
# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

## FACULTAD DE CIENCIAS

### CARRERA DE BIOFÍSICA

# RESOLUCIÓN DE IMÁGENES TOMOGRÁFICAS MEDIANTE MATLAB

**Trabajo de titulación**
Tipo: Trabajo Experimental

Presentado para optar el grado académico de:
**BIOFÍSICA**

**AUTORA:** XIMENA ABIGAIL VILLAGOMEZ HERRERA
**DIRECTORA:** Biof. VILMA NOHEMÍ YANCHAPANTA BASTIDAS

Riobamba – Ecuador

2020

Yo, Ximena Abigail Villagomez Herrera, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autora asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 11 de Junio del 2020.
**Ximena Abigail Villagomez Herrera**
**CI: 06064072496**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE CIENCIAS**

**CARRERA DE BIOFÍSICA**

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; Tipo Trabajo Experimental **RESOLUCIÓN DE IMÁGENES TOMOGRÁFICAS MEDIANTE MATLAB**, realizado por la señorita: XIMENA ABIGAIL VILLAGOMEZ HERRERA, ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

| | **FIRMA** | **FECHA** |
|---|---|---|
| Mat. Luis Marcelo Cortez Bonilla.<br>**PRESIDENTE DEL TRIBUNAL** | LUIS MARCELO CORTEZ BONILLA<br>Firmado digitalmente por LUIS MARCELO CORTEZ BONILLA<br>Nombre de reconocimiento (DN): c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION ECIBCE, l=QUITO, serialNumber=0000419075, cn=LUIS MARCELO CORTEZ BONILLA<br>Fecha: 2020.08.24 20:09:22 -05'00' | 2020-06-11 |
| Biof. Vilma Nohemí Yanchapanta Bastidas.<br>**DIRECTOR DE TRABAJO DE TITULACIÓN** | VILMA NOHEMI YANCHAPANTA BASTIDAS<br>Firmado digitalmente por VILMA NOHEMI YANCHAPANTA BASTIDAS<br>Fecha: 2020.08.24 11:38:35 -05'00' | 2020-06-11 |
| Dr. Richard Willians Pachacama Choca.<br>**MIEMBRO DEL TRIBUNAL** | RICHARD WILLIANS PACHACAMA CHOCA<br>Firmado digitalmente por RICHARD WILLIANS PACHACAMA CHOCA<br>Nombre de reconocimiento (DN): c=EC, serialNumber=0601921703, sn=PACHACAMA CHOCA, cn=RICHARD WILLIANS PACHACAMA CHOCA, 1.3.6.1.4.1.37442.10.4=0601921703, givenName=RICHARD WILLIANS, email=rpachacama@gmail.com, st=RIOBAMBA, l=CHIMBORAZO, ou=Certificado de Clase 2 de Persona Fisica EC (FIRMA) | 2020-06-11 |

## DEDICATORIA

El presente Trabajo de Titulación lo dedico con cariño, primero a Dios por darme la fuerza y sabiduría para poder concluir con mis estudios. A mis padres María Herrera y Pedro Villagomez, quienes con mucho sacrificio y amor durante mi niñez me han formado con buenos principios, valores, sentimientos y hábitos, por su apoyo incondicional que con sus palabras me han motivado a seguir adelante para poder culminar con mis metas.

Ximena.

## AGRADECIMIENTO

A Dios por bendecirme al permitirme estar junto a mi familia y amigos, darme la sabiduría y paciencia para finalizar con éxito mi meta anhelada. y por haber puesto en mi camino a personas muy buenas que han sido mi soporte y compañía durante todo el periodo académico.

A mis padres y hermanos, les doy gracias por el apoyo incondicional que me ha impulsado a salir adelante en los momentos más difíciles, enseñarme que con paciencia y dedicación todos mis sueños se pueden lograr y por estar conmigo en las buenas y en las malas.

Ximena.

**TABLA DE CONTENIDO**

**CAPÍTULO II**

**CAPÍTULO III**

**ÍNDICE DE TABLAS**

**ÍNDICE DE FIGURAS**

**ÍNDICE DE ANEXOS**

**RESUMEN**

El presente trabajo de titulación tuvo como objetivo realizar una simulación computacional mediante el lenguaje de programación de Matlab para mejorar la resolución de las imágenes obtenidas en Tomografía Computarizada. Para lo cual se empleó herramientas matemáticas como el análisis matricial, uso de programas computacionales que facilitan los cálculos matemáticos y el análisis comparativo basado en la visualización grafica de los resultados. Se comprobó la eficiencia de cada Método de filtrado empleados para obtener una restauración de las imágenes distorsionadas aproximada a la real pero no en su totalidad ya que cierta parte de información no se encuentra en la imagen de prueba. El rendimiento de cada uno de los filtros empleados se demostró mediante la evaluación de las pruebas experimentales e implementación de algoritmos matemáticos donde se procesan y analizan las imágenes correspondientes a una Tomografía craneal, los resultados que se obtuvieron de la aplicación de los algoritmos computacionales en base a un análisis matemático se visualizaron en el diseño de una Interfaz Gráfica para su respectiva evaluación. Se concluye que los resultados obtenidos de la aplicación de cada del filtro propuesto son excelentes en la restauración de imagen digitales, sin embargo, el filtro Inverso mostró ser el más eficaz en la detección de altas frecuencias o variaciones bruscas de los tonos de gris y evita la pérdida de información o detalles, lo cual permite obtener imágenes libres de ruido y de excelente calidad de resolución. Por lo tanto, el filtro Inverso es una excelente elección para el tratamiento o procesamiento de imágenes Tomográficas ya que genera resultados totalmente satisfactorios en cuanto a mantener una relación entre la resolución, nitidez y calidad de imagen. Se recomienda que se realice el cálculo de la relación señal-ruido para corroborar de forma teórica la efectividad del rendimiento cuantitativo de los diversos filtros implementados.

**Palabras claves:** <TOMOGRAFÍA COMPUTARIZADA>, <IMÁGENES DIGITALES>, <TRANSFORMADA INVERSA DE FOURIER>, <FILTROS ESPACIALES>, <MATLAB(SOFTWARE)>, <RESTAURACIÓN DE IMÁGENES>, <REDUCCIÓN DE ARTEFACTOS>, < SUAVIZADO DE IMÁGENES>

**ABSTRACT**

The objective of this titling work was to perform a computational simulation using the Matlab programming language to improve the resolution of the images obtained in Computed Tomography. For this, mathematical tools such as matrix analysis, the use of computer programs that facilitate mathematical calculations, and comparative analysis based on the graphic display of results were used. Verify the efficiency of each filter method used to obtain a restoration of the distorted images approximate to the real one but not in its entirety since some part of the information is not found in the test image. The performance of each of the filters used was demonstrated through the evaluation of experimental tests and the implementation of mathematical algorithms where the images corresponding to a cranial tomography are processed and analyzed, the results obtained from the application of computational algorithms in Based a mathematical analysis is visualized in the design of a Graphical Interface for its respective evaluation. It is concluded that the results obtained from the application of each application of the proposed filter are excellent in the restoration of the digital image, however, the reverse filter will be the most effective in detecting high frequencies or sudden variations in shades of gray and prevents the loss of information or details, allowing images to be obtained free of noise and with excellent resolution quality. Therefore, the Inverse filter is an excellent choice for the treatment or processing of tomographic images and that generates totally satisfactory results in terms of maintaining a relationship between resolution, sharpness and image quality. It is recommended that the calculation of the signal-to-noise ratio be performed to theoretically corroborate the determination of the quantitative performance of the various filters implemented.

**Keywords:** <COMPUTERIZED TOMOGRAPHY>, <DIGITAL IMAGES>, <REVERSE TRANSFORMED BY FOURIER>, <SPACE FILTERS>, <MATLAB (SOFTWARE)>, <IMAGE RESTORATION>, <ARTIFACTORY REDUCTION>, <IMAGE SMOOTHING

# INTRODUCCIÓN

En el presente Trabajo de Titulación se utilizó el lenguaje de programación de Matlab para realizar el procesamiento de imágenes médicas de tomografía Computariza y obtener un mejoramiento en la resolución, considerando que los distintos servicios de diagnóstico en la actualidad cuentan con imágenes en especial el área de Tomografía Computarizada digitalizada, debido a su facilidad y eficiencia para el almacenamiento, la transición y el procesamiento de datos que posterior a una técnica postprocesamiento que permite obtener una imagen digitalizada. Sin embargo, dichas Imágenes Digitalizadas se pueden ver afectadas por una serie de interferencias tanto externas como internas al sistema informático encargado del procesamiento de las imágenes.

Las Imágenes Tomográficas, son un conjunto de una infinidad de señales que se almacenan en un computador en forma de una matriz constituida por unidades o pixeles donde los perfiles de intensidad de señales están relacionados entre sí, dicho perfil de señales pueden ser imperfectos por las siguientes razones: 1: Interferencia Externas que desnaturalizan o cambian los datos del objeto estudiado, y 2: Interferencias o Perturbaciones Internas que se presentan en la señal durante la fase de detección, transición y digitalización de la información, las mismas que alteran la calidad de la señal. Estas dos razones se conocen como ruidos y artefactos que se encuentran ligados a la fase de digitalización de una señal y la reconstrucción tridimensional de los datos, que limitan la valoración patológica de un paciente después de una exploración Tomográfica.

En este trabajo de Titulación se desarrolla los antecedentes de acuerdo al tema que nos brinda un sustento en el objetivo planteado, planteamiento del problema, justificación donde se muestra la necesidad de obtener una imagen de buena calidad y los objetivos que se requieren para alcanzar este trabajo.

En el capítulo I, se expone el estudio de las pautas teóricas necesarias para el desarrollo del tema de trabajo de titulación, hace referencia a ciertas definiciones y métodos matemáticos que se emplean en las deducciones matemáticas de los distintos filtros, se realiza el estudio del formato de archivo gráfico de las imágenes procesadas y que se procesan, se detalla las causas y efectos de la acción de un agente externo que limitan la calidad de una imagen médica.

El capítulo II contiene el desarrollo detallado de los algoritmos y diagramas de bloque que se implementan en el procesamiento de imágenes mediante el software Matlab y las herramientas necesarias para el análisis de una imagen que se emplean en la tesis, además, se describe de forma

general el diseñó de la interfaz gráfica o GUIDE en Matlab para ejecutar programas mediante varias ventanas interactivas con el usuario de forma rápida y sencilla en su manejo.

El capítulo III describe principalmente la información de la imagen de prueba, se analiza y muestra los resultados obtenidos mediante la ejecución de las herramientas de la interfaz gráfica, previamente desarrollada de acuerdo al algoritmo descrito en el capítulo 2, posteriormente se hace una comparación con los diferentes filtros propuestos, y se compara los resultados obtenidos de la ejecución de todos los algoritmos propuestos.

Posteriormente, se describe de forma detalla las conclusiones y recomendaciones acerca del desarrollo del trabajo realizado y finalmente se muestra los anexos que fundamentan los resultados obtenidos en el capítulo III.

## ANTECEDENTES

En los procesos de Tomografía es difícil tener una señal eléctrica pura, ya que en algunos casos se afecta por presencia de ruido que se entremezclan o distorsionan la información de la señal para esto es necesario contar con algun tipos de filtros empleados como una herramienta matemática para la reducción de ruido que presenta una imagen digital. Su funcionamiento se centra en transmitir o bloquear señales dentro de un cierto rango o intervalo de frecuencias, trabajando sobre una señal de entrada produciendo una señal de salida, cuyo contenido espectral es función tanto de la señal de entrada y el tipo de filtro ( Jiménez, 2007b).
Se basa en los distintos avances matemáticos demostrados hace 2 siglos, como las transformadas de Laplace, transformadas de Fourier y las funciones para resolver integrales elípticas de Carl Jacob Jacobi, los mismo que han servido para fundamentar la teoría del uso de los filtros (Quiroz, 2007a).

La aplicacion adecuada del filtrado de las imágenes digitales obtenida en una exploración con Tomografia Compurarizada, permite mejorar el compromiso entre la calidad de las imágenes y la dosis de radiación ionizante empleada en dicha exploración.
En el año 1915 Wagner y Campbell realizaron las primeras investigaciones de los filtros de forma independiente, y dieron a conocer el concepto básico de filtro y lo construyeron a base de inductores y capacitores, posteriormente en 1923 Zobel Otto, plantea la teoría de Imagen, misma que consiste en el diseño práctico de los distintos parámetros de los filtros ( Jiménez, 2007).

Según los estudios realizados por Giraldo Juan, Arboleda Carolina y McCollough Cynthia, el potencial de los diferentes métodos de reducción de ruido presente en las imágenes digitales obtenidas mediante Tomografía Computarizada se atribuye al uso de filtro bilateral anisotrópica (FBA) aplicado a un estudio cuantitativo en las tres dimensiones (x, y, z), utilizando reconstrucciones para corroborar la disminución del ruidos de las imágenes obtenidas y evaluar la utilidad clínica de las imágenes resultantes. Este tipo de filtro bilateral permitió obtener una mejor relación entre el ruido, la resolución espacial y la detectabilidad de bajos contraste y además, puede disminuir el ruido presente en las imágenes y garantizar una mejor calidad para un adecuado diagnóstico clínico (Giraldo et al., 2010).

Por otra parte, Castillo Angélica M. propone el uso del método de filtrado de difusión anisotrópica resuelto mediante el proceso de deconvolución, con el fin de, mejorar la resolución de las imágenes obtenidas en TAC eliminando defectos y los niveles de ruidos que se han amplificado

durante el proceso, pero sin afectar la resolución espacial de una imagen. El filtro de difusión anisotrópica permitió suavizar las zonas homogéneas de una imagen manteniendo los bordes y eliminando el pixelado en imágenes procesadas con deconvolución, tuvo en cuenta los criterios de la distancia espacial y una medida de la diferencia en intensidades de grises, permitiendo mejorar la relación señal-ruido, resolución espacial y la defectibilidad de bajos contrastes y garantizó una adecuada calidad de imagen (Castillo, 2016).

## PLANTEAMIENTO DEL PROBLEMA

En los distintos servicios de diagnóstico por imágenes, la Tomografía Computarizada es una técnica básicamente digitalizada debido a su facilidad y eficiencia para el almacenamiento, la transición y el procesamiento de datos que posterior a una técnica postprocesamiento nos permite obtener una imagen digitalizada.

Las imágenes digitales de la anatomía de un paciente se obtienen mediante la técnica de exploración de Tomografía Computarizada a través de cortes transversales, las mismas que por lo general adquieren una adecuada relación señal-ruido y una alta resolución, sin embargo, existe la presencia de estructuras densas de dimensiones pequeñas y bordes difuminados conocidos como artefacto *blooming.* Los mismos que se encuentran ligados a la fase de digitalización de una señal y la reconstrucción tridimensional de los datos, que limitan la valoración patológica de un paciente.

En este proyecto de grado se pretende corregir el efecto negativo de la presencia de los artefactos en una imagen empleando una técnica matemática y computacional denominado método de filtrado, para mejorar la calidad, resolución de las imágenes y relación señal-ruido, con el objetivo de obtener una adecuada valorización y diagnóstico médico.

## JUSTIFICACIÓN DEL PROBLEMA

La gran importancia del uso de la Tomografía Computarizada se debe a la obtención de imágenes médicas que permite visualizar la anatomía de cuerpo y tener un diagnóstico de enfermedades, con el fin de prescribir un oportuno y adecuado tratamiento, sin embargo, debido a que el cuerpo humano está constituido por estructuras de pequeño tamaño y diferentes densidades no pueden ser observadas con facilidad, ya que la resolución no es suficiente o el nivel de ruido presente en la imagen es muy elevado, razón por la cual se pueden aplicar métodos sobre el procesamiento

de imágenes que nos permitan resaltar pequeños detalles que son importantes para el médico y reducir en gran porcentaje el ruido presente.

Este trabajo pretende emplear un método de filtrado o de corrección que permite mantener el compromiso de conseguir imágenes de buena calidad y alta resolución, mediante bajas dosis de radiación y menor tiempo de exposición. La estrategia empleada en esta investigación consiste en encontrar un filtro adecuado mediante un algoritmo o método matemático que permita evaluar la calidad de la imagen filtrada, es decir, se buscará el filtro o máscara adecuada a través de la cual la imagen original se transformará en una nueva imagen con una depreciación de las fluctuaciones del ruido que altere en lo menos posible la señal, además, nos permitirá reducir las diferencias entre pixeles y sus pixeles vecinos de la imagen original que conlleva a la disminución del contraste y de la resolución espacial de la nueva imagen obtenida después del filtrado, con el objetivo de evaluar el resultado del efecto del filtro en la calidad de la nueva imagen.

## OBJETIVOS

### Objetivo General

- Realizar una simulación computacional mediante el lenguaje de programación de Matlab para mejorar la resolución de las imágenes obtenidas en Tomografía Computarizada.

### Objetivos Específicos

- Estudio de los principales parámetros de una imagen digital.
- Realizar una simulación mediante el lenguaje de programación de Matlab capaz de corregir las "imperfecciones" que afectan en la resolución de una imagen digital.
- Modificar parámetros de las matrices (pixel x pixel) de una imagen para reducir el ruido que afecta a la señal.
- Aplicar filtros para mejorar la resolución de una Imagen Tomográfica.

# CAPÍTULO I

## 1. MARCO TEÓRICO

### 1.1. Tomografía Computarizada

La Tomografía Computada es una técnica de diagnóstico por imagen no invasiva y precisa que nos permite escanear o visualizar diferentes cortes de las estructuras del organismo mediante la combinación de un haz de rayos X y tecnología computacional e informática. Aprovecha al máximo la información presente en un haz de rayos X incidente, cubriendo las falencias procedentes de la radiología clásica mediante la delimitación de estructuras en base a las diferentes densidades de huesos, tejido blando y vasos sanguíneos de las estructuras directamente estudiadas (Calzado & Geleijns, 2010).

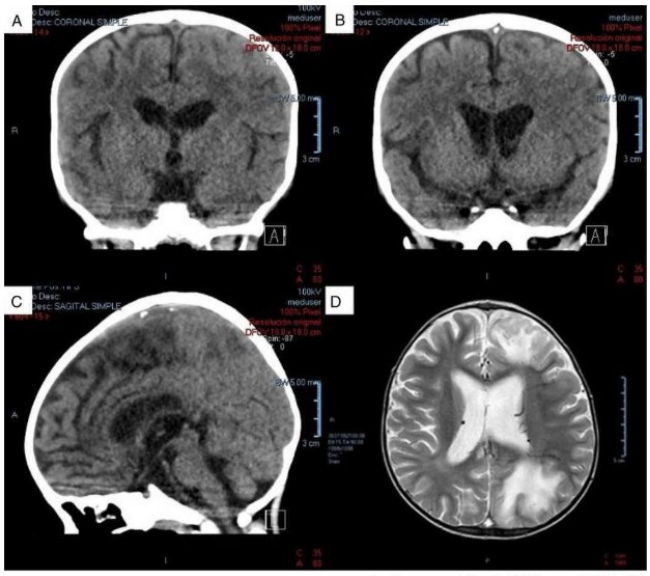



**Figura 1-1.** Estudio Tomográfico simple de Cráneo.
**Realizado por:** (Rivera et al., 2019).

Esta técnica tuvo su origen en el año de 1971 con Godfrey Hounsfield y Allan Cormack, quienes dieron a conocer el primer equipo de Tomografía Axial Computarizada instalado en Estados Unidos. Se utilizó para realizar exámenes mediante Tomografía durante un largo tiempo y empleado únicamente para generar imágenes craneales, las cuales eran indispensables para la neuroradiología. Con el paso del tiempo la tecnología ha ido avanzando y revolucionando el diagnóstico radiológico, hoy en día la Tomografía Computarizada (TC) se ha convertido en una técnica de exploración más utilizada debido a que se emplea intervalos de tiempos muy cortos y que proporciona información muy valiosa y fiable para el diagnóstico y tratamiento del paciente mediante la obtención de imágenes tridimensional de cualquier área anatómica (Calzado & Geleijns, 2010).

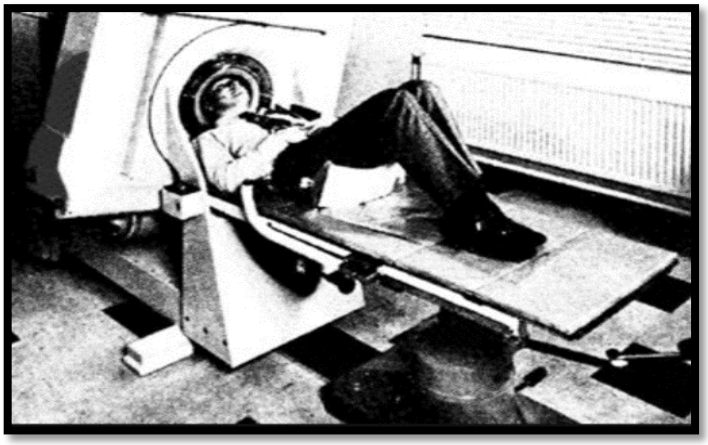


**Figura 2-1.** Primer Tomógrafo inventado G. Hounsfield.
**Realizado por:**(Plazas, 2018).

En 1971 se introdujo los primeros escáner de uso clínico que adquirían los datos del cerebro en aproximadamente 4 minutos, en dos secciones contiguas y con un tiempo de 7 minutos por cada imagen generada en la exploración; en 1976 se desarrolló escáneres Axiales con una sola fila de detectores, los cuales se podían aplicar a un estudio de cualquier parte del cuerpo y posteriormente a los escáneres Helicoidales o espirales que se contienen múltiples filas de detectores y una amplia difusión de exploraciones a nivel de todo el cuerpo de estudio.

Hoy en día existen varios tipos de sistemas Tomográficos como:

♦ Sistemas de primera generación: están constituidos por una fuente puntual y un detector, que se desplazan generando 160 trayectorias paralelas en pequeños incrementos con ángulo de apertura de 1 grado, se obtiene una proyección y datos en distintos ángulos de la proyección (Amed & Rojas, 2011, p. 20).

♦ Sistemas de segunda generación: el tubo emite radiación perpendicular al plano en diferentes direcciones, describe trayectorias que divergen desde un mismo punto como un abanico conocido como *fan beam* con un ángulo de apertura de 5 grados aproximadamente, cuanta con detectores que varían entre 10-30 formados linealmente y logran una exploración de 2 minutos (Amed & Rojas, 2011, p. 20).

♦ Sistemas de tercera generación: el método de *fan beam* se extiende a 3 dimensiones y el detector forma un plano, cuenta con 300 y 500 detectores, el haz de rayos X se proyecta con un ángulo entre 25 y 35 grados capaces de cubrir toda el área de exploración, tiene movimiento de rotación de 360 grados del tubo y el plano de detectores y permite una exploración de 2 a 3 segundos (Amed & Rojas, 2011, p. 20).

♦ Sistema de cuarta generación: es un equipo muy costoso, consta de un anillo de detectores dentro del cual, se ubica el tubo generador de rayos X, sus detectores giran a altas velocidades lo

cual permite una reducción del tiempo de exploración, obteniendo resultados de alta resolución en cuanto a la calidad (Amed & Rojas, 2011, p. 20).

### *1.1.1. Principios de la Tomografía Computarizada*

Tomografía computarizada (TC) viene del griego; tomé= cortar y grafo=dibujo y es conocida como una técnica o estudio a través del cual se obtiene representaciones gráficas tridimensionales de la estructura interna de un objeto, mediante cortes Tomográficos sin necesidad de realizar dicho corte (Molina, 2012).

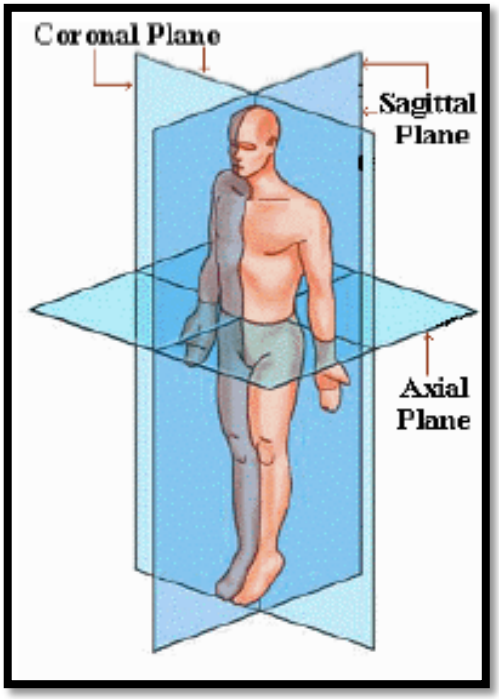



**Figura 3-1.** Tipos de cortes Tomográficos.
**Realizado por:** (Molina, 2012)

El objetivo de la TC es medir la transmisión de un haz de rayos X colimado y delgado que pasa a través del cuerpo de un paciente en una gran variedad de proyecciones con forma de abanico en dirección a un detector, el cual mide la intensidad de dicha transmisión, descarta la superposición de tejidos mediante la obtención de cortes Tomográficos del paciente.

### *1.1.2. Componentes que integra un equipo de Tomografía Computarizada*

Todos los equipos de Tomografía están constituidos básicamente por 3 partes muy importantes: el Gantry, la computadora y la consola, ver (**figura 4-1**).

***El Gantry:*** Es la parte física o cuerpo vertical de la unidad que presenta un orificio central en el que se introduce la camilla de exploración con el paciente, contiene el tubo emisor de rayos X, colimador, los detectores, el sistema de adquisición de datos (DAS) y toda la parte mecánica para

realizar el movimiento del mismo asociado con la exploración. Existen dos tipos de Gantry, los que rotan $360^0$ y cambian de dirección y los de rotación continua.

***Tubo de rayos X:*** es un tubo de vidrio al vacío con una cubierta de plomo, donde se generan los rayos X, dentro de este tubo se aceleran unos electrones en primer lugar, para después frenarlos bruscamente y posteriormente salir al exterior a través de una ventana pequeña, ***Colimador:*** es la parte del equipo que nos permite regular el tamaño del haz de rayos X, ***Detectores:*** los detectores reciben los rayos X que logran atravesar el cuerpo del paciente para posteriormente convertirlos en una señal eléctrica mediante un fotodiodo y ***DAS:*** realiza la conversión análoga de la señal eléctrica para que la computadora procese los datos almacenados.

***Computadora:*** parte más importante del equipo ya que el funcionamiento total del equipo depende del software del equipo instalado en la computadora, además se encarga del procesamiento para la reconstrucción de las imágenes a partir de los datos almacenados por el sistema de detección (Amed & Rojas, 2011).

***Consola:*** controla las operaciones del equipo, en algunos casos se encarga de la conversión de la imagen digital almacenada en el disco duro de la computadora en una señal que se puede visualizar en el monitor de TV (Amed & Rojas, 2011).

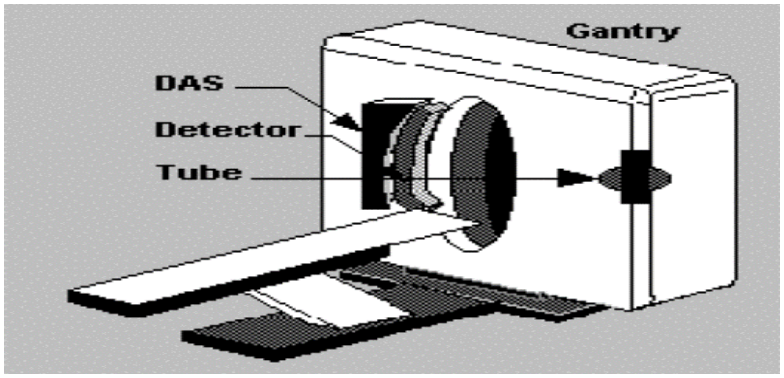



**Figura 4-1.** Componentes del equipo de Tomografía Computarizada.
**Realizado por:** (Amed & Rojas, 2011).

Las proyecciones se generan a partir de una relación entre el paciente y el sistema de detectores que generalmente tiene un ancho que varía de 1 a 20 mm en forma de un arco y se encuentran en movimiento perpendicular al colimador mientras la mesa donde está el paciente permanece inmóvil, esta técnica permite obtener múltiples datos que son procesados en una computadora para generar una serie de imágenes con una excelente resolución espacial, y son representadas por información tridimensional de la anatomía del paciente que posee un gran valor en el diagnóstico a partir de un estudio dinámico (Yáñez, 2016), ver figura **5-1**.

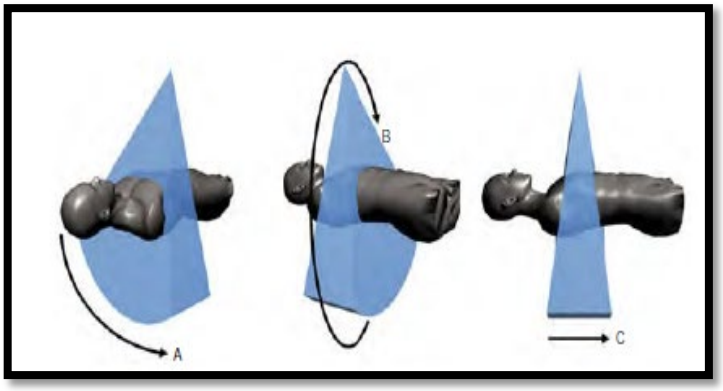



**Figura 5-1.** Perfiles de Transmisión de Rayos X.
**Realizado por:** (Calzado & Geleijns, 2010)

## 1.2. Parámetros que caracterizan una Imagen

### *1.2.1. Coeficiente de Atenuación*

El coeficiente de atenuación lineal μ es la capacidad que posee un material para detener fotones, es directamente proporcional al número atómico del material (Z) y su densidad es inversamente proporcional con la energía. El coeficiente de atenuación según la absorción de energía se puede clasificar de acuerdos a dos procesos generados durante la interacción de la radiación con la materia como: el efecto Compto que predomina en los tejidos blandos debido a que absorbe parte de la energía de su fotón incidente y el efecto Fotoeléctrico dominante en los materiales de alto número atómico.

El coeficiente de atenuación lineal de un material aumenta en la imagen observándose más blanco y si disminuye μ se verá más negro, otro efecto del cual depende el coeficiente de atenuación lineal se denomina endurecimiento del haz y se produce cuando los fotones de baja energía son absorbidos o fácilmente atenuados por el tejido a una profundidad mayor con un coeficiente de atenuación menor (Giraldo et al., 2008)

### *1.2.2. Adquisición de datos*

Una imagen de la TC es un conjunto bidimensional de una serie de datos o valores obtenidos del flujo de fotones emitidos con una intensidad $\mathbf{I_0}$ que logra atravesar un objeto para distintos ángulos de incidencia a lo largo de una trayectoria **L**, están cuantificados en escala de grises que representan al coeficiente de atenuación **μ** de cada elemento de volumen o vóxel, define una pequeña parte de la imagen por el espesor de corte donde se puede asignar un coeficiente de atenuación lineal efectiva **μ** que depende de la anchura del haz de radiación (García, 2015).

