



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

ELECTRÓNICA Y COMUNICACIONES

TEMA:

**MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG
MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN
DE DATOS.**

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

AUTOR: Wilson Daniel Vasco Gómez

TUTOR: Ing. Luis Pomaquero

Ambato – Ecuador

Septiembre 2012

APROBACION DEL TUTOR

En calidad de Tutor del Trabajo de Investigación sobre el tema:

“MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS”, de Wilson Daniel Vasco Gómez, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometidos a la evaluación del Tribunal de Grado, que el Honorable Consejo Directivo designe, para su correspondiente estudio y calificación.

Ambato, 10 de Septiembre 2012

EL TUTOR

Ing. Luis Pomaquero

AUTORÍA

El presente trabajo de investigación “MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS” es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, 10 de Septiembre de 2012

.....

Wilson Daniel Vasco Gómez

C.C. 1804110458

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Mario García e Ing. Franklin Silva, revisó y aprobó el Informe Final del trabajo de graduación titulado “MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS”, presentado por el señor Wilson Daniel Vasco Gómez de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Oswaldo Paredes

PRESIDENTE DEL TRIBUNAL

Ing. Mario García

DOCENTE CALIFICADOR

Ing. Franklin Silva

DOCENTE CALIFICADOR

DEDICATORIA

Dedicado a mis padres y hermanas por el apoyo que me han brindado.

Daniel Vasco

AGRADECIMIENTO

A Dios y a mi familia que han sido el pilar fundamental para mi formación personal y académica.

Daniel Vasco

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS	VI
ÍNDICE DE GRÁFICOS Y FIGURAS	XI
ÍNDICE DE TABLAS	XIII
GLOSARIO DE TÉRMINOS.	XIV
RESUMEN EJECUTIVO	XVI
INTRODUCCIÓN	XVII
CAPÍTULO I	1
1 EL PROBLEMA:	1
1.1. Tema:	1
1.2. Planteamiento del problema.	1
1.2.1. Contextualización.	1
1.2.2. Árbol del problema.	3
1.2.3. Análisis crítico.	3
1.2.4. Prognosis.	3
1.2.5. Formulación del problema.	4
1.3. Preguntas directrices.	4
1.4. Delimitación.	4
1.5. Justificación.	4
1.6. Objetivos.	5
1.6.1. Objetivo general.	5
1.6.2. Objetivos Específicos.	5
CAPÍTULO II	6
2 MARCO TEÓRICO.	6
2.1. Antecedentes Investigativos:	6

2.2.	Fundamentación Legal: _____	6
2.3.	Categorías Fundamentales: _____	7
2.3.1.	COMPRESIÓN DE DATOS. _____	7
2.3.1.1.	Algoritmos de compresión. _____	8
2.3.1.2.	Tipos de algoritmos. _____	9
2.3.1.2.1.	Algoritmos estadísticos. _____	9
2.3.1.2.2.	Algoritmos basados en diccionario. _____	12
2.3.1.3.	TRANSFORMADAS. _____	14
2.3.1.3.1.	Aplicación de la transformada en el procesado de una imagen.	15
2.3.1.3.2.	Codificación por Transformada. _____	16
2.3.1.3.3.	Transformada Wavelet. _____	17
2.3.1.3.4.	Wavelet madre. _____	19
2.3.1.3.5.	Historia de las Wavelet. _____	20
2.3.1.3.6.	Tipos de Wavelets. _____	22
2.3.1.4.	JPEG. _____	27
2.3.1.4.1.	Codificación de una imagen JPEG. _____	28
2.3.1.4.2.	Formatos. _____	28
2.3.2.	TRANSMISIÓN DE DATOS. _____	28
2.3.2.1.	Objetivos de la transmisión de datos. _____	28
2.3.2.2.	Representación de los datos. _____	29
2.3.2.3.	Comunicaciones locales y remotas. _____	29
2.3.2.4.	Medio de transmisión de datos. _____	30
2.3.2.5.	Formas de transmisión de datos entre dispositivos electrónicos. ___	30
2.3.2.6.	Codificación de señales de transmisión. _____	31
2.3.2.7.	Protocolos de comunicación. _____	31
2.3.2.8.	Modos de Transmisión. _____	31

2.3.3.	MATLAB.	32
2.3.3.1.	Formatos de imágenes soportados por MATLAB	32
2.3.3.2.	MATLAB GUIDE (Matlab Graphical User Interfaz Development Environment).	32
2.4.	Hipótesis.	34
CAPÍTULO III		35
3	MARCO METODOLÓGICO:	35
3.1.	Enfoque.	35
3.2.	Modalidad básica de investigación.	35
3.3.	Tipo o nivel de investigación.	36
3.4.	Población y muestra.	36
3.5.	Métodos y técnicas de la investigación.	36
3.6.	Operacionalización de las variables.	37
CAPÍTULO IV		40
4	ANÁLISIS DE RESULTADOS.	40
4.1.	Interpretación y análisis.	45
CAPÍTULO V		46
5	CONCLUSIONES Y RECOMENDACIONES.	46
5.1.	Conclusiones.	46
5.2.	Recomendaciones.	46
CAPÍTULO VI		47
6	PROPUESTA.	47
6.1.	Datos informativos.	47
6.2.	Antecedentes de la propuesta.	48
6.3.	Justificación.	49
6.4.	Objetivos.	49

6.4.1.	Objetivo general.	49
6.4.2.	Objetivos específicos.	49
6.5.	Análisis de factibilidad.	50
6.6.	Fundamentación.	50
6.6.1.	Transformada Discreta Wavelet (DWT) y Multirresolución.	50
6.6.2.	Eliminación de la redundancia.	51
6.6.3.	Histograma de una imagen.	52
6.6.4.	Conservación y compactación de la energía y de la entropía.	52
6.6.5.	Propiedades de los datos de la imagen.	54
6.6.5.1.	Entropía.	54
6.6.5.2.	Umbral de una imagen jpeg.	55
6.6.6.	Medidas de calidad.	56
6.6.6.1.	Medidas de Calidad Subjetivas.	56
6.6.6.2.	MOS TEST.	56
6.6.6.3.	Guía para evaluar la calidad de la imagen.	57
6.6.6.4.	Medidas de Calidad Objetivas.	57
6.7.	Metodología.	59
6.7.1.	Proceso de compresión de imágenes de extensión .jpeg mediante la transformada Wavelet.	59
6.7.1.1.	Descomposición de la frecuencia.	60
6.7.1.2.	Reconstrucción de la señal.	64
6.7.2.	Modelo de compresión de imágenes de extensión .jpeg realizado en guide de matlab.	65
6.7.2.1.	Interfaz realizada en guidede matlab.	67
6.7.2.2.	Histograma de una imagen en Matlab.	70
6.7.2.3.	Cálculo de la Entropía.	71

6.7.2.4. Cálculo del umbral.	72
6.7.2.5. Error Cuadrático Medio (MSE) y Relación Señal Pico a Ruido (PSNR).	72
6.7.3. Manual de usuario.	74
6.7.3.1. Requisitos del sistema.	74
6.7.3.2. Entorno de desarrollo.	75
6.7.3.3. Pasos para la ejecución de la aplicación:	75
6.7.4. Análisis e interpretación de resultados.	80
6.7.4.1. Resultados de la prueba objetiva.	80
6.7.4.2. Resultados de las pruebas subjetivas.	83
6.7.4.2.1. Análisis del proceso de compresión – descompresión de imágenes de extensión .jpeg	83
6.7.4.2.2. Análisis del proceso de transmisión de imágenes de extensión .jpeg.	84
CAPÍTULO VII	89
7 CONCLUSIONES Y RECOMENDACIONES.	89
7.1. Conclusiones.	89
7.2. Recomendaciones.	89
Bibliografía	90
ANEXOS	92

ÍNDICE DE GRÁFICOS Y FIGURAS

Figura 1.1	Árbol del problema	3
Figura 2.1	Categorías fundamentales (a) variable independiente (b) variable dependiente.	7
Figura 2.2	Tratamiento en el dominio transformado	15
Figura 2.3	Sistema típico de codificación por transformación: (a) codificador (b) decodificador.	16
Figura 2.4	Wavelet de Haar	22
Figura 2.5	Las Wavelets de Daubechies (dbN). Donde N es el orden y $N \in \mathbb{R}^+$	23
Figura 2.6	Symmlets (symN). Donde N es el orden.	24
Figura 2.7	Coiflets (coifN). Donde N indica el orden.	24
Figura 2.8	Wavelets Biortogonales (biorNr.Nd).	25
Figura 2.9	Wavelet Mexican Hat.	26
Figura 2.10	Wavelet Morlet	27
Figura 2.11	Matlab GUIDE.	33
Graf.4.1	Resultados - Pregunta 1	40
Graf. 4.2	Resultados - Pregunta 2	41
Graf. 4.3	Resultados - Pregunta 3	42
Graf. 4.4	Resultados - Pregunta 4	43
Graf. 4.5	Resultados - Pregunta 5	44
Figura 6.1	Codificación de los pixeles	54
Figura 6.2	Flujograma del programa.	59
Figura 6.3	Filtro pasa alto.	60
Figura 6.4	Filtro pasa bajos.	61
Figura 6.5	Estructura de filtros.	61
Figura 6.6	Estructura de las subbandas.	62
Figura 6.7	Descomposición de una imagen en tres niveles.	63
Figura. 6.8	Inserción de ceros a la componente de la señal.	64
Figura. 6.9	Tratamiento de imágenes en Matlab.	65
Figura 6.10	Creación de subimágenes.	66
Figura 6.11	Niveles de descomposición.	66

Figura 6.12 Interfaz gráfica del compresor.	67
Figura 6.13 Primera etapa de filtros (a) filtro pasa bajo, (b) filtro pasa alto.....	68
Figura 6.14 Primera etapa de filtros (a) filtro pasa bajo luego del diezmado, (b) filtro pasa alto luego del diezmado.	68
Figura 6.15. Componentes Wavelet (a) Aproximación (b) Componente Horizontal (c) Componente Vertical (d) Componente Diagonal.	69
Figura 6.16 Histogramas de las imágenes (a) Imagen original en escala de grises (b) Componentes Wavelet.....	71
Figura 6.17 Entorno de Matlab.	75
Figura 6.18 Selección del directorio de trabajo.	76
Figura 6.19 Ejecución de la interfaz gráfica del compresor.....	76
Figura 6.20 Interfaz gráfica ejecutada.....	77
Figura 6.21 Selección de la imagen que se va a abrir.	77
Figura 6.22 Imagen original cargada.	78
Figura 6.23 Proceso de compresión.	78
Figura 6.24 Proceso de descompresión.....	79
Figura 6.25 Guardar imagen (a) Imagen comprimida (b) Imagen descomprimida.	79
Figura 6.26 Salir de la aplicación.....	80
Figura 6.27 Menús de opciones.	80
Graf. 6.1 Resultados MOS TEST.....	82
Figura 6.28 HFS (HTTP File Server) versión 2.2f	85
Figura 6.29 Página del servidor creada con HFS.....	85

ÍNDICE DE TABLAS

Tabla 3.1 Operacionalización - Compresión de imágenes.....	37
Tabla 3.3 Operacionalización – Transmisión de datos	39
Tabla 4.1 Resultados - Pregunta 1	40
Tabla 4.2 Resultados - Pregunta 2	41
Tabla 4.3 Resultados - Pregunta 3	42
Tabla 4.4 Resultados - Pregunta 4	43
Tabla 4.5 Resultados - Pregunta 1	44
Tabla 6.1 Encuesta Mos Test	57
Tabla 6.2 Datos del proceso de compresión – descompresión.....	74
Tabla 6.3 Encuesta.	81
Tabla 6.4 Resultados encuesta MOS TEST.	82
Tabla 6.5 Resultados – proceso de compresión/descompresión	83
Tabla 6.6 Resultados – proceso de compresión/descompresión	86

GLOSARIO DE TÉRMINOS.

7-zip archivador de ficheros libre para Microsoft Windows.

Algoritmo Greedy adaptativo algoritmo de compresión que escoge el mejor código candidato para formar parte de una solución.

Ancho de banda cantidad de información que se puede enviar a través de una conexión de red en un período dado medida en bits por segundo.

ARJ Archived by Robert Jung (aplicación para crear archivos comprimidos).

BMP bitmap (imagen de mapa de bits).

Buffer espacio de memoria para evitar que un programa o recurso se quede sin datos durante una transferencia.

Convolución operador matemático que transforma dos funciones en una tercera función.

Correlación Relación recíproca o mutua entre dos o más cosas.

GIF Graphics Interchange Format.

HDF Hierarchical Data Format.

MSE Mean Squared Error (error cuadrático medio).

PCX formato de imagen digital Picture Exchange.

Pixel Picture Element (elemento de imagen).

PKZIP método de compresión de archivos desarrollado por Phillip Katz.

PSNR Peak Signal Noise Ratio (Relación señal pico a ruido).

RAR Roshal ARchive (formato de archivo de compresión de datos desarrollado por Eugene Roshal).

Resolución de una imagen cantidad de píxeles que componen una imagen.

RGB modelo de color Red Green Blue (Rojo Verde Azul).

TIFF Tagged Image File Format.

XWB XACT's Wave Bank (formato de archivos).

ZIP formato de almacenamiento creado originalmente por Phillip Katz.

RESUMEN EJECUTIVO

El presente trabajo investigativo que se detalla a continuación, corresponde al estudio sobre la compresión de imágenes de extensión .jpeg y como este proceso reduce los recursos esenciales para su almacenaje y mejoramiento de su transmisión.

Este informe cubre ciertos aspectos del análisis compresión de datos, profundizando acerca la Transformada Discreta Wavelet (DWT) y cómo ésta se puede utilizar para la compresión de imágenes de extensión .jpeg.

Su utiliza la herramienta Guide de Matlab para la elaboración de una aplicación que cumple el proceso de compresión – descompresión de las imágenes de extensión .jpeg, a más de incluir en este proyecto un manual de usuario para su fácil manipulación, permitiendo la visualización de las imágenes obtenidas durante el proceso. En la aplicación también se visualizan los datos obtenidos que sirven para el análisis de compresión de las imágenes.

Con el uso del software libre llamado HFS (HTTP File Server) versión 2.2f, se crea un servidor gratuito que permite subir y descargar archivos y es muy útil para analizar la transmisión de las imágenes originales con sus respectivas imágenes comprimidas y descomprimidas.

Por último, se realizan las conclusiones y recomendaciones luego de finalizar el desarrollo del proyecto y haber realizado pruebas para la comprobación del funcionamiento de la aplicación realizada.

INTRODUCCIÓN

Para comprimir las imágenes, se debe tratar de eliminar o disminuir las redundancias en las áreas donde hay un pequeño cambio entre los valores de los píxeles contiguos, esto usualmente sucede en áreas donde las imágenes tienen colores similares; pero con cambios grandes de colores las imágenes serán menos redundantes y por lo tanto serán más difíciles de comprimir.

El estándar JPEG ha sido pensado para dar mejor calidad a las imágenes, pero como consecuencia de las mejoras realizadas, se produjo un aumento bastante significativo en la complejidad de sus algoritmos y por consiguiente se convierte en una enorme exigencia para el hardware de los dispositivos de almacenamiento actuales.

El propósito de la presente tesis es diseñar e implementar un modelo de compresión – descompresión de imágenes de extensión .jpeg para la mejora de su transmisión, que servirá como material de consulta para los estudiantes de la carrera de Electrónica y Comunicaciones de la F.I.S.E.I. - U.T.A.

Este documento está dividido en seis capítulos, detallados de la siguiente manera:

Capítulo I: Se describe el problema de investigación indicando las causas y consecuencias, la justificación, y los objetivos a alcanzar en el presente trabajo.

Capítulo II: Corresponde al marco teórico, se explica los fundamentos sobre la compresión de datos, los tipos de algoritmos de compresión, Transformada Wavelet, el estándar JPEG. Conceptos de transmisión de datos y una breve explicación del software utilizado para la creación de una interfaz gráfica de usuario. En este capítulo se plantea la hipótesis a ser comprobada durante el desarrollo del presente trabajo y las variables de la misma.

Capítulo III: Se explica el marco metodológico, incluyendo el enfoque, la modalidad y el tipo de investigación; así como, la recolección y procesamiento de datos.

Capítulo IV: Se presentan y analizan los resultados de la encuesta realizada a los docentes de la carrera de Electrónica y Comunicaciones de la F.I.S.E.I. – U.T.A con el fin de evaluar la factibilidad de la realización de un compresor de imágenes de extensión .jpeg.

Capítulo V: Se presentan las conclusiones y las recomendaciones luego del análisis de los resultados obtenidos de la encuesta.

Capítulo VI: Consiste en la descripción del desarrollo de la propuesta; se describe el proceso de compresión – descompresión mediante el uso de la Transformada Discreta Wavelet (DWT) y su implementación mediante la interfaz gráfica de usuario desarrollada en Guide de Matlab. Se realiza además el análisis e interpretación de los resultados obtenidos en los procesos de compresión – descompresión y transmisión de las imágenes de extensión .jpeg. Se presentan y analizan los resultados de la encuesta MOS TEST realizada a los docentes de la carrera de Electrónica y Comunicaciones sobre la calidad de las imágenes luego del proceso de compresión – descompresión.

Capítulo VII: Se presentan las conclusiones y las recomendaciones del proyecto tras el análisis de los resultados obtenidos al finalizar el proyecto.

Para finalizar en el presente trabajo se presentan la bibliografía y los anexos.

CAPÍTULO I

1 EL PROBLEMA:

1.1. Tema:

“MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS”

1.2. Planteamiento del problema.

1.2.1. Contextualización.

Todo usuario al momento de transmitir o “pasar” datos (información), especialmente cuando se trata de enviar correos electrónicos, desea enviar la mayor cantidad de archivos posibles, siendo un inconveniente el tamaño del mismo, lo que influye en la velocidad de transmisión, para lo cual se requiere optimizar esta tarea usando varios programas o software, entre ellos: Winrar, Winzip, 7-Zip.

Una de las tareas más usuales realizadas es el envío de imágenes o fotografías vía correo electrónico, comprimiendo estos archivos para poder enviarlos en un solo archivo contenedor, reduciendo el tamaño de cada imagen para luego recuperarla al extraerla. Estos procesos se denominan: compresión y descompresión de datos.

"La compresión de imágenes aborda el problema de reducir la cantidad de datos para representar una imagen digital. La compresión se logra removiendo una o más de las tres posibles redundancias básicas. Redundancia de código: cuando se usan códigos de palabra menos óptimos (de menor longitud). Redundancia interpixelaria: Que resulta de la correlación entre pixeles de una imagen. Redundancia psicovisual: La cual se da debido a los datos que el sistema de visión humano ignora" ¹

Para comprimir una imagen se reduce el número de bits. Hay dos tipos de compresión de datos: con pérdidas, y sin pérdidas. La compresión con pérdidas que produce diferencias imperceptibles se llama también visualmente sin pérdidas.

La transmisión de imágenes a gran escala provoca molestias, en especial cuando se realiza esta tarea en repetidas ocasiones, tal es el caso de fotógrafos por citar un ejemplo, y una de las extensiones de imágenes que más problemas causa es la .jpeg que es el formato con el que trabajan muchos tipos de cámaras fotográficas, que tiene una muy buena calidad pero son de mayor tamaño que otros tipos de imágenes.

¹**Digital Image Processing 2nd Edition (DIP/2e)** by Gonzalez and Woods © 2002. Prentice Hall

1.2.2. Árbol del problema.

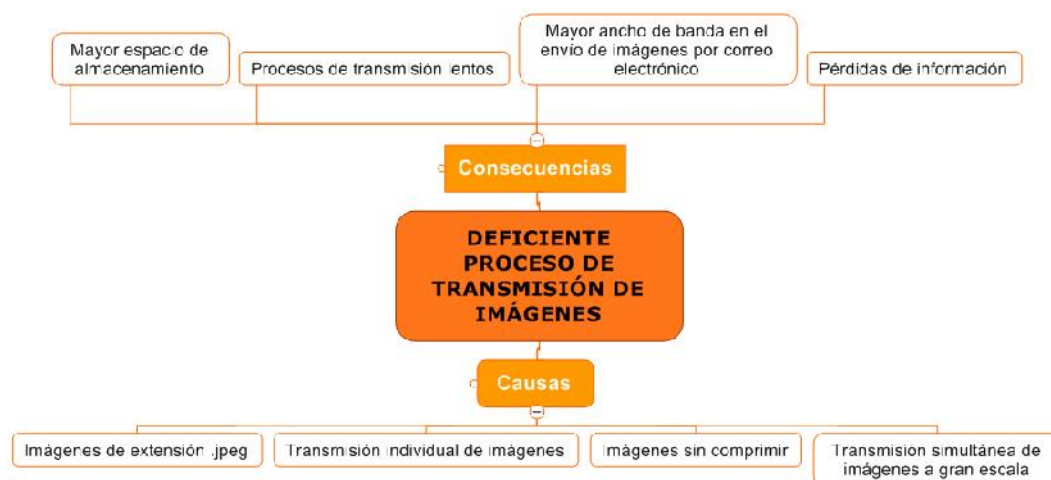


Figura 1.1 Árbol del problema

Elaborado por: Investigador

1.2.3. Análisis crítico.

La transmisión o envío de imágenes sin comprimir es un proceso deficiente, y requiere de mayor cantidad de memoria donde van a ser almacenadas (discos duros, pendrives, discos extraíbles, entre otros), así como, el envío de este tipo de archivos por correo electrónico se convierte en un proceso lento e inadecuado, donde cada imagen debe ser transmitida de manera individual, lo que requiere de un mayor ancho de banda y la velocidad de transmisión disminuye. Si el proceso se realiza de manera individual y de manera simultánea, se pueden ocasionar pérdidas de información.

1.2.4. Prognosis.

De continuar con los procesos inadecuados de transmisión de imágenes, se continuará desperdiciando recursos, como son: el tiempo, espacio en disco, y ancho de banda en envíos de correos electrónicos.

1.2.5. Formulación del problema.

¿Es posible diseñar un modelo de compresión de imágenes de extensión .jpeg para el mejoramiento de la transmisión de datos?

1.3. Preguntas directrices.

¿Es posible implementar un modelo de compresión de imágenes de extensión .jpeg?

¿Qué técnica será la adecuada para implementar un modelo de compresión de imágenes de extensión .jpeg?

¿Es posible realizar un modelo de compresión de imágenes de extensión .jpeg en Matlab?

1.4. Delimitación.

Campo: Ingeniería en Electrónica y Comunicaciones.

Área: Comunicaciones.

Aspecto: Compresión de imágenes.

1.5. Justificación.

Con el interés de analizar el proceso de compresión – descompresión de datos, nace la idea de implementar este novedoso proyecto que permitirá a los usuarios hacer un uso eficiente de sus imágenes al momento de comprimirlas para ser transmitidas o almacenadas, utilizando una interfaz gráfica fácil de usar que se desarrollará en Matlab, permitiendo ahorrar tiempo en la transmisión de imágenes.

Este proyecto además servirá como material de consulta para los alumnos de ingeniería en Electrónica y Comunicaciones, siendo una guía de estudio que fortalezca los conocimientos adquiridos durante el tiempo de estudio, el mismo que podría ser mejorado y servir como fuente para nuevas investigaciones.

1.6. Objetivos.

1.6.1. Objetivo general.

- Implementar un modelo de compresión de imágenes de extensión .jpeg mediante Matlab para el mejoramiento de la transmisión de datos.

1.6.2. Objetivos Específicos.

- Analizar el proceso de compresión de imágenes de extensión .jpeg.
- Estudiar la transmisión de datos para la compresión de imágenes de extensión .jpeg mediante Matlab.
- Proponer un modelo de compresión de imágenes de extensión .jpeg mediante Matlab que permita mejorar la transmisión de datos.

