

UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE FÍSICA



**SOFTWARE DE PROCESAMIENTO DIGITAL DE IMÁGENES  
BASADO EN MORFOLOGÍA MATEMÁTICA, PARA  
APLICACIONES DOCENTES EN EL ÁREA DE  
INSTRUMENTACIÓN.**

Trabajo Especial de Grado presentado por  
Danny José Cruz Blanco  
ante la Facultad de Ciencias de la  
Ilustre Universidad Central de Venezuela  
como requisito parcial para optar al título  
de: **Licenciado en Física**  
Con la tutoría de: Prof. Levi García

Mayo-2014  
Caracas-Venezuela

UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE FÍSICA



**SOFTWARE DE PROCESAMIENTO DIGITAL DE IMÁGENES  
BASADO EN MORFOLOGÍA MATEMÁTICA, PARA  
APLICACIONES DOCENTES EN EL ÁREA DE  
INSTRUMENTACIÓN.**

Trabajo Especial de Grado presentado por  
Danny José Cruz Blanco  
ante la Facultad de Ciencias de la  
Ilustre Universidad Central de Venezuela  
como requisito parcial para optar al título  
de: **Licenciado en Física**  
Con la tutoría de: Prof. Levi García

Mayo-2014  
Caracas-Venezuela

Quienes suscriben, miembros del Jurado que examinó el trabajo presentado por el Br. **Danny J. Cruz B.** titulado: **Software de Procesamiento Digital de Imágenes Basado en Morfología Matemática, para Aplicaciones Docentes en el Área de Instrumentación**, para optar al título de Licenciado en Física, consideramos que dicho trabajo cumple con los requisitos exigidos por los reglamentos respectivos y por lo tanto lo declaramos APROBADO en nombre de la Universidad Central de Venezuela.

  
\_\_\_\_\_  
MSc. Levi García, Tutor

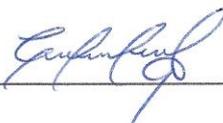
19/05/2014

Fecha

  
\_\_\_\_\_  
Dr. Miguel Martín

19/05/2014

Fecha

  
\_\_\_\_\_  
Lic. Zaida Medina

19/05/2014

Fecha

## **Resumen**

En el presente trabajo, hemos desarrollado un programa denominado PDI que procesa imágenes digitales almacenadas en formato BMP. Las operaciones más importantes que se implementaron en el programa son las de Morfología Matemática, y a partir de las fundamentales (erosión y dilatación) se implementaron funciones más complejas tales como: Filtros morfológicos, detección de bordes y granulometría. Uno de los objetivos de este trabajo es proporcionar un documento de Ayuda que suministre información sobre formato de imágenes BMP, Morfología Matemática y la implementación de estas funciones. En la ayuda también se encuentra disponible el código fuente del programa debidamente comentado.

## **Abstract**

In this paper, we have developed a program called PDI it processes digital images stored in BMP format. The most important operations that were implemented in the program are Mathematics Morphology, and from the basics (erosion and dilation) complex functions were implemented such as: Morphological filters, edge detection and granulometry. One objective of this paper is to provide a Help document to supply information on format BMP, Mathematical Morphology and implementation of these functions. In The help is also available the program source code properly commented.

*A mi Madre, este logro también es tuyo.*

*“Al igual que escalar una montaña  
el reto es personal, tu no la subes a  
ella, tú te escalas a ti mismo”.*

# Agradecimientos

Gracias a Dios por darme la fuerza para seguir adelante, por muy difícil y oscuro que el camino fue, no estaba solo, iba contigo Señor.

Gracias al profesor Levi García, tutor y amigo, fuiste el único que no me dio la espalda en el momento en que más necesitaba a alguien, confiaste en mí y me extendiste la mano para encarar juntos este proyecto, gracias por nunca rendirte y seguir hasta el final, estaré siempre agradecido contigo.

A mi querida madre muchas gracias por tu apoyo, me faltarían palabras para expresar mi inmensa gratitud para contigo, sólo humildemente quiero reconocer la fuerza y constancia que siempre has mostrado, sin ti no hubiese sido posible culminar este proyecto, tú me diste la energía para mantenerme y continuar.

Gracias a mi familia por su apoyo y respaldo incondicional, a mi Padre un luchador incansable que siempre tiene una sonrisa en los momentos de adversidad. Gracias a todos mis familiares que físicamente ya no están en este plano, honro la memoria de cada uno de ustedes. Gracias abuela, tu esfuerzo y sacrificio de años quiero recompensarlos con este trabajo.

Gracias a todos mis amigos, amigas y compañeros de la Universidad, me gustaría nombrarlos a todos pero no quisiera dejar a alguno por fuera, desde mis primeros amigos de la Universidad, compañeros de escuela, mis amigos de la orientación de Instrumentación, amigos de las otras escuelas, pasé momentos hermosos con su compañía, gracias por permitirme ser parte de sus vidas.

Un agradecimiento muy especial a mis profesores tutores y maestros, sus enseñanzas y consejos los llevo grabados en mi corazón, ustedes forjaron mi actitud y moldearon mi alma, sus palabras las recuerdo y las mantengo conmigo siempre.

Muchas gracias a mi amiga, novia y compañera Yessy Rodríguez, tú has sido muy importante en mi vida, gracias por brindarme tantas alegrías, por estar firme conmigo en las malas y por disfrutar con júbilo en las buenas.

Quiero agradecerme por ser fiel a un esfuerzo y sacrificio de años, a nunca bajar los brazos y mantenerme firme hasta el final. En la etapa más oscura y sombría de mi vida, el fuego que está dentro de mí se volvió a encender, cual Ave Fénix que renace de sus cenizas para emprender el vuelo nuevamente, con coraje y valentía.

En fin, gracias a todo aquel que de una u otra forma contribuyó para mi desarrollo y crecimiento personal, que me ayudó y esperó siempre lo mejor de mí, a todas esas personas que se preocuparon y me incentivaron, ustedes que me auparon para continuar, muchísimas gracias, saben que rendirse jamás ha estado ni estará en mis genes.

# Índice General

Introducción .....	1
Capítulo 1. Marco Teórico .....	3
1.1 Imágenes .....	3
1.1.1 Imágenes en Color .....	3
1.1.2 Imágenes en Escala de Grises .....	5
1.1.3 Imágenes Binarias.....	7
1.2 Formato de Imágenes: BMP.....	8
1.3 Brillo y Contraste.....	12
1.3.1 Brillo .....	12
1.3.2 Contraste .....	13
1.4 Morfología Matemática.....	14
1.5 El Elemento Estructurante.....	15
1.6 Vecindad entre Píxeles.....	17
1.7 Operadores Morfológicos.....	19
1.7.1 Erosión .....	20
1.7.1.1 Erosión en Imágenes Binarias .....	20
1.7.1.2 Erosión en Imágenes en Escala de grises.....	23
1.7.2 Dilatación .....	26
1.7.2.1 Dilatación en Imágenes Binarias .....	26
1.7.2.2 Dilatación en Imágenes en Escala de grises.....	30
1.7.3 Dualidad de la Erosión y Dilatación .....	32
1.7.4 Apertura y Cierre .....	34
1.7.4.1 Apertura .....	35
1.7.4.2 Cierre .....	36
1.8 Aplicaciones .....	38
1.8.1 Filtros Morfológicos .....	38
1.8.2 Extracción de bordes.....	40
1.8.2.1 Borde Interno.....	40
1.8.2.2 Borde Externo .....	41
1.8.2.3 Borde Total.....	42
1.8.3 Granulometría.....	43
1.8.4 Histograma .....	45
Capítulo 2. Marco Metodológico .....	46
2.1 Interfaz Gráfica de PDI.....	46
Archivo .....	47
Ajustes .....	47
Convertir.....	47
Op. Morfológicas .....	48
Aplicaciones .....	48
Ayuda .....	48
2.2 Visualización de imágenes digitales BMP .....	49
2.2.1 Comprobaciones .....	50
2.2.2 Carga de la Imagen en memoria.....	54
2.2.3 Mostrar la Imagen.....	59
2.3 Almacenamiento de la Imagen .....	60

2.4	Propiedades .....	61
2.5	Salir del Programa .....	62
2.6	Ajuste de la Imagen .....	63
2.6.1	Brillo de la Imagen .....	63
2.6.2	Contraste de la Imagen.....	63
2.7	Convertir Imagen .....	64
2.7.1	Imagen: Color a Escala Grises. ....	64
2.7.2	Imagen: Escala de Grises a Blanco y Negro .....	64
2.7.3	Imagen: Invertir Blanco y Negro.....	64
2.8	Operaciones Morfológicas .....	65
2.8.1	Erosión .....	65
2.8.2	Dilatación .....	70
2.9	Sistema de Ayuda .....	75
2.9.1	Ayuda Teórica .....	75
2.9.2	Ayuda Sobre el Programa .....	75
2.9.3	Código fuente .....	76
	Resultados y Conclusiones .....	77
	Bibliografía.....	78

# Índice de Figuras

Figura 1.1 Imagen de Lena en Color.-----	5
Figura 1.2 Imagen en escala de grises, sección, y conjunto de píxeles. -----	6
Figura 1.3 Imagen de Lena en escala de grises. -----	7
Figura 1.4 Conjunto de píxeles de una Imagen Binaria (a). Imagen Binaria (b). -----	8
Figura 1.5 Imagen de Lena en blanco y negro obtenida utilizando un TH de 127. -----	8
Figura 1.6 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises con el aumento del 50% de la escala máxima de brillo (b). -----	12
Figura 1.7 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises con el aumento del 50% de la escala máxima de contraste (b). -----	13
Figura 1.8 Elementos Estructurantes. Cuadrícula 3x3 (a). Cruz 3x3 (b). Equis 3x3 (c). Línea Diagonal Negativa 3x3 (d). Línea Diagonal Positiva 3x3 (e). Triangular 2x3 (f). Línea Horizontal 1x3 (g). Línea Vertical 3x1 (h). Círculo 5x5 (i).-----	16
Figura 1.9 Ejemplo de EE de volumen -----	17
Figura 1.10 Adyacencias de Píxeles. Adyacencia por Frontera (a). Adyacencia por Esquinas (b).18	
Figura 1.11 Vecindades del píxel “p”. Vecindad $N_4(p)$ (a). Vecindad $N_D(p)$ (b). Vecindad $N_8(p)$ (c). -----	19
Figura 1.12 Imagen original (a). Imagen erosionada por un elemento estructurante 3x1(b).-----	21
Figura 1.13 Conjunto de píxeles de una Imagen Binaria (a). Erosión del conjunto de píxeles de una Imagen Binaria por un EE 1x2 (b)-----	21
Figura 1.14 Imagen original (a). Imagen dilatada con un EE 1x3 (b). Imagen dilatada con un EE 3x1 (c). -----	22
Figura 1.15 EE en forma de cruz 3x3-----	24
Figura 1.16 Conjunto de píxeles de una Imagen en escala de grises (a). Erosión del conjunto de píxeles de una Imagen escala de grises por un EE en forma de cruz 3x3 (b)-----	24
Figura 1.17 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises erosionada con el EE de la Figura 1.15 (b)-----	25
Figura 1.18 Imagen original (a). Imagen original dilatada por un EE 3x1 (b).-----	27
Figura 1.19 EE 1x2 (a). EE 1x2 reflejado (b)-----	28
Figura 1.20 Conjunto de píxeles de una Imagen Binaria (a). Dilatación del conjunto de píxeles de una Imagen Binaria por un EE 1x2 (b). -----	28
Figura 1.21 Imagen original (a). Imagen dilatada con un EE 1x3 (b). Imagen dilatada con un EE 3x1 (c). -----	29
Figura 1.22 Conjunto de píxeles de una Imagen en escala de grises (a). Dilatación del conjunto de píxeles de una Imagen en escala de grises por el EE de la Figura 1.15 (b). -----	30
Figura 1.23 Imagen de Lena en escala de grises (a). Imagen dilatada de Lena en escala de grises por el EE de la Figura 1.15 (b).-----	31
Figura 1.24 Imagen erosionada de Lena con el EE de la Figura 1.15 (a). Imagen complemento de la Figura anterior (b). -----	33
Figura 1.25 Imagen complemento de Lena (a). Imagen complemento de Lena erosionada con el EE reflejado de la Figura 1.15 (b).-----	33
Figura 1.26 Imagen erosionada de Lena con el EE de la Figura 1.15 (a). Imagen complemento de la Figura anterior (b). -----	34
Figura 1.27 Imagen complemento de Lena (a). Imagen complemento de Lena erosionada con el EE reflejado de la Figura 1.15 (b).-----	34
Figura 1.28 Imagen de Lena en blanco y negro (a). Apertura de la imagen anterior con el EE de la Figura 1.15 (b). -----	35
Figura 1.29 Imagen de Lena en escala de grises (a). Apertura de la imagen anterior con el EE de la Figura 1.15 (b). -----	36
Figura 1.30 Imagen de Lena en blanco y negro (a). Cierre de la imagen anterior con el EE de la Figura 1.15 (b). -----	37
Figura 1.31 Imagen de Lena en escala de grises (a). Cierre de la imagen anterior con el EE de la Figura 1.15 (b). -----	37
Figura 1.32 Imagen original (a). Imagen con ruido (b). Imagen de filtrada (c).-----	39
Figura 1.33 Imagen original (a). Imagen con ruido (b). Imagen de filtrada (c).-----	40

Figura 1.34 Imagen blanco y negro de Lena (a). Borde interno de la imagen anterior con el EE de la Figura 1.15 (b). -----	41
Figura 1.35 Imagen blanco y negro de Lena (a). Borde externo de la imagen anterior con el EE de la Figura 1.15 (b). -----	42
Figura 1.36 Imagen blanco y negro de Lena (a). Borde total de la imagen anterior con el EE de la Figura 1.15 (b). -----	42
Figura 1.37 Imagen original (a). Imágenes de las múltiples apertura (b)–(l). -----	44
Figura 1.38 Histograma de la Figura 1.38 (a)-----	45
Figura 2.1 Menú del programa PDI-----	47
Figura 2.2 Diagrama de Flujo: Visualización de imágenes digitales BMP -----	49
Figura 2.3 Imagen 3x13 (a). Ampliación de la Imagen 3x13 (b) -----	51
Figura 2.4 Propiedades de la Imagen 3x13 -----	52
Figura 2.5 Diagrama de Flujo: Guardar Imagen-----	61
Figura 2.6 Diagrama de Flujo: Propiedades -----	62

## Índice de Tablas

Tabla 1 Información contenida en una archivo BMP.....	11
Tabla 2 Relación entre un pixel de la Imagen y un punto del EE para determinar la erosión binaria. ....	66
Tabla 3 Relación entre un pixel de la Imagen y un punto del EE para determinar la erosión binaria. ....	71

# Introducción

El procesamiento digital de imágenes es una herramienta computacional, que emplea un conjunto de técnicas para acondicionar y extraer información de imágenes en formato digital. Son muchas las áreas en las que su aplicación resulta de gran interés, entre ellas destacan: microscopía, metalurgia, meteorología, geología, robótica, tecnología militar y muy especialmente en técnicas modernas de diagnóstico por imagen, tales como la tomografía axial computarizada y la resonancia magnética nuclear.