La figura **6-1**, muestra el proceso de adquisición de la imagen representada por pixeles que contienen la atenuación efectiva de cada vóxel evaluado, el mismo que es proyectado a través de integrales lineales (García, 2015).

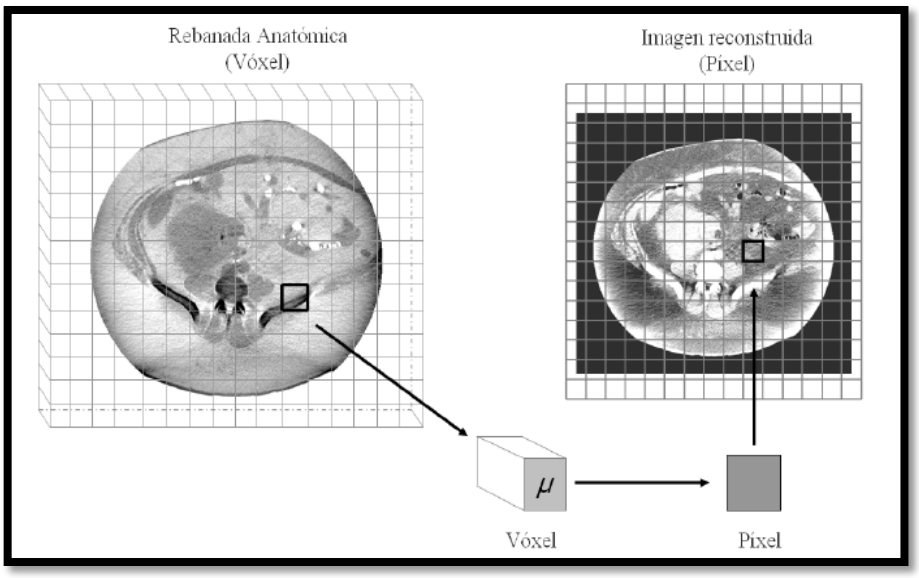



**Figura 6-1.** Proceso de Adquisición de una imagen.
**Realizado por:** (García, 2015)

La intensidad del haz de radiación transmitido por los diversos puntos de exploración en diferentes ángulos es recolectada en un equipo especializado en reconstrucción de imágenes de cada corte. La representación gráfica se observa en forma de una matriz plana con toda la información que se obtiene de cada vóxel.

Los valores de cada pixel que se asigna a cada matriz activa, sirven para reconstruir la imagen de TC, se relaciona con la atenuación en el tejido correspondiente al estudio, es decir, se relaciona concretamente con el coeficiente de atenuación lineal μ.

$$\boldsymbol{I(x) = I_0 e^{-\mu x}} \quad \textbf{(1)}$$

Dónde: $\boldsymbol{x}$ representa el espesor de un objeto o material, $\boldsymbol{\mu}$ depende de la composición y de la densidad del material y de la energía de los fotones del haz de radiación, su tamaño depende del campo de visión de la imagen reconstruida y del tamaño de la matriz de la imagen.

La ecuación **1** define la atenuación del haz primario que describe la intensidad del haz de radiación después de atravesar un espesor $\boldsymbol{x}$, pero no toma en cuenta la intensidad de la radiación dispersa causada por el efecto Compto, por lo tanto, se integra la ecuación 1 para todas las energías de los fotones polienergéticos presentes en el haz de radiación, sin embargo, la integración (2) no se aplica en la metodología de retroproyección de los algoritmos matemáticos para la reconstrucción

de TC, por lo tanto se asume que (1) representa el valor de la energía media o efectiva del espectro lo que genera la presencia de inexactitudes durante la reconstrucción (Calzado & Geleijns, 2010).

$$I(d) = I_0 e^{-\int_0^d \mu(x)dx}$$

**(2)**

Debido a las diferentes densidades del cuerpo del paciente a través del cual pasa el haz de rayos X, el coeficiente de atenuación lineal y la intensidad tendrán distintos valores después que el haz es atenuado después de atravesar cierto espesor del tejido, relacionando al paciente sometido al examen TC como una matriz de distintos valores de coeficientes de atenuación lineal $(\mu_{ij})$ (Calzado & Geleijns, 2010).

Una vez que los fotones del haz incidente han logrado atravesar el cuerpo del paciente son captados por un sistema de detectores que mide la intensidad transmitida y los transforma en una señal en función de los coeficientes de atenuación, posteriormente los datos obtenidos son enviados a un computador donde se procesan mediante algoritmos de reconstrucción.

Desde este punto de vista la formación de una imagen Tomografía se considera como una matriz de diferentes coeficientes de atenuación lineal. Durante los años setenta Hounsfield determinó que existen dos mil posibles coeficiente de atenuación que se representan en las diferentes partes del cuerpo humano como en tejido blando y huesos (Dillenseger & Moerschel, 2012).

## ESCALA DE HOUNSFIELD

Hounsfield propuso una nueva unidad que lleva su nombre unidad Hounsfield (UH), que en TC es utilizada para la reconstrucción de los coeficientes de atenuación lineal durante la transformación en una matriz de números medidos en UH del material o tejido estudiado tomando como referencia el valor del coeficiente de atenuación del agua como se formula en la ecuación 3, la atenuación de los tejidos en la escala Hounsfield se expresa en relación con el coeficiente de atenuación lineal del agua a temperatura ambiente según (Calzado & Geleijns, 2010):

$$UH_{tejido} = \frac{\mu_{tej} - \mu_{agua}}{\mu_{agua}} \times 1000$$

**(3)**

Además, la escala de Houndfield clasifica los tejidos, teniendo en cuenta su coeficiente de atenuación (**Tabla 1-1**).

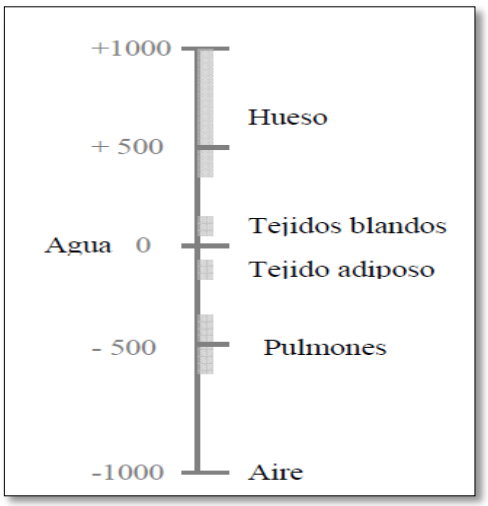



**Figura 7-1.** Valores en Escala Houndfield.
**Realizado por:** (Giraldo et al., 2010)

**Tabla 1-1:** Escala de Houndfield para diferentes Tipos de Tejido.

| TEJIDO | ESCALA DE HOUNDFIELD |
|---|---|
| **Hueso** | **+400. +1000** |
| **Tejidos Blandos** | **+10. +60** |
| Hígado | +49. +60 |
| Materia blanca | +43 |
| Materia gris | +40 |
| Músculo | +10. +40 |
| Riñones | +30 |
| **Agua** | **0** |
| **Tejidos Adiposo** | **-50.-100** |
| **Pulmones** | **-600. -400** |
| **Aire** | **-1000** |

**Fuente:** (Dillenseger & Moerschel, 2012)
**Realizado por:** Villagomez H, Ximena A. 2020

En la **Tabla 1-1** según (Dillenseger & Moerschel, 2012) representa la clasificación en la que se distingue 3 zonas sobre la escala Houndfield:

- Las estructuras de bajas densidades que se sitúa en la parte negativa de la escala definido por un valor teórico de -1000 UH para densidades menores que del agua.
- Los tejidos blandos que tienen coeficientes de atenuación agrupado ente -100 y +100 UH.
- Las estructuras densas que tienen de +100 a varios miles de UH

Cada incremento de una UH se asocia a un incremento del 0,1 % del coeficiente de atenuación lineal relativo al del agua. Durante la visualización de las imágenes de TC mediante la escasa de UH cuantitativamente está compuesta por miles de niveles de grises diferentes, sin embargo, el

ojo humano no puede visualizar los tejidos blando debido a la falta de un contraste, siendo necesario, atribuir una escala de grises a una porción específica de UH, con el fin de compensar el contraste visible a los tejidos blando, esto se realiza mediante una operación denominada ventana (Dillenseger & Moerschel, 2012).

La ventana compensa los límites fisiológicos del ojo humano modificando el nivel de la ventana (WL) y el ancho de la ventana (WW), permitiendo enfocar el estudio de los contrastes ya que modifica el brillo o ennegrecimiento de la imagen. Existen tres tipos de ventanas para la visualización de las densidades del cuerpo: las ventanas óseas; tiene un ancho muy extendido lo que permite visualizar un conjunto de tejidos óseos; las ventanas pulmonares tiene un ancho extendido y su nivel es inferior se sitúa en la parte negativa de la estaca de Hounsfield incluye tejido del parénquima pulmonar; y las ventanas blandas se sitúa apenas por encima del valor de referencia del agua en la escala de UH, su nivel es próxima al nivel cero (Dillenseger & Moerschel, 2012).

### *1.2.3. Fundamentos de una Imagen Médica*

La ciencia médica por mucho tiempo utiliza diversas formas de energía que penetran e interactúan hasta un cierto límite con el interior de diferentes tejidos, obteniendo como el resultado final una distribución de niveles de grises o tonos de colores para crear imágenes del cuerpo humano capaz de ser visualizada. La calidad de una imagen dentro del campo del diagnóstico médico es de suma importancia ya que la información presentada gráficamente tiene un valor de diagnóstico (Graffigna & Romo, 2003).

### *1.2.4. Parámetros de Calidad de una Imagen*

Una imagen digital es la representación de una función bidimensional $f(x,y)$ donde sus coordenadas son de tipo espaciales y el valor de la función $f$ es conocido como intensidad de luz o escala de grises, es una matriz tridimensional $n \times m \times z$ compuesta por un número finito de elementos denominados píxeles, cada uno con una localización y un valor específico entre 0-255 los mismo que indican el nivel de intensidad de la luz.

$$Matriz\ tridimensional\ \ representa\ matematicamente\ por:$$

$$\boldsymbol{nxmxk} \quad donde\ \boldsymbol{nxm} \gg dimensiones\ de\ la\ imagen\ y\ \boldsymbol{k}$$

$$\gg n\acute{u}mero\ color\ de\ la\ im\acute{a}gen\ en\ escala\ de\ grises$$

Las imágenes o cortes Tomográficos son de gran utilidad en la correcta interpretación médica, sin embargo, las mismas están expuestas a una amplia variedad de problemas conocidos como

artefactos que dan lugar a la degradación de la calidad visual de las imágenes. Para ello es necesario el análisis de ciertos parámetros que determinan la calidad de la imagen, entre estos tenemos:

### *1.2.4.1. Resolución Espacial:*

La resolución espacial es la capacidad que posee un sistema para distinguir todos los detalles o estructuras de dimensiones muy pequeñas o bordes nítidos, suele medirse en términos de pixeles por pulgadas (ppi) y de esto depende la calidad y el tamaño de la imagen generada, es decir, una imagen digital tiene mejor resolución espacial si posee mayor número de pixeles (Benito, 2015)

### *1.2.4.2. Artefactos en TAC*

El artefacto es cualquier tipo de discrepancia sistemática entre la imagen reconstruida y los verdaderos coeficientes de atenuación del objeto, debido a que la imagen se reconstruye a partir de 1 millón de medidas de detectores independientes, por lo tanto, los algoritmos de reconstrucción, asumen que todas las medidas son equilibradas y cualquier tipo de error de medida se refleja con un error en la reconstrucción de la imagen referenciar.
Los artefactos se pueden presentar en una imagen de acuerdo a cuatro categorías: de acuerdo al origen físico del proceso de adquisición de la imagen, originados por el paciente, originados por las imperfecciones en el escáner y originados durante el proceso de reconstrucción de la imagen (Molina, 2012, p. 47 a).

**Endurecimiento del haz:** es el proceso en que la energía media del haz de rayos X aumenta al atravesar un material donde los fotones de menos energía son absorbidos, estos rayos X que logran atraviesan un material por distintos caminos y salen con espectros diferentes. Generan medidas inconsistentes que se observan en la visualización de la imagen tomada, ejemplo: en una imagen se presentan líneas y bandas oscuras entre objetos densos en volúmenes heterogéneos (figura **8-1**) (Molina, 2012).

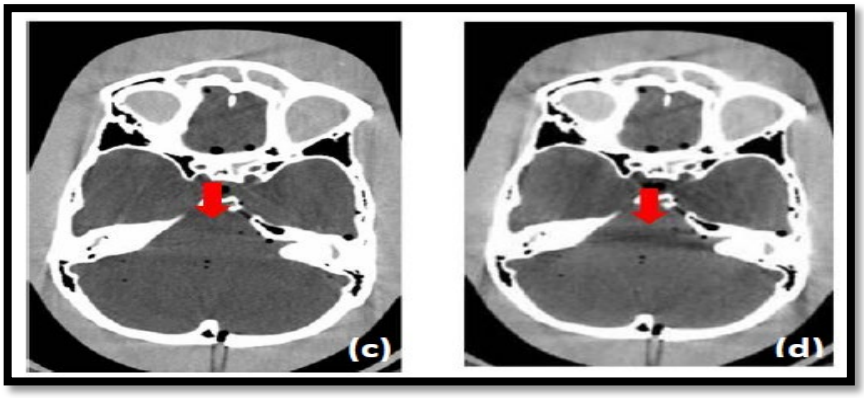



**Figura 8-1.** c) Imagen de TAC y d) Corte Axial Endurecimiento del haz.
**Realizado por:**(Molina, 2012, p. 48 c)

**Movimiento del paciente:** los movimiento involuntarios o provocados por el paciente generar inconsistencias en la proyección del equipo provocando la visualización de artefactos en la imagen reconstruida, observando la presencia de bandas que da lugar a una imagen difuminada, como se muestra en la figura **9-1.**

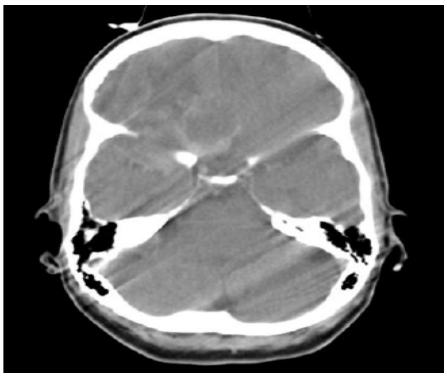


**Figura 9-1.** Ruido por Movimiento del paciente.
**Realizado por:** (Molina, 2012, p. 50 d)

*1.2.4.3. Tipo de Ruido presentes en las imágenes de TAC*

El ruido son fluctuaciones indeseables que muy frecuente aparecen de improvisto en la adquisición de las imágenes afectando su calidad, es un proceso aleatorio definida como variaciones estocásticas de pixeles erróneos que se mezclan con los pixeles aceptables que componen la imagen dándole un aspecto granuloso (Benito, 2015). Los tipos de ruidos más comunes que se presentes en la adquisición de imágenes digitales son:

**Ruido Gaussiano:**

Es el tipo de ruido que se presente en la adquisición de la imagen por las malas condiciones del equipo generando el emborronamiento de todos los pixeles de la imagen, como se muestra en la figura **10-1**, cambiando la información de sus características.

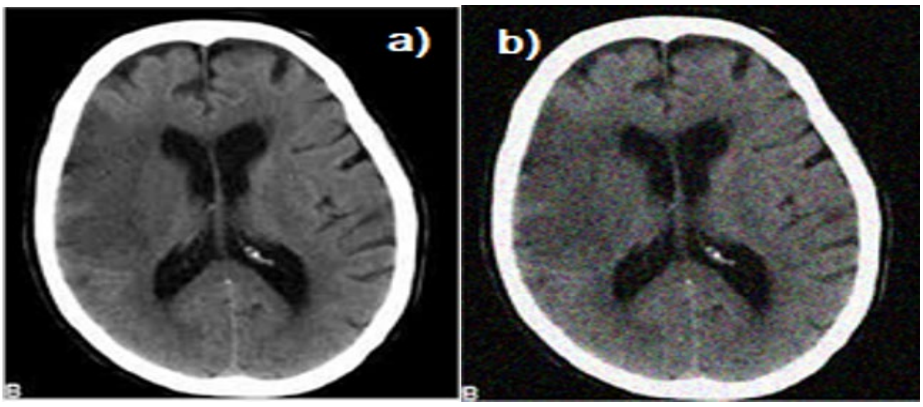



**Figura 10-1.** a) Imagen Original, b) Imagen con presencia de Ruido Gaussiano.
**Realizado por:** Villagomez H, Ximena A. 2020

El Ruido Gaussiano se refiere al ruido que se suma a la señal (imagen) debido a las condiciones de iluminación cambiando los valores de intensidad de cada pixel, cuya función de densidad de probabilidad es relaciona a la distribución normal o de Gauss.

Para generar Ruido Gaussiano es necesario realizar varias muestras aleatorias tomando en cuenta que el valor medio es igual a cero y el valor de la varianza puede variar entre los límites permitidos de la imagen el mismo que será escogido por el usuario, lo cual permite tener el control del nivel de ruido presente en la imagen, de esta manera se genera una distribución Gaussiana o normal.

Sea la imagen I con una función $F(x, y)$ y la función del ruido gaussiano $G(x, y)$ se obtiene una imagen resultante contaminada por señales de ruido gaussiano representada por la función (ecuación **4**):

$$E(x, y) = F(x, y) + G(x, y) \quad \textbf{(4)}$$

La ecuación **5**, expresa la Distribución de probabilidad gaussiana es:

$$p(G) = \frac{1}{\sqrt{2\pi}\sigma_G} e^{-\frac{1}{2}(\frac{G-\mu_G}{\sigma})^2} \quad \textbf{(5)}$$

**Donde:** $\boldsymbol{\sigma_G}$ Desviación estándar del ruido, dicho valor en la presente tesis deberá estar entre 0-255 que corresponde a los valores límites de los tonos de grises de la imagen J y $\boldsymbol{\mu_G}$ Valor medio del ruido.

La varianza del ruido V se define como el cuadrado de la desviación estándar, fórmula:

$$V = \sigma^2 \quad \textbf{(6)}$$

Se aplica la fórmula 4 para generar el ruido Gaussiano Blanco

$$G = \sqrt{\frac{12xV}{k}}\left(\sum_{i=1}^{k} r - \frac{k}{2}\right) + \mu$$

**(7)**

*Función de Ruido Gaussiano* $G(x, y)$, ecuación (**7**): Se define como un parámetro de la distribución Gaussiana de la ecuación **(5).** Dónde: G representa el valor del ruido gaussiano; V la Varianza del ruido, ecuación **(6),** R indica el muestreo aleatorio generado por el computador en función RND; k indica el valor del rango de valores de la variable G y $\mu$ es el Valor medio del ruido que en nuestro caso es igual a cero porque es necesario generar una distribución gaussiana o normal con una curva simétrica en el cero.

**Ruido Sal y Pimienta o Impulsivo:**

Al este tipo de ruido también se le conoce como **"sal y pimienta"** se presenta en la transmisión de datos obtenidos por las proyecciones, estas fluctuaciones son errores que afectan ciertos pixeles dejándolos totalmente blancos o negros, que se pueden apreciar en la visualización de imagen como se muestra en la figura **11-1**.

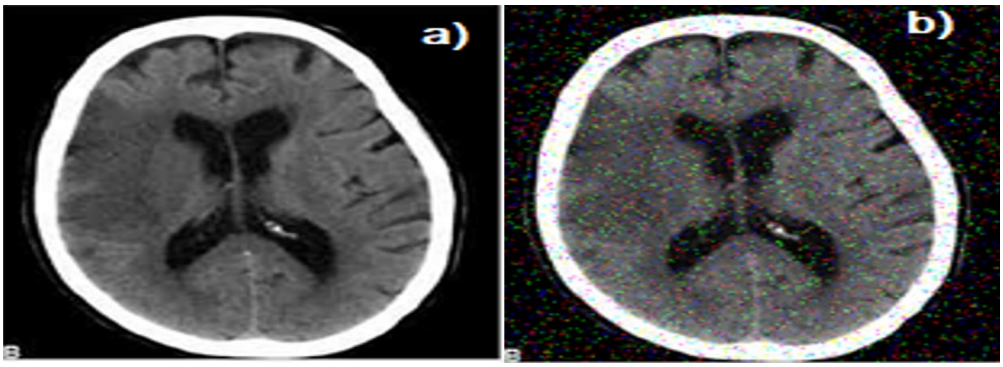



**Figura 11-1.** a) Imagen original b) Degradación por Ruido Sal y Pimienta.
**Realizado por:** Villagomez H, Ximena A. 2020

**Ruido Poisson**

Este tipo de ruido es el más frecuente en la mayoría de sistemas de adquisición de imágenes, para tratarlo de forma directa es difícil, ya que es un ruido no aditivo y correlacionado con la información útil del pixel, para su procesamiento es necesario la Transformada de Anscombe que mediante una varianza unitaria facilita la eliminación de ruido.

LA formulación del ruido Poisson, se expresa mediante la ecuación (**8**):

$$f(Z_i, Y_i) = \frac{Y_i^{Z_i} . e^{-Y_i)}}{Z_i!}$$

**(8)**

Donde $\boldsymbol{Z_i}$ Representa la variable independiente de la distribución de Poisson y $\boldsymbol{Y_i}$ Es el valor medio de las intensidades en cada pixel de la imagen de prueba.

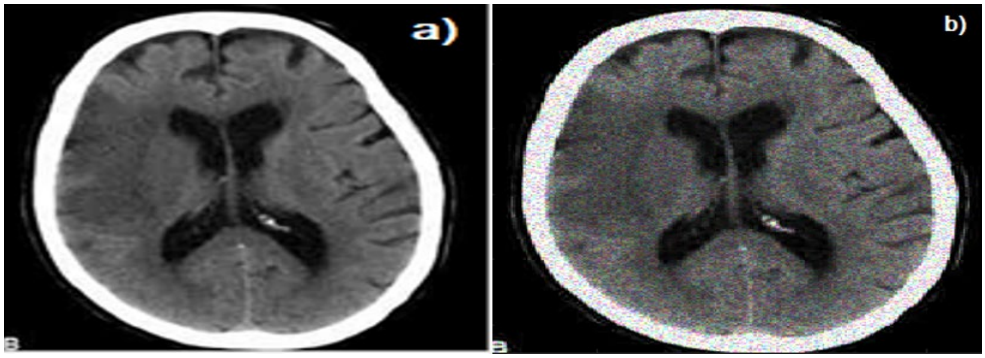



**Figura 12-1.** a) Imagen Original b) Degradación mediante Ruido Poisson**.**
**Realizado por:** Villagomez H, Ximena A. 2020

**Ruido Speckle o punto**

Las imágenes con este tipo de ruido multiplicativo, son difíciles de tratar debido a que es un ruido que se genera del sistema de captura de datos y se desarrolla mediante el producto de variables aleatorias como la varianza y la media.

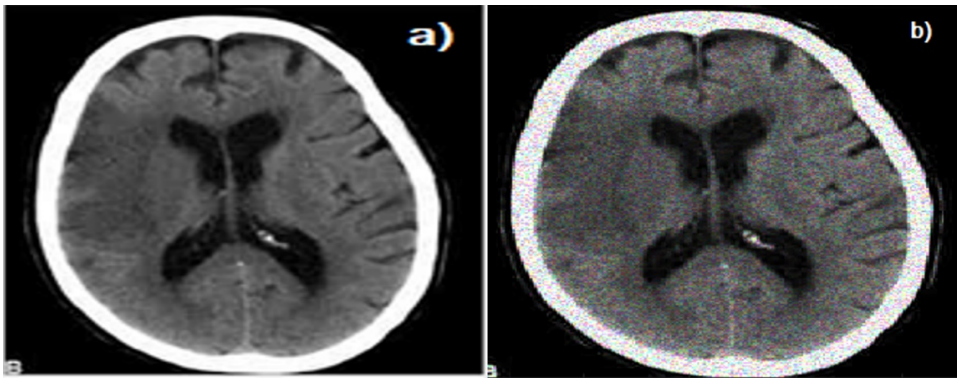



**Figura 13-1.** a) Imagen Original b) Degradación por Ruido Speckle o Punto.
**Realizado por:** Villagomez H, Ximena A. 2020

### *1.2.5. Histograma*

Los histogramas son representaciones estadísticas que muestra la probabilidad con la que un determinado nivel de gris aparece en la imagen, se representa por el número de pixeles que tiene el mismo nivel de gris dentro del rango dinámico de la imagen, figura **14-1b** (López, 2014).

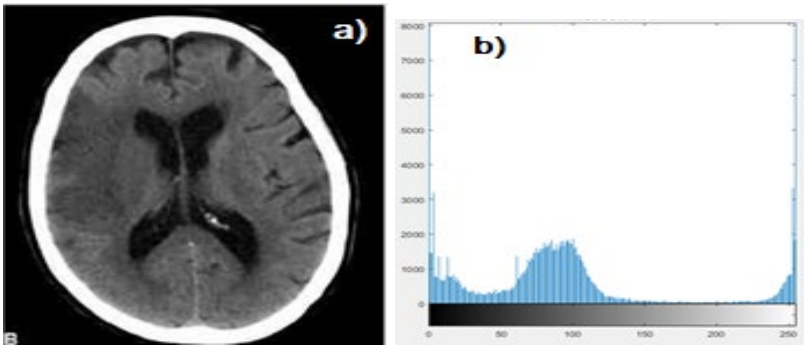



**Figura 14-1.** a) Imagen de Prueba, b) Histograma de la Imagen.
**Realizado por:** Villagomez H, Ximena A. 2020

### *1.2.6. Métodos de Reconstrucción de Imágenes*

La reconstrucción de imágenes Tomográficas se basa en una serie de proyecciones o datos medidos en intervalos de 0 a 2π en el sinograma o Transformada de Radon, siendo considerado como un conjunto de proyecciones que permiten obtener una estimación de la imagen real del objeto variando el ángulo de incidencia. La efectividad de la reconstrucción depende del proceso de adquisición de datos, la adecuada implementación de las fórmulas matemáticas para la reconstrucción y del postproceso de las imágenes reconstruidas.

El matemático australiano Johann Radon desarrolló en 1917, un método matemático denominado Teoría de Radón o transformada de Radón que sirvió de apoyo a los fundamentos teóricos de la tomografía, ya que establece la posibilidad de reconstruir un objeto por medio de sus proyecciones. En 1963 el físico Cormack utilizando la teoría de Radón, logró medir pequeñas diferencias de densidades y propuso la teoría de reconstrucción por computación (Dillenseger & Moerschel, 2012).

Existen diferentes métodos de reconstrucción ya sea analíticos e iterativos, toman en cuenta las bases matemáticas y hacen la inversa de las funciones $f(x, y)$ de las proyecciones hasta llegar al conjunto de datos originales obtenidos inicialmente.

#### *1.2.6.1. Método de reconstrucción iterativa o algebraica*

El algoritmo de interacción (repetición) se aplica a los problemas matriciales, consiste en dar una primera solución y efectuar cálculos simples, compara con los datos de entrada del problema y efectúa las correcciones necesarias para dar soluciones definitivas para la estimación de una imagen que es la fiel representación de los datos de entrada (Molina, 2012).

Este método iterativo de reconstrucción consta de 3 procesos sistematizados y continuos: el primero es el *proceso de resolución del sistema de ecuaciones* que se utiliza para obtener una imagen de solución aproximada empleando el Método de respuesta rápida (*Quick Response Code*) de mínimos cuadrados, luego realiza el *proceso de filtrado de la imagen obtenida* que sirve para disminuir o eliminar la presencia de ruido mejorando la calidad de la misma y finalmente el *proceso de aceleración* que permite repetir el método iterativo hasta lograr una convergencia en caso de que sea necesario (Vega, 2010).

El ordenador va haciendo intentos de sumas en vertical, horizontal y diagonal, hasta que obtiene la coincidencia de todos los datos y no se podrá reconstruir la imagen el ordenador hasta que tuviera todos los datos (Vega, 2010).

### *1.2.6.2. Métodos de reconstrucción Analíticos*

Estos métodos se basan en la búsqueda de una expresión analítica que nos permite obtener la transformada inversa de Radon para pasar del espacio los datos obtenidos en el Tomógrafo al espacio de la imagen tridimensional. Además se basa en el teorema del corte central que relaciona las proyecciones con la transformada de Fourier de la distribución $f(x, y)$ (Molina, 2012).

Los filtros que utilizan el método analítico para la reconstrucción de imágenes son conocidos como filtros Kernel que se basan en fórmulas matemáticas, los filtros más importantes son:

**Filtro Paso-Alto**: es el filtro que tiene un efecto opuesto al del filtro de pasa-baja, aísla o no atenúan los componentes o señal de alta frecuencia mientras que preserva los componentes de bajas frecuencias comprendidas entre 0 o por debajo de la frecuencia de corte ω. Para esto es necesario de una máscara de tres dimensiones que indica que el pixel central de la vecindad del pixel de entrada que se procesa aporta un valor de brillo muy diferente al de sus pixeles vecinos dándole una versión acentuada en la imagen de salida (Aldalur & Santamaría, 2002).

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Máscara $3x3$

**Filtro de Suavizado o pasa baja**: este tipo de filtro elimina o atenúa las altas frecuencias y preservan las bajas frecuencias, es decir elimina el ruido de tipo impulsivo (relación con las altas frecuencias) en la reconstrucción reduciendo las altas frecuencias, sin embargo, cuanto mayor sea la reducción de las altas frecuencias en comparación a la frecuencia de corte, menos importancia tendrá el ruido, pero más suavizada resulta la reconstrucción de las imágenes. Su forma general es expresada por la ecuación $T(s) = \frac{H}{D(s)}$ donde $T(s)$ es una función de frecuencia, $H$ es la

función de señal de frecuencia de entrada o de corte y $D(s)$ es un polinomio de componentes reales (Quiroz, 2007b).

**Filtro de Realce de Bordes**: o también se les denomina como filtros detectores de bordes, consiste en transformar la imagen original exhibiendo los detalles de bordes o frontera, de manera que se le atribuye nuevos valores de niveles grises a un pixel original según la influencia de sus pixeles vecinos, es decir, a mayor o menor influencia de la vecindad del pixel central se tendrá un realce mayor o menor de los detalles de la imagen original.
Máscaras de filtraje están constituida por valores positivos, nulos o negativos en influencia del pixel central, que se utilizan en el diseño del filtro para el realce de bordes.

$$\begin{bmatrix} -0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Grises Altos Grises Medios

La aplicación de las máscara de grises altos en el proceso de filtrado genera una imagen más clara y las máscaras medias dan como resultado imágenes con niveles de grises intermedios (Coca et al., 2008).

### 1.3. Definición de Proyecciones

Un paciente que se realiza una exploración de Tomografía computarizada se describe como una distribución $f(x, y, z)$ en 3 dimensiones con propiedades diferentes en cada punto $(x, y, z)$, en donde se hacen cortes transversales para obtener una imagen, es decir, un rayo X penetra un tejido con una intensidad media $I$, el cual es absorbido a lo largo de su trayectoria. A este conjunto de trayectorias del haz de rayos X se denomina como proyecciones paralelas, las mismas que con un ángulo θ, establecen una relación entre la transformada de Fourier unidimensional de una proyección y la transformada de Fourier bidimensional del objeto (Amed & Rojas, 2011).