CAPÍTULO II

2 MARCO TEÓRICO.

2.1. Antecedentes Investigativos:

Conforme a lo indagado en la biblioteca de la F.I.S.E.I. de la Universidad Técnica de Ambato, se encontró dos tesis de maestría con los siguientes temas: "COMPRESIÓN DE IMÁGENES EN MOVIMIENTO PARA TRANSMISIÓN EN REDES IP EN BASE A LA APLICACIÓN DE LA TRANSFORMADA DISCRETA COSENO CON DIFERENTES RELACIONES DE COMPRESIÓN", desarrollada por el Ing. José Gioberty Bucheli Andrade y, COMPRESIÓN DE AUDIO EN BASE A LA APLICACIÓN DE TÉCNICAS DE CUANTIFICACIÓN NO LINEAL PARA DETERMINAR EL EFECTO SONORO DE SU REPRODUCCIÓN EN LOS SERES HUMANOS, desarrollada por el Ing. César Alfredo Naranjo Hidalgo, las que servirán como una guía y material de consulta en la presente Tesis.

2.2. Fundamentación Legal:

La transmisión de datos en el Ecuador está regida bajo la LEY ESPECIAL DE TELECOMUNICACIONES Y SU REFORMA.

Disposiciones Fundamentales

“Art. 1.- AMBITO DE LA LEY.- La Ley Especial de Telecomunicaciones tiene por objeto normar en el territorio nacional la instalación, operación, utilización y

desarrollo de toda transmisión, emisión o recepción de signos, señales, imágenes, sonidos e información de cualquier naturaleza por hilo, radioelectricidad, medios ópticos y otros sistemas electromagnéticos.”²

2.3. Categorías Fundamentales:

Variable Independiente: Compresión de imágenes.

Variable dependiente: Transmisión de datos.

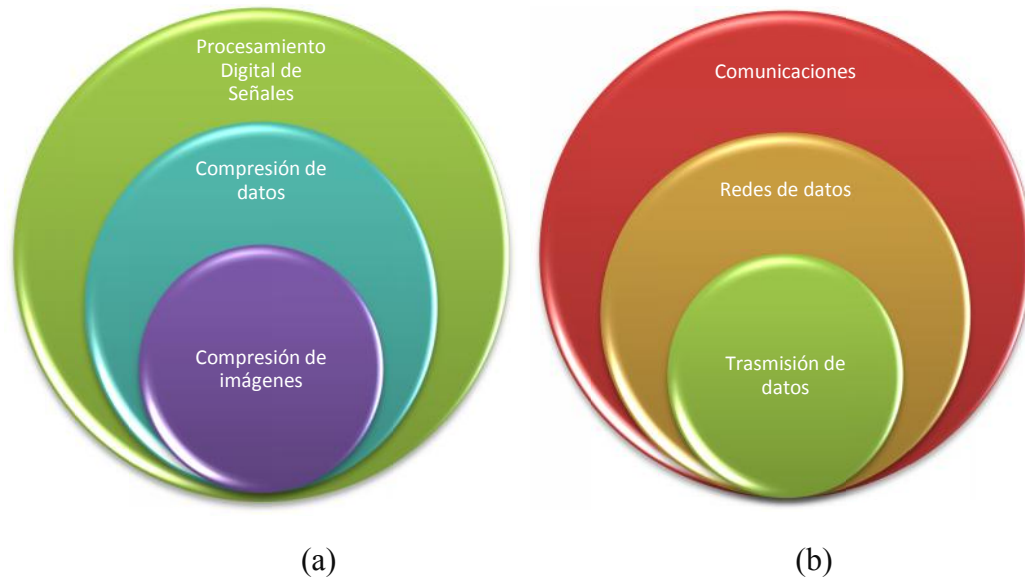


Figura 2.1 Categorías fundamentales (a) variable independiente (b) variable dependiente.

Elaborado por: Investigador

2.3.1. COMPRESIÓN DE DATOS.

La compresión de datos consiste en reducir el volumen de datos tratables para representar una determinada información.

² http://www.lexadin.nl/wlg/legis/nofr/oeur/arch/ecu/Ley_Teleco_reforma.pdf

Un **compresor** utiliza un algoritmo que optimiza los datos, hay que tener en cuenta el tipo de dato que se va a comprimir para usar la técnica adecuada de compresión (no se comprime una imagen del mismo modo que un archivo de audio).

Para reconstruir los datos se realiza un proceso llamado **descompresión**, por medio de un algoritmo opuesto al que se utiliza para la compresión.

2.3.1.1. Algoritmos de compresión.

La compresión de datos se basa fundamentalmente en la búsqueda de repeticiones en series de datos para luego almacenar solo el dato junto al número de veces que se repite. Por ejemplo, si en un fichero aparece una secuencia como "AAAAAA", ocupando 6 bytes se podría almacenar simplemente "6A" que ocupa solo 2 bytes, en algoritmo RLE.

A la hora de hablar de compresión hay que tener presentes dos conceptos:

Redundancia: Datos que son repetitivos o previsibles.

Entropía: La información nueva o esencial que se define como la diferencia entre la cantidad total de datos de un mensaje y su redundancia.

La información que transmiten los datos puede ser de tres tipos:

Redundante: Información repetitiva o predecible.

Irrelevante: Información que no podemos apreciar y cuya eliminación por tanto no afecta al contenido del mensaje.

Básica: Relevante, que no es ni redundante ni irrelevante. La que debe ser transmitida para que se pueda reconstruir la señal.

Teniendo en cuenta estos tres tipos de información, se establecen tres **tipologías de compresión de la información**:

Sin pérdidas reales: es decir, transmitiendo toda la entropía del mensaje (toda la información básica e irrelevante, pero eliminando la redundante). Se utiliza principalmente en la compresión de texto.

Subjetivamente sin pérdidas: es decir, además de eliminar la información redundante se elimina también la irrelevante.

Subjetivamente con pérdidas: se elimina cierta cantidad de información básica, por lo que el mensaje se reconstruirá con errores perceptibles pero tolerables. Se utiliza principalmente en la compresión de imágenes, videos y sonidos.

2.3.1.2. Tipos de algoritmos.

2.3.1.2.1. Algoritmos estadísticos.

Utilizan las propiedades estadísticas de la fuente para mejorar la codificación. Se trata de aprovechar la redundancia de información de la fuente para conseguir la compresión.

Algoritmo Huffman.

El algoritmo Huffman se basa en crear un árbol binario completo, que representa la codificación de los mensajes de la fuente, en el que cada nodo intermedio es menor que sus hijos (y la raíz el menor de todos). Los nodos hoja contienen cada uno de los mensajes emitidos por la fuente. El código para cada mensaje se construye siguiendo el camino desde el nodo raíz hasta el nodo hoja que representa el mensaje. Además, si el decodificador implementa el mismo árbol usado para comprimir, la decodificación no será más que leer bits e ir siguiendo el

camino desde la raíz del árbol hasta las hojas en función del valor de esos bits. Al llegar a la hoja se habrá llegado al mensaje.

La codificación es inversamente proporcional a la probabilidad de aparición del mensaje. Al mensaje más redundante, se le da una codificación más corta asignándole menos símbolos del alfabeto de salida y ahorrando espacio.

En cada paso se recogen los dos nodos con menor probabilidad del árbol y se crea un nodo padre para ambos que contendrá la probabilidad sumada de los dos. Los nodos con menor probabilidad se irán quedando al fondo.

Generalmente los descompresores de este tipo no tienen posibilidad de conocer previamente las probabilidades de los mensajes, pues sólo reciben los códigos asignados a los mensajes; en consecuencia el árbol ya procesado va a ser pasado al descompresor junto con los datos. Esto representa una carga adicional al fichero comprimido que resta en parte la eficiencia de esta técnica. Por ello una de las soluciones es hacer que estos algoritmos sean "adaptativos": se construye el árbol dinámicamente tanto por el compresor como por el descompresor, y así no estar obligados a pasarle el árbol al descompresor. Dependiendo del compresor, se suelen utilizar diferentes implementaciones "adaptativas" para los compresores.

Algoritmo Shannon-Fano.

El algoritmo Shannon Fano construye un árbol como Huffman, pero en vez de partir desde las hojas a la raíz, parte desde la raíz hacia las hojas.

Inicialmente considera a todos los mensajes en un solo conjunto. En cada etapa se intenta dividir el conjunto en dos subconjuntos con la probabilidad más parecida posible entre ambos (para conseguir conjuntos con menos elementos a mayor probabilidad). Se asigna un bit con valor 0 ó 1 a cada uno de estos dos nuevos subconjuntos y se va repitiendo el proceso de manera recursiva para cada nuevo

conjunto creado hasta llegar a los conjuntos con un solo elemento que representarán un mensaje.

Algoritmos Aritméticos.

Estos algoritmos se basan también en las probabilidades de repetición de los mensajes a la entrada, aunque su metodología es muy distinta. Lo que hacen es representar un valor del intervalo $[0,1]$ con mayor número de decimales mayor precisión cuanto mayor sea la información de los datos a comprimir. Básicamente va dividiendo el intervalo $[0,1]$ sucesivamente hasta obtener un número dentro de ese intervalo que utilice menos bits para representar toda la entrada. Así el descompresor podrá reconstruir la entrada con ese número más la información del número de elementos codificados y sus probabilidades correspondientes. Es un proceso bastante eficiente, aunque las probabilidades deben ser incluidas en el fichero comprimido. Por ello se suelen utilizar también modelos adaptativos, que cambian dinámicamente con la entrada.

Algoritmos Predictivos.

Estos algoritmos intentan predecir el siguiente mensaje de la entrada en base a lo que se sabe de la entrada procesada en ese momento. Si el mensaje de la entrada coincide con el predicho, la codificación se podrá hacer con menos bits. En caso contrario se hará con más bits (que permitan la sincronización con el descompresor a la hora de mantener las mismas tablas internas; así ya no necesitar incluirlas en el fichero comprimido). Estos algoritmos son adaptativos.

En consecuencia suelen ser más rápidos que las técnicas anteriores, tanto para compresión como descompresión, y además no son difíciles de programar (pero sí su mejora). Son muy malos cuando hay mucha redundancia, así que en la práctica se suelen usar en conjunción con otras técnicas de compresión para tratar los casos en que hay excesiva redundancia (por ejemplo el Run-Length).

2.3.1.2.2. Algoritmos basados en diccionario.

Algoritmo Run Length.

El algoritmo Run Length es también denominado RLE (Run Length Encoding). Es el más simple y a la vez el más ineficiente. Utiliza un diccionario deslizante para predecir el siguiente carácter de la entrada. Realmente se le considera ya algo "primitivo". Además hay diferentes formas de implementarlo, todas ellas patentadas.

Busca repeticiones consecutivas de un mismo símbolo y almacena en un byte el número de esas repeticiones consecutivas y en el segundo byte escribe el símbolo. Ejemplo: **17 48** (el byte 48 se repite 17 veces).

Demuestra gran eficiencia cuando hay un alto número de repeticiones consecutivas de un determinado byte. La unidad básica serían dos bytes, el primero indica el número de veces que se repite el segundo. Básicamente se utiliza para crear archivos tipo BMP sin gradaciones de color.

Algoritmo LZW.

El algoritmo LZW pertenece a la familia de los LZ78 (Liv&Zempel), es un algoritmo Greedy Adaptativo. Es el que implementan la mayoría de los programas de compresión.

En la codificación, al principio el compresor parte de un diccionario en el que se han introducido todas las cadenas de longitud 1, es decir, 256 cadenas que constan de un solo carácter (los caracteres ASCII). Añadir una cadena al diccionario significa asignarle un identificador secuencial consecutivo a partir de la anterior cadena. Como al principio las primeras 256 posiciones están ocupadas el siguiente número a asignar es el 257. Este número ya no cabe en 8 bits, por lo que las salidas (los identificadores de las cadenas ya reconocidas) del algoritmo son al

principio de 9 bits e irán aumentando conforme se vaya quedando pequeño el diccionario.

Durante el proceso, se van añadiendo nuevas cadenas de longitud mayor. Al identificarse cadenas más largas con un número de varios bits, la compresión resulta bastante efectiva.

La descompresión es más sencilla, aunque tiene que construir el mismo árbol. El descompresor recibe inicialmente códigos de 9 bits que usará como índices en el diccionario. Mantendrá el diccionario actualizado a partir de esa información.

Se ha visto que el diccionario iba creciendo a medida que se iban insertando nuevas cadenas en él. Sin embargo la memoria es finita y hay que poner tope a esa inserción. Además, a medida que el diccionario vaya creciendo, se utilizarán más bits para identificar a cada cadena, con lo que para las cadenas cortas (que son también las más probables) se conseguirá una codificación pobre e ineficiente, puesto que ahorrará pocos bits. Una vez puesto un tope (de 12 a 14 bits, es decir, entre 4096 y 16384 cadenas), está claro que el diccionario se llenará y eliminará las cadenas que guarda. Algunos modelos desechan todas las cadenas del diccionario y lo "vacían completamente", otros aprovechan las cadenas que más se han usado y la dejan en el diccionario; no hay una condición única que asegure que una u otra opción será mejor.

Algoritmo LZ77.

Este algoritmo es una técnica presente en multitud de modelos de compresión pues tiene tantas decisiones de diseño que cada uno es implementado de manera distinta a los demás. Esto hace que no se puedan patentar o que se puedan saltar de una forma u otra las patentes actuales. Como ejemplo de compresores que lo usan sería: ARJ, PKZIP, RAR, etc. Es más utilizado que el LZ78 (posiblemente por tener menos problemas de propiedad).

El LZ77 mantiene un registro de los últimos caracteres procesados de la entrada, pero no construye un diccionario explícito. En cada momento, el algoritmo se encuentra procesando en un punto de la entrada los "n" caracteres anteriores que forman la historia del algoritmo ó ventana (lo que equivale al diccionario). Los caracteres posteriores al punto actual forman el llamado buffer de adelantamiento. En cada paso, la cadena que comienza en el punto actual de la entrada se busca hacia atrás en la historia. Si se encuentra una coincidencia que sea lo suficientemente larga como para tenerla en cuenta, a la salida se sustituye la cadena coincidente por un par que indica el desplazamiento hacia atrás y la longitud de la coincidencia con la cadena hacia atrás. Como los pares *desplazamiento, longitud* ocupan menos espacio que la cadena que coincidió, se obtiene compresión. Si no se encuentra una coincidencia, la salida es una copia literal de la entrada. Posteriormente se avanza la ventana (es decir, se avanza en la entrada) ya sea de longitud si hubo coincidencia, o bien un carácter si no la hubo. El hecho de ir desplazando la ventana sobre la entrada hace que a estos algoritmos se les llame también de "ventana deslizante". Cuanto más grande sea la ventana, mayor será la compresión (más grande es la historia sobre la que se buscan posibles coincidencias y mayor será la probabilidad de encontrar una coincidencia más larga), aunque para una ventana grande se necesitarán más bits para codificar los valores del desplazamiento. Por ello el tamaño de la ventana más apropiado debería ser uno que haga que el campo de desplazamiento entre en pocos bits.

El descompresor es muy sencillo, ya que su ventana está formada por los datos que ha descomprimido anteriormente, y cuando recibe un par (desplazamiento, longitud), sólo tiene que copiar a la salida la longitud de bytes que están en el desplazamiento de los bytes antes. En el caso de recibir un literal, este es copiado tal cual a la salida.

2.3.1.3.TRANSFORMADAS.

De forma general, una transformada representa un cambio hacia un dominio diferente, que debido a sus propiedades, facilita de alguna manera la realización

de tareas determinadas. Este cambio puede caracterizarse mediante un operador, que genéricamente se denota como $T(\cdot)$. Al aplicar este operador de una señal f , se puede representar de la siguiente manera:

$$f(t) \xrightarrow{T(\cdot)} T(f)$$

$$T(f) \xrightarrow{T(\cdot)^{-1}} f(t)$$

Donde $T(\cdot)^{-1}$ es la transformada inversa.

2.3.1.3.1. Aplicación de la transformada en el procesamiento de una imagen.

El procesamiento de una imagen puede llevarse a cabo en el dominio espacial (amplitud en función de la posición) o en el dominio transformado. La figura 2.2 muestra la configuración general del procesamiento digital de imagen basado en transformadas.

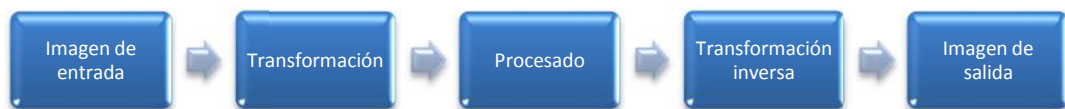


Figura 2.2 Tratamiento en el dominio transformado

Elaborado por: Investigador

La imagen digitalizada es previamente transformada antes de efectuar el procesamiento. Después de que la imagen es procesada, se somete a la correspondiente transformación inversa complementaria, para conseguir el retorno de la imagen al dominio original. Este tipo de análisis permite que las imágenes sean tratadas con métodos totalmente diferentes de aquellos utilizados sobre los datos originales.

El procesamiento, basado en el dominio transformado consigue codificar imágenes reduciendo los requerimientos de almacenamiento y ancho de banda para la

transmisión. Esto se realiza mediante el descarte o cuantificación de los coeficientes transformados de muy baja magnitud.

Un sistema típico de codificación basado en transformación es presentado en la figura 2.3. En la figura 2.3 (a), una imagen de $N \times N$ se subdivide en subimágenes de tamaño $n \times n$. Enseguida se aplica la transformación cuyo objetivo es deshacer la correlación de los píxeles de cada subimagen, o empaquetar tanta información como resulte posible en el menor número de coeficientes de la transformada. Se continúa con la cuantificación, que elimina selectivamente o cuantifica con menor precisión, los coeficientes que llevan la menor cantidad de información. Estos coeficientes influyen poco en la subimagen reconstruida. La de codificación finaliza codificando los coeficientes cuantificados. El proceso inverso, para obtener la imagen reconstruida, se muestra en la figura 2.3 (b).

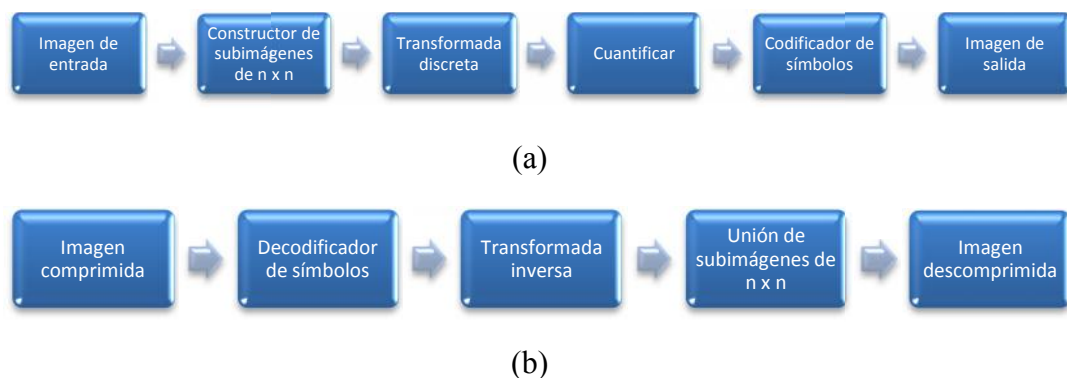


Figura 2.3 Sistema típico de codificación por transformación: (a) codificador (b) decodificador.

Elaborado por: Investigador

2.3.1.3.2. Codificación por Transformada.

La codificación por transformada es la base en la mayoría de los estándares y sistemas de codificación de imágenes. Las componentes espaciales de la imagen son llevadas al dominio de la transformada que se utilice.

Las razones para realizar la transformación de imágenes son:

- Los datos espaciales de la imagen son muy difíciles de comprimir.
- Las muestras de píxeles vecinos están altamente correlacionadas y la energía tiende a estar distribuida en toda la imagen, lo que hace difícil descartar datos o reducir la precisión de los datos sin afectar la calidad de la imagen.

La codificación por transformada ayuda a reducir la correlación de los datos, tratando de dejar un pequeño número de coeficientes visualmente importantes y un gran número de coeficientes insignificantes que puedan ser descartados o eliminados. Al elegir correctamente la transformada, los datos son fáciles de comprimir en el dominio de la transformada. La elección de la transformada depende de ciertos criterios:

- Los datos en el dominio de la transformada deben ser decorrelacionados, es decir, separados en componentes con mínima interdependencia.
- Compactación.- La energía en los datos transformados debe concentrarse en un pequeño número de valores.
- La transformada debe ser reversible.
- La transformada debe ser computacionalmente tratable lo que implica: bajos requerimientos de memoria, que sea realizable utilizando aritmética de precisión limitada, bajo número de operaciones aritméticas, etc.

2.3.1.3.3. Transformada Wavelet.

La Transformada Wavelet constituye una técnica relativamente nueva de procesamiento de señales e imágenes y ha sido propuesta por los investigadores como una poderosa herramienta en el análisis sobre el comportamiento local de una señal basándose en el análisis de multirresolución. Consiste en comparar la señal con ciertas funciones wavelet, las cuales se obtienen a partir de las wavelet

madre. La comparación permite obtener coeficientes que son fáciles de manipular, almacenar, transmitir e interpretar con la posibilidad de invertir la transformada, recuperando la señal a partir de esos coeficientes wavelet calculados. La transformada wavelet es un caso especial de la transformada de Fourier, y así como para la transformada de Fourier se usan unos algoritmos discretos como el FFT (Transformada rápida de Fourier), así mismo en la transformada Wavelet se usa la DWT (Transformada Wavelet Discreta).

El cálculo de la transformada wavelet para todas las posibles escalas supone una gran cantidad de información. Escoger solo aquellas escalas y posiciones que resulten interesantes para ciertos estudios es una tarea difícil. Si se escogen aquellas escalas y posiciones basadas en potencias de dos, los resultados serán más eficaces. Este análisis se denomina DWT.

El análisis de Wavelet se puede utilizar para dividir la información de una imagen en subseñales de aproximación y de detalles de ella misma. En el caso de dos dimensiones la subseñal de aproximación muestra la tendencia general de los valores de los píxeles y las subseñales de detalles muestran los detalles o los cambios verticales, horizontales y diagonales en la imagen.

Si estos detalles son muy pequeños entonces pueden ser fijados a cero sin cambiar perceptiblemente la imagen, mientras mayor es el número de ceros mayor es la compresión que se puede alcanzar. La cantidad de información conservada por la imagen después de la compresión y la descompresión se conocen como la energía retenida. Si la energía conservada es el 100% entonces la compresión es conocida como compresión sin pérdida (lossless), y la imagen puede ser reconstruida exactamente, esto ocurre cuando el valor de umbral se fija a cero, significando que los detalles no se han cambiado. Si algunos valores se cambian se perderá parte de la energía y ésta se conoce como compresión con pérdidas (lossy). Idealmente, durante la compresión el número de ceros y la retención de la energía serán tan altos como sea posible. Sin embargo, como se obtienen más ceros se pierde más energía, se necesita encontrar de un equilibrio entre las dos necesidades.

2.3.1.3.4. Wavelet madre.

El término Wavelet significa onda pequeña pues esta función es de duración finita y comportamiento oscilatorio. El término madre implica que las funciones con distintas regiones de soporte que se usan en el proceso de transformación se derivan de una función principal, lo que viene a ser la Wavelet madre. Es decir, la función Wavelet madre es un prototipo a partir del cual se obtienen las demás funciones ventana.

Esta función Wavelet debe cumplir con ciertos criterios o propiedades:

1) El área total sobre la curva de la función es cero, es decir.

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

Esta propiedad sugiere una función que oscila tanto arriba como abajo del eje del tiempo (t). Esta función tiende a tener una apariencia ondulada.

2) El área total de $(|\psi(t)|^2)$ es finita.

$$\int_{-\infty}^{\infty} (|\psi(t)|)^2 dt < \infty$$

Esta condición implica que la energía de la función es finita, sugiriendo que la función está localizada en algún intervalo finito y es cero o casi cero fuera de este intervalo.

Estas dos propiedades le dan el nombre a estas funciones de wavelet y presentan similitud con las ondas.

3) Condición de admisibilidad.

