que se está visualizando, etc.

- Servicio de alerta temprana como por ejemplo cuando ocurra una emergencia debido a un terremoto, sismo, tormenta o algún tipo de fenómeno meteorológico que atente contra la vida de las personas.

### 2.3.3. Tipos de interactividad

Poseemos tres tipos de interactividad divididas de la siguiente manera:

- Interactividad local:

El espectador únicamente interactúa con la información que tiene almacenada el receptor, la cual puede ser renovada en un período de tiempo.

La principal diferencia con los siguientes tipos de interactividad radica en que el usuario si bien es cierto, puede acceder a los contenidos interactivos, pero no puede enviar datos de vuelta, es decir, no tiene una comunicación en tiempo real o directa con el proveedor de servicios.

Ejemplos de este tipo de interactividad son las guías de programación, el teletexto digital o la información que se da de las personas que están participando en ese momento en algún tipo de programa de televisión

- Interactividad con canal de retorno:

El telespectador interactúa directamente con el proveedor de servicios exterior mediante el uso de un canal de retorno.

De esta manera, se permite no solo ver los contenidos adicionales a la programación televisiva, sino que también se puede enviar diferentes respuestas o requerimientos por parte de los usuarios incluso hacia otros



usuarios con el mismo tipo de necesidad. Por medio de esto, se permite a los espectadores votar, enviar mensajes o solicitudes.

Se pueden realizar diferentes tipos de servicio con la ayuda del canal de retorno como, por ejemplo, una guía electrónica de programación avanzada, teletexto mejorado, información básica del clima, o servicios como aeropuertos y el tráfico en alguna zona específica, etc., información tipo carrusel emitida por el operador, información particular y optimizada para el usuario, compras, votaciones y encuestas (Ministerio de Energía, Turismo y Agenda Digital, 2017)

#### **2.3.4. Taxonomía de aplicaciones interactivas**

Las aplicaciones interactivas se pueden clasificar de la siguiente manera:

- Canal de retorno en base a la estructura física
  - Local
  - Intermitente
  - Permanente
  
- Aplicación en base a la relación principal del video y las aplicaciones
  - Aplicación vinculada y sincronizada
  - Aplicación vinculada y sin sincronización
  - Aplicación desvinculada y sin sincronización
  
- GUI basado en el diseño básico
  - Video superpuesto
  - Video redimensionado
  - Solo GUI de la aplicación

## 2.4. Middleware GINGA NCL

Un middleware es una capa cuyo objetivo es el de comunicar la capa de aplicación y los servicios ofrecidos por las capas inferiores, de manera específica en el tema, un middleware para aplicaciones de TV digital está formado por máquinas de ejecución de los lenguajes que se ofrecen y librerías de funciones para realizar diferentes actividades en cuanto a programación se refiere.

Existen diversos middlewares existentes para interactividad como, por ejemplo:

- MHP (Multimedia Home Platform): diseñado por el proyecto DVB
- ARIB (Association of Radio Industries Businesses): relacionado con el sistema ISDB
- ACAP (Advanced Common Application Platform): creada por el Comité de Sistemas Avanzados de Televisión (ATSC)
- Ginga
- GEM (Globally Executable MHP): basada en MHP, elimina los elementos específicos orientados a DVB.

Ginga es el resultado de varios proyectos de investigación coordinados por los laboratorios Telemidia de la PUCRio y LAViD de la UFPB, es un middleware abierto del Sistema Nipo-Brasileño de TV Digital (ISDB-Tb), el mismo que ha sido adoptado en el país.

Está formado por un grupo de tecnologías estandarizadas e innovaciones brasileñas que lo convierten en un middleware muy avanzado en comparación a los antes mencionados.

Ginga consta de dos tipos principales:

- Ginga NCL basado en lenguaje declarativo
- Ginga J basado en lenguaje imperativo

Ginga NCL es un ambiente de presentación multimedia para aplicaciones escritas en lenguaje NCL y su scripting Lua (Ginga, 2017a). Se desarrolló en la Pontificia Universidad Católica de Río de Janeiro como un subsistema lógico necesario para Ginga ya que se encuentra a cargo de la presentación de documentos NCL (Nested Context Language) (Ginga, 2017b).

NCL es un lenguaje de aplicación XML (eXtensible Markup Language) que brinda facilidad en interactividad, sincronismo espacio-tiempo entre objetos, contenido multimedia y alternativas de presentación, la exposición en diversos dispositivos y producción de aplicaciones interactivas no lineales. Para todo esto, NCL define una separación contundente entre el contenido que posee la aplicación y su estructura para de esta manera poder definir objetos multimedia bien estructurados y relacionados tanto en espacio como en tiempo (Ginga, 2017a).

## **2.5. Procesamiento de imágenes digitales**

El procesamiento de imágenes se refiere a las técnicas aplicadas a imágenes digitales para de esta manera ayudar a mejorar la calidad o buscar información correspondiente a dicha imagen, las cuales son las dos áreas principales de aplicación frente a la necesidad de procesamiento de tal o cual imagen seleccionada.

Un aspecto muy importante del procesamiento es la percepción psicovisual que posee el ser humano ya que según el observador y como su nombre lo indica, según su percepción, se decide al final si la imagen le es agradable o no sin tomar en cuenta que clase de tratamiento o procesamiento se le realizo a la imagen en cuestión (Escalante Ramírez, 2006).

El principal instrumento utilizado para realizar algún tipo de procesamiento de imagen es las matemáticas. Por medio de algoritmos y una computadora funcional se puede llegar a diseñar e implementar un sin número de maneras necesarias para manipular de una u otra manera las imágenes (Escalante Ramírez, 2006)

### **2.5.1. Etapas del procesamiento de imágenes**

Las principales etapas que se siguen al realizar un procesamiento de una imagen determinada son las siguientes (Wainschenker, Massa, & Tristan, 2011):

- **Captura:** En esta etapa se realiza un diseño de las propiedades que se van a ver inmersas en el momento de capturar la imagen, por ejemplo, el tipo de cámara que se va a utilizar, la distancia a la que se va a encontrar del objeto, los pixeles que va a tener la imagen, etc.
- **Pre-procesamiento:** Etapa en la cual se va a disminuir el entorno que no se necesita para lo que se va a estudiar, por ejemplo, el fondo que tiene la imagen, el ruido que puede llegar a existir, etc.
- **Segmentación:** En esta etapa se logra identificar y obtener todos y cada uno de los objetos que están presentes en la imagen.
- **Extracción de características:** Etapa en la cual se elige y extrae, como su nombre lo dice, las características adecuadas para poder identificar de una u otra forma el objeto que se necesita

- Identificación de objetos: En esta etapa se utiliza un modelo que ayude a tomar decisiones en base a la clase a la que pertenece cada uno de los objetos seleccionados

### 2.5.2. Procesamiento en el dominio espacial

En este tipo de procesamiento de imágenes se hace referencia a la imagen en sí, es decir, los métodos utilizados se basan en la manipulación directa de los píxeles de la imagen (Jiménez Rodríguez, 2017b).

#### 2.5.2.1. Entorno 1 x 1: Operaciones pixel a pixel

- Operaciones aritméticas:

Se tiene dos imágenes en función de  $x, y$ ; las cuales se denotan de la siguiente manera  $f(x,y)$  y  $g(x,y)$ , cada una compuesta de  $M$  filas y  $N$  columnas, de esta manera se definen las siguientes operaciones (Jiménez Rodríguez, 2017b):

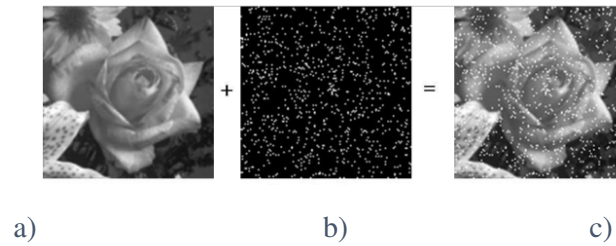
$$s(x, y) = f(x, y) + g(x, y) \quad (1)$$

$$d(x, y) = f(x, y) - g(x, y) \quad (2)$$

$$p(x, y) = f(x, y) \times g(x, y) \quad (3)$$

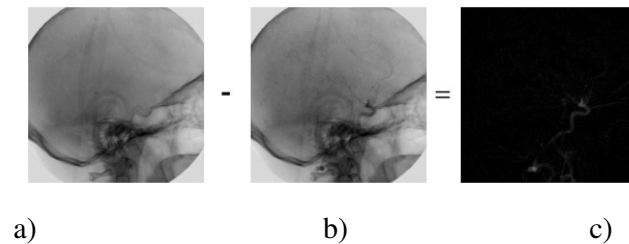
$$d(x, y) = f(x, y) \div g(x, y) \quad (4)$$

En la adición, como se muestra en la Figura 1, lo que se hace es reducir el ruido que posee una imagen a través de la adición de imágenes con ruido. Se tiene una imagen original  $f(x,y)$  y una imagen aleatoria con ruido  $g(x,y)=f(x,y)+n(x,y)$  (5), en donde la intensidad de los píxeles de  $n(x,y)$  son valores aleatorios no correlacionados y con esperanza nula (Jiménez Rodríguez, 2017b)



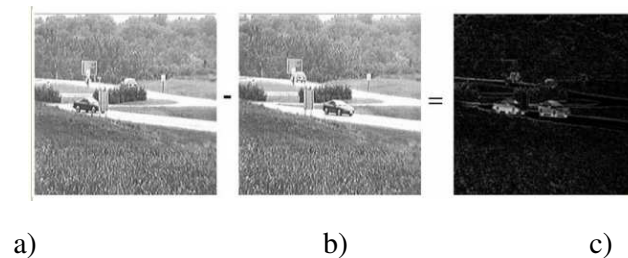
**Figura 1.** Adición: a) Imagen original, b) Ruido, c) Imagen ruido aditivo  
Fuente: (Jiménez Rodríguez, 2017b)

En la sustracción como se muestra en la Figura 2, se hace un realce importante a las diferencias que existen entre imágenes (Jiménez Rodríguez, 2017b)



**Figura 2.** Sustracción (diferencia entre imágenes): a) Imagen Rayos X, b) Imagen con contraste radiológico c) Diferencia entre imágenes  
Fuente: (Jiménez Rodríguez, 2017b)

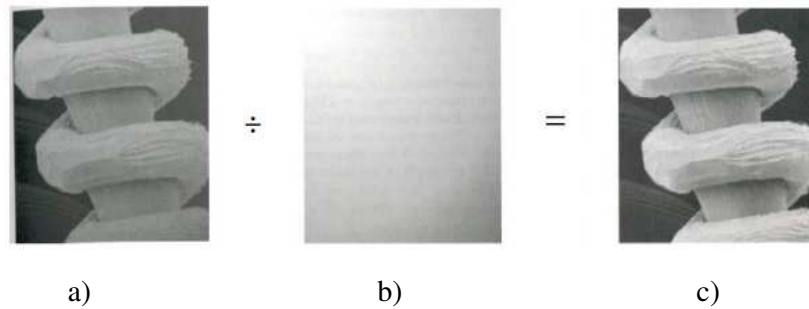
Así como también se puede realizar la detección del movimiento que ha ocurrido entre una imagen y otra, ambas de la misma escena, esto se puede visualizar a través de la Figura 3 mostrada a continuación (Jiménez Rodríguez, 2017b).



**Figura 3.** Sustracción (detección de movimiento): a) Imagen uno, b) Imagen dos c) Diferencia entre imágenes y detección del movimiento realizado entre imagen uno e imagen dos.

Fuente: (Jiménez Rodríguez, 2017b)

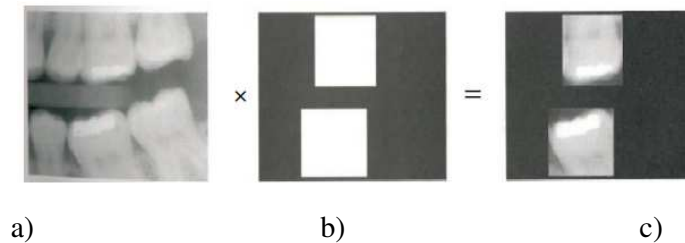
En la división y el producto mostrados en la Figura 4 y Figura 5 respectivamente, se puede realizar funciones como la de corrección de sombreado de una imagen (Jiménez Rodríguez, 2017b).



**Figura 4.** División: a) Imagen formada por una función perfecta y una función de sombreado, b) Función de sombreado conocida, c) Función formada por la función perfecta y la inversa de la función de sombreado conocida

Fuente: (Jiménez Rodríguez, 2017b)

También se puede destacar regiones de interés, esto se consigue mediante la multiplicación de una imagen máscara formada de unos en las regiones antes mencionadas y ceros en lo que sobra (Jiménez Rodríguez, 2017b).



**Figura 5.** Producto: a) Radiografía dental, b) Máscara de regiones de interés, c) Producto.

Fuente: (Jiménez Rodríguez, 2017b)

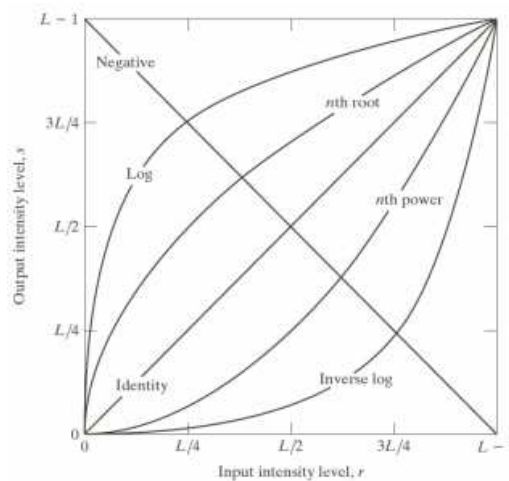
- Transformaciones de intensidad:

Dentro de las transformaciones de intensidad tenemos (Jiménez Rodríguez, 2017b):

- Transformaciones básicas:

Dentro de este grupo tenemos la transformación lineal, logarítmica, potencia-raíz y definidas a trozos

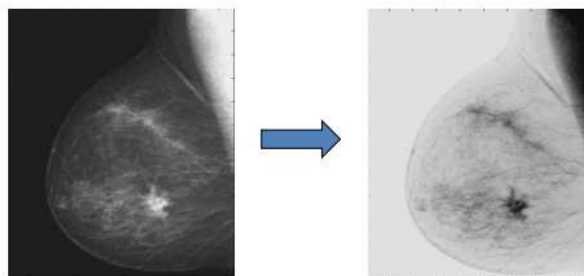
Cada una de ellas están formadas por otras diferentes, por ejemplo, la transformación lineal puede ser de identidad o negativa, la transformación logarítmica puede ser log e inversa de log y la transformación potencia-raíz puede ser potencia n-ésima y raíz n-ésima. Todas y cada una de ellas las podemos visualizar a continuación en la Figura 6.



**Figura 6.** Transformaciones básicas de intensidad.

Fuente: (Jiménez Rodríguez, 2017b)

El efecto de la transformación lineal  $T(r)=L-1-r$  (6) de imagen negativa es aquella que hace invertir el orden de los valores de intensidad como se puede mostrar en la Figura 7.  $L$  corresponde al nivel de intensidad ( $L=2^n$  donde  $n$  puede tomar valores desde 1 en adelante) y  $r$  es el ruido existente.

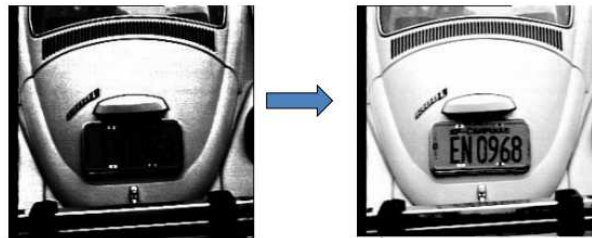


**Figura 7.** Transformación lineal (Imagen negativa).

Fuente: (Jiménez Rodríguez, 2017b)



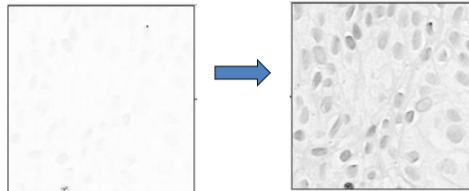
En la Figura 8 tenemos la transformación logarítmica  $T(r) = c \times \log(1+r)$  (7), la cual es usada para poder llegar a visualizar cuando se tiene bajos niveles de intensidad de manera más dinámica. Donde  $c$  es una constante y  $r$  (ruido) debe ser mayor o igual a 0.



**Figura 8.** Transformación logarítmica.

Fuente: (Jiménez Rodríguez, 2017b)

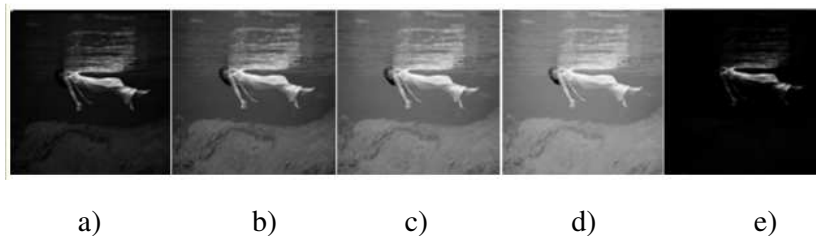
La transformación inversa de log  $T(r) = c \times e^{r-1}$  (8) ayuda a mejorar la discriminación visual en zonas que poseen alta luminosidad como en la Figura 9. Donde tanto  $c$  como  $e$  son constantes y  $r$  es el ruido existente.



**Figura 9.** Transformación inversa de log.

Fuente: (Jiménez Rodríguez, 2017b)

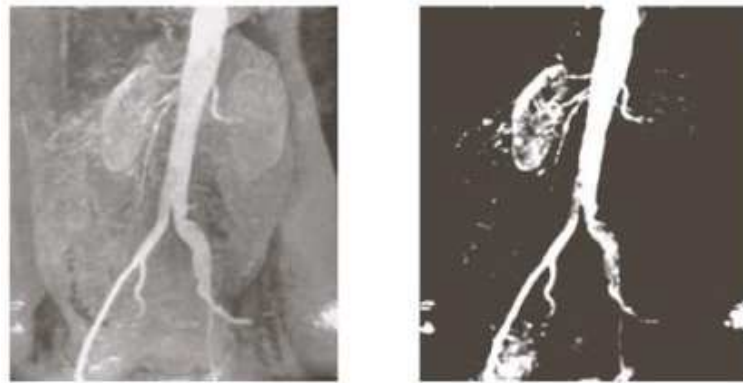
En la Figura 10 se muestra la transformación de potencia-raíz  $T(r) = c \times r^n$  (9), posee la ventaja de que se puede realizar una variedad de transformaciones al variar  $n$ . Donde  $c$  es una constante,  $r$  es el ruido existente y  $n$  va variando.



**Figura 10.** Transformación de potencia-raíz: a) Imagen original, b)  $n=1/2$ , c)  $n=1/3$ , d)  $n=1/4$ , e)  $n=2$ .

Fuente: (Jiménez Rodríguez, 2017b)

Por último, como se visualiza en la Figura 11, las transformaciones definidas a trozos ayudan a especificar un rango del nivel de gris en una imagen.

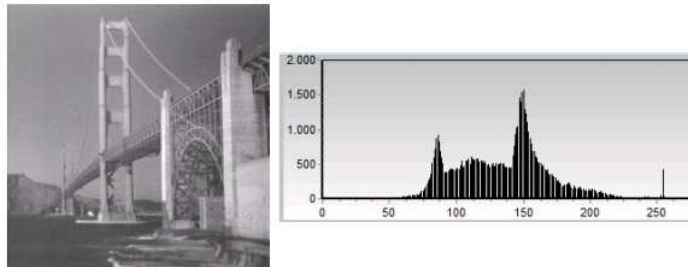


**Figura 11.** Transformación definida a trozos.

Fuente: (Jiménez Rodríguez, 2017b)

➤ Procesamiento del histograma:

El histograma, es la herramienta que nos ayuda a tomar una decisión en base a qué tipo de transformación es la adecuada para una imagen determinada. En este tipo de gráfica, Figura 12, se muestra el número de píxeles de cada nivel de gris que posee la imagen.



**Figura 12.** Representación del histograma de una imagen.

Fuente: (Jiménez Rodríguez, 2017b)

Por medio de un análisis realizado al histograma se puede comparar contraste e intensidades entre imágenes, así mismo, se lo puede alterar para de esta manera cambiar la imagen a nuestra conveniencia. De esta manera se puede determinar que para aumentar el contraste se expande el histograma o se lo ecualiza

### 2.5.2.2. Entorno $n \times n$ , $n > 1$ : Operaciones de contorno (filtros)

Las funciones de procesamiento de imagen en el dominio espacial están formadas mediante una imagen original, una imagen procesada de resultado y un operador que interviene en la imagen original (Jiménez Rodríguez, 2017c).

Dentro de este tipo de operaciones de contorno tenemos (Jiménez Rodríguez, 2017c):

- Filtros de suavizado:

Utilizados para eliminar detalles antes de la segmentación de un objeto determinado, para rellenar espacios pequeños y para eliminar ruido

Se clasifican en filtros lineales y filtros no lineales.

Los filtros lineales pueden ser (Jiménez Rodríguez, 2017c):

- Filtros de la media:

Se cambia el valor de cada pixel por el promedio de los valores que poseen los pixeles que están juntos o los pixeles vecinos. Esto se lo puede obtener mediante una convolución con una máscara definida. Ayuda a eliminar ruido, a suavizar imágenes reduciendo las variaciones de intensidad y a conseguir que la intensidad de objetos se mezcle con la del fondo.

- Filtro gaussiano:

Disminuye la nitidez, aumenta borrosidad, pierde detalles. Ayuda a obtener un suavizado más uniforme que el filtro mencionado anteriormente.

Los filtros no lineales pueden ser (Jiménez Rodríguez, 2017c):

- Filtros estadísticos ordenados:

Como su nombre lo indica, ayuda a ordenar los valores de los píxeles que están junto a un píxel determinado, de menor a mayor para de esta manera obtener un valor en la lista

Existen los filtros estadísticos ordenados máximo, mínimo y mediana.

Los filtros estadísticos ordenados máximo, como su nombre lo indica, selecciona el mayor valor dentro de valores ordenados de nivel de gris, ayudando de esta manera a reducir el ruido de píxeles negros.

Los filtros estadísticos ordenados mínimo selecciona el menor valor dentro de los valores ordenados de píxeles, ayudando a eliminar el ruido de píxeles blancos

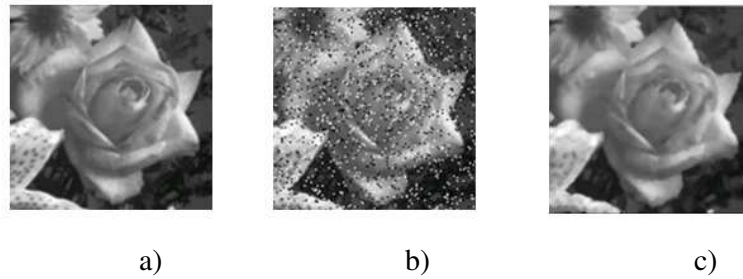
Ambos tipos de filtros estadísticos ordenados se muestran en la Figura 13.



**Figura 13.** Filtro estadístico ordenado mínimo y máximo.

Fuente: (Jiménez Rodríguez, 2017c)

Los filtros estadísticos ordenados mediana, Figura 14, selecciona el valor en la posición intermedia (mediana) dentro de valores ordenados de la vecindad de cada píxel, ayudando de esta manera a atenuar el ruido impulsional, a eliminar efectos engañosos y a guardar los bordes de la imagen



**Figura 14.** Filtro estadístico ordenado mediana: a) Imagen original, b) Imagen con ruido, c) Imagen aplicada filtro de mediana 3 x 3  
Fuente: (Jiménez Rodríguez, 2017c)

- Filtros de realce:

Este tipo de filtros ayuda a realzar los detalles que posee una imagen que por una u otra manera pudieron ser emborronados, por ende, detectan bordes.

Para detectar bordes, generalmente se calcula un operador de derivación, esto debido a que un pixel pertenece a un borde si existe un cambio brusco de niveles de grises en comparación con los pixeles que se encuentran a su alrededor. Un cambio más brusco ayuda a detectar de mejor manera un borde de la imagen.

### 2.5.3. Procesamiento en el dominio de la frecuencia

Este tipo de procesamiento de imágenes se ayuda de la utilización de una transformada de la imagen (Transformada de Fourier) para luego aplicar la transformada inversa y de esta manera obtener resultados (Jiménez Rodríguez, 2017a)

La transformada de Fourier tiene las siguientes propiedades (Jiménez Rodríguez, 2017a):

- Núcleo simétrico y separable:

La ventaja de esta propiedad es que se puede tener la transformada o su inversa únicamente en dos pasos, a través de la Transformada de Fourier 1-D o su inversa.

- Linealidad:

Se puede ver a la transformada de Fourier como una aplicación lineal, debido a que, teniendo dos imágenes  $f$  y  $g$  en función de  $(x,y)$ , la transformada de Fourier de la combinación lineal de las dos imágenes mencionadas anteriormente da como resultado la combinación lineal de las transformadas de Fourier de cada una de ellas. Cabe mencionar que la transformada inversa de Fourier también cumple esta propiedad.

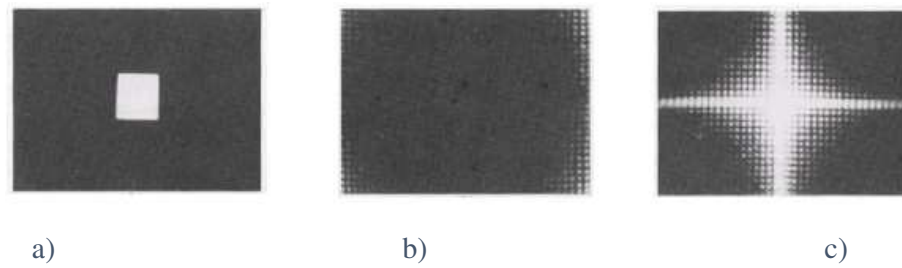
- Simetría y periodicidad:

En lo que se refiere a la simetría, si  $f(x,y)$  es real, lo cual siempre sucede en imágenes, se satisface que  $|F(u,v)|=|F(-u,-v)|$ , dado que  $F(u,v)$  es la transformada de Fourier y  $F(-u,-v)$  es su conjugada.

En cuanto a la periodicidad, la transformada de Fourier y su correspondiente inversa son funciones periódicas, las cuales tiene un período  $N$  en ambas direcciones.