La proyección matemáticamente se define con la ecuación (9) y es una señal unidimensional generada por la intensidad del rayo X que atraviesa al objeto en línea recta a lo largo de su trayectoria para cada ángulo θ pasando cada vóxel, dichas proyecciones se almacenan en una matriz $p(x, \theta)$. En la ecuación 4, se define a $t$ y $s$ como un nuevo sistema de coordenadas rotado por un ángulo $\theta$ y $\mathcal{R}$ representa al operador de proyección.

$$p(t,\theta) = \mathcal{R}_\theta f(t) = \int_{-\infty}^{\infty} f(t\,cos\theta - s\,sin\theta, s\,cos\theta + tsen\theta)ds$$

(9)

Y la transformada de Fourier unidimensional de la proyección $\mathcal{R}_\theta f(t)$ es formulada como la ecuación 10:

$$\widehat{\mathcal{R}_\theta f(t)} = \int_{-\infty}^{\infty} \mathcal{R}_\theta f(t) e^{-2\pi i \gamma t} dt$$

(10)

$$\widehat{\mathcal{R}_\theta f(t)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i \gamma (x cos\theta + y s\, sen\, \theta)} dxdy$$

(11)

Por consiguiente, la ecuación **10** se reescribe como la ecuación 11, definiendo a una proyección como el conjunto o la suma de las atenuaciones con la que un rayo X en dirección θ logra atravesar los pixeles de una imagen, tomando en cuenta que la atenuación puede ser diferente y distribuirse de manera uniforme en cada uno de los pixeles que logre atravesar dando como resultado una imagen borrosa referenciar.

## TEOREMA DEL CORTE CENTRAL O TEOREMA DE PROYECCIÓN

Para llegar a demostrar este teorema es necesario analizar la relación entre la Transformada de Fourier bidimensional (TF 2D) $f(u,v)$ y la Transformada de Fourier (TF) unidimensional $f(x,y)$ de la proyección obtenida en θ=0, esto quiere decir que, la TF unidimensional de una proyección generada a partir de rayos paralelos entre sí de una distribución $f(x,y)$ formando un ángulo θ con el eje X es igual a los cortes o valores de la Transformada de Fourier 2D de la distribución en la recta que pasa por el origen formando el mismo ángulo θ con el eje μ (Molina, 2012).

En la ecuación 9 se representa la relación a partir de un número infinito de proyecciones que existe entre la Transformada inversa de Fourier $f(x,y)$ definida en el plano $xy$ (continua e integrable), con su función compleja la Transformada de Fourier en dos dimensional (coordenadas polares) $F(u,v)$, para recuperar perfectamente la imagen del objeto $f(x,y)$ (dominio espacial) si se conoce todos los puntos de $F(u,v)$ (dominio de frecuencia).

$$f(x,y) = \iint_{-\infty}^{\infty} F(u,v) e^{2j\pi(\mu x + vy)} dudv$$

(**12)**

El teorema de corte central utiliza la ecuación 9 para relacionar las proyecciones $P_\theta(t)$ con la transformada de Fourier, como se muestra en la figura **15-1**.

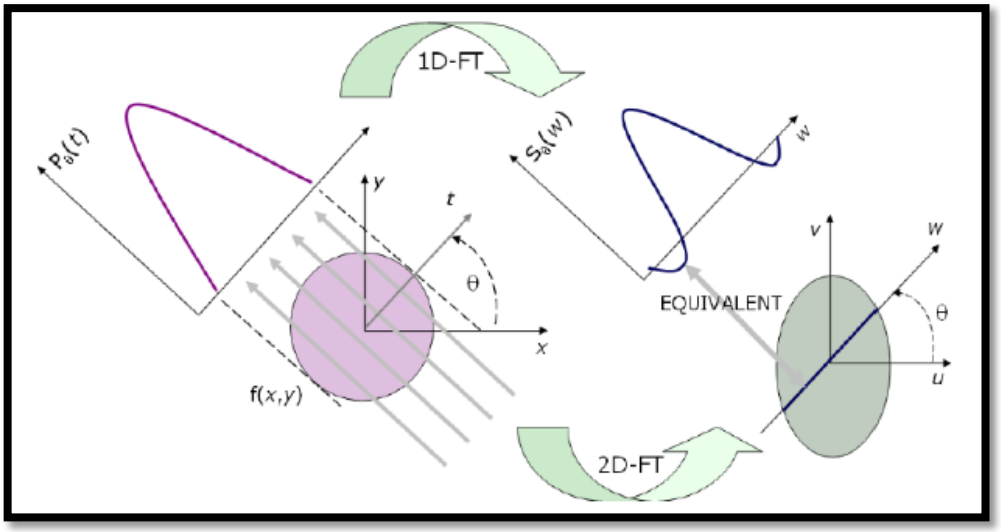



**Figura 15-1.** Teorema de Corte Central.
**Realizado por:** (Molina, 2012)

$$F(u,0) = \int_{-\infty}^{\infty} \mathcal{P}_{\theta=0}(x) e^{-2j\pi\mu x} dx \quad \textbf{(13)}$$

$$F(u,0) = TF\{\mathcal{P}_{\theta=0}(x)\} = S_{\theta=0}(u) \quad \textbf{(14)}$$

Analizando la ecuación 13, la transformada de Fourier en dos dimensiones $F(u,v)$ definida en el eje $v = 0$ del plano, coincide con la Transformada de Fourier de una proyección según su trayectoria $\theta = 0$ (Molina, 2012).

$$F\{\mathcal{P}(x,\theta+\alpha)\} = F(u,v+\alpha) \quad \textbf{(15)}$$

Según este teorema, la ecuación 14 establece que si se tiene todas las proyecciones de una función $f(u,v)$ en coordenadas polares se puede obtener la función $f(x,y)$ en coordenadas cartesianas calculando la transformada inversa de Fourier bidimensional como se muestra en la **figura 15-1**. Por lo tanto, con la ecuación **15** demuestra que, la transformada de Fourier de una proyección paralela de una distribución $f(x,y)$ según un ángulo θ es igual a los valores de la transformada de Fourier 2D de la distribución en la recta que pasa por el origen formando el mismo ángulo θ con el eje $u$.

$$f(x,y) = F_2^{-}(s_\theta(u)) \quad \textbf{(16)}$$

Partiendo de los resultados del Teorema (ecuación 16) existen 2 tipos de métodos que permiten la reconstrucción de imágenes invirtiendo el Teorema de Radon con el menor costo computacional estas son: Transformada inversa de Fourier y la Retroproyección Filtrada.

**TRANSFORMADA INVERSA DE FOURIER**

También se lo denomina como método directo de Fourier, establece que la transformada de Fourier unidimensional de una proyección $f(x,y)$ (ecuación 17) corresponde a una línea que

cruza el origen del espacio, las mismas que son interpoladas para obtener una representación cartesiana para esto es necesario tomar una transformada inversa de Fourier y obtener una imagen del objeto estudiado $f(x,y)$ (Giraldo et al., 2008).

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y)e^{2\pi(ux+vy)}dxdy \quad \textbf{(17)}$$

Con la ecuación **13** se obtiene la ecuación de la Proyección con la convolución de la proyección, definida por la ecuación **18**:

$$P_\theta(w) = \int_{-\infty}^{\infty} p_\theta(t)e^{2\pi iwt}dt \quad \textbf{(18)}$$

El cual permite reconstruir una imagen idéntica siempre y cuando las proyecciones pasen por un proceso denominado interpolación para un muestreo cartesiano, sin embargo, para altas frecuencias se tiene un número menor de muestras que genera un mayor error de interpolación y por lo tanto se observa la presencia de artefactos en la imagen, como se observa en la figura **16-1** (Molina, 2012).

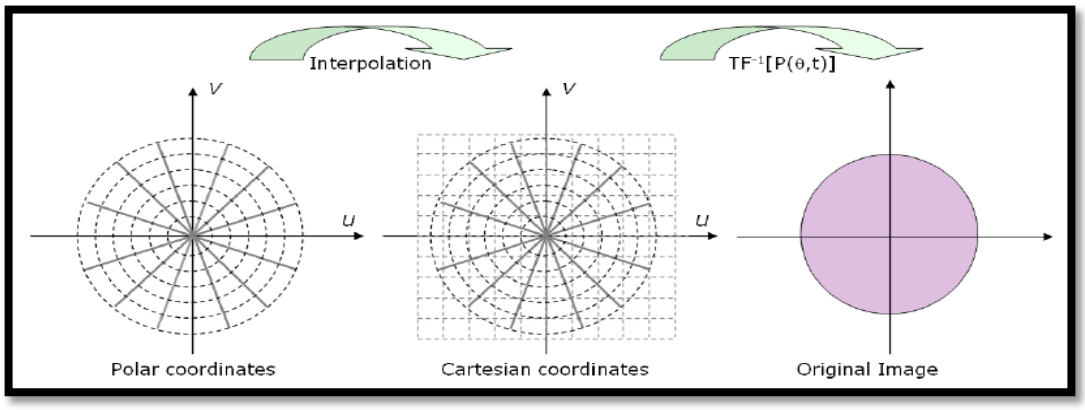



**Figura 16-1.** Método de Reconstrucción con la Transformada Inversa de Fourier 2D.
**Realizado por:** (Molina, 2012)

## 1.4. Retroproyección Filtrada (FBP)

La retroproyección consiste en generar una versión borrosa de una función debido a que no puede reconstruir por si sola a la función original, para corregir este problema existe el método de retroproyección filtrada que consta de dos etapas que se diferencian entre sí: la etapa de la retroproyección y la etapa del filtrado (Molina, 2012).

En la **Etapa de Retroproyección**, una imagen resultante consiste en la suma de las imágenes de retroproyección que se obtiene en diferentes ángulos calculando los valores de atenuación recogidos por los rayos horizontales pero representa una versión de una imagen altamente borrosa

por lo que es necesario una etapa de filtrado para recuperar completamente la imagen original (Molina, 2012).

La **Etapa de Filtrado o Retroproyección Filtrada**, consiste en un método a través del cual se corrige el problema del enfoque difuminado de la función retro proyectada, para esto es necesario aplicar un filtro sobre la transformada de Fourier de la retroproyección para invertir la nueva transformada de Fourier y obtener la función original (Jiménez, 2012).

A partir del teorema definido por la ecuación 15, la expresión de la retroproyección filtrada se puede escribir como 14, que da lugar a la transformada inversa de Fourier del producto de dos transformadas es decir, la transformada en el espacio de la proyección con la transformada de la convolución de la proyección $P_\theta(t)$ con una función cuya transformada de Fourier es igual a la función del filtro $|k|$ en frecuencia, por lo tanto esto es una proyección filtrada (ecuación 15) (Molina, 2012).

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} \mathcal{F}_1\{\widehat{f(p,\theta}\}e^{2\pi itk}|k|dkd\theta \quad (19)$$

$$\mathcal{Q}_\theta(p) = \int_{-\infty}^{\infty} \mathcal{S}_\theta(k)|k|e^{2\pi ikp}dk \quad (20)$$

Mediante la ecuacion 19 y 20 se obtiene la ecuacion 21:

$$f(x,y) = \int_0^{\pi} \mathcal{Q}_\theta\,(xcos\theta + ysen\theta)d\theta \quad (21)$$

La ecuación 21, corresponde a la formalización matemática del proceso de retrotracción de las proyecciones filtradas a lo largo de la línea de proyecciones.

## 1.5. Técnicas de Filtrado

### *1.5.1 Definición de Filtro*

Un filtro es una transformación de una imagen original en una nueva en forma de matrices de dimensiones reducidas, la cual se obtiene celda a celda (pixel a pixel) a partir de los valores de la imagen original y de los factores que constituye el filtro de manera en que mediante un filtro se considera un tamaño de vecindad relacionada con el pixel procesado y se realiza una operación sobre los pixeles abarcados por la vecindad para crear un nuevo pixel, el pixel obtenido se intercambiara por el pixel central de la vecindad escogida (Benito, 2015).

La aplicación de las técnicas de filtrado se relaciona estrechamente con los algoritmos de suavizado con el objetivo de obtener mejores resultados en la calidad de la imagen mediante la

eliminación de cualquier tipo de ruido o artefacto presente en la imagen. El proceso de filtrado se realiza sobre los dominios de frecuencia y espacio, debido a que el uso de la transformada de Fourier nos permite transformar señales entre el dominio espacial y el domino de frecuencias, entre estos procesos tenemos ciertos tipos de filtros que comúnmente son aplicados directamente en los pixeles de la imagen original, las técnicas de filtrado se clasifican en dos grupos: Técnicas de filtrado en el dominio espacial y las Técnicas de filtrado en el dominio de frecuencia
En este trabajo nos centraremos en el desarrollo de filtros que nos permita reducir el ruido, suavizar la imagen, realce de bordes y detectar bordes para mejorar la resolución de una imagen obtenida en el equipo de Tomografía Computarizada.

### *1.5.2. Implementación de Filtros*

#### *1.5.2.1. Definición de Filtrado en el dominio Espacial*

Se considera que las imágenes digitales están compuestas de detalles de frecuencias que varían de bajas a altas frecuencias y posee transiciones rápidas de intensidad que van del obscuro al claro, cuando se completa estos ciclos se denomina frecuencia espacial. En las imágenes las altas frecuencias representan los bordes o detalles, sin embargo, existen operaciones de procesamiento digital para eliminar una banda de frecuencias ya sean altas o bajas, el cual se conoce como filtrado espacial.

El **filtrado Espacial** o procesamiento espacial se aplica sobre un grupo de pixeles vecinos al pixel central sustituyéndolo por un pixel nuevo, este procesamiento se realiza mediante la implementación de 2 procesos denominados procesos de corrección y convolución espacial; el proceso de corrección consiste en transición de la máscara del filtro empleado por toda la matriz o arreglo de la imagen de entrada, y el proceso de convolución espacial recorre la imagen de entrada pixel a pixel ubicando los pixeles resultantes en la imagen de salida. El valor digital del brillo de cada pixel obtenido depende de la vecindad de pixeles de entrada el cual será procesado, esto quiere decir que la convolución espacial utiliza un promedio ponderado del pixel de entrada y de su vecindad para calcular la actividad de frecuencia del brillo del pixel de salida.
La reducción de ruido en una imagen se realiza mediante filtros con enfoque lineal y filtros con enfoques no lineales.

### *1.5.3. Filtros Espaciales Lineales*

El Filtrado Espacial también se lo denomina de paso bajos debido a que mantienes los componentes de frecuencias bajas con el objetivo de suavizar los contrastes atenuando los

componentes de altas frecuencias y consiste en realizar un promedio de los pixeles de la vecindad de 3x3 o 4x4 (figura 16) llamada ventana, considerando que todos los coeficientes de la matriz sean 1.

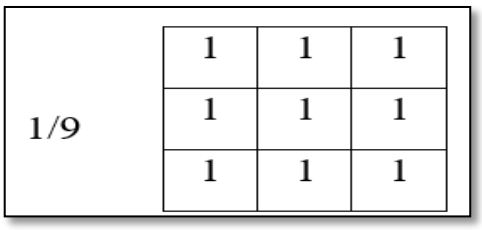



**Figura 17-1.** Vecindad de 3x3 filtrado
**Realizado por:** Villagomez H, Ximena A. 2020

*1.5.3.1. Filtro de Media*

El filtro medina se utiliza con mucha frecuencia ya que es muy efectiva para eliminar el ruido tipo 'sal y pimienta' y preservar los bordes de las imágenes. Este filtro consiste en recorrer toda la imagen pixel a pixel, reemplazando el valor de la intensidad de cada pixel por la mediana de los valores de intensidad de los pixeles vecinos. Esta vecindad de pixeles se conoce como ventana que recorre pixel por pixel sobre toda la imagen (Ferrero et al., 2017, p. 10).

*1.5.3.2. Filtro Laplaciano*

El Filtro Laplaciano es una herramienta que se emplea para destacar las regiones donde se observan cambios bruscos de intensidad de forma que generan escalones de brillo disperso sobre varios pixeles, por lo tanto, el filtro Laplaciano mostrara una respuesta aceptable en el suavizado de la imagen reduciendo la sensibilidad al ruido, desmarcando los bordes y puntos de brillo de la imagen de prueba.

El realce de bordes en una imagen es una operación omnidireccional independiente de su orientación que se basa en el cambio del brillo generado dentro de una vecindad de dimensiones $3x3$ de pixeles en función al pixel central (Aldalur & Santamaría, 2002).

El operador Laplaciano es un filtro que como resultado de su aplicación obtenemos la segunda derivada de una región de pixeles, diferentes entre el pixel central y su vecindad con respecto a sus tonos de grises.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{1.22}$$

Si consideramos la primera derivada (ecuación **1.22**) de la función obtendremos la formulación del operador Laplaciano, realizando la segunda derivada.

$$\nabla^2 x = [f(x+1,y) - f(x,y)] - [f(x,y) - f(x-1,y)]$$

(1.23)

$$\nabla^2 x = [f(x+1,y) - 2f(x,y) + f(x-1,y)]$$

(1.24)

$$\nabla^2 y = [f(x,y+1) - f(x,y)] - [f(x,y) - f(x,y-1)]$$

(1.25)

$$\nabla^2 y = [f(x,y+1) - 2f(x,y) + f(x,y-1)]$$

(1.26)

A partir de**1.23** y **1.25**, se suman los resultados obtenidos (ecuación 1.19 y 121) para definir la ecuación 1.29.

$$\nabla^2 x + \nabla^2 y = [f(x+1,y) - 2f(x,y) + f(x-1,y)] + [f(x,y+1) - 2f(x,y) + f(x,y-1)]$$

(1.27)

$$\nabla^2 x + \nabla^2 y = [f(x+1,y) + f(x-1,y)] - 4f(x,y) + [f(x,y+1) + f(x,y-1)]$$

$$\nabla^2 f(x,y) = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

(1.28)

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

(1.29)

Por lo tanto, la ecuación **1.28** de reduce a la ecuación **1.29** y representa la formulación adecuada del Laplaciano de una imagen digital $f(x,y)$

$$\nabla^2 f(x,y) = \nabla^2 f(x) + \nabla^2 f(y)$$

(1.29)

Para la deducción de la máscara Laplaciana es necesario realizar la expansión de la serie de Taylor, definida por la ecuación **1.30**.

$$f(x+\Delta x, y) \approx f(x,y) \pm \frac{\partial f}{\partial x}(x,y)\Delta x + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(x,y)[\Delta x]^2$$

(1.30)

La ecuación 1.27 es necesario para obtener las segundas derivadas de la función del operador Laplaciano tanto para $x$ como para $y$ respectivamente, como en la ecuación 1.31 y 1.32

$$\frac{\partial^2 f}{\partial x^2}(x,y) = \frac{f(x+\Delta x,y) - 2f(x,y)\Delta x + f(x-\Delta x,y)}{\Delta x^2}$$

(1.31)

$$\frac{\partial^2 f}{\partial y^2}(x,y) = \frac{f(x,y+\Delta y) - 2f(x,y)\Delta y + f(x,y-\Delta y)}{\Delta y^2}$$

(1.32)

Mediante las ecuaciones **1.24** y **1.26** se puede obtener el vector fila y el vector columna, posteriormente sumar los vectores y construir la máscara Laplaciana, de la siguiente forma:

$$\nabla^2 f(x) = [1 \quad -2 \quad 1] \qquad \nabla^2 f(y) = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$\nabla^2 f(x,y) = \nabla^2 f(x) + \nabla^2 f(y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Considerando que las ecuaciones **1.30** pueden ser positivo o negativo, la máscara (matriz) del filtro contiene coeficientes con valores tanto positivo como negativos, siempre y cuando el valor del pixel centrar sea igual al valor de la suma aritmética de sus pixeles vecinos. Por lo tanto, la estructura ideal del filtro de cualquier dimensión garantiza, que la suma de todos los coeficientes de la matriz dé como resultado cero, pero en este caso el resultado debe dividirse para la unidad (ecuación **1.31** y **1.32**).

A continuación, se presenta los tres tipos de máscaras Laplaciana espacial aplicadas para el procesamiento de imágenes de acuerdo a su funcionalidad.

$$1: \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad 2: \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad 3: \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -24 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Definición complementaria para el filtro Laplaciano 2:

$$\mathbf{2:} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{2':} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Durante la aplicación del filtro Laplaciano, se obtiene una imagen semejante pero no idéntica entre el filtro y su complemento. Por lo tanto, la función de la imagen procesada en base al operador Laplaciano se define por la ecuación 1.33.

$$g(x,y) = f(x,y) - c\nabla^2 f(x,y) \tag{1.33}$$

Donde $g(x,y)$ es la función de la imagen restaurada o procesada, $f(x,y)$ representa la función de la imagen de prueba, $c$ es una contante igual a 1 cuyo signo depende del signo del coeficiente centrar de la máscara Laplaciana espacial $\nabla^2 f(x,y)$.

### *1.5.3.3. Filtro Gaussiano*

El filtro gaussiano o filtro de suavizado es un ejemplo de la técnica de filtrado en el dominio espacial lineal y es el resultado de suavizado de una imagen de acuerdo a la forma de una función gaussiana, es muy útil en la reducción o eliminación del ruido, es adecuado en la detección de bordes sin embargo existe el riesgo de perder una gran cantidad de detalles debido a que mediante este filtro no se puede preservar los bordes obteniendo una imagen borrosa.

### *1.5.4. Filtro Espacial no Lineal*

Este tipo de filtros no son funciones lineales de los valores de intensidad de los pixeles d una imagen, se constituye como una técnica de procesamiento que opera sobre un núcleo de pixeles de entrada que rodea al pixel central utilizando un promedio ponderado de los valores del grupo de pixeles de entrada.

#### *1.5.4.1. Filtro Wiener*

Este tipo de filtro utiliza un filtrado inverso mediante el cual se elimina el ruido aditivo e invierte la difuminación de la imagen de entrada (Ferrero et al., 2017).

El filtro Wiener fue propuesta por Norbert Wiener en el año de 1942 con el propósito de reducir al mínimo el ruido que afecta a una señal de entrada de tal manera que la señal de salida del filtro se aproxime a una señal sin presencia de ruido, es decir este tipo de filtro se emplea para obtener una imagen estimada de tal forma que minimice el error cuadrático medio que existe entre la imagen original y la estimada.

Además, el método de filtrado Wiener considera a la imagen y el ruido como un proceso estadístico o aleatorio y el objetivo de implementar este filtro es tratar de encontrar una imagen estimada de tal forma que el error medio al cuadrado entre las imágenes sea de valor mínimo.

**Error Cuadrático Medio** (ECM): es el valor esperado del error al cuadrado, indica la cantidad con la que una imagen estimada difiere de la imagen original durante el proceso de restauración de imágenes, esto se debe a que no se conoce cierta parte de información corrompida por la presencia de ruido y se define con la siguiente ecuación:

$$e^2 = E\{(f'(x,y) - f(x,y))^2\} \tag{1.34}$$

Donde: $e^2$ Error cuadrático medio, $E$ operador del valor esperado de la esperanza matemática, $f(x,y)$ matriz de la imagen original y $f'(x,y)$ es la estimación de la imagen original

A partir de la ecuación 1.34 se puede obtener la varianza de la imagen estimada y el sesgo, reescribiendo la ecuación 1.34 en 1.35:

$$ECM\big(f'(x,y)\big) = E\{(f'(x,y)^2 - 2f'(x,y)f(x,y) + f(x,y)^2)\} \tag{1.35}$$

$$ECM\big(f'(x,y)\big) = E(f'(x,y)^2) - 2E(f'(x,y))f(x,y) + f(x,y)^2$$

$$ECM\big(f'(x,y)\big) = E(f'(x,y)^2) - 2E\big(f'(x,y)\big)f(x,y) + f(x,y)^2 + [E(f'(x,y)]^2 - [E(f'(x,y)]^2$$

$$ECM(f'(x,y) = E[f'(x,y)^2] - [E(f'(x,y)]^2 + E(f'(x,y)^2)2E(f'(x,y))f(x,y) + f(x,y)^2$$

(1.36)

Conocemos la ecuación de la varianza:

$$Var(x) = E[f[x] - f'[x]^2]$$

(1.37)

Sustituimos 1.37 en la ecuación 1.36

$$ECM\big(f'(x,y)\big) = Var\big(f`(x,y)\big) + (E(f'(x,y) + f(x,y))^2$$

**(**1.38**)**

Donde: $Var\big(f`(x,y)\big)$ es la Varianza de la imagen estimada y $(E(f'(x,y) + f(x,y))^2$representa el sesgo, es el cuadrado de la diferencia entre la imagen original y la estimada.

Con la ecuación 1.38 el ECM establece que la varianza estimada de la imagen estimada y el sesgo deben ser lo más pequeño posible para que la imagen estimada se aproxime a una fiel copia de la imagen original.

**Deducción de la Ecuación del Filtro Wiener**:

Tomando en cuenta que el objetivo general de este filtro es obtener una imagen estimada $f'(x,y)$ fiel copia de la imagen de la imagen original $f(x,y)$ siempre y cuando el valor del ECM sea lo más pequeño posible.

Sin embargo para encontrar una solución a la ecuación 1.34 que nos permita minimizar al máximo el ECM es necesario asumir 2 limitaciones: los valores de la escala de grises que representa a la imagen estimada es una función lineal de los valores de la escala de grises de la imágenes degradada y que el ruido, además la imagen no tienen correlación entre sí y uno de ellos tiene una media igual a 0, la estimación de la imagen original se da en el dominio de frecuencia, teniendo en cuanta estas suposiciones será posible minimizar el ECM, estas suposiciones nos permite expresar expresa la forma siguiente:

$$E[f(x,y)f'(x,y)] = E[f(x,y)]E[f'(x,y)] = 0$$

(1.39)

Se considera que la función de la imagen degradada o corrompida por cualquier tipo de artefacto es igual a:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

(1.40)

Donde: $h(x,y)$ es función de degradación y $n(x,y)$ la función del ruido que afecta a la imagen digital $f(x,y)$. Por lo tanto, el estimado de la imagen original deberá ser igual a 1.8:

$$f'(x,y) = w(x,y) * g(x,y)$$

(1.41)

Donde: $w(x,y)$= función del filtro Wiener

Si la ecuación 1.2 la separamos en factores y con la ecuación 1.8 obtendremos la ecuación siguiente:

$$e^2 = E[f(x,y)^2] + E[f'(x,y)^2] - 2E[f(x,y)f'(x,y)]$$

Sustituyendo 1.41 se obtiene 1.42

$$e^2 = E[f(x,y)^2] + E[\{w(x,y) * g(x,y)\}^2] - 2E[f(x,y)\{w(x,y) * g(x,y)\}]$$

(1.42)

Sustituimos la ecuación 1.40 en la ecuación 1.42, tendremos:

$$= E[f(x,y)^2] + E\left[\{w(x,y) * (h(x,y) * f(x,y) + n(x,y))\}^2\right] - 2E[f(x,y)\{w(x,y) * (h(x,y) * f(x,y) + n(x,y))\}]$$

$$= E[f(x,y)^2] + E[\{w(x,y) * h(x,y) * f(x,y) + w(x,y) * n(x,y)\}^2] - 2E[f(x,y)\{w(x,y) * h(x,y) * f(x,y) + w(x,y) * n(x,y)\}]$$

$$= E[f(x,y)^2] + E[\{w(x,y) * h(x,y) * f(x,y) + w(x,y)n(x,y)\}^2] - 2E[f(x,y) * \ w(x,y) * h(x,y) * f(x,y) + f(x,y) * w(x,y) * n(x,y)]$$

(1.43)

Si conocemos que $E[f(x,y)^2] = R_f(x,y)$ y $E[n(x,y)^2] = R_n(x,y)$ sustituimos en 1.44 y la reducimos en:

$$= E[f(x,y)^2] + \{E * w(x,y) * h(x,y) * f(x,y) + E * w(x,y)n(x,y)\}^2 - [2E * f(x,y) * w(x,y) * h(x,y) * f(x,y) + 2E * f(x,y) * w(x,y) * n(x,y)]$$

$$= R_f(x,y) + E[w(x,y) * h(x,y) * f(x,y)]^2 + E[w(x,y)n(x,y)]^2 - [2E * f(x,y) * w(x,y) * h(x,y) * f(x,y) + 2E * f(x,y) * w(x,y) * n(x,y)]$$

$$= R_f(x,y) + E[w(x,y) * h(x,y) * f(x,y)]^2 + [w(x,y) * E * n(x,y)]^2 - [2E * f(x,y)^2 * w(x,y) * h(x,y) + 2E * f(x,y) * w(x,y) * n(x,y)]$$

$$= R_f(x,y) + \big(w(x,y) * h(x,y)\big)^2 * E * (f(x,y)^2) + w(x,y)^2 * E * (n(x,y)^2) - [2E * f(x,y)^2 * w(x,y) * h(x,y) + 2E * f(x,y) * w(x,y) * n(x,y)]$$

$$e^2 = R_f(x,y) + \big(w(x,y) * h(x,y)\big)^2 * R_f(x,y) + w(x,y)^2 * R_n(x,y) - 2\big[R_f(x,y) * w(x,y) * h(x,y)\big]$$

(1.44)

De otra forma el ECM puede ser escrito de forma matricial como 1.45:

$$e^2 = \big[R_f\big] + [W][H]\big[R_f\big][W]^T[H]^T + [W][R_n][W]^T - 2[W][H]\big[R_f\big]$$

(1.45)

Si derivamos la ecuación 1.40 con respecto a $w(x,y)$ he igualamos a 0 encontraremos el filtro Wiener Optimo para la restauración de imágenes:

$$\frac{d}{dw(x,y)}\Big[R_f(x,y) + \big(w(x,y) * h(x,y)\big)^2 * R_f(x,y) + w(x,y)^2 * R_n(x,y)$$

$$- 2\big[R_f(x,y) * w(x,y) * h(x,y)\big]\Big] = 0$$

$$2 * w(x,y) * \ h(x,y)^2 * \ R_f(x,y) + 2 * w(x,y) * \ R_n(x,y) - 2 * h(x,y) * R_f(x,y) = 0$$

$$2w(x,y)\big[h(x,y)^2 * \ R_f(x,y) + \ R_n(x,y)\big] = \ 2 * h(x,y) * R_f(x,y) \qquad \text{Se suprime 2}$$

$$w(x,y) = \frac{h(x,y) * R_f(x,y)}{h(x,y)^2 * R_f(x,y) + R_n(x,y)}$$

(1.46)

La ecuación 1.42 se puede escribir de forma matricial:

$$\big[W_{opt}\big] = \big[R_f\big][H]^T\Big[[H]\big[R_f\big][H]^T + [R_n]\Big]^-$$

Suponemos que H es invariante por desplazamiento lineal y f es estacionario, la aplicación de transformadas de Fourier conduce a la ecuación 1.47

$$W(u,v) = \frac{H^*(u,v)S_f(u,v)}{|H(u,v)|^2 S_f(u,v) + S_n(u,v)}$$

(1.47)

Donde: $S_f(u,v)$ densidad espectral de potencia del ruido, $S_n(u,v)$ densidad espectral de la imagen original y $H^*(u,v)$ es el complejo conjugado de $H(u,v)$.