Un número infinito de funciones satisfacen estas condiciones, y algunas de estas han sido investigadas y son comúnmente usadas por transformadas wavelets.

Transformar los valores de los píxeles, y codificar los valores de la transformada. Aquí cabe resaltar que la compresión es alcanzada por la remoción de redundancia. De esta forma es posible pensar en la transformada en términos de la entropía de una imagen. En una imagen altamente correlacionada, los píxeles tienden a tener valores equiprobables, lo que resulta en la obtención de una máxima entropía. Si la transformada de píxeles son decorrelacionados, ciertos valores de píxeles llegan a ser comunes, por consiguiente teniendo grandes probabilidades, mientras que los otros no tanto. Esto resulta en la obtención de una pequeña entropía. Cuantizando los valores de la transformada se puede producir una compresión de las imágenes con pérdida de eficiencia.

2.3.1.3.5. Historia de las Wavelet.

La primera referencia al concepto de Wavelet se asocia a Jean Baptiste Joseph Fourier por sus trabajos en 1807 sobre la transformada de Fourier por medio de senos y cosenos de distintas frecuencias que permitieron representar casi cualquier señal o función periódica, significando que Fourier abrió la puerta hacia un nuevo mundo.

En 1909 fue la primera vez que se mencionó una Wavelet en el trabajo de tesis del matemático húngaro Alfred Haar. Una propiedad de la Wavelet de Haar es su soporte compacto. La principal desventaja de esta Wavelet resulta en el hecho de no poder ser continuamente diferenciable de manera continua, lo que puede limitar sus aplicaciones.

En 1930 Paul Levy notó en una de sus aplicaciones sobre el movimiento Browniano que la Wavelet de Haar era más apropiada para la descomposición de la señal que el tratamiento a través de senos y cosenos de Fourier.

En 1946 el físico Dennis Gabor descompuso una señal en paquetes de frecuencia-tiempo.

En algunos trabajos realizados entre 1960 y 1980 Guido Weiss y Ronald R. Coifman estudiaron lo que denominaron “átomos”, elementos a partir de los que era posible la reconstrucción de una señal.

En 1981 el ingeniero Jean Morlet encuentra el modo de descomponer una señal sísmica en cierto tipo de “Wavelets” de forma constante.

En 1984 Morlet desarrolla su modelo con la ayuda del físico cuántico Alex Grossman, definiendo las Wavelet en el contexto de la física cuántica. El término Wavelet aparece por primera vez.

En 1985 Yves Meyer descubre la primer Wavelet ortogonal suave.

En 1986 Stéphane Mallat dio un salto importante a través de su trabajo sobre el procesamiento digital de señales y muestra que los métodos de Haar, Gabor, Morlet, están relacionados por el mismo algoritmo de Wavelets. Por otro lado, Meyer construyó la primer Wavelet no trivial la cual era continuamente diferenciable.

En 1987 Ingrid Daubechies se apoyó en el trabajo de Mallat para construir el primer Wavelet ortogonal con soporte compacto, que resulta ser, la más depurada versión de las Wavelets y pilar de las Wavelets actuales y sus aplicaciones.

En 1990 David Donoho y Johnstone usan los wavelets para eliminar el ruido de una señal.

En 1992 el FBI (Federal Bureau of Investigation) elige un método de Wavelets desarrollado por Tom Hopper, de la división de Servicios de información criminal

del FBI, y Jonathan Bradley y Chris Brisan, del Laboratorio Nacional de Los Álamos, para comprimir su enorme base de datos de huellas dactilares.

En 2004, una vez superada la gran revolución de los años 90, se ve que no todo se puede hacer con wavelets, pero que sí suponen una nueva herramienta útil de cálculo y análisis.

2.3.1.3.6. Tipos de Wavelets.

Wavelet de Daubechies.

Dentro de la familia de las Wavelets Daubechies, encontramos la notación 'dbN' donde N indica el orden con $N \in \mathbb{Z}$.

Esta wavelet posee soporte compacto y con N momentos de desvanecimiento.

Puede ser ortogonal, biortogonal y no posee simetría (de hecho en algunas wavelets la asimetría es muy pronunciada).

El número de momentos de desvanecimiento para Ψ es N. El número de filtros es $2N$. Estas Wavelets no tienen una expresión determinada, excepto por la Wavelet de Haar o db1, la cual se muestra en la figura 2.4:

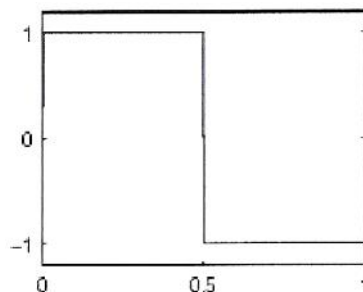


Figura 2.4 Wavelet de Haar

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

La Wavelet de Haar o Daubechies de orden 1, es la primera y la más sencilla de las Wavelets. Y aunque tiene soporte compacto, no tiene buena localización tiempo-frecuencia. Esta Wavelet no es continua, y por ende es no diferenciable.

La apariencia de las Wavelets de Daubechies se muestra en la figura 2.5.

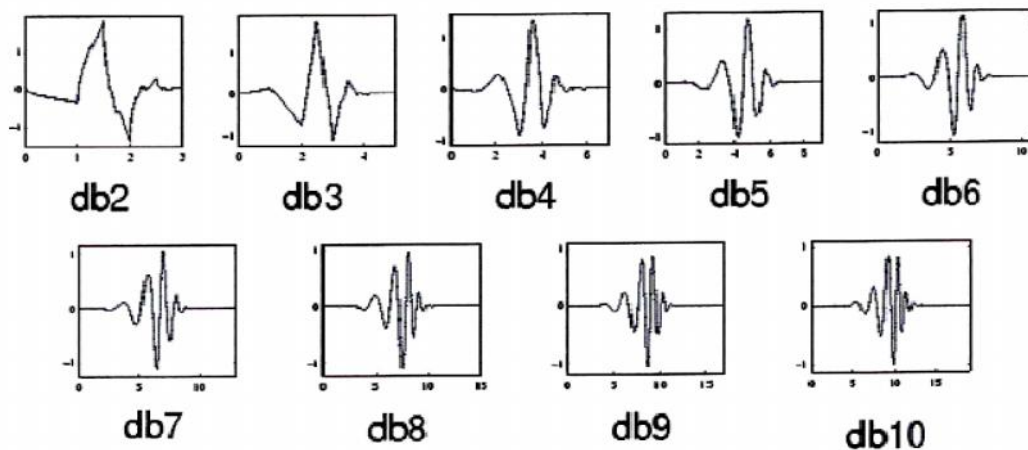


Figura 2.5 Las Wavelets de Daubechies (dbN). Donde N es el orden y $N \in \mathbb{R}^+$

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

Con el fin de agregar un poco de simetría a sus wavelets, Daubechies creó la familia de Wavelets Symmlets. Esta familia es de soporte compacto y puede realizar la transformada continua y discreta de Wavelet. Las Symmlets pueden ser ortogonales, biortogonales, y están cerca de ser simétricas. El número de momentos de desvanecimiento es N. Y su apariencia se muestra en la figura 2.6:

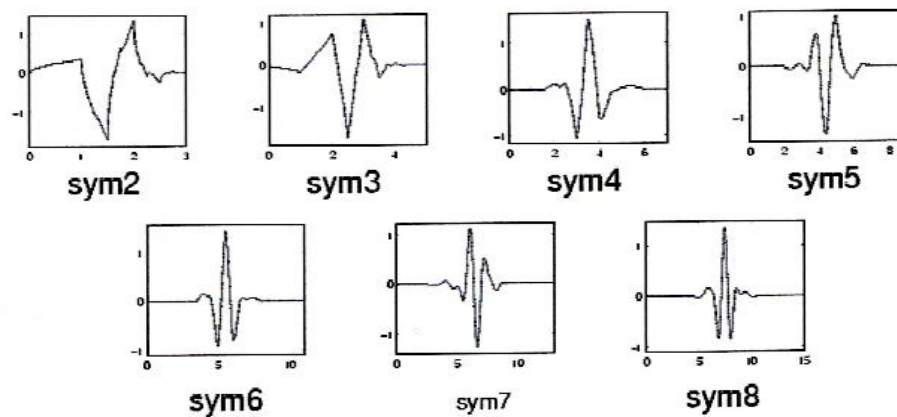


Figura 2.6 Symmlets (symN). Donde N es el orden.

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

Wavelet Coiflet.

Estas Wavelets fueron igualmente creadas por Daubechies con ayuda de Coifman.

Esta familia de Wavelets posee un mayor número de momentos de desvanecimiento: $2N$. La función Wavelet y la de escalamiento son mucho más simétricas que las Wavelets Symmlets. Poseen soporte compacto y el número de filtros es $6N$. Su apariencia se muestra en la figura 2.7.

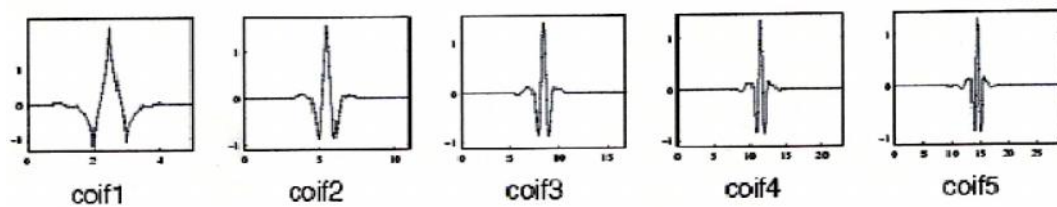


Figura 2.7 Coiflets (coifN). Donde N indica el orden.

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

Wavelets Biortogonales.

Estas Wavelets poseen soporte compacto y simetría, así como la reconstrucción exacta de la señal que son posibles con filtros FIR (Finite Impulse Response), lo cual en las Wavelets ortogonales es imposible excepto en la Wavelet Haar. El orden de estas Wavelets está dado por N_r y N_d , la primera para la reconstrucción y la segunda para la descomposición.

Los momentos de desvanecimiento de Ψ están dados por N_r . La apariencia de estas Wavelets se muestra en la figura 2.8:

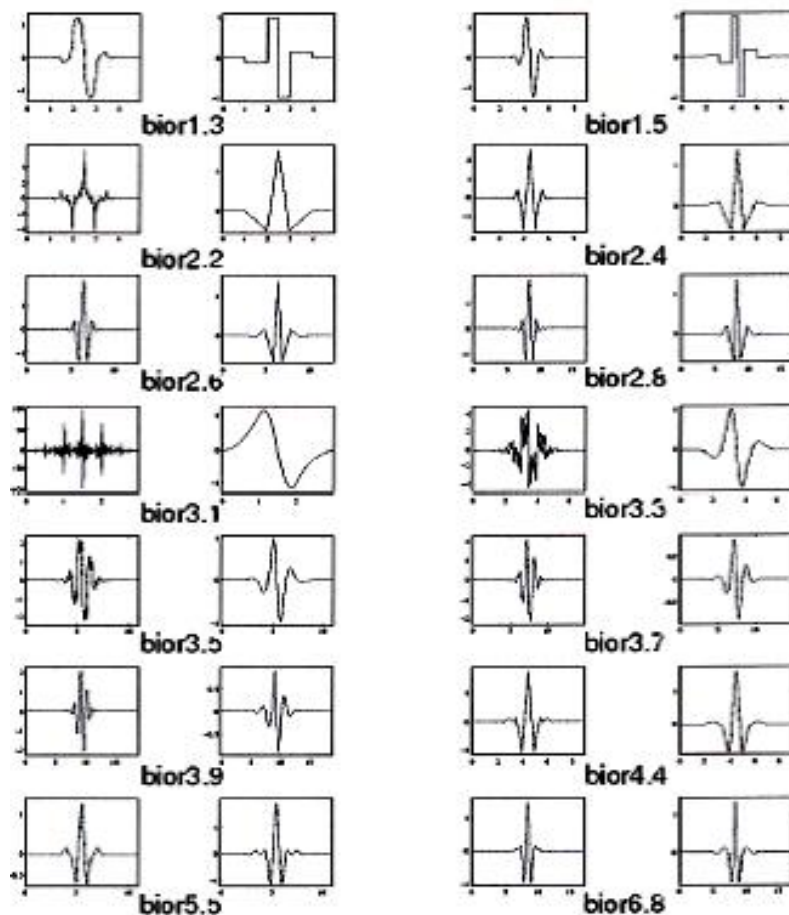


Figura 2.8 Wavelets Biortogonales ($\text{bior}N_r.N_d$).

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

Wavelet Mexican Hat.

Otra wavelet muy recurrida para el análisis de señales es la Wavelet Mexican Hat por la forma de su gráfico.

Esta wavelet posee fase lineal y su apariencia se muestra en la figura 2.9:

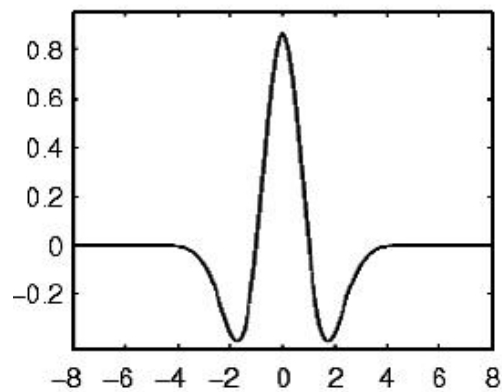


Figura 2.9 Wavelet Mexican Hat.

Fuente:http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo_2.pdf

Wavelet Morlet.

En la figura 2.10 se puede ver que la Wavelet Morlet es simétrica. Por otro lado no posee características de ortogonalidad ni biortogonalidad, además que sólo es útil para realizar la Transformada Continua Wavelet.

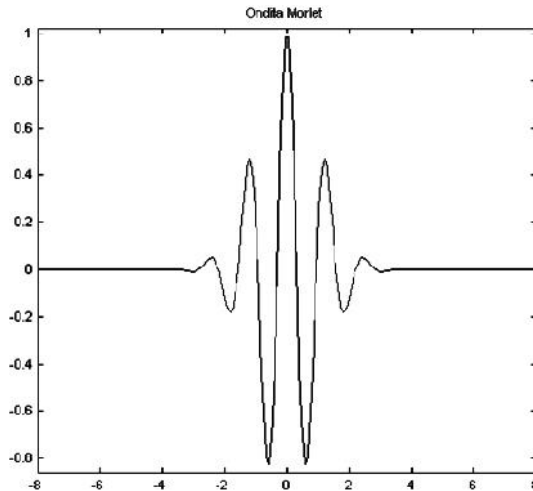


Figura 2.10 Wavelet Morlet

Fuente:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/hernandez_d_m/capitulo2.pdf

Esta gran variedad de funciones Wavelets comparten la característica de ser funciones finitas, lo que se aprovecha para manipularlas mediante las variables de dilatación y traslación. Estos valores determinan que tan abierta o cerrada estará la señal y también dará información sobre el lugar donde se centrará la señal en el plano que se está utilizando.

2.3.1.4. JPEG.

“El JPEG (Joint Photographic Experts Group) es el método de compresión más utilizado actualmente para la compresión de imágenes con pérdida. Este método utiliza la transformada discreta del coseno (DCT), que se calcula empleando números enteros, por lo que se aprovecha de algoritmos de computación veloces. El JPEG consigue una compresión ajustable a la calidad de la imagen que se desea reconstruir.”³

³ <http://coco.ccu.uniovi.es/immed/compresion/descripcion/jpeg/jpeg.htm>

2.3.1.4.1. Codificación de una imagen JPEG.

Codificando con longitud variable los coeficientes, la imagen se puede comprimir aún más. El codificador más utilizado es el algoritmo de Huffman, que se encarga de transmitir los coeficientes ordenados. Una razón para utilizar el codificador de Huffman es que es fácil de implementar. Para comprimir los símbolos de los datos, el codificador de Huffman crea códigos más cortos para símbolos que se repiten frecuentemente y códigos más largos para símbolos que ocurren con menor frecuencia.

2.3.1.4.2. Formatos.

“**JPEG/Exif** es el formato de imagen más común utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen, junto con **JPG/JFIF**, que también es otro formato para el almacenamiento y la **transmisión** de imágenes fotográficas en la World Wide Web. Estas variaciones de formatos a menudo no se distinguen, y se llaman JPEG. Los archivos de este tipo se suelen nombrar con la extensión **.jpg**.”⁴

2.3.2. TRANSMISIÓN DE DATOS.

La transmisión de datos es el intercambio o distribución de información digital entre dos o más ordenadores y entre distintos dispositivos de almacenamiento.

2.3.2.1. Objetivos de la transmisión de datos.

Los principales objetivos que debe satisfacer un sistema de transmisión de datos son:

- Reducir tiempo y esfuerzo.
- Aumentar la velocidad de entrega de la información.

⁴ http://es.wikipedia.org/wiki/Joint_Photoshopping_Group

- Reducir costos de operación.
- Aumentar la capacidad de las organizaciones a un costo incremental razonable.
- Aumentar la calidad y cantidad de la información.

2.3.2.2.Representación de los datos.

El propósito de una **red** es transmitir información desde un equipo otro. Para lograr esto, primero se debe decidir cómo se van a codificar los datos que serán enviados. En otras palabras, la representación informática. Esta variará según el tipo de datos, los cuales pueden ser: audio, video, texto, gráficos.

La representación de datos puede dividirse en dos categorías:

Representación digital: que consiste en codificar la información como un conjunto de valores binarios, en otras palabras, en una secuencia de 0 y 1.

Representación analógica: que consiste en representar los datos por medio de la variación de una cantidad física constante.

2.3.2.3.Comunicaciones locales y remotas.

La transmisión de datos consiste en el movimiento de información de un punto a otro. El destinatario puede encontrarse cerca o lejos del emisor.

Según la ubicación geográfica se puede hablar de dos tipos de transmisión de datos:

Transmisión de datos local. También denominada "en planta". Las distancias son pequeñas. En este caso es la propia organización (empresa, universidad, factoría) la que construye las líneas de comunicaciones.

Transmisión de datos remota. La distancia entre los equipos que se quieren comunicar es mucho mayor. Es necesario acceder a las líneas de telecomunicaciones para que se realice. Normalmente se accede a las líneas proporcionadas por el servicio telefónico.

2.3.2.4. Medio de transmisión de datos.

Para que ocurra la transmisión de datos, debe haber una línea de transmisión entre los dos equipos, también denominada canal de transmisión o canal.

Estos canales de transmisión están compuestos por varios segmentos que permiten la circulación de los datos en forma de ondas electromagnéticas, eléctricas, luz y hasta ondas acústicas. Es, de hecho, un fenómeno de vibración que se propaga a través de un medio físico.

2.3.2.5. Formas de transmisión de datos entre dispositivos electrónicos.

Transmisión Analógica: Estas señales se caracterizan por el continuo cambio de amplitud de la señal. En la ingeniería de control de procesos la señal oscila entre 4 a 20 mA, y es transmitida en forma puramente analógica. En una señal analógica el contenido de información es muy restringido; tan solo el valor de la corriente y la presencia o no de esta puede ser determinado.

Transmisión Digital: Estas señales no cambian continuamente, sino que es transmitida en paquetes discretos. No es tampoco inmediatamente interpretada, sino que debe ser primero decodificada por el receptor. El método de transmisión también es otro: como pulsos eléctricos que varían entre dos niveles distintos de voltaje. En lo que respecta a la ingeniería de procesos, no existe limitación en cuanto al contenido de la señal y cualquier información adicional.

2.3.2.6. Codificación de señales de transmisión.

A fin de que sea posible el intercambio de datos, se debe elegir una codificación para transmitir las señales. Esto depende, básicamente, del medio físico que se utilice para transmitir datos, de la garantía de la integridad de los mismos y de la velocidad de transmisión.

2.3.2.7. Protocolos de comunicación.

Un protocolo es un lenguaje común utilizado por todos los actores en la comunicación para intercambiar datos. Sin embargo, su función no se detiene allí.

Un protocolo también permite:

- El inicio de las comunicaciones.
- El intercambio de datos.
- La detección de errores.
- Una finalización "educada" de las comunicaciones.

2.3.2.8. Modos de Transmisión.

Transmisión Paralela: Es el envío de datos de byte en byte sobre un mínimo de ocho líneas paralelas por medio de una interfaz paralela como por ejemplo la Interfaz Paralela Centronics utilizada para impresoras.

Transmisión en Serie: Es el envío de datos de bit a bit sobre una interfaz serial. Requiere menos cables que la transmisión paralela, con el inconveniente que el tiempo de transmisión al ser transmitida se incrementa conforme el tamaño de la cadena de los bits aumenta. Por ejemplo la RS232 y la RS485.

2.3.3. MATLAB.

“El nombre de **MATLAB** proviene de la contracción de los términos MATrixLABoratory, es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.”⁵

2.3.3.1. Formatos de imágenes soportados por MATLAB

Matlab soporta los siguientes formatos de imágenes:

.BMP, .HDF, .JPEG, .PCX, .TIFF, .XWB

La mayoría de las imágenes disponibles hoy en día, se encuentran en el formato JPEG, el cual es el nombre para el estándar de compresión de imágenes que más se utiliza.

Al trabajar en Matlab se deben considerar múltiples factores como la manera en la que se carga una imagen, el formato correcto, guardar la información de diferentes maneras, como desplegar una imagen, conversiones entre diferentes formatos de imágenes, etc.

2.3.3.2. MATLAB GUIDE (Matlab Graphical User Interface Development Environment).

Matlab Guide es un entorno de programación visual que posee Matlab para poder realizar y ejecutar programas de simulación a medida y de forma simple, tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

⁵ http://www.usc.es/gir/docencia_files/tdd/tutorial_matlab.pdf

El comando `guide` abre la consola de edición de la parte gráfica de la aplicación a implementar (.fig), es decir, colocar botones, cuadros de dialogo, graficas, texto, etc. La figura 2.11 muestra el entorno de la interfaz gráfica de Matlab.

Cada uno de estos elementos tiene un conjunto de propiedades a las cuales podemos acceder con el botón derecho del mouse.

Para editar las propiedades de cada elemento seleccionamos la opción *Properties Inspector* y se abre una consola (la cual variará según que elemento se esté editando) con todas las propiedades que podemos editar, ejemplo: color, posición, tamaño, fuente, etc.

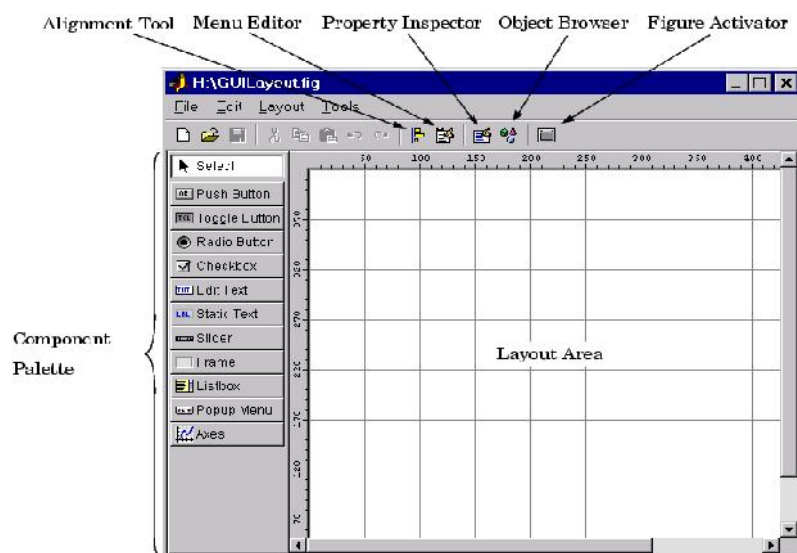


Figura 2.11 Matlab GUIDE

Fuente:

<http://www.google.com.ec/url?sa=t&source=web&cd=4&sqi=2&ved=0CCgQFjAD&url=http%3A%2F%2Fwww.ingelec.uns.edu.ar%2Ficd2763%2Ftut.doc>

2.4. Hipótesis.

¿Con la implementación de un modelo de compresión de imágenes de extensión .jpeg mediante MATLAB se mejorará la transmisión de datos?