- Traslación y rotación:

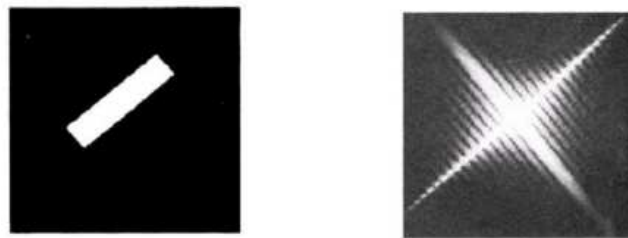
Al tener una imagen en función de  $(x,y)$  y multiplicarla por una función exponencial, su transformada de Fourier se traslada, centrándose de esta manera en un nuevo origen definido en la función mencionada anteriormente, como por ejemplo, la visualización de la Figura 15. De esta manera, se mueve el origen al centro de la matriz  $N \times N$  que corresponda. Es importante mencionar que un desplazamiento de la función, no hace que exista un cambio de magnitud de su correspondiente transformada de Fourier.



**Figura 15.** Traslación: a) Imagen simple, b) Espectro de Fourier sin desplazar, c) Espectro de Fourier desplazado hacia el centro.

Fuente: (Jiménez Rodríguez, 2017c)

La rotación de la función hacia un ángulo determinado hace que la transformada de Fourier también se afecte con la rotación del mismo ángulo como se puede ver en la Figura 16, esto se aplica también a su inversa.

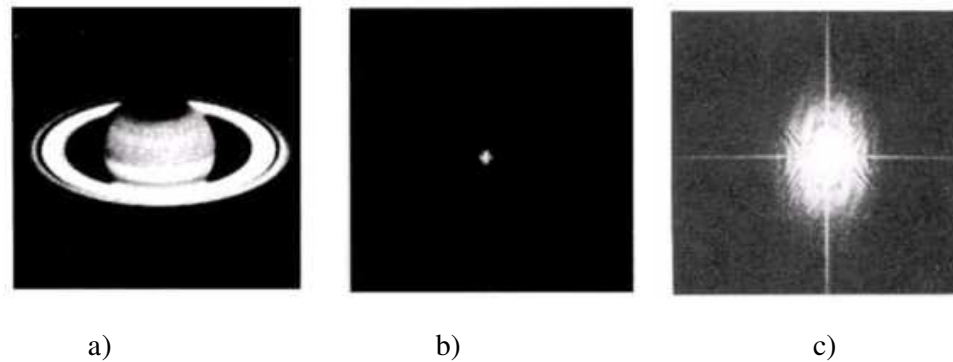


**Figura 16.** Rotación

Fuente: (Jiménez Rodríguez, 2017c)

- Visualización del espectro de Fourier:

El espectro de Fourier posee un rango muy amplio que los que se visualizan usualmente al momento de mostrar una imagen. Para poder evitar esto, se aplica la transformación de intensidad logarítmica para de esta manera dar como resultados los valores de la paleta de colores que se encuentran disponibles, Figura 17.



**Figura 17.** Representación del logaritmo del espectro: a) Imagen original, b) Espectro de Fourier, c) Logaritmo del espectro a 8 bits, escala de grises de 256.

Fuente: (Jiménez Rodríguez, 2017c)

Los filtros en el dominio de la frecuencia se utilizan para poder eliminar altas o bajas frecuencias de la imagen, de esta manera se suaviza a la misma o bien se realza o detecta bordes.

#### 2.5.4. Segmentación de imágenes

Su objetivo es el dividir la imagen en regiones que sean significativas en base a la aplicación determinada que se desea emplear (Jiménez Rodríguez, 2017d)

La subdivisión y su nivel dependerá de lo que se desea estudiar o del problema que se desee resolver, es decir, la segmentación se detiene cuando el o los objetos de interés para la aplicación se hayan aislado (Jiménez Rodríguez, 2017d)

El resultado es una imagen más simplificada en base a los grupos significativos determinados anteriormente, los cuales componen la imagen original (Jiménez Rodríguez, 2017d)

Los algoritmos utilizados para la segmentación se basan en las propiedades básicas que poseen los niveles de gris (Jiménez Rodríguez, 2017d):



- **Discontinuidad:**

Los bordes de las regiones son distintos del fondo, lo cual ayuda a la detección de los mismos al basarse en cambios bruscos de niveles de intensidad

- **Similitud:**

Divide la imagen en referencia a la búsqueda de zonas que tengan valores similares con parámetros fijados anteriormente.

- **Umbralización:**

Ayuda a diferenciar un objeto del fondo que tiene la imagen a través de una binarización

### 2.5.5. Áreas de aplicación

Las áreas en las que actualmente se están ocupando el procesamiento de imágenes se muestran en la Tabla 2

**Tabla 2**  
*Áreas de aplicación del procesamiento de imágenes*

Área	Detalle
Biología y Genética	Al realizar el análisis de huesos, tejidos y células. Análisis y clasificación de ADN
Defensa / Inteligencia	Interpretación automática de imágenes obtenidas de manera satelital para ayudar en la búsqueda de objetivos militares
Procesamiento de documentos	Detectar y reconocer de manera automática los caracteres existentes en un documento
Automatización Industrial	Inspección visual de manera automática Análisis de las características de piezas manufacturadas en una línea de producción

CONTINÚA 

Forense / Criminología	Análisis de huellas dactilares
	Análisis de comparación entre fotografías
Análisis de materiales	Visualizar internamente un material Conteo automático y clasificación de objetos
Fotografía / Video	Componer una escena con la ayuda de varios objetos Aumentar efectos especiales
Imágenes Satelitales	Análisis de paisajes, características de vegetación
Medicina	Rayos X, resonancias magnéticas, tomografías, etc. Ayuda en sistemas de diagnóstico

Fuente: (Jiménez Rodríguez, 2017a)

## 2.6. Tipos de imágenes

De manera general, existen dos tipos de imágenes:

- **Imágenes vectoriales:**

Son aquellas que están formada por figuras u objetos geométricos independientes como, por ejemplo, líneas, curvas, polígonos, etc. los cuales están representados mediante funciones matemáticas (vectores) que ayudan a especificar su forma, color, posición, etc. (González Arribas, 2017)

Este tipo de imágenes se crean a través de programas de diseño o dibujo vectorial como son Adobe Illustrator, Corel Draw, Inkscape, etc., teniendo como principal ventaja que al aumentar el tamaño de la misma no se produce el “pixelado” que poseen las imágenes de mapa de bits (González Arribas, 2017)

- **Imágenes de mapa de bits:**

Son aquellas que están formadas por un conjunto de puntos llamados píxeles, en donde cada uno de estos posee información de color y luminosidad para al unirlos formar la imagen determinada (González Arribas, 2017)

Poseen una mayor gama de colores y de tonos en comparación con las imágenes de tipo vectoriales y es por esto que se las usa en fotografías creadas a través de cámaras de fotos, escáneres y programas de edición de imagen como Adobe Photoshop, Gimp, etc. Es por este motivo, que este tipo de imágenes pesan más (bytes) que una imagen vectorial (González Arribas, 2017)

Cualquier tipo de imagen para poder utilizarla o visualizarla en cualquier momento es guardada en dispositivos de almacenamiento a través de un conjunto de datos llamados archivos o ficheros. Cada archivo gráfico se puede identificar por medio de su nombre y su extensión, este último indica el formato que posee la imagen determinada (González Arribas, 2017)

Entre los tipos de formatos de imagen vectorial se tiene (González Arribas, 2017):

- AI (Adobe Illustrator):

A través de este programa se puede crear gráficos vectoriales y editarlos, luego de esto, se lo debe exportar a otro archivo para tener una correcta visualización. Cabe recalcar que Adobe Illustrator permite exportar la imagen a diversos formatos para la impresión, pero el formato AI es el propio de este tipo de editor (Beatrizxe, 2016)

- CDR (Corel Draw):

Es el formato por defecto de las imágenes realizadas a través de la aplicación Corel Draw que fue desarrollada en 1987 por primera vez. Incluye apoyo a la gestión de color y colores planos, así como también, un soporte básico para el texto (Zamzar, 2006a)

- DXF (Autodesk):

Drawing Exchange Format es un formato de archivo creado principalmente para habilitar la interoperabilidad entre los archivos DWG de AutoCAD y el resto

de programas existentes en el mercado. En los últimos años se ha reducido su utilidad debido a que no puede soportar los elementos de CAD más avanzados. Los datos que contiene un archivo DXF se conocen con el nombre de datos etiquetados ya que como su nombre lo indica, la etiqueta es la encargada de decir cómo se van a interpretar dichos datos (Zamzar, 2006b)

- EMF (Enhanced Metafile):

Es una versión más reciente de 32 bits, del formato WMF que tenía 16 bits y el cual se lo introdujo en Windows 3.0. Ambos son meta-archivos que poseen un conjunto de funciones de llamadas que ayudan a que una aplicación represente una imagen o un gráfico (Zamzar, 2006c). Adicionalmente cabe mencionar que, EMF es utilizado como lenguaje de gráficos para dispositivos de impresión.

- EPS (Encapsulated PostScript):

Este formato puede incluir vectores e imágenes de mapa de bits. Su uso se lo recomienda únicamente para enviar lo que se necesita a la imprenta o al cliente. Se debe tomar en cuenta que para la impresión de este tipo de formato de imagen se debe tener una impresora específica llamada PostScript (Beatrizxe, 2016)

- ODG (Open Office Draw):

El objetivo de este tipo de archivo desarrollado por OASIS, es el de tener un tipo de archivo gratuito y no propietario para que de esta manera todos puedan utilizarlo. Debido a esto, este tipo de archivo se lo puede abrir mediante los programas de código abierto como OpenOffice.org Draw y LibreOffice Draw (Zamzar, 2006d)

- SVG (Inkscape):

Este tipo de formato pesa muy poco y tiene un estilo CSS, el cual es un lenguaje de programación que da estilo a las páginas web. Su ventaja radica en que puede ser visualizado en navegadores, obteniendo de esta manera una mayor calidad, sin embargo, su desventaja es que está basado en lenguaje de

programación XML y para su correcto funcionamiento se debe tener un conocimiento óptimo del mismo (Beatrizxe, 2016)

- SWF (Adobe flash):

Los archivos Shockwave Flash pueden ser creados por Adobe Animate (antes conocido como Adobe Flash) y pueden ser ejecutados mediante un navegador con un plugin denominado Adobe Flash Player. Admite programación mediante el lenguaje actionscript. Su objetivo principal es la creación de archivos pequeños pero que permitan tener interactividad y que además funcionen con cualquier tipo de plataforma.

- WMF (Microsoft):

Windows Metafile permite inclusión de mapa de bits, aunque actualmente ya no se lo utiliza debido a la aparición de formatos más comunes como los que se mencionarán a continuación para el uso de imágenes de mapa de bits. Este tipo de archivo almacena un conjunto de llamadas a diferentes funciones que se envían al nivel gráfico de Windows denominado GDI para de esta manera llegar a generar la imagen. Es un formato de 16 bits.

Antes de mencionar los formatos que se usan en imágenes de mapa de bits, se debe tomar en cuenta los siguientes parámetros para elegir el más adecuado (González Arribas, 2017):

- El contenido que tiene la imagen ya sea una foto, un dibujo, un gráfico, un logotipo, etc.
- El tamaño que tiene el archivo que aloja la imagen determinada
- La calidad que se desea tener en base al destino en el que se vaya a usar la imagen como, por ejemplo, publicación en la web, impresión en impresora doméstica o profesional, etc.

Dependiendo de estos parámetros se puede elegir entre los siguientes formatos más comunes al trabajar con una imagen de mapa de bits (González Arribas, 2017):

- **BMP:**

Este formato fue introducido por Microsoft y usado generalmente en su sistema operativo llamado Windows para de esta manera almacenar imágenes.

- **GIF:**

Este formato es antiguo, desarrollado por CompuServe para tener archivos de tamaño reducido. Una desventaja es que solo admite 256 colores y por ende no es adecuado para fotografías, pero si es muy usado en logotipos, dibujos, etc. Una ventaja de este formato es que permite crear animaciones y transparencias.

- **JPEG:**

Este formato es uno de los más conocidos y usado para fotos digitales ya que a diferencia del formato anterior, admite millones de colores. Se debe tener en cuenta en este tipo de formato que a más compresión existe menor calidad y por ende archivos más pequeños (menos Kbytes); así como también que, a menos compresión, mayor calidad y archivos más grandes (más Kbytes).

La compresión que realiza este tipo de formato tiene pérdidas y, por lo tanto, afecta a la calidad de imagen. Al abrir el archivo con este formato y manipularlo en algún ordenador, la imagen al realizar la compresión y descompresión ya obtiene un tipo de degradación en base a la imagen original, es por esto que se recomienda no guardarla en formato JPEG si la imagen va a ser modificado o manipulada varias veces.

- PNG:

Este formato fue creado con el objetivo de sustituir a GIF debido a su antigüedad. Tuvo mejoras ya que admite más colores que GIF y posee sistemas de compresión gratuitos, sin embargo, no admite animaciones, pero si transparencias.

- PSD:

Este formato es por defecto al ser utilizado Adobe Photoshop y por ende es adecuado para poder editar imágenes con este tipo de programa y otros que sean compatibles con el mismo. Admite un sin número de colores, capas, canales, etc.

- RAW:

Este tipo de formato está formado por todos los píxeles de la imagen original, tal cual como ha sido captada, por ende, brinda mayor calidad en fotografía y se puede admitir en cámaras de aficionados avanzados y profesionales.

A diferencia de las cámaras que guardan las fotos en otros formatos para procesarlas e interpretarlas, en RAW, los píxeles no son procesados y se mantienen originales para luego mediante un software procesarlos. Al no tener una compresión, se tiene mayor detalle de la imagen y, por ende, se ocupa mayor espacio (Mbytes).

- TIFF:

Este formato es utilizado para el escaneado de imágenes, edición e impresión. Una ventaja es que es compatible con varios sistemas operativos y con diversos tipos de editores de imágenes. Admite millones de colores, capas, canales, así como también, es incluido en diversas cámaras y escáneres.

## 2.7. Calidad en imágenes

Como punto general debemos considerar que existen diversos formatos de archivo de imágenes disponibles como se mencionó anteriormente, cada formato da como resultado diversos tamaños, compresión y calidad del archivo determinado. Para poder determinar cuál es el más óptimo en cuanto a lo que calidad se refiere, se puede mencionar las ventajas y desventajas que posee cada uno para una elección óptima.

El formato JPEG (Joint Photographic Experts Group) posee imágenes a full color ocupando 24 bits de memoria por cada pixel usado, teniendo como resultado alrededor de 16.8 millones de colores en cada imagen guardada con esta extensión (Durán, 2014).

Se debe tomar en cuenta que estas imágenes tienen pérdidas debido a que como está formada por toda la información de color y la compresión hacen que exista como resultado la obtención de pérdida de calidad (Durán, 2014).

Cuando se guarda en este formato, el algoritmo hace un seguimiento de cada línea de píxeles y va tomando decisiones. Cuando encuentra en la misma línea dos píxeles contiguos bastante similares en valores de colorimetría decide cual se guarda y cual se elimina. Al abrir de nuevo la imagen, el pixel que era igual al otro ya no se encuentra, ocupando su lugar un pixel inventado o estimado similar al que estaba a su lado en la imagen visualizada originalmente (Gráfica, 2014).

Este proceso se repite en cada una de las líneas de pixel en donde exista este tipo de similitud ahorrando de esta manera gran cantidad de memoria, sin embargo, hace que la calidad de la imagen se vaya perdiendo (Gráfica, 2014).



En cualquier tipo de programa en donde se maneja este tipo de formato se puede elegir el grado o rango de compresión con el que se desea guardar la imagen teniendo en cuenta que la opción de máxima calidad igualmente altera la calidad de la misma, si bien es cierto, puede ser indistinguible de la fotografía original, pero en su contra poseerá un mayor tamaño de archivo (Durán, 2014).

Al decidir una calidad muy baja lo que se está haciendo es ampliar el rango de pixeles similares que se eliminaran al momento de la compresión lo que da como resultado el perder matices de la imagen y por ende calidad de la misma (Gráfica, 2014)

En la Tabla 3 se muestra tanto los aspectos positivos como negativos de este tipo de formato para archivar imágenes (Durán, 2014):

**Tabla 3**

*Aspectos positivos y negativos del formato JPEG*

<b>Aspectos Positivos:</b>	<b>Aspectos negativos:</b>
<ul style="list-style-type: none"> <li>• Hasta 16 millones de colores</li> <li>• Excelente para fotografías que tengan más de 256 colores</li> <li>• Por lo general es ideal para hacer archivos más pequeños (menor tamaño en bytes)</li> </ul>	<ul style="list-style-type: none"> <li>• Elimina gran cantidad de datos</li> <li>• No puede ser animada</li> <li>• No tiene transparencias</li> <li>• No es bueno para imágenes de texto</li> <li>• Genera pérdida</li> </ul>

Fuente: (Durán, 2014)

Las imágenes en GIF (Graphics Interchange Format) por su parte se utilizan para visualizar gráficos con colores indizados e imágenes para documentos HTML en páginas web, por lo tanto, solo se muestran un máximo de 256 colores y por ende son ideales para ilustraciones y bloques de colores simples como logotipos e íconos (Durán, 2014).

La calidad depende de un buen afinamiento entre el nivel de color y la compresión, teniendo como característica principal la aparición de pequeños puntitos o granulado en la imagen (Gráfica, 2014)

En la Tabla 4 se muestra tanto los aspectos positivos como negativos de este tipo de formato para archivar imágenes (Durán, 2014):

**Tabla 4**

*Aspectos positivos y negativos del formato GIF*

<b>Aspectos Positivos:</b>	<b>Aspectos negativos:</b>
<ul style="list-style-type: none"> <li>• Incluye transparencias</li> <li>• Puede ser animada</li> <li>• Es excelente cuando se tiene una imagen con colores simples o limitados</li> <li>• Excelente para imágenes de texto</li> </ul>	<ul style="list-style-type: none"> <li>• Únicamente soporta 256 colores</li> <li>• Gran tamaño de archivo en bytes</li> <li>• No es adecuado para el uso de fotografías</li> </ul>

Fuente: (Durán, 2014)

La principal ventaja es que como soporta transparencias, al editar la imagen se puede poner sobre un fondo de color o incluso sobre una fotografía y no se tendrá el problema de visualizar bordes alrededor de dicha imagen (Durán, 2014)

En lo que respecta al formato PNG (Portable Network Graphics), se ofrece varias características que atraen a los usuarios como por ejemplo una amplia gama de profundidades de color, soporte para transparencias y correcciones automáticas para monitores (Durán, 2014)

Es posible en este formato tener un pequeño detalle de texto del contenido de la imagen lo que ayuda a la búsqueda de las mismas en base a estas pequeñas descripciones de texto, es así como por ejemplo trabaja Google a partir de este tipo de contexto (Durán, 2014)

PNG al tener imágenes con masas de color actúa mucho mejor que JPEG ya que no genera el típico ruido o distorsión de píxeles que se generan al descomprimir un archivo JPEG (Gráfica, 2014)

En la Tabla 5 se muestra tanto los aspectos positivos como negativos de este tipo de formato para archivar imágenes (Durán, 2014):

**Tabla 5**  
*Aspectos positivos y negativos del formato PNG*

<b>Aspectos Positivos:</b>	<b>Aspectos negativos:</b>
<ul style="list-style-type: none"> <li>• Menos pérdida de calidad</li> <li>• Tamaño de archivo más pequeño (bytes)</li> <li>• Soporta transparencias</li> <li>• Excelente para imágenes de texto</li> <li>• Excelente para logotipos</li> </ul>	<ul style="list-style-type: none"> <li>• No es adecuado en comparación con JPEG para grandes imágenes</li> <li>• No acepta animación</li> <li>• No lo admiten todos los navegadores</li> </ul>

Fuente: (Durán, 2014)

## 2.8. Métodos para análisis de calidad

### 2.8.1. Métodos subjetivos

Estos métodos se encuentran normalizados por la ITU en las recomendaciones ITU BT.500 [ITUR02] para video e ITU P.800 [ITUP96] para audio (Salvador Moreno, 2012). Se hacen a través de encuestas realizadas a un grupo de usuarios en un ambiente controlado, tomando en cuenta el seguir los lineamientos dados por la ITU y las condiciones e interpretaciones de los resultados (Cuéllar, Ortiz, & Arciniegas, 2014)

Se pueden clasificar básicamente en (Salvador Moreno, 2012):

- ACR (Absolute Category Rating) que tiene como resultado el MOS (Mean Opinion Score)
- DCR (Degradation Category Rating) que tiene como resultado el DMOS (Degradation Mean Opinion Score)

En el ACR los usuarios que participan deben asignar un valor general de calidad a la señal, ya sea esta una imagen, un video o un audio, el cual es presentado después de la transmisión sin tener ningún tipo de comparación con la señal original (Salvador Moreno, 2012). Los valores de calidad de este método se muestran en la Tabla 6 a continuación:

**Tabla 6**  
*Valores de calidad de ACR*

<b>Valor de calidad</b>	<b>Calidad de la señal</b>
5	Excelente
4	Buena
3	Regular
2	Pobre
1	Mala

Fuente: (Salvador Moreno, 2012)

El valor medio que den los participantes es lo que se conoce como MOS (Salvador Moreno, 2012). Cabe recalcar que en este método se evalúa la calidad de diferentes muestras y se pondera resultados.

La mejor ventaja que posee este tipo de método es que se evalúa la calidad directamente del usuario que es el beneficiario final, de esta manera se logra obtener los atributos esperados de acuerdo a la medición otorgada. Sin embargo, la desventaja que genera es el costo y el tiempo que conlleva el realizar las encuestas para la población seleccionada (Cuéllar et al., 2014)

Por otro lado, el DCR se encarga de que a los participantes se les muestre dos señales, debiendo de esta manera asignar un valor a la degradación que posee la calidad de una en comparación con la otra (Salvador Moreno, 2012), todo esto se puede evaluar de acuerdo a la Tabla 7

**Tabla 7**  
*Valores de calidad de DCR*

<b>Valor de calidad</b>	<b>Nivel de degradación</b>
5	Imperceptible
4	Perceptible pero no molesta
3	Un poco molesta
2	Molesta
1	Muy molesta

Fuente: (Salvador Moreno, 2012)

El valor medio que asignen los participantes es lo que se conoce como DMOS (Salvador Moreno, 2012)

Lo más común en este método es presentar primero la señal original y luego la señal que ya se transmitió para de esta manera medir de manera adecuada el sistema de comunicación que se utilizó, sin embargo, existen diferentes variantes de acuerdo a como se presenten las señales (Salvador Moreno, 2012)

### **2.8.2. Métodos objetivos**

En los métodos objetivos debemos tomar en cuenta que existe dos maneras para medir la calidad. Una es la que se basa en tener una señal de referencia o comparación para poder medir y la otra depende del análisis que se desee realizar (Cuéllar et al., 2014)

En base a la que requiere el uso de una señal de referencia o comparación existen tres enfoques en los cuales se puede manejar (Cuéllar et al., 2014):

- Referencia completa:

Este método posee una señal original guardada en el receptor para de esta manera ser comparada con la señal que se está recibiendo y poder de esta manera medir la calidad. Por medio de el podemos obtener resultados de correlación y distorsión, pero no una calidad total. Requiere algoritmos complejos y por ende es costoso en tiempo (Rifai, Mohammed, & Mellouk, 2011; Winkler, 2009)

- Referencia reducida:

Este método tiene en el receptor información no completa de la señal original para medir la calidad. Es económico en recursos y tiempo, pero menos preciso (Rifai et al., 2011)

- Sin referencia:

En este método solo se posee la señal recibida para poder determinar la calidad, la cual está basada únicamente en parámetros definidos con anterioridad

para poder obtener degradación útil cuando se hace transmisión en tiempo real (Rifai et al., 2011)

Entre los métodos objetivos más utilizados tenemos:

- PSQM (Perceptual Speech Quality Measurement):

Es un algoritmo computacional definido en la Recomendación ITU-T P.861 (Disqus, 2017), se realiza mediante una comparación entre la señal a evaluar y la señal de origen a través de un dispositivo electrónico. Su escala va desde 0 equivalente en MOS al 5, y el 6.5 equivalente en MOS al 1. Cabe mencionar que existen actualmente software y hardware que facilitan este tipo de pruebas, causando de esta manera un ahorro en costos y tiempo favorables en comparación a los resultados brindados, los cuales son considerados confiables (Disqus, 2017)

Entre las desventajas que posee PSQM están el hecho de que no fue desarrollado desde un principio para poder entender las perturbaciones que existen en servicios de VoIP, pérdidas de paquetes, tardanza en el envío o paquetes no sincronizados, obteniendo de esta manera resultados no favorables que en el equivalente a valores MOS son falsamente inflados (Disqus, 2017)

- PESQ (Perceptual Evaluation of Speech Quality):

Se basa en PSQM mencionado anteriormente, pero mejorando ciertos parámetros, es decir, compara una señal inicial con una señal degradada después de la transmisión en un sistema de comunicaciones (Joskowicz & Sotelo, n.d.), posee un rango que va desde -0.5 a 4.5, aunque en la mayoría de casos la salida es entre 1 y 4.5. Un valor de 4.5 quiere decir que no posee distorsión alguna, es decir que es exactamente igual a la señal presentada como original. Un valor de 1 indica una degradación severa (Konecny, 2017). Como resultado obtenemos una predicción de calidad, la cual es percibida por los usuarios que realizan la prueba subjetiva (Joskowicz & Sotelo, n.d.).

Lo importante en este proceso es que se realiza una transformación de la señal original y de la señal transmitida a través de una representación interna (Joskowicz & Sotelo, n.d.)

- PAMS (Perceptual Analysis Measurement System):

Su objetivo es medir el nivel de esfuerzo necesario para comprender lo que se está evaluando, haciendo comparaciones entre la señal original y el resultante, su escala va del 1 al 5 (Carballar Falcon, 2008)

- RMSE (Root Mean Squared Error):

La raíz del error cuadrático medio es una medida de desempeño cuantitativa, la cual es utilizada generalmente para evaluar métodos de pronóstico de demanda. Consiste en, como dice su nombre, la raíz cuadrada de la sumatoria de los errores cuadráticos., amplificando de alguna manera los errores de mayor magnitud.