Si tomamos en cuenta que el producto de una cantidad compleja con su conjugado es igual a la magnitud del complejo al cuadrado y dividimos por $S_f(u,v)$ al numerador y denominar tendremos:

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}}$$

(1.48)

Por lo tanto, si la ecuación 1.48 multiplicamos y dividimos por $H(u,v)$ tenemos la siguiente expresión:

$$W(u,v) = \frac{1}{H(u,v)} \frac{\big|H(u,v)^2\big|}{\big|H(u,v)^2\big| + \frac{S_n(u,v)}{S_f(u,v)}}$$

(1.49)

La ecuación 1.49 es conocida como el filtro Wiener, por lo tanto, es posible definir la ecuación 1.41 función de la imagen estimada de la original en el dominio de frecuencia de la siguiente forma:

$$F'(u,v) = \left[\frac{1}{H(u,v)} \frac{\big|H(u,v)^2\big|}{\big|H(u,v)^2\big| + \frac{S_n(u,v)}{S_f(u,v)}}\right] * G(u,v)$$

(1.50)

Dónde: la expresión $\frac{S_n(u,v)}{S_f(u,v)}$ se la conoce como la relación señal-ruido que representa el margen que existe entre la información de la imagen y el ruido.

Sin embargo debemos tomar en consideración que muchas de las veces no se conoce con exactitud la información que posee una imagen de original como para calcular la relación señal-ruido, por esta razón es necesario sustituirla por una constante $K$ que mediante interacciones repetidas de prueba el error llega a un mejor resultado, por lo contrario si se tiene la matriz de datos de referencia la relación señal-ruido puede ser calculado y no es necesario realizar múltiples interacciones (Vettorazzi, 2007).

### *1.5.4.2. Método del Filtrado Lagrangiano*

Para la restauración de imágenes es preciso la minimización de un criterio de rendimiento que nos permita obtener la estimación de la imagen $\hat{f}(x,y)$, por lo cual, es necesario la utilización de funciones de costes por mínimos cuadrados dependiendo de la restricción que se le aplique a la función de coste.

Utilizando la ecuación 1.40 de la función de degradación de la imagen, se despeja la función del ruido obteniendo la ecuación 1.51

$$g(x,y) - H(x,y)f(x,y) = n(x,y) \tag{1.51}$$

De manera que la ecuación 1.18 se reescribe como (ecuación 1.49) si no se tiene ningún conocimiento sobre la naturaleza del origen de $n(x,y)$, por lo tanto se aplica una función de coste que nos permita encontrar la función de la imagen estimada $\hat{f}(x,y)$, tal que, $H(x,y)\hat{f}(x,y)$ acerque a $g(x,y)$ al sentido de mínimos cuadrados, haciendo que la norma $n(x,y)$ sea lo menor posible.

$$g(x,y) - H(x,y)\hat{f}(x,y) = n(x,y) \tag{1.52}$$

Por lo tanto, se le aplica el cuadrado de la norma a ambos lados de la ecuación 1.52 respectivamente, obteniendo 1.53 para minimizar o reducir el valor de $n(x,y)$

$$\left\|g - H\hat{f}\right\|^2 = \|n\|^2 \tag{1.53}$$

Por lo tanto, a 1.53 se le conoce como función de restricción, la cual, trata de minimizar la función $\left\|Q\hat{f}\right\|^2$ sujeta a la restricción 1.53, donde $Q$ es el operador lineal de $f$. Para facilitar el proceso minimización adecuada de $n$, podemos utilizar el método de los ***Multiplicadores de Lagrange,*** establece que si la función $J(\hat{f}) = \left\|Q\hat{f}\right\|^2$ (ecuación **1.54**) tiene un punto extremo (mínimo o

máximo), la cual, está sometida a una restricción (ecuación **1.53**) igualada a cero, entonces existe una escalar $\lambda$ conocido como operador Lagrangiano.

Por lo tanto, la ecuación 1.53 se reescribe de la forma 1.55 y se le aplica el operador $\lambda$ obteniendo 1.56.

$$\left\|g - H\hat{f}\right\|^2 - \|n\|^2 = 0 \tag{1.55}$$

$$\lambda\left(\left\|g - H\hat{f}\right\|^2 - \|n\|^2\right) = 0 \tag{1.56}$$

Si le añadimos la función $J(\hat{f})$ (**1.54**) a la ecuación 1.56, se obtiene la ecuación:

$$J(\hat{f}) = \left\|Q\hat{f}\right\|^2 + \lambda\left(\left\|g - H\hat{f}\right\|^2 - \|n\|^2\right) \tag{1.57}$$

La ecuación 1.57 busca una función que minimice la función de coste $J(\hat{f})$, para ello, es necesario derivar 1.57 con respecto $\hat{f}$.

$$\frac{\partial(J(\hat{f})}{\partial\hat{f}} = \frac{\partial}{\partial\hat{f}}\left[\left\|Q\hat{f}\right\|^2 + \lambda\left(\left\|g - H\hat{f}\right\|^2 - \|n\|^2\right)\right] = 0 \tag{1.58}$$

$$\frac{\partial}{\partial\hat{f}}\left[\left\|Q\hat{f}\right\|^2\right] + \frac{\partial}{\partial\hat{f}}\left[\lambda\left(\left\|g - H\hat{f}\right\|^2 - \|n\|^2\right)\right] = 0$$

$$\frac{d}{d\hat{f}}\left[\left\|Q\hat{f}\right\|^2\right] + \lambda\frac{d}{d\hat{f}}\left[\left(\left\|g - H\hat{f}\right\|^2 - \|n\|^2\right)\right] = 0$$

Después de derivar 1.29 y tomando en cuenta que la norma Euclidiana de una función $\left\|g - H\hat{f}\right\|^2 = \left(g - H\hat{f}\right)^T\left(g - H\hat{f}\right)$, se aplica en la ecuación obtenida después de derivar 1.55.

$$2Q^T\left(Q\hat{f}\right) + 2\lambda H^T\left(g - H\hat{f}\right) = 0 \tag{1.59}$$

$$2Q^TQ\hat{f} + 2\lambda H^Tg - 2\lambda H^TH\hat{f} = 0 \tag{1.60}$$

Si despejamos $\hat{f}$ de la ecuación 1.560 obtenemos

$$\widehat{2f}(Q^T\,Q\, - \lambda H^T\,H) + 2\lambda H^Tg = 0 \tag{1.61}$$

$$\hat{f} = \frac{2\lambda H^Tg}{2(Q^T\,Q\, - \lambda H^T\,H)} \tag{1.62}$$

Suprimimos el 2 de la ecuación 1.62, obtenemos

$$\hat{f} = \frac{\lambda H^Tg}{Q^T\,Q\, - \lambda H^T\,H} \tag{1.63}$$

Por lo tanto 1.63 se puede expresar de la forma

$$\hat{f} = (\lambda H^T g)(Q^T Q - \lambda H^T H)^{-1} \quad (1.64)$$

$$\hat{f} = (\lambda H^T g)(Q^T Q - \lambda H^T H)^{-1}\left(\frac{1}{\lambda}\right) \quad (1.65)$$

Para simplificar el operador de Lagrange de 1.65 se le multiplica por $\gamma = \frac{1}{\lambda}$

$$\hat{f} = (H^T g)\left(\left(\frac{1}{\lambda}\right) Q^T Q - \left(\frac{1}{\lambda}\right)\lambda H^T H\right)^{-1} \quad (1.66)$$

$$\hat{f} = (H^T g)(\gamma Q^T Q - H^T H)^{-1} \quad (1.67)$$

La ecuación 1.67 es la función óptima para encontrar la imagen estimada, ya que $\gamma$ es un valor que se puede ajustar a la restricción.

*1.5.4.3. Método del Filtrado Inverso*

El método de filtrado inverso permite una aproximación sobre una imagen original que se visualizan borrosas y fuera de foco, se basa en obtener una imagen estimada ideal sin imperfecciones dividiendo la transformada de la imagen degradada por una función de degradación conocida suponiendo que se conoce una función de desenfoque o *a priori* (Vettorazzi, 2007).

Se conoce que una imagen restaurada en su versión en el dominio de frecuencia está definida por 1.41:

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \quad (1.41)$$

Donde: $F'(u,v)$ Representa a la estimación de la imagen original, $G(u,v)$ Transformada de Fourier de la función de degradación, $H(u,v)$ es la Transformada de Fourier de la imagen degradada.

Si se conoce la función $H(u,v)$ ya sea por la observación o experimentación, la imagen restaurada mejorará, pero no será la ideal, por lo tanto, si se tiene un mejor conocimiento del fenómeno que causa la degradación la respuesta del filtro inverso sobre la imagen de prueba será mejor.

Sabemos que la transformada de Fourier que representa a la imagen degradad en el dominio de frecuencia se representa de la siguiente forma es igual a:

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \quad (1.68)$$

Sustituyendo la ecuación 1.42 en la ecuación 1.41, obteniendo la siguiente expresión:

$$F'(u,v) = \frac{H(u,v)*F(u,v)+N(u,v)}{H(u,v)} \quad (1.69)$$

$$F'(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \quad (1.70)$$

Si se conoce con exactitud la función de degradación y $N(u,v)$=0 (ruido) será posible tener una fiel aproximación de la imagen original por lo contrario si la imagen contiene ruido es muy probable que en el momento de analizar la imagen no se obtenga buenos resultados ya que la presencia de ruido corresponde a una función aleatoria cuya transformada de Fourier no se conoce, por lo tanto el efecto del filtro inverso solo se podrá observar cuando el ruido se presente en altas frecuencia.

Este tipo de filtro sobre sale de otros métodos de restauración desarrollados por su mayor sencillez de aplicación, además presenta ciertas limitaciones y puede aplicarse en casos específicos como, por ejemplo, se puede aplicar en imágenes fuera de foco y libres de contaminación de ruido.

## 1.6. Formato gráfico

### *1.6.1. Definición de Imágenes Digitales*

Una imagen representa la descripción de la variación de un parámetro sobre un plano en dos dimensiones, convirtiéndose en imágenes de luz capaces de ser valoradas por el ojo humano. Matemáticamente una imagen se representa como una función bidimensional $f(x,y)$ donde $x\ e\ y$ son las coordenadas espaciales y $f$ es la amplitud en cualquier punto de coordenadas del espacio $(x,y)$ llamada intensidad o escala de grises de la imagen.

Una imagen digital se denomina de esta forma cuando los valores de $x, y$ ý $f$ son cantidades discretas, lo cual se obtiene mediante el proceso de digitalización de la imagen análoga con valores continuos con respecto a $x$ ý $y$ , generando una imagen en formato digital a partir de dos procesos como el muestreo y la cuantización ya que son una herramienta matemática muy importante para el procesamiento de imágenes.

Para obtener una imagen digital es importante convertir la información continua obtenida por los detectores por medio del muestreo y la cuantización, obteniendo como resultado una matriz $MxN$, los valores de las coordenadas $(x,y)$ son convertidos en valores discretos o valores enteros que representan la posición de los pixeles a lo largo de la matriz de la imagen como se muestra en la figura **18-1**:

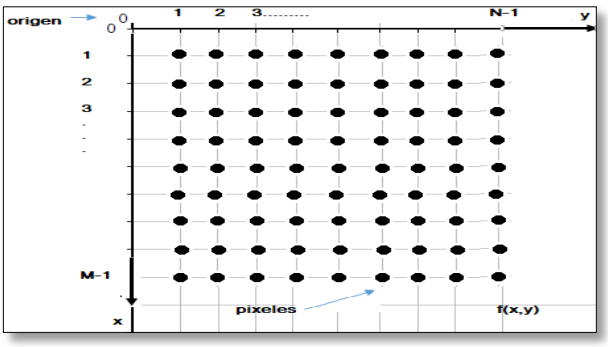



**Figura 18-1.** Representación de la Imagen
**Realizado por:** Villagomez H, Ximena A. 2020

La matriz de la imagen $MxN$ (notación **22**) se puede representar de la siguiente forma teniendo en cuenta la notación matemática que representa a una imagen digital:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \downarrow & \downarrow & \downarrow & \downarrow \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad \textbf{(71)}$$

Una imagen digital es una representación bidimensional de un objeto mediante una matriz numérica compuesta de un número finitos de elementos conocidos como pixeles los cuales tienen una localización especificada y un valor propio de intensidad o niveles de grises que varía en ese punto entre 0 y 255 (Benito, 2015).

*Una imagen en escala de grises de define mediante una función* (**notación 72**)*:*

$$f(x,y):\ \Omega \subset \mathbb{R}^2 \rightarrow \{0,\dots,255\}$$

**(72)**

*Donde* $f(x,y)$*= intensidad positiva de cada punto de la imagen*

La representación más común es como una matriz $NxMxk$ donde $N$ $y$ $M$ representan las dimensiones de la imagen y $k$ es el número de bits que corresponde a los colores en escala de grises de la imagen, por lo tanto, para el proceso de digitalización se requiere el número de bits para guardar una imagen digitalizada que define como la ecuación 73:

$$b = NxMxk$$

**(73)**

La ecuación **73** nos proporciona el tamaño de una imagen digital lo cual es indispensable para el procesamiento de imágenes y el número de niveles de grises como un entero se define con la ecuación **74**.

$$L = 2^k$$

**(74)**

Donde L representa los niveles enteros o discretos de la escala de grises permitidos para cada pixel como por ejemplo si tenemos una imagen de 8 bits $L = 2^8 = 256$ valores que se distribuyen para cada pixel en un intervalo de 0 a 255 y de tamaño $1024x1024$, podemos observar que la imagen tendrá 8,388.608 bits acuerdo a la ecuación **73**, lo cual es importante para el procesamiento de imágenes.

### *1.6.2. Los Criterios para la elección de un Formato Digital*

La ecuación **74** nos proporciona el tamaño de una imagen digital lo cual es indispensable para el procesamiento de imágenes, por lo general una imagen compuesta por más de $1000x1000\ pixeles$ son consideradas como las imágenes de mejor calidad, sin embargo en el procesamiento de imágenes es habitual utilizar imágenes con 256 niveles de cuantización en la escala de grises debido a fácil manejo de datos para que una computadora guarde la información, además, que el tamaño de paso 1/256 luminosidades es muy pequeño de lo que el ojo humano puede visualizar.

Una imagen digitalizada se presenta en un formato que pueda ser manipulado por varias aplicaciones es decir que los datos de la imagen debes estar codificados de forma estándar para que puedan ser interpretados por otras aplicaciones de acuerdo a su conveniencia, por lo tanto el mejor formato será aquel que se adopte a las necesidades del usuario como un buen manejo por parte del sistema gráfico, mayor nitidez menor tiempo de visualización, menor espacio de almacenamiento y rapidez de descompresión (Aguilar, 1995).

### *1.6.3. Tipos de Formato gráficos de Imagen Digital*

Los formatos gráficos digitales son archivos donde se almacena cierta información útil para conformar una imagen de forma independiente, en la actualidad existen múltiples tipos de formatos de archivos gráficos, pero ninguno supera al otro debido a que cada uno tiene sus propias características y funciones, por lo mismo no se tiene la certeza de que uno es mejor que el otro para establecer como un formato estándar.

Analizando la situación se determina que el mejor formato es aquel que se adapte a las necesidades del manejo del procesador gráfico como si tiene mayor nitidez, menor tiempo para su visualización y menor espacio para su almacenamiento. Sin embargo, todos los formatos de

archivos gráficos se rigen a ciertas reglas que en algunos casos son más complejas que en otras, lo que define la estructura del formato ya sea similar a otros formatos, más flexible o fija.

En el análisis de las diferentes estructuras de los formatos de archivos gráficos se tiende a encontrarse con problemas en cuanto a la información presente en la imagen en el momento de ser editada permanezca constante, otro problema es que en el momento de editar la imagen se requiere de mucho tiempo para un análisis secuencial de los datos debido a que es muy grande.

Otro problema es el cambio de formato de archivo de imágenes es decir es difícil que el formato de una imagen en blanco y negro se adapte a una imagen de 16 o 256 colores sin embargo a medida que se desarrollen el campo de los formatos de archivos digitales se emplean estas consideraciones tratando de compensar los requerimientos propuestos.
Los formatos gráficos se clasifican de acuerdo a la forma en que los datos de una imagen son propiamente almacenados y visualizados en una pantalla. Existen 2 categorías de clasificación: los formatos reticulares y formatos vectoriales.

### *1.6.3.1. Formato Reticular-Pixelares*

El formato Reticular-Pixelares se caracteriza por componer de una serie de elementos de imágenes o pixeles que cubren un área completa de visualización en la pantalla, los pixeles no necesariamente se relacionan unos con otros (figura **19-1 b**), estas imágenes son utilizadas muy a menudo por sus principales ventajas en cuanto a su calidad de resolución como:

- La visualización de los datos en la pantalla es más rápida permitiéndonos visualizar partes o secciones que a simple vista no se pueden observar.
- Es fácil la resolución de la salida de los datos sobre un dispositivo como monitor de computador.

### *1.6.3.2. Formato Vectorial*

Un formato vectorial se caracteriza porque involucra a elementos compuestos por segmentos lineales en lugar de pixeles para conformar una imagen, es decir, una imagen se conforma por polígonos unidos mediante puntos que para su interpretación digital se miden la distancia que existe entre los puntos del polígono.

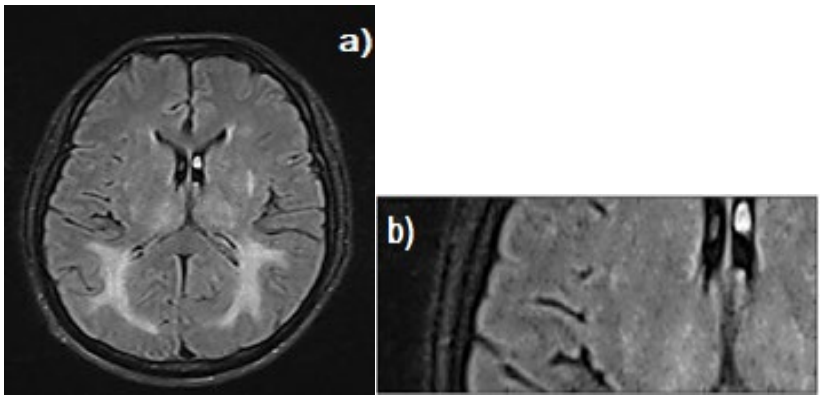



**Figura 19-1.** a) Imagen Digital, b) Parte de la Imagen pixelado.
**Realizado por:** Villagomez H, Ximena A. 2020

**FORMATOS DE IMÁGENES**

En la tabla **2-1** se presenta los formatos gráficos reticulares siendo los más utilizados en el campo del procesamiento de imágenes digitales y los mismos que son soportados por la mayoría de aplicaciones gráficas para PC.

**Tabla 2-1:** Formatos gráficos reticulares más utilizados en el procesamiento de Imágenes.

| Formato | Extensión | Características |
|---|---|---|
| BMP | .bmp | Bitmap o Mapa de bits para Windows es un formato no comprimido y sin perdidas de información y es utilizado por Windows para imágenes de dimensiones pequeñas. |
| FIF | .fif | Fue diseñado para importar imágenes en editores gráficos y permite al programador satisfacer sus necesidades, debido a que son muy complejo pocos programas decodificadores hacen uso de toda la información presente en el archivo. |
| PCX | pcx | Fue desarrollado por *Zsoft Corporation* implementado en su programa PC *Paintbrush* se consideró que era un formato estándar necesario para el intercambio de imágenes entre aplicaciones con un ahorro de espacio de almacenamiento. |

**Fuente:** (Aguilar, 1995)
**Realizado por:** Villagomez Herrera, Ximena, 2020.

## 1.7. Formato Gráfico utilizado

### *1.7.1. Criterio para la elección de un formato*

Cabe señalar que se consiguió imágenes en tono de grises grabadas en un formato de mapa de bits sin comprimir y sin ningún dato adicional, esto nos permite ahorrar el espacio de memoria que ocupa los bytes de acuerdo a la definición de la paleta de colores y otras características, ya que

posee un soporte de 256 tonos de gris-8 bits y se conoce que hay datos que no son de utilidad para la utilidad de la tesis.

Al utilizar imágenes con este tipo de formato es importante conocer que tiene un soporte de aplicaciones de uso común bajo Windows, debido a que si otras aplicaciones no reconocen el archivo generado por el programa procesador de imágenes. Además, nos permite ahorra tiempo y disminuir la complejidad en el desarrollo de los programas computacionales para codificar y descodificación.

La figura **20-1** representa una imagen en formato BMP, con la siguiente información de acuerdo a la estructura de su formato obtenida mediante la función *imageinfo* de Matlab: tamaño de archivo 1081134, anchura 571, altura 630, desplazamiento de datos de imagen 54, tamaño del encabezado del mapa de bits 108108, tipo de compresión ninguna, resolución horizontal 2835 y resolución vertical 2835.

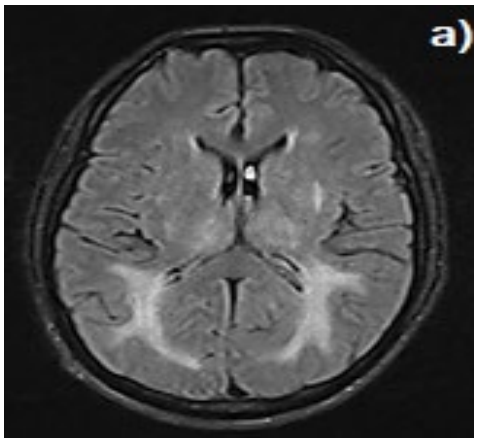



**Figura 20-1.** Imagen en formato BMP.
**Realizado por**: Villagomez H, Ximena A. 2020.

Para que tenga un buen rendimiento el programa computacional que se desarrollará en este Trabajo de Titulación se requiere imágenes para procesar en formato gráfico de tipo reticular, además deben cumplir los siguientes requerimientos:

- Identificar el tipo de archivo y disponibilidad de la información completa sobre el formato es decir obtener una descripción detallada sobre la forma de escribir el archivo de estudio.
- Tamaño de la imagen
- Fecha y nombre original de la imagen.
- Posibilidad del lenguaje de programación mediante el cual se desarrollará el programa soporte el tamaño del formato gráfico de la imagen.
- El formato gráfico debe ser soportado por múltiples aplicaciones gráficos presentes en nuestro PC, esto permitirá que las imágenes procesadas en nuestro programa sean recuperadas por otros programas.

♦ Emplear una menor complejidad computacional para la codificación de una imagen y que tenga la posibilidad de soportar una imagen de 256 tonos de grises sin problemas.

### *1.7.2. Factibilidad de utilizar un formato de archivo gráfico*

El formato estándar DICOM es el mecanismo de codificación, almacenamiento, transmisión e impresión de imágenes aceptado universalmente por la comunidad médica, además contiene una parte del fichero que almacena información del paciente, tipo de escáner, las dimensiones de la imagen entre otras y otra parte contiene todos los datos correspondientes a la imagen tridimensional almacenada. Los datos de la imagen en formato DICOM pueden ser comprimidas a través de otros formatos con o sin pérdida de datos como (Fenoll, 2010):

#### *1.7.2.1. Formato BMP*

**Formato BMP**

**BMP:** Este formato gráfico que se conoce como Windows Bitmap ya que sirve a base de Microsoft Windows, por lo tanto de lo denomina como mapa de bits independiente del dispositivo, fue desarrollado por Windows y es un registro de los bits reales en el archivo, no pueden ser fácilmente ajustados si se cambian la resolución de la pantalla a demás son rápidos para ser cargados y desplegados por Windows (Aguilar, 1995).

### *1.7.3. Estructura del Formato Gráfico*

Los archivos gráficos BMP tienen una estructura que prescriben ciertas reglas esenciales donde parte de la información debe estar en una sección denominada cabecera que va al inicio del archivo seguida por los datos de la imagen. De acuerdo a la información limitada acerca de estos formatos se concluye que los archivos gráficos BMP tienen una estructura que consta de 3 0 4 partes como el encabezado o cabecera principal, cabecera de información del mapa de bits, la paleta que es opcional y el cuerpo de datos de la imagen.

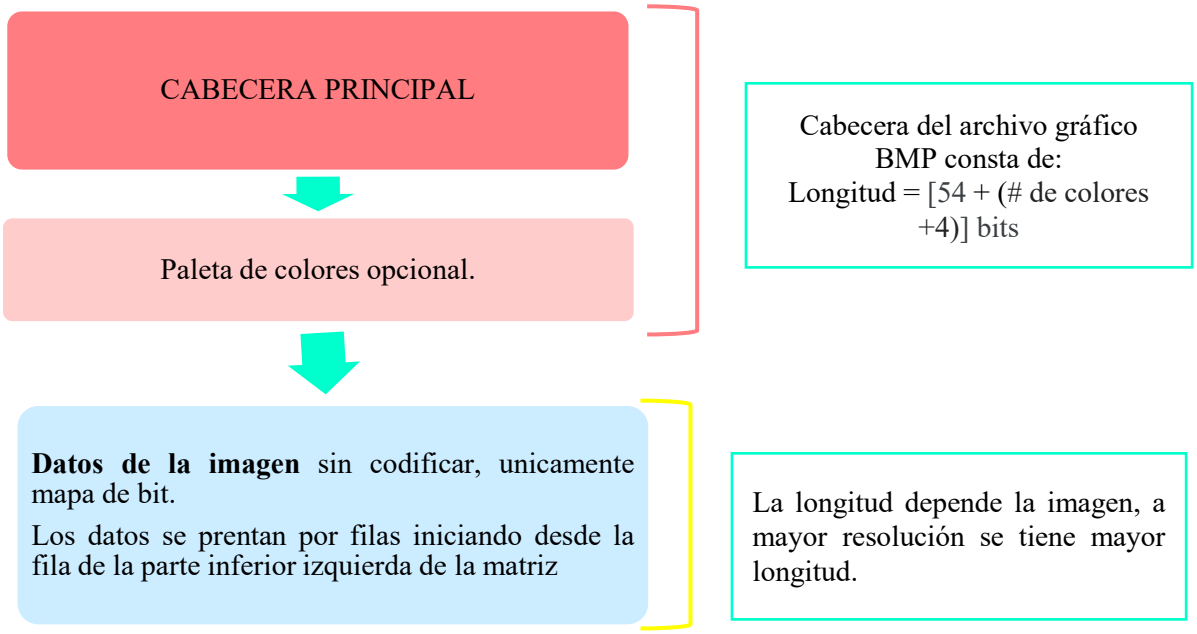



**Figura 21-1.** Estructura del Formato Gráfico BMP.
**Realizado por:** (Aguilar, 1995)

La **cabecera** de un archivo MBP está constituida por 54 bytes de longitud que conforman la cabecera principal del archivo seguida por la información correspondiente a la paleta de colores.

**Tabla 3-1:** Estructura de la cabecera principal del formato BMP.

| Byte | | Tamaño | Nombre del campo | Descripción |
|---|---|---|---|---|
| dec | hex | | | |
| ---- | ---- | 2 | Siglas del formato | BM42 4D(hex) indica el archivo BMP |
| 18 | 12 | 4 | Numero de filas | Ancho de la imagen con su valor maximo en X+1. |
| 22 | 16 | 4 | Numero de columnas | Altura de la imagen con su valor maximo en Y+1. |
| 28 | 1C | 1 | Bits por pixel | Numero de bits de color por pixel |
| 38 | 26 | 4 | Resolucion horizontal | Resolucion horizontal del dispositivo usado para generar la imagen |
| 42 | 2ª | 4 | Resolucion vertical | Resolucion vertical del dispositivo usado para generar la imagen |
| 46 | 2E | 4 | Tonos de grises | Numero de tonos de grises en la imagen |

**Fuente:** (Aguilar, 1995)
**Realizado por:** Villagomez Herrera, Ximena, 2020.

La **Tabla 3-1**, contiene la información aproximada de la estructuración de la cabecera principal del formato gráfico BMP toma en cuenta los datos que más similitud poseen la estructura de varias imágenes en formato BMP.

Después de la cabecera se presentan **los datos de la imagen**, estos datos se graban por filas de forma matricial iniciando por la última fila de la matriz hasta la primera fila, considerando la esquina inferior izquierda de la matriz como el origen de la imagen. Otro punto importante de este formato es que soporta desde imágenes binarias hasta imágenes de 24 bits.

**Ventajas del formato BMP:**

♦ Es un formato gráfico que no contiene datos de imágenes comprimidas, por lo tanto, su tiempo de resolución y despliegue en la pantalla es mínimo.

♦ Es un formato gráfico muy flexible que se pueden adaptar a cualquier tipo de imágenes que se utilice.

**Desventaja**

♦ En imágenes muy grandes este formato requiere de archivos de tamaños excesivamente grandes para su almacenamiento de información.

## 1.8. Análisis matemático en el procesamiento de Imágenes.

En esta sección se realiza una descripción de las operaciones reales que se desarrollara sobre las imágenes, de aquí se derivan los algoritmos que se van a efectuar en el programa procesador de imágenes, después se mu3estra algunas de las operaciones matemáticas y computacionales para el uso de los filtros, demostrando ciertas consecuencias con su aplicación.
El filtrado de señal tiene como objetivo tratar de eliminar o disminuir total o parcial cualquier tipo de ruido presente las señales, observando la eficacia de los filtros aplicados a las imágenes ruidosas, lo que trata de realizar es simular el ruido en imágenes y luego observar el efecto que generan los filtros sobre la eliminación del ruido (Vettorazzi, 2007).

Es necesario tomar en cuenta que los sistemas lineales son de suma importancia para la creación de modelos matemáticos para la Degradación y Restauración de imágenes y modelos que nos permitan entender el comportamiento y efecto en las imágenes digitales, para esto se toma en cuenta que un sistema produce una señal de salida en respuesta a una señal de entrada de forma lineal, es decir, este proceso es lineal debido a que posee dos propiedades de homogeneidad y aditiva.

Además, existen operaciones lineales que se basan en dos métodos que se emplear para resaltar ciertas características importantes en la imagen y para tratar de eliminar o discriminar otras características, estos métodos son la convolución y el análisis de Fourier los cuales nos

proporciona las herramientas principales para el procesamiento y restauración de imágenes además nos permite visualizar las características que posee una imagen digital (Vettorazzi, 2007).

Las operaciones básicas que se realizan sobre las imágenes se clasifican en operaciones lógicas y aritméticas. Operaciones lineales: Son útiles para la detección de características y cubrir ciertas áreas de una imagen mediante la definición de máscaras y un proceso de convolución tomando el mínimo valor de los pixeles entre los cuales se realizará la operación.