En la Escuela de Física de la U.C.V, existe una orientación llamada Instrumentación, allí se dicta un curso denominado Instrumentación I, el cual contiene tres temas principales: control, señales e imágenes. Un objetivo fundamental de esta materia, es enseñarles a los estudiantes del curso a programar, haciendo uso del programa Visual Basic. Esos conocimientos luego se ponen a prueba en la parte del curso relacionada con imágenes, donde los estudiantes deben desarrollar operaciones básicas de la morfología matemática y sus aplicaciones.

En este trabajo se desarrolló un software que permite procesar imágenes digitales, empleando una técnica no lineal denominada morfología matemática<sup>1</sup>, permitiendo almacenar y recuperar imágenes digitales siguiendo el formato BMP<sup>2</sup>. El proyecto reviste de importancia y surge como una necesidad de apoyar la docencia dictada en el curso de Instrumentación I.

---

<sup>1</sup> La Morfología matemática es una técnica empleada para el análisis y tratamiento de las estructuras geométricas, basada en la teoría de conjuntos.

<sup>2</sup> Los archivos con extensión .BMP, en los sistemas operativos Windows, representan la sigla BitMaP (o también Bit Mapped Picture), es decir, mapa de bits. Los archivos de mapas de bits se componen de direcciones asociadas a códigos de color, uno para cada cuadro en una matriz de píxeles.

# Objetivos

## Objetivo General

Desarrollar un Software de Procesamiento Digital de Imágenes Basado en Morfología Matemática, para Aplicaciones Docentes en el Área de Instrumentación.

## Objetivos Específicos

- Desarrollar un software de Procesamiento Digital de Imágenes (PDI) basado en Morfología Matemática.
- Incorporar en el software de PDI, un documento de ayuda que brinde información de la teoría de Morfología Matemática y el formato de imágenes BMP y su implementación a nivel de programación.
- Proporcionar el código fuente del software de PDI, debidamente comentado.

# Capítulo 1. Marco Teórico

## 1.1 Imágenes

Una imagen es una representación visual bidimensional que manifiesta la apariencia de un conjunto de objetos reales que conforman una escena tridimensional. La discretización de una imagen analógica produce una imagen digital, éste proceso se realiza de forma espacial, sobre las dimensiones de la imagen y en la cromaticidad, característica que envuelve el matiz y la saturación del color. Luego de esa operación, la imagen se puede almacenar en un medio digital, utilizando algún formato como: BMP, JPG, GIF, PNG, etc.

Cuando hablamos de imágenes digitales surge un importante elemento que denominaremos “píxel”, que es un término que proviene de la expresión "Picture Element". “Un píxel es un punto de luz en un monitor o pantalla, representa la unidad más pequeña de medida gráfica y con un conglomerado de ellos se construye la imagen digital” [Garat, Fotografía Digital].

Se puede imaginar cada píxel en un monitor como la superposición de tres puntos luminosos, que corresponden a los colores primarios: rojo, verde y azul; y cada uno de ellos posee 256 niveles de intensidad.

### 1.1.1 Imágenes en Color

Las imágenes a color poseen una tonalidad cromática que varían desde 16 colores para imágenes de 4 bits por píxel, hasta 16.777.216 colores para imágenes de 24 bits por píxel.

Aunque existen imágenes de 32 bits, éstas no proporcionan colores adicionales con respecto a las de 24 bits, los 8 bits adicionales suministran información sobre la transparencia de la imagen, éste byte adicional suele ser llamado: máscara o canal alfa y es independiente del color. Cada píxel de la imagen es representado con 256 niveles de transparencia posible. Un valor de “0” en la máscara indica que ese píxel es completamente transparente (deja ver el fondo), mientras que, un valor de “255” indica que el píxel será totalmente opaco (no deja ver el fondo), los valores intermedios proporcionan transparencias parciales.

Los modelos cromáticos o espacios de color, codifican las sensaciones cromáticas en vectores tridimensionales que representan coordenadas espaciales [Ortiz, 2002]. Entre los modelos cromáticos más utilizados están: RGB, CYM, HSI, etc. La mayoría de éstos, se derivan de transformaciones directas del espacio RGB. Dependiendo de los requerimientos y la finalidad de su implementación, un modelo podría ser más idóneo que otro.

## **Modelo RGB**

El modelo RGB es uno de los modelos cromáticos más empleados, y es el que se utiliza en el presente trabajo. Las siglas RGB provienen del idioma inglés Red, Green, Blue, y se fundamenta en el modelo de triestímulos y en la síntesis aditiva del color. “La Síntesis aditiva del color consiste en mezclar los colores primarios (verde, rojo, azul) de tal manera que el color resultante es más luminoso que los empleados para dicha formación” [Ortiz, 2002].

En el ojo humano existen dos tipos de células fotorreceptoras: los bastones y los conos, cada una cumple con un papel específico en el proceso de la visión. Los bastones son responsables de la visión en condiciones de poca luminosidad, son muy sensibles a la luz, se saturan en condiciones de mucha iluminación y no detectan los colores. Por su parte los conos, son sensibles a la luz para frecuencias que se

corresponden con los colores rojo, verde y azul, se asume entonces que tres imágenes monocromáticas (verde, roja y azul) son formadas por los conos, para luego ser transmitidas al cerebro en donde se forma una imagen cromática, con la información aportada de cada monocroma.

A continuación se muestra la imagen de Lena<sup>3</sup> en color, ésta imagen servirá de ejemplo para mostrar otros tipos de imágenes que se utilizan (binarias y escala de grises) así como los efectos que el procesamiento digital puede producir sobre las mismas.



Figura 1.1 Imagen de Lena en Color.

### 1.1.2 Imágenes en Escala de Grises

Las imágenes en escala de grises pueden verse como un caso particular de las imágenes a color. Supongamos que tenemos los valores de los colores rojo, verde, azul de un píxel, lo más probable es que estos tres valores sean diferentes. Si se promedia el valor de los tres colores, y se asigna al píxel en cuestión, el proceso dará

---

<sup>3</sup> Lena es un fragmento de fotografía de una playmate reproducida en el número de Noviembre (miss Noviembre) de 1972 de la revista Playboy. Sirve como imagen de prueba para los algoritmos de compresión de imagen y se ha convertido de facto en un estándar industrial y científico. La imagen es tan conocida que Lena Soderberg (de soltera Sjööblom), una sueca, fue la invitada de honor de la 50ª Conferencia Anual de la Society for Imaging Science in Technology, en 1997[Wikipedia, Lena].

como consecuencia un píxel de color gris. Si se repite la operación para cada uno de los píxeles de la imagen, el resultado será una imagen en escala de grises.

A continuación en la Figura 1.2, se muestra un ejemplo de una imagen en escala de grises, de la cual se extrae una pequeña sección de interés, y se muestran los valores de algunos píxeles presentes de cada píxel presente en esa sección. Los valores de luminosidad están entre 0 y 255. Si un píxel es oscuro su valor será cercano a “0”, si por lo contrario es claro su valor será cercano a “255”.

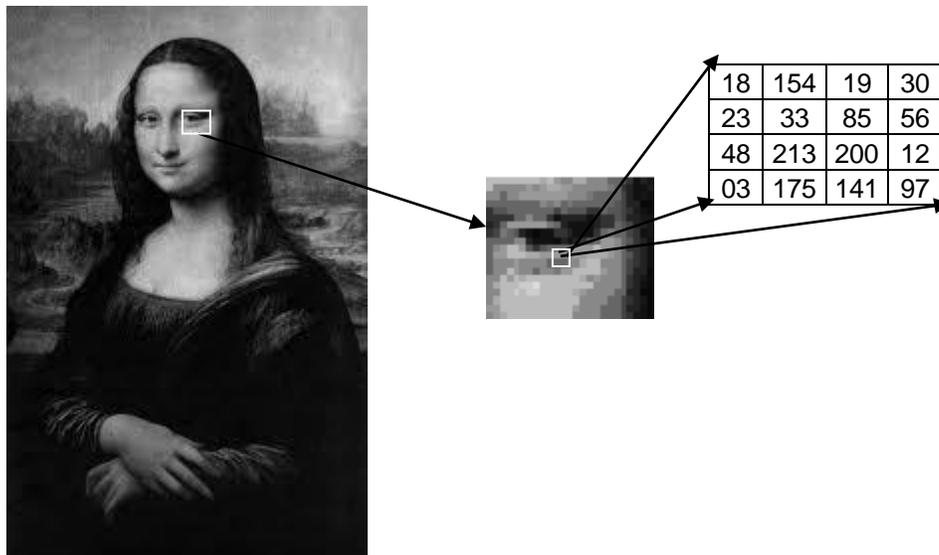


Figura 1.2 Imagen en escala de grises, sección, y conjunto de píxeles.

A continuación en la Figura 1.3 se muestra la imagen de Lena en escala de grises, ésta fue creada a partir de la Figura 1.1, donde se promediaron los planos: rojo, verde y azul.

A partir de la imagen de la Figura 1.3, todas las imágenes que se muestran en el presente trabajo, fueron generadas usando el Programa de Procesamiento Digital de Imágenes PDI.



Figura 1.3 Imagen de Lena en escala de grises.

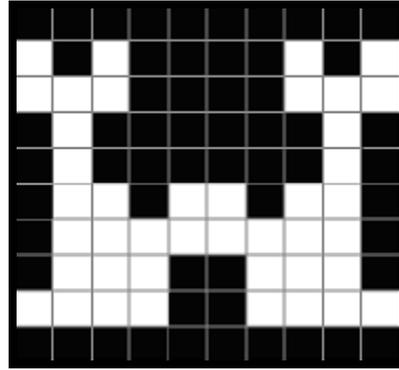
### **1.1.3 Imágenes Binarias**

Las imágenes binarias son aquellas dispuestas de tal manera que, los valores posibles de luminosidad para cada píxel están restringidos a los extremos: negro ó blanco, es decir, “0” ó “255” respectivamente. Por convención, denotaremos con “0” a los puntos del fondo de la imagen, y el valor de “1” estará asignado a los puntos que conforman la imagen. Si se tiene una imagen en escala de grises y se desea obtener una imagen en blanco y negro a partir de ésta, es necesario definir un valor de referencia llamado: umbral (en inglés: Threshold “TH”), si el píxel (que tiene valores entre 0-255) es igual o está por encima del valor asignado en el TH, este píxel será puesto en “255”, si por el contrario es menor a ese valor se le asigna “0”.

A continuación en la Figura 1.4 y Figura 1.5 se muestran ejemplos de imágenes binarias.

0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	1
1	1	1	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0
0	1	1	0	1	1	0	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	1	1	1	0
1	1	1	1	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0

(a)



(b)

Figura 1.4 Conjunto de píxeles de una Imagen Binaria (a). Imagen Binaria (b).



Figura 1.5 Imagen de Lena en blanco y negro obtenida utilizando un TH de 127.

## 1.2 Formato de Imágenes: BMP

El formato de almacenamiento de imágenes digitales BMP fue desarrollado por las compañías Microsoft e IBM para ser utilizado en los sistemas operativos Windows y OS/2. El BMP o **bitmap** (mapa de Bits) es una estructura que almacena de forma organizada una correspondencia entre cada píxel de la imagen digital y su color. En algunos casos se emplea una paleta, la cual posee un número determinado de colores, a los que se accede a través de un índice.

Cuando se emplea la paleta, la información de los colores está contenida en una matriz auxiliar, ésta paleta posee un número limitado de combinaciones de colores (en general, menos de 65.000 colores). En este caso, la imagen digital posee un solo plano, y el valor de cada píxel hace referencia de forma indirecta a su color, es decir, el valor es empleado como un índice, que apunta hacia un color específico definido en la paleta de colores.

## **Estructura del Formato de Imágenes: BMP**

Un archivo BMP almacena toda la información necesaria para poder recuperar una imagen digital, ésta información está dividida en tres bloques:

- Encabezado.
- Paleta (opcional).
- Información de la imagen.

Windows utiliza el BMP para almacenar imágenes como mapa de bits. La información de cada línea horizontal de la imagen se guarda como múltiplos de 4 bytes. Si el ancho de la imagen no cumple con esta condición, se completa con los bits o bytes que sean necesarios. La operación se realiza asignándole el valor de cero “0” a cada bit o byte añadido, éste detalle es importante cuando se lee el archivo, ya que se debe considerar para discriminar la información que pertenece a la imagen.

Otro aspecto importante, es que la información en un BMP se guarda de izquierda a derecha y de abajo hacia arriba, así que la primera línea que se lee del archivo realmente es la información de la última línea de la imagen.

A continuación se presenta una tabla que muestra la estructura de un BMP. Cada campo está identificado con su nombre, acompañado de su dirección de registro, tamaño y una breve descripción de su contenido.

<b>Dir</b>	<b>Campo</b>	<b>Tamaño</b>	<b>Contenido</b>
0	Identificador	2 bytes	<p>Caracteres que identifican a un mapa de bit. Los siguientes entradas son válidas</p> <p>"BM" → Windows 3.1x, 95, NT  "BA" → OS/2 Bitmap Array  "CI" → OS/2 Color Icon  "CP" → OS/2 Color Pointer  "IC" → OS/2 Icon  "PT" → OS/2 Pointer</p>
2	Tamaño del Archivo	4 bytes	Longitud completa del archivo en bytes
6	Reservado	4 bytes	Reservado para uso posterior
10	Desplazamiento de Datos del Mapa de Bits	4 bytes	Desplazamiento desde el principio del archivo hasta el comienzo de los datos del mapa de bits
14	Tamaño del encabezado del Mapa de Bits	4 bytes	<p>Longitud del encabezado del mapa de bits. Los siguientes tamaños son posibles:</p> <p>40 → Windows  12 → OS/2 1.x  240 → OS/2 2.x</p>
18	Ancho	4 bytes	Longitud horizontal del mapa de bits en píxeles
22	Altura	4 bytes	Longitud vertical del mapa de bits en píxeles
26	Planos	2 bytes	Numero de Planos del mapa de bits
28	Bit por Píxel	2 bytes	<p>Bits por píxel utilizado para almacenar información de la paleta de entrada. También se identifica de manera indirecta el número de posibles colores.</p> <p>1 → mapa de bits monocromático  4 → 16 colores de mapa de bits  8 → 256 colores de mapa de bits  16 → 16 bits (color de alta densidad) de mapa de bits  24 → 24 bits (color verdadero) de mapa de bits  32 → 32 bits (color verdadero) de mapa de bits</p>

30	Compresión	4 bytes	<p>Especificaciones de compresión. Los siguientes valores son posible:</p> <p>0 → ninguno (también identificados por BI_RGB)</p> <p>1 → RLE de 8 bits / píxel (también identificados por BI_RLE8)</p> <p>2 → RLE de 4 bits / píxel (también identificados por BI_RLE4)</p> <p>3 → Campos de bits (también identificados por BI_BITFIELDS)</p>
34	Tamaño de los Datos Del Mapa de Bits	4 bytes	Tamaño en bytes del conjunto de datos del mapa de Bits, éste número debe ser redondeado hasta completar múltiplos de 4 Bytes
38	Resolución H	4 bytes	Resolución horizontal expresada en píxeles por metro
42	Resolución V	4 bytes	Resolución vertical expresada en píxeles por metro
46	Colores	4 bytes	Número de colores utilizados en el mapa de bits.
50	Colores Importantes	4 bytes	Cantidad de colores importantes. Este número será igual a el número de colores cuando todos los colores sean importantes.
54	Paleta	N * 4 bytes	<p>La especificación de la paleta. Por cada entrada en la paleta cuatro bytes se utilizan para describir los valores RGB del color de la siguiente manera:</p> <p>1 byte para el componente azul</p> <p>1 byte para el componente verde</p> <p>1 byte para el componente rojo</p> <p>1 byte de relleno que se establece en 0 (cero)</p>
1078	Datos del Mapa Bits	X bytes	Dependiendo de las especificaciones de compresión, este campo contiene todos los datos del mapa de bits, que representan índices en la paleta de colores.