El Error Cuadrático Medio mide el promedio de los errores al cuadrado, es decir, la diferencia existente entre el estimador y lo que se estima

- MSSIM:

SSIM (Índice de Similitud Estructural) es una métrica que se basa en que las estructuras de los objetos de las escenas son independientes de la iluminación, de forma que la influencia de esta debe ser aislada en el cálculo de la calidad de la imagen, el promedio de los valores SSIM se conoce como MSSIM

Este índice universal es un método para medir la similitud entre dos imágenes al tomar una de ellas como referencia, para esto, consta de tres componentes que son la comparación de luminancia, distorsión del contraste y similitud estructural, proporcionando de esta manera una similitud más completa

que otros índices como el RMSE mencionado anteriormente (Sola, González-Audicana, Álvarez-Mozos, & Torres, 2014)

Se obtiene como resultado valores entre -1 y 1, siendo los más cercanos a 1 los que poseen mayor similitud entre ambas imágenes (Sola et al., 2014)

## **2.9. Infraestructura de la TDT**

Un sistema de TDT posee los siguientes elementos; en cuanto a la infraestructura del proveedor tenemos el servidor de aplicaciones y contenidos y el servidor de playout y en la infraestructura del usuario se posee de un decodificador o set top box

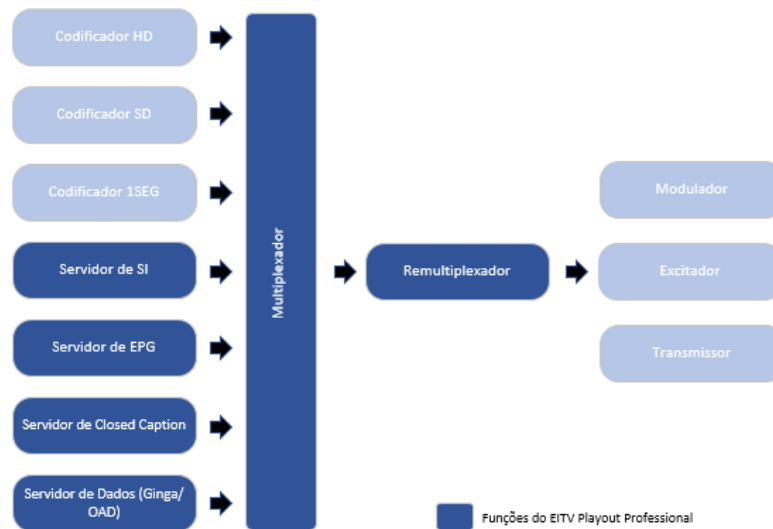
### **2.9.1. Servidor de aplicación y contenidos**

Este servidor guarda las aplicaciones de NCL que se transmitirán a la par del contenido audiovisual. Dentro de este equipo debe almacenarse de manera adicional, el software para desarrollar las aplicaciones interactivas (Dávila Sacoto, 2012)

### **2.9.2. Servidor de playout**

A través de este equipo se realiza la modulación y transmisión de los contenidos que posee la televisión digital. Crea el flujo de datos de transporte (TS) a través del uso del formato MPEG-2. Posee la capacidad de unir la aplicación NCL con el flujo de datos de audio y video. Un servidor de playout posee la siguiente funcionalidad mostrada en la Figura 18 a continuación (Dávila Sacoto, 2012).





**Figura 18.** Funcionalidad de un servidor plegado.  
Fuente: (Dávila Sacoto, 2012)

### 2.9.3. Set top box

Equipo cuya funcionalidad es la de recibir la señal a través de la antena del receptor, decodificarla y mostrarla en un televisor. Al utilizar el estándar ISDB-Tb, se tiene precargado en este dispositivo el middleware Ginga para poder visualizar las aplicaciones interactivas (Dávila Sacoto, 2012)



**Figura 19.** Set top box  
Fuente: (Dávila Sacoto, 2012)

## **CAPÍTULO III**

### **3. DISEÑO E IMPLEMENTACIÓN**

#### **3.1. Técnica mejorada de transmisión de imágenes segura a través de imágenes en mosaico mediante transformaciones de color casi reversibles**

Las imágenes que se visualizarán en la aplicación de Ginga, la cual se explicará su diseño posteriormente, fueron obtenidas mediante este tipo de técnica que fue desarrollada como se indica en (Acosta, 2017), de la cual se dará una breve explicación.

Si bien es cierto hoy en día los sistemas de comunicación que trabajan con multimedia intentan transmitir varios tipos de imágenes a través de un solo flujo de datos, para lo cual se requiere de un ancho de banda óptimo sin verse involucrado en esto, la pérdida de seguridad o de información. Recientemente se está usando lo que se refiere a imagen mosaico contenida por fragmentos secretos, es decir, mediante ésta se puede ocultar una imagen llamada secreta. La técnica empleada se basa en reducir de manera estadística el número de bits que se necesitan para poder recuperar la imagen oculta o secreta utilizando la imagen mosaico, obteniendo como resultado una simplificación de 3 a 1 (Acosta, 2017)

Se da a conocer un enfoque de tipo estadístico para de esta manera obtener la desviación estándar de la imagen secreta con sus respectivos parámetros estadísticos y estructurales de la imagen que se tiene por objetivo. Al conocer estos parámetros y la clave secreta, se puede recuperar sin ningún problema la imagen secreta sin obtener mucha pérdida al hacerlo para de esta manera reducir el ancho de banda requerido al transmitir dicha imagen (Acosta, 2017)

El método se basa en el desarrollado por (Lee & Tsai, 2014) donde se propone una técnica de transmisión de imágenes a través de Transformación de colores casi reversibles (NRCT) para de esta manera transformar una imagen

llamada secreta en una imagen mosaico, las cuales poseen el mismo tamaño. Para lograr el camuflaje mencionado, se divide la imagen secreta en bloques para poder transformar sus características de color haciéndolas similares a la imagen que se tiene por objetivo (Acosta, 2017)

Los datos de la imagen secreta vinculados con los índices de posición del bloque, los valores medios y los ángulos de rotación se generan tal como en (Lee & Tsai, 2014). Los datos de la imagen destino que se requieren no se encuentran en la secuencia de bits que se va a transmitir, sino que más bien serán estimados al momento de la recepción. Mediante la pdf (función de densidad de probabilidad exponencial) se puede llegar a estimar la desviación estándar de los bloques que posee la imagen secreta, reduciendo el número de bits a transmitir. Cabe mencionar que para que todo esto sea factible, el receptor debe ser capaz de poder reconstruir la imagen secreta por sí solo (Acosta, 2017)

Todo el proceso consta de dos fases, la una es la creación de imágenes en mosaico acompañada del ensamblaje del flujo de datos y la otra es la recuperación de las imágenes secretas. En lo que respecta la primera fase, se produce una imagen mosaico que incluirá la secuencia de datos que será obtenida en la recepción, esta imagen está formada por bloques de una imagen secreta que va a la entrada con NRCT y flujo de datos. Dentro de esta fase se realiza cuatro subprocesos, primeramente se identifican los bloques similares en comparación de la imagen secreta con la imagen objetivo, después se debe rotar cada bloque de la imagen secreta hasta obtener el mínimo error cuadrático medio (RMSE) en base al bloque de la imagen objetivo que corresponda para de esta manera hacer la transformación de color de cada uno de los bloques, luego se simplifica todo lo que tiene que ver con la desviación estándar en cada bloque de la imagen secreta por medio del uso de la pdf exponencial y por último se obtiene el flujo de datos para recuperar la imagen con la ayuda del receptor (Acosta, 2017)

Con la ayuda de la explicación mencionada anteriormente se hizo uso del algoritmo para la obtención de las imágenes que se evaluarán posteriormente.

El algoritmo diseñado consta de cinco funciones y dos programas de los cuales nos vamos a enfocar únicamente en el programa principal y el programa de recuperación de imágenes secretas debido a que es en el programa principal en donde se ingresa las imágenes a ser procesadas (secreta y portadora) así como también el número de bloques; por otro lado, el programa de recuperación de imágenes como su nombre lo dice, nos permite recuperar la imagen secreta a partir de la imagen mosaico. Por ende, con la ayuda de estos dos programas obtenemos las imágenes que posteriormente serán evaluadas.

Las funciones mencionadas anteriormente son las siguientes:

- Angrot: Toma cada uno de los bloques en los cuales se ha dividido la imagen secreta para girarlos en 4 posiciones distintas que van desde los  $0^\circ$  hasta los  $270^\circ$ , cada giro lo va comparando con el bloque correspondiente de la imagen portadora calculando entre ellos el RMSE y en el giro que se obtiene el menor valor, con ese se queda.
- Angrec: Realiza el mismo procedimiento que angrot pero a la inversa, de manera que se pueda recuperar la imagen secreta a partir de la imagen mosaico
- Construir4: Se realiza la conversión de color a través de los principios NRCT y la fórmula correspondiente para calcular los valores de cada uno de los componentes en los píxeles por medio del uso del coeficiente de desviación estándar y los valores medios de cada bloque tanto de la imagen secreta como de la imagen portadora respectivamente.
- Matdiv: Obtiene el valor divisible e igual de los ejes de una imagen teniendo como variables de entrada el tamaño del eje x, el tamaño del eje y, una imagen y el número de partes en que se va a dividir la imagen (N). Entrega una matriz de los bloques rectangulares en que se ha dividido la imagen.

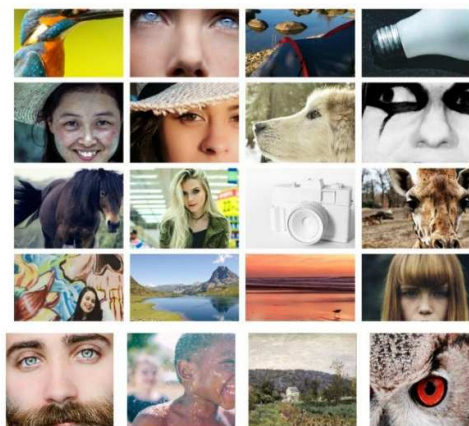
- Ordenar2: Ordena las imágenes de menor a mayor valor de desviación estándar de cada uno de los bloques en que se la dividió. Almacena los índices originales para luego poder recuperar las posiciones de dichos bloques.

En el programa principal las líneas de código que se iban cambiando son las siguientes:

```
S = imread('Imgs/TT.jpg'); % Imagen Secreta
P = imread('Imgs/barras.jpg'); % Imagen Portadora
```

En donde en S se va a leer la imagen secreta guardada en la carpeta Imgs dentro de la carpeta principal en donde se encuentra el programa, al igual que en P se va a leer la imagen portadora que se encuentra en el mismo directorio.

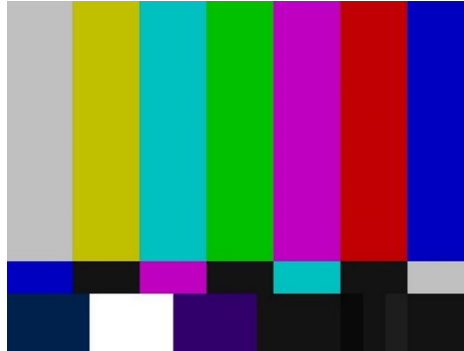
Las imágenes secretas fueron seleccionadas al azar, obteniendo un total de 20 imágenes para su posterior procesamiento. En la Figura 20 se muestran las imágenes que fueron utilizadas.



**Figura 20.** Imágenes a ser procesadas

Cada una de ellas tenía un nombre específico, el cual iba desde la AA hasta la TT, su tamaño es de 1024 x 768 píxeles y se las guardó en formato .jpg debido a las razones que se explicarán posteriormente.

En lo que respecta a la imagen portadora, fue escogida una sola para todo el procesamiento y fue la imagen de barra de colores que ocasionalmente aparece cuando la transmisión de un canal de televisión finaliza. En la Figura 21 se puede visualizar la imagen escogida como portadora.



**Figura 21.** Imagen portadora

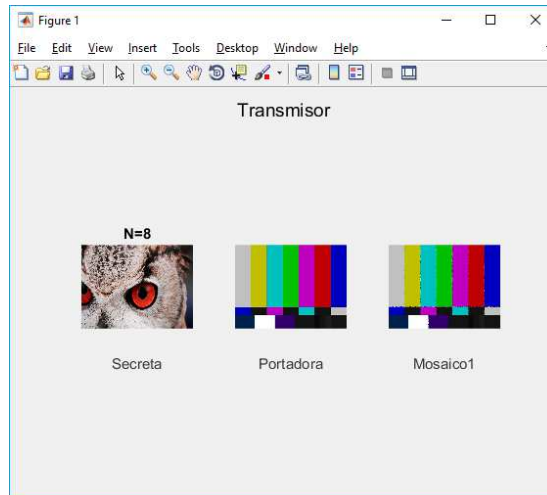
Para el correcto procesamiento se debió redimensionar la imagen en cuestión debido a que no se encontraba en el mismo tamaño de las imágenes secretas (1024 x 768), para lo cual se utilizaron las siguientes líneas de código en Matlab

```
S = imread('Imgs/barras.jpg');  
K = imresize(S, [768 1024])  
imwrite(K, 'barras.jpg')
```

En S como se explicó anteriormente, se lee la imagen de las barras obtenidas originalmente que se encuentra dentro de la carpeta principal del programa para luego en K redimensionarla a los pixeles necesarios y de esta manera guardarla en formato .jpg con las dimensiones correspondientes.

Al correr el programa principal tenemos la visualización de la siguiente pantalla como resultado, la cual se muestra en la Figura 22, en donde se encuentra

la imagen portadora, la imagen secreta y su respectiva imagen mosaico creada. El valor de N indica el número de bloques en que se dividieron las imágenes.



**Figura 22.** Ventana de resultado al correr el programa principal del algoritmo empleado

Cuando esta pantalla se ha visualizado se puede comprobar que dentro de la carpeta en donde se encuentra el programa principal se ha creado un archivo denominado Dc1532 en donde esta guardada la imagen mosaico como matriz y los parámetros necesarios a ser utilizados al momento de recuperar la imagen secreta como se muestra en la Figura 23 a continuación.

Imgs	14/11/2017 21:14	Carpeta de archivos	
angRec.m	13/12/2016 5:52	Archivo M	1 KB
angRot.m	05/11/2016 23:35	Archivo M	1 KB
barras.m	14/11/2017 21:12	Archivo M	1 KB
construir4.m	29/03/2017 21:34	Archivo M	1 KB
Dc1532	04/01/2018 21:02	Microsoft Access ...	499 KB
matdiv.m	27/03/2017 19:41	Archivo M	2 KB
ordenar2.m	27/03/2017 19:52	Archivo M	1 KB
recuperar_v0FR.m	23/11/2017 11:53	Archivo M	2 KB
V1programa_completo_V0FR.m	23/11/2017 11:49	Archivo M	3 KB

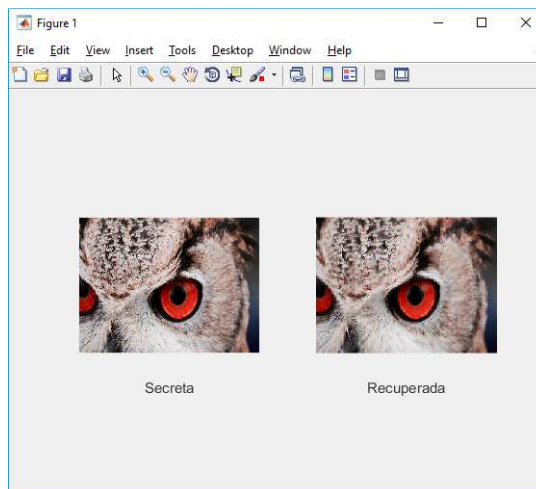
**Figura 23.** Creación del archivo Dc1532

Al programa de recuperación de imágenes secretas se le añadió la siguiente línea de código

```
imwrite(Recuperada,'recuperada.jpg','quality',0)
```

Por medio de la cual, se guarda la imagen, que momentáneamente se encuentra en forma de matriz (Recuperada), en formato .jpg dándole un valor de calidad el cual puede variar de 0 a 100, donde 0 significa menor calidad y mayor compresión y 100 es de mayor calidad con menor compresión, cabe recalcar que por default el valor es 75 (MathWorks, 1994).

Al correr el programa mencionado anteriormente, se obtiene la visualización de la siguiente pantalla, como se indica en la Figura 24, donde se muestra ambas imágenes, la secreta original y la secreta recuperada después del procesamiento. Cabe recalcar, que la calidad de la imagen solo se visualiza en la imagen guardada, es decir, en el archivo que se genera con formato .jpg en la carpeta en donde se encuentra el programa como se muestra en la Figura 25



**Figura 24.** Ventana de resultado al correr el programa de recuperación de imágenes del algoritmo empleado



Imgs	14/11/2017 21:14	Carpeta de archivos	
angRec.m	13/12/2016 5:52	Archivo M	1 KB
angRot.m	05/11/2016 23:35	Archivo M	1 KB
barras.m	14/11/2017 21:12	Archivo M	1 KB
construir4.m	29/03/2017 21:34	Archivo M	1 KB
Dc1532	04/01/2018 21:02	Microsoft Access ...	499 KB
matdiv.m	27/03/2017 19:41	Archivo M	2 KB
ordenar2.m	27/03/2017 19:52	Archivo M	1 KB
recuperada	04/01/2018 21:23	Archivo JPG	16 KB
recuperar_v0FR.m	23/11/2017 11:53	Archivo M	2 KB
V1programa_completo_V0FR.m	23/11/2017 11:49	Archivo M	3 KB

**Figura 25.** Creación del archivo .jpg

El procedimiento mencionado, se realizó para cada una de las 20 imágenes con cada uno de los valores de calidad determinados y por ende se realizaron 4 aplicaciones interactivas, la diferencia de cada una de ellas era el valor de calidad que las imágenes pre-procesadas iban a poseer. Cada una de las imágenes obtenidas como resultado se fueron almacenando en una carpeta para posteriormente añadirlas en la carpeta multimedia de cada una de las aplicaciones diseñadas en Ginga.

### 3.2. Diseño de la aplicación en Ginga

La aplicación interactiva diseñada permite visualizar las imágenes pre-procesadas en diferentes tamaños y regiones. Cabe mencionar que una región es el lugar en donde se va a mostrar el elemento multimedia, es decir, representa como su nombre lo dice, una región de la pantalla y se la puede definir mediante porcentaje o en píxeles.

El diseño, como ya se mencionó anteriormente, consiste en la visualización de cada una de las 20 imágenes, dividiéndolas en grupos de 5 para cada uno de los botones del control remoto utilizados de la siguiente manera:

- Botón rojo: dentro de este botón se encontrarán las 5 primeras imágenes ocupando el 20% de la pantalla

- Botón verde: dentro de este botón se encontrarán las 5 siguientes imágenes ocupando el 40% de la pantalla
- Botón amarillo: dentro de este botón se encontrarán las 5 siguientes imágenes ocupando el 3,5 : 2 de la pantalla
- Botón azul: dentro de este botón se encontrarán las últimas 5 imágenes ocupando el 4,5 : 3 de la pantalla

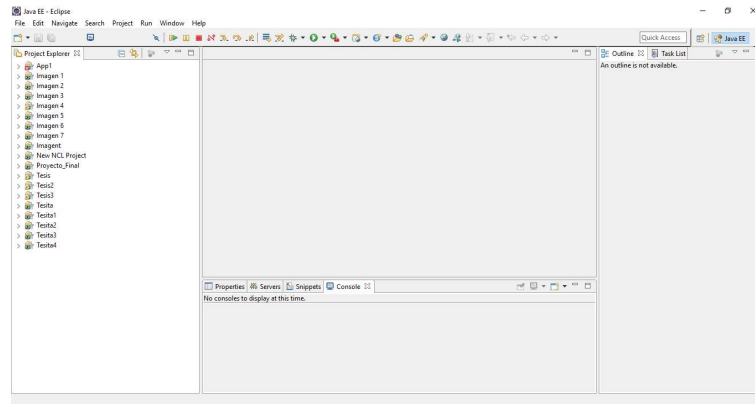
El porcentaje de ocupación de la pantalla en cada uno de los casos se refiere a la imagen que se muestra en la esquina superior izquierda de la pantalla ya que en la parte inferior de la misma se muestra las 5 imágenes pertenecientes al botón seleccionado, entre las cuales el usuario puede navegar aplastando los botones hacia la derecha o hacia la izquierda del control remoto. La imagen posicionada en el centro de la parte inferior de la pantalla es la imagen que aparece en el tamaño indicado en la esquina superior izquierda y a medida que se va navegando con las flechas las imágenes que se vayan posicionando en este lugar será las que se vayan cambiando en la parte superior.

En cuanto al video que se utiliza, el cual representa la programación que normalmente se visualizaría en un televisor al mostrar la pantalla principal estará mostrado en toda la pantalla y los botones de los controles se visualizarán en la esquina inferior derecha, al momento de seleccionar uno de los botones con el control remoto el video se redimensiona a un 50% de la pantalla.

Al seleccionar cada uno de los botones se puede volver al menú principal aplastando la flecha hacia arriba, de esta manera el video que se encontraba redimensionado al 50% de la pantalla volverá a ocupar el 100% de la misma y el usuario tendrá la posibilidad de escoger otro botón.

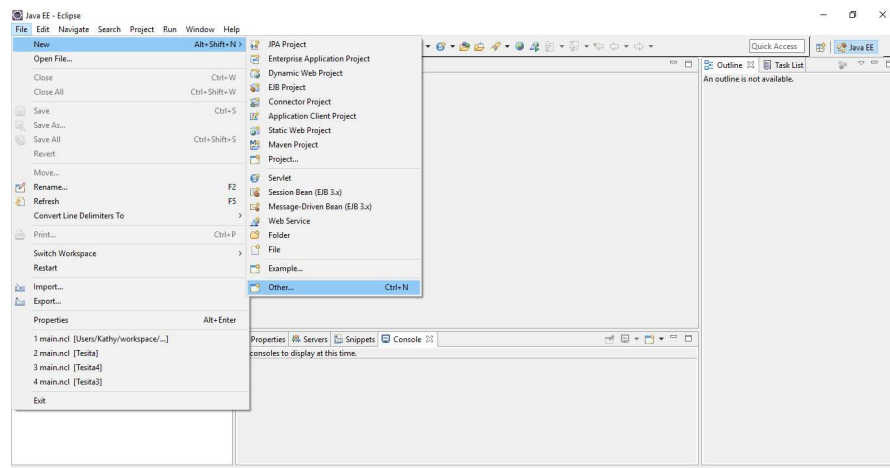
A continuación, se mencionará paso a paso como se fueron diseñando las aplicaciones interactivas por medio de la ayuda de Eclipse y su herramienta NCL.

1. Se cuenta con la ayuda del programa Eclipse, el cual es una plataforma de software que posee varias herramientas de programación con código abierto. La pantalla al iniciar el programa se indica en la Figura 26.



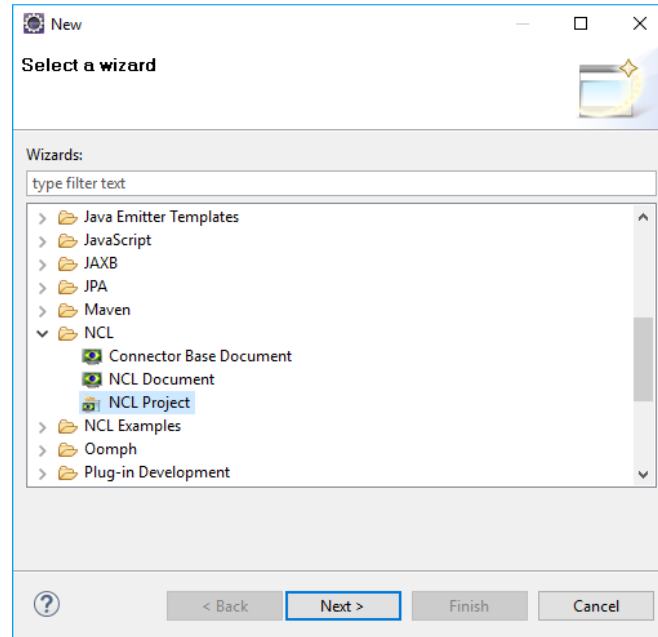
**Figura 26.** Pantalla principal de Eclipse Mars

2. Para crear un nuevo proyecto se dirige a la pestaña File, se escoge New y posteriormente se da click en Other... como muestra la Figura 27



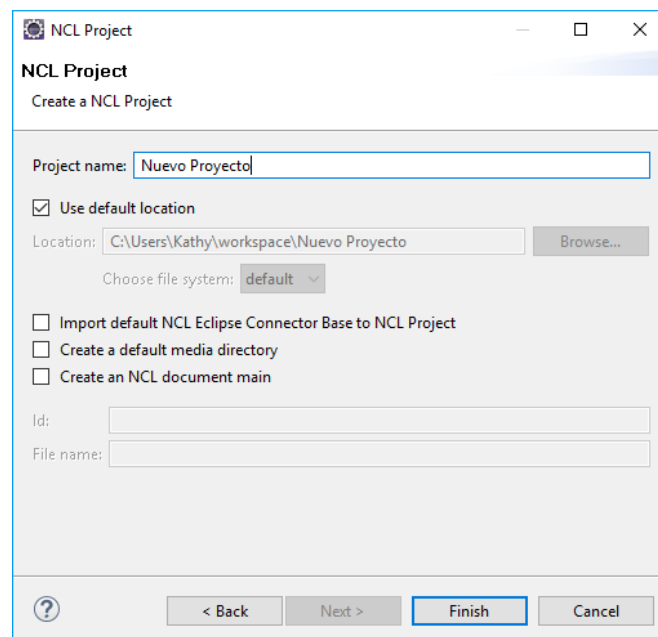
**Figura 27.** Creación de un nuevo proyecto NCL

3. La ventana que aparecerá a continuación se muestra en la Figura 28 en donde se escogerá dentro de la carpeta NCL la opción NCL Project y se da click en Next >



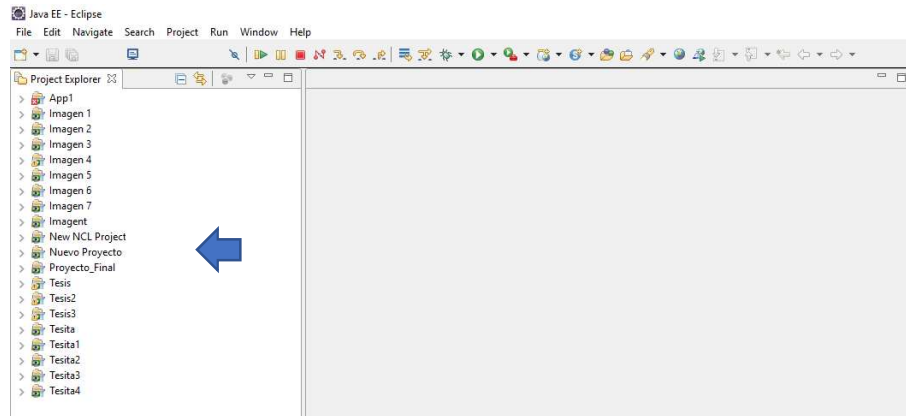
**Figura 28.** Selección de nuevo proyecto NCL

4. Posteriormente se visualizará una nueva pantalla, como en la Figura 29, la cual nos permitirá poner el nombre que se le quiere dar al proyecto y finalmente se dará click en Finish