### *1.8.1. Convolución*

La convolución se denomina a una función lineal y continua, es un proceso mediante el cual se puede aplicar un filtro a una imagen, utilizando el promedio ponderado (proceso lineal) del pixel de entrada y el de los pixeles vecinos para calcular el valor digital del brillo de cada pixel en la imagen de salida, el grupo de pixeles de la vecindad se conoce como núcleo o kernel, el cual es una matriz móvil con un número impar de valores en cada dimensión $(1x1, 3x3, 5x5, ..)$ para el procesamiento digital pixel a pixel, es decir la convolución consiste en la transformación de la señal de entrada en una señal nueva en el caso de las imágenes digitalizadas la convolución de la función $f(x, y)$ por $g(x, y)$ es una matriz $MxN$.

$$f(x,y)^* g(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) g(x-m, y-n)$$

La convolución espacial consiste en combinar los valores de los elementos del núcleo de convolución y los valores de los coeficientes de convolución en forma de una matriz denominada matriz o máscara de convolución, los valores del pixel central del núcleo y sus pixeles vecinos se multiplican por los respectivos coeficientes de convolución para luego ser sumados y su resultado ubicado en la posición del pixel central procesado de la imagen de salida, de esta manera el proceso recorre la imagen pixel a pixel hasta encontrar los nuevos valores de los pixeles de la nueva imagen sin afectar los bordes de la imagen de entrada debido a que no tienen el número de pixeles vecinos necesarios para realizar el cálculo. La convolución para el caso de una función continua y bidimensional viene dada por la función:

$$f(x,y)^* g(x,y) = \iint_{-\infty}^{\infty} f(x' - y') g(x - x', y - y') dx' dy'$$

**Dónde:** $(x, y)$ son coordenadas del pixel y las variables $x', y'$ representan el desplazamiento de la matriz de convolución. El proceso de convolución de una imagen mediante la aplicación de un filtro da como resultado la imagen filtrada.

Operaciones Aritméticas: Son utilizadas para eliminar ruido, en este proceso implica el uso de dos imágenes y se efectúa pixel a pixel de la primera imagen con los de la segunda mediante operaciones de suma y resta pero tan solo de $nxm$ donde $n$ es el ancho y $m$ es el alto de la imagen en pixeles (López, 2014).

$$S(x,y) = I_1(x,y) \pm I_2(x,y) \quad \textbf{(75)}$$

La imagen de salida depende de la efectividad de la aplicación de las operaciones aritméticas, esto es útil para detectar los cambios producidos después de un intervalo de tiempo o eliminar defectos de la imagen.

### *1.8.2. Transformada de Fourier*

Una transformada se define como el mapeo de un sistema de coordenadas a partir de otro sistema, ejemplo la rotación de una imagen se considera como una transformada debido a que las coordenadas de la imagen rotada corresponden a la imagen original, además la transformada codifica de forma minuciosa la información de la imagen original y puede ser reconstruida (Vettorazzi, 2007).

La transformada de Fourier es una herramienta matemática comúnmente utilizado para trabajar dentro y fuera del dominio de frecuencias, esto implica que una imagen contiene información de alta y bajas frecuencias, además se considera que si tenemos la información contenida en la imagen en términos de frecuencia es posible diseñar cualquier tipo de filtro para la restauración de las imágenes.

## 1.9. Distorsión en las Imágenes Digítales

Para la degradación de una imagen es necesario tomar en cuenta que al tener una imagen digital $f(x,y)$ que se ve afectada por una función de degradación y por cualquier tipo de ruido posiblemente puede ser restaurada parcialmente mientras pase por una etapa de filtrado. El proceso de degradación de una imagen, (figura **22-1**) consiste en que por medio de una función de degradación conjunto a los efectos del ruido operen sobre una imagen $f(x,y)$ para generar una imagen degradada $g(x,y)$.

**Figura 22-1.** Modelo de degradación de una imagen.
**Realizado por:** Villagomez H, Ximena A. 2020

$f(x, y)$=imagen de prueba

$g(x, y)$=imagen distorsionada

⊗ = ruido aditivo

## EFECTO BORROSO EN UNA IMAGEN DE TOMOGRAFÍA

La distorsión de una imagen es el efecto borroso de la misma debido a inconsistencias en las proyecciones que se generan durante un intervalo de tiempo provocando artefactos en la imagen reconstruida visualizando la presencia de bandas que da lugar a una imagen difuminada por el movimiento involuntario del cuerpo del paciente, como se muestra en la figura **23-1.**

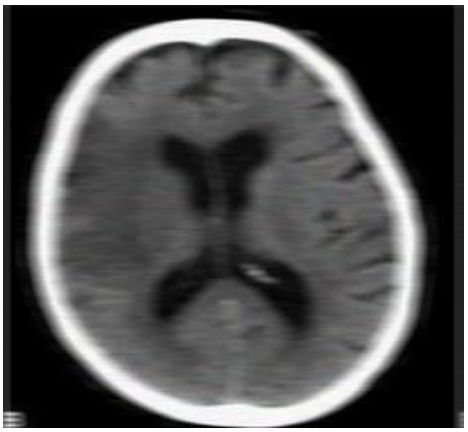


**Figura 23-1.** Efecto de la Distorsión.
**Realizado por:** Villagomez H, Ximena A. 2020

# CAPÍTULO II

## 2. MARCO METODOLÓGICO

En el presente trabajo se utiliza Matlab por la flexibilidad que nos proporciona en cuanto la realización de diversas operaciones de programación nos permite el desarrollo de funciones matemáticas mediante algoritmos para facilitar los procedimientos matemáticos necesario en el menor tiempo posible para la implementación de los filtros propuestos, también nos permite desarrollar una Interfaz Gráfica denominada GUI a través del cual se desarrolla un programa que establezca una buena relación entre el usuario y la PC, y se realiza la simulación de ruido aditivo y se genera funciones de degradación que causa un efecto borroso en las imágenes de prueba causado por el movimiento voluntario o involuntario, para capturar una imagen es necesario aplicar la función "*fspecial*" no que nos permite modificar de forma lineal el desplazamiento de los pixeles así como el ángulo de desplazamiento en la imagen de prueba .

### 2.1. Restauración de las Imágenes obtenidas en Tomografía Computarizada

Las imágenes generadas por la exploración de Tomografía Computarizada son de suma importancia del campo del diagnóstico médico, la calidad de la imagen se ven afectadas por una serie de imperfecciones que se generan a partir de su procesamiento y limitan la valoración patológica de un paciente después de una exploración

En la restauración de las imágenes digitales es necesario tomar en consideración el análisis del fenómeno físico que causa los diversos tipos de ruido (estudiados en el capítulo 2) los mismos que pueden afectar en la resolución de las imágenes Tomográficas, sin embargo, estas imágenes distorsionadas pueden ser restauradas implementando una etapa de filtrado que nos permiten remover ciertas imperfecciones presentes en las imágenes.

En el presente trabajo de titulación se presenta el estudio de 4 tipos de filtros más utilizado y estudiados en la restauración de imágenes digitales, como: el Método del filtrado de Wiener. Método de filtrado Lagrangiano, Método del filtrado Laplaciano y el Método del filtrado de Inverso, con el objetivo de obtener una nueva imagen $f'(x, y)$ a través del filtrado de la imagen de prueba minimizando los niveles de la borrosidad y eliminar la presencia de ruido, mediante la comprensión de los fenómenos que ocasionan la perdida de información en la imagen.

Para lograr nuestro objetivo es necesaria la comprensión de las herramientas matemáticas más utilizadas en el procesamiento de imágenes como análisis matricial y el análisis de la transformada de Fourier. Durante la implementación de los métodos de restauración de una imagen también es

necesario el uso de programas computacionales como en nuestro caso Matlab, que nos facilita el cálculo matemático como el análisis matricial a nivel avanzado y nos proporciona con las herramientas matemáticas indispensables para el análisis de imágenes digitales.

Sin embargo no es posible restaurar una imagen distorsionada al 100% ya que cierta parte de información original no se encuentra en la imagen degradada aun así los métodos de filtrados aplicados en la actualidad nos permiten llegar a lograr una restauración que se aproxima a la imagen original, por lo tanto para obtener un imagen restaurada en su totalidad igual a la imagen original es necesario de la utilización de estos programas o métodos con el apoyo de computadoras con mejor capacidad de procesamiento y análisis de imagen como por ejemplo los telescopios utilizados para explorar varias zonas del espacio que se localizan a grandes distancia.

## 2.2. Generación de Ruido

Al trabajar con imágenes médicas estamos expuestos a diversos tipos de perturbaciones o ruido que se presentan en la formación de la imagen y que generan una degradación de la calidad de la misma. En el presente trabajo se simularán 4 tipos de ruidos, como el Ruido Gaussiano, el Ruido Sal y Pimienta, Poisson y *Speckle* o punto, cada uno descritos con su respectivo algoritmo para observar el efecto que producen en la resolución de la imagen de prueba.

### *2.2.1. Algoritmo para implementar Ruido Gaussiano*

**Algoritmo:**

*Entrada:* Imagen original J, con una función $F(x, y)$

*Salida:* Imagen con un nivel de ruido J1, $E(x, y)$

Definir:

- Valor de la *varianza* del ruido V
- Lectura del valor de intensidad de un pixel de la imagen original.
- Se aplica la fórmula 22 para generar el ruido Gaussiano Blanco

Este diagrama de flujo se realizó tomando en cuenta las siguientes variables: V= varianza D= densidad estándar del ruido.

- El valor del pixel de la imagen original se suma con el valor del ruido gaussiano.

***Código fuente del programa para simulación***

Los comandos que se utilizó en el presente trabajo son:

```
set(handles.text23,'String','Varianza')
 set(handles.text24,'String','Media')
```

*set(handles.edit1,'String',0)* *%* Media

*set(handles.edit2,'String',0.01) %* Varianza

Mediante la función `set` para especificar los nuevos valores de las propiedades de la imagen original. Luego utilizamos el comando *imnoise* para añadir ruido a la imagen de prueba con los nuevos valores especificados:

*J = imnoise (imagen,'gaussian', edit1, edit2);*

### *2.2.2. Algoritmo para implementar Ruido Sal y pimienta*

**Algoritmo:**

*Entrada:* Imagen original J, con una función $F(x, y)$

*Salida:* Imagen con un nivel de ruido J1, $E(x, y)$

Definir:

D= Densidad, valor aleatorio entre 0 y 1 para nuestro caso es igual a 0.05.

- Asigna el valor de probabilidad aleatoria $p$ a cada pixel en un intervalo abierto de (0-1)
- Se lee un pixel de la imagen para agregar ruido de este tipo con una densidad
- Estos pasos se repiten hasta que haya terminado de leer todos los pixeles o byte de la imagen original o de prueba a excepción del primer pixel.

***Código fuente del programa para simulación***

Los comandos que se utilizó en el presente trabajo son:

*set(handles.text23,'String','Densidad')*

*set(handles.text24,'String','')*

*set(handles.edit1,'String',0.05) %* Densidad

*set (handles. edit2,'String',0)*

Utilizamos el comando *imnoise* para añadir ruido a la imagen de prueba con los nuevos valores especificados mediante la función *set*:

*J = imnoise(imagen,'salt & pepper',edit1);*

### *2.2.3. Algoritmo para implementar Ruido Poisson*

**Algoritmo:**

*Entrada:* Imagen original J, con una función $F(x, y)$

*Salida:* Imagen con un nivel de ruido J1, $E(x, y)$

Definir:

$l =$ Valor de la longitud igual a 21

$theta = 11$

La estimación del ruido Poisson depende directamente de la magnitud de los pixeles de la imagen de prueba.

***Código fuente del programa para simulación***

Los comandos que se utilizó en el presente trabajo son:

*set(handles.text24,'String','Longitud')*

*set(handles.text23,'String','Theta')*

*set(handles.edit1,'String',21) %* Longitud

*set(handles.edit2,'String',11) %* Theta

Mediante la función `set` para especificar los nuevos valores de las propiedades de la imagen original. Luego utilizamos el comando *imnoise* para añadir ruido a la imagen de prueba con los nuevos valores especificados:

*J = imnoise (imagen,'poisson');*

### *2.2.4. Algoritmo para implementar Ruido Speckle o punto*

**Algoritmo:**

*Entrada:* Imagen original J, con una función $F(x, y)$

*Salida:* Imagen con un nivel de ruido J1, $E(x, y)$

Definir:

$\mu =$ Valor medio del ruido para nuestro caso es igual a cero

V= Varianza del ruido igual a 0.04, ecuación **(24)**

***Código fuente del programa para simulación***

Los comandos que se utilizó en el presente trabajo son:

*set(handles.text23,'String','Varianza')*

*set(handles.text24,'String','Media')*

*set(handles.edit1,'String',0)* % Media

*set(handles.edit2,'String',0.04)* % Varianza

Mediante la función `set` para especificar los nuevos valores de las propiedades de la imagen original. Luego utilizamos el comando `imnoise` para añadir ruido a la imagen de prueba con los nuevos valores especificados:

*J = imnoise(imagen,'speckle',edit2);*

**MODELO DE RESTAURACIÓN DE UNA IMAGEN**

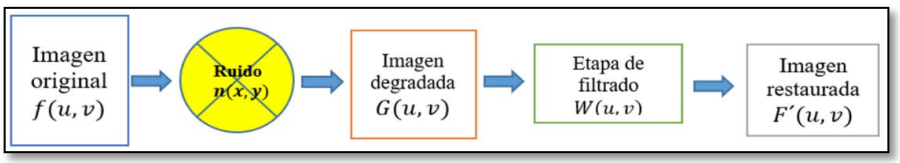



**Figura 1-2.** Diagrama de Bloques para la Restauración de una Imagen Degradada

**Realizado por:** Villagomez H, Ximena A. 2020

## 2.3. Métodos de filtrado - Implementación del programa

El desarrollo de la aplicación nos permitirá realizar el proceso de restauración de imágenes distorsionadas para lo cual, es necesario obtener un algoritmo de programación a través del cual será posible observar la forma en la que el filtro Wiener e Inverso se encargan de recupera cierta información contenida en la imagen de prueba con la ayuda del lenguaje de programación Matlab.

## 2.4. Desarrollo de algoritmos para la aplicación del filtro

En el proceso de captura de imagen su algoritmo es igual tanto para todos los métodos de filtrado propuestos en este Trabajo de Titulación, esta etapa de inicio es muy importante ya que proporciona al software Matlab las herramientas necesarias para el análisis de una imagen, como se muestra en la **figura 2-2**:

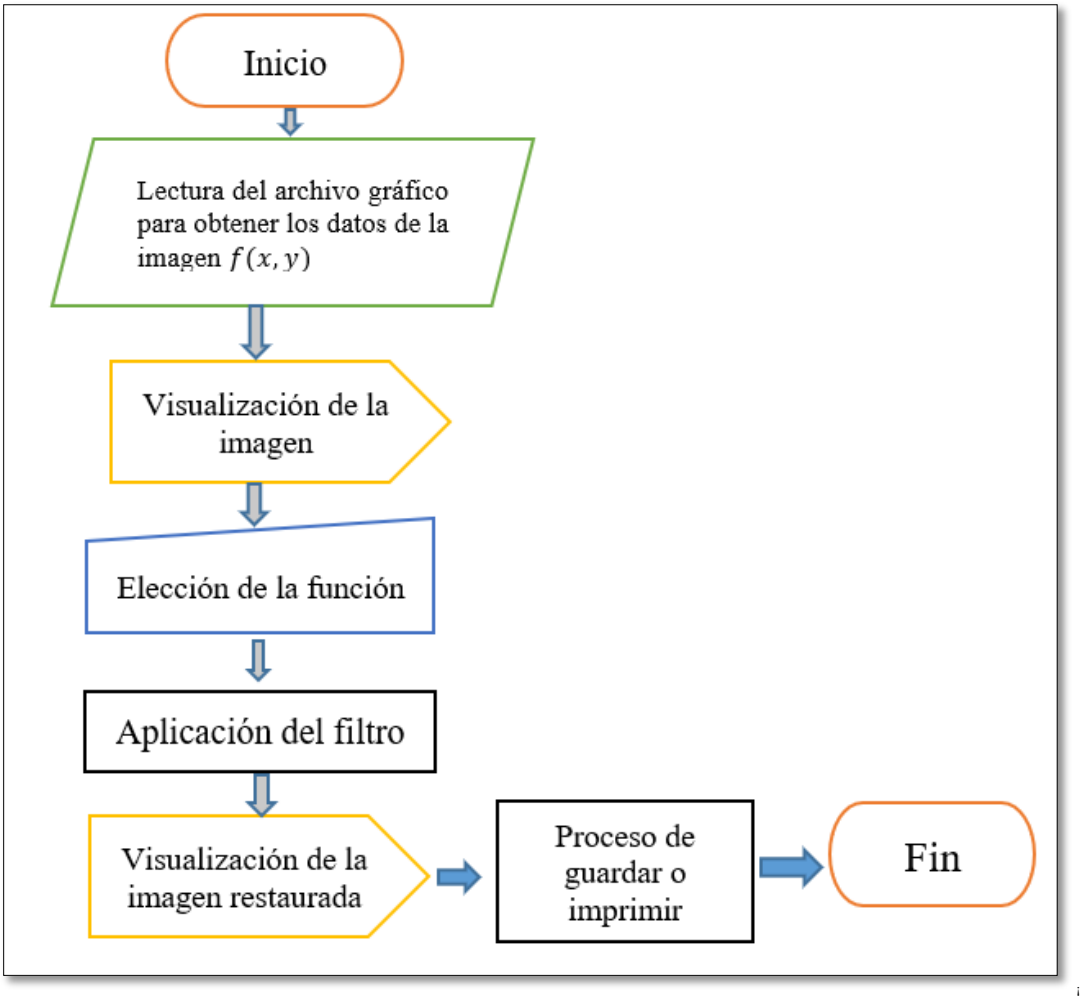



**Figura 2-2.** Diagrama General del Proceso de Captura de la Imagen Digital.

**Realizado por:** Villagomez H, Ximena A. 2020

### *2.4.1. Filtrado Wiener*

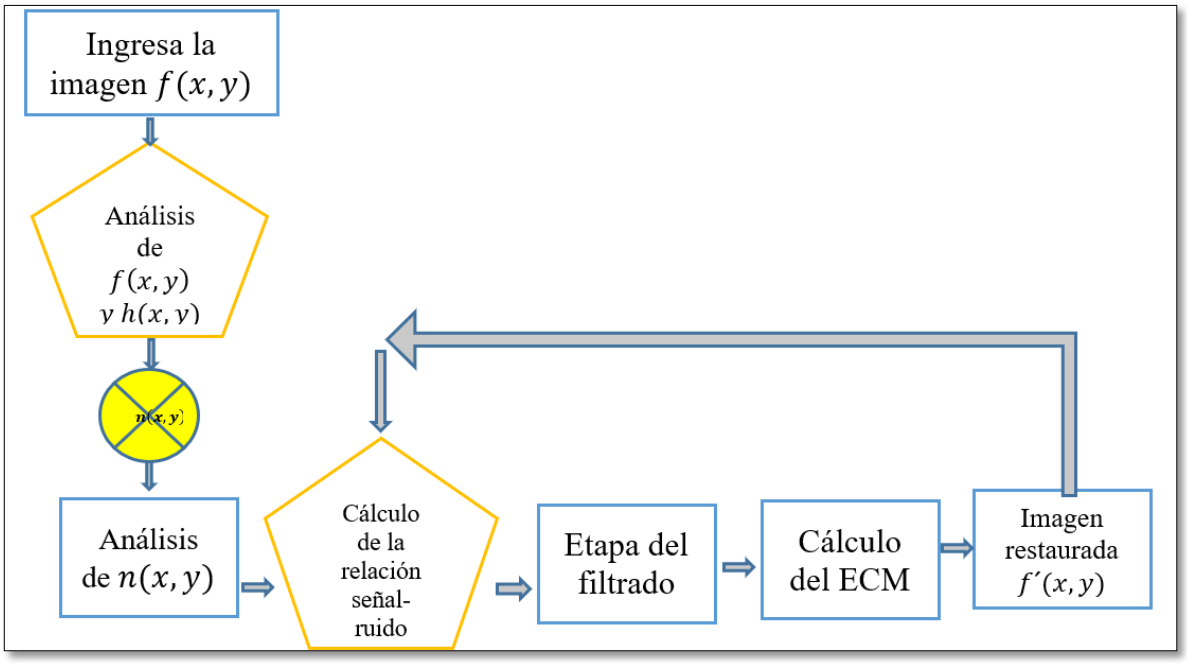



**Figura 3-2.** Diagrama de bloques empleado en el Filtro Wiener

**Realizado por:** Villagomez H, Ximena A. 2020

**Algoritmo desarrollado para la aplicación del Filtro Wiener:**

*Entrada:* Imagen original con una función $f(x,y)$

*Salida:* Imagen estimada, $f'(x,y)$

- Lectura de la imagen para convertirle en un arreglo matricial MxN

*[nombre ruta]=uigetfile* % Acepta todo Formato de imágenes compatible con Matlab

*imread_(fullfile (ruta, nombre); %Lectura de la imagen*

- Análisis de la imagen

*k=fft2(j); %Transformada de Fourier de imagen original*

*size(j) % tamaño de la imagen*

*otf = psf2otf %Transformada de función de degradación*

*d=otf. *k; %Aplicación de degradación a imagen dominio de la frecuencia*

- Cálculo del error ECM

*immse %Calcula el MSE o ECM*

- Etapa de Filtrado

Durante la etapa de filtrado se desarrolla el algoritmo que nos permite realizar la restauración de la imagen $f'(x,y)$.

### 2.4.2. *Filtrado Laplaciano*

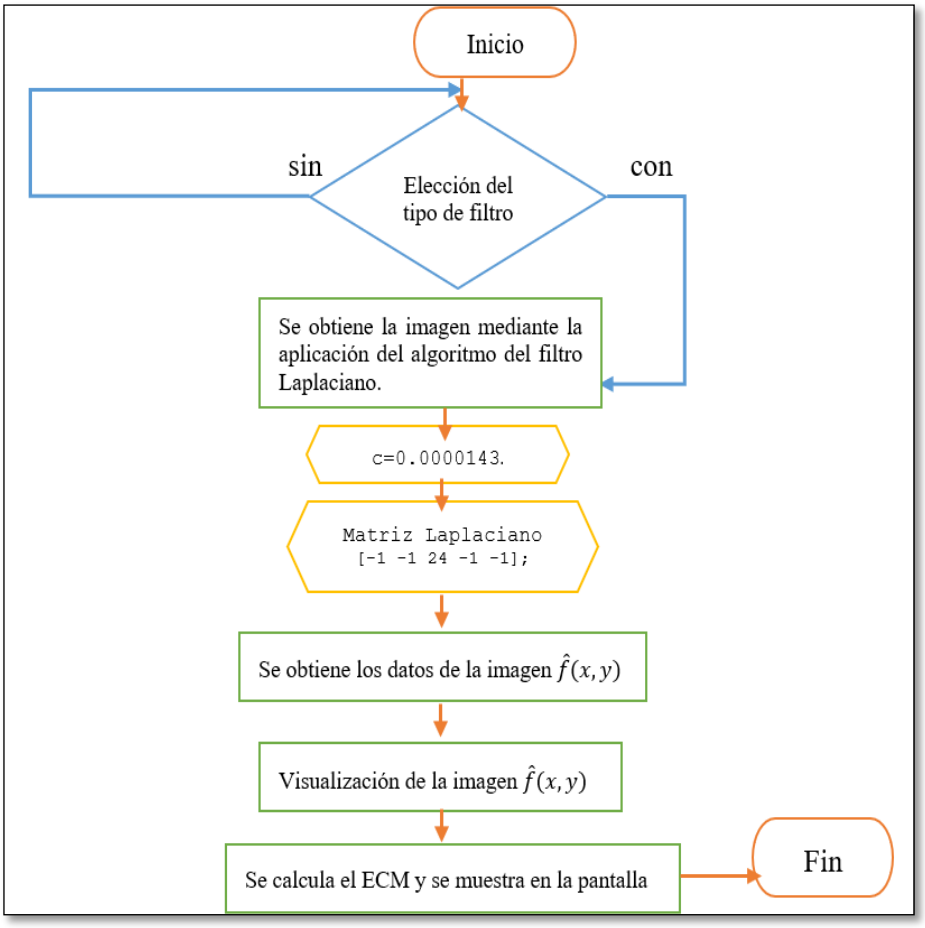



**Figura 4-2.** Diagrama de bloques para la restauración de Imágenes con el Filtro Laplaciano

**Realizado por:** Villagomez H, Ximena A. 2020

**Algoritmo desarrollado para la aplicación del Filtro Laplaciano:**

*Entrada:* Imagen original con una función $g(x,y) = h(x,y) * f(x,y) + n(x,y)$
*Salida:* Imagen estimada, $g'(x,y) = f(x,y) - c\nabla^2 f(x,y)$

- Lectura del archivo gráfico en formato bmp.

*imread (fullfile (ruta, nombre);* %Lectura de la imagen

- Se aplica la función *Edged = edgetaper(imagennoise,PSF1)*; para reducir la amplificación del ruido a lo largo del contorno de la imagen antes de la deconvolución.
- Se declara la constante C=0.0000143.

C=0.0000143;

- Se declara la máscara Laplaciana, la misma que se multiplica por el valor de c.

*REGOP* = [-1 -1 24 -1 -1];

- Mediante la función *imagenfilt* se filtra la imagen de prueba obteniendo la imagen reconstruida.

*imagenfilt = deconvreg(imagennoise,PSF,[],C,REGOP)*

### *2.4.3. Filtro Lagrangiano*

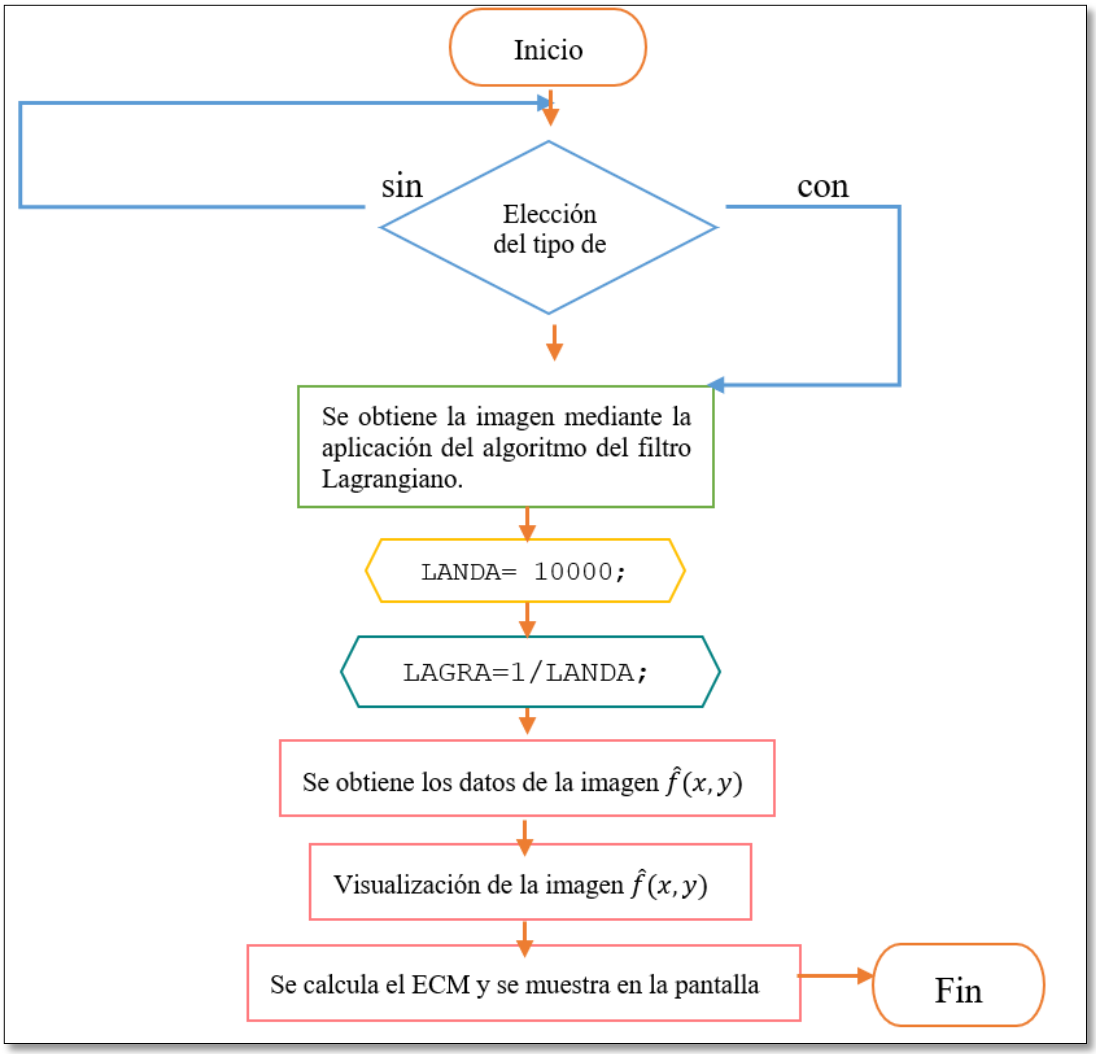



**Figura 5-2.** Diagrama de bloques para la restauración de Imágenes con el Filtro Lagrangiano.

**Realizado por:** Villagomez H, Ximena A. 2020

**Algoritmo desarrollado para la aplicación del Filtro Lagrangiano :**

*Entrada:* Imagen original con una función $g(x,y) = h(x,y) * f(x,y) + n(x,y)$

*Salida:* Imagen estimada, $\hat{f} = (H^T g)(\gamma Q^T Q - H^T H)^{-1}$

- Lectura de la imagen para convertirle en un arreglo matricial MxN

*imread (fullfile (ruta, nombre); %Lectura de la imagen*

- Se declara el valor del operador de Lagrange λ

LANDA= 10000;

- Definimos la variable γ(LAGRA) cuyo valor depende de la división de la unidad sobre el valor del operador de Lagrange.

LAGRA=1/LANDA;

- Mediante la función imagenfilt se obtiene la imagen filtrada.