Variable Independiente: Compresión de imágenes.

Variable dependiente: Transmisión de datos.

CAPÍTULO III

3 MARCO METODOLÓGICO:

3.1. Enfoque.

Este proyecto de MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS, tiene un enfoque analítico – deductivo en lo que respecta al análisis de las técnicas de compresión de imágenes y a la determinación de un código adecuado para la compresión de imágenes de extensión .jpeg.

3.2. Modalidad básica de investigación.

Bibliográfica- documental.

Las fuentes de consulta más utilizadas son: libros y documentos consultados en internet, obteniendo documentos de gran utilidad con un muy buen contenido de lo requerido para la explicación de las variables utilizadas en este trabajo.

Aplicada.

Con la implementación de este proyecto se puso en práctica lo aprendido en la carrera de electrónica y comunicaciones, en especial en lo que se refiere al área de comunicaciones.

Experimental.

Antes de la implementación definitiva, se analizaron los diferentes tipos de compresión de imágenes, sus ventajas y desventajas para obtener un buen rendimiento.

3.3. Tipo o nivel de investigación.

Descriptiva.

Se realizó un estudio sobre los diversos tipos y algoritmos utilizados en el proceso de compresión y descompresión de imágenes.

Explicativa.

Se analizaron las causas y efectos de tener una imagen sin comprimir y comprimida, permitiendo hallar una técnica adecuada para su fácil portabilidad, almacenamiento y transmisión.

3.4. Población y muestra.

El número de docentes de la carrera de Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato es de 20, a los cuales se les preguntó acerca de la calidad de la imagen antes y después de comprimirla.

3.5. Métodos y técnicas de la investigación.

Deductivo – Inductivo.- Para poder llegar a la solución del problema, se dedujeron las causas que originan los problemas al momento de transmitir y almacenar imágenes de extensión .jpeg.

Analítico – Sintético.- Debido a la deficiencia en el almacenamiento y transmisión de imágenes de extensión .jpeg, se tomó la decisión de utilizar la interfaz guide de Matlab para desarrollar una aplicación que permita realizar el proceso de compresión – descompresión mediante el uso de la Transformada Discreta Wavelet.

Encuesta.- Se preparó una lista de preguntas dirigida a los docentes de la carrera de Ingeniería en Electrónica y Comunicaciones de la F.I.S.E.I. -U.T.A. para poder evaluar la factibilidad del proyecto con mayor facilidad. (Anexo A).

3.6. Operacionalización de las variables.

Variable Independiente: Compresión de imágenes.

Tabla 3.1 Operacionalización - Compresión de imágenes.

Elaborado por: Investigador

Conceptualización	Dimensión	Indicadores	Ítems	Instrumento
La compresión de imágenes consiste en reducir el volumen de datos tratables con la menor pérdida de calidad posible.	Compresor	Reduce el número de bits de las imágenes para optimizarlas.	¿Conoce usted acerca del formato de imágenes de extensión .jpeg?	Encuesta
	Imagen	Representación visual de una cosa.	¿Conoce usted que es un compresor de imágenes?	Encuesta

Variable dependiente: Transmisión de datos.

Tabla 3.3 Operacionalización – Transmisión de datos

Elaborado por: Investigador

Conceptualización	Dimensión	Indicadores	Ítems	Instrumento
La transmisión de datos es el intercambio o distribución de información digital entre dos o más ordenadores y entre distintos dispositivos de almacenamiento.	Información digital	Todo aquello que está representado mediante ceros y unos dentro de una computadora	¿Cree usted que se mejorará la transmisión de imágenes de extensión .jpeg al comprimirlas?	Encuesta
	Ordenador	Máquina capaz de tratar información automáticamente mediante operaciones matemáticas y lógicas realizadas con mucha rapidez y controladas por programas informáticos.		
	Dispositivos de almacenamiento de datos.	Componentes que leen o escriben datos en medios o soportes de almacenamiento		

CAPÍTULO IV

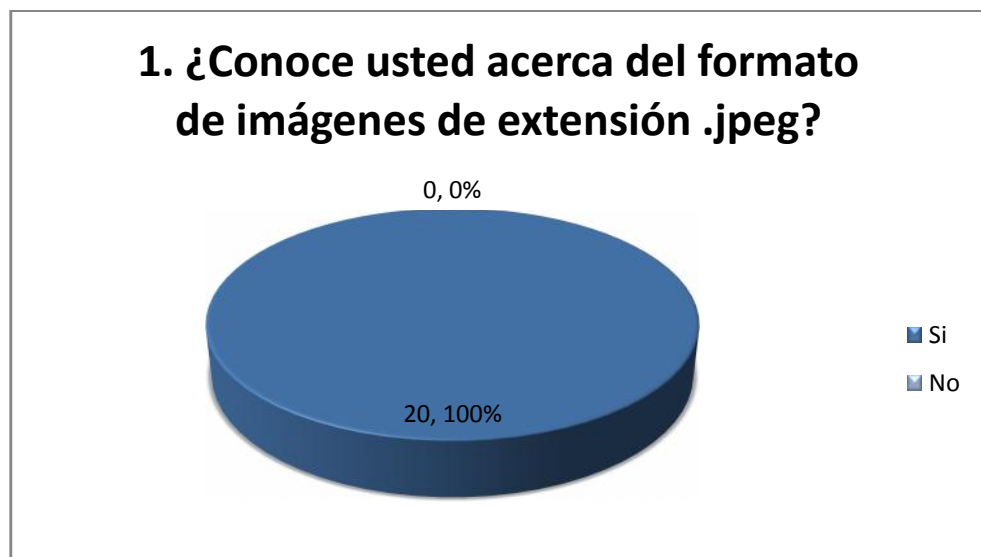
4 ANÁLISIS DE RESULTADOS.

1. ¿Conoce usted acerca del formato de imágenes de extensión .jpeg?

Tabla 4.1 Resultados - Pregunta 1

Elaborado por: Investigador

Ítem	Frecuencia	Porcentaje
Si	20	100%
No	0	0%
Total	20	100%



Graf. 4.1 Resultados - Pregunta 1

Elaborado por: Investigador

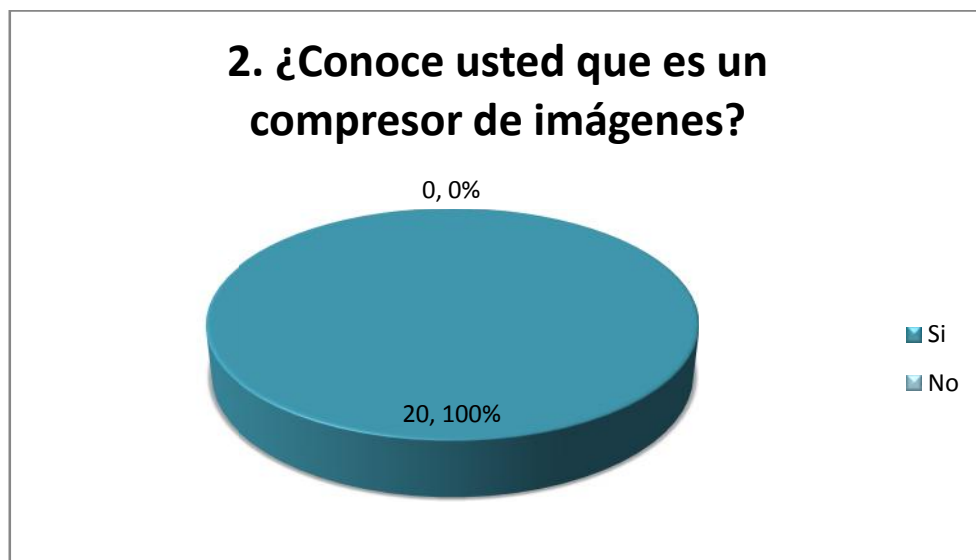
Con un porcentaje del 100%, se interpreta que el total de los docentes entrevistados conoce acerca del formato de imágenes de extensión .jpeg.

2. ¿Conoce usted que es un compresor de imágenes?

Tabla 4.2 Resultados - Pregunta 2

Elaborado por: Investigador

Ítem	Frecuencia	Porcentaje
Si	20	100%
No	0	0%
Total	20	100%



Graf. 4.2 Resultados - Pregunta 2

Elaborado por: Investigador

Un número de 20 docentes entrevistados que representa al 100%, conoce lo que es un compresor de imágenes.

3. ¿Cree usted que se mantendrá la calidad de las imágenes de extensión .jpeg luego de descomprimirlas?

Tabla 4.3 Resultados - Pregunta 3

Elaborado por: Investigador

Ítem	Frecuencia	Porcentaje
Si	16	80%
No	4	20%
Total	20	100%



Graf. 4.3 Resultados - Pregunta 3

Elaborado por: Investigador

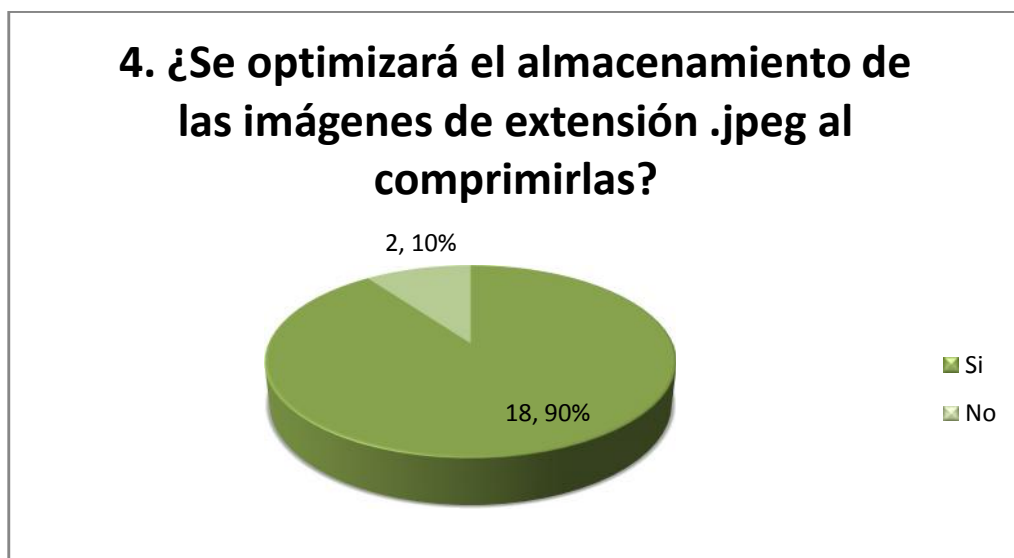
Tan solo 4 personas que representa el 20% del total de los encuestados creen que no se mantendrá la calidad de las imágenes de extensión .jpeg luego de descomprimirlas, mientras que, 16 personas que representa al 80% creen que si se mantendrá la calidad de las imágenes luego de finalizar el proceso de compresión – descompresión.

4. ¿Se optimizará el almacenamiento de las imágenes de extensión .jpeg al comprimirlas?

Tabla 4.4 Resultados - Pregunta 4

Elaborado por: Investigador

Ítem	Frecuencia	Porcentaje
Si	18	90%
No	2	10%
Total	20	100%



Graf. 4.4 Resultados - Pregunta 4

Elaborado por: Investigador

Un total de 18 personas que representa el 90% de los encuestados creen que se optimizará el almacenamiento de las imágenes de extensión .jpeg al comprimirlas, mientras que, tan solo 2 personas representando el 10% no lo considera de esa manera.

5. ¿Cree usted que se mejorará la transmisión de las imágenes de extensión .jpeg al comprimirlas?

Tabla 4.5 Resultados - Pregunta 1

Elaborado por: Investigador

Ítem	Frecuencia	Porcentaje
Si	18	90%
No	2	10%
Total	20	100%



Graf. 4.5 Resultados - Pregunta 5

Elaborado por: Investigador

Un total de 18 personas que representa el 90% de los encuestados creen que se mejorará la transmisión de las imágenes de extensión .jpeg al comprimirlas, frente a un número de 2 personas representando un 10% considera que no se mejorará la transmisión de las mismas al comprimirlas.

4.1. Interpretación y análisis.

De acuerdo a los resultados obtenidos, el total de los encuestados conocen acerca del formato de imágenes de extensión .jpeg; y así mismo, conocen lo que es un compresor de imágenes; mientras que, una pequeña cantidad del número total de encuestados desconocen sobre si se mantendrá la calidad de las imágenes de extensión .jpeg luego de descomprimirlas. Por otro lado, la mayor parte coincide con el mejoramiento, almacenamiento y transmisión de las imágenes de extensión .jpeg comprimidas.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES.

5.1. Conclusiones.

- Luego de realizar la encuesta, los docentes encuestados consideran factible la realización de un compresor de imágenes de extensión .jpeg.
- Con la implementación de un compresor de imágenes de extensión .jpeg se podrá reducir el tamaño de ese tipo de archivos.
- La transmisión de este tipo de imágenes mejorará debido a su menor volumen, permitiendo disminuir el tiempo de transmisión.

5.2. Recomendaciones.

- Implementar una interfaz gráfica mediante matlab que permita realizar la compresión de imágenes de extensión .jpeg.
- Utilizar una técnica de compresión de imágenes de extensión .jpeg que alcance un buen nivel de compresión manteniendo la calidad de las mismas.
- Transmitir las imágenes de extensión .jpeg descomprimidas en menor tiempo que las imágenes originales.

CAPÍTULO VI

6 PROPUESTA.

“MODELO DE COMPRESIÓN DE IMÁGENES DE EXTENSIÓN .JPEG MEDIANTE MATLAB PARA EL MEJORAMIENTO DE LA TRANSMISIÓN DE DATOS”

6.1. Datos informativos.

Institución ejecutora: Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

Ubicación: Campus Huachi

Teléfono: (03) 2415288

Visión: La Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, por sus niveles de excelencia, se constituirá como un centro de formación superior con liderazgo y proyección nacional e internacional.

Misión: Formar profesionales líderes competentes, con visión humanista y pensamiento crítico, a través de la Docencia, la Investigación y la Vinculación, que apliquen, promuevan y difundan el conocimiento respondiendo a las necesidades del país.

Beneficiarios: Estudiantes de la carrera de Electrónica y Comunicaciones de la F.I.S.E.I. – U.T.A.

6.2. Antecedentes de la propuesta.

Se pudo observar que al momento de transmitir imágenes de extensión .jpeg, el tamaño de los archivos provoca lentitud en la velocidad de transmisión, haciendo del envío de este tipo de datos un proceso ineficiente y tedioso. Además, este tipo de imágenes requieren de mayor cantidad de memoria para almacenarlas, espacio que podría ser mejor utilizado con otros tipos de archivos o a su vez con mayor cantidad de imágenes de extensión .jpeg.

El formato jpeg es un formato abierto, cuyos derechos son libres y puede ser usado o implementado en un programa sin tener que pagar derechos por ello.

El formato JPEG suele usar nombres de archivo con las extensiones *.JPEG o *.JPG y surgió como una respuesta a las limitaciones de otros formatos, entre ellos el GIF, en cuanto a calidad y tamaño.

JPEG es un formato de compresión con pérdida, esto quiere decir que al guardar la imagen algo de la información se reduce, perdiéndose un poco de calidad, aunque, es imperceptible al ojo humano. Con ello se consigue reducir el tamaño del archivo y, por tanto, mejorar la velocidad de transmisión. El sistema de compresión que usa jpg se basa en reducir información promediándola en las zonas de degradado. A grandes rasgos, esto quiere decir que se calcula el valor de color de algunos pixeles en función del color de los pixeles que les rodean. Debido a ello, este formato es muy eficiente a la hora de almacenar imágenes que posean muchos degradados y matices de color, mientras que es casi inútil cuando se enfrenta a dibujos con grandes extensiones de colores planos y uniformes o con bordes muy definidos.

6.3. Justificación.

A través de este proyecto se logrará mejorar el proceso de transmisión de imágenes de extensión .jpeg mediante el diseño de un modelo de compresión realizado en guie de matlab, manteniendo la calidad de la imagen con cambios imperceptibles para el ojo humano luego de recuperarla.

Los usuarios de internet que suelen enviar este tipo de archivos mediante correo electrónico, o quienes las transfieren entre distintos dispositivos de almacenamiento, verán a esta aplicación como una herramienta muy útil para optimizar esos procesos.

Los principales beneficiados con el desarrollo de este proyecto, serán los estudiantes de los últimos semestres de la carrera de Electrónica y Comunicaciones de la F.I.S.E.I. – U.T.A., quienes podrán utilizarlo como material de consulta en el estudio de la asignatura de Procesamiento Digital de Señales (DSP), tomando este documento como una guía introductoria a la Transformada Wavelet especialmente en la DWT (Transformada Discreta Wavelet) y en su aplicación en la compresión de imágenes.

6.4. Objetivos.

6.4.1. Objetivo general.

- Implementar un modelo de compresión de imágenes de extensión .jpeg mediante Matlab para el mejoramiento de la transmisión de datos utilizando la Transformada Discreta Wavelet.

6.4.2. Objetivos específicos.

- Explicar el proceso de compresión de imágenes de extensión .jpeg mediante la Transformada Discreta Wavelet.

- Realizar un modelo de compresión de imágenes de extensión .jpeg mediante la interfaz guide de Matlab.
- Elaborar un manual que permita utilizar la interfaz realizada con mayor facilidad.

6.5. Análisis de factibilidad.

Luego de indagar y seleccionar la Transformada Discreta Wavelet como el método adecuado y eficiente para implementar un compresor de imágenes de extensión .jpeg mediante matlab, se pudo concluir que el proyecto es factible de realizarse.

6.6. Fundamentación.

6.6.1. Transformada Discreta Wavelet (DWT) y Multirresolución.

Dentro de los desarrollos más importantes de la teoría de Wavelets se encuentra el análisis de *multirresolución*. Dada una señal se puede obtener una versión de menor resolución perdiendo poca información, luciendo similar a la original requiriendo menos datos buscando aumentar la redundancia entre los valores de la señal.

Para muchas señales la información más importante se encuentra en las frecuencias bajas, mientras que en las altas frecuencias se encuentran los detalles o matices de la señal. Por ejemplo, en el caso de la voz humana, si se eliminan los componentes con altas frecuencias, la voz suena diferente pero se sigue entendiendo su mensaje. En cambio, si lo que se elimina son las componentes de bajas frecuencias, el mensaje se vuelve irreconocible. El análisis wavelet permite descomponer la señal en aproximaciones y detalles. Este filtrado proporciona el doble de datos de los que son necesarios, este problema se soluciona con la operación de diezmado.

6.6.2. Eliminación de la redundancia.

La redundancia de una imagen es causada por la correlación entre píxeles, de modo que la transformación de píxeles a una representación donde ellos son decorrelacionados elimina la redundancia.

La redundancia se elimina en la descomposición, debido a que las funciones son ortogonales. Esta descomposición define una representación multiresolución denominada representación por wavelets, que para el caso discreto se calcula por medio del empleo de un algoritmo piramidal basado en convoluciones por filtros espejo en cuadratura. La transformada bidimensional por wavelets se puede calcular a partir de una extensión separable del algoritmo de descomposición unidimensional.

En DWT se analizan partes diferentes de la onda que pueden ser vistas a través de ventanas de diferentes anchos en lugar de examinar la señal completa a través de una misma ventana. Las partes de alta frecuencia de la señal usan ventanas pequeñas para obtener una buena resolución en tiempo, las partes de baja frecuencia de la señal usan una ventana grande para obtener una buena información en frecuencia. Una cuestión importante que hay que notar es que las ventanas tienen igual área a pesar de que se varíe la altura y el ancho durante el análisis wavelet. El área de las ventanas es controlada por el principio de incertidumbre de Heisenberg.

“**El principio de incertidumbre de Heisenberg** es un teorema sobre ciertos pares de operadores matemáticos. En mecánica cuántica se aplica a la posición y momento de cualquier partícula. En procesamiento de señales se aplica a la representación temporal y frecuencial de cualquier señal. Ambos son casos particulares de pares de operadores que cumplen las hipótesis del teorema. En mecánica cuántica, significa que no es posible determinar simultáneamente la posición y velocidad (o posición y energía) de una partícula. En procesamiento de señales, el principio de incertidumbre establece una cota a la localización en el

tiempo y en la frecuencia de cualquier señal. Esto significa que si se busca concentrar la mayor parte de la energía de la señal en un intervalo lo más reducido posible de tiempo y en un intervalo lo más reducido posible de frecuencia, una mejora en un dominio implica una pérdida en el otro”.⁶

6.6.3. Histograma de una imagen.

“El histograma de una imagen indica los valores de sus píxeles. Una imagen en escala de grises tendrá en su histograma "x" píxeles con el valor 0, "y" píxeles con el valor 1. El histograma es la representación de la densidad de probabilidad de cada valor de gris que aparece en esa imagen.

En general se representa como un gráfico de barras en el que las abscisas (eje x) representa los distintos colores de la imagen y las ordenadas (eje y) representa la frecuencia relativa con la que cada color aparece en la imagen. El histograma proporciona información sobre el brillo y el contraste de la imagen, y puede ser utilizado para ajustar estos parámetros, eliminar ciertas tonalidades molestas, etc.”⁷

Ventaja de los histogramas.

Los histogramas reducen la carga computacional, pasando de una función bidimensional $f(x,y)$ con $N \times M$ valores a una función unidimensional $H(p)$ con 2^g valores. (Siendo g el número de bits empleados en la digitalización).

6.6.4. Conservación y compactación de la energía y de la entropía.

Una característica importante del análisis de wavelet es la de conservar la energía y la entropía. La energía se define como la suma de los cuadrados de los valores del píxel.

⁶ <http://www.jvuletich.org/Research/TesisVuletich.pdf>

⁷ http://www.ayc.unavarra.es/miguel.pagola/P_histograma.htm

La energía en la transformada wavelet de una imagen es la suma de los cuadrados de los coeficientes transformados. Durante el análisis de wavelet la energía de una señal y la entropía se dividen entre la imagen de aproximación y las imágenes de los detalles, pero la energía total y la entropía no cambian. Durante la compresión sin embargo, se pierde la energía porque el umbral cambia los valores de los coeficientes y por lo tanto la versión comprimida contiene menos energía.

La DWT utilizada en este trabajo fue implementada usando wavelets o filtros ortogonales. Si los filtros están normalizados la DWT conserva la energía y la entropía. Para una señal de entrada $x[n]$ de longitud N , la propiedad de conservación de la energía, puede ser vista como:

$$\sum_{n=0}^{N-1} x^2(n) = \sum_{l=0}^{L-1} w^2$$

La propiedad de conservación de la energía es conveniente para el diseño de los sistemas de compresión puesto que significa que la distorsión cuadrática introducida por la cuantificación de los coeficientes es igual a la distorsión cuadrática en la señal reconstruida. Por lo tanto la propiedad de conservación de la energía simplifica el diseño del compresor debido a que el diseño del cuantificador puede ser desarrollado completamente en el dominio de la transformada.

En el proceso de la imagen hay 256 niveles de la intensidad (escalas) de gris. 0 es negro y 255 es blanco. Cada nivel es representado por un número binario 8-bit (negro 00000000 y blanco 11111111) (Figura 6.1). Una imagen se puede interpretar como una rejilla de pixeles, donde cada pixel puede representarse por el valor binario de 8-bit para una escala de gris. La resolución de una imagen está dada por los pixeles por pulgada cuadrada (es decir, 500dpi significa que un pixel es 1/500ava parte de una pulgada).

Para digitalizar una imagen cuadrada de una pulgada de 500 dpi requiere $8 \times 500 \times 500 = 2$ millones de bits del almacenaje. Usando esta representación la compresión de datos de la imagen es una gran ventaja si se van a almacenar muchas imágenes para ser transmitidas o para ser procesadas.