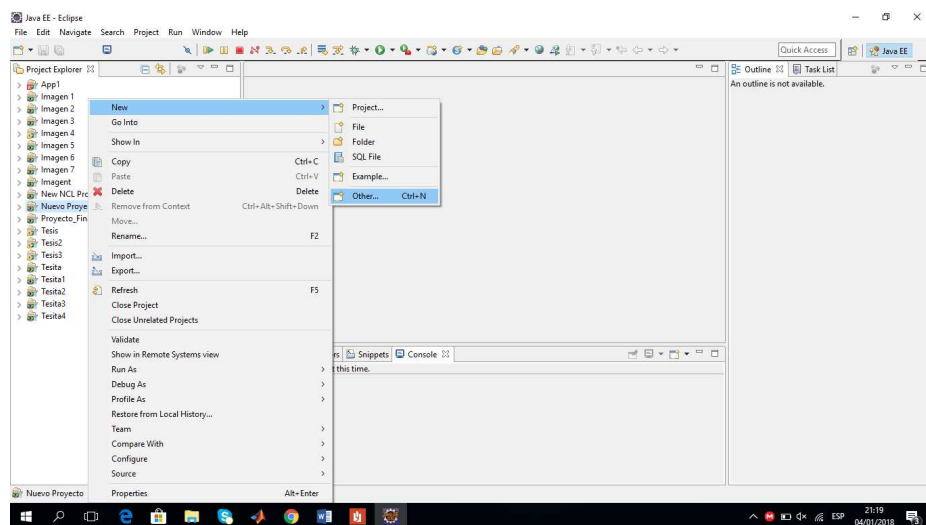
**Figura 29.** Configuración del nombre del proyecto

5. Se puede ver que, en la pantalla principal del programa, en la parte izquierda, ya se ha creado el nuevo proyecto que hemos elegido, Figura 30



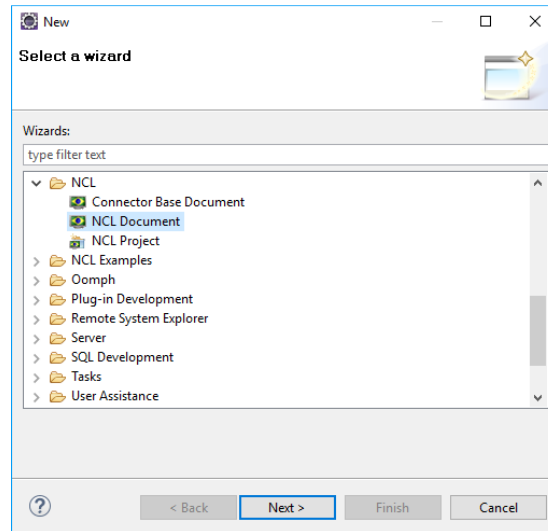
**Figura 30.** Visualización de la correcta creación del nuevo proyecto NCL

6. Para crear un nuevo documento dentro del proyecto únicamente damos click derecho en el proyecto creado, escogemos New y luego Other..., tal como en la Figura 31



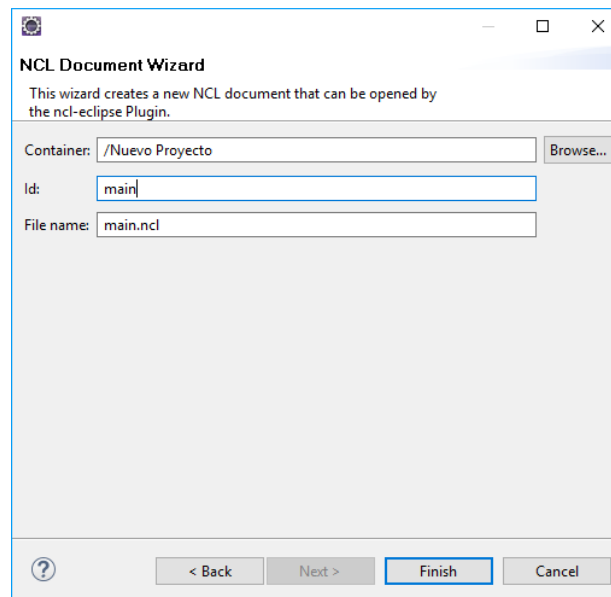
**Figura 31.** Creación de un documento nuevo dentro del proyecto creado

7. La Figura 32 muestra la pantalla que saldrá a continuación en la cual escogeremos NCL Document y damos click en Next >



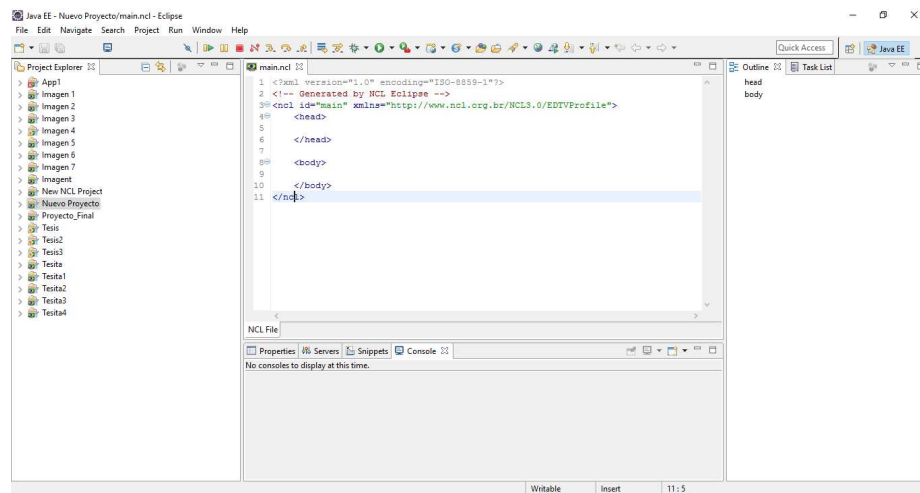
**Figura 32.** Pantalla donde es escoge el tipo de documento a crear

8. En el campo Id ponemos el nombre que deseamos para nuestro documento y damos click en Finish, Figura 33. Por lo general se le da el nombre de main ya que es el documento principal en donde se va a programar



**Figura 33.** Configuración del nombre del documento

9. Una vez generado el documento NCL en la pantalla principal del programa podemos ver el código que por default se genera también, tal como en la Figura 34. Podemos visualizar que se ha creado las dos partes importantes de las cuales está constituido, el head o cabecera y el body o cuerpo. En la cabecera van las regiones (lugar en donde se va a mostrar), descriptores (la manera en que se va a mostrar) y conectores (cuando se va a mostrar), mientras que en el cuerpo van las puertas (lugar por donde ingresará), contextos (como se puede reutilizar el código), elementos multimedia (que es lo que se va a mostrar) y enlaces (cuando se va a mostrar) que se vayan a utilizar



**Figura 34.** Generación del código por default al crear el documento NCL

10. Con la ayuda de la explicación dada anteriormente se empezó a programar para poder conseguir todos los requerimientos necesarios con el objetivo de una correcta visualización de cada una de las imágenes pre-procesadas. Primeramente, se declaró todo lo que tiene que ver con las regiones que se van a utilizar para la visualización de los diversos elementos multimedia, todo esto va en el head como se indica en la Figura 35

```

<head>
<regionBase>
<region id="z0" right="0%" top="0%" height="100%" width="100%" zIndex="1"/>

<region id="z1" right="0%" top="60%" height="10%" width="25%" zIndex="2"/>
<region id="z2" right="0%" top="70%" height="10%" width="25%" zIndex="2"/>
<region id="z3" right="0%" top="80%" height="10%" width="25%" zIndex="2"/>
<region id="z4" right="0%" top="90%" height="10%" width="25%" zIndex="2"/>

<region id="z5" right="0%" left="46%" top="93%" height="5%" width="3%" zIndex="2"/>
<region id="z6" right="0%" left="51%" top="93%" height="5%" width="3%" zIndex="2"/>

<region id="z7" right="0%" left="73%" top="92.5%" height="7.5%" width="27%" zIndex="2"/>

<region id="z8" right="0%" left="5%" top="70%" height="20%" width="15%" zIndex="2"/>
<region id="z9" right="0%" left="22.5%" top="70%" height="20%" width="15%" zIndex="2"/>
<region id="z10" right="0%" left="40%" top="65%" height="25%" width="20%" zIndex="2"/>
<region id="z11" right="0%" left="62.5%" top="70%" height="20%" width="15%" zIndex="2"/>
<region id="z12" right="0%" left="80%" top="70%" height="20%" width="15%" zIndex="2"/>

<region id="z13" width="20%" height="20%" left="15%" top="20%" zIndex="2"/>
<region id="z14" width="40%" height="40%" left="5%" top="10%" zIndex="2"/>
<region id="z15" width="35%" height="20%" left="7.5%" top="20%" zIndex="2"/>
<region id="z16" width="45%" height="30%" left="2.5%" top="15%" zIndex="2"/>

</regionBase>

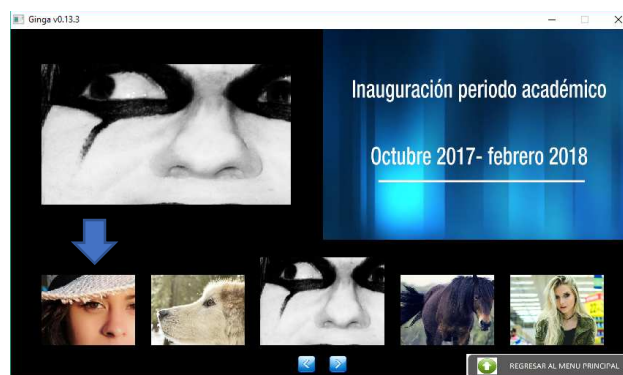
```

**Figura 35.** Declaración de regiones a utilizarse

A continuación, se explicará una de las regiones con sus respectivos parámetros, las demás se las realiza de igual manera dependiendo de la ubicación que se desee en la pantalla:

Región r8: imagen 1 para la parte inferior (carrusel de imágenes), declarada con un alto del 20%, un ancho del 15%, un espacio entre el margen izquierdo de la pantalla del 5%, un espacio entre el margen superior de la pantalla del 70% y un espacio entre el margen derecho de la pantalla del 0%.

La Figura 36 muestra en el caso de la región r8 en donde estaría ubicado el elemento multimedia que corresponde a la explicación mencionada anteriormente.



**Figura 36.** Región r8 programada para la primera imagen en el carrusel de imágenes



11. Inmediatamente después han sido declarados los descriptores como se visualiza en la Figura 37, igualmente en el head del documento

```
<descriptorBase>
  <descriptor id="d0" region="r0"/>

  <descriptor id="d1" region="r1" />
  <descriptor id="d2" region="r2" />
  <descriptor id="d3" region="r3" />
  <descriptor id="d4" region="r4" />

  <descriptor id="d5" region="r5" />
  <descriptor id="d6" region="r6" />

  <descriptor id="d7" region="r7" />

  <descriptor id="d8" region="r8" />
  <descriptor id="d9" region="r9" />
  <descriptor id="d10" region="r10" />
  <descriptor id="d11" region="r11" />
  <descriptor id="d12" region="r12" />

  <descriptor id="d13" region="r13" />
  <descriptor id="d14" region="r14" />
  <descriptor id="d15" region="r15" />
  <descriptor id="d16" region="r16" />

</descriptorBase>
```

**Figura 37.** Declaración de descriptores a utilizarse

Cada región que se vaya a utilizar, debe ir ligada a un descriptor correspondiente, es por ello que existe igual número de regiones como de descriptores.

12. En la Figura 38 se muestra que por último en lo que respecta al head va la declaración del Conector Base que se va a utilizar el cual es proporcionado ya por default por parte del programa de Eclipse, por ende, solo se realiza la importación del mismo

```
<connectorBase>
  <importBase documentURI="ConnectorBase.nc1" alias="conector"/>
</connectorBase>

</head>
```

**Figura 38.** Importación del conector base

13. Dentro del body se comenzó con la declaración de cada uno de los elementos multimedia que se van a utilizar como se muestra en la Figura 39

```

<body>

  <port id="p0" component="m0"/>
  <media id="m0" src="media/V1.mp4" descriptor="d0">
    <area id="inf" begin="1s"/>
    <property name="bounds"/>
  </media>

  <media id="mrojo" src="media/red.png" descriptor="d1">
    <property name="transparency" value="40%"/>
  </media>

  <media id="mverde" src="media/green.png" descriptor="d2">
    <property name="transparency" value="40%"/>
  </media>

  <media id="mamarillo" src="media/yellow.png" descriptor="d3">
    <property name="transparency" value="40%"/>
  </media>

  <media id="mazul" src="media/blue.png" descriptor="d4">
    <property name="transparency" value="40%"/>
  </media>

  <media id="mizquierda" src="media/left.png" descriptor="d5" />
  <media id="mderecha" src="media/right.png" descriptor="d6" />

  <media id="mregreso" src="media/flecha.png" descriptor="d7"/>

  <media id="m1.1" src="media/i1.jpg" descriptor="d8" />
  <media id="m1.2" src="media/i2.jpg" descriptor="d9" />
  <media id="m1.3" src="media/i3.jpg" descriptor="d10" />
  <media id="m1.4" src="media/i4.jpg" descriptor="d11" />
  <media id="m1.5" src="media/i5.jpg" descriptor="d12" />

  <media id="m1.6" src="media/i6.jpg" descriptor="d8" />
  <media id="m1.7" src="media/i7.jpg" descriptor="d9" />
  <media id="m1.8" src="media/i8.jpg" descriptor="d10" />
  <media id="m1.9" src="media/i9.jpg" descriptor="d11" />
  <media id="m1.10" src="media/i10.jpg" descriptor="d12" />

  <media id="m1.11" src="media/i11.jpg" descriptor="d8" />
  <media id="m1.12" src="media/i12.jpg" descriptor="d9" />
  <media id="m1.13" src="media/i13.jpg" descriptor="d10" />
  <media id="m1.14" src="media/i14.jpg" descriptor="d11" />
  <media id="m1.15" src="media/i15.jpg" descriptor="d12" />

  <media id="m1.16" src="media/i16.jpg" descriptor="d8" />
  <media id="m1.17" src="media/i17.jpg" descriptor="d9" />
  <media id="m1.18" src="media/i18.jpg" descriptor="d10" />
  <media id="m1.19" src="media/i19.jpg" descriptor="d11" />
  <media id="m1.20" src="media/i20.jpg" descriptor="d12" />

  <media id="m7.1" src="media/i3.jpg" descriptor="d13" />
  <media id="m7.2" src="media/i8.jpg" descriptor="d14" />
  <media id="m7.3" src="media/i13.jpg" descriptor="d15" />
  <media id="m7.4" src="media/i18.jpg" descriptor="d16" />

```

**Figura 39.** Declaración de los elementos multimedia

A continuación, se explicará uno de los elementos multimedia con sus respectivos parámetros, los demás se los realiza de igual manera dependiendo de lo que se desee mostrar en la pantalla:

Media mverde: botón verde en formato png, ligado al descriptor d2, al cual se le dio una transparencia del 40%

La Figura 40 muestra en el caso del id mverde que clase de multimedia se va a visualizar en la pantalla.



**Figura 40.** Multimedia mverde programado para el botón verde

14. Para mostrar el MENU principal al momento de correr la aplicación diseñada se utilizó el conector especificado en la Figura 41, el cual ayuda a que, al iniciar el video, se inicie también los botones en sus respectivas regiones declaradas anteriormente.

```

<!--..... mostrar MENU .....-->

<link xconnector="conector#onBeginStartN" id="vid">
  <bind role="onBegin" component="m0" interface="inf"/>
  <bind role="start" component="mazul"/>
  <bind role="start" component="mamarillo"/>
  <bind role="start" component="mrojo"/>
  <bind role="start" component="mverde">

  </bind>
</link>

```

**Figura 41.** Conector para mostrar el MENU principal

15. Para el redimensionamiento de la pantalla se utilizó el conector de la Figura 42, el cual ayuda a que al momento de seleccionar una tecla del control remoto sea esta, rojo, verde, amarillo o azul, deje de mostrar los demás botones y redimensione el video a través de la propiedad bound, ocupando ahora el 50% de la pantalla para que en la esquina superior izquierda se visualice la imagen pre-procesada y en la parte inferior se visualice el carrusel de imágenes. En este caso se muestra el redimensionamiento del botón verde pero para los demás es exactamente igual con la única condición de que en vez de value="GREEN" se va cambiando el color correspondiente a cada botón con su nombre en inglés.

```

<!--..... redimensionar GREEN .....-->

<link xconnector="conector#onKeySelectionSetNStopN" id="verdedos">
  <bind role="onSelection" component="mverde"/>
  <linkParam name="keyCode" value="GREEN"/>
  <bind role="stop" component="mverde"/>
  <bind role="stop" component="mazul"/>
  <bind role="stop" component="mamarillo"/>
  <bind role="stop" component="mrojo"/>

  <bind role="set" component="m0" interface="bounds">
    <bindParam name="var" value="50%,0%,50%,60%"/>
  </bind>
</link>

```

**Figura 42.** Redimensionamiento de la pantalla al presionar un botón

16. La programación en cada uno de los botones se la realiza como en la Figura 43, en donde en este caso del botón rojo se utiliza un conector que permite que

al seleccionar la tecla roja en el control remoto, se inicie el contexto 1 que se explicara posteriormente, se inicie el carrusel de imágenes al igual que la visualización de la imagen en la esquina superior izquierda que es la imagen pre-procesada, se iniciaran las flechas hacia la derecha e izquierda así como también la flecha hacia arriba para regresar al MENU principal y por el contrario dejarán de visualizarse los demás botones que no han sido seleccionados, los demás contextos que son de los otros botones y las demás imágenes que serán visualizadas al presionar un botón de otro color. Este procedimiento se repite para cada uno de los botones cambiando lo que se debe o no se debe visualizar en cada caso.

```

<!--..... boton ROJO .....-->

<link xconnector="conector#onKeySelectionStopNStartN" id="rojo">
  <bind role="onSelection" component="mrojo"/>
  <linkParam name="keyCode" value="RED"/>
  <bind role="start" component="contexto1"/>
  <bind role="start" component="m1.1"/>
  <bind role="start" component="m1.2"/>
  <bind role="start" component="m1.3"/>
  <bind role="start" component="m1.4"/>
  <bind role="start" component="m1.5"/>
  <bind role="start" component="m7.1"/>
  <bind role="start" component="mregreso"/>
  <bind role="start" component="mizquierda"/>
  <bind role="start" component="mderecha"/>
  <bind role="stop" component="mrojo"/>
  <bind role="stop" component="mverde"/>
  <bind role="stop" component="mazul"/>
  <bind role="stop" component="mamarillo"/>
  <bind role="stop" component="contexto2"/>
  <bind role="stop" component="contexto3"/>
  <bind role="stop" component="contexto4"/>

  <bind role="stop" component="m1.6"/>
  <bind role="stop" component="m1.7"/>
  <bind role="stop" component="m1.8"/>
  <bind role="stop" component="m1.9"/>
  <bind role="stop" component="m1.10"/>
  <bind role="stop" component="m1.11"/>
  <bind role="stop" component="m1.12"/>
  <bind role="stop" component="m1.13"/>
  <bind role="stop" component="m1.14"/>
  <bind role="stop" component="m1.15"/>
  <bind role="stop" component="m1.16"/>
  <bind role="stop" component="m1.17"/>
  <bind role="stop" component="m1.18"/>
  <bind role="stop" component="m1.19"/>
  <bind role="stop" component="m1.20"/>
</link>

```

**Figura 43.** Programación para cada uno de los botones dependiendo de lo que se debe o no se debe visualizar en cada caso

17. La tecla que permite regresar al MENU principal es la de la flecha hacia arriba y en este caso se muestra la programación que se utilizó en la Figura 44, en donde por medio de la ayuda de un conector se puede hacer que al presionar la

tecla de la flecha hacia arriba primeramente el video se redimensione para que ocupe de nuevo el 100% de la pantalla, después se hace que se visualice de nuevo todos los botones para que el usuario pueda escoger una nueva opción y por último se hace que se deje de visualizar el carrusel de imágenes, las imágenes pre-procesadas que se muestran en la esquina superior izquierda y los diferentes contextos

```

<!--..... regresar tecla MENU .....-->

<link xconnector="conector#onKeySelectionStartNSetN" id="regresomenu1">
  <bind role="onSelection" component="mregreso"/>
  <linkParam name="keyCode" value="CURSOR_UP"/>
  <bind role="start" component="mazul"/>
  <bind role="set" component="m0" interface="bounds">
  <bindParam name="var" value="0%,0%,200%,200%"/> <!-- left,top,width,height -->
  </bind>
</link>

<link xconnector="conector#onKeySelectionStartNStopN" id="regresomenu2">
<bind role="onSelection" component="mregreso"/>
<linkParam name="keyCode" value="CURSOR_UP"/>
  <bind role="start" component="mrojo"/>
  <bind role="start" component="mverde"/>
  <bind role="start" component="mamarillo"/>
  <bind role="start" component="mazul"/>
  <bind role="stop" component="mizquierda"/>
  <bind role="stop" component="mderecha"/>
  <bind role="stop" component="mregreso"/>
  <bind role="stop" component="m1.1"/>
  <bind role="stop" component="m1.2"/>
  <bind role="stop" component="m1.3"/>
  <bind role="stop" component="m1.4"/>
  <bind role="stop" component="m1.5"/>
  <bind role="stop" component="m1.6"/>
  <bind role="stop" component="m1.7"/>
  <bind role="stop" component="m1.8"/>
  <bind role="stop" component="m1.9"/>
  <bind role="stop" component="m1.10"/>
  <bind role="stop" component="m1.11"/>
  <bind role="stop" component="m1.12"/>
  <bind role="stop" component="m1.13"/>
  <bind role="stop" component="m1.14"/>
  <bind role="stop" component="m1.15"/>
  <bind role="stop" component="m1.16"/>
  <bind role="stop" component="m1.17"/>
  <bind role="stop" component="m1.18"/>
  <bind role="stop" component="m1.19"/>
  <bind role="stop" component="m1.20"/>
  <bind role="stop" component="m7.1"/>
  <bind role="stop" component="m7.2"/>
  <bind role="stop" component="m7.3"/>
  <bind role="stop" component="m7.4"/>
</link>

<link xconnector="conector#onKeySelectionStartNStopN" id="regresomenu3">
  <bind role="onSelection" component="mregreso"/>
  <linkParam name="keyCode" value="CURSOR_UP"/>
  <bind role="start" component="mrojo"/>
  <bind role="start" component="mverde"/>
  <bind role="start" component="mamarillo"/>
  <bind role="start" component="mazul"/>
  <bind role="stop" component="contexto1"/>
  <bind role="stop" component="contexto2"/>
  <bind role="stop" component="contexto3"/>
  <bind role="stop" component="contexto4"/>
</link>

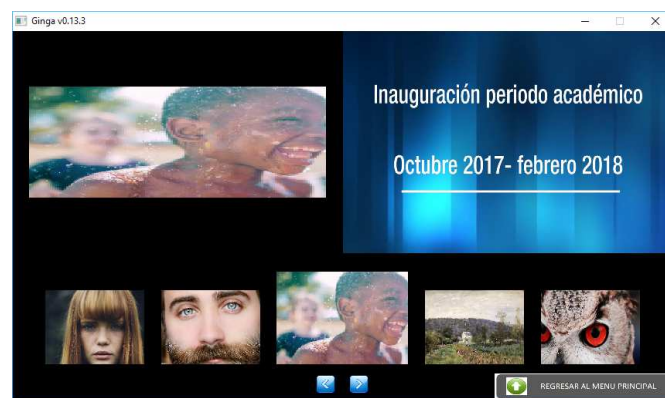
```

**Figura 44.** Programación para regresar al MENU principal

18. Finalmente se programó lo que tiene que ver con los contextos de cada uno de los botones, en este parte del programa se maneja todo lo que tiene que ver con el carrusel de imágenes, es decir, en qué posición debe estar cada imagen cuando se aplaste la flecha a la derecha o la flecha a la izquierda según corresponda. Se va jugando con los conectores mencionados anteriormente en donde se van dejando de visualizar las imágenes que no correspondan y se van iniciando las que correspondan de acuerdo a la visualización que se desea entregar al usuario.
19. En las Figuras 45 y 46 se muestra el resultado de la programación del menú principal, así como de uno de los botones respectivamente, para el correcto diseño de la aplicación interactiva que será utilizada posteriormente para el análisis de calidad de las imágenes pre-procesadas.



**Figura 45.** Pantalla del MENU principal de la aplicación interactiva



**Figura 46.** Pantalla visualizada al presionar el botón azul

Cabe mencionar que el mismo procedimiento se utilizó para la creación de las demás aplicaciones interactivas, 4 en total, con la única diferencia que lo que se iba cambiando son las imágenes obtenidas a través del algoritmo con su respectivo valor de N.

### 3.3. Generación del Transport Stream

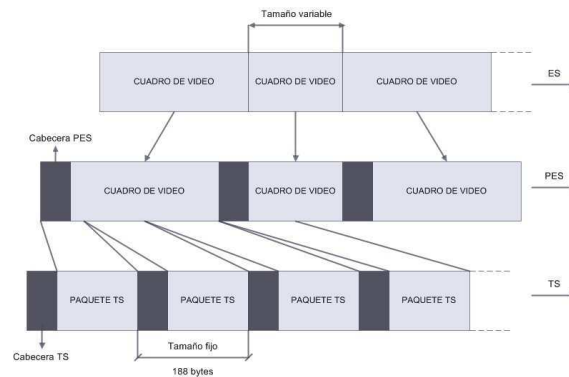
#### 3.3.1. Transport Stream (TS)

Es un protocolo de transmisión de audio, video y datos especificado en el estándar MPEG-2. El flujo de transporte ayuda a la multiplexación del audio y video combinando todos los datos en un solo flujo de bits sincronizados para su correspondiente transmisión en el medio que se requiera.

En el sistema ISDB-T los paquetes TS tienen una longitud fija de 188 bytes para de esta manera obtener compatibilidad con el modo de transferencia asíncrono que utiliza el estándar MPEG-2

#### 3.3.2. Estructura del Transport Stream

Los flujos de datos ES (Elementary Stream), PES (Packetized Elementary Stream) y TS están formados como se refleja en la Figura 47



**Figura 47.** Formación de flujos de datos

Fuente: (Villamarín, Illescas, Olmedo, & Cueva, 2013)



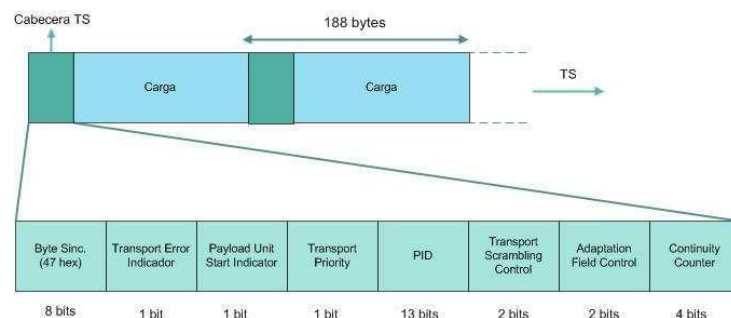
De los 188 bytes que forman el TS, 4 son de cabecera y 184 son de carga para transportar información como PES, servicios, datos y paquetes nulos para obtener una tasa de transmisión fija (Villamarín et al., 2013)

Por medio de la codificación el audio y el video obtienen la compresión de manera aislada para formar un ES con paquetes de tamaño variable dependiendo de la calidad que tengan tanto el audio como el video (Villamarín et al., 2013)

En la paquetización se consiguen los bits que vienen de la codificación del audio y video para formar como su nombre lo dice, paquetes PES, los cuales de la misma manera que los TS poseen una cabecera que contiene información y una carga. Su tamaño es variable ya que su compresión se lo relaciona con el tiempo de cuadro y su tasa de compresión aleatoria.(Villamarín et al., 2013)

Los paquetes PES se dividen en 188 bytes formando de esta manera un flujo de datos TS lo cual ayuda a que los errores sean detectados de manera más precisa y a tener un flujo de datos único para varios programas al momento de multiplexarlos. Por ende, el TS transporta tanto video como audio en conjunto con la información adicional requerida.