Tabla 1 Información contenida en una archivo BMP

## 1.3 Brillo y Contraste

El brillo y el contraste son propiedades características de las imágenes. Variando apropiadamente los niveles de éstos atributos, se puede mejorar considerablemente la apariencia de una imagen.

### 1.3.1 Brillo

El brillo es la cantidad de energía luminosa emitida o reflejada por un objeto o imagen. “En términos cualitativos, puede decirse que el brillo de una imagen debe ser suficiente como para que ésta pueda apreciarse satisfactoriamente en un recinto moderadamente iluminado y percibir sin esfuerzo toda la información visual”. [Pérez Vega, 2012]. Para aumentar o disminuir el brillo de una imagen, es necesario sumar o restar una constante, al valor de cada uno de sus píxeles.



Figura 1.6 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises con el aumento del 50% de la escala máxima de brillo (b).

La imagen de la Figura 1.6 (a) posee el lado derecho más claro que el izquierdo. Al aumentar el brillo al 50 %, en la Figura 1.6 (b) se aprecia como el lado izquierdo de la imagen luce más iluminado, en cambio, la zona derecha de la imagen

se observa “saturada” de brillo. El brillo es el grado de luz que tiene una imagen, y está muy relacionado con el contraste. Las imágenes en blanco y negro presentan brillo y contraste máximos.

### 1.3.2 Contraste

El contraste es la diferencia relativa de intensidad entre un punto de una imagen y sus alrededores, es decir, un aumento de contraste incrementa la diferencia de luminosidad entre áreas más oscuras y más claras de una imagen. Para variar el contraste de una imagen, primero, es necesario restar el promedio de intensidad de la imagen a cada uno de sus píxeles, luego se multiplica por un factor o una constante, y para finalizar se suma el promedio sustraído. Este proceso garantiza la variación del contraste, sin afectar el brillo de la imagen. Si el valor de la constante es mayor a uno “1” el contraste aumenta, en caso contrario se atenúa.



Figura 1.7 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises con el aumento del 50% de la escala máxima de contraste (b).

En la Figura 1.7 (b) se aprecia notablemente el efecto que produce el aumento de contraste de la imagen de Lena, las zonas más oscuras de la imagen, se aprecian mucho más oscurecidas que antes, y las zonas más claras, se han aclarado aún más.

## 1.4 Morfología Matemática

El término PDI proviene del idioma inglés (Digital Image Processing) y comúnmente es traducido como: procesamiento digital de imágenes, y aunque no es la mejor traducción, ésta se ha aceptado como tal. Una traducción que concuerda correctamente con dicho término es: Procesamiento de imágenes digitales, ya que el procesamiento es sobre imágenes digitales. No obstante, es evidente que este procesamiento se efectúa en un sistema digital. Para realizar el PDI se cuenta entre otras con una importante herramienta denominada morfología matemática, y se emplea cuando las formas en las imágenes contienen la información de relevancia.

La morfología matemática se utiliza para cumplir múltiples aspectos entre los cuales se encuentra:

- a) **Pre y post procesamiento de imágenes.** (Ej.: supresión de ruido, simplificación de formas).
- b) **Destacar la estructura de los objetos.** (Ej.: extraer el esqueleto, detección de objetos, envolvente convexa, ampliación, reducción).
- c) **Descripción de objetos.** (Ej.: área, perímetro, y filtrado morfológico).

La morfología matemática está basada en la teoría de conjuntos. Una imagen digital puede ser tratada como un conjunto, y los píxeles que la conforman como los elementos de dicho conjunto. Valiéndonos de esto, cualquier transformación sobre un conjunto puede ser aplicada a las imágenes digitales.

Para imágenes binarias, los conjuntos pertenecen al espacio bidimensional  $Z^2$ , donde cada elemento del conjunto posee dos coordenadas  $(x, y)$ .

Las imágenes digitales en escala de grises pueden ser representadas como conjuntos en las cuales sus elementos pertenecen al espacio  $Z^3$ , donde dos componentes de cada elemento hacen referencia a las coordenadas de cada píxel, y la tercera concierne a su valor de intensidad.

El objetivo de realizar transformaciones morfológicas es reconocer ciertas estructuras geométricas en las imágenes, este proceso es posible mediante operaciones con otro conjunto conocido como elemento estructurante.

## **1.5 El Elemento Estructurante**

El Elemento Estructurante o elemento de estructura (EE) es un “patrón” predefinido, que se aplica sobre las imágenes con la finalidad de reconocer, resaltar, extraer, etc. estructuras geométricas presentes en ella. El tamaño y la forma del elemento estructurante se deben elegir de acuerdo a la morfología de la imagen sobre la cual va a actuar.

“Para emplear un elemento estructurante, es necesario definir su origen, el origen del EE es un concepto muy importante, ya que define las orientaciones de las traslaciones” [Ortiz, 2002]. Generalmente se elige el origen del EE de tal manera que coincida con su centro geométrico, aunque no siempre se puede o se quiere cumplir con esta condición. El EE opera ubicando su origen en un píxel de la imagen, y a partir de allí se estudia la influencia en la vecindad o conectividad del píxel en cuestión, éste proceso se realiza secuencialmente sobre cada píxel de la imagen.

A continuación se muestran algunas formas de elemento estructurante. Ver Figura 1.8

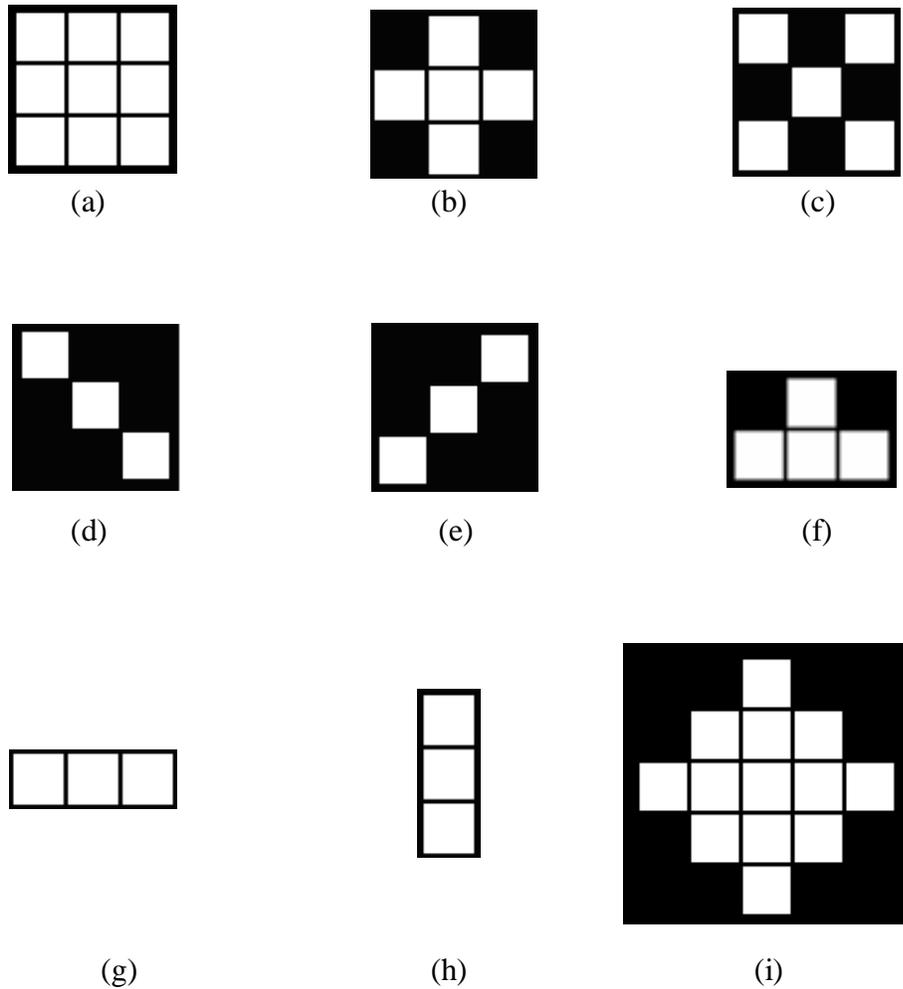


Figura 1.8 Elementos Estructurantes. Cuadrícula 3x3 (a). Cruz 3x3 (b). Equis 3x3 (c). Línea Diagonal Negativa 3x3 (d). Línea Diagonal Positiva 3x3 (e). Triangular 2x3 (f). Línea Horizontal 1x3 (g). Línea Vertical 3x1 (h). Círculo 5x5 (i).

Para realizar operaciones morfológicas, se debe desplazar el EE sobre cada pixel dentro de los límites de la imagen. No siempre será posible colocar el origen del EE en los píxeles que se encuentren en los bordes de la imagen, ya que el EE se excede de la misma, debido que algunos de sus elementos no están contenidos dentro de la imagen. Efectuar este proceso daría como resultado una reducción del tamaño de la imagen, y si se aplica de manera sucesiva, terminaría destruyéndola completamente.

Para preservar la imagen de ser destruida, se realiza una excepción que sólo afecta a los píxeles que se encuentran en los bordes de la imagen. Si el EE excede los límites de la imagen, los elementos que estén fuera de ella son ignorados, de ésta manera, el origen del EE puede recorrer cada uno de los píxeles de la imagen.

Hasta ahora los Elementos Estructurantes que se han mencionado han sido planos, es decir, son considerados solamente aquellos elementos del EE con valor igual a uno. En escala de grises el EE no tiene por qué ser plano, cada elemento del EE podría estar entre “1” y “255”, por lo tanto si un valor es igual a cero se considera que no forma parte del EE. En consecuencia, si un EE presenta dicha condición se le considera un EE de volumen, recordamos que para imágenes binarias no existe EE de volumen. Según [Ortiz, 2002] los EE de volumen modifican el efecto de las operaciones morfológicas, respecto a un EE plano.

En la Figura 1.9 se muestra un ejemplo de un EE de volumen, el elemento superior derecho y el inferior izquierdo no forman parte del EE, ya que sus valores son iguales a cero.

25	25	0
25	1	1
0	1	3

Figura 1.9 Ejemplo de EE de volumen

## 1.6 Vecindad entre Píxeles

Dos píxeles son adyacentes si, y sólo si, tienen en común algún punto de sus fronteras. En la Figura 1.10 se muestran los dos tipos de adyacencia, adyacencia por frontera y adyacencia por esquinas.



Figura 1.10 Adyacencias de Píxeles. Adyacencia por Frontera (a). Adyacencia por Esquinas (b).

Los vecinos de un píxel “p” situado en coordenadas del sistema  $(x, y)$  son todos aquellos píxeles que rodean directamente al píxel “p”. El conjunto de 4 píxeles que limitan con el píxel “p” (2 vecinos horizontales y 2 vecinos verticales) recibe el nombre de vecindad de 4, en notación  $N_4(p)$  y están localizados de la siguiente manera:

$$(x + 1, y); (x - 1, y); (x, y + 1); (x, y - 1).$$

Si consideramos sólo los píxeles localizados diagonalmente al píxel “p”, estamos en presencia de una vecindad de diagonal, en notación  $N_D(p)$ . Las coordenados de esos píxeles son:

$$(x + 1, y + 1); (x + 1, y - 1); (x - 1, y + 1); (x - 1, y - 1).$$

Dos píxeles son vecinos siempre y cuando sean adyacentes, si los píxeles comparten una de sus fronteras, se denominan vecinos directos, si sólo colindan en una de sus esquinas son llamados vecinos indirectos. El conjunto de píxeles que forman una Vecindad  $N_4(p)$  más el conjunto de píxeles que forman una vecindad  $N_D(p)$ , es decir, vecinos directos e indirectos del píxel “p” se consideran la vecindad  $N_8(p)$ . En la Figura 1.11 se ilustra las vecindades del píxel “p”

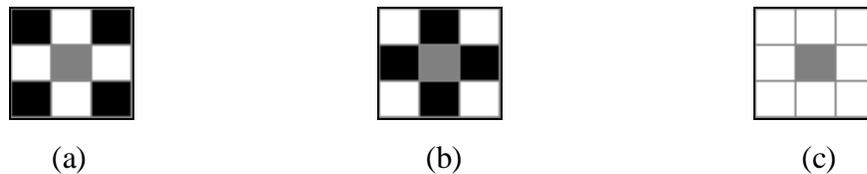


Figura 1.11 Vecindades del píxel “p”. Vecindad  $N_4(p)$  (a). Vecindad  $N_D(p)$  (b). Vecindad  $N_8(p)$  (c).

## 1.7 Operadores Morfológicos

La morfología matemática es una técnica de procesamiento no lineal, basada en teoría de conjuntos y con un marcado interés en la geometría de los objetos. Los operadores morfológicos elementales son la erosión y la dilatación, y están estrechamente relacionados con la resta y la suma de Minkowsky<sup>4</sup> respectivamente. A partir de los operadores morfológicos elementales, se pueden elaborar otros operadores como aplicación en secuencia de éstos. Existe un tercer operador llamado “Hit-Miss”, que no es posible crearlo a partir de la erosión y la dilatación, así que éste podría ser considerado también como un operador fundamental.

Realizar en secuencia las operaciones de erosión y dilatación da como resultado un operador llamado Apertura, en cambio realizar las operaciones de dilatación y erosión generan un operador denominado Cierre. La implementación sucesiva de operadores generan aplicaciones como: Filtros Morfológicos, Extracción de Bordes, etc. Dependiendo si las imágenes digitales son binarias, en escala de grises ó color, surgen cambios en el mecanismo para aplicar las operaciones morfológicas. Tanto en la erosión como la dilatación, el tamaño y forma finales del conjunto resultante, dependerá fuertemente del tamaño y forma del elemento de estructura.