*imagenfilt = deconvreg(Edged,PSF2,[],LAGRA/100);*

### *2.4.4. Filtrado Inverso*

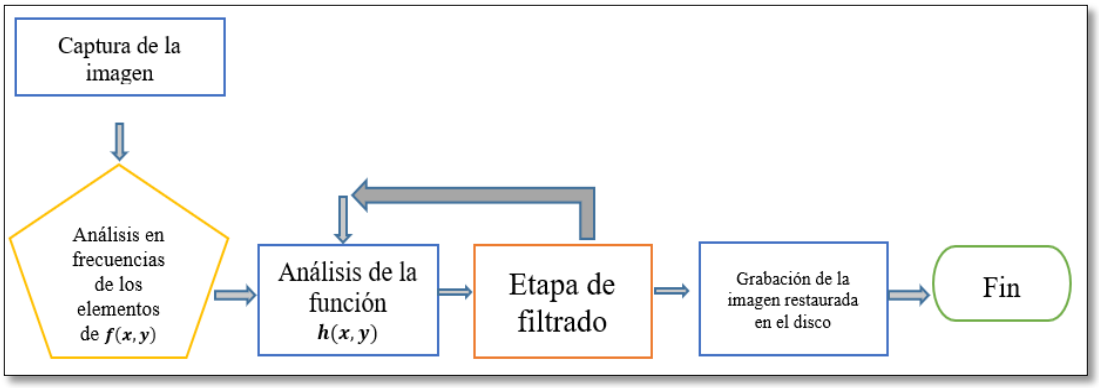



**Figura 6-2.** Diagrama de bloques para la restauración de imágenes mediante el Filtro Inverso

**Realizado por:** Villagomez H, Ximena A. 2020

**Algoritmo desarrollado para la aplicación del Filtro Inverso**

---

*Entrada:* Imagen original con una función $f(x,y)$

*Salida:* Imagen estimada, $f'(x,y)$

- Lectura de la Imagen
- Imagen para convertirle en un arreglo matricial MxN

*[nombre ruta]=uigetfile % Acepta todo Formato de imágenes compatible con Matlab*

*imread (fullfile (ruta, nombre); %Lectura de la imagen*

- Análisis en frecuencias de los elementos que constituye una imagen

*k=fft2(j); %Transformada de Fourier de imagen original*

*psf=fspecial %Función de desenfoque-degradación*

*size(j) % tamaño de la imagen*

*otf = psf2otf %Transformada de función de degradación*

*d=otf.*k; %Aplicación de degradación a imagen dominio de la frecuencia*

*imagennoise=abs(ifft2(d)) % calcula el espectro de Fourier*

- Análisis de la función de Degradación

Para el desarrollo de esta etapa se debe conocer el fenómeno que causa el efecto borroso en la imagen de prueba ya sea por observación o experimentación, en nuestro caso las imágenes no contienen ruido por lo tanto $n(x,y)$ tiende a cero permitiendo tener un mejor efecto el filtro sobre la imagen de prueba.

- Etapa de Filtrado

Usando la deducción matemática de cada filtro realizamos la respectiva codificación en Matlab, para la valoración y optimación de las imágenes resultantes.

## 2.5. Software

Para la realización de la interfaz gráfica se utilizó el software MATLAB versión R2017b, debido a que es un software que posee una serie de funciones, capacidades para la interpretación, manejo de datos y visualización de datos, desarrollo de algoritmos, análisis y cálculos numéricos, es más accesible en su manejo.

Para su funcionalidad de la interfaz fue instalado y verificado en un procesador INTEL CORE i5 con 4GB de memoria RAM.

### *2.5.1. Descripción del equipo utilizado*

En este capítulo se desarrolla una interfaz gráfica de trabajo que permite al usuario procesar imágenes en el formato bmp mediante una computadora portátil, con ciertas características fundamentales para facilitar el procesamiento de imágenes, se muestra en la **Tabla 1-2**.

**Tabla 1-2: Características Técnicas del Equipo Portátil.**

| **Laptop HP CORE i5** | |
|---|---|
| Sistema Operativo | El sistema operativo es responsable por la administración de los recursos de su computador y permite que el software y el hardware trabajen juntos. • Windows 7 Home Premium (64 Bits) |
| Procesador | Un procesador más rápido le permite terminar sus tareas en menos tiempo. • Intel Core i5-430M (2.26 GHz, 3 MB de caché), con Turbo Boost hasta 2.53 GHz. |
| Pantalla | Una pantalla clara y brillante donde quiera que la lleve. El tamaño de la pantalla esta dado en pulgadas. • 14.5" LED Widescreen con tecnología BrightView Infinity (1366 x 768) |
| Memoria | Más memoria le permite ejecutar más programas de software al mismo tiempo. • 4 GB, DDR3 (2 x 2048MB) |
| Disco Duro | Discos duros le dan el espacio de almacenamiento para sus datos y programas. • 500 GB, 7200 rpm, SATA |
| Gráficos | El controlador de gráficas acelera el procesamiento de las imágenes, ofreciéndole una alta calidad y mejor nitidez. • Intel HD Graphics |
| Lector de Memorias | Para utilizar la tarjeta de memoria de otro dispositivo (como una cámara digital o un MP3) para acceder a fotos, películas, archivos, música, etc. |

**Realizado por:** Villagomez Herrera, Ximena, 2020.

## 2.6. Descripción de la Interfaz Gráfica de usuario

En la aplicación de los algoritmos para la restauración de una imagen degradada se diseñó una interfaz gráfica o GUIDE (*Graphical User Interface Development Environment*) en Matlab, que nos permite generar un entorno de programación visible para ejecutar programas mediante varias ventanas interactivas con el usuario de forma rápida y sencilla en su manejo, una aplicación GUIDE consta de 2 tipos de archivos: un *archivo.m* donde se muestra el código ejecutable con los correspondientes botones de control de la interfaz y el *archivo.fig.*

Para poner en funcionamiento la aplicación desarrollado se realiza de dos maneras la primera abriendo el cuadro de dialogo de la GUIDE (figura 5-2) y la segunda forma desde la ventada de comandos de Matlab el cual posee los elementos gráficos necesarios para el diseño y ejecución de nuestra interfaz gráfica etiquetada con el nombre "**PRESENTACIÓN**".

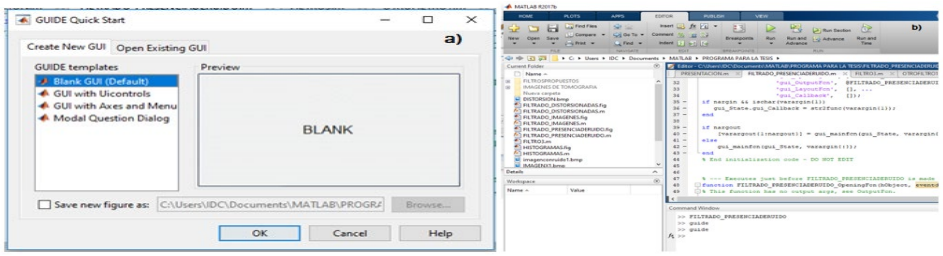



**Figura 7-2.** a) Inicio de la GUIDE y b) Editor de Matlab.

**Realizado por:** Villagomez H, Ximena A. 2020

## ***PRESENTACIÓN***

Una vez ejecutando la aplicación de cualquier forma que se muestra en la figura 5-2, aparece el entorno o cuadro de presentación de la aplicación desarrollada, en el cual consta de tres botones que nos dirige a distintos entornos con sus respectivos nombres en donde se lleva a cabo el procesamiento de imágenes mediante herramientas específicas.

El usuario puede seleccionar el tipo de entorno en el que quiera trabajar, donde se puede cargar la imagen de prueba con su respectivo formato al presionar el botón ***Examinar***, se selecciona la imagen mediante la ventana del navegador de archivos. Para que se realice la tarea de examinar se emplea la función *imread* de Matlab que nos permite abrir la imagen de prueba.

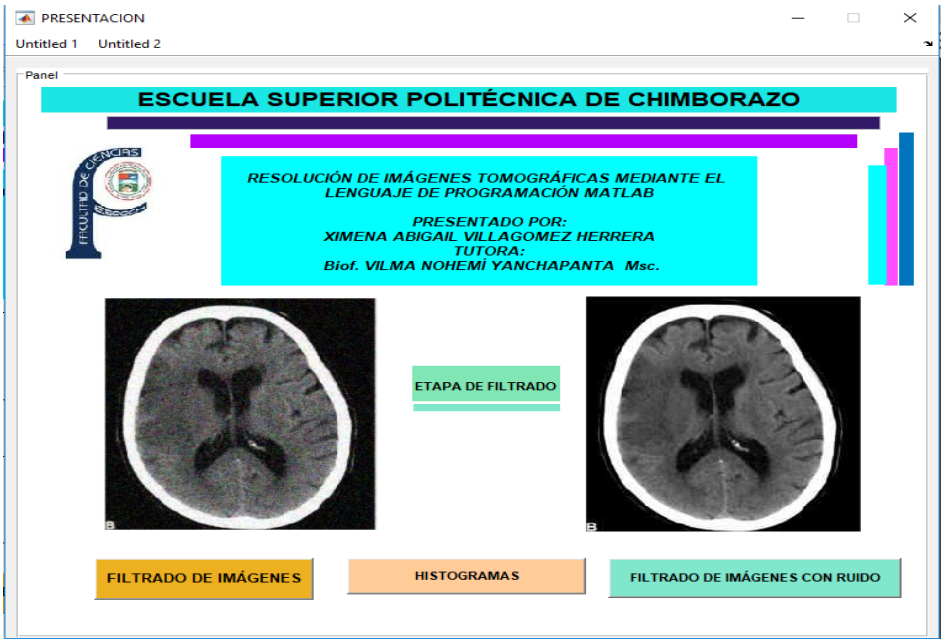



**Figura 8-2.** Cuadro de Diálogo de la Presentación del Programa.

**Realizado por:** Villagomez H, Ximena A. 2020

La imagen seleccionada se puede visualizar mediante el botón ***IMAGEN A FILTRAR*** en la parte izquierda del entorno principal de la interfaz; mientras que en la parte derecha se observa la imagen filtrada con sus respectivos cambios a medida que se utilice las herramientas propuestas en la programación.

- **Cuadros de diálogo**

Son ventanas específicas que indican al usuario suministrar la informacion necesaria para la aplicación del programa. En la ventana principal se presentan una barra de herramientas que facilita al usuario realizar diversas operaciones en el menor tiempo posible, las cuales estan constituidas por botones que dan acceso a las opciones principales en los menus de la aplicación. Los botones que posee una barra de herramieta son:

**Retornar al inicio de la presentación:**

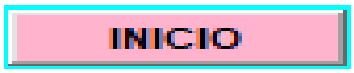



**Figura 9-2.** Botón Inicio

**Realizado por:** Villagomez H, Ximena A. 2020

**Cerrar todas las ventanas del cuadro de diálogo del programa:**


**Figura 10-2.** Botón Salir.

**Realizado por:** Villagomez H, Ximena A. 2020

**Aplicar – añadir la opción escogida del menú**


**Figura 11-2.** Botón Añadir o Aplicar.

**Realizado por:** Villagomez H, Ximena A. 2020

**Examinar o abrir imagen desde el disco**, es un cuadro de diálogo que nos permite cargar un archivo gráfico desde la Pc.

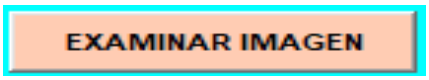



**Figura 12-2.** Examinar la PC.

**Realizado por:** Villagomez H, Ximena A. 2020

**Guardar imagen en el disco**: facilita al usuario que puede guardar una imagen en el disco de su PC en cualquier tipo de formato compatible con el programa.


**Figura 13-2.** Guardar en la PC.

**Realizado por:** Villagomez H, Ximena A. 2020

## SISTEMA DE MENÚS

Presenta una lista de todas las opciones que cuenta el programa y que puede acceder el usuario con facilidad.

Selección de elementos-filtros: nos permite escoger un elemento de las características que necesitemos para el procesamiento de imágenes, este programa fue diseñado para escoger un tipo de filtro para ser aplicado en una imagen.


**Figura 14-2.** Menú de Filtros.

**Realizado por:** Villagomez H, Ximena A. 2020

**Añadir ruido a la imagen:** nos permite seleccionar el tipo de ruido que se requiere simular en la imagen de prueba, la figura 13-2 presenta el diseño del cuadro de dialogo.


**Figura 15-2.** Menú para añadir Ruido.

**Realizado por:** Villagomez H, Ximena A. 2020

## PROGRAMA FILTRADO_IMAGENES

**Cuadros de ejes:** muestra la imagen de prueba, la imagen restaurada después de ser aplicada el filtro y el histograma dependiendo en que cuadro de diálogo se encuentre.

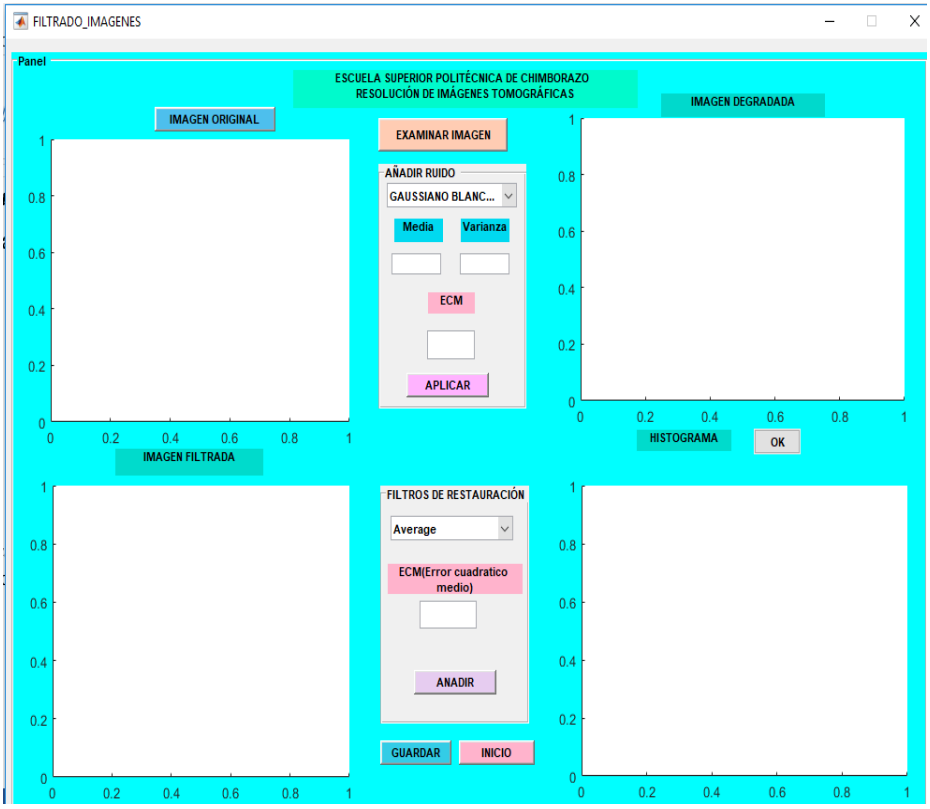



**Figura 16-2.** Entorno Grafico para Generación de Ruido.

**Realizado por:** Villagomez H, Ximena A. 2020

## PROGRAMA FILTRADO_PRESENCIADERUIDO

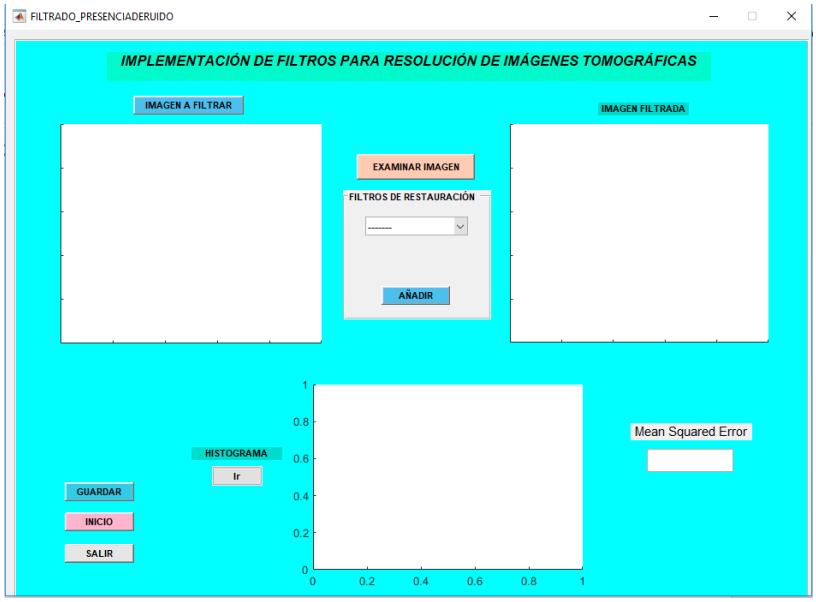



**Figura 17-2.** Entorno de Prueba de Filtros.

**Realizado por:** Villagomez H, Ximena A. 2020

# CAPITULO III

## 3. MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En este capítulo principalmente se da a conocer la información de la imagen de prueba, se analizan y muestran los resultados obtenidos mediante la ejecución de las herramientas de la interfaz gráfica previamente desarrollada de acuerdo al algoritmo descrito en el capítulo 2, posteriormente se hace una comparación con los diferentes filtros propuestos.

### 3.1. Información de las Imágenes de prueba mediante Matlab

Mediante la función *imfinfo* de Matlab se conoce la información detallada de la imagen de prueba (Tabla **1-3**).

CÓDIGO FUENTE: Obtención de la información de la imagen de prueba

```
a=imread('TAC normal.bmp');
whos a
imfinfo('TAC normal.bmp')
```

**INFORMACIÓN DE LA IMAGEN PROPUESTA**

La imagen (**figura 1-3**) corresponde al diagrama axial de una Tomografía Computarizada craneal a nivel de los ganglios basales tomada a un paciente masculino de 30 años que no presenta ningún tipo de trauma ya sea intra o extra axial, se puede observar diferencias entre el tono gris y blanco de acuerdo a la densidad del tejido, presenta una normal morfología en el sistema ventricular (1), diámetro entre las asta frontal (2), normalidad en el diámetro de las astas occipitales (3), diámetro antero-posterior de la encrucijada ventricular (4) y normalidad en el diámetro interhemisférico frontal (5).

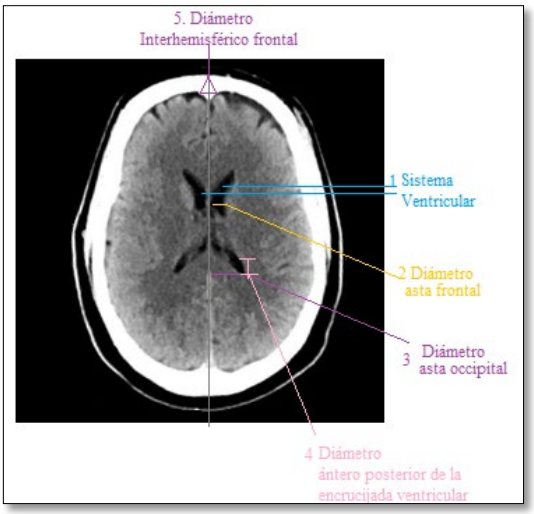


**Figura 1-3.**Tomografía Craneal.

**Realizado por:** Villagomez H, Ximena A. 2020

**Tabla 1-3:** Información de la Imagen en formato bmp.

| INFORMACIÓN GENERAL | |
|---|---|
| *Name Size Bytes Class Attributes*<br><br>*a 444x352x3 468864 uint8*<br>*ans =*<br>*struct with fields:*<br>*Filename: 'C:\Users\IDC\Documents\MATLAB\TAC normal.bmp'*<br>*FileModDate: '25-Apr-2020 14:32:04'*<br>*FileSize: 468918*<br>*Format: 'bmp'*<br>*FormatVersion: 'Version 3 (Microsoft Windows 3.x)'*<br>*Width: 352*<br>*Height: 444*<br>*BitDepth: 24*<br>*ColorType: 'truecolor'* | *FormatSignature: 'BM'*<br>*NumColormapEntries: 0*<br>*Colormap: []*<br>*RedMask: []*<br>*GreenMask: []*<br>*BlueMask: []*<br>*ImageDataOffset: 54*<br>*BitmapHeaderSize: 40*<br>*NumPlanes: 1*<br>*CompressionType: 'none'*<br>*BitmapSize: 468864*<br>*HorzResolution: 0*<br>*VertResolution: 0*<br>*NumColorsUsed: 0*<br>*NumImportantColors: 0* |

**Realizado por:** Villagomez Herrera, Ximena, 2020.

## 3.2. Degradación de la Imagen de prueba mediante la simulación de Ruido

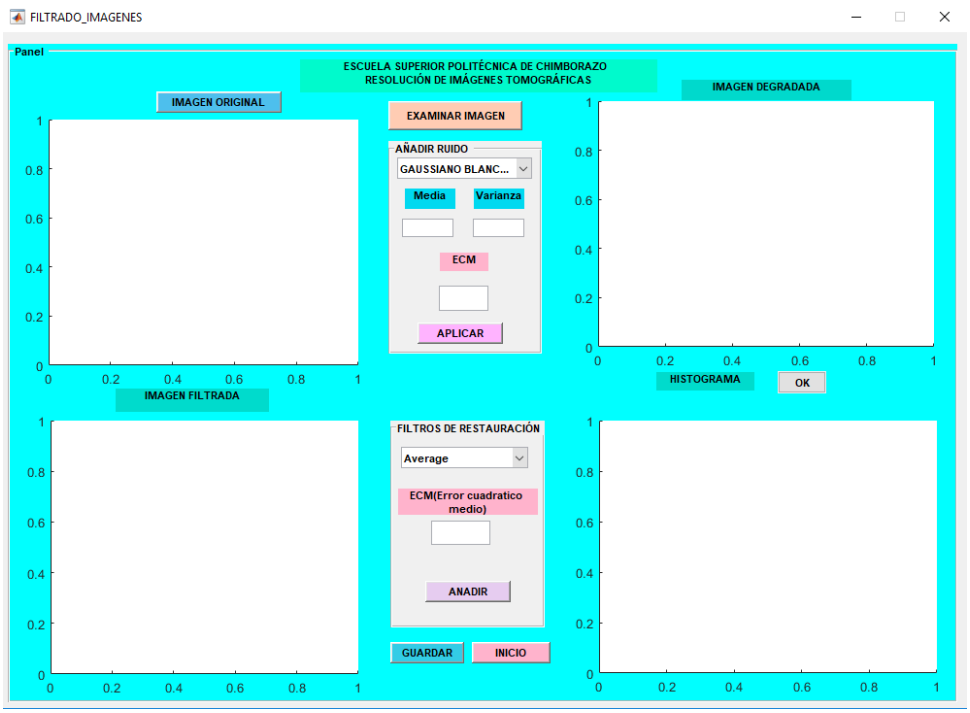



**Figura 2-3.** Interfaz Gráfica para la Simulación de Ruido.

**Realizado por:** Villagomez H, Ximena A. 2020

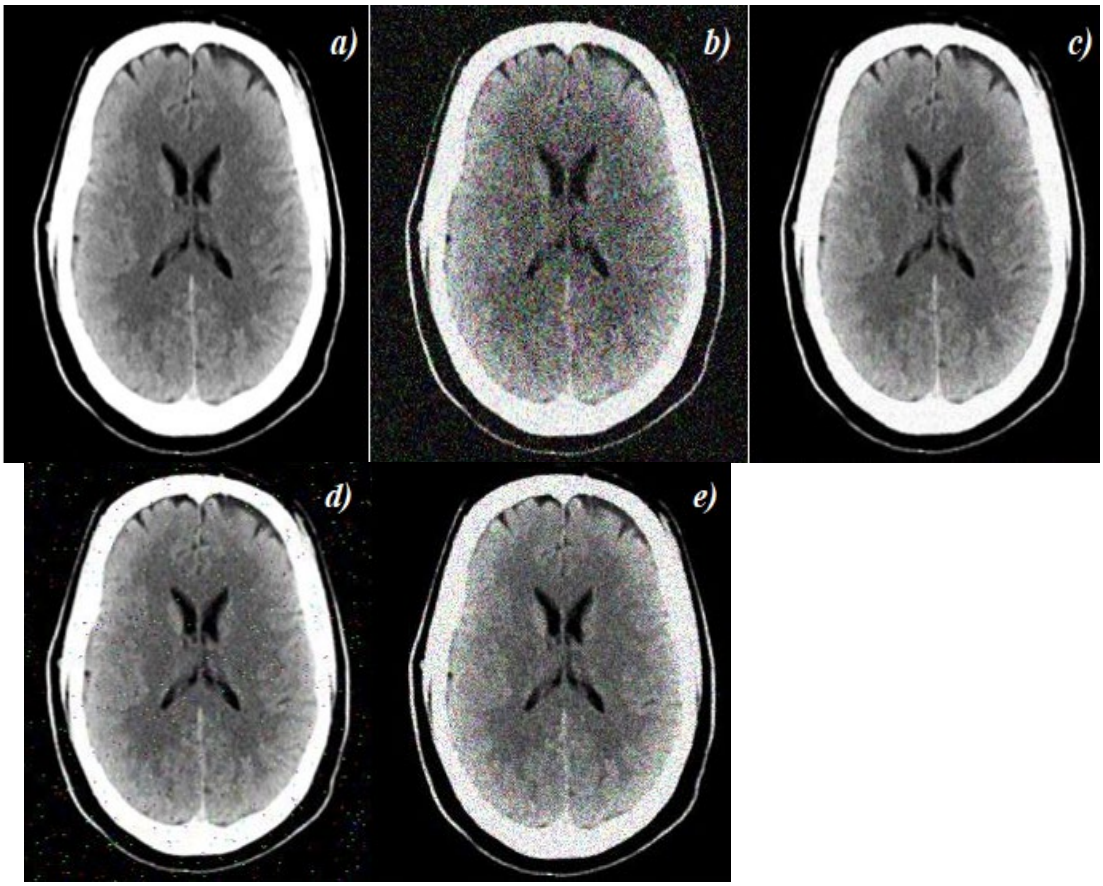



**Figura 3-3.** Simulación de Ruido, a) Imagen Original, b) Ruido Gaussiano, c) Ruido Poisson, d) Ruido Sal y Pimienta y e) Ruido Speckle.

**Realizado por:** Villagomez H, Ximena A. 2020

La **figura 3-3,** muestra el resultado obtenido de simulación del efecto que genera los diferentes tipos ruido al contaminar la imagen de prueba (**figura 1-3**), tomando en cuenta a los algoritmos propuestos en el capítulo II en la sección 2.2. Generación de ruido del presente Trabajo de Titulación.

En la **figura 3-3 b),** se presente el resultado alcanzado al contaminar o degradar la imagen de prueba con el Ruido Gaussiano, con los siguientes valores, media igual a cero y varianza igual a 0.03. En la figura 3-3 b se observa que evidentemente hay pérdida de información que limita y disminuye la resolución de estructuras pequeñas, mala iluminación en los bordes y emborrazamiento de todos los pixeles de la imagen.

La **figura 3-3 c),** demuestra como la imagen de prueba (**figura 3-3 a**) es contamina con el Ruido Poisson tomando como valor de K igual a 21 (número de ocurrencia de un evento) y Theta igual a 11 (número positivo); se observa que la resolución de estructuras pequeñas y bordes se conservan.

La **figura 3-3 d)**, muestra el efecto de la aplicación del Ruido Sal y Pimienta o impulsivo en la degradación de la imagen (**figura 3-3 a**), mediante un valor de densidad de ocurrencia igual 0.05; se observa que ciertos pixeles se ven afectados dejándolos completamente blancos o negros generando perdida de información, la resolución de estructuras pequeñas y bordes es aceptable en la imagen.

El resultado de la simulación del Ruido Speckle con varianza igual a 0.04 y media igual a 0 se observa en la **figura 3-3 e**, además, se muestra la presencia de ruido que se manifiesta de forma granulada (puntos negros y blancos) que se distribuye de manera aleatoria en toda la imagen, lo cual, no permite conservar bordes y estructuras importantes para su estudio.

### *3.2.1. Resultados Analizando el valor de MSE y SNR*

**Tabla 2-3:** Análisis Teórico para evaluar el nivel de Ruido simulado en la Imagen.

| Tipo de Ruido | MSE | SNR |
|---|---|---|
| Gaussiano | 57 | 16.8844 |
| Poisson | 27 | 29.9037 |
| Sal y Pimienta | 3 | 16.8664 |
| Speckle | 51 | 21.8747 |

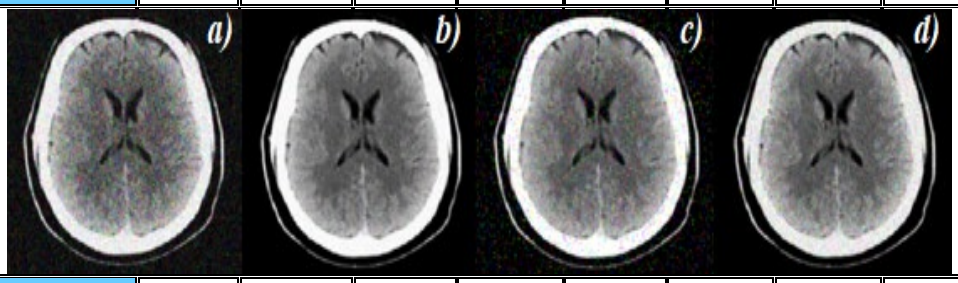
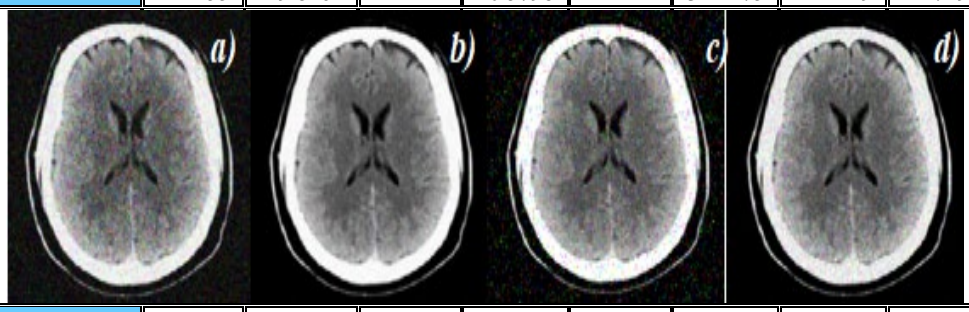
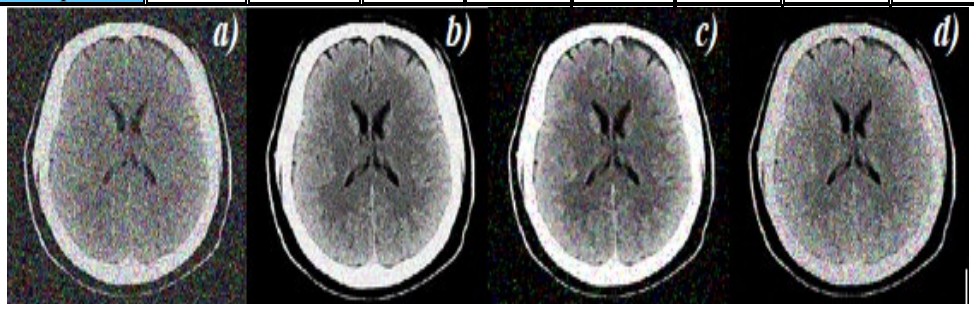
**Realizado por:** Villagomez Herrera, Ximena, 2020.

Después de aplicar los diferentes tipos de ruidos se analizaron los resultados teóricos proporcionados en el cálculo de Error Cuadrático Medio (MSE) de las imágenes originales con las imágenes distorsionadas y SNR mediante el programa con el código de Matlab, en este caso el ruido que género menor distorsión en la imagen de prueba de acuerdo al dato del MSE es el ruido Sal y Pimienta, usando una densidad de 0.05 obtenemos un MSE de 3 y además obtenemos un valor menor del SNR igual a 16.8664, al evaluar el máximo valor de tono de gris en la imagen de prueba dividido por su error cuadrático medio.