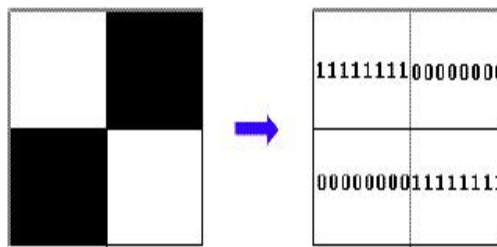


Figura 6.1 Codificación de los píxeles

Fuente: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

Los algoritmos de compresión de imagen apuntan a quitar la redundancia en los datos de una manera que haga posible la reconstrucción de la misma, lo que significa básicamente que los algoritmos de la compresión intentan explotar redundancias en los datos; calculan qué datos necesitan ser guardados para reconstruir la imagen original y por lo tanto que datos se pueden eliminar. Quitando los datos redundantes, la imagen se puede representar en un número más pequeño de bits, y por lo tanto puede ser comprimida.

6.6.5. Propiedades de los datos de la imagen.

6.6.5.1. Entropía.

Una imagen altamente correlacionada tendrá una entropía baja. Por ejemplo a frecuencias muy bajas, una imagen altamente correlacionada será comprimida por muchas diversas técnicas; es más la característica de la imagen y no el algoritmo de compresión que da las buenas tasas de compresión. También, un algoritmo de compresión que es bueno para algunas imágenes no será necesariamente bueno para todas las imágenes.

Una forma de calcular la entropía es: si una imagen tiene G niveles de grises y la probabilidad del nivel de gris k es $P(k)$ la entropía H es:

$$H = - \sum_{k=0}^{G-1} P(k) \log_2(P(k))$$

6.6.5.2. Umbral de una imagen jpeg.

El umbral es un valor que se puede establecer como límite con algún fin. En compresión de imágenes todos los píxeles cuyo nivel de gris se encuentren por debajo de dicho valor se llevan a cero. El umbral hace posible destacar formas u objetos en una imagen.

Es el método más simple de la segmentación de imágenes. Desde una imagen en escala de grises, el umbral se puede utilizar para crear imágenes binarias.

“Las imágenes en escala de grises no se pueden comprimir tanto. Como el ojo humano es mucho más sensible a las variaciones de la luminosidad que de la coloración, el método JPEG puede comprimir los datos de los colores más que los datos de la luminosidad (escala de grises). Generalmente, un archivo JPEG de una imagen en escala de grises es solo un 10-25 % más pequeño que su contrapartida a todo color. Como los valores de la escala de grises sin comprimir son de tan solo 8-bit/píxel (un tercio de los valores del color) la proporción del nivel de compresión es mucho menor que con imágenes en color. El umbral de compresión a partir del cual se nota una pérdida de calidad en imágenes en escala de grises es alrededor de 1:5.”⁸

⁸ <http://www.google.com/ec/url?sa=t&source=web&cd=4&sqi=2&ved=0CCgQFj>

AD&url=http%3A%2F%2Fwww.ingelec.uns.edu.ar%2Ficd2763%2Ftut.doc/telepatologia/telepatologia05.pdf

6.6.6. Medidas de calidad.

6.6.6.1. Medidas de Calidad Subjetivas.

La percepción de una imagen está formada por una compleja interacción en los componentes del sistema visual humano, el ojo y el cerebro. Entre los factores que influyen la percepción están la fidelidad espacial y temporal. Otros factores externos como el ambiente, el estado de ánimo del observador y la distancia a la cual está de la pantalla, pueden jugar un papel muy importante en la opinión del observador sobre la calidad de la imagen. Todos estos factores pueden hacer extremadamente difícil medir la calidad visual en forma exacta y cuantitativa.

Para evaluar subjetivamente la calidad de la imagen luego de realizar el proceso de compresión – descompresión a partir de la interfaz diseñada en Matlab, se hizo uso del test MOS TEST, mediante el cual se pidió a espectadores puntuar la calidad de la imagen descomprimida con respecto a la original en una escala conforme a su percepción. Las calificaciones asignadas por cada espectador son promediadas para así obtener la puntuación de opinión media (MOS, Mean Opinion Score).

6.6.6.2. MOS TEST.

La técnica a usarse para determinar y calificar la calidad de las imágenes es el MOS TEST, que es un test internacional de procesamiento de voz e imágenes.

MOS (Mean Opinion Score). Nota media de opinión. Se refiere a una valoración cuantitativa de la calidad subjetiva de una señal (por ejemplo, voz, imágenes o vídeo).

Según la recomendación de “UIT-R BT.500-11”⁹, una prueba MOS realiza comparaciones entre la imagen a comprimir con la imagen descomprimida, las

⁹http://www.dii.unisi.it/~menegaz/DoctoralSchool2004/papers/ITU-R_BT.500-11.pdf

mismas que visualiza el observador, con una valoración numérica en cuanto a la calidad de las imágenes desde 1 (pobre) hasta 5 (excelente).

6.6.6.3. Guía para evaluar la calidad de la imagen.

A la hora de evaluar las diferencias observadas entre la imagen original y la imagen descomprimida, se utiliza la escala discreta de apreciación de cinco notas que se indica en la tabla 3.1:

Tabla 6.1 Encuesta Mos Test

Elaborado por: Investigador

ESCALA DE CINCO NOTAS DE CALIDAD	
CALIFICACIÓN	DEFINICIÓN
5	Imperceptible
4	Perceptible pero no molesta
3	Ligeramente Molesta
2	Molesta
1	Muy molesta

La recopilación de la información, se la realizó en las instalaciones de la F.I.S.E.I. de la U.T.A., la encuesta fue aplicada a los docentes del área de Electrónica y Comunicaciones.

6.6.6.4. Medidas de Calidad Objetivas.

Relación señal pico a ruido (PSNR).

El parámetro de la razón señal pico a ruido (**PSNR**) es una medida relativa en escala logarítmica de la calidad de la imagen, se basa en el error cuadrático medio

(MSE) entre la imagen original y la imagen modificada. El MSE se calcula con la siguiente fórmula:

$$MSE = \frac{1}{mn} \sum [s(x,y) - \tilde{s}(x,y)]^2$$

Donde m y n representan el largo y ancho de la imagen, $s(x,y)$ es la señal original y $\tilde{s}(x,y)$ es la señal reconstruida. La sumatoria se realiza sobre todos los pixeles.

La raíz del Error Cuadrático Medio (RMSE), es la raíz cuadrada del MSE.

Una vez calculado el MSE, se puede obtener la PSNR. Un valor bajo de MSE, significa menos error en la señal reconstruida con respecto a la señal original; lo cual se traduce en un valor grande de PSNR (en decibeles). Es decir un valor grande de PSNR, es bueno lo que significa que la razón señal a ruido es grande.

$$PSNR_{dB} = 20 \log_{10} \frac{(255)}{RMSE}$$

El valor real de PSNR no es significativo, pero la comparación entre dos valores para diversas imágenes reconstruidas da una medida de calidad. Si se obtiene un método de reducción de ruido en imágenes con un valor de MSE bajo y un valor de PSNR alto, se reconoce como el de mejor desempeño.

La PSNR puede ser calculada fácil y rápidamente y además es una medida de calidad muy popular, ampliamente usada para comparar la calidad de imágenes comprimidas y descomprimidas. Sin embargo esta medida sufre de un número de limitaciones. Una de ellas es la necesidad de una imagen original no modificada la cual no es fácil de conseguir en todos los casos. Otra limitación es que la PSNR no se correlaciona bien con las medidas de calidad subjetiva.

Los valores típicos que adopta este parámetro están entre 30 y 50 dB, siendo mayor cuando la codificación mejora.

Para una imagen en formato RGB, la definición del PSNR es la misma, pero el MSE se calcula como la media aritmética de los MSEs de los tres colores (R, G y B).

6.7. Metodología.

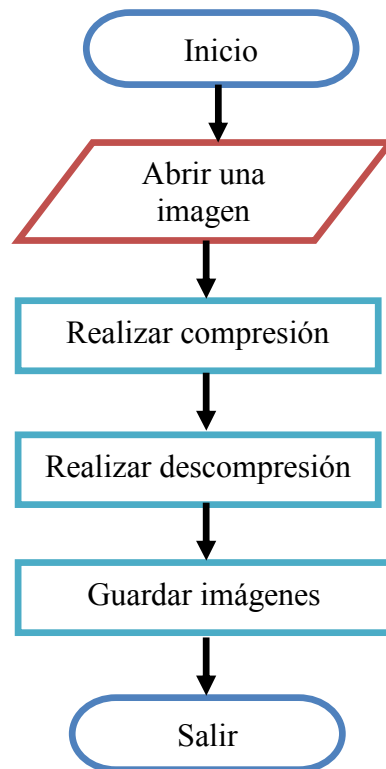


Figura 6.2 Flujograma del programa.

Elaborado por: Investigador

6.7.1. Proceso de compresión de imágenes de extensión .jpeg mediante la transformada Wavelet.

La elección de la Transformada Discreta Wavelet como la técnica adecuada para la realización del modelo de compresión de imágenes de extensión .jpeg es debido a que ofrece información en tiempo - frecuencia instantánea y las representaciones mediante Wavelets se aproximan a la sensibilidad visual del humano, logrando que las pérdidas en la imagen sean imperceptibles.

6.7.1.1. Descomposición de la frecuencia.

El análisis discreto Wavelet en dos dimensiones es realizado usando el concepto de banco de filtros con diferente frecuencia de corte.

Una manera de implementar la DWT (Transformada Discreta Wavelet) es utilizando filtros. Se aplican dos filtros, uno de baja frecuencia y otro de alta frecuencia:

- Un filtro pasa-alto $g[n]$

Elimina todas las frecuencias desde cero hasta la frecuencia de corte, f_c y permite el paso de todas las frecuencias por encima de dicho valor. (Figura 6.3).

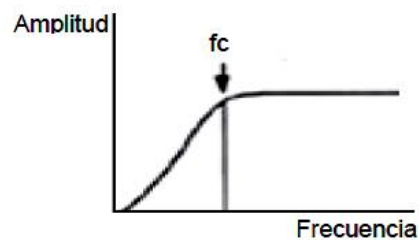


Figura 6.3 Filtro pasa alto.

Fuente: <http://www.dspace.espol.edu.ec/bitstream/123456789/13513/1/D-42644.pdf>

- Un filtro pasa-bajos $h[n]$

Hace exactamente lo contrario que el filtro pasa alto, es decir se caracteriza por permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas. (Figura 6.4).

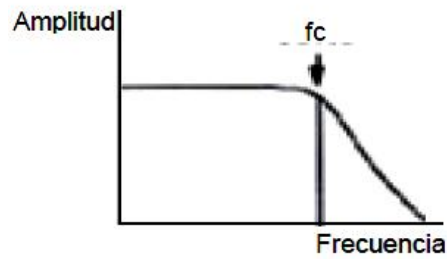


Figura 6.4 Filtro pasa bajos.

Elaborado por: <http://www.dspace.espol.edu.ec/bitstream/123456789/13513/1/D-42644.pdf>

Estos filtros pasabajos $h[n]$ y pasa altos $g[n]$ que actúan primero en una dirección y luego en la otra, se muestra en la figura 6.5 como la unidad básica que realiza la descomposición en multirresolución:

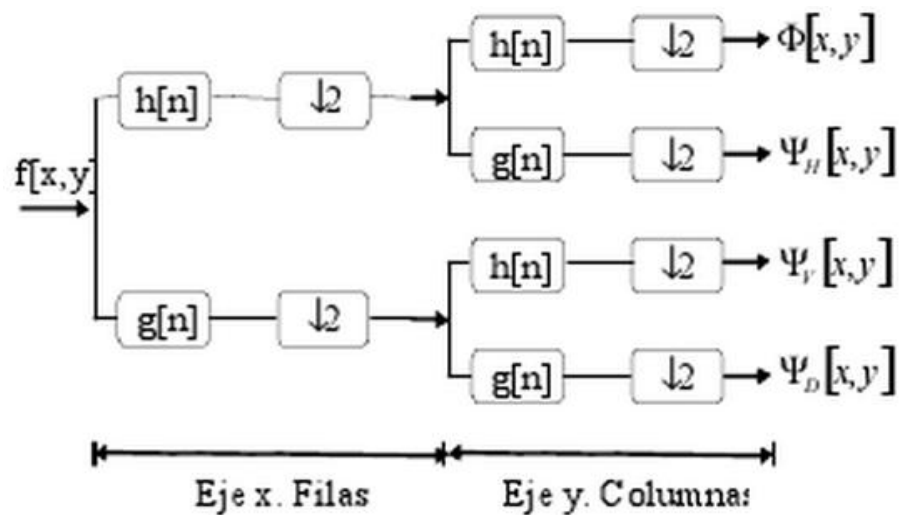


Figura 6.5 Estructura de filtros.

Elaborado por: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

- Con el de baja frecuencia se obtienen datos de aproximación, el contenido es importante y le da a la señal su identidad.
- Con el de alta frecuencia se obtienen datos de detalle que imparten complementación.

Visto desde una perspectiva bidimensional. (Figura 6.6).

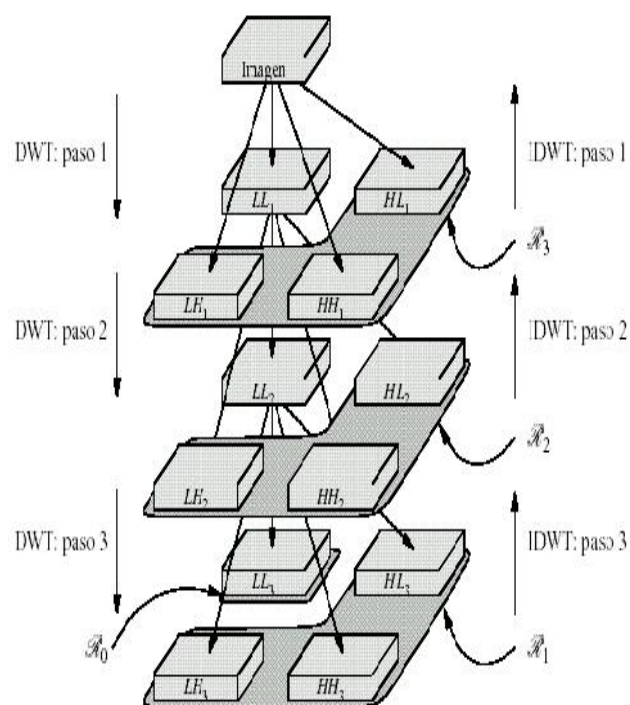


Figura 6.6 Estructura de las subbandas.

Fuente: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

La subbanda LL (del inglés Low, Low) o $\Phi(x, y)$, proviene de la aplicación del filtro de análisis paso bajo en ambas direcciones, la horizontal y la vertical. La subbanda HL (del inglés High, Low) o $\Psi_H(x, y)$, se obtiene aplicando el filtro paso bajo en la dirección vertical y el filtro paso alto en horizontal. Las otras subbandas, se identificarían como LH (del inglés Low, High) o $\Psi_V(x, y)$ y HH (del inglés High, High) o $\Psi_D(x, y)$, respectivamente. El nivel D de la DWT se obtiene aplicando D veces la DWT de la manera que se indica en la Figura 6.6. Esto es, a la subbanda LL producida por la primera aplicación de la DWT se le aplica otra vez la DWT y así sucesivamente. Las subbandas producidas por el paso D de la transformada se denotan como LLd, HLd, LHd y HHd. Las subbandas intermedias no tienen muestras por sí mismas, sino que son un nodo en el árbol y podemos considerar que sus muestras son las de las hojas de las ramas que cuelgan de él.

La descomposición wavelet puede ser interpretada entonces como una descomposición de la imagen en un conjunto de canales o bandas de frecuencia independientes entre sí y espacialmente orientados. Mientras una banda (pasabajo) contiene una versión borrosa de la imagen original, las otras bandas resaltan la información de alta frecuencia (bordes y texturas). Pero más importante aún es que esta última se encuentra separada en las bandas de acuerdo a su ubicación en el espacio: horizontal $\Psi_H(x,y)$, vertical $\Psi_V(x,y)$ y diagonal $\Psi_D(x,y)$.

Al aplicar una descomposición como la que se muestra en la figura 6.6 a una imagen cualquiera, se obtiene una división en subbandas como la que se muestra en la figura 6.7.

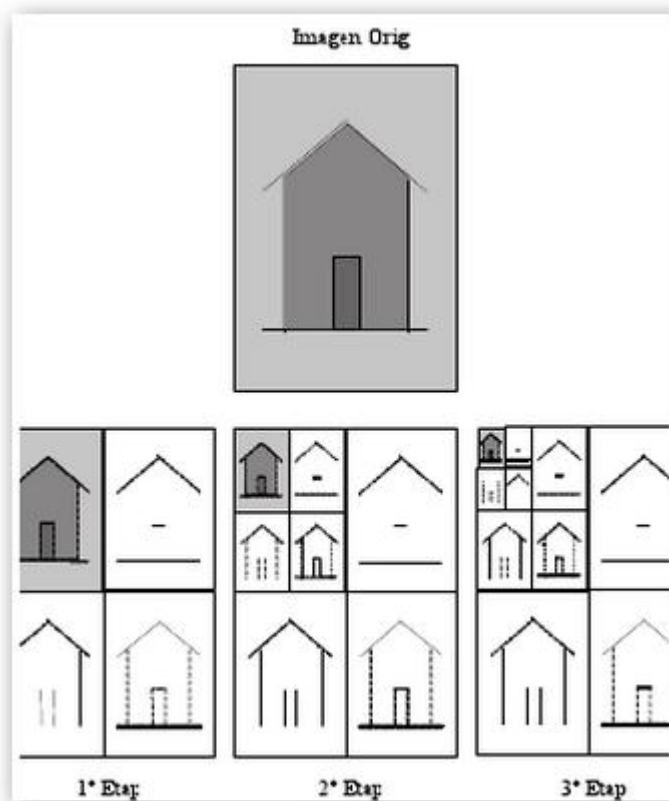


Figura 6.7 Descomposición de una imagen en tres niveles.

Fuente: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

Calcular los coeficientes wavelet en todas las posibles escalas es mucho trabajo. Por ello se elige un subconjunto de estiramientos y translaciones. Estos subconjuntos serán potencia de dos.

Mediante un submuestreo se elimina la duplicación de los datos.

6.7.1.2. Reconstrucción de la señal.

Consiste en ensamblar los componentes obtenidos en el proceso de descomposición, obteniendo la señal original sin pérdida de información. A este proceso se le denomina Transformada Inversa de Wavelet (IDWT).

Mientras que en la descomposición se realizaba filtrado y submuestreo, y en la reconstrucción se realiza sobremuestreo y filtrado.

El sobremuestreo es el proceso de alargar la señal componente insertando ceros entre muestreos. (Figura 6.8)

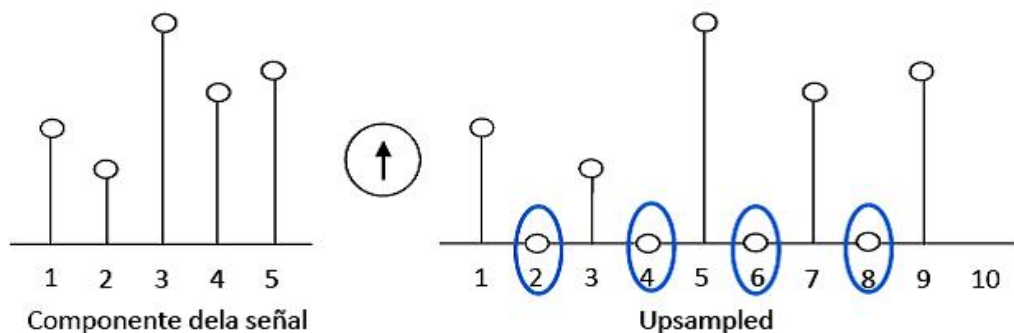


Figura. 6.8 Inserción de ceros a la componente de la señal.

Fuente: <http://www.dspace.espol.edu.ec/bitstream/123456789/13513/1/D-42644.pdf>

6.7.2. Modelo de compresión de imágenes de extensión .jpeg realizado en guide de matlab.

En MATLAB las imágenes son tratadas como señales bidimensionales, consideradas como matrices de N filas y M columnas. En cada nivel de descomposición en la Transformada Discreta Wavelet los datos horizontales son filtrados, de esta manera la matriz de aproximación y las de detalles que se producen son filtradas sobre las columnas, tal como se muestra en la figura 6.9:

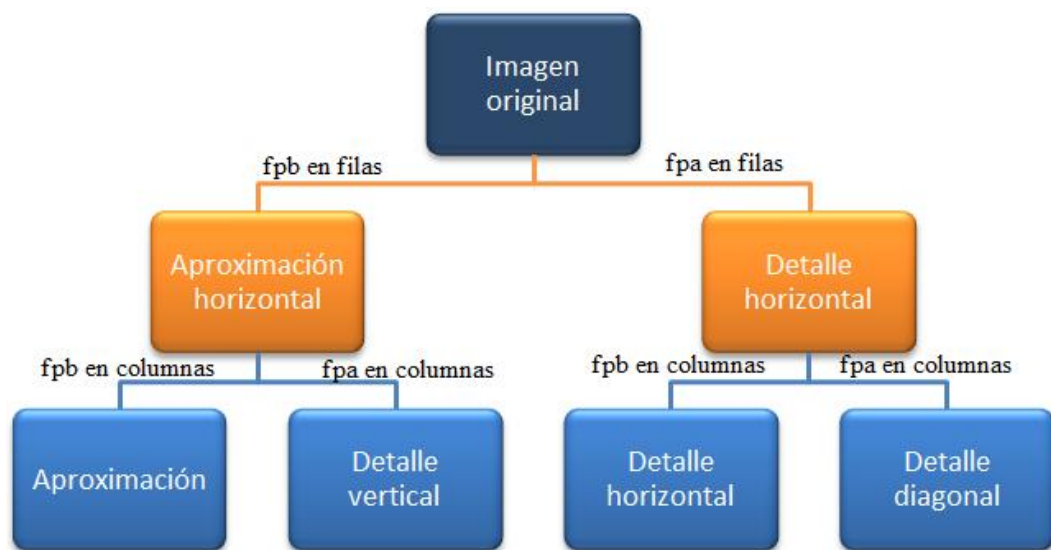


Figura. 6.9 Tratamiento de imágenes en Matlab.

Elaborado por: Investigador

En cada nivel se obtienen cuatro subimágenes: la aproximación que contiene la mayor parte de la energía de la imagen y los detalles vertical, horizontal y diagonal, que contienen valores próximos a cero. En la figura 6.10 se puede observar como a través de Matlab, la imagen se descompone en el primer nivel. De esta manera con la transformada discreta wavelet se obtienen las aproximaciones verticales, horizontales y diagonales.

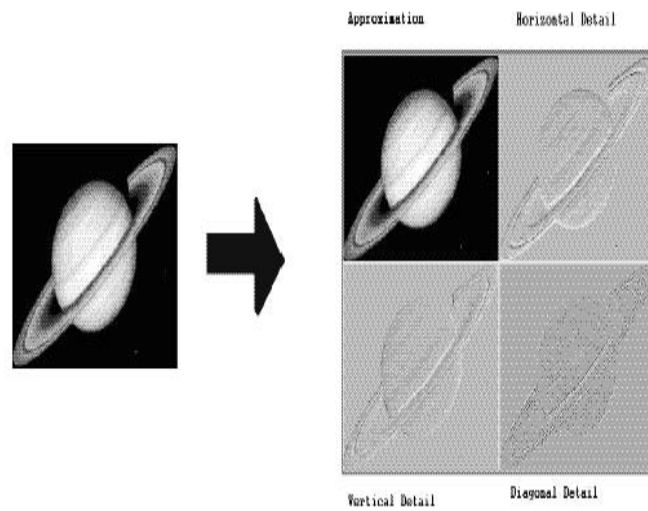


Figura 6.10 Creación de subimágenes.

Fuente: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

Para obtener el siguiente nivel de descomposición la subimagen se descompone una vez más, tal como se observa en la figura 6.11.