---

<sup>4</sup> La descripción básica de la Morfología Matemática descansa en la “teoría de conjuntos” cuyos primeros trabajos se deben a Minkowsky y Hadwiger. La continuación de la investigación bajo la impulsión y reformulación de Matheron y Serra, se darían posteriormente a conocer bajo la denominación de Morfología Matemática, como una técnica no lineal de tratamiento de señales [Ortiz, 2002].

## 1.7.1 Erosión

La operación de erosión consiste en hacer decrecer un conjunto A (la imagen) a través de un proceso controlado de eliminación de elementos, tomando como referencia un elemento de estructura B [Gamino, 2004].

### 1.7.1.1 Erosión en Imágenes Binarias

La operación de erosión se define con la siguiente ecuación:

$$A \ominus B = \{x \mid B_x \subseteq A\} \quad (1)$$

La ecuación (1) indica que la erosión de A por B, es el conjunto de todos los puntos x tales que B, trasladado por x, está contenido en A.

Para imágenes binarias, la operación de erosión de un conjunto A por un elemento estructurante B, se define como el conjunto de puntos x, pertenecientes a A, de tal manera que, cuando el elemento estructurante B se traslada a un punto x, el EE queda incluido en A.

El efecto de la erosión puede observarse en la imagen de la Figura 1.12 (b), donde un EE 3x1 en forma de línea vertical, como el mostrado en la Figura 1.8 (h) desaparece los píxeles de la imagen donde no está presente el elemento estructurante.

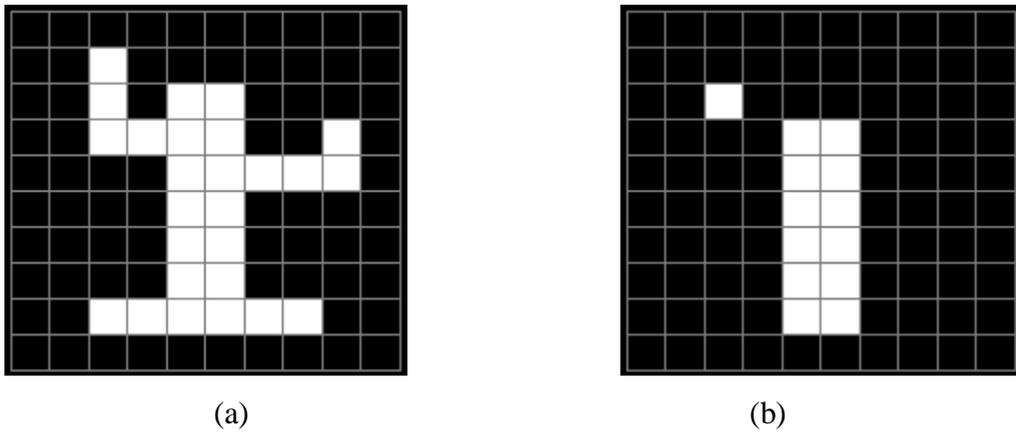


Figura 1.12 Imagen original (a). Imagen erosionada por un elemento estructurante 3x1(b).

A continuación se presenta un ejemplo de erosión binaria, representado por un recuadro que muestra el valor de los píxeles de una imagen, otro recuadro que representa los píxeles de la imagen erosionada, y el EE con forma de línea horizontal de 1 fila y 2 columnas (1x2), con origen en el punto de la izquierda del EE. Si todos los puntos del EE están contenidos en la imagen, el píxel donde se centró el EE será colocado en uno (1) en la imagen erosionada.

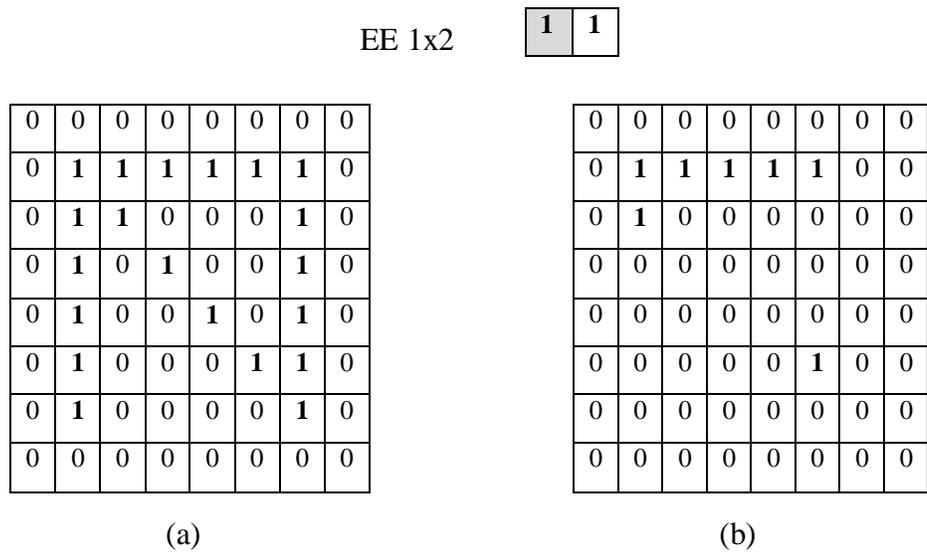


Figura 1.13 Conjunto de píxeles de una Imagen Binaria (a). Erosión del conjunto de píxeles de una Imagen Binaria por un EE 1x2 (b)

A continuación la imagen de la Figura 1.14 (a) se muestra como ejemplo de erosión binaria, ella está compuesta por líneas horizontales y verticales y fue erosionada por dos Elementos Estructurantes, uno de forma línea horizontal y otro de forma línea vertical. Los resultados pueden apreciarse en las imágenes de la Figura 1.14 (b) y Figura 1.14 (c) respectivamente. La imagen original fue erosionada con el EE mostrado en la Figura 1.8 (g) y el resultado es una imagen donde sólo están presentes líneas horizontales, y éstas a su vez han decrecido ligeramente en su longitud, (Ver Figura 1.14 (b)). En cambio, si se emplea el EE mostrado en la Figura 1.8 (h) la imagen resultado será una en la que los objetos que están presentes sólo son de forma vertical y éstos también han decrecido ligeramente en su longitud (Ver Figura 1.21 (c)). Si bien la erosión decrece el tamaño de los objetos, podemos constatar que más allá de la operación, el EE determina fuertemente en qué dirección se produce dicho decrecimiento.

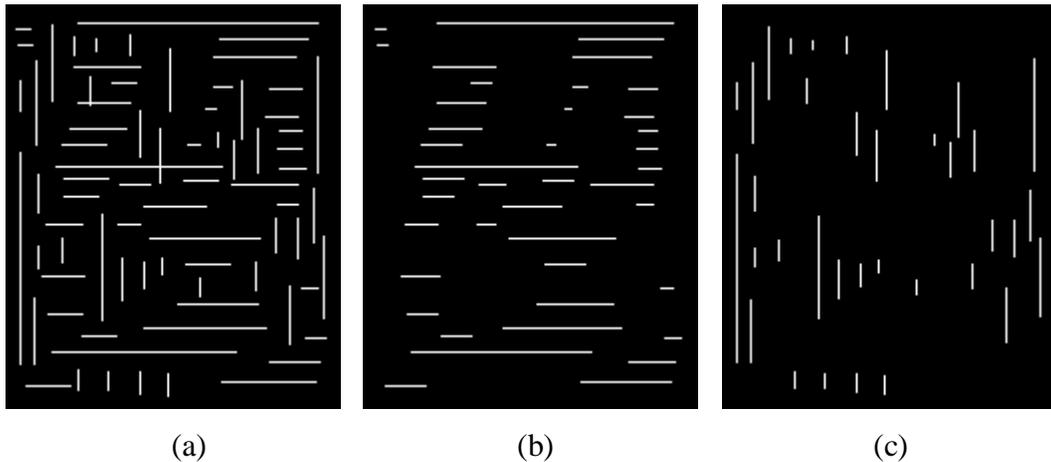


Figura 1.14 Imagen original (a). Imagen dilatada con un EE 1x3 (b). Imagen dilatada con un EE 3x1 (c).

Cuando ampliamos el concepto de erosión para imágenes en escala de grises, surgen diferencias con respecto a la implementación realizada en imágenes binarias.

## **Implementación de Erosión Binaria**

A continuación se explica brevemente los pasos a seguir para implementar la operación de erosión sobre imágenes binarias.

El origen del EE debe ser colocado sobre cada uno de los píxeles de la imagen. Una vez puesto sobre un pixel, se verifica si todos los puntos que conforman el EE se encuentran incluidos en imagen, es decir, que la forma del EE se encuentra totalmente contenida en la imagen. De cumplirse esta condición, el valor del píxel en el cual se ubicó el origen del EE tomará el valor de “1” en la imagen erosionada, de lo contrario su valor será “0”.

### **1.7.1.2 Erosión en Imágenes en Escala de grises**

A diferencia del caso binario donde se busca la presencia del EE, en escala de grises se busca el valor más pequeño de la sustracción pixel a pixel, de la imagen menos el EE. El valor mínimo será colocado en la imagen erosionada, en las mismas coordenadas donde el origen del EE fue situado en la imagen original. Recordemos que el EE en imágenes en escala de grises por lo general es volumétrico, pudiéndose ser particularmente plano.

Adicionalmente al cambio de la forma de los objetos, la erosión en escala de grises afecta el brillo de la imagen. La imagen erosionada es más oscura que la original, debido que sustraemos a los valores de intensidad de la imagen el valor del EE, quedándonos con el mínimo valor de esta operación.

A Continuación se muestra una tabla, que representa los niveles de intensidad de una imagen en escala de grises. Para realizar la erosión utilizaremos el EE que

previamente se muestra en la Figura 1.15, donde el origen del EE será el centro geométrico de la figura, recordamos que no necesariamente estos dos puntos deben coincidir.

0	1	0
1	1	1
0	1	0

Figura 1.15 EE en forma de cruz 3x3

178	178	54	1	42	134	226
22	33	150	34	59	79	31
165	88	255	86	45	89	4
99	247	2	48	90	187	15
41	123	33	7	1	177	145
46	89	156	2	30	25	86
109	0	23	10	12	42	252

(a)

21	32	0	0	0	41	30
21	21	32	0	33	30	3
21	32	1	33	44	3	3
40	1	1	1	0	14	3
40	40	1	0	0	0	14
40	0	1	1	0	24	24
0	0	0	1	9	11	41

(b)

Figura 1.16 Conjunto de píxeles de una Imagen en escala de grises (a). Erosión del conjunto de píxeles de una Imagen escala de grises por un EE en forma de cruz 3x3 (b)

En la imagen de la Figura 1.16 (b) se observa una reducción muy significativa en los valores de cada píxel con respecto al valor inicial, salvo los píxeles que de entrada ya tenían valores bajos, la imagen globalmente se ha oscurecido. Otro aspecto importante que se puede apreciar, es la presencia marcada del EE, (zonas sombreadas), el resultado de la erosión, refleja en muchas zonas de la imagen, la forma del EE empleado para la operación.



Figura 1.17 Imagen de Lena en escala de grises (a). Imagen de Lena en escala de grises erosionada con el EE de la Figura 1.15 (b)

Para la erosión de la imagen de Lena en escala de grises se utilizó el EE de la figura 1.15. En la imagen erosionada (Figura 1.17 (b)) las zonas oscuras han crecido y queda implícito que las zonas blancas han decrecido. El crecimiento de las zonas oscuras desaparece las pequeñas zonas blancas de la imagen, eso genera una imagen más oscura y ligeramente borrosa, se ha perdido brillo y definición.

## **Procedimiento para realizar la erosión en escala de grises**

A continuación se explica brevemente los pasos a seguir para realizar la operación de erosión sobre imágenes en escala de grises.

El origen del EE debe ser colocado sobre cada uno de los píxeles de la imagen. En cada posición se resta a la imagen el EE, esta operación se efectúa punto a punto y se obtiene tantos valores como puntos tenga el EE. El valor mínimo obtenido será asignado en la imagen resultado (imagen erosionada), en las mismas coordenadas donde se situó el origen del EE en la imagen original. Vale la pena recordar que para una imagen en escala de grises los valores de los píxeles están acotados entre “0” y “255”. Para el caso de imágenes a colores, los diferentes planos que la conforman serán procesados como imágenes en escala de grises.

## 1.7.2 Dilatación

De igual manera que la erosión, la dilatación es una de las operaciones morfológicas fundamentales. La dilatación aumenta el tamaño de un objeto, y la forma en que aumenta depende del EE. La dilatación es el operador morfológico complementario de la erosión y el mecanismo para su empleo nuevamente dependerá si se aplica a una imagen binaria, en escala de grises ó color.

### 1.7.2.1 Dilatación en Imágenes Binarias

La dilatación de un conjunto A por medio de un conjunto B (EE) se define como, el conjunto de todos los puntos x, tal que la reflexión de B se superpone por al menos un elemento con el conjunto A. La ecuación (2) resume el enunciado anterior:

$$A \oplus B = \{x \mid ((\hat{B})_x \cap A) \subseteq A\} \quad (2)$$

“El resultado de la dilatación es el conjunto de puntos origen del EE B tales que el EE contiene algún elemento del conjunto A, cuando el elemento se desplaza por el espacio que contiene a ambos conjuntos” [Ortiz, 2002].

En el proceso de dilatación para imágenes binarias, los objetos de la imagen dilatada poseen más píxeles que la imagen original. Los píxeles adicionales surgen cuando el origen del EE se ubica en un píxel que pertenece al fondo, y existe algún punto del EE que se superpone con la imagen. Ese píxel ahora formará parte de la imagen dilatada. La dilatación cambia píxeles negros a blancos en la periferia de los objetos presentes en la imagen, causando un aumento de su área, inclusive puede hacer que diferentes objetos lleguen a unirse.

Para explicar el efecto de la dilatación en imágenes binarias, a continuación se muestra un ejemplo de este proceso. La imagen de la figura 1.18 (a) fue dilatada con un EE de forma línea vertical 3x1, el resultado se muestra en la imagen de la figura 1.18 (b). Tanto la imagen como el EE fueron utilizados en la sección de erosión binaria, de tal manera que, permite comparar el efecto que produce la erosión y la dilatación sobre una imagen binaria.