### *3.2.2. Comparación del MSR Y SNR de la imagen distorsionada por los tipos de ruidos y la imagen restaurada.*

En este trabajo de titulación se realizó la aplicación de 3 tipos de filtros que se utiliza principalmente para suavizar o realzar-detectar los bordes de la imagen de prueba.

**Tabla 3-3:** Valor del MSR y SNR de acuerdo al filtro utilizado para la eliminación de ruido presente en la imagen.

| VALOR DEL MSR Y SNR | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Ruido / filtro | Gaussiano | | Poisson | | Sal y Pimienta | | Speckle | |
| | MSR | SNR | MSR | SNR | MSR | SNR | MSR | SNR |
| Disk | 53 | 21.716 | 21 | 23.3324 | 19 | 31.4521 | 46 | 26.072 |
|   | | | | | | | | |
| Gaussiano | 55 | 19.8464 | 14 | 20.5798 | 21 | 32.1275 | 49 | 24.7195 |
|   | | | | | | | | |
| Unsharp | 63 | 8.4416 | 38 | 15.2322 | 56 | 17.1519 | 64 | 11.5649 |
|   | | | | | | | | |

**Realizado por:** Villagomez H, Ximena A. 2020

La **Tabla 3-3** representa los valores del MSR, SNR y la imagen restaurada obtenida en el momento de filtrar la imagen distorsionada por el efecto del ruido presente (a-gaussiano, b-Poisson, c-Sal y pimienta, d-Speckle), consiguiendo un nivel aceptable en la restauración de la misma.

Después de analizar de forma visual todas las imágenes resultantes (**Tabla 3-3**) del proceso del filtrado, se determina que los filtros Disk, Gaussiano y Ursharp muestran una mejor respuesta en la recuperación de la imagen distorsionada con presencia del ruido de tipo Poisson (b) y Speckle (c), sin embargo, el valor del MRS de la imagen distorsionada con el ruido Poisson y Speckle es

menor al valor MRS de la imagen filtrada después de aplicar el filtro Disk y Gaussiano, por lo tanto de acuerdo con los datos logrados podemos considerar que los filtros Disk y Gaussiano revelan mayor equilibrio en el nivel de eliminación de ruido y la perdida de detalles de la imagen filtrada.

## INFORMACIÓN DE LA IMAGEN DE PRUEBA

La imagen corresponde a una Tomografía de cerebro, corte axial realizada a un paciente de 42 años de edad, no se puede visualizar de forma correcta por la presencia de artefactos generados por el movimiento que realizo el paciente (**figura 6-3a**).

**Tabla 4-3:** Descripción Detallada de la Información de la Imagen de Prueba.

| **Información General** | |
|---|---|
| *>> informacion1* | *FormatSignature: 'BM'* |
| *Name Size Bytes Class Attributes* | *NumColormapEntries: 0* |
| *a 272x236x3 192576 uint8* | *Colormap: []* |
| *ans =* | *RedMask: []* |
| *struct with fields:* | *GreenMask: []* |
| *Filename: 'C:\Users\IDC\Documents\MATLAB\PROGRAMA PARA LA TESIS\DISTORSION.bmp'* | *BlueMask: []* |
| *FileModDate: '10-Feb-2020 21:56:55'* | *ImageDataOffset: 54* |
| *FileSize: 192630* | *BitmapHeaderSize: 40* |
| *Format: 'bmp'* | *NumPlanes: 1* |
| *FormatVersion: 'Version 3 (Microsoft Windows 3.x)'* | *CompressionType: 'none'* |
| *Width: 236* | *BitmapSize: 192576* |
| *Height: 272* | *HorzResolution: 0* |
| *BitDepth: 24* | *VertResolution: 0* |
| *ColorType: 'truecolor'* | *NumColorsUsed: 0* |
| | *NumImportantColors: 0* |

**Realizado por:** Villagomez Herrera, Ximena, 2020.

## ENTORNO FILTRADO_PRESENCIADERUIDO

A continuación, se presenta la ventana principal del entorno gráfico **FILTRADO_PRESENCIADERUIDO (figura 2-3),** para el inicio del procesamiento de imágenes mediante el filtrado.


**Figura 4-3.** Interfaz Gráfica para Implementación de Filtros en la Restauración.

**Realizado por:** Villagomez H, Ximena A. 2020

## 3.3. Restauración de la imagen de prueba mediante el proceso de filtrado.

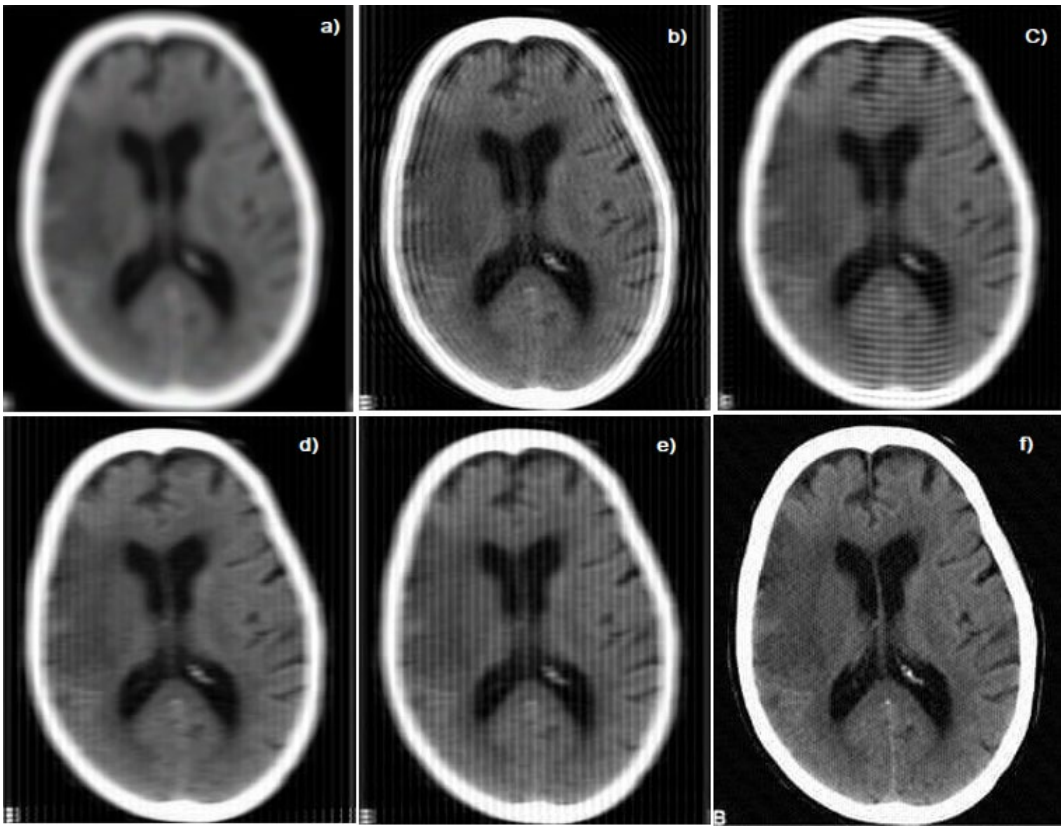



**Figura 5-3.** Restauración mediante Filtros. a) Imagen Original, b) Wiener, c) Regularizado, d) Filtro Laplaciano, e) Filtro Lagrangiano y f) Filtro Inverso.

**Realizado por:** Villagomez H, Ximena A. 2020

La **figura 5-3,** muestra los resultados obtenidos en cada uno de los procesos de filtrado con el fin de evaluar la efectividad de los filtros empleados para el procesamiento de imágenes almacenadas con el formato bmp.

Los resultados obtenidos de la aplicación del filtro Wiener (**figura 5-3 b**) demuestran que este tipo de filtro no logra conseguir una restauración aceptable ya que impide que se puedan filtrar las altas frecuencias de la imagen degradada por lo mismo no se puede recuperar ciertos detalles.
La **figura 5-3 c,** muestra la imagen resultante después de aplicar el filtro Regularizado, en la imagen se observa un importante nivel de atenuación de ruido, pero la presencia de bandas de color blanco a lo ancho de la imagen provoca la pérdida de detalles importantes de la misma por lo tanto este filtro no es óptimo para el procesamiento de imágenes digital.

Este tipo de filtro generalmente se utiliza por la gran sensibilidad que posee al ruido presente en una imagen digital, por lo tanto, la imagen resultante de la filtración (**figura 5.3 d**), demuestra una restauración considerable, aunque es evidente la presencia de ruido, producen bordes dobles y no brinda la información necesaria, la cual, no permite obtener una restauración total de la imagen original.
En la **figura 5.3 e,** se observa los cambios obtenidos en el procesamiento de imágenes mediante el filtro Lagrangiano, al analizar visualmente la imagen resultante se observa un nivel de suavizado muy aceptable que permite una mejor resolución de la imagen, pero al igual que el filtro Laplaciano aún hay presencia de ruido en forma de bandas blancas a lo largo de la imagen, en nuestro caso el valor de landa es muy alto sin embargo no se muestra una óptima estimación de la imagen procesada.

En la **figura 5-3 f,** se demuestra el resultado de una imagen procesada mediante el filtro Inverso, donde se logra remover totalmente el ruido de la imagen degradada, se conserva los detalles de los bordes, realce de contraste donde se acentúan los detalles y no hay perdida de información, lo cual nos permite generar una estimación bastante aceptable de la imagen procesada. En la misma ya se puede observar que el diámetro Interhemisférico frontal es de normal morfología, el sistema ventricular es normal.
Sin embargo, el diámetro de la asta frontal derecha varía en comparación a la asta izquierda el diámetro de las astas occipitales varía y de igual forma el diámetro antero-posterior de la encrucijada ventricular derecha es mayor al de lado izquierdo, además se muestra una calcificación en la asta occipital derecho.

### 3.4. Comparación de Resultados del valor de MSE de cada Filtro.

**Tabla 5-3:** Variación del valor de MSE para cada filtro propuesto.

| Desempeño del Filtrado de Imágenes | |
|---|---|
| **FILTRO** | **MSE** |
| Wiener | 0.00959479 |
| Regularizado | 0.00979593 |
| Laplaciano | 0.0077639 |
| Lagrangiano | 0.013881 |
| Inverso | 0.00091249 |

**Realizado por:** Villagomez Herrera, Ximena, 2020.

De acuerdo con los resultados obtenidos después de la simulación del filtrado de imágenes (**tabla 4-3**), se puede medir el efecto y desempeño de cada uno de los algoritmos correspondiente a cada filtro propuesto. Para esto es necesario la utilización de ciertos parámetros que nos permita medir la efectividad del algoritmo propuesto.

El parámetro de la razón señal – ruido (SNR) se basa en el error cuadrático medio (MSE) para medir la calidad de una imagen.

A partir de los valores de MSE obtenidos mediante la función *immse(imagenfilt,imagen)* (**1**) de Matlab se obtiene el error que existe entre la señal de la imagen filtrada con respecto a la señal de la imagen de prueba, con la ecuación **SNR** se puede calcular los valores del parámetro SNR correspondientes a cada filtro.

$$SNR = 20\ log_{10}\left(\frac{256-1}{\sqrt{MSE}}\right)$$

Los resultados de aplicar la fórmula 1 se muestran en la **Tabla 5-3**.

**Tabla 6-3:** Valor de SNR para cada Filtro.

| Filtrado de Imágenes | | |
|---|---|---|
| **Filtro** | **MSE** | **SNR** |
| Wiener | 0.00959479 | 68.31044 |
| Regularizado | 0.00979593 | 68.22034 |
| Laplaciano | 0.0077639 | 69.22861 |
| Lagrangiano | 0.013881 | 66.70659 |
| Inverso | 0.00091249 | 78.52512 |

**Realizado por:** Villagomez Herrera, Ximena, 2020.

La **Tabla 6-3** muestra los resultados del cálculo de MSE y del SNR después de ejecutar la función (**1**) de Matlab y aplicar la ecuación ***2*** de forma manual que evalúa la efectividad de los filtros propuestos para el procesamiento de imágenes. Los valores obtenidos de SNR y MRS indican que el filtro Wiener (**figura 5-3 b**) presenta una ligera mejoría que se puede apreciar en la calidad de restauración de la imagen.

Los resultados obtenidos demuestran claramente la eficiencia del filtro Inverso en el proceso de restauración de imágenes degradadas o distorsionadas por el movimiento en diferentes grados de intensidad. Sin embargo para determinar que este filtro produce una respuesta excelente en el filtrado no solo es necesario una percepción visual de los detalles de la imagen procesada, sino también es necesario analizar los valores de SNR y MSR (**Tabla 6-3**) se logra remover el ruido de la imagen degradada de una forma bastante aceptable lo cual nos permite obtener una buena estimación de la misma, ya que el valor de MRS es cercano a cero (*0.000912492*) y el valor de SNR es muy alto (*78.52512*).

Esto quiere decir que la relación señal-ruido al ser muy alto se tiene mayor equilibrio entre la eliminación del ruido y mejor se conserva los detalles e información de la imagen procesada por lo que es totalmente factible la utilización de este filtro.

### 3.5. Histogramas obtenidos durante el proceso de filtrado

El histograma de una imagen describe la frecuencia relativa de los varios niveles de gris en término de número de *pixel*es en cada nivel de concurrencia, en el eje x se presenta la distribución de los tonos de grises y en el eje y se representa la frecuencia de ocurrencia. El histograma es una función de probabilidad que no contiene información espacial de la imagen, como se muestra en la **figura 7-3 a**.

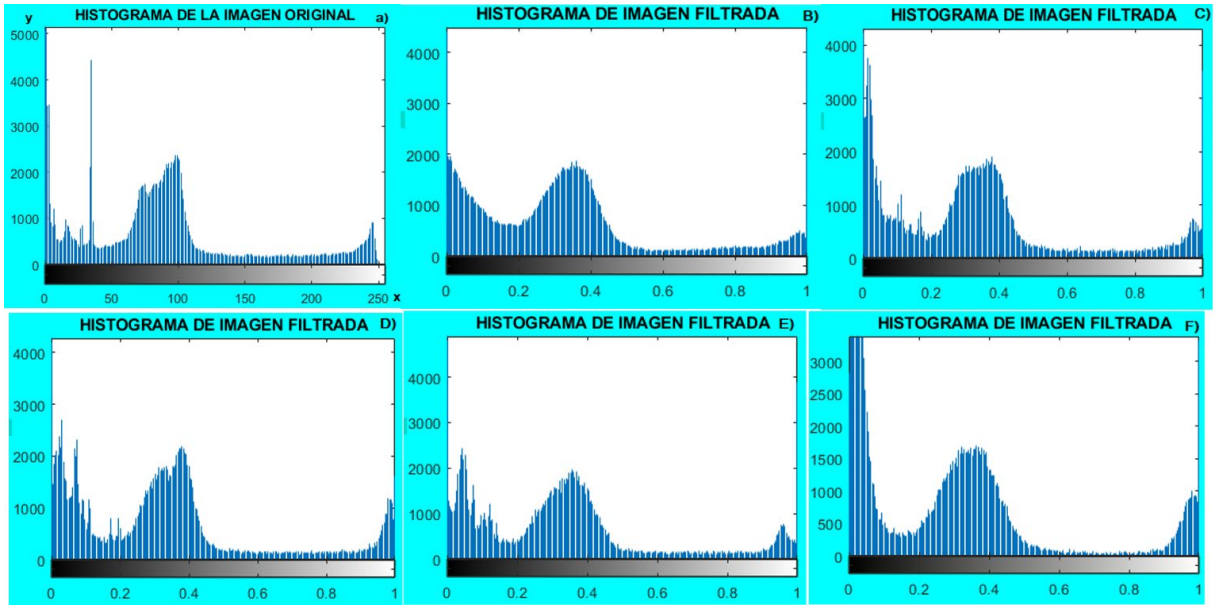



**Figura 6-3.** Histogramas de la Imagen Filtrada. a) Imagen de prueba, b) Wiener. c) Regularizado, d) Filtro Laplaciano, e) Filtro Lagrangiano y f) Filtro Inverso.

**Realizado por:** Villagomez H, Ximena A. 2020

La **figura 6-3** se determina que en el eje de las X de las imágenes se muestra la diferencia tonal que se distribuye entre el punto más obscuro y el más claro de la escala de grises, esto no significa que la imagen posea todos los tonos dentro de este intervalo dinamico de grises.

El histograma de la **figura 6-3 f** corresponde a la imagen procesada mediante el filtro Inverso, el cual describe la distribución homogénea de los niveles de gris en termino de numero d pixeles en cada nivel, provee información importante para determinar la efectividad del filtro como la intensidad media y la medida de contraste de la imagen que mientras mayor sea la dispersión a lo largo del eje X mayor es el contraste.

Además, podemos considerar que la imagen restaurada posee la diferencia entre sus niveles de gris que se encuentran en el rango de diferencia tonal desde el punto más claro al más oscuro de una imagen (eje X).

# CONCLUSIONES

En el presente capítulo se detalla las conclusiones a las que se llegó después analizar los resultados obtenidos durante el desarrollo de los métodos matemático y computacional propuestos en el presente Trabajo de Titulación.

1. Con el estudio y desarrollado de los principales parámetros que determinan la calidad de una imagen digital, se determinó que la calidad de una imagen se basa en la efectividad de los parámetros de reconstrucción como el filtrado y la previa visualización del observador.

2. El proceso de restauración de imágenes es un método especialmente para generar imágenes de mejor calidad, ya sea en el dominio del espacio como en el dominio frecuencial tomando en consideración su complejidad.

3. Para realizar una simulación mediante Matlab es importante la aplicación de algoritmos propuestos a fin de optimizar el tiempo de procesamiento de imágenes utilizando un método con el menor número de operaciones posible y además el diseño de diagramas para una mejor comprensión de los efectos de los filtros sobre las imágenes de prueba.

4. El cálculo de SNR (relación señal-ruido) determina la efectividad del rendimiento cuantitativo de los diversos filtros implementados.

5. Los resultados obtenidos en la eliminación de ruido mediante el filtro Inverso fueron excelentes debido a que presenta las mejores condiciones y características para eliminar la cantidad de ruido conservando detalles de la imagen, demostrando un rendimiento superior al resto de filtros puestos a prueba pues el valor de SNR que se obtuvo fue el máximo en comparación a cada caso. Los filtros Wiener, Laplaciano y Lagrangiano cancelan o disminuye una gran cantidad de ruido sin embargo no evita que haya perdida de detalles e información importante en la visualización de la imagen procesada, logrando imágenes con el doble del valor de SNR.

6. El tiempo de procesamiento y restauración de imágenes dependen del tamaño, formato y tipo de imagen de prueba, el tipo de filtro aplicado y el computador compatible con imágenes de 256 tonos de grises con un tamaño de $256x256$ pixeles donde se ejecuta el programa desarrollado. Por lo tanto, los tiempos de restauración de imágenes fueron muy buenos que van de 5 a 12 segundos.

# RECOMENDACIONES

1. Obtener evidencia teórica adecuada acerca de las diversas causas que provocan la presencia de ruido o distorsión en las imágenes digitales, con la finalidad de generar un algoritmo computacional eficiente en solucionar este problema en específico.

2. Utilizar funciones computacionales fáciles de Matlab para implementar algoritmos de programación que ayude reducir el tiempo de procesamiento de imágenes.

3. Se recomienda no utilizar valores iguales a 1 en cada parámetro necesario para la ejecución de los filtros debido a que con el valor máximo es con el que se pierde menor cantidad de detalles de la imagen y con valores menores se producen perdida de información y detalles generando presencia de ruido en la imagen de prueba.

## GLOSARIO

**FILTRO DIGITAL:** es un filtro que opera sobre señales digitales. Es una operación matemática que toma una secuencia de números (la señal de entrada) y la modifica produciendo otra secuencia de números (la señal de salida) con el objetivo de resaltar o atenuar ciertas características (Jure et al., 2011).

**COEFICIENTE DE ATENUACIÓN LINEAL** μ: es la capacidad que posee un material para detener fotones, es directamente proporcional al número atómico del material (Z) y su densidad es inversamente proporcional con la energía (Giraldo et al., 2008)

**RESOLUCIÓN ESPACIAL:** es la capacidad que posee un sistema para distinguir todos los detalles o estructuras de dimensiones muy pequeñas o bordes nítidos (Benito & Conejero, 2015).

**PROYECCIÓN:** Es la dirección del rayo central respecto de las caras y planos del cuerpo. Describe el trayecto del RC del haz de rayo X cuando atraviesa al paciente y proyecta una imagen sobre el Receptor de imagen («Posiciones y Proyecciones», 2016).

**BIBLIOGRAFÍA**

**AGUILAR, G**., Procesamiento Digital de Imágenes utilizando Filtros Morfológicos. [en línea]. (Trabajo de Titulación) (Pregrado), Escuela Politécnica Nacional. Quito, Ecuador. 1995. pp. 23-30. [Consulta: 10 diciembre 2019]. Disponible en: https://bibdigital.epn.edu.ec/bitstream/15000/5249/1/T171.pdf. Ingenieria Electrica

**ALDALUR, B. y SANTAMARÍA, M.**, Filtrado Espacial., vol. 17, no. 2. 2002. pp. 12.

**AMED, A. y ROJAS, L**., Algoritmo de tomografía local basado en la transformada discreta wavelet [en línea]. (Trabajo de Titulación) (Maestría), Universidad EAFIT. Medellín, Colombia. 2011. pp. 176-230. [Consulta: 4 junio 2019]. Disponible en: https://repository.eafit.edu.co/xmlui/bitstream/handle/10784/152/AmedAlfonso_AlfonsoCristancho_2011.pdf;jsessionid=47AA8163BAD997E857FD00B471C27C1D?sequence=3.

**BENITO, C.**, Diseño de Filtros para el Procesado de Imágenes basados en Teoría de Grafos. Valencia, España.: Universidad Politécnica de Valencia. 2015. pp. 103.

**CALZADO, A. y GELEIJNS, J**., "Tomografía Computarizada". *Revista de Física Médica* [en línea], vol. 11, no. 3 (2010). [Consulta: 8 julio 2019]. ISSN 1576-6632. Disponible en: https://revistadefisicamedica.sefm.es/index.php/rfm/article/view/115.

**CASTILLO, A.**, Mejoramiento de Imágenes de Tomografía Axial Computarizada del Corazón empleando Filtrado de Difusión Anisotrópica. (Trabajo de Titulación) (Grado), Universidad Tecnológica de Bolívar. Cartagena, Colombia. 2016. pp. 17-23.

**COCA, L., FRANCO, Z. y PATETI, A**., Implementación de Filtros Morfológicos utilizados en el Procesamiento de Imágenes Digitales en un Dispositivo Lógico Programable. [en línea], vol. 12, no. 18 (2008) pp. 171-182. [Consulta: 10 abril 2020]. Disponible en: https://pdfs.semanticscholar.org/8f83/363841c9e840e485b06f36039f4389b4ae3e.pdf.

**DILLENSEGER, J. y MOERSCHEL, E.**, Manual para Técnicos Radiólogo. Argentina: Journal. 324-330. ISBN 98789871259618. 2012.

**FENOLL, I.**, Aportaciones a la Segmentación y Caracterización de Imágenes Médicas 3D [en línea]. (Trabajo de Titulación) (Grado), Universidad de Sevilla. Sevilla, España. 2010. pp. 171-

180. [Consulta: 6 abril 2020]. Disponible en: http://bibing.us.es/proyectos/abreproy/11854/fichero/Volumen+1%252FCapitulo+2.pdf.

**FERRERO, J**., Filtrado de artefactos en imágenes médicas de TAC cuando se emplean pocas proyecciones. (Trabajo de Titulación) (Grado), Universidad Politécnica de Valencia. Valencia. Valencia-España. 2017. pp. 5-12

**GARCÍA, J.**, La Transformada de Radon y su Aplicación en la Tomografía Axial Computarizada. [en línea]. (Trabajo de Titulación) (Licenciatura), Universidad Tecnológica de la Mixteca. Oaxaca, México. 2015. pp. 1-3. [Consulta: 27 noviembre 2019]. Disponible en: http://jupiter.utm.mx/~tesis_dig/12549.pdf.

**GIRALDO, J., ARBOLEDA, C. y MCCOLLOUGH, C.**, "Tomografía computarizada por rayos X". En: Universidad CES, *Revista Ingeniería Biomédica*, vol. 2, no. 4 (2008) pp. 19. ISSN 1909-9762.

**GIRALDO, J., FLETCHER, J. y MCCOLLOUGH, C**., "Reducción del ruido en imágenes de tomografía computarizada usando un filtro bilateral anisotrópico". *Revista Ingeniería Biomédica*, vol. 4, no. 7 (2010). pp. 1-8. ISSN 1909-9762.

**GRAFFIGNA, J. y ROMO, R**., 2003. Fundamentos de Imágenes Médicas. [en línea]. S.l.: s.n. Disponible en: http://dea.unsj.edu.ar/imagenes/recursos/fundamentos.pdf.

**JIMÉNEZ, C**., Interpolación de los datos de Radón calculados a partir de proyecciones tomográficas Cone-Beam en trayectoria circular [en línea]. Tesis de Maestría, Universidad Nacional de Colombia. Bogotá-Colombia. 2012. pp. 8-14. [Consulta: 6 abril 2020]. Disponible en: http://bdigital.unal.edu.co/6399/1/299710.2012.pdf.

**JIMÉNEZ, J**., Filtro de Impedancia con Característica de Magnitud Máximamente Plana [en línea]. Universidad de las Américas Puebla. Puebla, México. 2007. pp. 1-3. [Consulta: 27 noviembre 2019]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/meie/jimenez_u_jr/capitulo1.pdf.

**JURE, L., LÓPEZ, E. y ROCAMORA, M.**, Introducción a los Filtros Digitales. [en línea]. S.l. vol. 2, no. 4 (2011). pp. 18. [Consulta: 11 junio 2020]. Disponible en: https://www.eumus.edu.uy/eme/ensenanza/electivas/dsp/presentaciones/clase10.pdf.

**LÓPEZ, G.**, Implementación de Algoritmos de Procesamiento de Imágenes FPGA [en línea]. (Trabajo de Titulación) (Grado), Instituto Politécnico Nacional. 2014. México, D.F.: pp. 23-30.

[Consulta: 3 diciembre 2019]. Disponible en: http://148.204.63.111/SABERv3/Repositorios/webVerArchivo/26041.

**MOLINA, C.**, Corrección del Artefacto de Endurecimiento de Haz (BEAM HARDENING) para Imágenes obtenidas con Tomógrafo de Rayos X de pequeños animales [en línea]. Universidad Autónoma de Madrid. Madrid, España. 2012. pp. 30-55. [Consulta: 27 noviembre 2019]. Disponible en: https://drive.google.com/file/d/0B1CbuFUAAF7tME5pWWh4dXFwUE0/view?usp=drive_open&usp=embed_facebook.

**PLAZAS, L.** Inventos de la medicina 2. *Enfermería Buenos Aires* [en línea]. 2018. [Consulta: 8 julio 2019]. Disponible en: https://enfermeriabuenosaires.com/historia-de-la-medicina-2.

**Posiciones y Proyecciones**. Radiodiagnóstico e Imagenología [en línea], 2016. [Consulta: 14 junio 2020]. Disponible en: https://radiodiagnosticoeimagenologia.wordpress.com/posiciones-y-proyecciones/.

**QUIROZ, G.**, Teoría de Filtros [en línea]. (Trabajo de Titulación) (Grado), Universidad de las Américas Puebla. Cholula, México: 2007a. pp. 30-55. [Consulta: 20 mayo 2020]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/quiroz_c_g/capitulo1.pdf. Ingeniería y Ciencias

**QUIROZ, G**., Teoría de Filtros [en línea]. (Trabajo de Titulación) (Grado), Universidad de las Américas Puebla. Cholula, México: 2007b. pp. 30-55. [Consulta: 20 mayo 2020]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/quiroz_c_g/capitulo2.pdf.

**RIVERA, Y., ARGÜELLO, I. y VÁZQUEZ, R.**, "Trombosis del seno venoso sagital superior en paciente pediátrico con enfermedad inflamatoria intestinal". *Revista de Gastroenterología*, vol. 2, no. 4 (2019). pp. 3. ISSN 0375-0906. DOI 10.1016/j.rgmx.2019.07.007.
**VEGA, C**., 2010. Tomografía Computada. pp. 31.

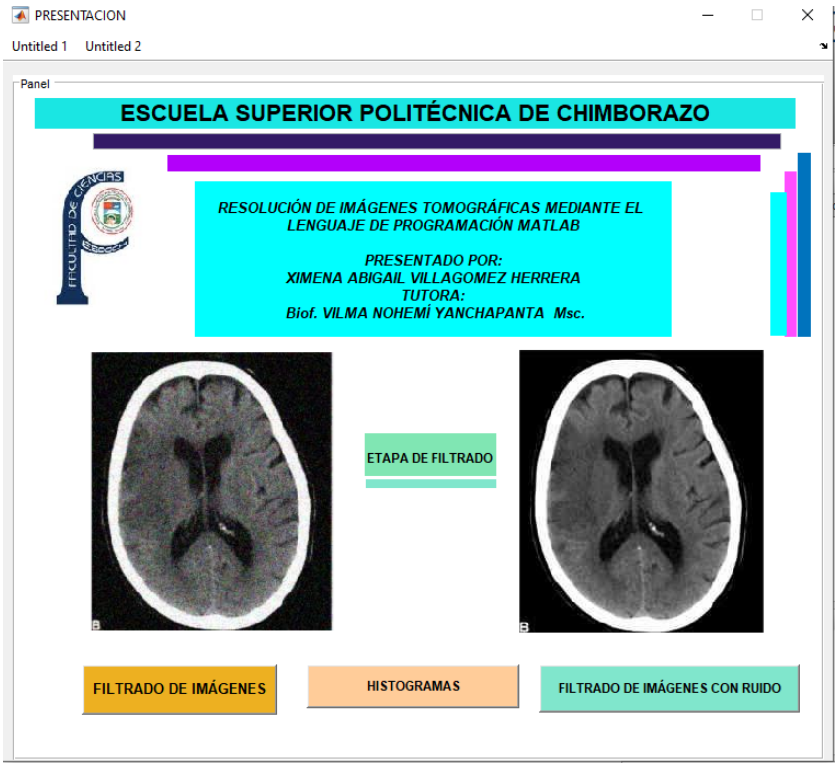
**VETTORAZZI, J.**, Restauración de Imágenes Distorsionadas mediante Técnicas de Procesamiento Digital y Comparación entre dos Métodos de Restauración. [en línea]. Guatemala: (Trabajo de Titulación) (Grado), Universidad de San Carlos. Guatemala. 2007. pp. 2-30. [Consulta: 4 febrero 2020]. Disponible en: http://biblioteca.usac.edu.gt/tesis/08/08_0203_EO.pdf. Escuela de Ingeniería Mecánica Eléctrica

**YÁÑEZ, A.**, *Formación de Imagen en TC.* [Blog]. 2016.[Consulta: 9 julio 2019]. Disponible en: https://es.slideshare.net/beleeny/formacin-de-imagen-en-tc.