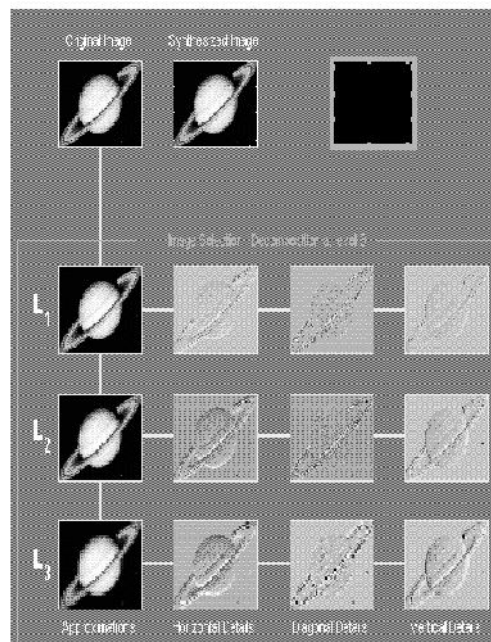


Figura 6.11 Niveles de descomposición.

Fuente: <http://ebookbrowse.com/difusion-multimedia-quijada-doc-d1301563>

6.7.2.1. Interfaz realizada en guide de matlab.

Para un mejor manejo y visualización del proceso de compresión - descompresión de imágenes de extensión .jpeg, se desarrolló una aplicación utilizando el guide de matlab. (Figura 6.12). (El flujograma y el código del programa se presentan en los Anexos B y C, respectivamente).

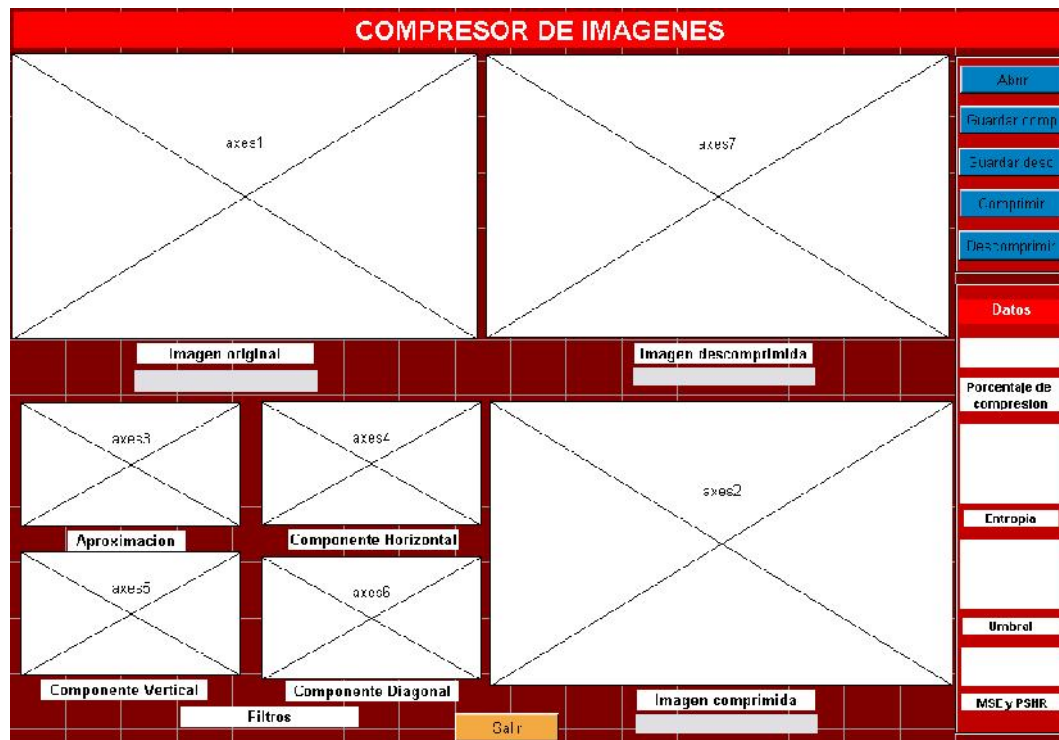


Figura 6.12 Interfaz gráfica del compresor.

Elaborado por: Investigador

Para realizar el proceso de compresión, primero seleccionamos una imagen de extensión .jpeg y convertirla a escala de grises mediante el comando **rgb2gray**, para luego aplicar la primera etapa de filtros. Las imágenes resultantes antes de ser diezmadas se muestran en la Figura 6.13.



(a)

(b)

Figura 6.13 Primera etapa de filtros (a) filtro pasa bajo, (b) filtro pasa alto.

Elaborado por: Investigador

Luego de obtener estas dos imágenes, se realiza el **diezmado vertical**, que consiste en eliminar una columna de cada dos columnas, partiendo del hecho de que entre pixel y pixel la diferencia es despreciable. El resultado se muestra en la Figura 6.14.



(a)

(b)

Figura 6.14 Primera etapa de filtros (a) filtro pasa bajo luego del diezmado, (b) filtro pasa alto luego del diezmado.

Elaborado por: Investigador

En la segunda etapa de filtros se obtienen cuatro imágenes de la mitad del ancho y alto de la imagen original. Cada una de estas cuatro imágenes representa cada una de las componentes de la imagen.

El siguiente paso consiste en repetir el proceso, pero usando cada una de estas dos imágenes con la diferencia de que luego de filtrarlas, el **diezmado** se hará de forma **horizontal**, consiguiendo los coeficientes de Wavelet, estas imágenes se muestran en la Figura 6.15.

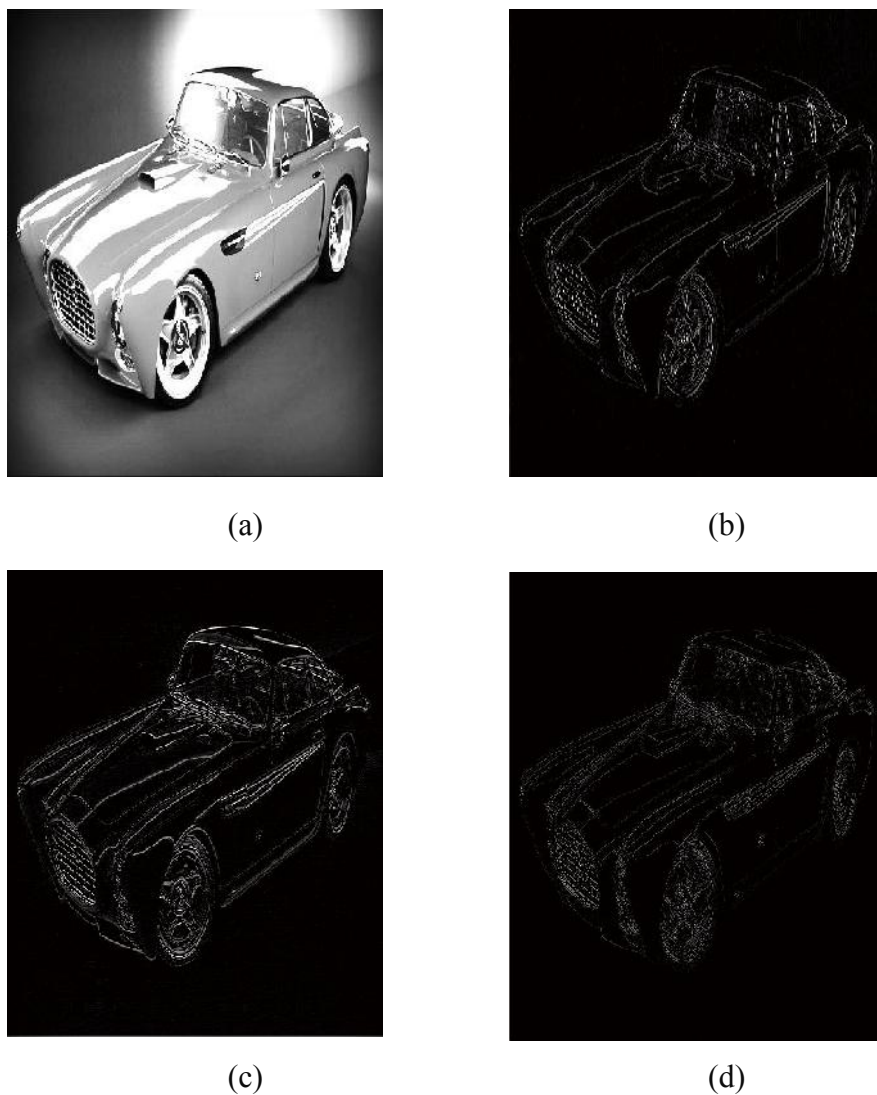


Figura 6.15. Componentes Wavelet (a) Aproximación (b) Componente Horizontal (c) Componente Vertical (d) Componente Diagonal.

Elaborado por: Investigador

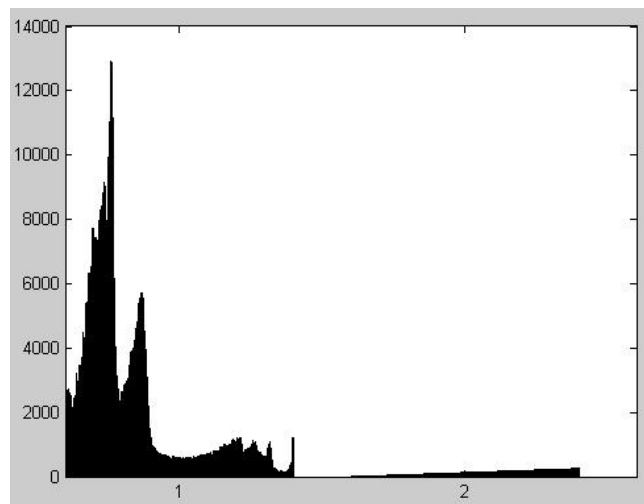
Las imágenes de la parte izquierda pasaron por un filtro pasa bajo y las de la derecha pasaron por un filtro pasa Alto. Las imágenes de la parte superior son las equivalentes a las salidas de los filtros de la imagen de la izquierda de la figura 6.14. Las de la parte inferior son las salidas de los filtros de la imagen de la derecha de la Figura 6.14.

6.7.2.2. Histograma de una imagen en Matlab.

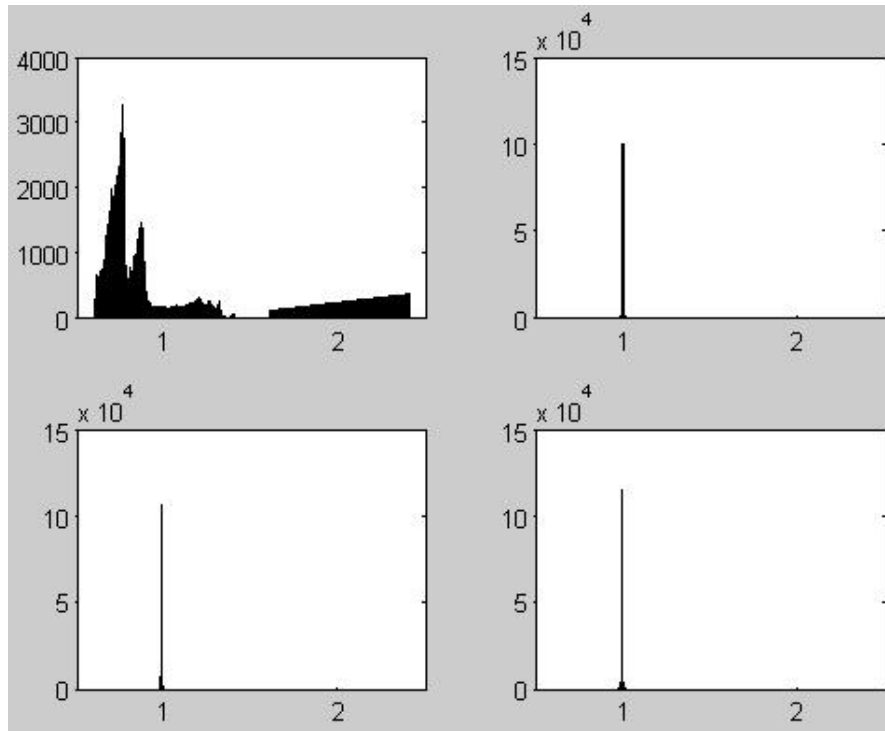
Con un histograma es posible ver la distribución de intensidades en una imagen, lo que se logra utilizando la función **imhist** de matlab.

Al descomponer la imagen, se logra tener una imagen que contendrá la mayor parte de la información, además se tendrán otras tres imágenes que contendrán los detalles, las que tendrán valores próximos a cero, lo que permitirá obtener un histograma concentrado.

Los histogramas de las imágenes se muestran en la figura 6.16.



(a)



(b)

Figura 6.16 Histogramas de las imágenes (a) Imagen original en escala de grises

(b) Componentes Wavelet

Elaborado por: Investigador

De acuerdo a los histogramas se puede observar que la componente Wavelet correspondiente a la aproximación (imagen superior izquierda de la figura 6.16 (b)) contiene la mayor parte de la energía de la imagen original y los histogramas de las componentes: horizontal, vertical y diagonal, se aproximan a cero.

6.7.2.3. Cálculo de la Entropía.

Para calcular la entropía de las imágenes es necesario obtener el histograma y se lo realizó mediante el siguiente proceso en matlab:

```

histograma=imhist(imagen);
histograma=histograma/sum(histograma);
histograma(find(histograma==0))=1;

```

```
loghistograma= log2(histograma);  
H= - sum(histograma.*loghistograma);
```

Dónde:

imhist → Calcula el histograma

find → permite encontrar un valor siempre y cuando se cumpla una condición

log → calcula el logaritmo

sum → Realiza la operación de sumatoria.

6.7.2.4. Cálculo del umbral.

El umbral de compresión puede ser calculado solo para las componentes Wavelet de la imagen y se utilizó la siguiente expresión de matlab:

```
Deadapx=max(max(abs(imagen)))*threshold/100;
```

Dónde:

deadapx → Nombre de la variable que contendrá el valor del umbral.

Max → máximo valor.

abs → Da el modulo o valor absoluto de un valor, en este caso de cada una de las componentes Wavelet de la imagen.

threshold → Calcula el umbral de las imágenes.

6.7.2.5. Error Cuadrático Medio (MSE) y Relación Señal Pico a Ruido (PSNR).

El MSE es el error acumulativo cuadrado entre la imagen descomprimida y la imagen original, y se calculó mediante el siguiente procedimiento en matlab:

```
error = I - i;
```

$MSE = \text{sum}(\text{sum}(\text{error} .* \text{error})) / (h * w);$

$MSE_R = MSE(:, :, 1);$

$MSE_G = MSE(:, :, 2);$

$MSE_B = MSE(:, :, 3);$

$MSE_t = (MSE_R + MSE_G + MSE_B) / 3;$

Donde:

$I \rightarrow$ Imagen original.

$i \rightarrow$ Imagen reconstruida.

$\text{sum} \rightarrow$ Realiza la operación de sumatoria.

h y $w \rightarrow$ Dimensiones de la imagen (alto y ancho).

$MSE_R \rightarrow$ MSE correspondiente a las componentes Red de las imágenes.

$MSE_G \rightarrow$ MSE correspondiente a las componentes Green de las imágenes.

$MSE_B \rightarrow$ MSE correspondiente a las componentes Blue de las imágenes.

$MSE_t \rightarrow$ MSE total (valor promedio).

El PSNR es una medida del error de pico y depende del valor del MSE, calculado con la siguiente expresión:

$\text{psnr} = 20 * \log_{10}((255) / \text{sqrt}(MSE_t));$

Para el caso de la imagen “ferrari-340-competizione-xl.jpg” se obtuvo los siguientes resultados:

Tabla 6.2 Datos del proceso de compresión – descompresión

Elaborado por: Investigador

Imagen	Entropía	Umbral	MSE	PSNR
Original	7.2465		0.222	54.712
Aproximación	7.0335	0.022748		
Componente horizontal	1.8596	0.041248		
Componente vertical	1.5878	0.049594		
Componente diagonal	1.2375	0.046265		

6.7.3. Manual de usuario.

En este manual se explicará acerca del funcionamiento de la interfaz desarrollada para el compresor de imágenes de extensión .jpeg. Se comenzará por explicar cuáles son los requisitos del sistema para la ejecución de la interfaz, así mismo, el entorno que se ha utilizado para su desarrollo y se explicará el procedimiento a realizar sobre la interfaz de usuario del programa para llevar a cabo las distintas funciones que realiza el programa.

6.7.3.1. Requisitos del sistema.

Para el desarrollo de la interfaz se utilizó un computador personal con las siguientes características:

- Procesador Pentium 4 de 2.40 MHZ.
- RAM de 512 MB.
- Sistema operativo: Windows XP.

6.7.3.2. Entorno de desarrollo.

La aplicación se desarrolló utilizando guide de Matlab en su siguiente distribución:

MATLAB. The Language of Technical Computing

Versión 7.1.0.246 (R14) Service Pack 3

Para esta aplicación no se desarrolló un archivo ejecutable, por lo tanto, para poder ejecutarla se deberá tener instalada la versión de Matlab antes mencionada o una versión superior.

6.7.3.3. Pasos para la ejecución de la aplicación:

1. Al ejecutar Matlab7.1, se abre una ventana como se muestra en la figura 6.17:

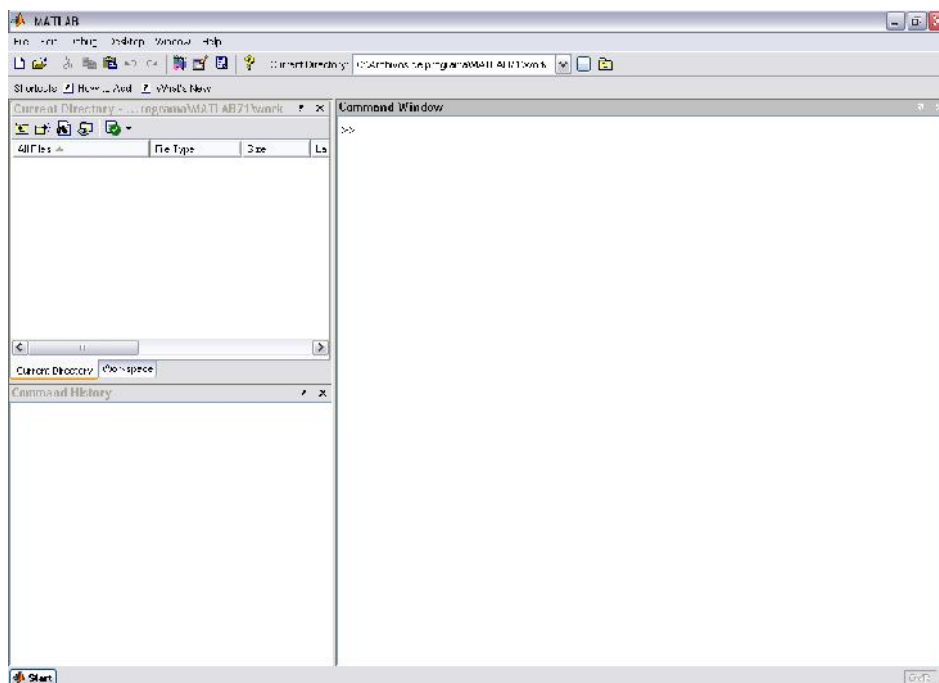


Figura 6.17 Entorno de Matlab.

Elaborado por: Investigador

2. Seleccionar el directorio donde se encuentran los ficheros de la aplicación en **Current Directory** tal y como se muestra en la figura 6.18.



Figura 6.18 Selección del directorio de trabajo.

Elaborado por: Investigador

3. A continuación se ejecuta el comando correspondiente al nombre del programa principal (Matlab distingue las mayúsculas de las minúsculas, por lo tanto, el comando ejecutado debe ser exactamente el mismo con el que se lo haya guardado), y se presiona la tecla Enter. En este caso el nombre de la aplicación es **Compresor** (figura 6.19).

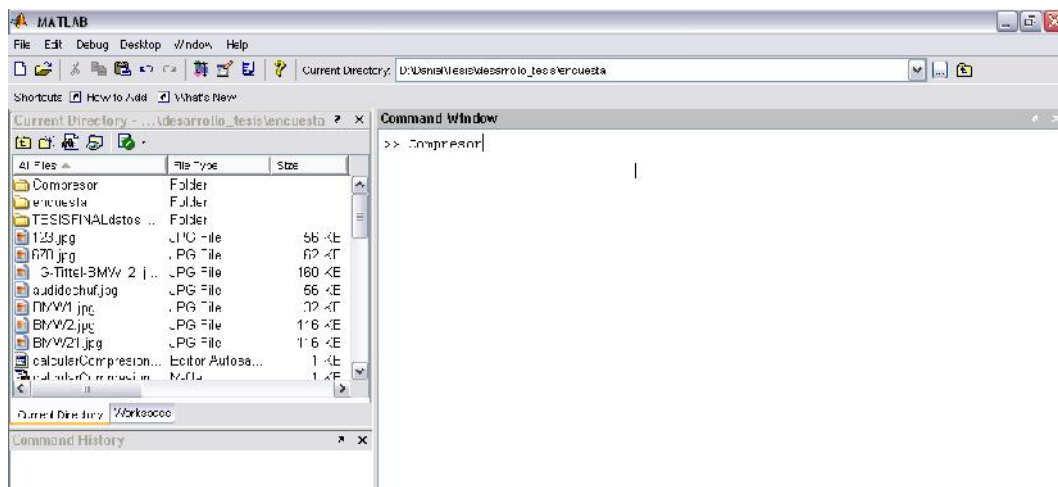


Figura 6.19 Ejecución de la interfaz gráfica del compresor.

Elaborado por: Investigador

4. Cuando se ejecuta el programa aparecerá la interfaz realizada. (Figura 6.20).

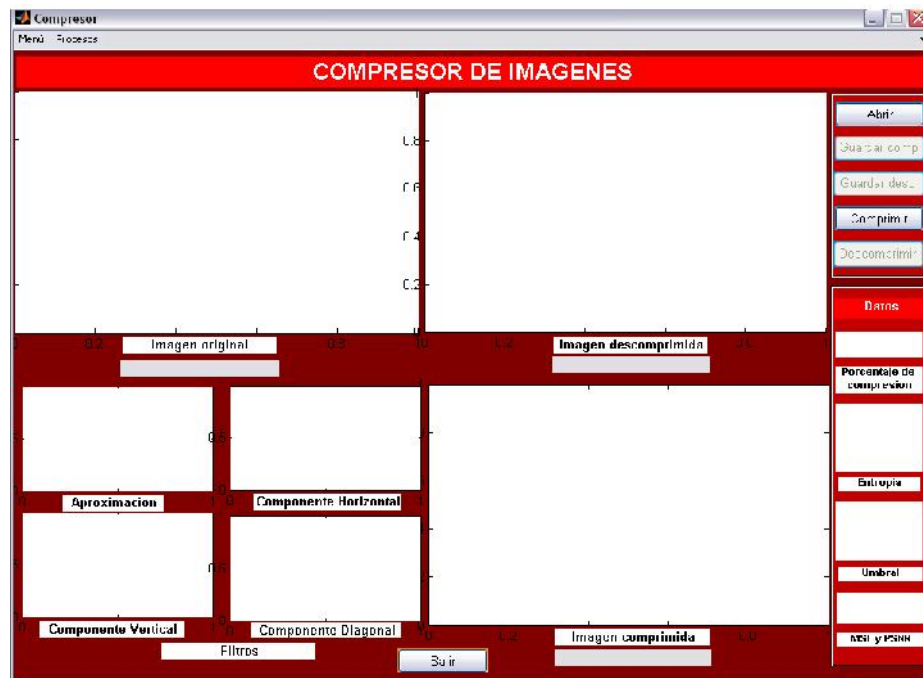


Figura 6.20 Interfaz gráfica ejecutada.

Elaborado por: Investigador

5. Para cargar una imagen se debe hacer clic en el botón **Abrir**, la imagen debe tener extensión .jpeg o jpg. (Figura 6.21).