La cabecera que tiene un flujo de datos TS se muestra en la Figura 48 y a continuación se explica de manera general cada una de las partes



**Figura 48.** Cabecera TS

Fuente: (Villamarín et al., 2013)

- Byte de sincronismo: utilizado como referencia del inicio del paquete TS
- Transport Error Indicator: existencia de error en el paquete. Cuando está en 1 el decodificador descarta el paquete
- Payload Unit Start Indicator: cuando está en 1 el paquete transporta tablas o un PES
- Transport Priority: da a conocer el nivel de prioridad de transporte que tienen los diferentes paquetes que están en el mismo PES. Mayor prioridad es indicada mediante un 1
- PID: identificador de cada paquete del TS, depende del tipo de información que lleva la carga
- Transport Scrambling Control: indica el modo de codificación de la carga útil
- Adaptation Field Control: señala la presencia o ausencia en el TS de campos de adaptación y carga útil como se indica en la Tabla 8

**Tabla 8***Valores del campo de control de adaptación*

<b>Valor</b>	<b>Detalle</b>
00	Reservado
01	Solo carga útil sin campo de adaptación
10	Solo campo de adaptación sin carga útil
11	Campo de adaptación y carga útil

- Continuity Counter: contador progresivo de paquetes con el mismo PID y que llevan carga útil

La multiplexación tiene como objetivo juntar varios tipos de información en uno o diversos flujos independientes. Es importante tomar en cuenta que en este proceso se debe ordenar el envío de paquetes de cada uno de los TS, así como también el hecho de que las tablas de cada flujo creado debe tener todos los programas que resultan de la multiplexación (Villamarín et al., 2013)

La información adicional que transportan algunos paquetes de cómo está formado el flujo de datos, cuantos programas se van a transmitir y a que programa pertenece cada audio y cada video hace que la utilización de PID (identificador)

sea indispensable para la correcta visualización en el receptor. (Villamarín et al., 2013)

Las informaciones sobre el flujo se las denomina tablas de información de cada uno de los sistemas, las cuales son transportadas en conjunto con la carga que llevan los paquetes. Cada una de las tablas se identifican con un PID diferente como en la Tabla 9 se muestra

**Tabla 9**  
*PIDs usados en la transmisión de Tablas del Sistema*

<b>PID</b>	<b>Detalle</b>
0x0000	Paquetes de transporte de la tabla PAT (Program Association Table)
0x0001	Paquetes de transporte de la tabla CAT (Conditional Access Table)
0x0003-0x000F	PIDs reservados
0x0010	Paquetes de transporte de la tabla NIT (Network Information Table)
0x0011	SDT (Service Description Table) y BAT (Bouquet Information Table)
0x0012, 0x0026 y 0x0027	EIT (Event Information Table)
0x0013	RST (Running Status Table)
0x0014	TDT (Time and Date Table) y TOT (Time Offset Table)
0x0011 – 0x1FFE	Tablas que no incluyen PAT, CAT, NIT y paquetes nulos
0x0022	PCAT (Partial Content Announcement Table)
0x0024	BIT (Broadcaster Information Table)
0x0025	NBIT (Network Board Information Table) y LDT (Linked Description Table)
0x1FFF	Paquetes nulos

Fuente: (Villamarín et al., 2013)

Las tablas PSI/SI permiten que el receptor sea capaz de identificar y decodificar todo lo que contiene el TS, para esto, se transmiten en una frecuencia alta de manera que el receptor encendido en cualquier instante realice lo mencionado anteriormente. Cada una de ellas poseen secciones de tamaño variable y se las transporta dentro del TS (Villamarín et al., 2013)

En las Tablas 10 y 11 se da una breve explicación de la funcional de cada una de las tablas, tanto PSI como SI respectivamente

**Tabla 10**

*Tablas PSI principales para el sistema ISDB-Tb*

Nombre	Detalle
PAT	Tabla de asociación de programas. A cada programa le da un PID de la PMT
PMT	Tabla de mapeamiento de programas. Indica los componentes de un programa o servicio
CAT	Tabla de acceso condicional. Da información sobre el acceso condicional a diversos programas

Fuente: (Villamarín et al., 2013)

**Tabla 11**

*Tablas SI para el sistema ISDB-Tb*

Nombre	Detalle
AIT	Tabla de información de aplicación. Transmite información y control de aplicaciones que se transmiten en el flujo
BIT	Tabla de información de radiodifusor. Se encarga de designar las radiodifusoras y los parámetros SI correspondientes
EIT	Tablas de información de eventos. Transporta información sobre eventos de cada servicio y su orden cronológico
NIT	Tabla de información de red. Lleva información de cómo se formó el TS y su medio de transmisión
RST	Tabla de estado de evento. Actualiza el estado de eventos
SDT	Tabla de descripción de servicios. Reconoce los servicios en el TS
TDT	Tabla de fecha y horario. Indica fecha y hora del sistema
TOT	Tabla de cambio de fecha y hora. Indica al receptor la hora, fecha y huso horario

Fuente: (Villamarín et al., 2013)

### 3.3.3. Generación del TS

Para la generación del Transport Stream de cada una de las aplicaciones interactivas creadas se siguió el procedimiento mencionado en (Villamarín et al.,

2013), en donde se da a conocer que de una manera óptima el software OpenCaster2.4 ayuda a realizar la tarea antes mencionada.

Se utilizó este software debido a que es libre y por ende no representa un costo al proyecto que se realizó. Fue desarrollado por AVALPA Digital Engineering SRL (Avalpa Digital Engineering Srl, 2018) y permite la creación de TS en formato MPEG2 para el sistema de televisión DVB. Posteriormente varios investigadores de LIFIA en la Universidad de la Plata en Argentina hicieron cambios para que OpenCaster2.4 pueda ser utilizado en el sistema de televisión ISDB-T (Villamarín et al., 2013)

Los archivos que se necesitan para poder instalar este software se pueden bajar del sitio web de LIFIA (<ftp://tvd.lifia.info.unlp.edu.ar/OpenCaster2.4/>), los cuales incluye la versión original de OpenCaster2.4 y el archivo para aplicar los cambios de LIFIA. Ambos se deben copiar dentro del mismo directorio, cabe recalcar que la instalación se la realiza en Linux a través del terminal. El siguiente paso a seguir es la descompresión del archivo original para posteriormente aplicar las modificaciones a través del patch y finalmente instalar todo como usuario root para su correcto funcionamiento (Villamarín et al., 2013)

De forma general y para una mejor comprensión cabe mencionar que para pasar de información analógica a digital se usan los códec para de esta manera analizar los fotogramas y ayudar a la compresión de datos (Villamarín et al., 2013)

Dentro de los formatos de audio tenemos dos tipos principales, los que tiene pérdida de información y los que no. Dentro de los formatos con pérdida están: AAC, MP2, MP3, WMA; y dentro de los formatos sin pérdida están: AIFF, FLAC, WAV y MIDI. (Villamarín et al., 2013)

En los formatos de video existen tres de los más principales que son: AVI (permite audio y video, multiplataforma), MOV (creado por Apple, multiplataforma y permite visualizar películas en formato 3D) y MPEG (existe una pequeña pérdida de calidad en la compresión)

La resolución de video tiene que ver con el número de píxeles que se muestran en la pantalla. Se lo representa como un producto en donde se visualiza el ancho y alto en píxeles para de esta manera tener una relación de aspecto. A continuación, se muestra las relaciones de aspecto o resolución de acuerdo a cada tipo de formato (Villamarín et al., 2013):

- LDTV (Televisión en baja definición) con una resolución de 320x240 y una relación de aspecto de 4:3
- SDTV (Televisión en definición estándar) con una resolución de 640x480 y una relación de aspecto de 4:3
- EDTV (Televisión en definición mejorada) con una resolución de 1280x720 y una relación de aspecto de 16:9
- HDTV (Televisión en alta definición) con una resolución de 1920x1080 y una relación de aspecto de 16:9

La codificación se la realiza por separado tanto para el video como para el audio más sin embargo se parte de una fuente común de audio y video (Villamarín et al., 2013)

A continuación, se explicará paso a paso como se realizó la creación del TS y que parámetros se deben tomar en cuenta:

1. Partimos de un archivo .AVI, pero como no todo archivo se encuentra en este formato, se toma en cuenta una herramienta que nos proporciona Linux para transformar al formato requerido. Esta herramienta se llama mencoder que

codifica diversos tipos de formatos a través de líneas de comandos en el terminal mostrados a continuación

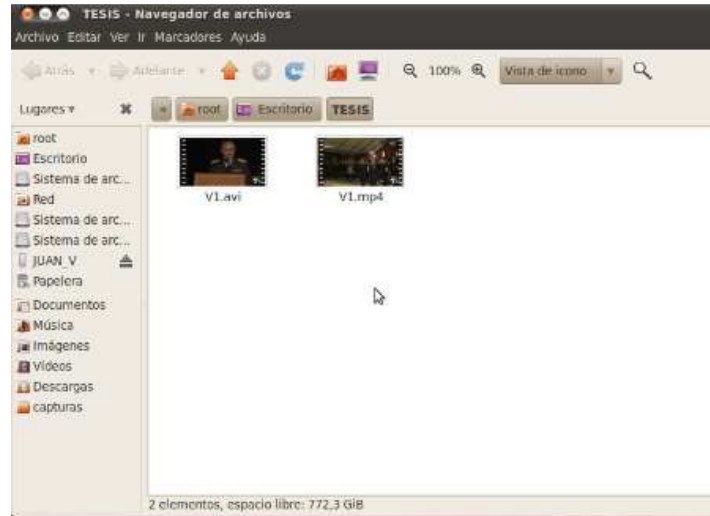
```
mencoder V1.mp4 -ofps 30 -ovc xvid -oac mp3lame -lameopts  
abr:br=192 -srate 48000 -xvidencopts fixed_quant=4 -o V1.avi
```

Previamente se debe tomar en cuenta el ubicarnos en el directorio adecuado dentro del cual se encuentra el archivo .mp4 para de esta manera proceder a convertir a formato .avi. En la Figura 49 y 50 de manera respectiva, se ve la finalización del proceso de transformación por medio del Terminal y la creación del nuevo archivo en la carpeta correspondiente.

```
mencoder SVN-r1.0-rc3+svn28098426-4.4.3 (C) 2000-2009 MPlayer Team
success: format: 0 data: 0x0 - 0x1282dc
libavformat file format detected.
[Lavf] Video stream found, -vid 0
[Lavf] Audio stream found, -aid 1
VCodec: [xvid] 1280x720 -4bpp_29.970 fps 0.0 khps (0.0 kbyte/s)
[V] filefmt:44 fourcc:0x31637661 size:1280x720 fps:29.970 ftime:=0.0334
=====
Opening audio decoder: [faad] AAC (MPEG2/4 Advanced Audio coding)
FAAD: compressor input bitrate missing, assuming 128kbits/s!
AUDIO: 44100 Hz, 2 ch, s16le, 128.0 kbit/9.07% (ratio: 16000->176400)
Selected audio codec: [faad] aka: faad (FAAD AAC (MPEG-2/MPEG-4 Audio))
=====
xvid: using library version 1.2.2 (build xvid-1.2.2)
Opening video filter: [expand osd=1]
Expand: 1 x 1, 1 : 1, osd: 1, aspect: 0.000000, round: 1
=====
Opening video decoder: [ffmpeg] FFmpeg's libavcodec codec family
Selected video codec: [fh264] vfm: ffmpeg (FFmpeg H.264)
=====
MP3 audio selected.
VDec: vo config request - 1280 x 720 (preferred colorspace: Planar YV12)
VDec: using Planar YV12 as output csp (no 0)
Movie-Aspect is 1.78:1 - prescaling to correct movie aspect.
videocodec: Xvid [1280x720 fourcc=44495658 [XVID]]
xvid: par=0/0 (vgall), displayed=1280x720, sampled=1280x720
xvid: Fixed Quant Rate Control -- quantizer=4/1-4.00
Writing header...
ODM: vprp aspect is 16:9.
Setting audio delay to 0.024s.
Writing header...
ODM: vprp aspect is 16:9.
Setting audio delay to 0.024s.
Pos: 0.6s 21f (0%) 0.00fps Trem: 0min 6mb A-V:0.067 [0:72]
Skipping frame!
Pos: 1.6s 32f (1%) 0.00fps Trem: 1min 15mb A-V:0.068 [0:102]
Skipping frame!
Pos: 66.5s 2000f (66%) 24.09fps Trem: 0min 31mb A-V:0.025 [2506:181]
1 duplicate frame(s)!
Pos: 99.9s 3000f (99%) 24.11fps Trem: 0min 32mb A-V:0.029 [2493:182]
1 duplicate frame(s)!
Pos: 100.0s 3021f (100%) 24.06fps Trem: 0min 32mb A-V:0.015 [2489:182]
Flushing video frames.
Writing index...
Writing header...
ODM: vprp aspect is 16:9.
Setting audio delay to 0.024s.

Video stream: 2407.074 kbit/s (310984 B/s) size: 31316112 bytes 100.700 secs 3021 frames
Audio stream: 192.468 kbit/s (22808 B/s) size: 2298900 bytes 100.752 secs
root@espe-tv-desktop:~/Escritorio/TEST5#
```

**Figura 49.** Finalización del proceso de transformación



**Figura 50.** Creación del archivo en formato AVI

Los parámetros que se utilizaron en el comando se visualizan en la Tabla 12

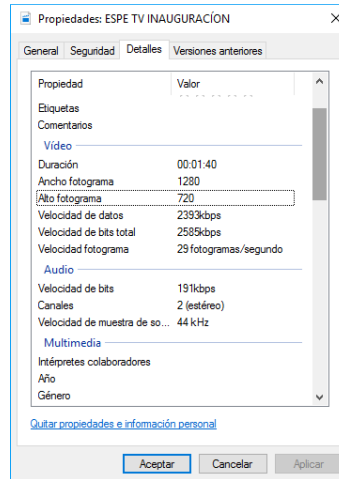
**Tabla 12**

*Parámetros utilizados para mencoder*

Parámetro	Detalle
V1.mp4	Video a convertir
-ofps 30	Número de cuadros por segundo
V1.avi	Video convertido

Para poder transformar el video original en .AVI a través de mencoder se debe conocer previamente la información con la que se codificó el video que se posee para de esta manera generar de manera correcta el TS posteriormente y para esto se da click derecho sobre el archivo y se escoge Propiedades en donde se visualiza una pantalla como en la Figura 51





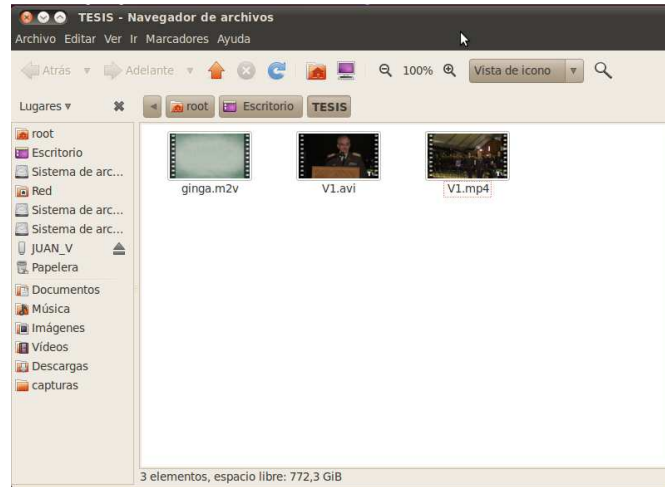
**Figura 51.** Propiedades del video en formato MP4

2. Como la codificación de audio y video se realiza por separado, se empezó por la codificación de video a través del comando mostrado a continuación. En la Figura 52 y 53 de manera respectiva, se visualiza la finalización del proceso y la creación del archivo respectivo en la carpeta

```
ffmpeg -i V1.avi -an -vcodec mpeg2video -f mpeg2video -s 1280x720 -r 30 -aspect 16:9 -deinterlace -b 4000k -minrate 4000k -maxrate 4000k -bf 2 -bufsize 1835008 ginga.m2v
```

```
ffmpeg version 3.4.4 Copyright (c) 2000-2019 FFmpeg developers
  configuration: extra-version=3.4.4 libavutil 55.17.100 libavcodec 57.243.138 libavformat 57.25.102 libavfilter 7.2.0 libavresample 4.0
  libavutil 55.17.100
  libavcodec 57.243.138
  libavformat 57.25.102
  libavfilter 7.2.0
  libavresample 4.0
  Input #0: avformat, format=avi, codec=unrecognized, 30 fps, 30 tbr, 30 tbc, 30 tbc
  Stream #0: Video: mpeg2video, yuv420p, 1280x720 [PAR: 16:9], 30 tbr, 30 tbn, 30 tbc
  Output #0: mpeg2video, format=mpeg2video, 30 fps, 30 tbr, 30 tbn, 30 tbc
  Stream #0: Video: mpeg2video, yuv420p, 1280x720 [PAR: 16:9], 30 tbr, 30 tbn, 30 tbc
  Stream mapping:
    Stream #0:0 -> #0:0
  Press [q] to stop encoding
  [m2v @ 0:0:0] Frame non-key: 0, time=180.67, bitrate=4001.4kbits/s
  video:491718 audio:0kB global headers:0kB muxing overhead 0.00000%
  ffmpeg: AVOutputFormatContext::write_header
```

**Figura 52.** Finalización del proceso de codificación de video



**Figura 53.** Creación del archivo ginga.m2v

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 13

**Tabla 13**  
*Parámetros para la codificación de video*

Parámetro	Detalle
-i	Archivo fuente que contiene audio y video en formato .AVI
-an	Ignora la señal de audio
-vcodec	Códec de video, se necesita el mpeg2video para compatibilidad con el TS
-f	Formato de salida, se necesita en este caso mpeg2video
-s	Resolución del video en pixeles
-r	Frames por segundo
-aspect	Relación de aspecto
-deinterlace	Desentrelazado de imágenes
-b	Tasa de bits de la señal codificada en Kbps, este mismo valor va en maxrate y minrate para mantener constancia
-bf	Número de cuadros B para cada GOP
-bufsize	Tamaño del buffer en bits relacionada con la expresión $vbv\_buffersize * 1024 * 16 = 112 * 1024 * 16 = 1835008$
ginga.m2v	Nombre del archivo que se genera

3. El comando utilizado para la codificación de audio se visualiza a continuación. En la Figura 54 y 55 de manera respectiva, se puede visualizar la finalización del proceso y la creación del archivo correspondiente

```
ffmpeg -i V1.avi -vn -ac 2 -acodec mp2 -f mp2 -ab 128000 -ar 48000
ginga.mp2
```

```
ffmpeg version 2.8.2 Copyright (c) 2000-2016 FFmpeg developers
  built on Mar 17 2016 00:37:26; gcc: 4.4.3
  input #0, av1, from 'V1.avi'
  duration: 00:01:40.79, start: 0.00000, bitrate: 2043 kb/s
  Stream #0:0: video: av1, yuv420p, 1280x720 (PAR:1:1 DAR:4:3), 30 fps, 30 tbn, 30 tbc
  Stream #0:1: audio: mp2, 48000 Hz, stereo, s16, 50 kb/s
  Output #0, mp2, to 'ginga.mp2'
  Stream #0:0: video: mp2, 48000 Hz, stereo, s16, 128 kb/s
  Stream #0:1: audio: mp2, 48000 Hz, stereo, s16, 128 kb/s
  Stream #0:2: data: none
  Stream #0:3: data: none
  Press [q] to stop encoding
  size= 137681 size=100.75 bitrate= 128.00kbits/s
  video:100.75kB audio:128.00kB encoder: none writing overhead 0.000000
  ffmpeg -i V1.avi -vn -ac 2 -acodec mp2 -f mp2 -ab 128000 -ar 48000
ginga.mp2
```

Figura 54. Finalización del proceso de codificación de audio

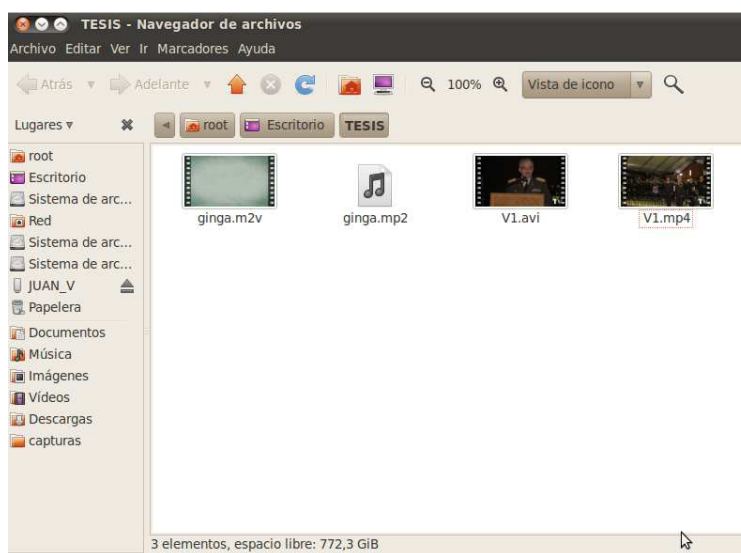


Figura 55. Creación del archivo ginga.mp2

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 14

**Tabla 14**  
Parámetros para la codificación de audio

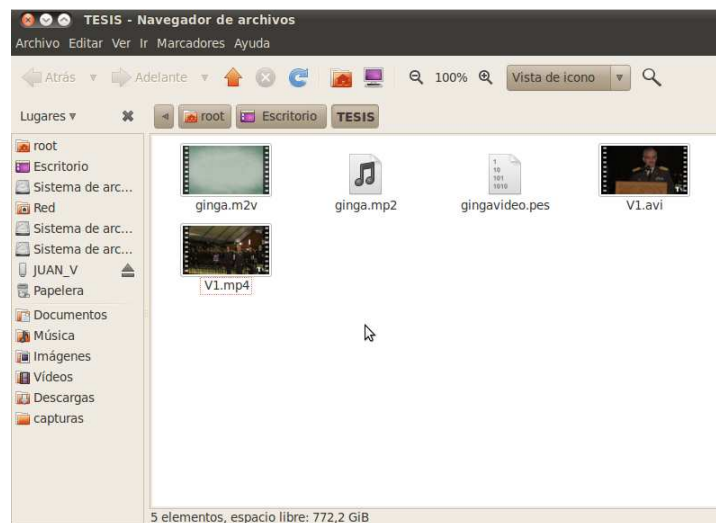
Parámetro	Detalle
-i	Archivo fuente que contiene audio y video en formato .AVI
-vn	Ignora la señal de video
-ac	Número de canales, por default es 1
-acodec	Códec de audio, se necesita el mp2 para compatibilidad con el TS
-f	Formato de salida, se necesita en este caso mp2
-ab	Tasa de bits en bps
-ar	Frecuencia de muestreo de la señal de audio
ginga.mp2	Nombre del archivo que se genera

4. Se realizó la paquetización de video partiendo del archivo ES que se obtuvo después de la codificación. Se indica el comando utilizado para este propósito. En la Figura 56 y 57 de manera respectiva, se visualiza la finalización del proceso y la creación del archivo correspondiente.