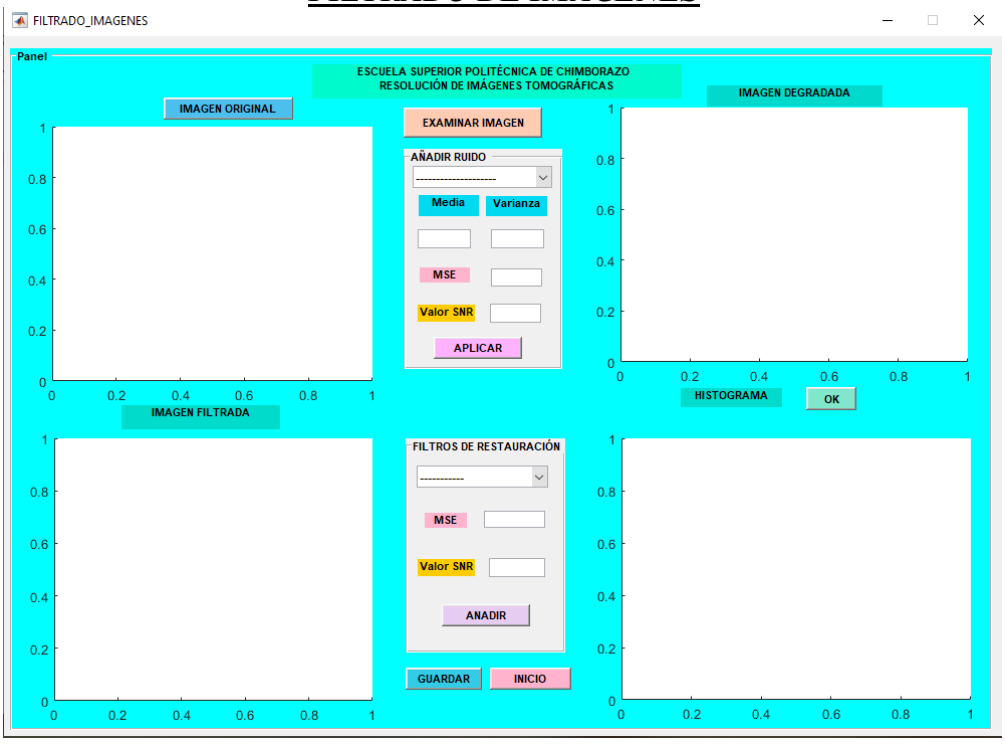
**ANEXOS**

ANEXO A. Manual de la interfaz gráfica

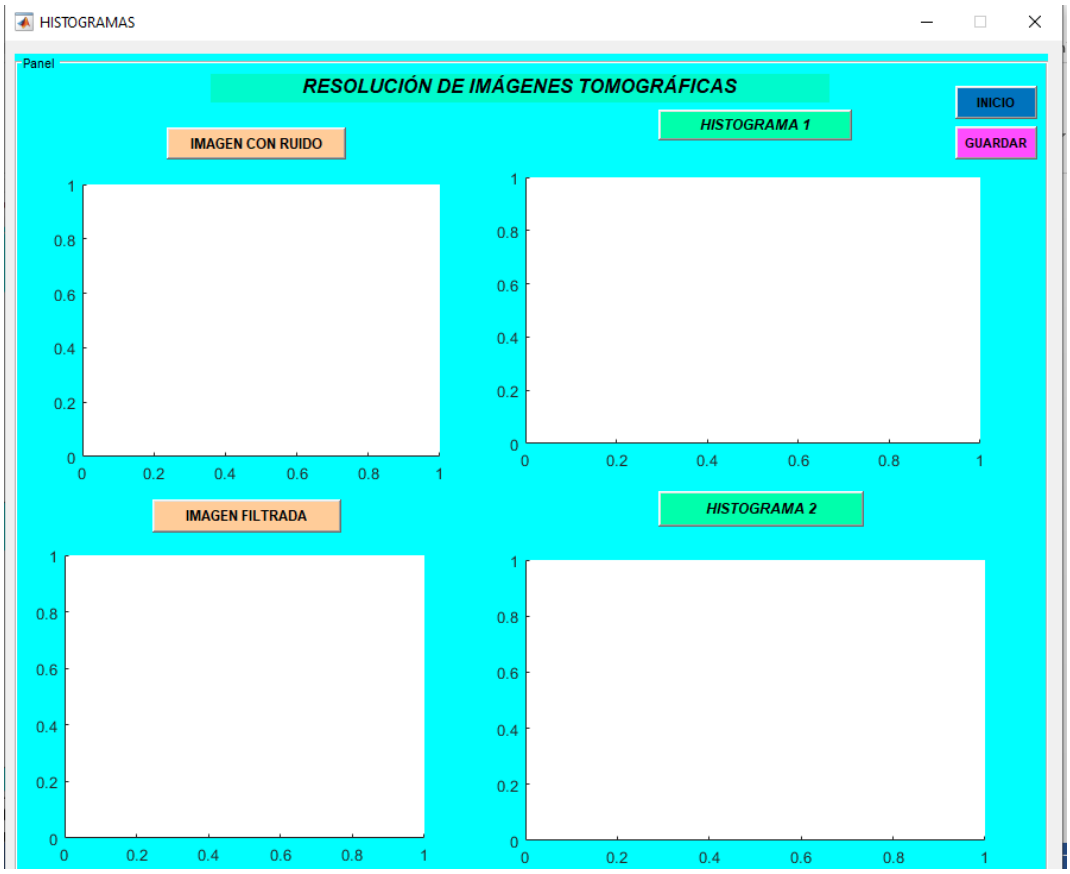
***PRESENTACIÓN***



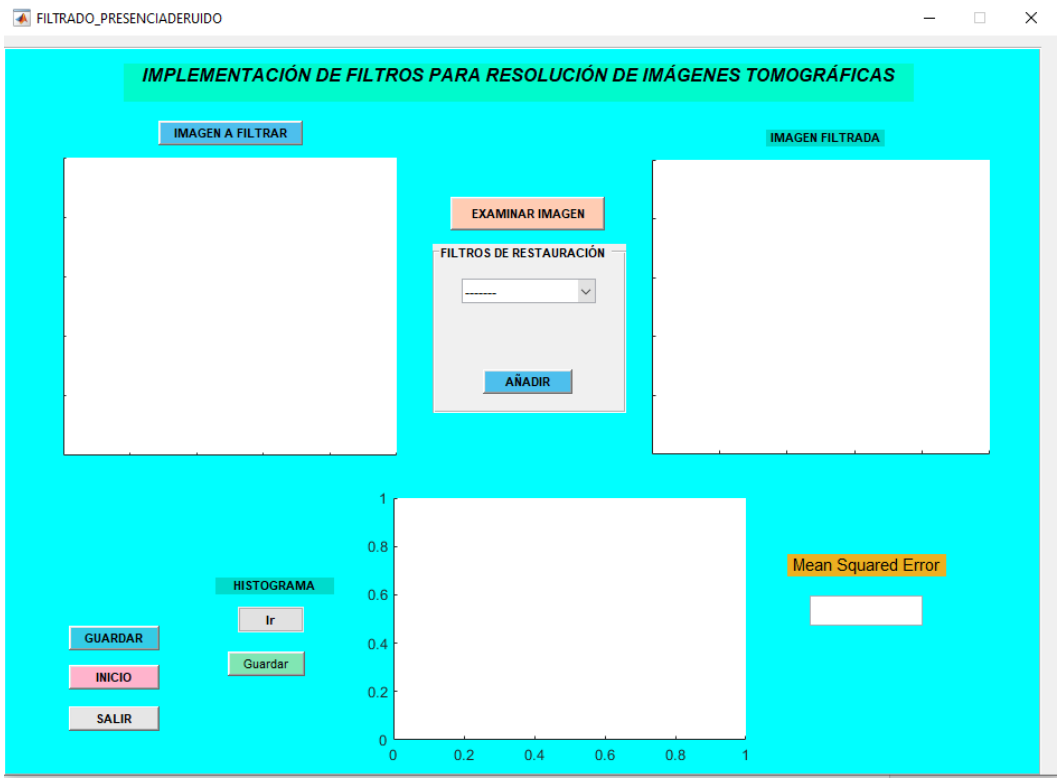

***FILTRADO DE IMÁGENES***




***HISTOGRAMAS***




# *FILTRADO DE IMÁGENES CON RUIDO*



ANEXO B. *Código fuente de la GUIDE "PRESENTACIÓN"*

```
function varargout = PRESENTACION(varargin)
% PRESENTACION MATLAB code for PRESENTACION.fig
%      PRESENTACION, by itself, creates a new PRESENTACION or
raises the existing
%      singleton*.
%
%      H = PRESENTACION returns the handle to a new PRESENTACION
or the handle to
%      the existing singleton*.
%
%      PRESENTACION('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in PRESENTACION.M with the given
input arguments.
%
%      PRESENTACION('Property','Value',...) creates a new
PRESENTACION or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before PRESENTACION_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to PRESENTACION_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
PRESENTACION

% Last Modified by GUIDE v2.5 11-Feb-2020 11:12:30

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @PRESENTACION_OpeningFcn,
...
                   'gui_OutputFcn',  @PRESENTACION_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before PRESENTACION is made visible.
function PRESENTACION_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PRESENTACION (see
VARARGIN)
%Representamos imagen en figura, con su mapa de colores

% Choose default command line output for PRESENTACION
handles.output = hObject;
imshow('TAC.bmp')
axis off
axes(handles.axes2);

imshow('imagenconruido.jpg')
axis off
axes(handles.axes1);

imshow('ESPOCH.bmp')
axis off
axes(handles.axes5);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PRESENTACION wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout = PRESENTACION_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on mouse press over axes background.
function axes1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on mouse press over axes background.
function axes2_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in FILTRADO_IMAGENES.
function FILTRADO_IMAGENES_Callback(hObject, eventdata, handles)
% hObject    handle to FILTRADO_IMAGENES (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
FILTRADO_IMAGENES
close PRESENTACION

% --- Executes on button press in IMAGENES_DEGRADADAS.
function IMAGENES_DEGRADADAS_Callback(hObject, eventdata,
handles)
% hObject    handle to IMAGENES_DEGRADADAS (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
FILTRADO_PRESENCIADERUIDO
close PRESENTACION

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
HISTOGRAMAS
close PRESENTACION
```

```
function varargout = FILTRADO_IMAGENES(varargin)
% FILTRADO_IMAGENES MATLAB code for FILTRADO_IMAGENES.fig
%      FILTRADO_IMAGENES, by itself, creates a new
FILTRADO_IMAGENES or raises the existing
%      singleton*.
%
%      H = FILTRADO_IMAGENES returns the handle to a new
FILTRADO_IMAGENES or the handle to
%      the existing singleton*.
%
%
FILTRADO_IMAGENES('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in FILTRADO_IMAGENES.M with the
given input arguments.
%
%      FILTRADO_IMAGENES('Property','Value',...) creates a new
FILTRADO_IMAGENES or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before FILTRADO_IMAGENES_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to
FILTRADO_IMAGENES_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
FILTRADO_IMAGENES

% Last Modified by GUIDE v2.5 27-Apr-2020 18:27:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@FILTRADO_IMAGENES_OpeningFcn, ...
                   'gui_OutputFcn',
@FILTRADO_IMAGENES_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before FILTRADO_IMAGENES is made visible.
function FILTRADO_IMAGENES_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to FILTRADO_IMAGENES (see
VARARGIN)

% Choose default command line output for FILTRADO_IMAGENES
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FILTRADO_IMAGENES wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout = FILTRADO_IMAGENES_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton15.


% --- Executes on button press in EXAMINAR_IMAGEN.
function EXAMINAR_IMAGEN_Callback(hObject, eventdata, handles)
% hObject    handle to EXAMINAR_IMAGEN (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nombre ruta]=uigetfile('*.*','Seleccionar imagen');% Formato de
imágenes (todos)
```

```
if nombre == 0
    return
end
imagen=imread(fullfile(ruta,nombre)); %Lee imagen desde ruta
set(handles.IMAGEN_ORIGINAL,'Enable','on')
handles.img_orig=imagen;
guidata(hObject, handles)

%image(imagen,'parent',handles.axes1)
set(handles.APLICAR,'Enable','on')
set(handles.RECUPERAR,'Enable','on')


guidata(hObject, handles)

function IMAGEN_ORIGINAL_Callback(hObject, eventdata, handles)
% hObject    handle to IMAGEN_ORIGINAL (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;

image(imagen,'parent',handles.axes17)
%imshow(imagen,'parent',handles.axes1)

guidata(hObject,handles)
% --- Executes on button press in HISTOGRAMA.
function HISTOGRAMA_Callback(hObject, eventdata, handles)
% hObject    handle to HISTOGRAMA (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;
axes(handles.axes17)
% Mostrar el histograma I
axes(handles.axes21)
imhist(imagen);
title ('HISTOGRAMA DE LA IMAGEN ORIGINAL');
% --- Executes on button press in GUARDAR.
function GUARDAR_Callback(hObject, eventdata, handles)
% hObject    handle to GUARDAR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

rgb = getimage(handles.axes19); %Obtiene imagen desde cuadro
mencionado
if isempty(rgb), return, end
%Almacenamiento de imagen deseada
formatos = {'*.jpg','JPEG
(*.jpg)';'*.bmp','BMP(*.bmp)';'*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(rgb,fName);
```

```
% --- Executes on selection change in popupmenu9.
function popupmenu9_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu9 contents as cell array
%        contents{get(hObject,'Value')} returns selected item
from popupmenu9
val=get(handles.popupmenu9,'Value');
switch val
    case 1 % SIN ELECCION

    case 2% RUIDO DE TIPO GAUSSIANO
        set(handles.text23,'String','Varianza')
        set(handles.text24,'String','Media')

    case 3 % RUIDO POISSON
        set(handles.text24,'String','K n.ocur')
        set(handles.text23,'String','Theta')

    case 4 % RUIDO DE TIPO SAL Y PIMIENTA

        set(handles.edit1,'String',0) % Probabilidad
        set(handles.edit2,'String',0.05)
    otherwise % % RUIDO SPECKLE O PUNTO
        set(handles.text23,'String','Varianza')
        set(handles.text24,'String','Media')
        set(handles.edit1,'String',0)      % Media
        set(handles.edit2,'String',0.04) % Varianza
end

% --- Executes during object creation, after setting all
properties.
function popupmenu9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in APLICAR.
function APLICAR_Callback(hObject, eventdata, handles)
% hObject    handle to APLICAR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;
% Obtener el tipo de ruido
valor=get(handles.popupmenu9,'value');
% Obtener la media y la varianza
edit1=str2double(get(handles.edit1,'String'));
edit2=str2double(get(handles.edit2,'String'));
% Control de errores
if edit1<0||isempty(edit1)||isnan(edit1)
    errordlg('Error de ingreso en el primer parametro','ERROR')
    return
end
if valor ~=2
    if edit2<0||isempty(edit2)||isnan(edit2)
        errordlg('Error de ingreso en el segundo
parámetro','ERROR')
        return
    end
end
% Aplicar el ruido
switch valor
    case 1 % SIN ELECCION

    case 2
        J = imnoise(imagen,'gaussian',edit1,edit2);


end
handles.distorsionada=J;
image(J,'parent',handles.axes19)
guidata(hObject,handles)
MSE = mean(mean(mean((imagen-J).^2)));
S = int2str(MSE);
set(handles.ECM_1, 'String', S)

%Calcule el PSNR.
SNR = psnr(J, imagen);
set(handles.edit6,'string',SNR);



% --- Executes on selection change in popupmenu10.
function popupmenu10_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu10 contents as cell array
%        contents{get(hObject,'Value')} returns selected item
from popupmenu10
val=get(handles.popupmenu10,'Value');
switch val

    case 3 % Disk
```

```
    case 4 % Gaussian
    case 5 % Motion
    otherwise % Unsharp
end

% --- Executes during object creation, after setting all
properties.
function popupmenu10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in RECUPERAR.
function RECUPERAR_Callback(hObject, eventdata, handles)
% hObject    handle to RECUPERAR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
J=handles.distorsionada;
valor=get(handles.popupmenu10,'value');
switch valor
    case 1 % SIN ELECCION

    case 2
        PSF = fspecial('disk',1);%point-spread function
    case 3
        PSF = fspecial('gaussian');%point-spread function

    otherwise
        PSF = fspecial('unsharp');%point-spread function
        J1 = imfilter(J,PSF,'circular','conv');
end
image1 = handles.img_orig;
image(J1,'parent',handles.axes18)
guidata(hObject,handles)
MSE = mean(mean(mean((image1-J1).^2)));
S = int2str(MSE);
set(handles.ECM_2, 'String', S)
%Calcule el PSNR.
SNR = psnr(J1, image1);
set(handles.edit7,'string',SNR);

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double


% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of
edit2 as a double


% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function ECM_1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to ECM_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ECM_1 as text
%        str2double(get(hObject,'String')) returns contents of
ECM_1 as a double


% --- Executes during object creation, after setting all
properties.
function ECM_1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ECM_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ECM_2_Callback(hObject, eventdata, handles)
% hObject    handle to ECM_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ECM_2 as text
%        str2double(get(hObject,'String')) returns contents of
ECM_2 as a double
% --- Executes during object creation, after setting all
properties.
function ECM_2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ECM_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in INICIO.
function INICIO_Callback(hObject, eventdata, handles)
% hObject    handle to INICIO (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
PRESENTACION
close FILTRADO_IMAGENES


% --- Executes on button press in SNR.
function SNR_Callback(hObject, eventdata, handles)
% hObject    handle to SNR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of
edit6 as a double
```

```
% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of
edit7 as a double


% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all
properties.
function text33_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text33 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
function varargout = FILTRADO_PRESENCIADERUIDO(varargin)
%FILTRADO_PRESENCIADERUIDO MATLAB code file for
FILTRADO_PRESENCIADERUIDO.fig
%      FILTRADO_PRESENCIADERUIDO, by itself, creates a new
FILTRADO_PRESENCIADERUIDO or raises the existing
%      singleton*.
%
%      H = FILTRADO_PRESENCIADERUIDO returns the handle to a new
FILTRADO_PRESENCIADERUIDO or the handle to
%      the existing singleton*.
%
%      FILTRADO_PRESENCIADERUIDO('Property','Value',...) creates
a new FILTRADO_PRESENCIADERUIDO using the
%      given property value pairs. Unrecognized properties are
passed via
%      varargin to FILTRADO_PRESENCIADERUIDO_OpeningFcn.  This
calling syntax produces a
%      warning when there is an existing singleton*.
%
%      FILTRADO_PRESENCIADERUIDO('CALLBACK') and
FILTRADO_PRESENCIADERUIDO('CALLBACK',hObject,...) call the
%      local function named CALLBACK in
FILTRADO_PRESENCIADERUIDO.M with the given input
%      arguments.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
FILTRADO_PRESENCIADERUIDO

% Last Modified by GUIDE v2.5 28-Apr-2020 21:43:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@FILTRADO_PRESENCIADERUIDO_OpeningFcn, ...
                   'gui_OutputFcn',
@FILTRADO_PRESENCIADERUIDO_OutputFcn, ...
                   'gui_LayoutFcn',  [], ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
```

```
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before FILTRADO_PRESENCIADERUIDO is made
visible.
function FILTRADO_PRESENCIADERUIDO_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from
the
%            command line (see VARARGIN)

% Choose default command line output for
FILTRADO_PRESENCIADERUIDO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FILTRADO_PRESENCIADERUIDO wait for user response
(see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout =
FILTRADO_PRESENCIADERUIDO_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in EXAMINAR_1.
function EXAMINAR_1_Callback(hObject, eventdata, handles)
% hObject    handle to EXAMINAR_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nombre ruta]=uigetfile('*.*','Seleccionar imagen');% Formato de
imágenes (todos)
if nombre == 0
    return
end
```

```
imagen=imread(fullfile(ruta,nombre)); %Lee imagen desde ruta
handles.img_orig=imagen;
guidata(hObject, handles)

% --- Executes on button press in IMAGEN_ORIGINAL.
function IMAGEN_ORIGINAL_Callback(hObject, eventdata, handles)
% hObject    handle to IMAGEN_ORIGINAL (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;
psf=fspecial('motion',8,4); %Función de desenfoque-degradacion
otf = psf2otf(psf,size(j)); %Transformada de función de
degradación
d=otf.*k; %Aplicación de degradación a imagen dominio de la
frecuencia
imagennoise=abs(ifft2(d)); %Transformada inversa de imagen
degradada para regresar al dominio del espacio
image(imagennoise,'parent',handles.axes20) %Muestra imagen
degradada


% --- Executes on button press in HISTOGRAMA_.
function HISTOGRAMA__Callback(hObject, eventdata, handles)
% hObject    handle to HISTOGRAMA_ (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagenfilt=handles.img_filt;
axes(handles.axes19)
imhist(imagenfilt); % Muestra histograma de imagen filtrada
title ('HISTOGRAMA DE IMAGEN FILTRADA');

% --- Executes on button press in INICIO.
function INICIO_Callback(hObject, eventdata, handles)
% hObject    handle to INICIO (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
PRESENTACION %Muestra portada de proyecto
close FILTRADO_PRESENCIADERUIDO


% --- Executes on button press in GUARDAR_IMAGEN1.
function GUARDAR_IMAGEN1_Callback(hObject, eventdata, handles)
% hObject    handle to GUARDAR_IMAGEN1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
rgb = getimage(handles.axes27); %Obtiene imagen desde cuadro
mencionado
if isempty(rgb), return, end
%Almacenamiento de imagen deseada
formatos = {'*.jpg','JPEG
(*.jpg)';'*.bmp','BMP(*.bmp)';'*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
```

```
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(rgb,fName);


function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit18 as
text
%        str2double(get(hObject,'String')) returns contents of
edit18 as a double


% --- Executes during object creation, after setting all
properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenu10.
function popupmenu10_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;
imagennoise=handles.img_dist;
noise_var = 0.00035; %Varianza Ruido
PSF = fspecial('motion',6,3); %Función de dispersión de puntos
imagend=im2double(imagen);
val = get(hObject,'Value');


%Opciones de filtros
switch val

case 1 %Sin selección

case 2 %Selección Filtro Wiener en menú

    V = .000013; %Varianza
```

```
    imagennoise1 = imnoise(imagennoise,'gaussian',0,V);
%Aplicación de ruido Gaussiano de varianza V
    estimated_nsr = noise_var / var(imagend(:));
    imagenfilt = deconvwnr(imagennoise1, PSF, estimated_nsr);
%Aplicación deconvolución filtro Wiener

case 3  % Seleccion Filtro Regularizado

   PSF = fspecial('gaussian',6,3);
%%Reduzca la amplificación de ruido a lo largo del contorno de
la imagen antes de la deconvolución.
   %Se observa cómo la restauración de la imagen se vuelve menos
sensible al parámetro de energía de ruido
   imagenfilt = deconvreg(Edged,PSF);%devuelve la imagen
desborrosa restringido en el sentido de menor error cuadrado de
la imagen estimada.

case 4  % Seleccion Filtro Reg. Laplaciano

    PSF1 = fspecial('gaussian',6,3);
    Blurred = imfilter(imagennoise,PSF1);
%Reduzca la amplificación de ruido a lo largo del contorno de la
imagen antes de la deconvolución.
    %Se observa cómo la restauración de la imagen se vuelve
menos sensible al parámetro de energía de ruido
    C=0.0000143; %el multiplicador de Lagrange correspondiente,
LAGRA
    REGOP = [-1 -1 24 -1 -1]; % [-1 4 -1],[-1 -1 24 -1 -1],[-1 8
-1]
    imagenfilt = deconvreg(imagennoise,PSF,[],C,REGOP);
%LAGRANGE debe ser una matriz escalar o de dos escalas.


case 5 % Seleccion Filtro Reg. Lagrange

    PSF2 = fspecial('motion',4,3);
    Blurred = imfilter(imagennoise,PSF2,'conv');
    LAGRA=1/LANDA;
    imagenfilt = deconvreg(Edged,PSF2,[],LAGRA/100); %para
debilitar la restriccion se utiliza LAGRA/100

case 6 %Selección Filtro Inverso en menú

    psf2=fspecial('motion',8,4.1);
    otf2=psf2otf(psf2,size(imagennoise));%Función de
transferencia de función de degradación
    p=fft2(imagennoise); %Función de transferencia de imagen
degradada
    R=p./otf2; %Filtro Inverso
(TFImagenDegradada/TFFunciónDegradación)
    imagenfilt=abs(ifft2(R)); %Transformada inversa de imagen
degradada para regresar al dominio del espacio

end

handles.img_dist=imagennoise;
```

```
handles.img_filt=imagenfilt;
guidata(hObject,handles)



% --- Executes during object creation, after setting all
properties.
function popupmenu10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in SALIR.
function SALIR_Callback(hObject, eventdata, handles)
% hObject    handle to SALIR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(handles.output); %Salir de Aplicación


% --- Executes on button press in EJECUTAR.
function EJECUTAR_Callback(hObject, eventdata, handles)
% hObject    handle to EJECUTAR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen=handles.img_orig;
imagen=im2double(imagen);
imagenfilt=handles.img_filt;
image(imagenfilt,'parent',handles.axes27) %Muestra imagen
filtrada
MSE = immse(imagenfilt, imagen); %Calcula el MSE entre imagen
filtrada e imagen original
%MSE = mean(mean(mean((imagen-imagenfilt).^2)));
set(handles.edit18,'string',MSE); %Muestra el MSE en pantalla


function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as
text
```

```
%        str2double(get(hObject,'String')) returns contents of
edit19 as a double


% --- Executes during object creation, after setting all
properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in guardarhisto.
function guardarhisto_Callback(hObject, eventdata, handles)
% hObject    handle to guardarhisto (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
hist = getimage(handles.axes19); %Obtiene imagen desde cuadro
mencionado
if isempty(hist), return, end
%Almacenamiento de imagen deseada
formatos = {'*.jpg','JPEG
(*.jpg)';'*.bmp','BMP(*.bmp)';'*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(hist,fName);

function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as
text
%        str2double(get(hObject,'String')) returns contents of
edit20 as a double

% --- Executes during object creation, after setting all
properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

ANEXO E. *Código fuente de la GUIDE "HISTOGRAMA"*

```
function varargout = HISTOGRAMAS(varargin)
% HISTOGRAMAS MATLAB code for HISTOGRAMAS.fig
%      HISTOGRAMAS, by itself, creates a new HISTOGRAMAS or
raises the existing
%      singleton*.
%
%      H = HISTOGRAMAS returns the handle to a new HISTOGRAMAS
or the handle to
%      the existing singleton*.
%
%      HISTOGRAMAS('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in HISTOGRAMAS.M with the given
input arguments.
%
%      HISTOGRAMAS('Property','Value',...) creates a new
HISTOGRAMAS or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before HISTOGRAMAS_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to HISTOGRAMAS_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help HISTOGRAMAS

% Last Modified by GUIDE v2.5 07-Feb-2020 17:28:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @HISTOGRAMAS_OpeningFcn,
...
                   'gui_OutputFcn',  @HISTOGRAMAS_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
```

```
end
% End initialization code - DO NOT EDIT


% --- Executes just before HISTOGRAMAS is made visible.
function HISTOGRAMAS_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to HISTOGRAMAS (see
VARARGIN)

% Choose default command line output for HISTOGRAMAS
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes HISTOGRAMAS wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout = HISTOGRAMAS_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in IMAGEN_RUIDO.
function IMAGEN_RUIDO_Callback(hObject, eventdata, handles)
% hObject    handle to IMAGEN_RUIDO (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nombre ruta]=uigetfile('*.*','Seleccionar imagen');% Formato de
imágenes (todos)
if nombre == 0
    return
end
imagen=imread(fullfile(ruta,nombre)); %Lee imagen desde ruta
set(handles.IMAGEN_RUIDO,'Enable','on')
image(imagen,'parent',handles.axes1)

% --- Executes on button press in HISTOGRAMA1R.
```

```
function HISTOGRAMA1R_Callback(hObject, eventdata, handles)
% hObject    handle to HISTOGRAMA1R (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen2=handles.img_orig;
axes(handles.axes3)
% Mostrar el histograma I
imhist(imagen2);
title ('HISTOGRAMA IMAGEN CON RUIDO');


% --- Executes on button press in IMAGEN_FILTRADA.
function IMAGEN_FILTRADA_Callback(hObject, eventdata, handles)
% hObject    handle to IMAGEN_FILTRADA (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nombre ruta]=uigetfile('*.*','Seleccionar imagen');% Formato de
imágenes (todos)
if nombre == 0
    return
end
imagen=imread(fullfile(ruta,nombre)); %Lee imagen desde ruta
set(handles.IMAGEN_FILTRADA,'Enable','on')
handles.img_orig=imagen;
guidata(hObject, handles)
image(imagen,'parent',handles.axes2)
% --- Executes on button press in HISTOGRA2F.
function HISTOGRA2F_Callback(hObject, eventdata, handles)
% hObject    handle to HISTOGRA2F (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
imagen1=handles.img_orig;
axes(handles.axes2)

% Mostrar el histograma I
axes(handles.axes4)
imhist(imagen1);
title ('HISTOGRAMA DE LA IMAGEN FILTRADA');


% --- Executes on button press in GUARDAR.
function GUARDAR_Callback(hObject, eventdata, handles)
% hObject    handle to GUARDAR (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
formatos = {'*.bmp','BMP (*.bmp)';'*.jpg','JPEG
(*.jpg)';'*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR HISTOGRAMA');
if nomb==0, return, end
% Crear nueva figura
imagen1 = imagen;
% Unidades y posición
```

```
unidades  = get(handles.axes4,'Units');
objeto_2  = copyobj(handles.axes4,imagen);
% Modificar la nueva figura
% Ajustar la nueva figura
set(imagen,'Units',unidades);
set(imagen,'Position',[15 5 posicion(3)+30 posicion(4)+10]);
% Guardar la gráfica
saveas(imagen,[ruta nomb])
%Cerrar figura
close(imagen)


% --- Executes on button press in INICIO.
function INICIO_Callback(hObject, eventdata, handles)
% hObject    handle to INICIO (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
PRESENTACION
close HISTOGRAMAS


% --- Executes during object deletion, before destroying
properties.
function text2_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS PARA EL APRENDIZAJE Y LA INVESTIGACIÓN**

**UNIDAD DE PROCESOS TÉCNICOS**

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

**Fecha de entrega:** 11/ 06 / 2020

| **INFORMACIÓN DE LA AUTORA** |
|---|
| **Nombres – Apellidos:** XIMENA ABIGIAL VILLAGOMEZ HERRERA |
| **INFORMACIÓN INSTITUCIONAL** |
| **Facultad:** CIENCIAS |
| **Carrera:** BIOFÍSICA |
| **Título a optar:** BIOFÍSICA |
| **f. Analista de Biblioteca responsable:** |