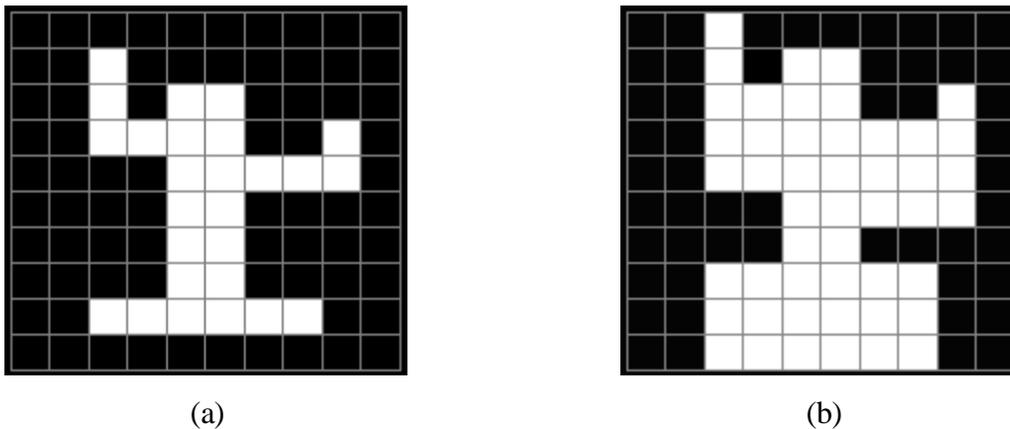


Figura 1.18 Imagen original (a). Imagen original dilatada por un EE 3x1 (b).

En la figura 1.18 (b) se aprecia un notable engrosamiento del objeto con respecto a la imagen original. El objeto ha crecido en la misma dirección del EE, es decir, de forma vertical, causando que la imagen dilatada posea más píxeles blancos que la imagen original. Recordamos que el EE empleado es simétrico, y su origen fue elegido de tal manera que coincidiera con su centro geométrico, entonces al realizar su reflexión, el origen de éste se mantiene en el mismo píxel.

A continuación en la Figura 1.20 (a) se muestra un conjunto de píxeles, similar a los de la Figura 1.13 (a) perteneciente a la sección de erosión en imágenes binarias, pero ahora dicho conjunto se empleará para la dilatación. Usando el mismo EE 1x2 con origen en el píxel de la izquierda (Ver Figura 1.19), nótese que al hacer la reflexión del EE, éste cambia hacia el píxel de la derecha. Recordemos que la reflexión del EE se debe efectuar cada vez que se realiza una operación de dilatación.

1	1
---	---

(a).

1	1
---	---

(b)

Figura 1.19 EE 1x2 (a). EE 1x2 reflejado (b)

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	0	0	1	0	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1
0	1	1	1	0	0	1	1
0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1
0	1	1	0	0	1	1	1
0	1	1	0	0	0	1	1
0	0	0	0	0	0	0	0

(b)

Figura 1.20 Conjunto de píxeles de una Imagen Binaria (a). Dilatación del conjunto de píxeles de una Imagen Binaria por un EE 1x2 (b).

A continuación la imagen de la Figura 1.21 (a) se muestra como ejemplo de dilatación binaria, ella está compuesta por líneas horizontales y verticales (Figura 1.14 (a) utilizada en el ejemplo de erosión binaria) y fue dilatada por dos Elementos Estructurantes, uno de forma línea horizontal y otro de forma línea vertical. Los resultados pueden apreciarse en las imágenes de la Figura 1.21 (b) y Figura 1.21 (c) respectivamente. La imagen original fue dilatada con el EE mostrado en la Figura 1.8 (g) y el resultado es una imagen donde las líneas verticales aumentaron de grosor y las líneas horizontales ligeramente en su longitud (Ver Figura 1.21 (b)). En cambio, si se emplea el EE mostrado en la Figura 1.8 (h) la imagen resultado será una en la que los objetos han crecido verticalmente, las líneas verticales en su longitud y las horizontales en su grosor (Ver Figura 1.21 (c)). Si bien la dilatación aumenta el tamaño de los objetos, podemos constatar que más allá de la operación, el EE determina fuertemente en qué dirección se produce dicho aumento.

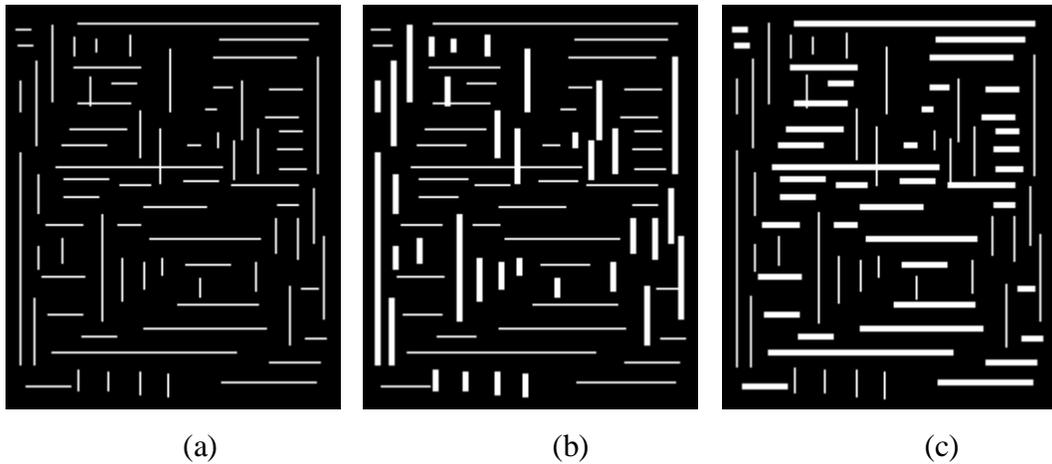


Figura 1.21 Imagen original (a). Imagen dilatada con un EE 1x3 (b). Imagen dilatada con un EE 3x1 (c).

## Implementación de Dilatación Binaria

A continuación se explica brevemente los pasos a seguir para implementar la operación de dilatación sobre imágenes binarias.

El origen del EE debe ser colocado sobre cada uno de los píxeles de la imagen. En cada posición, se verifica que al menos un punto del EE coincida con un pixel de la imagen, es decir, que el EE y la imagen se superpongan en al menos un elemento, De cumplirse esta condición, el valor del píxel en el cual se ubicó el origen del EE tomará el valor de “1” en la imagen erosionada, de lo contrario su valor será “0”.

### 1.7.2.2 Dilatación en Imágenes en Escala de grises

“El resultado de la dilatación de imágenes en escala de grises es, generalmente, una imagen de mayor valor, es decir, una imagen más clara, puesto que la dilatación maximiza el valor de los píxeles de la imagen” [Ortiz, 2002].

A Continuación se muestra una tabla, que representa los niveles de intensidad de una imagen en escala de grises, la misma que se usó en la sección de erosión en escala de grises (Ver Figura 1.16 (a)). El EE empleado para hacer la dilatación es el de la Figura 1.15, este elemento por ser simétrico con respecto a su origen, al realizar la reflexión de él, su origen no se ve modificado.

178	178	54	1	42	134	226
22	33	150	34	59	79	31
165	88	255	86	45	89	4
99	247	2	48	90	187	15
41	123	33	7	1	177	145
46	89	156	2	30	25	86
109	0	23	10	12	42	252

(a)

179	179	179	55	135	227	227
179	179	255	151	80	135	227
166	255	255	255	91	188	90
248	248	255	91	188	188	188
124	248	157	49	178	188	178
110	157	157	157	31	178	253
110	110	157	24	43	253	253

(b)

Figura 1.22 Conjunto de píxeles de una Imagen en escala de grises (a). Dilatación del conjunto de píxeles de una Imagen en escala de grises por el EE de la Figura 1.15 (b).

Se puede observar en la imagen de Figura 1.22 (b) la presencia total o parcial de la forma del EE algunas zonas de la imagen, las cuales fueron sombreadas para su identificación. Si comparamos ésta tabla con respecto al resultado de la erosión (Ver Figura 1.16 (b)), resulta que las zonas oscuras disminuyeron su presencia, ya que los valores de cada píxel han aumentado.

A continuación en la Figura 1.23, se muestra un ejemplo de dilatación de imágenes en escala de grises, empleando el EE de la Figura 1.15.



Figura 1.23 Imagen de Lena en escala de grises (a). Imagen dilatada de Lena en escala de grises por el EE de la Figura 1.15 (b).

En la imagen de la Figura 1.23 (b) (imagen dilatada) las zonas claras se han ensanchado y esto produce un aumento del brillo de la imagen. Al igual que en la erosión en escala de grises la imagen resultado se ve borrosa comparada con la imagen original.

## **Procedimiento para realizar la dilatación en escala de grises**

A continuación se explica brevemente los pasos a seguir para realizar la operación de dilatación sobre imágenes en escala de grises.

El origen del EE debe ser colocado sobre cada uno de los píxeles de la imagen. En cada posición se suma a la imagen el EE, esta operación se efectúa punto a punto y se obtiene tantos valores como puntos tenga el EE. El valor máximo obtenido será asignado en la imagen resultado (imagen dilatada), en las mismas coordenadas donde se situó el origen del EE en la imagen original. Vale la pena recordar que para una imagen en escala de grises los valores de los píxeles están

acotados entre “0” y “255”. Para el caso de imágenes a colores, los diferentes planos que la conforman serán procesados como imágenes en escala de grises.

### 1.7.3 Dualidad de la Erosión y Dilatación

La relación que existe entre la erosión y dilatación es de dualidad con respecto a la complementación. “Esto significa que una erosión es equivalente a la complementación de la dilatación de la imagen complementada con el mismo EE y viceversa” [Ortiz, 2002]. En otras palabras, lo que un operador le hace a la imagen el otro lo hace al fondo:

$$(A \ominus B)^c = A^c \oplus \hat{B} \quad (3)$$

En consecuencia se cumple que:

$$(A \oplus B)^c = A^c \ominus \hat{B} \quad (4)$$

“la acción de erosionar un conjunto A por un elemento de estructura B dado, es equivalente a dilatar el complemento del conjunto A con el reflejado del elemento de estructura B. Recíprocamente, dilatar un conjunto A por un elemento de estructura B, es equivalente a erosionar su complemento  $(A)^c$  con el reflejado del elemento de estructura B”. [Gamino, 2004].

A continuación se muestra un ejemplo de la dualidad entre la erosión y la dilatación. Este ejemplo fue elaborado empleando imágenes binarias. Se erosionó la imagen de Lena con el EE de la Figura 1.5 y se tomó su complemento, éste resultado debe ser igual, a tomar el complemento de la imagen original, y dilatar esta con el reflejado del EE, tal como se señala en la ecuación (3).



Figura 1.24 Imagen erosionada de Lena con el EE de la Figura 1.15 (a). Imagen complemento de la Figura anterior (b).

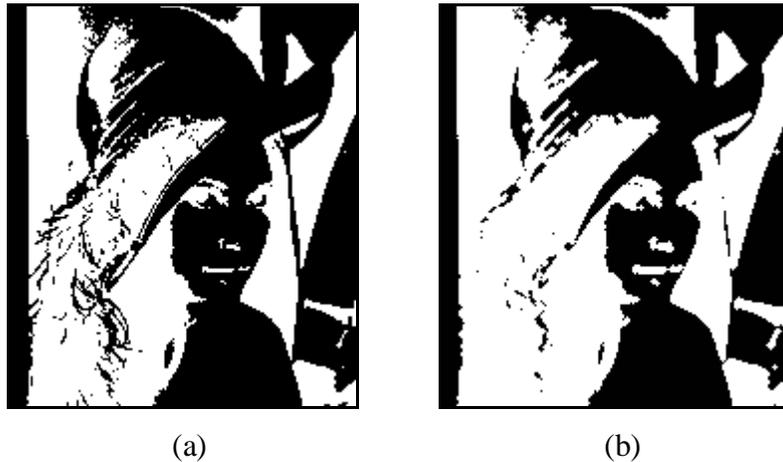


Figura 1.25 Imagen complemento de Lena (a). Imagen complemento de Lena erosionada con el EE reflejado de la Figura 1.15 (b).

Si se comparan las imágenes de las Figuras 1.24 (b) y 1.25 (b) resultarían que son idénticas, esto como consecuencia de la relación de dualidad entre la erosión y la dilatación. A continuación se muestra otro ejemplo, esta vez empleando la ecuación (4), nuevamente si se comparan pixel a pixel las imágenes (b) resultarían que son idénticas. Ver Figuras 1.26 y Figuras 1.27.

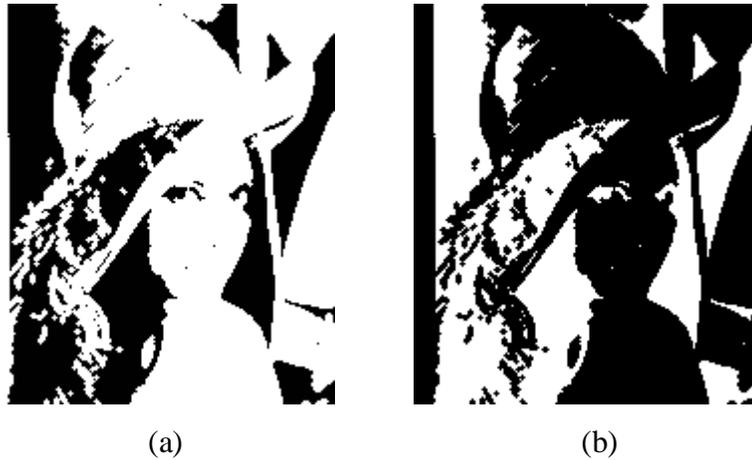


Figura 1.26 Imagen erosionada de Lena con el EE de la Figura 1.15 (a). Imagen complemento de la Figura anterior (b).

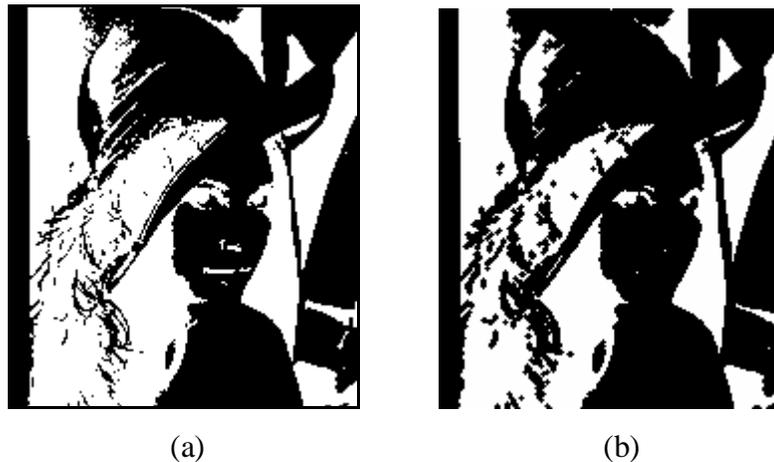


Figura 1.27 Imagen complemento de Lena (a). Imagen complemento de Lena erosionada con el EE reflejado de la Figura 1.15 (b).

### 1.7.4 Apertura y Cierre

Existen otras operaciones importantes que se derivan de aplicar los operadores básicos (erosión y dilatación), esas operaciones son llamadas: Apertura y Cierre. La propiedad más relevante de estas transformaciones es la ídem potencia (una vez hecha la transformación, la aplicación sucesiva de ésta operación no altera el primer resultado). Algunos autores consideran que la apertura y el cierre son tan básicos como la erosión y dilatación.