Figura 6.21 Selección de la imagen que se va a abrir.

Elaborado por: Investigador

6. La imagen cargada aparecerá de la siguiente manera, como se muestra en la parte superior izquierda de la figura 6.22



Figura 6.22 Imagen original cargada.

Elaborado por: Investigador

7. Para realizar el proceso de compresión se debe hacer clic en el botón **Comprimir**, obteniendo como resultado las imágenes mostradas en la figura 6.23.



Figura 6.23 Proceso de compresión.

Elaborado por: Investigador

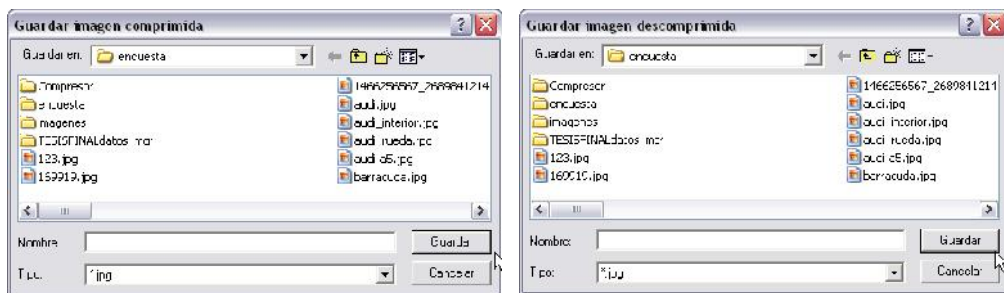
- Para realizar el proceso de descompresión se debe hacer clic en el botón **Descomprimir**, finalizando así el proceso, obteniendo la imagen reconstruida (parte superior derecha de la figura 6.24).



Figura 6.24 Proceso de descompresión.

Elaborado por: Investigador

- Para guardar las imágenes comprimida y descomprimida hay que hacer clic en los botones **Guardar comp** y **Guardar desc** respectivamente. (Figura 6.25).



(a)

(b)

Figura 6.25 Guardar imagen (a) Imagen comprimida (b) Imagen descomprimida.

Elaborado por: Investigador

10. Al hacer clic en el botón **Salir** aparecerá el siguiente mensaje: (Figura 6.26).



Figura 6.26 Salir de la aplicación.

Elaborado por: Investigador

Si se elige la opción **Si**, la aplicación y las ventanas que se generaron al correr la aplicación se cerrarán, de lo contrario, al elegir la opción **No**, la aplicación no se cerrará y se podrá seguir trabajando con la misma.

11. Se creó también dos menús de opciones que realizan las mismas funciones de los botones de la aplicación “Compresor”. (Figura 6.27).



Figura 6.27 Menús de opciones.

Elaborado por: Investigador

6.7.4. Análisis e interpretación de resultados.

6.7.4.1. Resultados de la prueba objetiva.

Una vez realizada la encuesta (Anexo D) a los 20 observadores, se calculan los valores medios, en función del MOST TEST que se codifica con valores de 1 a 5

(desde muy molesto a imperceptible) las diferencias entre la imagen original y la imagen descomprimida.

Entonces para un intervalo de confianza del 100%, los resultados obtenidos tras el procesamiento de los datos de la prueba subjetiva se resumen en la tabla 6.3.

Tabla 6.3 Encuesta.

Elaborado por: Investigador

Pregunta	Imperceptible (5)	Perceptible pero no molesta (4)	Ligeramente molesta (3)	Molesta (2)	Muy Molesta (1)
¿Cuál es su criterio sobre la diferencia entre el archivo descomprimido con el archivo original?	15	5			

La puntuación de opinión media se calcula con la siguiente fórmula:

$$\hat{p} = \frac{a}{n} = \bar{x}$$

Dónde:

$a \rightarrow$ es el número de unidades de la muestra que poseen la característica a observar.

$n \rightarrow$ Número de unidades de la población o muestra.

$\hat{p} \rightarrow$ Proporción de unidades en la muestra que poseen la característica a estudiar.

Con los resultados de la encuesta descrita en la tabla 6.3, se determinaron los siguientes valores:

Tabla 6.4 Resultados encuesta MOS TEST.

Elaborado por: Investigador

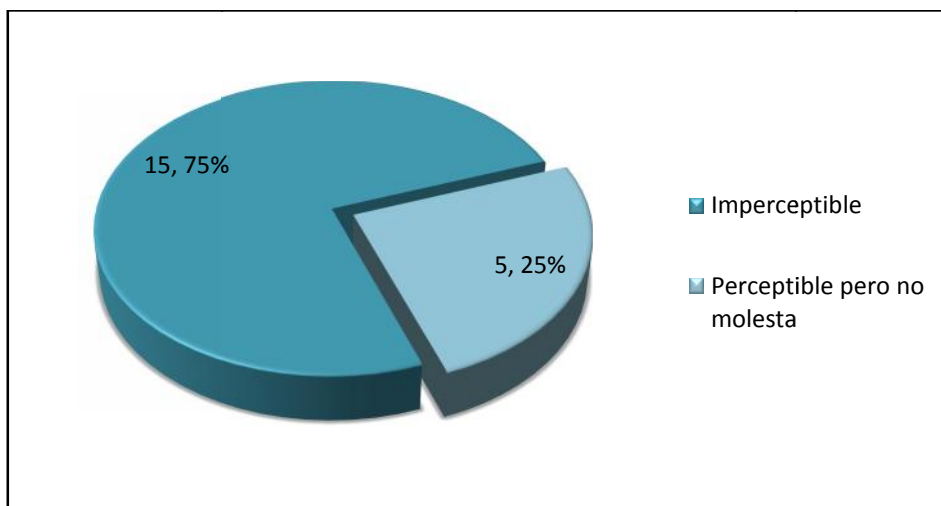
Nota	Frecuencia	Porcentaje (%)
Imperceptible	15	75
Perceptible pero no molesta	5	25
	20	100

Evaluación de la Calidad de las imágenes con respuesta Imperceptible que responden 15 observadores.

$$\hat{p} = \frac{a}{n} = \frac{15}{20} = 0.75 = 75\%$$

Evaluación de la Calidad de las imágenes con respuesta Perceptible pero no molesta que responden 5 observadores.

$$\hat{p} = \frac{a}{n} = \frac{5}{20} = 0.25 = 25\%$$



Graf. 6.1 Resultados MOS TEST

Elaborado por: Investigador

6.7.4.2. Resultados de las pruebas subjetivas.

6.7.4.2.1. Análisis del proceso de compresión – descompresión de imágenes de extensión .jpeg

Para analizar el proceso de compresión – descompresión, se repitió el proceso para 20 imágenes (Anexo E) desde la aplicación realizada en Guide – Matlab, obteniendo los siguientes resultados:

Tabla 6.5 Resultados – proceso de compresión/descompresión

Elaborado por: Investigador

NOMBRE DE LA IMAGEN	Porcentaje Relativo de compresión (%)	Porcentaje Total de compresión (%)	MSE	PSNR (dB)
audi	46,851	41,234	0,108	57,794
audi interior	56,638	55,241	0,164	55,994
audi_rueda	43,133	38,216	0,101	58,075
audi-a5	60,953	57,812	0,188	55,381
barracuda	48,519	40,910	0,199	55,136
BMW	65,819	60,475	0,123	57,249
BMW_M3	57,299	54,202	0,188	55,391
bmw-i3-i8-series-045	43,912	40,996	0,138	56,726
casca	60,294	58,126	0,195	55,231
compu	64,866	38,472	0,099	58,189
corvette_turn_signal_mirror_lrg	59,342	58,129	0,171	55,808
dj-school	67,726	64,536	0,193	55,272
ferrari_aliante_07	52,725	43,355	0,159	56,105
ferrari-340-competizione-xl	48,671	44,259	0,220	54,711
Ferrari-612-GTO-3-lg	47,316	44,069	0,160	56,082
flash	61,665	27,041	0,058	60,479
helicoptero	63,715	58,035	0,191	55,319

Continuación de la Tabla 6.5

mercedes-sls-amg-gt3	56,409	45,652	0,143	56,574
pomo	55,620	40,679	0,129	57,015
viudas	60,604	59,635	0,193	55,271
DESVIACION TIPICA	7,554	10,060	0,042	1,416
PROMEDIO	56,304	49,352	0,156	56,390

De acuerdo con los resultados obtenidos al calcular el promedio y la desviación típica de los datos, se obtienen valores aproximados de: 58% para el porcentaje relativo de compresión de las imágenes con una desviación típica de 8, y un porcentaje total de 50% con una desviación típica de 10; por consiguiente, se cumple con el porcentaje de compresión para el primer nivel de compresión de la DWT.

Los valores aproximados obtenidos para la descompresión son: 0.156 para el Error Cuadrático Medio (MSE) con una desviación típica de 0.042, y una relación señal pico a ruido (PSNR) de 56.390 dB con una desviación típica de 1.416; esto indica que hay una buena relación de señal pico a ruido, superando los valores típicos de PSNR que están en el rango de 30 a 50 dB, y por lo tanto, la calidad de las imágenes es buena.

6.7.4.2.2. Análisis del proceso de transmisión de imágenes de extensión .jpeg.

Para analizar el proceso de transmisión, se utilizó el software libre HFS (HTTP File Server) versión 2.2f (Figura 6.28), mediante el cual se crea un servidor gratuito (Figura 6.29) donde se pueden compartir archivos entre usuarios, en este caso, se creó dos usuarios en dos ordenadores conectados en red vía inalámbrica.

El proceso fue realizado para las 20 imágenes que pasaron el proceso de compresión – descompresión, incluyendo sus respectivas imágenes comprimidas y descomprimidas (en total 60 imágenes), obteniendo los siguientes resultados:

Tabla 6.6 Resultados – proceso de compresión/descompresión

Elaborado por: Investigador

Imagen	Tamaño (kB)	Tiempo (seg)	Velocidad (kbps)
audi	55,24	0,29	1203
comprimida	16,55	0,09	546
descomprimida	55,20	0,29	1823
audi interior	95,09	0,50	1545
comprimida	22,45	0,12	364
descomprimida	70,57	0,37	2330
audi_rueda	75,99	0,40	1235
comprimida	16,23	0,09	536
descomprimida	50,16	0,27	1092
audi-a5	206,95	1,10	1502
comprimida	31,54	0,17	1009
descomprimida	117,98	0,63	2570
barracuda	36,45	0,19	298
comprimida	9,43	0,05	643
descomprimida	32,55	0,17	1075
BMW	159,73	0,85	3480
comprimida	31,69	0,17	1014
descomprimida	115,69	0,61	3821
BMW_M3	70,09	0,37	1139
comprimida	18,81	0,10	621
descomprimida	69,32	0,37	1144
bmw-i3-i8-series-045	109,26	0,58	1434
comprimida	18,05	0,10	596
descomprimida	56,75	0,30	1236
casada	499,63	2,65	761
comprimida	89,49	0,47	1454
descomprimida	383,92	2,04	680
compu	13,78	0,07	300
comprimida	4,30	0,02	142
descomprimida	13,73	0,07	878

Continuación de la Tabla 6.6

corvette_turn_signal_mirror_lrg	63,29	0,34	1378
comprimida	9,52	0,05	207
descomprimida	33,85	0,18	1118
dj-school	210,04	1,11	625
comprimida	11,31	0,06	723
descomprimida	43,52	0,23	1437
ferrari_aliante_07	64,71	0,34	1068
comprimida	8,08	0,04	266
descomprimida	26,98	0,14	891
ferrari-340-competizione-xl	136,78	0,73	897
comprimida	18,41	0,10	1178
descomprimida	72,84	0,39	684
Ferrari-612-GTO-3-lg	60,93	0,32	1006
comprimida	15,55	0,08	1061
descomprimida	49,06	0,26	1068
Flash	127,67	0,68	4085
comprimida	20,13	0,11	644
descomprimida	65,66	0,35	4202
Helicóptero	51,12	0,27	1113
comprimida	9,87	0,05	673
descomprimida	39,06	0,21	1290
mercedes-sls-amg-gt3	224,19	1,19	2106
comprimida	57,23	0,30	1246
descomprimida	213,26	1,13	2799
Pomo	92,83	0,49	1508
comprimida	4,15	0,02	132
descomprimida	12,71	0,07	419
Viudas	61,26	0,33	2023
comprimida	18,25	0,10	602
descomprimida	61,24	0,32	3919
Promedio imagen original		0,64	1435,30
Promedio imagen comprimida		0,11	682,85
Promedio imagen descomprimida		0,42	1723,80

La velocidad de descarga de las imágenes depende del tamaño del archivo y del ancho de banda del proveedor de servicios de internet, en esta ocasión, 1 Mbps en ADSL. La velocidad también depende de problemas de hardware como: memoria,

conflictos con el navegador, problemas con otras aplicaciones instaladas o hasta debido a la actividad virus.

De acuerdo con los resultados obtenidos al calcular el tiempo de transmisión de las imágenes, se obtienen valores promedio de: 0.64 segundos para las imágenes originales, 0.11 segundos para las imágenes comprimidas y 0.42 segundos para las imágenes descomprimidas; logrando tiempos de transmisión reducidos comparando entre la imagen original y la imagen descomprimida.

Los valores promedio obtenidos para la velocidad de descarga de las imágenes son: 1435.30 kbps para las imágenes originales, 682.85 kbps para las imágenes comprimidas y 1723.80 kbps para las imágenes descomprimidas; alcanzando velocidades de transmisión mayores para las imágenes descomprimidas.

CAPÍTULO VII

7 CONCLUSIONES Y RECOMENDACIONES.

7.1. Conclusiones.

- La calidad de las imágenes de extensión .jpeg se mantuvo debido a que las representaciones mediante Wavelets se aproximan a la sensibilidad visual del humano.
- La información contenida en las componentes Wavelets que pasaron por filtros pasa altos poseen gran cantidad de ceros, permitiendo realizar la codificación con pocos bits.
- Al obtener menores tiempos y mayores velocidades de transmisión de las imágenes de extensión .jpeg, se mejora la transmisión luego de realizado el proceso de compresión – descompresión.

7.2. Recomendaciones.

- El uso del presente proyecto en biblioteca servirá como manual de guía para profundizar en el estudio de las transformadas Wavelets y sus diferentes aplicaciones.
- Analizar los distintos tipos de DWT y realizar pruebas de compresión a distintos niveles.
- Elegir el medio adecuado para realizar la transmisión de las imágenes y no tener inconvenientes en la obtención de resultados.

Bibliografía

Libros.

CARNE, E. Bryan. (1999). *Telecommunications Primer: Data, Voice And Video Communications*(2ed). Upper Saddle River, New Jersey: Prentice-Hall PTR

COUCH, Leon.(1998). *Sistemas de Comunicación Digitales y Analógicos* (5 ed.). México: Prentice Hall

GONZÁLEZ, Rafael C. &WOODS, Richard E. (1992). *Digital Image Processing*.Boston – USA: Addison-Wesley Publishing Company, Inc

QUEZADA, G. (1994). *Diseño y evaluación de proyectos*. Loja - Ecuador: Editorial Universidad Técnica Particular de Loja.

SOLOMON, C. & Breckon, T. (2011). *Fundamentals of Digital Image Processing*. United Kingdom: John Wiley & Sons Ltd.

WALKER, J.S. (1999), *A Primer on Wavelets and Their Scientific Applications*. Boca Raton – Londres – New York – Washington D.C. : Chapman & Hall/CRC

Enlaces.

Kioskea.net

Recuperada en octubre 16, 2008 desde:

<http://es.kioskea.net/contents/video/compress.php3>

Wikipedia.org

Recuperada en octubre 5, 2011 desde:

http://es.wikipedia.org/wiki/Compresi%C3%B3n_de_datos

Wikipedia.org

Recuperada en julio 7, 2011 desde:

http://es.wikipedia.org/wiki/Transmisi%C3%B3n_de_datos

Galeon.com

Recuperada en junio, 2011 desde:

<http://www.galeon.com/odiseus/info/algorithm.htm>

Lexandin.nl

Recuperada en Agosto 10, 1992

http://www.lexadin.nl/wlg/legis/nofr/oeur/arch/ecu/Ley_Teleco_reforma.pdf

Kioskea.net

Recuperada en octubre 16, 2008 desde:

<http://es.kioskea.net/contents/transmission/transintro.php3>

Wikipedia.org

Recuperada en octubre 10, 2011 desde:

http://es.wikipedia.org/wiki/Transmisi%C3%B3n_de_datos

Uaq.mx

Recuperada en septiembre 21, 1999

<http://www.uaq.mx/matematicas/estadisticas/xu5.html>

CIRIA UDLAP

Recuperada en diciembre 6, 2004

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/perez_r_oa/capitulo2.pdf

ANEXOS

ANEXO A

ENCUESTA.

La encuesta está dirigida a los docentes de la carrera de Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, realizada con preguntas de carácter general y fáciles de entender, por lo tanto también fáciles de responder.

El esquema utilizado es el siguiente:

UNIVERSIDAD TÉCNICA DE AMBATO
F.I.S.E.I.
ELECTRÓNICA Y COMUNICACIONES
ENCUESTA

Objetivo: Con el fin de evaluar la factibilidad de la realización de un compresor de imágenes de extensión .jpeg, usted tendrá que responder el siguiente cuestionario.

Instructivo: Lea detenidamente cada pregunta y marque su respuesta con una X.

1. ¿Conoce usted acerca del formato de imágenes de extensión .jpeg?

SI () NO ()

2. ¿Conoce usted que es un compresor de imágenes?

SI () NO ()

3. ¿Cree usted que se mantendrá la calidad de las imágenes de extensión .jpeg luego de descomprimirlas?

SI () NO ()

4. ¿Se optimizará el almacenamiento de las imágenes de extensión .jpeg al comprimirlas?

SI () NO ()

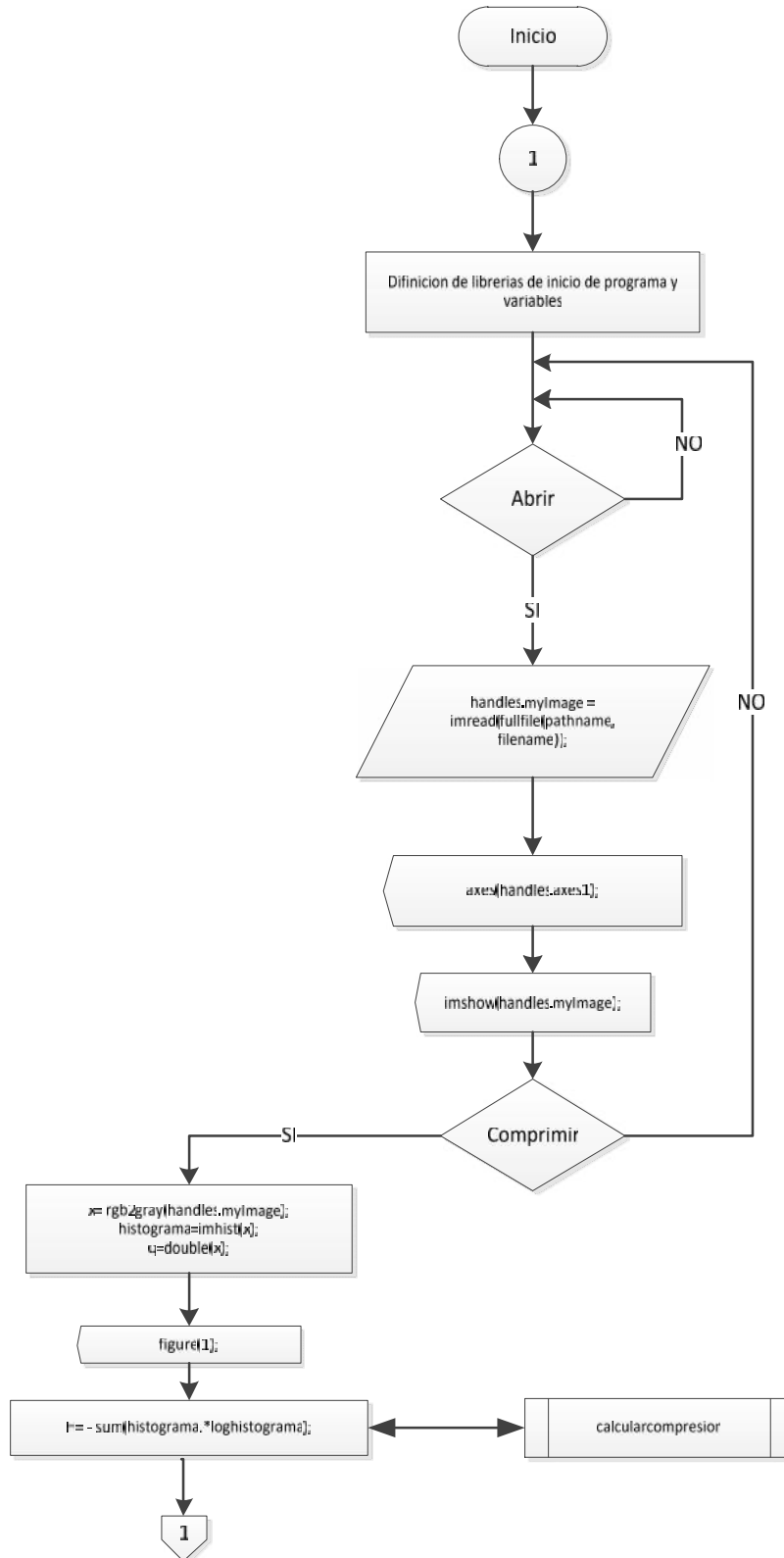
5. ¿Cree usted que se mejorará la transmisión de imágenes de extensión .jpeg al comprimirlas?