*esvideo2pes ginga.m2v > gingavideo.pes*

```
root@espe-tv-desktop:~/Escritorio/TESIS# esvideo2pes ginga.m2v > gingavideo.pes
10875600
root@espe-tv-desktop:~/Escritorio/TESIS#
```

**Figura 56.** Finalización del proceso de paquetización de video



**Figura 57.** Creación del archivo gingavideo.pes

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 15

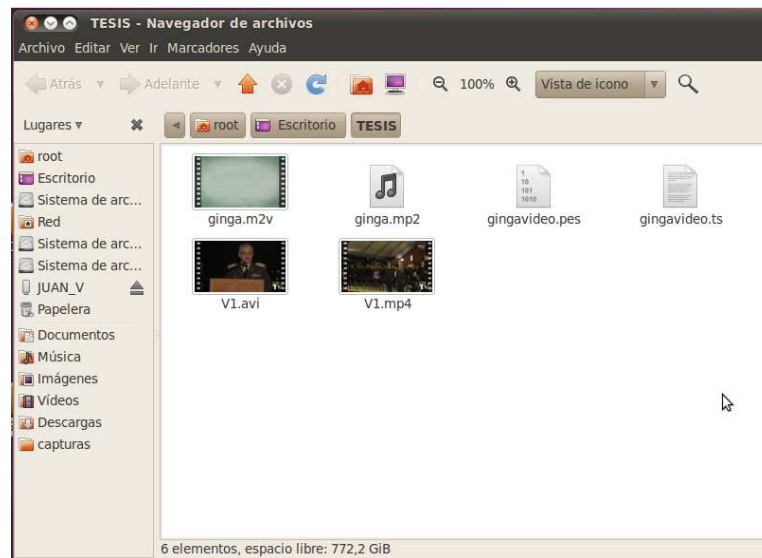
**Tabla 15**

*Parámetros utilizados en la paquetización de video*

Parámetro	Detalle
ginga.m2v	Video ES que se va a convertir
gingavideo.pes	Video convertido a PES

5. El siguiente comando que nos brinda OpenCaster ayuda a pasar el archivo PES a TS en este caso, de video. En la Figura 58 se visualiza la creación del archivo correspondiente

```
pesvideo2ts 2065 30 112 4600000 0 gingavideo.pes > gingavideo.ts
```



**Figura 58.** Creación del archivo `gingavideo.ts`

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 16

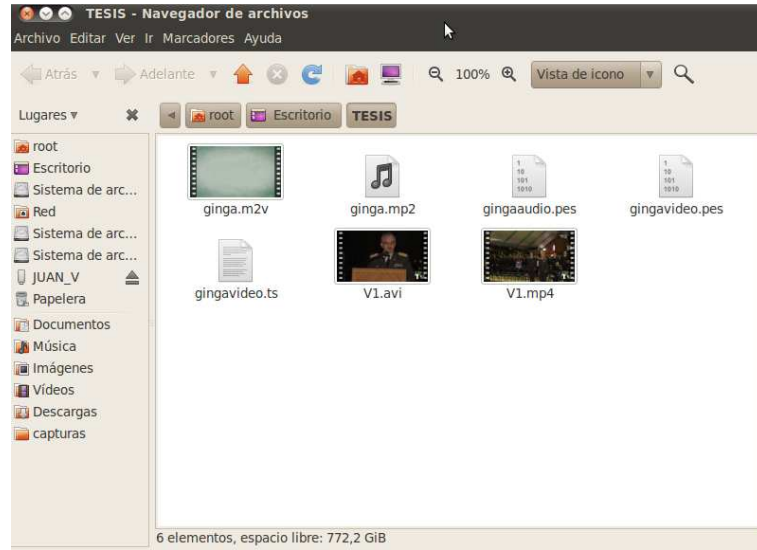
**Tabla 16**

*Parámetros utilizados en la paquetización de video*

Valor del parámetro	Detalle
2065	PID del video
30	Frames por segundo
112	Valor del Video Buffer Verifier definido en MPEG-2
4600000	Ancho de banda del TS. Recomendable un 15% más que la tasa de bits de la señal de video
0	El video no va a estar en un lazo cerrado
	Video PES que se va a convertir
<code>gingavideo.ts</code>	Video de salida en TS

- Para la paquetización de audio se utilizó el comando mostrado a continuación, partiendo de un archivo ES (Elementary Stream) que se obtuvo como resultado de la codificación de audio. En la Figura 59 se visualiza la creación del archivo correspondiente.

```
esaudio2pes ginga.mp2 1152 48000 384 3600 > gingaaudio.pes
```



**Figura 59.** Creación del archivo *gingaaudio.pes*

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 17

**Tabla 17**

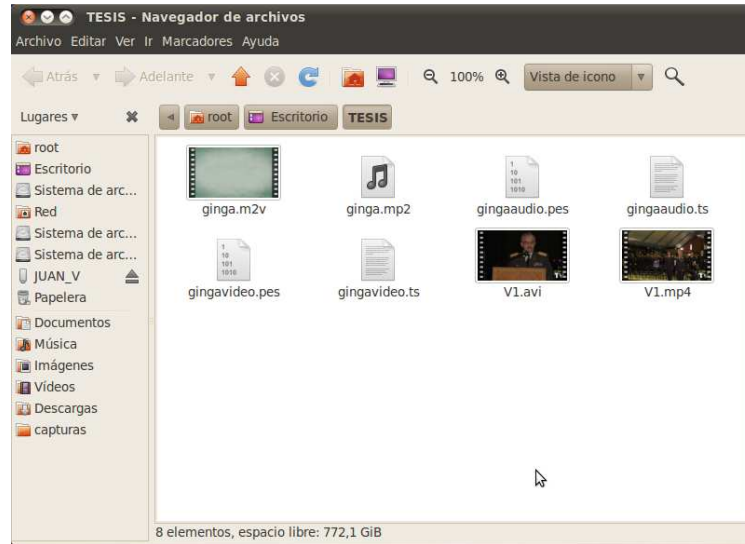
*Parámetros para la paquetización de audio*

Valor del parámetro	Detalle
<i>ginga.mp2</i>	Audio ES a ser convertido
<i>gingaaudio.pes</i>	Audio convertido a PES
1152	Muestras por frame
48000	Frecuencia de muestreo
384	Tamaño del frame de audio
3600	Desplazamiento del Presentation Time Stamp para sincronizar con el video

7. Para pasar de un archivo PES de audio a TS se utilizó el comando siguiente.

En la Figura 60 se visualiza la creación del archivo correspondiente

```
pesaudio2ts 2075 1152 48000 384 0 gingaaudio.pes > gingaaudio.ts
```



**Figura 60.** Creación del archivo gingaaudio.ts

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 18

**Tabla 18**

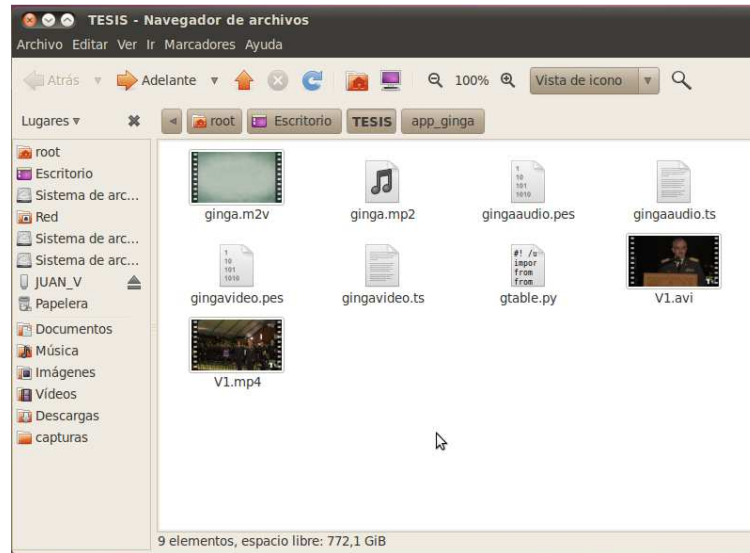
*Parámetros para la paquetización de audio*

Valor del parámetro	Detalle
2075	PID para el audio
1152	Muestras por frame
48000	Frecuencia de muestreo
384	Tamaño de frame de audio
0	El audio no va a estar en un lazo cerrado
gingaaudio.pes	Audio PES a convertirse
gingaaudio.ts	Audio de salida en formato TS

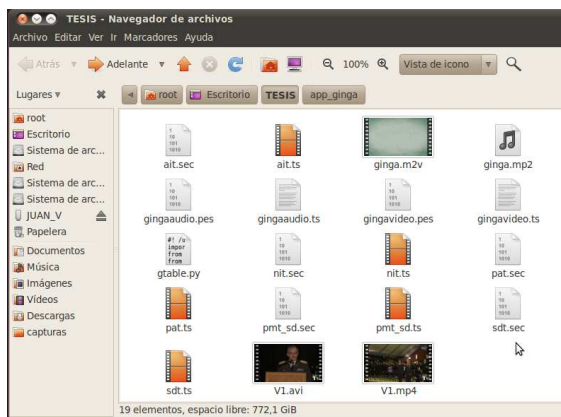
- Para generar las tablas PSI/SI necesarias, se lo realizó mediante un script en Python para que al momento de compilarlo cada una de las secciones de la tabla se conviertan en paquetes, las mismas que son indispensables al momento de la multiplexación para que el TS sea el correcto a transmitirse (Villamarín et al., 2013). El Script completo debe ser guardado en la misma carpeta en donde se han generado el resto de archivos como se visualiza en la Figura 61. A continuación se muestra el comando para compilarlo y ejecutarlo, generando de esta manera los archivos de cada tabla en formato TS, Figura 62

*chmod u+x gtable.py*

*./gtable.py*



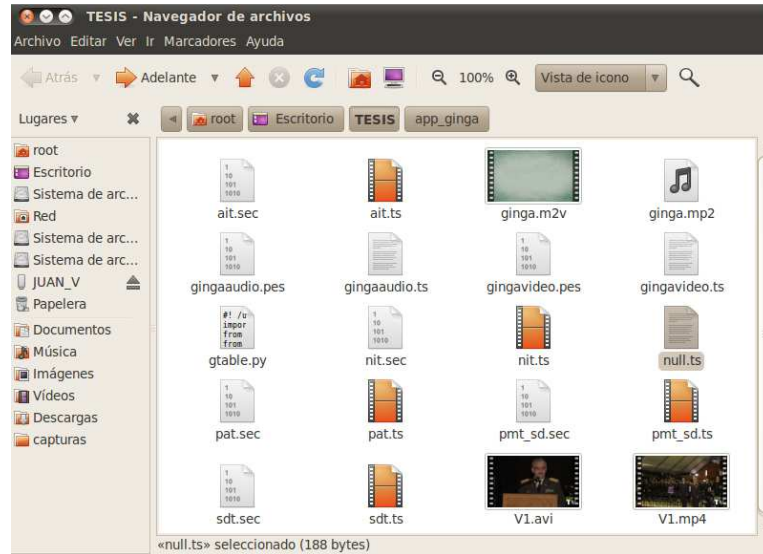
**Figura 61.** Creación del script gtable.py para las tablas PSI/SI



**Figura 62.** Creación de los archivos de cada tabla en formato TS

9. Se generó un archivo vacío para de esta manera guardar los paquetes nulos con el objetivo de completar el ancho de banda de los 6MHz como se visualiza en la Figura 63

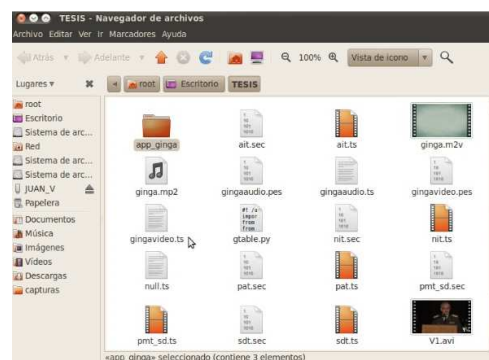




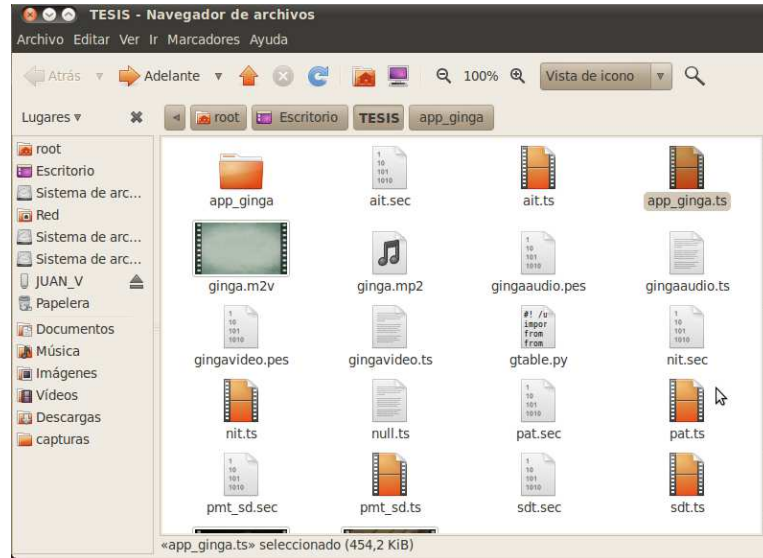
**Figura 63.** Creación del archivo para paquetes nulos

10. En cuanto a la aplicación en Ginga se debe tener una carpeta llamada `app_ginga` debido a que de esta manera se lo definió en la tabla AIT, la cual contiene todo lo relacionado a la aplicación diseñada, Figura 64; y con ayuda de la herramienta de mostrada a continuación se genera el carrusel de objetos en la carpeta correspondiente, Figura 65. Cabe mencionar que en esta carpeta no debe estar contenido el video con el que se trabajó al principio al momento de la codificación de audio y video, además el programa principal diseñado debe contener un cambio correspondiente al momento de declarar el media de video como se muestra en la Figura 66.

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



**Figura 64.** Carpeta que contiene todo lo relacionado a la aplicación en Ginga



**Figura 65.** Creación del archivo TS de la aplicación interactiva

```
<media id="m0" src="sbtvd-ts://0" descriptor="d0">
  <area id="inf" begin="1s"/>
  <property name="bounds"/>
</media>
```

**Figura 66.** Cambio en el programa principal para la correcta generación del TS de la aplicación en Ginga

Los parámetros que se utilizaron en este comando se visualizan en la Tabla 19

**Tabla 19**

*Parámetros utilizados para la generación del carrusel de objetos*

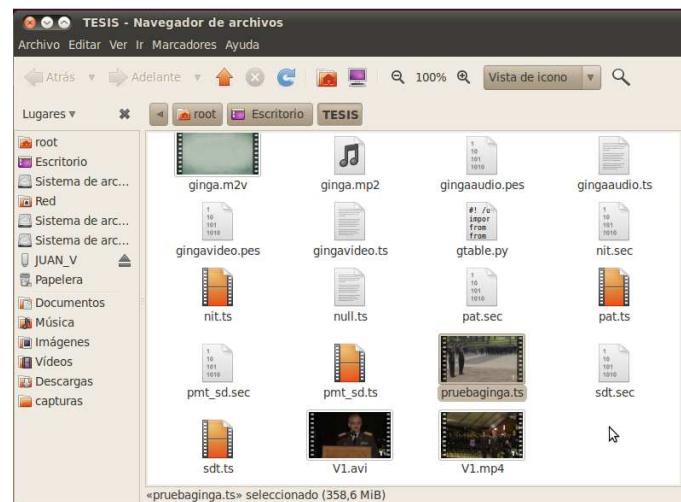
Valor del parámetro	Detalle
app_ginga	Nombre del archivo generado y directorio de la aplicación
0x0C	Association_tag del carrusel que se va a generar
1	Número de versión de los módulos generados
2004	PID del carrusel
2	Identificador del carrusel

11. Para la multiplexación se utiliza el comando mostrado a continuación, obteniendo como resultado el archivo final en formato TS, el cual es un solo flujo de datos que contiene video, audio y la aplicación interactiva. En la Figura 67 y 78 de manera respectiva, se puede visualizar la finalización del proceso y la creación del archivo correspondiente.

```
tscbrmuxer 2000000 b:15040 pat.ts b:15040 pmt_sd.ts b:3008 sdt.ts
b:3008 nit.ts b:3008 ait.ts b:1000000 app_ginga.ts b:4600000
gingavideo.ts b:188000 gingaaudio.ts b:24131190 null.ts >
pruebaginga.ts
```

```
.ts
output bitrate: 29958294.000000
new pid entry 0
new pid entry 1031
new pid entry 17
new pid entry 16
new pid entry 2001
new pid entry 2004
new pid entry 2065
new pid entry 8191
new pid entry 2075
root@espe-tv-desktop:~/Escritorio/TESIS#
```

**Figura 67.** Finalización del proceso de multiplexación



**Figura 68.** Creación del archivo pruebaginga.ts

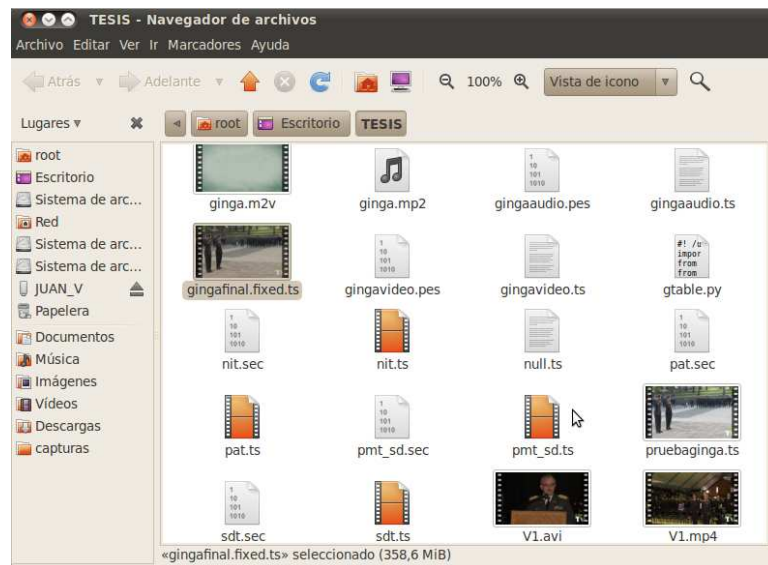
Los parámetros que se utilizaron en este comando se visualizan en la Tabla 20

**Tabla 20**  
*Parámetros para la multiplexación de archivos*

Valor del parámetro	Detalle
2000000	Cantidad de paquetes que se van a multiplexar. Se transmite en ISDB-T alrededor de 20000 paquetes por segundo
b:15040	PAT y PMT se envía a 10 veces por segundo. Se envía 10 paquetes por segundo y cada paquete de 188 bytes
b:3008	NIT y SDT se envían 2 veces por segundo
b:4600000	Ancho de banda de video
b:188000	Ancho de banda de audio
b:24131190	Ancho de banda para paquetes nulos
pruebaginga.ts	Archivo de salida en formato TS

12. Finalmente se realiza la sincronización debido a que al multiplexar se cambia el lugar de los paquetes de video y al momento de la recepción será impreciso. Se visualiza a continuación el uso de `tsstamp` para este propósito. En la Figura 69 se visualiza la creación del archivo final en formato TS listo para su transmisión.

```
tsstamp pruebaginga.ts 29958294 > gingafinal.fixed.ts
```



**Figura 69.** Creación del archivo `gingafinal.fixed.ts`

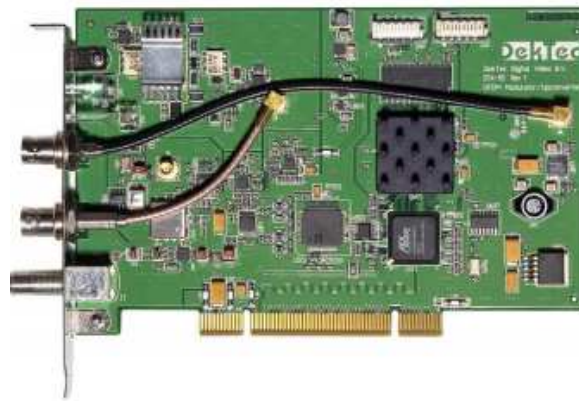
### 3.4. Transmisión en Tiempo Real

Para la modulación del TS se utilizó la tarjeta moduladora DTA-115, la cual se encuentra instalada en una computadora del Laboratorio de Televisión Digital de la Universidad de las Fuerzas Armadas – ESPE. Dicha computadora cuenta con dos sistemas operativos: Windows que tiene compatibilidad para un uso adecuado de la tarjeta y Linux para la generación del TS mencionado en la sección anterior.

Con la ayuda de Stream Xpress TS Player se puede realizar la transmisión de manera óptima por medio de la computadora que en este caso llega a ser el

servidor de contenidos TS y debe tener mínimo un Core 2@2GHz para ISDB-T, 512 MB en RAM y un puerto PCI (Villamarín et al., 2013)

La tarjeta mencionada anteriormente, Figura 70, es multi-estándar y cuenta con un convertidor de frecuencia VHF / UHF, soporta la mayoría de estándares de modulación que están basados en QAM, OFDM y VSB admitiendo todas las constelaciones para cada uno de los estándares. Además, cubre toda la banda de VHF y UHF. Se la puede utilizar para generar pruebas multi-estándar, para reemplazar algún tipo de modulador y para obtener una salida RF que se necesite en algún tipo de aplicación (DekTec, 2011)



**Figura 70.** Tarjeta moduladora DTA-115

En la Figura 71 se puede apreciar la manera en que la tarjeta se encuentra instalada en la computadora (servidor) del Laboratorio.



**Figura 71.** Instalación de la tarjeta

Todo lo mencionado anteriormente forma parte en lo que respecta al transmisor. La parte de recepción está formada por tres tipos de televisores (uno a la vez) y un Set Top Box. La Tabla 21 indica algunas de las características técnicas de los elementos utilizados para el proyecto

**Tabla 21**  
*Elementos de recepción*



- Modelo: KV-27FS16
- Alimentación de energía: AC120 60Hz
- Peso: 47 kg
- Consumo de energía: 170W
- Salida de sonido: 5W x 2

**Televisor de Rayos Catódicos Sony 27"**



- Modelo: KDL-40BX425
- Resolución: 1920x1080
- Tamaño: 101.6 cm
- Frecuencia de imagen: 60/50Hz
- Sintonizador ISDB-T
- Alimentación de energía: AC110-240V ~ 50/60Hz
- Consumo de energía: 177W
- Peso: 13.3 kg

**Televisor LCD Sony Bravia 40"**



- Resolución: 1366x768
- Modelo: UN32H4303
- Tamaño: 80 cm
- Peso: 5.4 kg
- Salida de sonido: 5Wx2 (RMS)
- Alimentación de energía: AC100-240V ~ 50/60Hz
- Consumo de energía: 69W

**Televisor LED Samsung 32"**



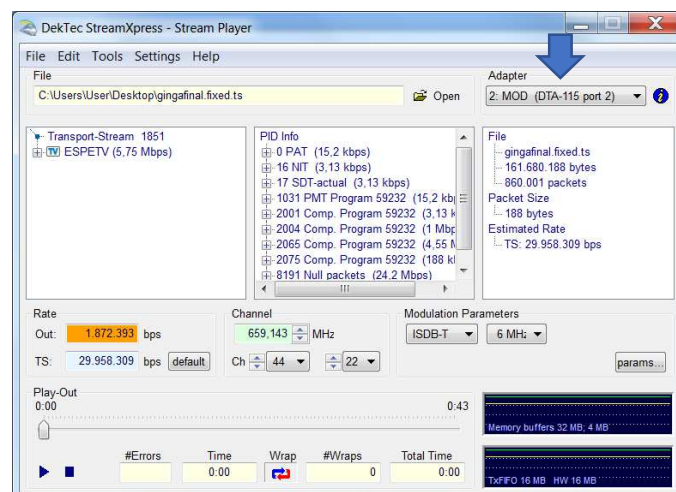
**Set Top Box EITV**

STB híbrido tanto para ISDB-T e IPTV, está diseñado para soportar aplicaciones Ginga-NCL y Ginga-J. Se puede probar este tipo de aplicaciones por red, aire o embebidas.

- Frecuencia de entrada UHF: 470MHz a 806MHz
- Frecuencia de entrada VHF: 174MHz a 216MHz
- Ancho de banda: 5.6MHz

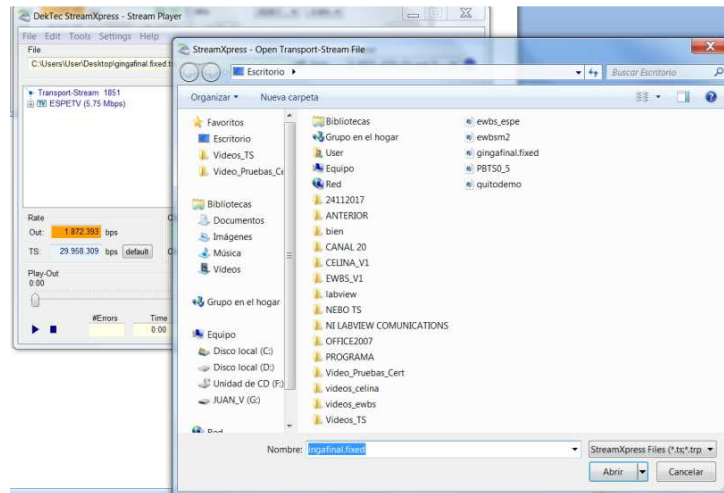
El software Stream Xpress TS Player es un paquete de Windows de fácil uso, el cual fue diseñado para reproducir en tiempo real transmisiones compatibles con MPEG-2. Se carga por el usuario final en cualquier tipo de computadora trabajando en conjunto con los dispositivos de salida DekTec como es la tarjeta moduladora DTA-115. Soporta varias interfaces de transmisión como con ASI. ToSIP, salidas moduladas RF y SDI para de esta manera dar a conocer información necesaria acerca del flujo que se va a transportar (DekTec, 2017)

Para su instalación se puede encontrar los archivos necesarios en la página web de DekTec, la cual es <http://www.dektec.com/Downloads/Applications.asp>, luego de su respectiva descarga y descompresión se obtiene el archivo .exe en el cual se da doble click para que inicie la instalación correspondiente. En nuestro caso la instalación de este software ya se encontraba realizada en la computadora del Laboratorio de Televisión Digital y en la Figura 44 se visualiza la interface que posee al inicializarlo. Cabe mencionar que es importante que StreamXpress reconozca de manera adecuada la tarjeta modulada DTA-115 para que la transmisión del TS sea haga de manera correcta, Figura 72

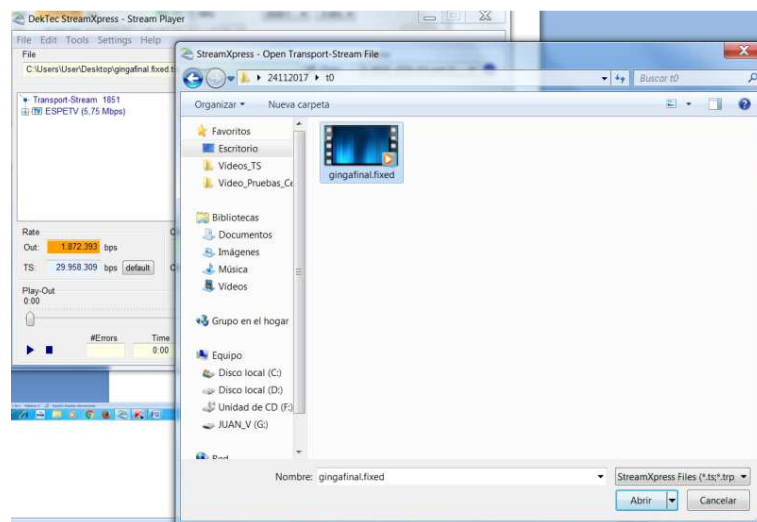


**Figura 72.** Interfaz gráfica StreamXpress y reconocimiento de tarjeta moduladora DTA-115

Para poder transmitir el TS a través de StreamXpress se da click en la pestaña Open y de esta manera se busca la dirección en donde se encuentra almacenado el archivo en formato TS como se indica en la Figura 73 y 74



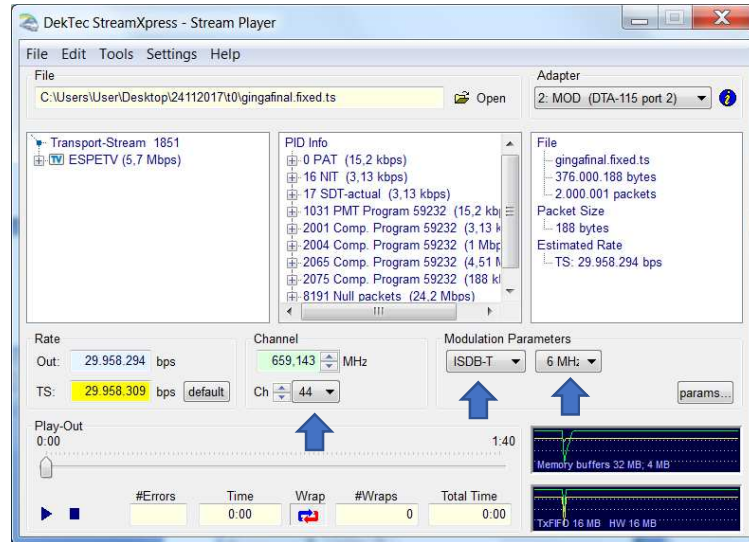
**Figura 73.** Búsqueda en el Explorador



**Figura 74.** Elección del TS

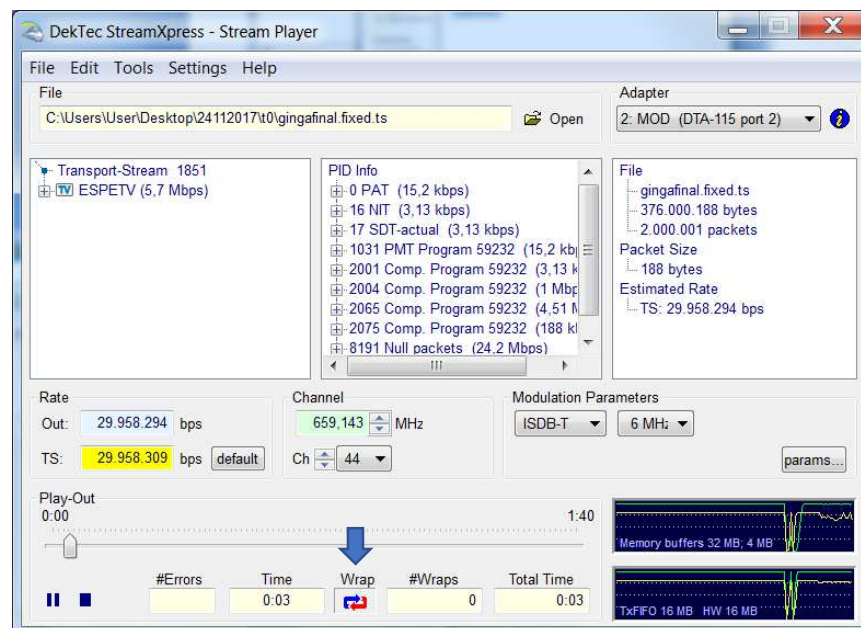
Posteriormente se configura los parámetros de transmisión: canal en el que se va a transmitir, estándar empleado y frecuencia de modulación, Figura 75.