### 1.7.4.1 Apertura

Es la acción de efectuar las operaciones de erosión y dilatación, cumpliendo con ese orden. La ecuación que se presenta a continuación resume esta definición:

$$A \circ B = (A \ominus B) \oplus B \quad (5)$$

“La apertura de una imagen es independiente del origen del EE, puesto que si la erosión se corresponde con una intersección de traslaciones, la dilatación que sigue es una unión de traslaciones en dirección opuesta. El tamaño y la forma de los elementos estructurantes empleados en la apertura deben ser acordes con las estructuras de la imagen que se desean eliminar. En ocasiones, tamaños elevados de elementos estructurantes eliminarán formas indeseables en una imagen pero afectarán al resto de las estructuras. Tamaños reducidos serán óptimos cuando las imágenes contengan pequeños detalles” [Ortiz, 2002]. A continuación se muestran un par de ejemplos donde se aplicó la operación de apertura, usando el EE de la Figura 1.15.

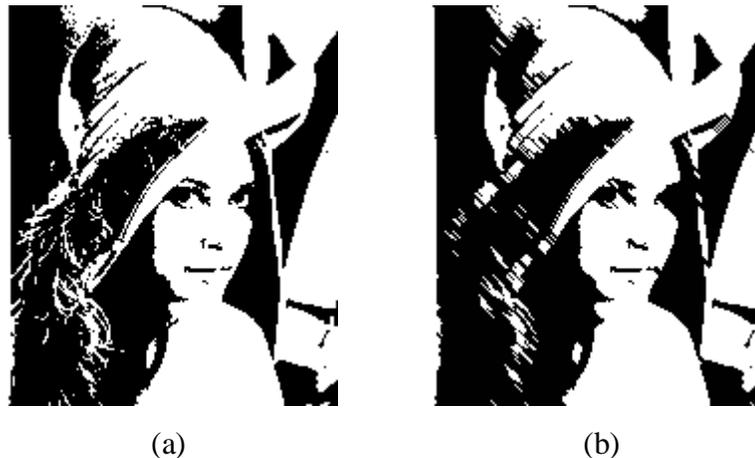


Figura 1.28 Imagen de Lena en blanco y negro (a). Apertura de la imagen anterior con el EE de la Figura 1.15 (b).



Figura 1.29 Imagen de Lena en escala de grises (a). Apertura de la imagen anterior con el EE de la Figura 1.15 (b).

“La apertura al igual que la erosión cumple con las propiedades de ser una operación antiextensiva y creciente. Son variados los efectos que produce la apertura en una región de interés: Se eliminan islas de tamaño menor al elemento de estructura, se eliminan picos o cabos más delgados que el elemento de estructura se rompen istmos cuya anchura sea menor al diámetro del elemento de estructura” [Gamino, 2004].

### 1.7.4.2 Cierre

Es la acción de efectuar las operaciones dilatación y erosión, cumpliendo con ese orden. La ecuación que se presenta a continuación resume esta definición

$$A \bullet B = (A \oplus B) \ominus B \quad (6)$$

“Al igual que la apertura, el cierre de una imagen es independiente del origen del EE”. La dilatación que se realiza primero une estructuras que la erosión no puede separar. En una imagen de niveles de grises el cierre morfológico ayuda a eliminar estructuras oscuras menores en tamaño al EE [Ortiz, 2002]. A continuación se

muestran un par de ejemplos donde se aplicó la operación de cierre, usando el EE de la Figura 1.15.



Figura 1.30 Imagen de Lena en blanco y negro (a). Cierre de la imagen anterior con el EE de la Figura 1.15 (b).



Figura 1.31 Imagen de Lena en escala de grises (a). Cierre de la imagen anterior con el EE de la Figura 1.15 (b).

“El cierre al igual que la dilatación cumple con las propiedades de ser una operación extensiva y creciente y produce efectos diferentes a los que produce la apertura. El cierre produce los siguientes efectos: se rellenan los lagos o huecos de tamaño menor al elemento de estructura, se rellenan rajaduras o golfos más delgados

que el elemento de estructura, se funden estrechos cuya anchura sea menor al diámetro del elemento de estructura” [Gamino, 2004].

## 1.8 Aplicaciones

Las aplicaciones de PDI son operaciones secundarias que se derivan de las operaciones básicas: erosión y dilatación. Estas operaciones secundarias son herramientas que nos permiten extraer información de las imágenes digitales. Entre las aplicaciones más utilizadas podemos mencionar:

- Filtros Morfológicos.
- Extracción de Bordes.
- Granulometría.

### 1.8.1 Filtros Morfológicos

Llamamos filtros morfológicos a la aplicación de las operaciones apertura-cierre y cierre-apertura, estas operaciones también pueden identificarse por las operaciones más básicas que le dieron origen, es decir, el primer filtro es erosión-dilatación-dilatación-erosión (E\_D\_D\_E), y el segundo filtro es dilatación-erosión-erosión-dilatación (D\_E\_E\_D), respectivamente:

- Filtro Morfológico: Apertura-Cierre (E\_D\_D\_E)
- Filtro Morfológico: Cierre-Apertura (D\_E\_E\_D)

“Las aplicaciones básicas de los filtros morfológicos son la atenuación del ruido y la extracción selectiva de objetos en la imagen. Ambas aplicaciones son importantes en visión artificial, pues técnicas de interpretación o medida fallaran en

presencia de ruido, mientras que la selección de estructuras y objetos es fundamental en los procesos de segmentación” [Ortiz, 2002].

A continuación en las Figura 1.32 y Figura 1.33 se muestran ejemplos de Filtros Morfológicos. La imagen binaria de un objeto de forma circular servirá como imagen original. A esa imagen se le añadieron puntos blancos en el fondo, y puntos negros al círculo, para simular ruido en la imagen.

En la Figura 1.32 se utilizó el filtro morfológico Apertura-Cierre con el EE de la figura 1.15. Se observa que después de aplicar el filtro, el ruido presente en la imagen resultado fue eliminado por completo, tanto dentro del objeto como fuera del círculo (ver Figura 1.32 (c)).

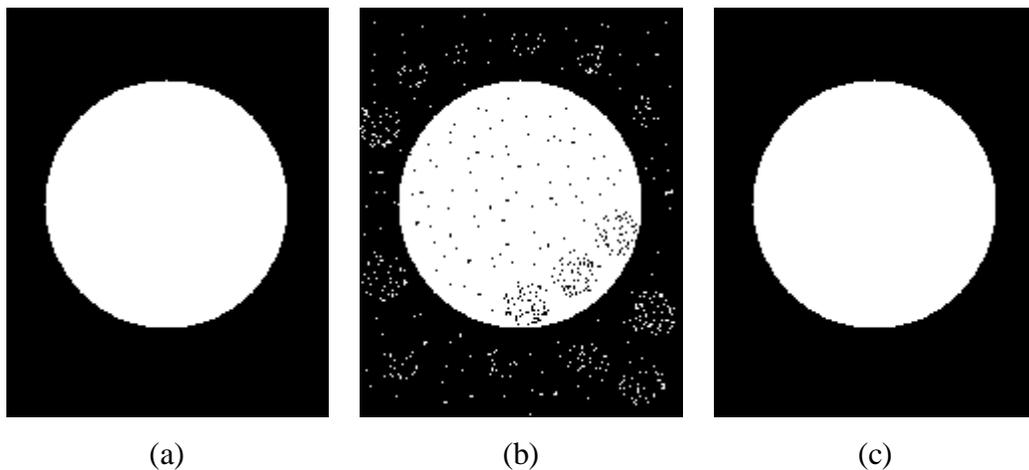


Figura 1.32 Imagen original (a). Imagen con ruido (b). Imagen de filtrada (c).

En cambio, en la Figura 1.33 se utilizó el filtro morfológico Cierre-Apertura con EE de la figura 1.15. Nuevamente se observa que después de aplicar el filtro, el ruido fue suprimido completamente de la imagen (ver Figura 1.33 (c)). Con la utilización de los filtros morfológicos, tanto el ruido que se encuentra dentro y fuera del círculo se eliminó por completo.

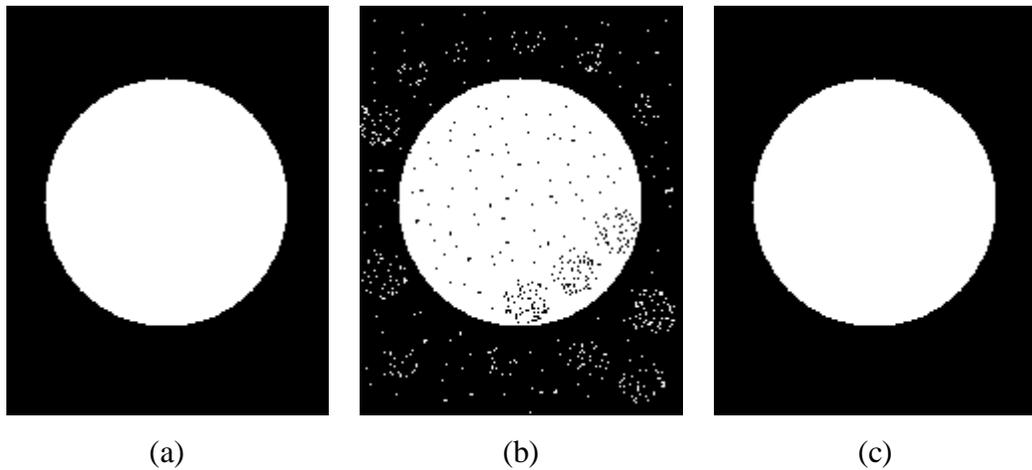


Figura 1.33 Imagen original (a). Imagen con ruido (b). Imagen de filtrada (c).

Los filtros morfológicos apertura y cierre, permiten suprimir ruido indeseable presente en las imágenes, acondicionándolas para la posterior aplicación de operaciones morfológicas.

## 1.8.2 Extracción de bordes

Llamamos bordes de una imagen, al contorno de los objetos que pertenecen a ella. Con la extracción de bordes se obtiene la silueta de la imagen, que corresponde a los puntos de frontera del objeto y del fondo, en ese sentido se puede hablar de un borde interno y externo respectivamente. Cuando es de interés tener en cuenta a ambos bordes (interno y externo) estamos considerando el borde total de la imagen.

### 1.8.2.1 Borde Interno

El borde interno son todos los puntos que pertenecen a los objetos de la imagen y colindan con el fondo. Para extraer el borde interno primero se realiza una erosión de la imagen, luego la imagen resultante se sustraerá a la imagen original, es

decir, se resta el valor píxel a píxel de cada una de las imágenes, la imagen original “menos” la imagen erosionada.

A continuación se muestra un ejemplo del borde interno de una imagen.

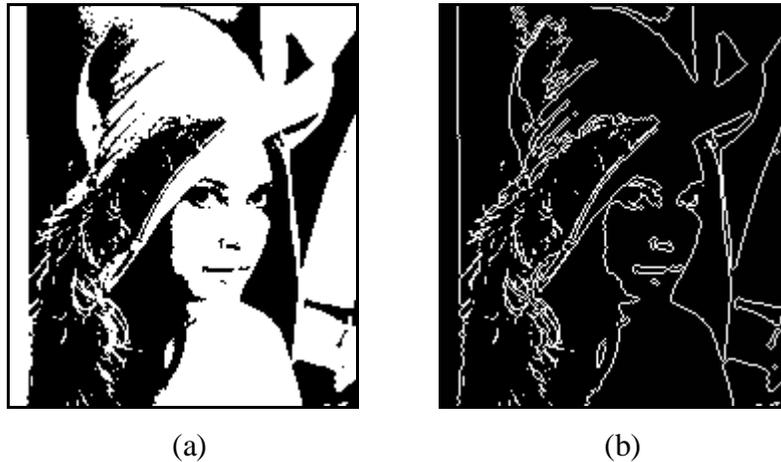


Figura 1.34 Imagen blanco y negro de Lena (a). Borde interno de la imagen anterior con el EE de la Figura 1.15 (b).

### 1.8.2.2 Borde Externo

El borde externo son todos los puntos que pertenecen al fondo y colindan con los objetos de la imagen. Para extraer el borde externo primero se realiza una dilatación de la imagen, luego la imagen original será sustraída de la imagen resultante (la dilatada), es decir, se resta el valor píxel a píxel de cada una de las imágenes, la imagen dilatada “menos” la imagen original.

A continuación se muestra un ejemplo del borde externo de una imagen.

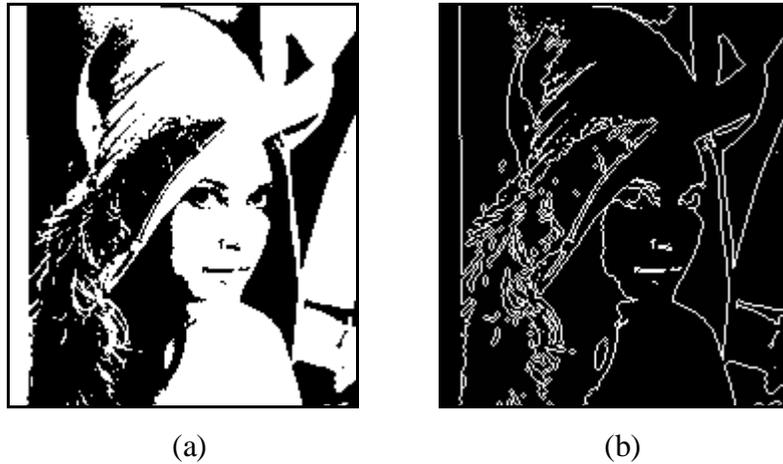


Figura 1.35 Imagen blanco y negro de Lena (a). Borde externo de la imagen anterior con el EE de la Figura 1.15 (b).

### 1.8.2.3 Borde Total

El borde total de una imagen es la unión de su borde externo e interno. El borde total puede obtenerse directamente la sustrayendo a la imagen dilatada la imagen erosionada, es decir, se resta el valor píxel a píxel de cada una de las imágenes, la imagen dilatada “menos” la imagen erosionada. A continuación se muestra un ejemplo del borde total de una imagen.

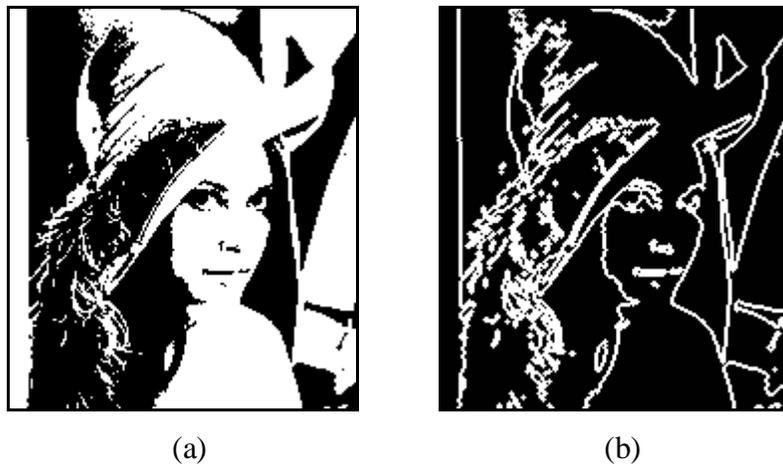


Figura 1.36 Imagen blanco y negro de Lena (a). Borde total de la imagen anterior con el EE de la Figura 1.15 (b).