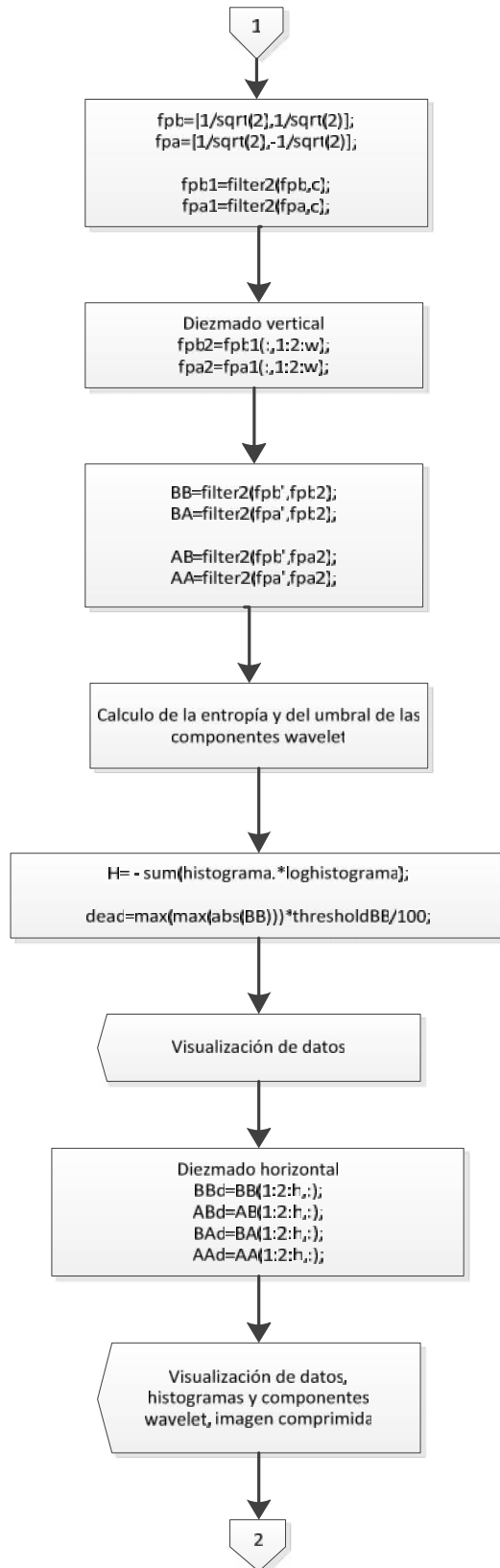
SI () NO ()

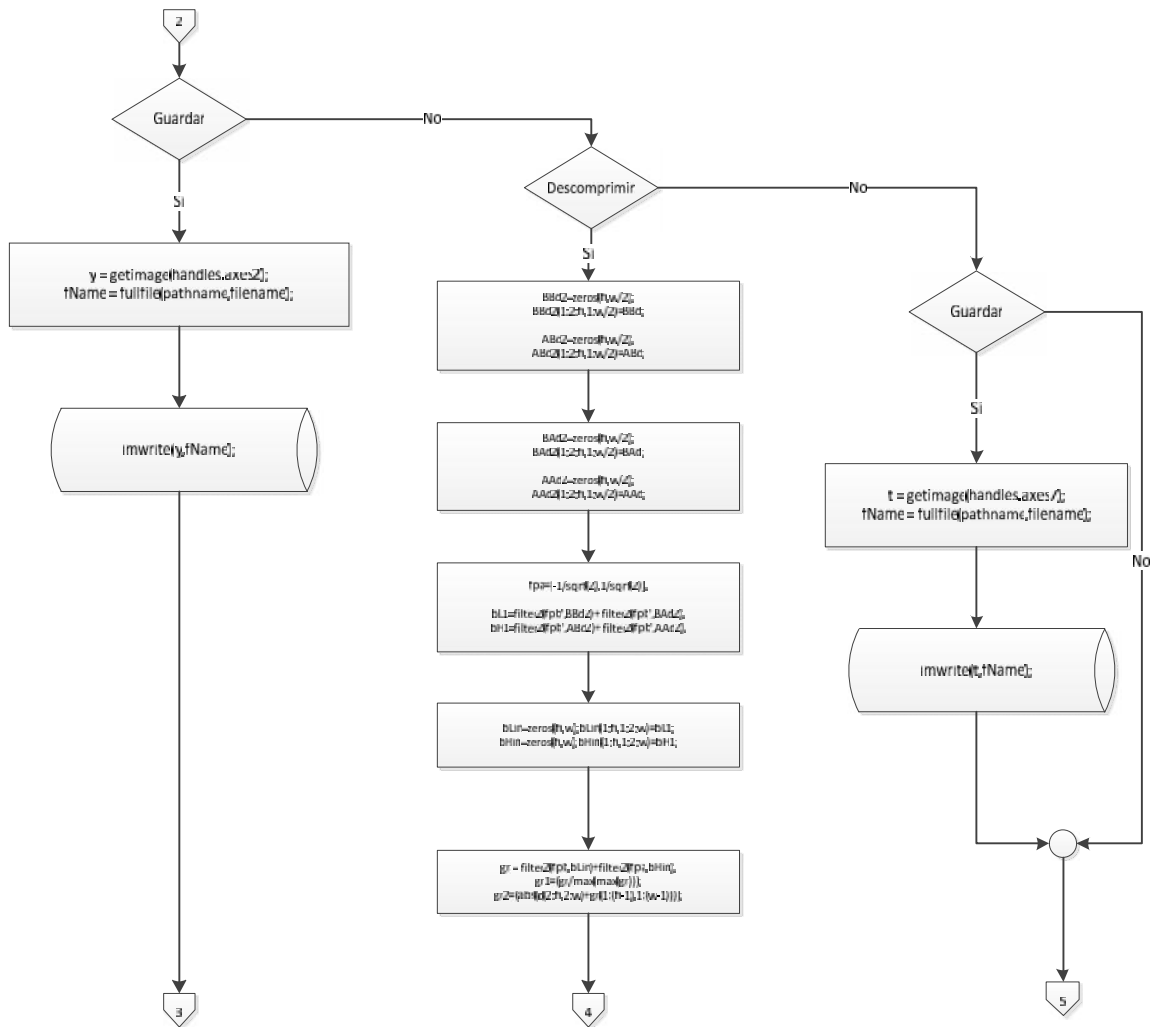
¡Gracias por su colaboración!

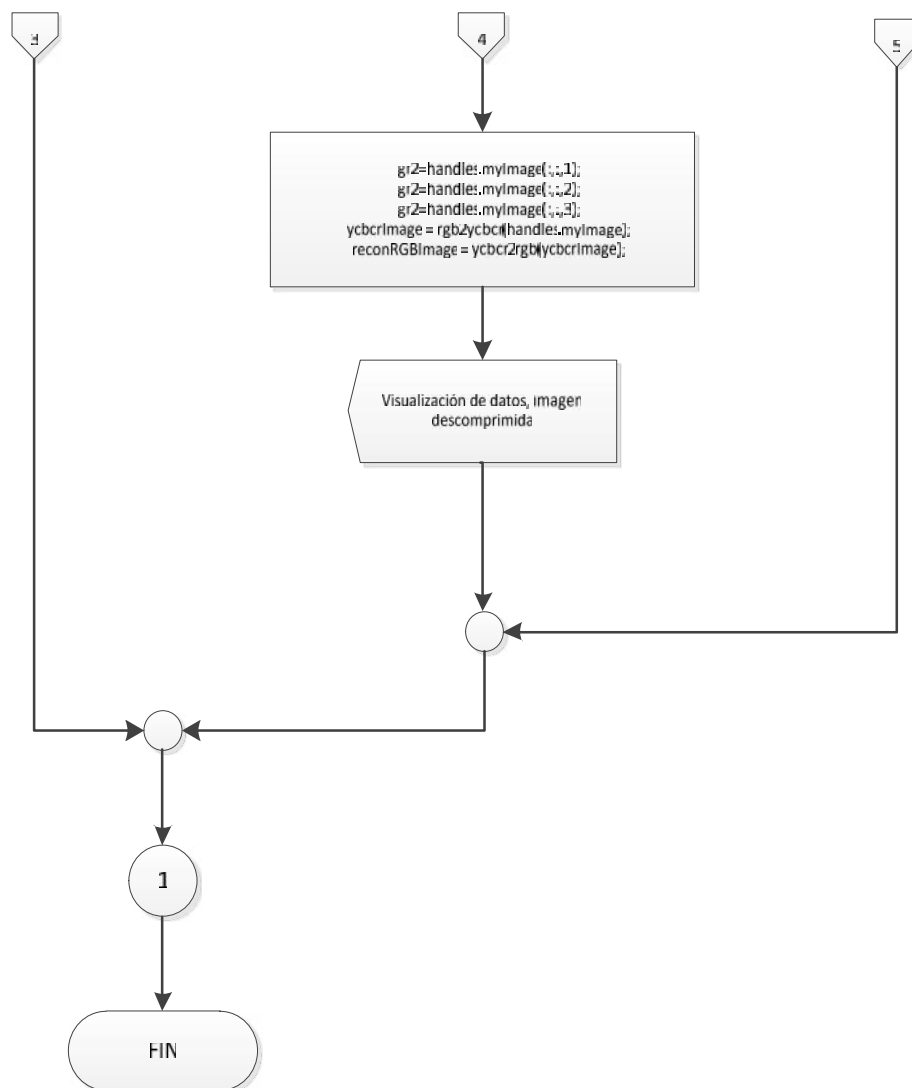
ANEXO B

FLUJOGRAMA









ANEXO C

Código del programa “Compresor”

```
function varargout = Compresor(varargin)

% Código de inicialización - NO EDITAR

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @Compresor_OpeningFcn, ...
'gui_OutputFcn',  @Compresor_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);

If nargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
Else
    gui_mainfcn(gui_State, varargin{:});
end

% Fin del código de inicialización- NO EDITAR

Function Compresor_OpeningFcn(hObject, eventdata, handles,
varargin)

clc;      % limpia la ventana de comandos

imtool closeall; % Cierra todas las figuras creadas por imtool.

handles.output = hObject;% se guardan las variables de salida en
la variable hObject

%Agregar variables
handles.myImage = [];%Variable que guarda la imagen original

guidata(hObject, handles);% se guardan las variables que van a ser
usadas en todo el programa.

% -----Habilitación/deshabilitación de los botones-----
set(handles.Abrir, 'Enable', 'on');
set(handles.comprimir, 'Enable', 'on');
set(handles.guardar, 'Enable', 'off');
```

```

set(handles.descomprimir,'Enable','off');
set(handles.salir,'Enable','on');
set(handles.guardardsc,'Enable','off');

% Las salidas de esta función son retornadas a la línea de
comandos.

% Función de salida del programa Compresor

Function varargout = Compresor_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;% variables de salida

% ----- Botón Abrir -----

function Abrir_Callback(hObject, eventdata, handles)

clc %limpia la pantalla

try
    [filename,pathname] = uigetfile('*.jpg','Selecciona imagen
para abrir');% seleccionar el directorio donde se abrirá la
imagen.

If isequal(filename,0)

else

handles.myImage = imread(fullfile(pathname, filename));% leer la
imagen desde el directorio seleccionado.

[d,e,f]=size(handles.myImage); %calcular el tamaño de la imagen
original

set(handles.text17,'String',{'size = ',num2str([d,e,f])}); %
mostrar el tamaño de la imagen en un cuadro de texto.

axes(handles.axes1); % visualizar la imagen en un axes
axis off; % deshabilita el axes
imshow(handles.myImage); % muestra la imagen original en axes1

end
guidata(hObject, handles); % se guardan las variables que van a
ser usadas en todo el programa.

catch
msgbox('Error') % aparece un error si el archivo cargado no es una
imagen .jpeg
end

% ----- Botón Comprimir -----

Function comprimir_Callback(hObject, eventdata, handles)

```

```

global x q h w BBd ABd BAd AAd fpb fpa c; % variables globales

%-----convertir la imagen a escala de grises-----
x= rgb2gray(handles.myImage);

% -----histograma-----
histograma=imhist(x);
q=double(x); % transformar la imagen de escala de grises a double
figure(1); % se muestra el histograma en una ventana

% datos de la compresión:

% -----entropía-----
histograma=histograma/sum(histograma);
histograma(find(histograma==0))=1; % busca valores en el rango de
0 a 1 dentro del histograma
loghistograma= log2(histograma);
H= - sum(histograma.*loghistograma); % fórmula de la entropía

% -----calcular la compresión-----

[huffOriginal,tamHuffOriginal]=calcularcompresion(x);
c=q/(max(max(q))); %normalizar la matriz de la imagen
z=c;

%----- Primera etapa de filtros -----
fpb=[1/sqrt(2),1/sqrt(2)]; %filtro pasa bajos
fpa=[1/sqrt(2),-1/sqrt(2)];%filtro pasa alto

fpb1=filter2(fpb,c);%aplicación del filtro pasa bajos a la imagen
fpa1=filter2(fpa,c); %aplicación del filtro pasa alto a la imagen

% ----- tamaño de la imagen -----
[h,w]=size(x);
tam=h*w*8;

%----- Segunda etapa de filtros -----
% Se eliminan la mitad de las columnas

fpb2=fpb1(:,1:2:w);
fpa2=fpa1(:,1:2:w);

%-----obtención de las componentes Wavelet-----
BB=filter2(fpb',fpb2); % Obtención de la aproximación
BA=filter2(fpa',fpb2); % Obtención de la componente horizontal

AB=filter2(fpb',fpa2); % Obtención de la componente vertical
AA=filter2(fpa',fpa2); % Obtención de la componente diagonal

```



```

% -----entropía de las componentes wavelet-----

% -----cálculo de la entropía de la aproximación-----
histoapx=imhist(BB);
histoapx=histoapx/sum(histoapx); % normalización del histograma
histoapx(find(histoapx==0))=1; % busca valores en el rango de 0 a
1 dentro del histograma
loghistoapx= log2(histoapx);
Hapx= - sum(histoapx.*loghistoapx);% fórmula de la entropía

% -----cálculo del umbral de la aproximación-----
thresholdBB=8/Hapx;%cálculo del threshold
deadapx=max(max(abs(BB)))*thresholdBB/100; % cálculo de la zona
muerta
dyA1= BB;
index=find(abs(dyA1)<=deadapx); % buscar valores menores a la
zona muerta
dyA1(index)=zeros(size(index));%relleno de ceros
density1 = nnz(dyA1); %cálculo de la densidad

% -----cálculo de la entropía de la componente horizontal-----
histoh=imhist(BA);
histoh=histoh/sum(histoh); % normalización del histograma
histoh(find(histoh==0))=1; % busca valores en el rango de 0 a 1
dentro del histograma
loghistoh= log2(histoh);
Hh = - sum(histoh.*loghistoh);% fórmula de la entropía

% -----cálculo del umbral de la componente horizontal-----
thresholdBA=8/Hh; ;%cálculo del threshold
deadH=max(max(abs(BA)))*thresholdBA/100; % calculo de la zona
muerta
dyh1= BA;
index=find(abs(dyh1)<=deadH); % buscar valores menores a la zona
muerta
dyh1(index)=zeros(size(index)); %relleno de ceros
density2= nnz(dyh1); %cálculo de la densidad

% -----cálculo de la entropía de la componente vertical-----
histov=imhist(AB);
histov=histov/sum(histov); % normalización del histograma
histov(find(histov==0))=1; % busca valores en el rango de 0 a 1
dentro del histograma
loghistov= log2(histov);
Hv = - sum(histov.*loghistov);% fórmula de la entropía

% -----cálculo del umbral de la componente vertical-----
thresholdAB=8/Hv; ;%cálculo del threshold
deadV=max(max(abs(AB)))*thresholdAB/100; % calculo de la zona
muerta
dyv1= AB;

```

```

index=find(abs(dyv1)<=deadV); % buscar valores menores a la zona
muerta
dyv1(index)=zeros(size(index)); %relleno de ceros
density3= nnz(dyv1); %cálculo de la densidad

% -----cálculo de la entropía de la componente diagonal-----
histod=imhist(AA);
histod=histod/sum(histod); % normalización del histograma
histod(find(histod==0))=1; % busca valores en el rango de 0 a 1
dentro del histograma
loghistod= log2(histod);
Hd = - sum(histod.*loghistod);% fórmula de la entropía

% -----cálculo del umbral de la componente diagonal-----
thresholdAA=8/Hd; ;%cálculo del threshold
deadD=max(max(abs(AA)))*thresholdAA/100; % calculo de la zona
muerta
dyd1= AA;
index=find(abs(dyd1)<=deadD); % buscar valores menores a la zona
muerta
dyd1(index)=zeros(size(index)); %relleno de ceros
density4 = nnz(dyd1); %cálculo de la densidad

% -----visualización de la entropía y del umbral-----

set(handles.text20,'String',{'Horig = ',num2str(H)};['Haprox =
',num2str(Hapx)};['Hhoriz = ',num2str(Hh)};['Hvert =
',num2str(Hv)};['Hdiag = ',num2str(Hd)}]);
set(handles.text22,'String',{'Uapx = ',num2str(deadapx)};['Uh =
',num2str(deadH)};['Uv = ',num2str(deadV)};['Ud =
',num2str(deadD)}]);% los datos se visualizan en un cuadro de
texto

%-----densidad de la imagen comprimida-----
density = density1+density2+density3+density4

% ----- diezmando horizontal -----
% Elimina la mitad de las filas
BBd=BB(1:2:h,:);
ABd=AB(1:2:h,:);
BA d=BA(1:2:h,:);
AA d=AA(1:2:h,:);

%----- normalización de la imagen-----
BBdn=(BBd/max(max(BBd)));

% -----tamaño de la imagen-----
[a,l,p]=size(BBd);
tamano=a*1*8;

```

```

% -----visualización del tamaño de la imagen-----
set(handles.text19,'String',{'size = ',num2str([a,l,p])});
% los datos se visualizan en un cuadro de texto

% -----visualización de los histogramas-----

        figure(14);
subplot(2,2,1);

        [pbpb,tamBB]=calcularcompresion(uint16((BBd+1)*127));
subplot(2,2,2);

        [pbpa,tamBA]=calcularcompresion(uint16((BAd+1)*127));
subplot(2,2,3);

        [papb,tamAB]=calcularcompresion(uint16((ABd+1)*127));
subplot(2,2,4);

        [papa,tamAA]=calcularcompresion(uint16((AAd+1)*127));

% -----cálculo de los porcentajes de compresión-----
sumTam=a*1*8*4;
sumaTamanios=tamBB+tamBA+tamAB+tamAA;
porcentajeRelativo=sumaTamanios*100/tamHuffOriginal;
PorcentajeTotal=sumaTamanios*100/tam;

% -----visualización de los porcentajes de compresión-----
set(handles.salida,'String',{'PR =
',num2str(porcentajeRelativo),'%'];['PT =
',num2str(PorcentajeTotal),'%']});% los datos se visualizan en un
cuadro de texto

% -----visualización de las componentes wavelets en los axes-----

axes(handles.axes2);
axisoff;
imshow(BBdn);
axes(handles.axes3);
axisoff;
imshow(BBd);
axes(handles.axes4);
axisoff;
imshow(BAd);
axes(handles.axes5);
axisoff;
imshow(ABd);
axes(handles.axes6);
axisoff;
imshow(AAd);

guidata(hObject, handles); % se guardan las variables que van a
ser usadas en todo el programa.

```

```

% Habilitación de los botones

set(handles.guardar, 'Enable', 'on');
set(handles.descomprimir, 'Enable', 'on');
set(handles.guardardsc, 'Enable', 'on');

% ----- Botón Guardar -----

Function guardar_Callback(hObject, eventdata, handles)

try
y = getimage(handles.axes2); % obtener la imagen guardada en axes2

if isempty(y), return, end % retornar la variable "y".

    [filename,pathname] =
uiputfile('*.jpg', 'Guardarimagencomprimida'); % seleccionar el
directorio donde se guardará la imagen comprimida.

if filename==0, return, end

fName = fullfile(pathname, filename);
imwrite(y, fName); % guardar la imagen en el directorio seleccionado

% mensaje de error cuando no se da un nombre al archivo

catch
msgbox('Error')
end

guidata(hObject, handles); % se guardan las variables que van a
ser usadas en todo el programa.

Function descomprimir_Callback(hObject, eventdata, handles)

% habilitación de los botones

set(handles.guardar, 'Enable', 'on');
set(handles.descomprimir, 'Enable', 'off');
set(handles.guardardsc, 'Enable', 'on');

global x q h w BBd ABd BAd AAd fpb fpa c; % variables globales

% intercalación de zeros en la dirección vertical de las imágenes
iniciales

BBd2=zeros(h,w/2);
BBd2(1:2:h,1:w/2)=BBd;

ABd2=zeros(h,w/2);
ABd2(1:2:h,1:w/2)=ABd;

```

```

BAd2=zeros(h,w/2);
BAd2(1:2:h,1:w/2)=BAd;

AAd2=zeros(h,w/2);
AAd2(1:2:h,1:w/2)=AAd;

% -----filtrado de las imágenes resultantes-----

fpa=[-1/sqrt(2),1/sqrt(2)];

bL1=filter2(fpb',BBd2)+ filter2(fpb',BAd2);
bH1=filter2(fpb',ABd2)+ filter2(fpb',AAd2);

% Intercalación de ceros en la dirección horizontal a cada una de
las imágenes

bLin=zeros(h,w);bLin(1:h,1:2:w)=bL1;
bHin=zeros(h,w);bHin(1:h,1:2:w)=bH1;

% --filtrado y combinación de las imágenes para obtener la imagen
descomprimida-----

gr = filter2(fpb,bLin)+filter2(fpa,bHin);
gr1=(gr/max(max(gr)));
gr2=(abs(c(2:h,2:w)+gr(1:(h-1),1:(w-1)))));

% ---conversión de gray a RGB-----
gr2=handles.myImage(:,:,1);
    gr2=handles.myImage(:,:,2);
    gr2=handles.myImage(:,:,3);
ycbcrImage = rgb2ycbcr(handles.myImage);
reconRGBImage = ycbcr2rgb(ycbcrImage);

%tamaño de la imagen
[al,an,pr]=size(reconRGBImage);
tam=al*an*8;

% ----visualización del tamaño de la imagen descomprimida-----
set(handles.text18,'String',{'size = ',num2str([al,an,pr])});
imwrite(gr2,'imdescgray.jpg','jpg');

% ----visualización de la imagen descomprimida en un axes-----
axes(handles.axes7);
axisoff;
imshow(reconRGBImage);

%-----cálculo del MSE y PSNR -----
error = handles.myImage - reconRGBImage;
MSE = sum(sum(error .* error)) / (h * w);
MSER=MSE(:,:,1);
MSEG=MSE(:,:,2);
MSEB=MSE(:,:,3);
MSEt=(MSER+MSEG+MSEB)/3;
psnr = 20*log10((255)/sqrt(MSEt));

```

```

% ----visualización del MSE y PSNR-----
set(handles.text21,'String',{['MSE = ',num2str(MSEt)];['PSNR = ',num2str(psnr)]});

guidata(hObject, handles); % se guardan las variables que van a
ser usadas en todo el programa.

% -----Botón salir-----

Function salir_Callback(hObject, eventdata, handles)

opc=questdlg('¿Desea salir del programa?','SALIR','Si','No','No');
if strcmp(opc,'No')

return;
end

clear,clc,close all% limpia la pantalla y cierra las figuras
creadas

close (gcbf)% cerrar las figuras

% -----creación del menú-----

Function menu_Callback(hObject, eventdata, handles)

% -----Botón guardar-----
function guardardsc_Callback(hObject, eventdata, handles)

try
t = getimage(handles.axes7); % obtener la imagen guardada en axe7

if isempty(t), return, end% retornar la variable "t".
[filename,pathname] = uiputfile('*.jpg','Guardar imagen
descomprimida'); % seleccionar el directorio donde se guardará la
imagen comprimida.

if filename==0, return, end
fName = fullfile(pathname,filename);
imwrite(t,fName); %guardar la imagen en el directorio seleccionado

% mensaje de error cuando no se da un nombre al archivo
catch
msgbox('Error')
end

guidata(hObject, handles); % se guardan las variables que van a
ser usadas en todo el programa.

% -----creación del menu procesos-----

Function procesos_Callback(hObject, eventdata, handles)

```

ANEXO D

ENCUESTA MOS TEST

UNIVERSIDAD TECNICA DE AMBATO

F.I.S.E.I.

ELECTRÓNICA Y COMUNICACIONES

ENCUESTA SOBRE LA CALIDAD DE LAS IMÁGENES LUEGO DEL PROCESO DE DESCOMPRESIÓN.

Objetivo: Con el fin de evaluar la calidad de las imágenes luego de comprimir las, ud. tendrá que responder el siguiente cuestionario en una escala de Imperceptible a Muy molesta en base a la diferencia entre la imagen original y la imagen descomprimida.

Instructivo:

- Observe la imagen detenidamente para una mejor percepción de cambios en la imagen descomprimida.
- Marque con una X en la casilla correspondiente de acuerdo a su criterio.
- Realice una sola marca de acuerdo a la escala, de lo contrario la puntuación no será válida.

Pregunta	Imperceptible (5)	Perceptible pero no molesta (4)	Ligeramente molesta (3)	Molesta (2)	Muy Molesta (1)
¿Cuál es su criterio sobre la diferencia entre el archivo descomprimido con el archivo original?					

¡Gracias por su colaboración!

ANEXO E

IMÁGENES UTILIZADAS EN EL TRABAJO.



audi
1024 x 576



audi interior
800 x 500



audi_rueda
1024 x 683



audi-a5
1024 x 587



barcuda
640 x 480



BMW
900 x 601



BMW_M3
800 x 533



bmw-i3-i8-
series-045
944 x 667



cascada
1600 x 1200



compu
358 x 297



corvette_turn_si
gnal_mirror_lrg
500 x 375



dj-school
500 x 333



ferrari_aliant
e_07
600 x 403



ferrari-340-
competizione-xl
970 x 600



Ferrari-612-
GTO-3-lg
900 x 508



flash
950 x 713



helicoptero
480 x 360

mercedes-
sls-amg-gt3
1600 x 1067

pomo
376 x 278

viudas
640 x 480