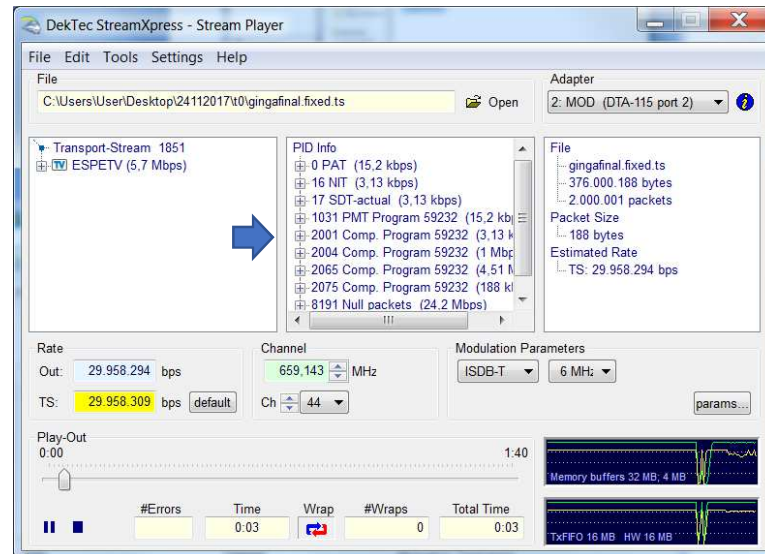
**Figura 75.** Configuración de los parámetros de transmisión

Finalmente se da click en Play y la señal se empieza a transmitir en broadcast no sin antes haber activado la opción de Wrap que ayuda a que la transmisión se encuentre en un bucle infinito hasta que el usuario de click en Stop, Figura 76



**Figura 76.** Transmisión del TS

Es importante mencionar que como en la Figura 77, al cargar el archivo TS, se muestran las tablas y secciones que contiene la interactividad.



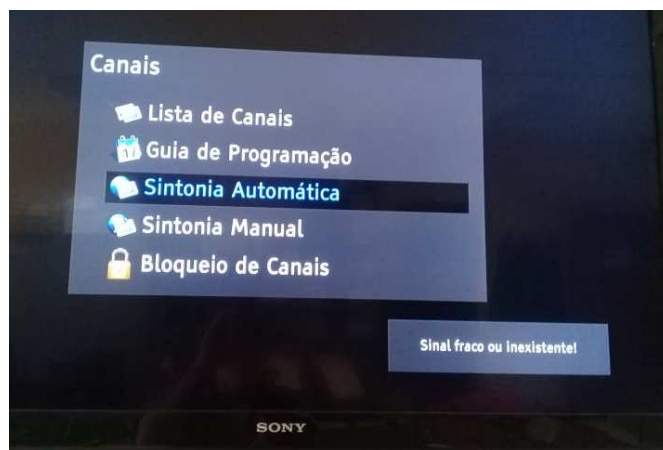
**Figura 77.** Contenido de la aplicación interactiva

El Set Top Box también debe ser configurado para poder recibir la señal proveniente de la tarjeta DTA-115 y para ello al encenderlo se debe hacer un scaneo de frecuencias para de esta manera sintonizar los canales digitales que se están emitiendo por broadcast, esto se hace mientras el StreamXpress se encuentra enviando información, es decir, mientras este en Play. Para esto se presiona en el control remoto la tecla Menú y se selecciona la opción de Canales como se visualiza en la Figura 78

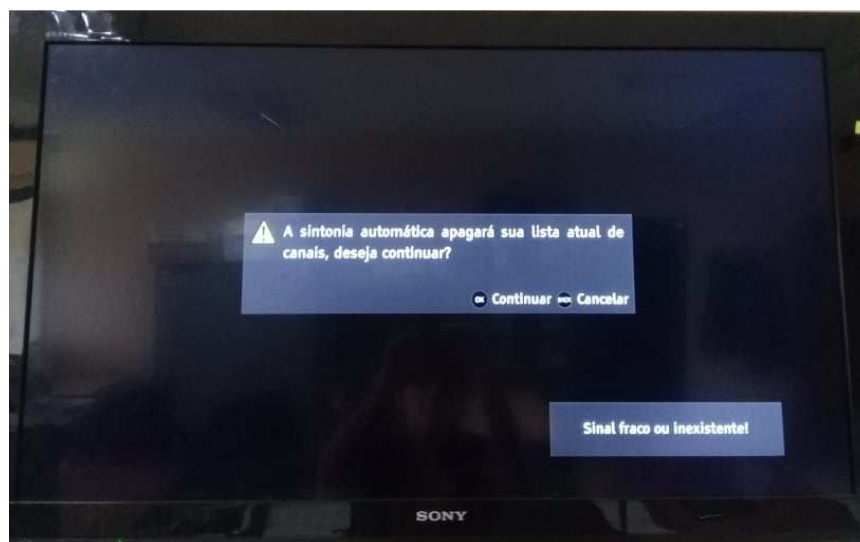


**Figura 78.** Menú del set top box y selección de la opción Canales

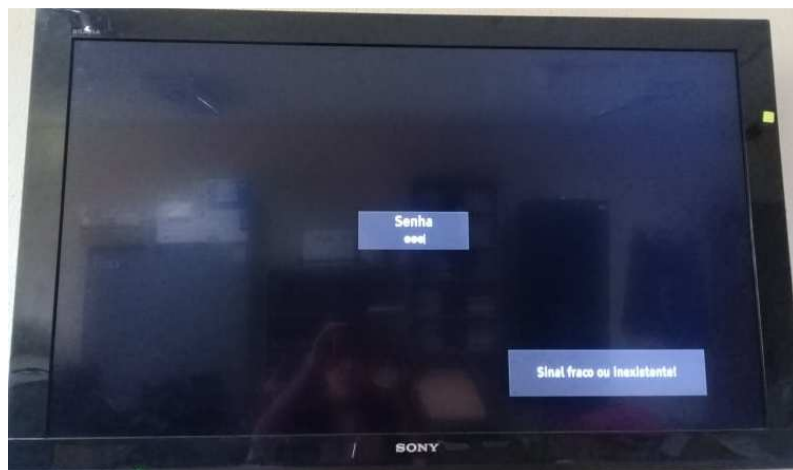
En la pantalla nueva que aparece dentro de la opción de Canales se escoge la sintonización automática, Figura 79, la cual necesita de una confirmación por parte del usuario ya que todos los canales existentes hasta el momento se van a proceder a borrar, Figura 80. La contraseña que pide a continuación por default es 0000, Figura 81



**Figura 79.** Submenú al seleccionar la opción Canales



**Figura 80.** Confirmación necesitada para iniciar la sintonización automática



*Figura 81.* Contraseña para iniciar el proceso

La Figura 82 y 83 muestra que se ha sintonizado de manera correcta el Set Top Box, encontrando el canal en el que se está transmitiendo la señal requerida.

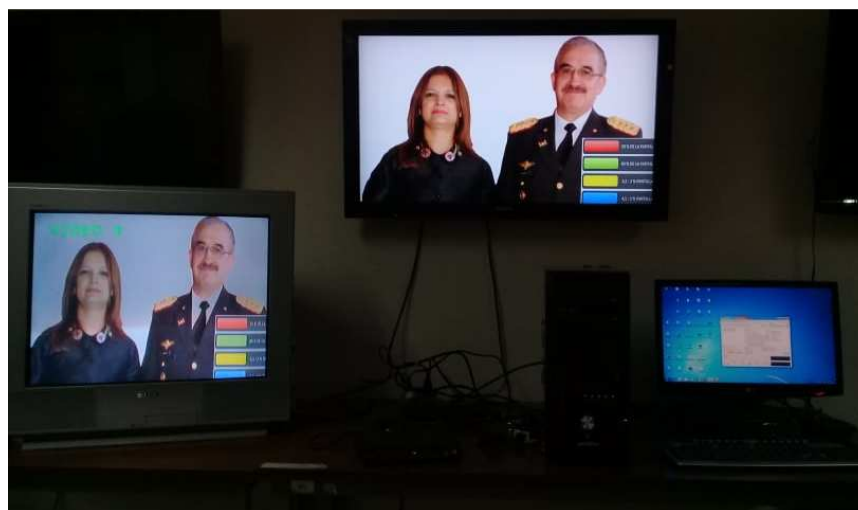


*Figura 82.* Búsqueda de canales, uno encontrado



**Figura 83.** Sintonización exitosa del canal en ambas TVs

Por medio de la Figura 84 se muestra la recepción del servicio de TV Digital en alta definición y su respectiva aplicación interactiva creada a través de Ginga en conjunto con Eclipse



**Figura 84.** Recepción del servicio de TV Digital y la aplicación interactiva

## CAPÍTULO IV

### 4. ANÁLISIS DE RESULTADOS

#### 4.1. Análisis MOS

Este tipo de análisis como ya se explicó de manera breve en el Capítulo II, es un método subjetivo debido a que depende de la percepción que tenga el usuario para visualizar las imágenes presentadas y de esa manera darles una ponderación adecuada en base a lo que se desea evaluar.

Para este objetivo se realizó un tipo de encuesta en donde se dio a conocer al usuario todas las posibilidades que tenía para poder calificar de manera óptima a cada una de las 20 imágenes en sus diferentes presentaciones y dependiendo del número de bloques en el cual se dividió la imagen secreta al ser procesada.

La Tabla 22 presentada a continuación representa de manera general las opciones que tenía el usuario para poder llegar a dar una ponderación óptima a cada una de las imágenes que se le presentaba en los receptores de TV.

**Tabla 22**  
*Opciones presentadas al usuario para el análisis MOS*

<b>20 % DE LA PANTALLA</b>					
5	4	3	2	1	
Excelente Imperceptible	Buena (Casi) perceptible pero no molesto	Regular (Perceptible) y ligeramente molesto	Pobre Molesto (pero no cuestionable)	Mala Muy molesto (cuestionable)	<b>Imagen</b>
					1
					2
					3
					4
					5

Como se puede visualizar la tabla consta de un título, en este caso, 20% de la pantalla, lo cual tiene que ver con el porcentaje que utilizaba la imagen a ser calificada en la pantalla, cabe recalcar que también se lo puede manejar en pixeles, pero al momento de programar todo se trabajo en porcentajes y se lo presentó así en el MOS también para que no genere ningún tipo de confusión. Como al momento de programar se utilizaron los 4 botones del control remoto con sus respectivos colores, se tiene 4 tipos de tamaños de las imágenes pre-procesadas mostradas al usuario, es decir, el título de la tabla cambia para 20% de la pantalla, 40% de la pantalla, 3,5:2 de la pantalla y 4,5:3 de la pantalla respectivamente para cada uno de los colores.

Cada botón contiene en su carrusel de imágenes, 5 imágenes en total, por eso es cada tabla solo posee imagen de la 1 a la 5 para su respectiva evaluación.

Dentro de las opciones o la calificación que el usuario podía dar a la imagen tenemos las siguientes:

- Ponderación de 5: la imagen tendrá esta calificación cuando tenga una calidad excelente, es decir, la distorsión que puede llegar a tener es imperceptible para el usuario que está calificando
- Ponderación de 4: la imagen tendrá esta calificación cuando tenga una calidad buena, es decir, la distorsión que puede llegar a tener es casi perceptible pero no genera molestia alguna al usuario que se encuentra calificando
- Ponderación de 3: la imagen tendrá esta calificación cuando tenga una calidad regular, es decir, la distorsión que puede llegar a tener es perceptible y ligeramente molesta para el usuario que está calificando
- Ponderación de 2: la imagen tendrá esta calificación cuando tenga una calidad pobre, es decir, la distorsión que puede llegar a tener es molesta, pero no genera cuestionamiento alguno al usuario que se encuentra calificando

- Ponderación de 1: la imagen tendrá esta calificación cuando tenga una calidad mala, es decir, la distorsión que puede llegar a tener es muy molesta y por ende es cuestionable al usuario que está calificando

En total en el documento entregado a cada usuario se obtuvo 16 tablas, debido a que se tomó en cuenta la calidad de la imagen recuperada, basándose en el número de bloques en que se dividía la imagen secreta para transformarla a mosaico. Se tomo en cuenta 4 opciones: 4x4, 8x8, 16x16 y 32x32 bloques en que se dividía la imagen y por ende cada una de estas opciones evaluaba las 20 imágenes en total, de las cuales 5 se repartían en cada botón y cada botón tenía su respectiva tabla. Por todo esto es 16 son las tablas que se mostraron al usuario para la ponderación de cada imagen.

Era importante evaluar la percepción en diferentes receptores y para ello se utilizó 3 tipos de televisores: un LCD, un LED y uno de Rayos Catódicos para de esta manera poder comprobar si la calidad de la imagen aparte de depender del número de bloques en que se la dividió, también dependía del tipo de receptor en el cual se visualizó la imagen.

Se tuvo la participación de 30 personas en total, las cuales fueron divididas en grupos de 10 para que cada uno de los grupos evalúe las imágenes pre-procesadas en cada uno de los diferentes receptores mencionados anteriormente. La Figura 85 muestra la implementación del MOS en el escenario de recepción de la aplicación interactiva diseñada en Ginga para la visualización de imágenes pre-procesadas.





**Figura 85.** Implementación del MOS con el grupo de estudiantes

Luego de realizar el MOS en cada uno de los estudiantes dependiendo del tipo de receptor, se procedió a revisar la calificación obtenida para cada una de las imágenes dependiendo del número de bloques en el cual se dividió, obteniendo de esta manera los resultados mostrados en las Tablas de la 23 a la 34. Cabe mencionar que los resultados que se muestran son el promedio en base a las 10 personas que fueron destinadas para cada uno de los grupos

**Tabla 23**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=4*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 4x4	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	4,3
Imagen 2	4,4
Imagen 3	4,3
Imagen 4	4,1
Imagen 5	3,9
Imagen 6	4,3
Imagen 7	4,4
Imagen 8	4,2
Imagen 9	4,2
Imagen 10	4,3

CONTINÚA →

Imagen 11	3,4
Imagen 12	3,6
Imagen 13	3,1
Imagen 14	3,5
Imagen 15	3,2
Imagen 16	3,7
Imagen 17	4,3
Imagen 18	4,1
Imagen 19	4,4
Imagen 20	3,2
<b>Promedio total</b>	<b>3,945</b>

**Tabla 24**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=8*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 8x8	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	4,1
Imagen 2	4,3
Imagen 3	3,9
Imagen 4	3,7
Imagen 5	3,7
Imagen 6	2,8
Imagen 7	2,8
Imagen 8	2,7
Imagen 9	2,3
Imagen 10	2,6
Imagen 11	2,4
Imagen 12	2,4
Imagen 13	2,4
Imagen 14	2,9
Imagen 15	2,4
Imagen 16	2,7
Imagen 17	2,8
Imagen 18	2,4
Imagen 19	2,9
Imagen 20	1,8
<b>Promedio total</b>	<b>2,9</b>

**Tabla 25**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=16*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 16x16	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	2,4
Imagen 2	2,5
Imagen 3	2,4
Imagen 4	1,5

CONTINÚA 

Imagen 5	2,2
Imagen 6	1,5
Imagen 7	1,1
Imagen 8	1,4
Imagen 9	1,4
Imagen 10	1,3
Imagen 11	1,1
Imagen 12	1,3
Imagen 13	1,4
Imagen 14	2,0
Imagen 15	1,3
Imagen 16	1,5
Imagen 17	1,5
Imagen 18	1,4
Imagen 19	2,0
Imagen 20	1,0
<b>Promedio total</b>	<b>1,61</b>

**Tabla 26**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=32*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 32x32	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	1,3
Imagen 2	1,4
Imagen 3	1,3
Imagen 4	1,1
Imagen 5	1,3
Imagen 6	1,3
Imagen 7	1,0
Imagen 8	1,1
Imagen 9	1,2
Imagen 10	1,1
Imagen 11	1,1
Imagen 12	1,1
Imagen 13	1,3
Imagen 14	1,3
Imagen 15	1,2
Imagen 16	1,4
Imagen 17	1,1
Imagen 18	1,1
Imagen 19	1,1
Imagen 20	1,1
<b>Promedio total</b>	<b>1,195</b>

**Tabla 27**

*Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=4*

<b>Tipo de Receptor: Rayos Catódicos</b>	
Número de bloques en el que se dividió la imagen secreta: 4x4	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	2,9
Imagen 2	3,8
Imagen 3	2,4
Imagen 4	2,7
Imagen 5	2,6
Imagen 6	3,3
Imagen 7	3,8
Imagen 8	4,4
Imagen 9	3,6
Imagen 10	4,3
Imagen 11	1,3
Imagen 12	2,1
Imagen 13	1,4
Imagen 14	1,3
Imagen 15	1,7
Imagen 16	3
Imagen 17	4
Imagen 18	3,6
Imagen 19	3,8
Imagen 20	1,8
<b>Promedio total</b>	<b>2,9</b>

**Tabla 28**

*Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=8*

<b>Tipo de Receptor: Rayos Catódicos</b>	
Número de bloques en el que se dividió la imagen secreta: 8x8	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	2,2
Imagen 2	3,3
Imagen 3	2,2
Imagen 4	2
Imagen 5	1,8
Imagen 6	2,9
Imagen 7	2,8
Imagen 8	3,2
Imagen 9	1,7
Imagen 10	2,7
Imagen 11	1,3
Imagen 12	1,8
Imagen 13	1,3
Imagen 14	1,3
Imagen 15	1,9
Imagen 16	2,8
Imagen 17	3,1

CONTINÚA 

Imagen 18	2,6
Imagen 19	3
Imagen 20	1,2
<b>Promedio total</b>	<b>2,26</b>

**Tabla 29**

*Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=16*

<b>Tipo de Receptor: Rayos Catódicos</b>	
Número de bloques en el que se dividió la imagen secreta: 16x16	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	1,8
Imagen 2	2,4
Imagen 3	1,8
Imagen 4	1,2
Imagen 5	1,1
Imagen 6	1,2
Imagen 7	1
Imagen 8	1,1
Imagen 9	1,1
Imagen 10	1,1
Imagen 11	1
Imagen 12	1
Imagen 13	1,2
Imagen 14	1,8
Imagen 15	1,2
Imagen 16	1,4
Imagen 17	1,2
Imagen 18	1,1
Imagen 19	1,4
Imagen 20	1
<b>Promedio total</b>	<b>1,32</b>

**Tabla 30**

*Resultados obtenidos en la evaluación MOS del televisor de tipo Rayos Catódicos con N=32*

<b>Tipo de Receptor: Rayos Catódicos</b>	
Número de bloques en el que se dividió la imagen secreta: 32x32	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	1,2
Imagen 2	1,1
Imagen 3	1,3
Imagen 4	1
Imagen 5	1,3
Imagen 6	1,1
Imagen 7	1
Imagen 8	1
Imagen 9	1
Imagen 10	1,1

CONTINÚA 

Imagen 11	1
Imagen 12	1
Imagen 13	1
Imagen 14	1,4
Imagen 15	1,1
Imagen 16	1,2
Imagen 17	1
Imagen 18	1,1
Imagen 19	1,3
Imagen 20	1
<b>Promedio total</b>	<b>1,12</b>

**Tabla 31**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=4*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 4x4	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	4,7
Imagen 2	4,3
Imagen 3	4
Imagen 4	3,2
Imagen 5	3
Imagen 6	3,8
Imagen 7	4,2
Imagen 8	3,9
Imagen 9	4,1
Imagen 10	3,9
Imagen 11	2,8
Imagen 12	3,2
Imagen 13	2,7
Imagen 14	3,1
Imagen 15	3,9
Imagen 16	3,7
Imagen 17	4
Imagen 18	3,6
Imagen 19	4,8
Imagen 20	2,9
<b>Promedio total</b>	<b>3,68</b>

**Tabla 32**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=8*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 8x8	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	4
Imagen 2	3,8
Imagen 3	3,1
Imagen 4	2,8
Imagen 5	2,8

CONTINÚA 

Imagen 6	2,4
Imagen 7	2,8
Imagen 8	2,3
Imagen 9	1,8
Imagen 10	2,6
Imagen 11	1,9
Imagen 12	1,9
Imagen 13	2,3
Imagen 14	2,2
Imagen 15	2,6
Imagen 16	2,8
Imagen 17	2,8
Imagen 18	2,7
Imagen 19	2,9
Imagen 20	1,4
<b>Promedio total</b>	<b>2,59</b>

**Tabla 33**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=16*

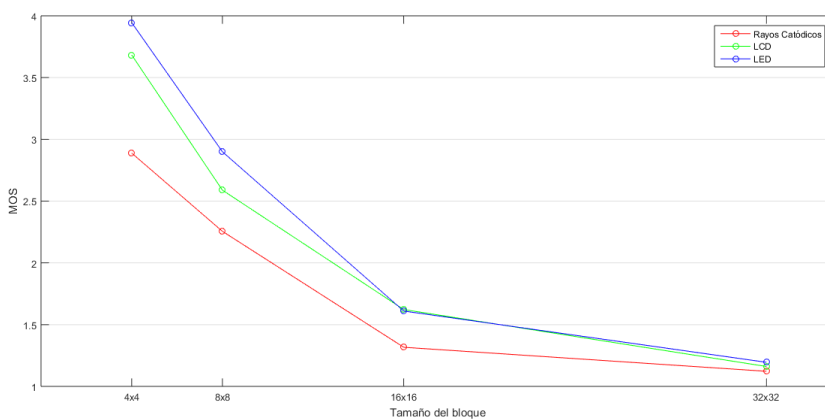
<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 16x16	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	2,1
Imagen 2	2,6
Imagen 3	2,2
Imagen 4	1,8
Imagen 5	2
Imagen 6	1,1
Imagen 7	1
Imagen 8	1,2
Imagen 9	1
Imagen 10	1,3
Imagen 11	1
Imagen 12	1,5
Imagen 13	1,7
Imagen 14	2
Imagen 15	1,3
Imagen 16	2,1
Imagen 17	1,8
Imagen 18	1,7
Imagen 19	1,8
Imagen 20	1,1
<b>Promedio total</b>	<b>1,62</b>

**Tabla 34**

*Resultados obtenidos en la evaluación MOS del televisor de tipo LED con N=32*

<b>Tipo de Receptor: LED</b>	
Número de bloques en el que se dividió la imagen secreta: 32x32	
<b>Imagen</b>	<b>Promedio de calificación</b>
Imagen 1	1
Imagen 2	1,5
Imagen 3	1,2
Imagen 4	1
Imagen 5	1,3
Imagen 6	1,1
Imagen 7	1
Imagen 8	1
Imagen 9	1
Imagen 10	1,2
Imagen 11	1
Imagen 12	1
Imagen 13	1,4
Imagen 14	1,5
Imagen 15	1,1
Imagen 16	1,2
Imagen 17	1
Imagen 18	1,2
Imagen 19	1
Imagen 20	1,2
<b>Promedio total</b>	<b>1,16</b>

En la Figura 86 se muestra la comparación de resultados del análisis MOS dependiendo del tamaño del bloque en el que se divide la imagen secreta y el tipo de receptor en el que se visualizó las imágenes pre-procesadas.



**Figura 86.** Comparación del análisis MOS dependiendo del tamaño del bloque y el tipo de receptor



Se puede notar que, dependiendo del tipo de receptor, el que mayor calificación obtuvo es el del tipo de pantalla LED, sin tomar en cuenta aun el tamaño que tiene el bloque al ser dividido, esto debido a que las características que posee este tipo de pantalla supera considerablemente a las dos opciones restantes en las cuales se trabajó.

En cuanto a lo que respecta al tamaño del bloque se tiene una mejora en calidad en un tamaño de 4x4 con una ponderación dada por el usuario en promedio de 3.68 aproximadamente, si bien es cierto, frente a los demás tamaños del bloque siempre el de 4x4 es superior, pero en específico en el receptor de tipo LED se obtiene el valor mencionado anteriormente debido a que en conjunto se trabaja con una pantalla con características mejoradas y un tamaño de bloque óptimo que permite que la imagen secreta sea dividida en menos bloques para su correcta recuperación posteriormente.

#### **4.2.Análisis RMSE**

Al contrario del método de análisis presentado anteriormente, este es un método objetivo debido a que se tiene un resultado obtenido de la comparación de dos imágenes, el cual se lo calcula mediante una fórmula matemática que representa la raíz cuadrada de la sumatoria de los errores cuadráticos. De esta manera, se mide el promedio de errores al cuadrado, es decir, la diferencia que existe entre un conjunto de datos originales (imagen secreta original) y lo que se estima (imagen recuperada a través del mosaico).