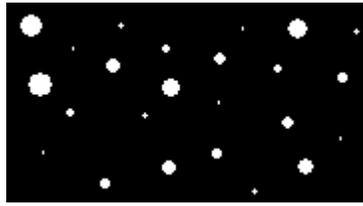
### 1.8.3 Granulometría

Aunque el uso del término “granulometría” es empleado en gran medida para la medición y graduación de granos de formación sedimentaria, este concepto puede ser usado para referirse a la acción de contabilizar el número de objetos (granos) de diferentes tamaños que se encuentran presentes en una imagen.

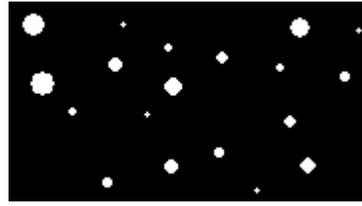
La granulometría se realiza efectuando operaciones de apertura con elementos estructurantes de diámetros crecientes, al finalizar cada proceso de apertura se restan las dos últimas imágenes, la imagen previa menos la última, y se contabiliza el número de píxeles blancos que resultan de la resta. El resultado de cada sustracción se divide entre el área del EE de diámetro “d-1”, el cociente indica el número de objetos de diámetro “d-1” presentes en la imagen. Esta operación se debe realizar de forma sucesiva aumentando el diámetro del EE de tal manera que se eliminen todos los objetos presentes en la imagen, hasta que todos hayan sido contabilizados.

A continuación se muestra una serie de imágenes que son producto de las operaciones de apertura empleando elementos estructurantes crecientes sobre la imagen “Granos-01”. En cada imagen se observa cómo se eliminaron objetos debido a las aperturas, y debajo de cada una de ellas se muestra el valor del diámetro del EE que se usó para producir dicha imagen, desde diámetro = 3 píxeles hasta diámetro = 13 píxeles. En la última imagen no se encuentra ningún objeto en la imagen, ya que todos se han sido eliminados, sólo se aprecia el fondo de la imagen.

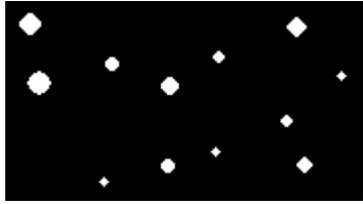
Para obtener el resultado de la granulometría es necesario utilizar una herramienta para contabilizar los tipos de objetos y la cantidad de ellos presentes en la imagen, esto puede ser representado mediante el uso de un histograma.



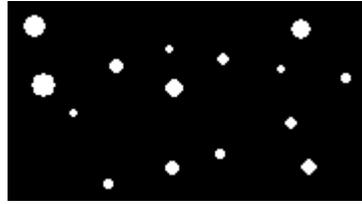
Granos-01 (a)



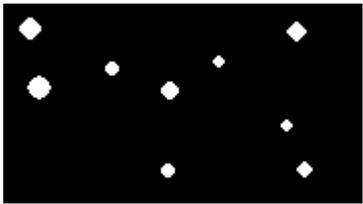
Diámetro = 3 (b)



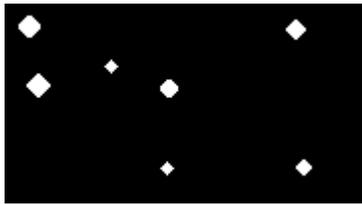
Diámetro = 4 (c)



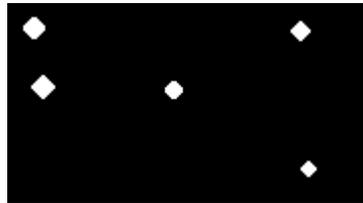
Diámetro = 5 (d)



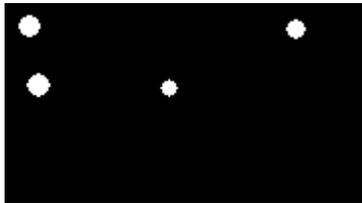
Diámetro = 6 (e)



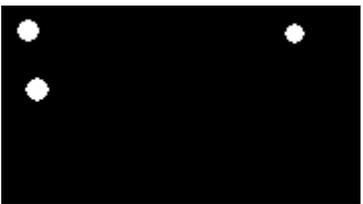
Diámetro = 7 (f)



Diámetro = 8 (g)



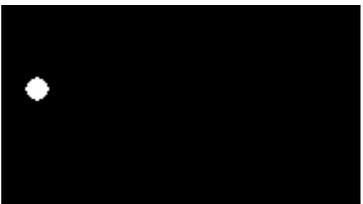
Diámetro = 9 (h)



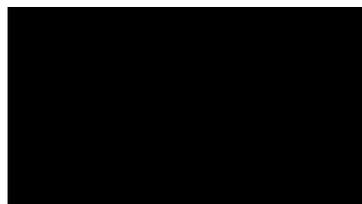
Diámetro = 10 (i)



Diámetro = 11 (j)



Diámetro = 12 (k)



Diámetro = 13 (l)

Figura 1.37 Imagen original (a). Imágenes de las múltiples apertura (b)–(l).

## 1.8.4 Histograma

“Un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. En el eje vertical se representan las frecuencias, y en el eje horizontal los valores de las variables” [Plan Ceibal]. El histograma puede ser usado para representar el resultado de la granulometría, en el eje vertical estará la frecuencia o número de objetos presentes en la imagen y en el eje horizontal el tipo de objetos, es decir, el valor de cada diámetro de los objetos presentes en la imagen. Contabilizando los parámetros nombrados anteriormente, podemos elaborar el histograma.

A continuación se muestra la Figura 1.38, el histograma de la imagen “Granos-01.bmp” la imagen que se utilizó para ejemplificar el proceso de granulometría, este es el resultado de dicho proceso:

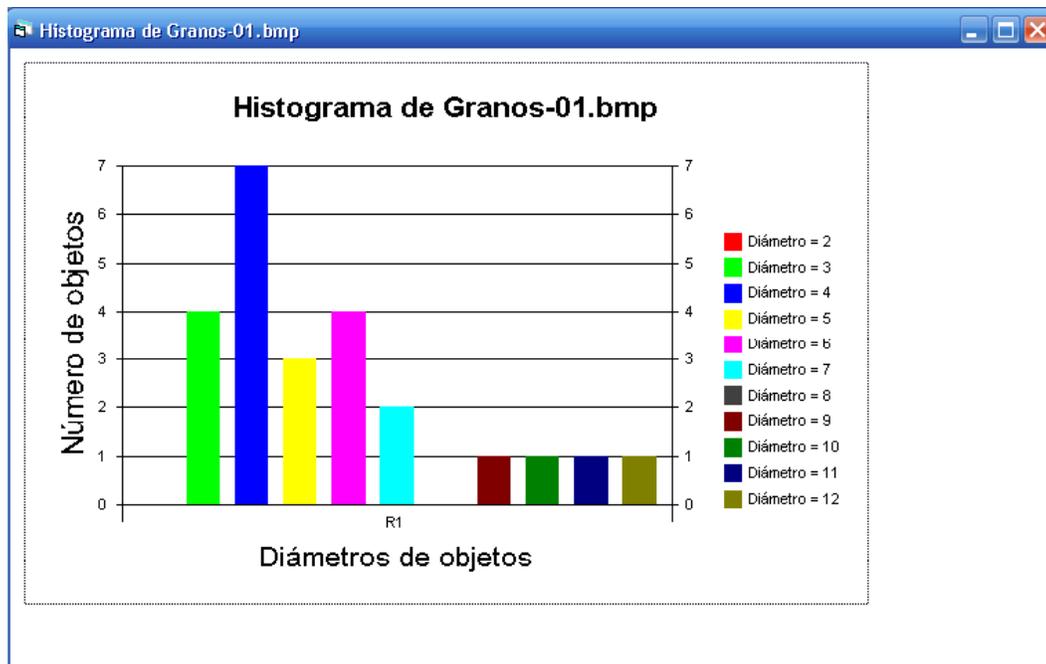


Figura 1.38 Histograma de la Figura 1.38 (a)

## Capítulo 2. Marco Metodológico

En este capítulo se explica la estructura del programa PDI, su interfaz gráfica, la implementación de sus rutinas y del sistema de ayuda. Para visualizar una imagen tendremos una rutina que verifica primero su información, de estar correcta, ésta se carga en memoria dependiendo del tipo de imagen (desde 1 bit hasta 32 bits), por último la imagen es mostrada en el monitor. Una vez que la imagen está abierta se puede procesar morfológicamente, ajustar su brillo-contraste o convertirla en escala de grises o blanco y negro, etc. En este capítulo se explica cómo fue la implementación de las operaciones básicas erosión y dilatación, que son el fundamento de todas las operaciones morfológicas.

### 2.1 Interfaz Gráfica de PDI

La interfaz gráfica es la parte del software que permite la interacción con el usuario. En el PDI la interfaz consta de una ventana principal donde se encuentra una barra de menú con múltiples opciones: *Archivo*, *Ajustes*, *Convertir*, *Op. Morfológicas*, *Aplicaciones* y *Ayuda* (Ver Figura 2.1). La ventana cumple con características estándar, intuitivamente y sin mayor experiencia el usuario puede usarla.

El programa PDI consta de rutinas que permiten leer archivos con formato BMP y mostrar las imágenes en la pantalla. Una vez que una imagen ha sido mostrada y acondicionada, se le puede aplicar operaciones básicas de morfología matemática así como operaciones más complejas, entre ellas se encuentran: filtros morfológicos, extracción de bordes y granulometría. Para cada operación, el resultado es mostrado en la pantalla como una nueva imagen, de ésta manera la imagen resultado puede ser comparada directamente con la imagen previa.



Figura 2.1 Menú del programa PDI

## Archivo

En el menú *Archivo* se encuentran las opciones más generales del programa como: *Abrir Imagen* y *Guardar Imagen*, *Propiedades*, *Salir*. Las primeras dos como sus nombres lo mencionan permiten abrir y guardar las imágenes. Cuando una imagen se visualiza en la pantalla, la opción *Propiedades* muestra el valor de sus atributos: altura, ancho, número de bits por píxel, etc. Por último, la opción *Salir* permite cerrar el programa.

## Ajustes

En el menú *Ajustes*, está contenida solamente una opción que se encarga de modificar el brillo y el contraste de una imagen, y es importante porque frecuentemente es necesario acondicionar la imagen original para su posterior procesamiento. Aunque este menú sólo contiene un ítem, se ha dejado intencionalmente así para posteriormente añadir otros ajustes como por ejemplo: filtro gamma, brillo y contraste por zonas, etc.

## Convertir

En el menú *Convertir* se encuentran tres opciones que modifican la apariencia cromática de la imagen, éstos son: *Color a Escala de Grises*, *Escala de Grises a*

*Blanco y Negro*, e *Invertir Blanco y Negro*. En la primera opción se transforma de una imagen a color a una imagen en escala de grises. Para la segunda opción es necesario establecer un valor de umbral, luego la imagen a color o gris, es convertida a una imagen blanco y negro (imagen binaria). La última opción es *Invertir Blanco y Negro*, allí se intercambian la imagen con su fondo.

## Op. Morfológicas

En este menú se encuentran las operaciones morfológicas elementales: *Erosión* y *Dilatación*, así como las operaciones de *Apertura* y *Cierre*. En éste menú también se encuentra la opción de establecer el EE que se usará en las operaciones morfológicas (El programa utiliza por defecto un EE 3x3 en forma de Cruz).

## Aplicaciones

En el menú *Aplicaciones* se encuentran las operaciones morfológicas más complejas, ellas son: *Granulometría*, *Filtro E\_D\_D\_E*, *Filtro D\_E\_E\_D*, *Borde Interno*, *Borde Externo* y *Borde Total*. Los filtros morfológicos nombrados anteriormente se corresponden con las operaciones de Apertura-Cierre y Cierre-Apertura respectivamente. Todas las aplicaciones son funciones más complejas basadas en la implementación en secuencia de los operadores más esenciales.

## Ayuda

El menú *Ayuda* es una parte fundamental de este trabajo. Éste menú está compuesto por un documento guía sobre el formato de imágenes BMP y una explicación de cómo fueron implementadas las operaciones morfológicas más importantes: erosión y dilatación. Además, se ha incorporado el código fuente debidamente comentado para facilitar su comprensión y seguimiento.

## 2.2 Visualización de imágenes digitales BMP

La rutina que se encarga de la visualización de la imagen en la pantalla, lee la información contenida en el archivo BMP y verifica la integridad de la imagen, de tal manera de extraer la información requerida para mostrarla en la pantalla. A continuación se muestra en la Figura 2.2 el diagrama de flujo del conjunto de rutinas que permiten la visualización de la imagen, éste diagrama de flujo es una visión general de las acciones que se realizan para la visualización de la imagen.