Para obtener resultados óptimos se realizó un programa en Matlab, el cual nos devuelve el valor RMSE entre dos imágenes. A continuación, se muestran las líneas de código implementadas:

```
data=imread('Imgs/AA.jpg'); %original - secreta
```

```
estimate=imread('N4/i1.jpg'); %recuperada
valuermse = sqrt( sum( (data(:)-estimate(:)).^2) / numel(data) )
```

En data y en estimate se leen cada una de las imágenes que se van a utilizar para la correcta comparación entre ambas. Data almacena la imagen tomada como la secreta original, es decir, antes de ser procesada para convertirse en mosaico. Estimate por su parte, almacena la imagen recuperada a través del mosaico.

Es importante mencionar que en cada uno de estos parámetros se va cambiando el nombre de la imagen que se desea comparar. En data se tiene desde la imagen AA hasta la imagen TT y en estimate se tiene desde la imagen i1 hasta i20, cada una de las cuales se van alternando de manera conjunta, por ejemplo, AA se compara con i1, BB se compara con i2, CC se compara con i3 y así respectivamente. La carpeta N4 se refiere al número de bloques en que se dividieron las imágenes secretas para ser procesadas, por ende, se tendrá desde la carpeta N4 hasta la carpeta N32, teniendo en total 4 carpetas, cada una de las cuales contiene 20 imágenes.

En valuermse se va actualizando el valor correspondiente que va a ser utilizado para el análisis posterior, en esta línea de código primero se realiza la diferencia entre la imagen original y la imagen recuperada, estos errores posteriormente se elevan al cuadrado para luego realizar el promedio a través del comando sum para la sumatoria y dividiendo para el número de elementos obtenidos mediante numel, el cual devuelve el número de elementos en un array. Finalmente se saca la raíz cuadrada de todo lo mencionado anteriormente a través de sqrt.

Utilizando este programa para cada una de las imágenes dependiendo del número de bloques en que se la dividió previamente, se obtuvo los resultados mostrados en las Tablas 35, 36, 37 y 38.

**Tabla 35***Resultados obtenidos en el método RMSE con N=4*

Número de bloques en el que se dividió la imagen secreta: 4x4	
<b>Imagen</b>	<b>RMSE</b>
Imagen 1	3,9997
Imagen 2	2,3407
Imagen 3	4,2698
Imagen 4	4,6071
Imagen 5	3,3588
Imagen 6	3,1023
Imagen 7	4,1013
Imagen 8	2,4574
Imagen 9	5,0591
Imagen 10	3,736
Imagen 11	2,6081
Imagen 12	5,4687
Imagen 13	3,7652
Imagen 14	5,3332
Imagen 15	2,9044
Imagen 16	3,2877
Imagen 17	4,8588
Imagen 18	3,1186
Imagen 19	5,1192
Imagen 20	5,774
<b>Promedio total</b>	<b>3,963505</b>

**Tabla 36***Resultados obtenidos en el método RMSE con N=8*

Número de bloques en el que se dividió la imagen secreta: 8x8	
<b>Imagen</b>	<b>RMSE</b>
Imagen 1	4,6201
Imagen 2	3,0195
Imagen 3	4,9046
Imagen 4	4,9236
Imagen 5	4,4896
Imagen 6	4,0901
Imagen 7	5,0999
Imagen 8	3,2941
Imagen 9	5,4654
Imagen 10	4,9819
Imagen 11	3,3317
Imagen 12	6,4790
Imagen 13	5,0375
Imagen 14	5,6203
Imagen 15	3,9351
Imagen 16	4,1993
Imagen 17	5,3852
Imagen 18	3,9324
Imagen 19	5,8972
Imagen 20	6,8057
<b>Promedio total</b>	<b>4,77561</b>

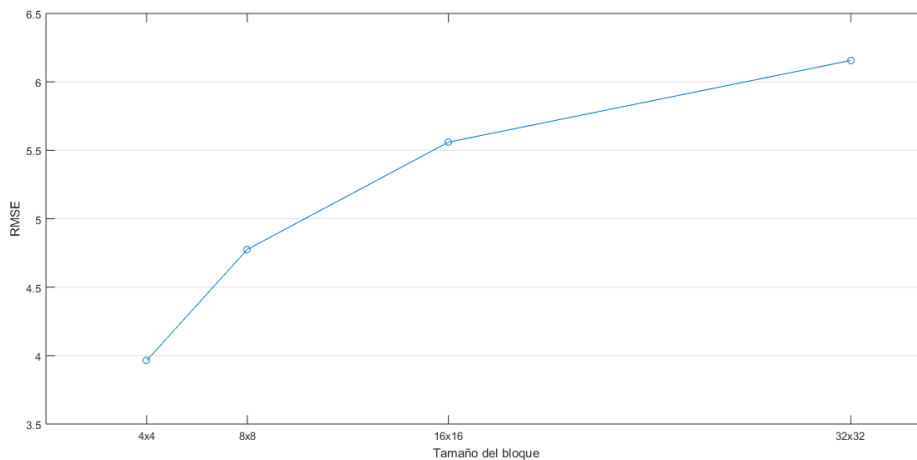
**Tabla 37***Resultados obtenidos en el método RMSE con N=16*

Número de bloques en el que se dividió la imagen secreta: 16x16	
<b>Imagen</b>	<b>RMSE</b>
Imagen 1	5,2698
Imagen 2	3,811
Imagen 3	5,5349
Imagen 4	5,1732
Imagen 5	5,4225
Imagen 6	5,0296
Imagen 7	5,9178
Imagen 8	4,2304
Imagen 9	5,9552
Imagen 10	6,223
Imagen 11	3,983
Imagen 12	7,555
Imagen 13	6,3745
Imagen 14	5,9673
Imagen 15	4,6801
Imagen 16	5,157
Imagen 17	5,891
Imagen 18	4,8714
Imagen 19	6,4777
Imagen 20	7,6729
<b>Promedio total</b>	<b>5,559865</b>

**Tabla 38***Resultados obtenidos en el método RMSE con N=32*

Número de bloques en el que se dividió la imagen secreta: 32x32	
<b>Imagen</b>	<b>RMSE</b>
Imagen 1	5,6432
Imagen 2	4,5159
Imagen 3	6,0125
Imagen 4	5,2111
Imagen 5	6,023
Imagen 6	5,7224
Imagen 7	6,5786
Imagen 8	5,254
Imagen 9	6,227
Imagen 10	7,6144
Imagen 11	4,2318
Imagen 12	8,3278
Imagen 13	7,5703
Imagen 14	6,2817
Imagen 15	5,1449
Imagen 16	5,8635
Imagen 17	6,417
Imagen 18	5,6781
Imagen 19	6,698
Imagen 20	8,1205
<b>Promedio total</b>	<b>6,156785</b>

En la Figura 87 se muestra la comparación de los valores promedios obtenidos a través del método objetivo RMSE dependiendo del tamaño del bloque.



**Figura 87.** Comparación del análisis RMSE dependiendo del tamaño del bloque

Se puede visualizar que el tamaño del bloque de 4x4 es el que menor error posee en comparación con los demás, con un valor RMSE de 3.963505, debido a que como la imagen secreta se divide únicamente en 4 bloques, es más fácil poderla recuperar posteriormente a partir de la imagen mosaico, sin perder mucha calidad de la imagen al momento de hacerlo.

### 4.3. Análisis MSSIM

Este método al igual que el anterior es un método objetivo debido a que es una métrica que permite medir cuanta similitud tiene una imagen comparada con otra, se toma en cuenta tres componentes fundamentales, los cuales se explicaron de manera previa en el Capítulo II, por ende, se hace un análisis mucho más preciso que el RMSE mencionado anteriormente.

Para obtener resultados óptimos se realizó un programa en Matlab, el cual nos devuelve el valor SSIM entre dos imágenes, mediante los cuales de manera posterior se realizará el promedio para de esta manera obtener el valor MSSIM. A continuación, se muestran las líneas de código implementadas:

```
A=imread('N4/i1.jpg'); %recuperada  
ref=imread('Imgs/AA.jpg'); %original - secreta  
  
valuessim = ssim(A,ref)
```

En A y en ref se leen cada una de las imágenes que se van a utilizar. A almacena la imagen recuperada por medio del mosaico y ref guarda la imagen secreta original, es decir, antes de ser procesada para convertirse en mosaico.

Es importante mencionar que en ambos parámetros se va cambiando el nombre de la imagen que se desea comparar. En ref se tiene desde la imagen AA hasta la imagen TT y en A se tiene desde la imagen i1 hasta i20, cada una de las cuales se van alternando de manera conjunta, por ejemplo, AA se compara con i1, BB se compara con i2, CC se compara con i3 y así respectivamente. La carpeta N4 se refiere al número de bloques en que se dividieron las imágenes secretas para ser procesadas, por ende, se tendrá desde la carpeta N4 hasta la carpeta N32, teniendo en total 4 carpetas, cada una de las cuales contiene 20 imágenes.

En valuessim se va actualizando el valor correspondiente que va a ser utilizado para el análisis posterior, en esta línea de código se utiliza el comando ssim, el cual calcula y por ende nos devuelve el valor del Índice de Similitud Estructural de una imagen A usando como referencia la imagen en ref.

Utilizando este programa para cada una de las imágenes dependiendo del número de bloques en que se la dividió previamente, se obtuvo los resultados mostrados en las Tablas 39, 40, 41 y 42.

**Tabla 39***Resultados obtenidos en el método MSSIM con N=4*

Número de bloques en el que se dividió la imagen secreta: 4x4	
<b>Imagen</b>	<b>SSIM</b>
Imagen 1	0,9479
Imagen 2	0,9791
Imagen 3	0,9375
Imagen 4	0,8258
Imagen 5	0,9317
Imagen 6	0,9536
Imagen 7	0,9461
Imagen 8	0,9371
Imagen 9	0,8396
Imagen 10	0,9482
Imagen 11	0,9346
Imagen 12	0,9081
Imagen 13	0,9539
Imagen 14	0,8867
Imagen 15	0,9855
Imagen 16	0,9581
Imagen 17	0,9123
Imagen 18	0,9562
Imagen 19	0,8552
Imagen 20	0,8246
<b>Promedio total (MSSIM)</b>	<b>0,92109</b>

**Tabla 40***Resultados obtenidos en el método MSSIM con N=8*

Número de bloques en el que se dividió la imagen secreta: 8x8	
<b>Imagen</b>	<b>SSIM</b>
Imagen 1	0,9212
Imagen 2	0,965
Imagen 3	0,9001
Imagen 4	0,7848
Imagen 5	0,8703
Imagen 6	0,9174
Imagen 7	0,9095
Imagen 8	0,8786
Imagen 9	0,7988
Imagen 10	0,8962
Imagen 11	0,8953
Imagen 12	0,8478
Imagen 13	0,9048
Imagen 14	0,8623
Imagen 15	0,9698
Imagen 16	0,9349
Imagen 17	0,8773
Imagen 18	0,9283
Imagen 19	0,799
Imagen 20	0,7301
<b>Promedio total (MSSIM)</b>	<b>0,879575</b>

**Tabla 41***Resultados obtenidos en el método MSSIM con N=16*

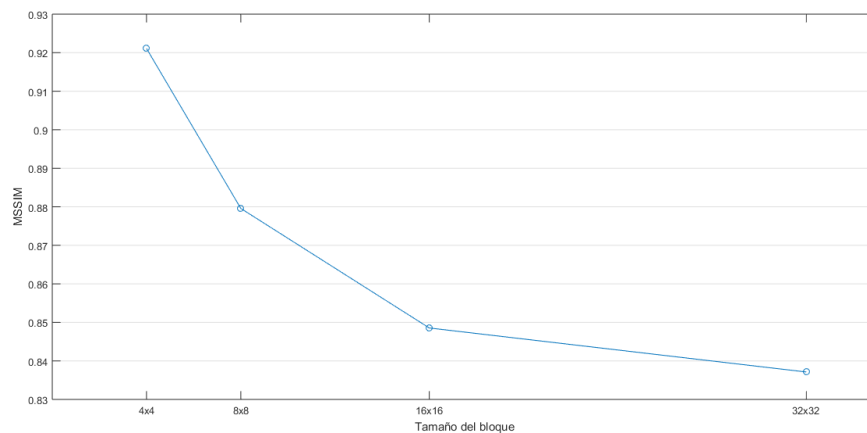
Número de bloques en el que se dividió la imagen secreta: 16x16	
<b>Imagen</b>	<b>SSIM</b>
Imagen 1	0,8939
Imagen 2	0,9507
Imagen 3	0,8635
Imagen 4	0,774
Imagen 5	0,8228
Imagen 6	0,8901
Imagen 7	0,8879
Imagen 8	0,8236
Imagen 9	0,7689
Imagen 10	0,849
Imagen 11	0,8747
Imagen 12	0,8005
Imagen 13	0,8509
Imagen 14	0,8454
Imagen 15	0,9586
Imagen 16	0,9125
Imagen 17	0,8633
Imagen 18	0,8998
Imagen 19	0,7703
Imagen 20	0,6705
<b>Promedio total (MSSIM)</b>	<b>0,848545</b>

**Tabla 42***Resultados obtenidos en el método MSSIM con N=32*

Número de bloques en el que se dividió la imagen secreta: 32x32	
<b>Imagen</b>	<b>SSIM</b>
Imagen 1	0,8827
Imagen 2	0,9401
Imagen 3	0,8313
Imagen 4	0,7869
Imagen 5	0,804
Imagen 6	0,8764
Imagen 7	0,8856
Imagen 8	0,7878
Imagen 9	0,7728
Imagen 10	0,8168
Imagen 11	0,8809
Imagen 12	0,7767
Imagen 13	0,812
Imagen 14	0,8353
Imagen 15	0,9529
Imagen 16	0,8993
Imagen 17	0,869
Imagen 18	0,8818
Imagen 19	0,7809
Imagen 20	0,6692
<b>Promedio total (MSSIM)</b>	<b>0,83712</b>



En la Figura 88 se muestra la comparación de los valores promedios obtenidos a través del método objetivo MSSIM dependiendo del tamaño del bloque en el que de manera previa se dividió a la imagen secreta para que posteriormente se transforme en imagen mosaico



**Figura 88.** Comparación del análisis MSSIM dependiendo del tamaño del bloque

Se puede visualizar que una vez más, el tamaño del bloque más óptimo y por ende el que ayuda a obtener una mejor calidad de imagen es el de 4x4 con un valor de Índice de Similitud Estructural Media (MSSIM) de 0.92109, siendo este el más cercano a 1 (mayor similitud) y por ende llegando a tener una similitud suficientemente buena entre ambas imágenes comparadas. Este tipo de análisis ayuda a obtener con mayor precisión una comparación adecuada debido a que toma en cuenta la comparación de luminancia, la distorsión del contraste y la similitud estructural en sí.

## CAPÍTULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

- Se ha realizado el análisis de calidad por medio de métodos cualitativos y cuantitativos, de imágenes pre-procesadas mediante transformaciones de color reversibles en un escenario real TVi compuesto del servidor de aplicaciones y contenidos, servidor de playout y un decodificador o set top box; teniendo como resultado que de manera general el tamaño del bloque adecuado para obtener una mejor calidad en la recuperación de la imagen secreta a partir de la imagen mosaico es el de 4x4 debido a que se divide la imagen en una cantidad menor de partes que en las demás opciones trabajadas y por ende, se facilita la correcta recuperación después del procesamiento; cabe mencionar que el tipo de receptor en el que se visualiza la imagen es también un factor importante para que la calidad sea mejor, siendo el mejor el televisor de tipo LED en comparación con las dos opciones restantes.
- La aplicación interactiva desarrollada en GINGA favoreció la correcta observación de cada una de las imágenes en sus diferentes tamaños y regiones (ubicación en la pantalla), de manera que la programación seleccionada para la aplicación interactiva, no interfiera en el análisis que se va a realizar de manera posterior.
- La correcta generación del Transport Stream mediante la utilización del software OpenCaster en Ubuntu facilitó la transmisión de las diferentes imágenes incluidas en la interactividad para que de esta manera puedan ser transmitidas a cada tipo de receptor (televisor), entre los cuales se tenía: Rayos Catódicos, LED y LCD.

- Se ha realizado el método de análisis MOS por medio de la utilización de encuestas entregadas a un grupo de usuarios, de las cuales se procesó la información de manera óptima para obtener una puntuación media en base a la ponderación dada de manera subjetiva por cada participante, obteniendo como resultado de manera general que el mejor tamaño de bloque al momento de procesar la imagen secreta es el de 4x4 y el mejor receptor es el de tipo LED debido a que brinda mejores características que los dos restantes con los que se trabajó, todo esto para que al momento de recuperar la imagen secreta a partir de la imagen mosaico se tenga una mejor calidad comparando con la imagen secreta original.
- Al utilizar el método de análisis objetivo RMSE con la ayuda de la programación realizada en Matlab se pudo notar eficazmente que, para obtener una mejora de calidad significativa, el tamaño de bloque que se debe utilizar es el de 4x4 debido a que en éste la imagen secreta es mucho más fácil de recuperar teniendo como origen la imagen mosaico, dando como resultado que la imagen recuperada tenga menor cantidad de errores en comparación a los que se obtuvieron al aumentar el tamaño del bloque.
- El método objetivo que ayuda a obtener una mejor precisión de calidad en base a la comparación de la imagen secreta recuperada y la imagen secreta original es el del análisis MSSIM debido a que toma en cuenta más aspectos relevantes que únicamente la similitud estructural que puedan tener ambas imágenes en cuestión, de esta manera con la ayuda de Matlab se llegó a conocer y confirmar que al utilizar el tamaño de bloque 4x4 se llegó a obtener un mayor acercamiento al valor de 1 (mayor similitud), es decir, la semejanza que tenía la una imagen con la otra era lo suficientemente buena como para evaluar a este método como el de mayor exactitud

- El algoritmo que se propuso en (Acosta, 2017) para reducir la cantidad de bits requeridos al momento de recuperar una imagen secreta permite tener una optimización del ancho de banda al momento de transmitir, estimando parámetros estadísticos de la imagen mosaico para de esta manera obtener los parámetros necesarios de la imagen secreta. El uso del mismo permitió obtener una reducción de 3 a 1 sin la necesidad de perder calidad en la imagen recuperada, sin embargo, por medio del análisis realizado en este trabajo se pudo determinar que el tamaño del bloque es indispensable para que la calidad de la imagen sea mejor así como también la visualización dependiendo del receptor que se escoja al momento de transmitir.

## **5.2.Recomendaciones**

- Es importante ir realizando pruebas en la transmisión a medida que se vaya diseñando la aplicación interactiva, debido a que el tamaño de la pantalla de uno u otro receptor no es el mismo y por ende la relación de aspecto del mismo perjudica a la visualización optima en cuanto a botones o cualquier tipo de multimedia que se esté utilizando.
- Es necesario estar al tanto de manera previa sobre la correcta sintaxis que poseen los lenguajes de programación destinados a realizar aplicaciones interactivas para TV Digital, en este caso, conocer la manera en que se programa en Ginga NCL.
- Al momento de generar el Transport Stream es importante tomar en cuenta cada uno de los parámetros que el video que se va a utilizar posee como, por ejemplo, la duración, la cantidad de fotogramas por segundo, la velocidad de datos, etc. debido a que son datos importantes para obtener un correcto TS al momento de multiplexar el video, el audio y la aplicación interactiva.

- Conocer qué tipo de decodificador es el adecuado para que soporte aplicaciones interactivas y las diferentes necesidades que se puede llegar a tener para brindar al usuario una mayor interactividad
- Tomar en cuenta que para poder configurar el set top box y recibir la señal que proviene de la tarjeta moduladora DTA-115, se debe hacer un barrido o escaneo de frecuencias para sintonizar los canales y para esto el software StreamXpress debe estar enviando información, es decir, debe estar en Play, caso contrario no se encontrará ningún canal para poder transmitir.
- Es indispensable que en si la aplicación diseñada no posea tanto contenido que pueda llamar mucho la atención del usuario debido a que en si la programación cotidiana del televisor debe ser el centro de atención de la persona que esta interactuando con la aplicación mencionada.

## 6. BIBLIOGRAFÍA

- Avalpa Digital Engineering Srl. (2018). OpenCaster 3.2.2: the free digital tv software. Retrieved January 7, 2018, from <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- Beatrizxe. (2016). Los 11 tipos de formatos de imágenes más usados. Retrieved December 26, 2017, from <https://beatrizxe.com/es/blog-diseno-ilustracion/blog-diseno-grafico/11-tipos-formatos-imagenes-mas-usados.html>
- Carballar Falcon, J. A. (2008). *VoIP : la telefonía de Internet*. Paraninfo.
- Cuéllar, J. C., Ortiz, J. H., & Arciniegas, J. L. (2014). Clasificación y Análisis de Métodos para medir Calidad de la Experiencia del Servicio de Televisión sobre Protocolo IP (IPTV). *Información Tecnológica*, 25(5), 121–128. <https://doi.org/10.4067/S0718-07642014000500017>
- Dávila Sacoto, M. A. (2012). Diseño de una plataforma de software para televisión digital interactiva de un canal de deportes utilizando Ginga-NCL Lua. Retrieved from <https://dspace.ups.edu.ec/handle/123456789/1736>
- DekTec. (2011). DTA-115.
- DekTec. (2017). DTC-300-SP StreamXpress Software.
- Disqus. (2017). PSQM—helpes.eu. Retrieved December 26, 2017, from <http://www6.helpes.eu/01104780/PSQM>
- Durán, A. (2014). PNG, GIF, o JPEG. ¿Cuál es el mejor formato de imagen para los emails? Retrieved December 25, 2017, from <http://blog.masterbase.com/blog-masterbase/png-gif-o-jpeg.-cuál-es-el-mejor-formato-de-imagen-para-los-emails>
- Escalante Ramírez, B. (2006). Procesamiento Digital de Imágenes. Retrieved from <http://verona.fi-p.unam.mx/boris/teachingnotes/Introduccion.pdf>

- Ginga. (2017a). inicio | Ginga. Retrieved December 25, 2017, from <http://www.ginga.org.br/es/inicio>
- Ginga. (2017b). inicio | Ginga-NCL. Retrieved December 25, 2017, from <http://www.gingancl.org.br/en>
- González Arribas, Á. (2017). Tipos de imágenes y formatos - APRENDE TIC. Retrieved December 25, 2017, from <https://sites.google.com/site/ticvalcarcel/optimizacion-de-imagenes-para-internet/tipos-de-imagenes-y-formatos>
- Gráfica. (2014). TIFF, JPEG, GIF, PNG... ¿Por qué guardar en un formato u otro? Retrieved December 25, 2017, from <https://graffica.info/tiff-jpeg-gif-png-formatos-archivos/>
- Jiménez Rodríguez, M. J. (2017a). Procesamiento en el Dominio de la Frecuencia. *Universidad de Sevilla*. Retrieved from <http://asignatura.us.es/imagendigital/Tema3-Frecuencia.pdf>
- Jiménez Rodríguez, M. J. (2017b). Procesamiento en el Dominio Espacial (Parte 1). *Universidad de Sevilla*.
- Jiménez Rodríguez, M. J. (2017c). Procesamiento en el Dominio Espacial (Parte 2). *Universidad de Sevilla*.
- Jiménez Rodríguez, M. J. (2017d). Segmentación. *Universidad de Sevilla*. Retrieved from [http://asignatura.us.es/imagendigital/Tema5-1\\_SegmentacionDiscontinuidades.pdf](http://asignatura.us.es/imagendigital/Tema5-1_SegmentacionDiscontinuidades.pdf)
- Joskowicz, J., & Sotelo, R. (n.d.). Medida de la calidad de voz en redes IP. Retrieved from [https://iee.fing.edu.uy/~josej/docs/Medida de la calidad de voz en redes IP.pdf](https://iee.fing.edu.uy/~josej/docs/Medida%20de%20la%20calidad%20de%20voz%20en%20redes%20IP.pdf)
- Konectado. (2017). Qué es la Medición de la calidad de audio y cómo realizarlas?, 593–9. Retrieved from [http://www.konectado.com/file/Medicion\\_MOS.pdf](http://www.konectado.com/file/Medicion_MOS.pdf)

Lee, Y.-L., & Tsai, W.-H. (2014). A New Secure Image Transmission Technique via Secret-Fragment-Visible Mosaic Images by Nearly Reversible Color Transformations. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(4), 695–703. <https://doi.org/10.1109/TCSVT.2013.2283431>

MathWorks. (1994). Write image to graphics file - MATLAB imwrite - MathWorks España. Retrieved January 4, 2018, from <https://es.mathworks.com/help/matlab/ref/imwrite.html>

Ministerio de Telecomunicaciones y de la Sociedad de la Información. (2017). ¿Qué es la TDT? - TDT - Televisión Digital Terrestre. Retrieved December 25, 2017, from <https://tdtecuador.mintel.gob.ec/que-es-la-tdt/>

Rifai, H., Mohammed, S., & Mellouk, A. (2011). A brief synthesis of QoS-QoE methodologies. In *2011 10th International Symposium on Programming and Systems* (pp. 32–38). IEEE. <https://doi.org/10.1109/ISPS.2011.5898880>

Salvador Moreno, U. (2012). Estudio experimental de calidad de servicio de voz sobre IP: comparativa subjetiva versus objetiva.

Sola, I., González-Audicana, M., Álvarez-Mozos, J., & Torres, J. L. (2014). Evaluación multitemporal de métodos de corrección topográfica mediante el uso de imágenes sintéticas multispectrales Multitemporal evaluation of topographic correction methods using multispectral synthetic images. <https://doi.org/10.4995/raet.2014.2246>

Villamarín, D., Illescas, M. A., Olmedo, G., & Cueva, R. L. (2013). Generating a transport stream for digital terrestrial television system in conformance with ISDB-Tb standard. *2013 IEEE Colombian Conference on Communications and Computing, COLCOM 2013 - Conference Proceedings*, (May 2015). <https://doi.org/10.1109/ColComCon.2013.6564814>

Wainschenker, R., Massa, J. M., & Tristan, P. (2011). Procesamiento Digital de Imágenes. Retrieved from <http://www.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf>

Winkler, S. (2009). Video quality measurement standards — Current status and trends. In *2009 7th International Conference on Information, Communications*



*and Signal Processing (ICICS)* (pp. 1–5). IEEE.  
<https://doi.org/10.1109/ICICS.2009.5397585>

Zamzar. (2006a). CDR a JPG | Zamzar - Conversión gratis de archivos en línea.  
Retrieved December 26, 2017, from <http://www.zamzar.com/es/convert/cdr-to-jpg/>

Zamzar. (2006b). DXF a PDF | Zamzar - Conversión gratis de archivos en línea.  
Retrieved December 26, 2017, from <http://www.zamzar.com/es/convert/dxf-to-pdf/>

Zamzar. (2006c). EMF a JPG | Zamzar - Conversión gratis de archivos en línea.  
Retrieved December 26, 2017, from <http://www.zamzar.com/es/convert/emf-to-jpg/>

Zamzar. (2006d). ODG a JPG | Zamzar - Conversión gratis de archivos en línea.  
Retrieved December 26, 2017, from <http://www.zamzar.com/es/convert/odg-to-jpg/>