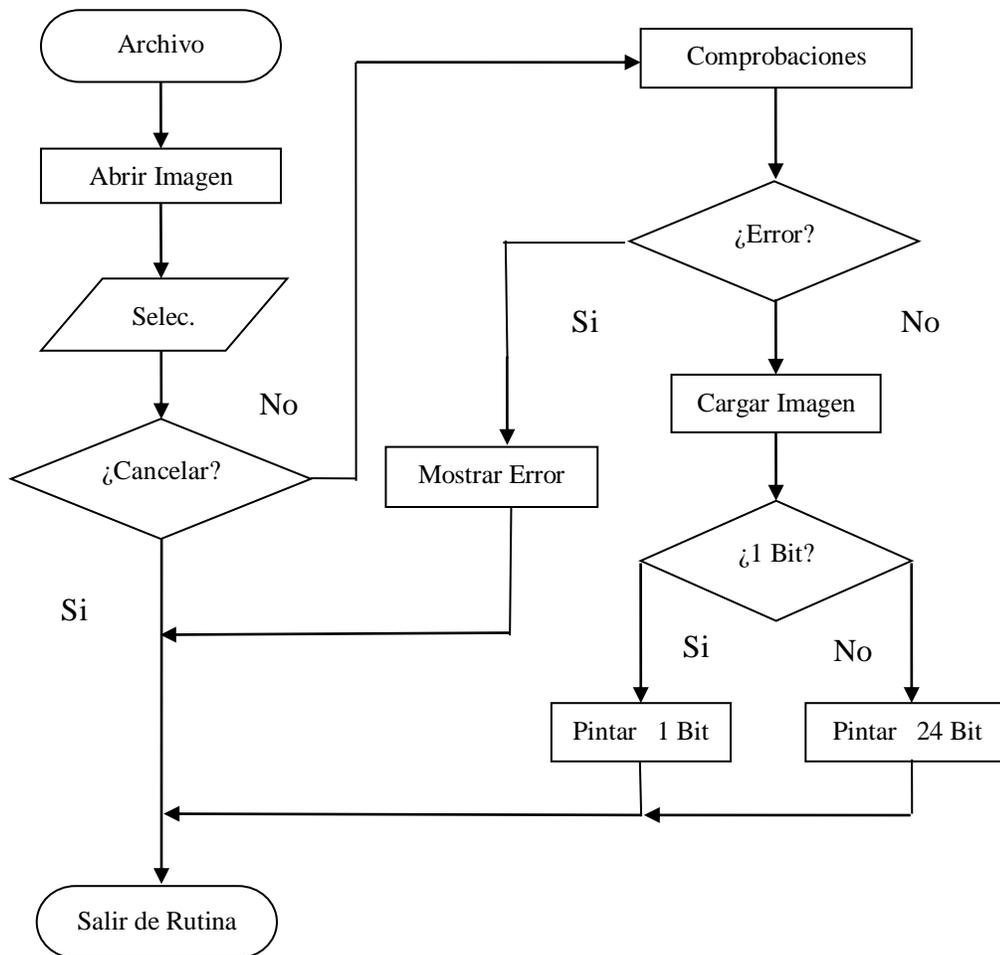


Figura 2.2 Diagrama de Flujo: Visualización de imágenes digitales BMP

Para poder visualizar la imagen se deben cumplir tres etapas, éstas son: comprobaciones, carga de la imagen y mostrar la imagen. La primera etapa son las comprobaciones, allí nos aseguramos que por lo menos la información que se encuentra en el encabezado, no se encuentre dañada. En la siguiente etapa se carga la información de la imagen en la memoria del computador; Luego en la última etapa se muestra la imagen en la pantalla del monitor.

### 2.2.1 Comprobaciones

En el bloque de comprobaciones, se lee y verifica parte de la información presente en el encabezado de la imagen, por medio de una función que devolverá un resultado al concluir la comprobación. Si alguno de los registros no supera la comprobación, se mostrará un mensaje de “error” en la pantalla indicando cual es la irregularidad, en caso contrario si la comprobación es exitosa, el programa continuará ejecutándose. La información que se chequea es la siguiente:

- a. Identificador.
- b. Tamaño del Archivo.
- c. Compresión.
- d. Tamaño de los Datos del Mapa de Bits.

#### Identificador:

Los primeros dos bytes que se leen de un BMP son referente a su identificación, allí deben encontrarse las letras “**BM**” para que efectivamente sea un archivo BMP. El programa PDI verifica la condición anterior, si ésta no se cumple se mostrará en pantalla el siguiente error: *"No es una imagen tipo: '.bmp'",* y no se abrirá el archivo.

## Tamaño del Archivo:

El tamaño del archivo indica el número de bytes necesarios para almacenar la imagen (la información del encabezado y la información de la imagen en sí). Si existe alguna irregularidad en la verificación de este ítem, el programa mostrará el siguiente error: *"El archivo está dañado"*. Para ejemplificar este ítem se usará la imagen binaria de la Figura 2.3.

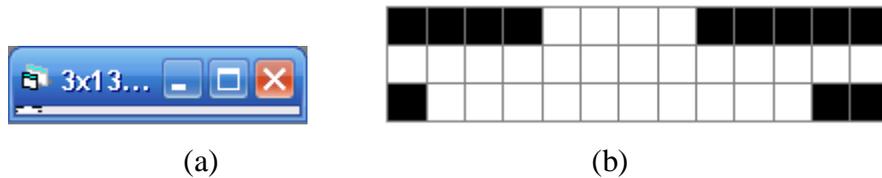


Figura 2.3 Imagen 3x13 (a). Ampliación de la Imagen 3x13 (b)

El archivo de la imagen binaria “3x13” tiene un tamaño de 74 bytes, la información de su encabezado posee una longitud de 62 bytes, y ésta se indica en el campo *Desplazamiento de Datos del BMP*. Por su parte, la información que concierne a la imagen ocupa una longitud de 12 bytes, como lo muestra el campo el de *Tamaño de Datos del BMP*, la suma de los dos campos mencionados anteriormente conforman el tamaño total del archivo de la imagen “3x13”.

A continuación en la Figura 2.4 se muestran las propiedades de la imagen “3x13” y se señalan los campos correspondientes al *Desplazamiento de Datos del BMP*, *Tamaño de Datos del BMP* y *Tamaño del Archivo*.

Propiedades	
Identificador	BM
Tamaño del Archivo	74
Reservado	0
Desplaz. de Datos del BMP	62
Tamaño del Encab. del BMP	40
Ancho	13
Altura	3
Planos	1
Bit Por Pixel	1
Compresión	0
Tamaño de Datos del BMP	12
Resolución H	0
Resolución V	0
Colores	0
Colores Importantes	0

$62 + 12 = 74$

Figura 2.4 Propiedades de la Imagen 3x13

### Compresión:

La información de un archivo BMP puede estar comprimida o no. Las rutinas del programa PDI que se encargan de abrir y guardar imágenes BMP están implementadas para trabajar con imágenes sin comprimidas.

Cuando una imagen BMP no es comprimida, en el campo de *Compresión* está colocado un valor de cero (0), cualquier otro valor indicaría que la imagen está comprimida y el programa devolverá el siguiente mensaje de error: *"No se puede abrir archivos comprimidos"* y la imagen no se abrirá.

## Tamaño de los Datos del Mapa de Bits:

Como se mencionó anteriormente, el *Tamaño de los Datos del Mapa de Bits* es el campo que almacena el producto de las dimensiones de la imagen, ese número es guardado en el archivo en múltiplos de 4 bytes. La imagen “3x13” servirá de ejemplo para mostrar cómo se utiliza la información de este campo.

La imagen “3x13” posee 13 bits de ancho y 3 líneas de bits de alto. Los 13 bits de ancho de la imagen son almacenados en múltiplos de 4 bytes en el archivo, por ésta razón, cada línea de ancho será completada con “ceros” hasta que obtenga esta característica. Si leemos la información de la imagen “3x13” sólo el campo referente al *Tamaño de Datos del BMP* obtendremos 12 bytes, el valor de cada uno de ellos se refleja a continuación:

```
127; 224; 00; 00. = 0111 1111; 1110 0000; 0000 0000; 0000 0000.  
255; 248; 00; 00. = 1111 1111; 1111 1000; 0000 0000; 0000 0000.  
15; 00; 00; 00. = 0000 1111; 0000 0000; 0000 0000; 0000 0000.
```

Cada uno de los números que están a la izquierda del “=” representan 1 byte, sumándolos, resultan ser los 12 bytes que componen el campo *Tamaño de los Datos del Mapa de Bits* de la imagen “3x13” (ver Figura 2.4).

La imagen “3x13” es binaria, el valor de cada píxel está guardado en cada bit respectivamente, por lo tanto los números mostrados en negritas representan los valores de los pixeles de la imagen. Debemos recordar que la información de la imagen se guarda al revés, de abajo hacia arriba, así que la primera línea que leemos es realmente la última línea de la imagen.

Si al finalizar la etapa de las comprobaciones no se ha producido ningún error, el programa continua con las rutinas de carga de la imagen, de lo contrario se mostrará el siguiente error en la pantalla: *"la imagen está dañada"*.

Para cargar la información de la imagen en memoria utilizaremos 6 rutinas, una por cada tipo de imagen (campo *Bit Por Píxel*). En éste punto ya hemos comprobado que el archivo que se está empleando en efecto es una imagen de formato BMP y por lo menos la información de su encabezado no se encuentra dañada.

### **2.2.2 Carga de la Imagen en memoria**

Una vez que la imagen haya superado las comprobaciones, se procede a abrir la comunicación con el disco para recuperar la información de la imagen, y así cargarla en memoria, en forma de una matriz, que tendrá uno o tres planos dependiendo si es una imagen en blanco y negro o si la imagen es gris o color respectivamente. Los tipos de imágenes son diferenciados dependiendo su valor de bit por píxel, y éstos son:

- Bit Por Píxel = 1
- Bit Por Píxel = 4
- Bit Por Píxel = 8
- Bit Por Píxel = 16
- Bit Por Píxel = 24
- Bit Por Píxel = 32

Al concluir con la selección anterior, todas las imágenes excepto las que poseen 1 bit por píxel son convertidas a imágenes de 24 bits y dibujadas como tal. Este ajuste se realiza para hacer un estándar en las operaciones morfológicas, teniéndose así dos grupos de operaciones morfológicas: las que operan sobre imágenes de 1 bit por píxel y las que operan sobre imágenes de 24 bits.

Cuando se estudia imágenes digitales, por lo general se empiezan con las imágenes en blanco y negro hasta llegar a las imágenes a color. Entonces por razones didácticas, las imágenes de 1 bit no fueron convertidas a 24 bits, se conservó el formato porque es fundamental e importante en el proceso de enseñanza del tema.

Como existen seis posibilidades de bit por pixel hay seis rutinas diferentes para procesar imágenes. No obstante, hay aspectos que son comunes en todas ellas, en general poseen dos matrices, en una se recupera y se carga la información del archivo y en la otra matriz se almacena la imagen lista para ser procesada morfológicamente.

### Bit Por Píxel = 1

Las imágenes más elementales del formato BMP son las que poseen una profundidad de 1 bit por píxel, a pesar de su sencillez por sólo ser en blanco y negro, éstas guardan una pseudo compresión, ya que en un byte se almacena la información de 8 pixeles. Para descompilar la información de cada byte emplearemos la operación lógica “AND” usando una máscara<sup>5</sup> adecuada. Este recurso permite obtener la información de un bit en particular descartando la información de los otros, haciendo el proceso progresivamente se obtiene la información de todo el byte.

Cuando se chequea cada bit, se empieza con el bit más significativo de cada byte, hasta llegar con el bit menos significativo. A continuación se muestra cual es la secuencia que se sigue desde el primer bit que se chequea hasta el último bit:

---

<sup>5</sup> La máscara en informática es el conjunto de datos que, junto con una operación permiten extraer selectivamente ciertos datos almacenados en otro conjunto. El tipo de máscara más frecuente es la llamada máscara de bits (bitmask, en inglés), que extrae ciertos bits particulares de las cadenas binarias. [Wikipedia, Máscara (informática)].

1000 0000  
0100 0000  
0010 0000  
0001 0000  
0000 1000  
0000 0100  
0000 0010  
0000 0001

Cada vez que se testea un bit el resultado de la operación será True o False, uno o cero. Si el resultado es igual a “1” se guardará como “255” en una tercera matriz, que complementa a las dos matrices generales que fueron previamente explicadas. La tercera matriz se emplea para almacenar la información del color después de realizar la descompilación de la información del archivo.

El ancho más pequeño que puede tener un archivo BMP es de 4 bytes (32 bits), debido a la forma como fue creado el formato, así que si tuviéramos una imagen de 1 píxel por 1 píxel ésta se guardaría en 1 bit de los 4 bytes mencionados anteriormente, dejando los otros 31 bits en cero.

Bit Por Píxel = 4

En las imágenes de 4 BPP la información de un pixel se guarda en medio byte, es decir, cada byte contiene la información de 2 píxeles, pero no se guarda directamente el color del pixel sino el índice asociada a una Paleta de colores.

Al comenzar con ésta subrutina la información de la paleta se guarda en una matriz, se lee la información de la imagen y se determina su ancho real. Nuevamente se efectúa una operación lógica AND, esta vez para seleccionar cada mitad del byte, la más significativa y la menos significativa. El proceso se realiza como se muestra a continuación:

**1111 0000**  
**0000 1111**

El índice asociado a la paleta es un número que va desde el 0 al 15. En la segunda mitad (la menos significativa) el resultado es directamente un número entre 0 y 15. Pero para la mitad más significativa del byte debemos darle el “peso” correspondiente, es decir, dividir entre 16 ese valor, para desplazar el índice del color que se encuentra en los bits más significativos hacia los bits menos significativos.

Para finalizar se emplea la matriz que contendrá la información de la imagen, cada coordenada de ancho y alto es cargada con su color respectivo. Luego de este proceso la imagen se encuentra lista para ser mostrada.

**Bit Por Píxel = 8**

El procedimiento para abrir imágenes de 8 BPP es parecido al que se empleó para abrir imágenes de 4 bits. Sin embargo, dado que se utilizan 8 bits para el índice, la paleta tiene 256 colores a diferencia de las imágenes de 4 bits que la paleta era tan solo de 16 colores.

En esta subrutina se emplean las dos matrices generales explicadas anteriormente, e igual que en el caso de las imágenes de 4BBP se usa una matriz adicional que almacena la información de la paleta. La matriz que contendrá la información procesada de la imagen, es cargada en cada coordenada ancho y alto con su con su color respectivo. Luego de este proceso la imagen se encuentra lista para ser mostrada.

## Bit Por Píxel = 16

Las imágenes de 16 bits (2 bytes) emplean tres bloques de cinco bits para almacenar la información, se utilizan 5 bits para el color rojo, 5 bits para el color verde y 5 bits para el color azul, dejando al bit más significativo sin emplearse. La información del color se encuentra y se chequea como se muestra a continuación:

Rojo: 0**111** **1100** 0000 0000

Verde: 0000 00**11** **1110** 0000

Azul: 0000 0000 000**11111**

El color que está hacia los bits más significativo es el rojo, luego le sigue el verde (la información de éste se encuentra entre los dos bytes) y por último el azul. A diferencia de las imágenes de 4 y 8 BPP en las de 16 BPP no se emplea paleta, así que el color es cargado directamente en la matriz de la imagen.

## Bit Por Píxel = 24

Este es el mismo formato que usa la pantalla (RGB), es la información exacta que ella necesita. En cada tres bytes se almacena la información de un píxel, de tal manera que cada byte guarda la información de una componente de color Rojo, Verde y Azul de cada píxel.

## Bit Por Píxel = 32

El formato de 32 bits es parecido al anterior de 24 bits, nuevamente cada byte almacena la información de cada color, rojo, verde y azul respectivamente. El cuarto byte nos da información de cuan transparente es cada píxel en relación al fondo.

### 2.2.3 Mostrar Imagen

Como se mencionó anteriormente, hay dos maneras para “pintar” o simplemente visualizar las imágenes en la pantalla. La elección de una sobre la otra depende del formato de la imagen, es decir, si es una imagen de un 1 BBP o no, (recordemos que todas las imágenes que no sean de 1 BPP son convertidas a imágenes de 24 BPP) por ésta razón muchas de las rutinas del programa además de la de pintar, poseen una “etiqueta” que hace ésta distinción, por ejemplo:

```
Public Sub BMP_Pintar_1()
```

```
Public Sub BMP_Pintar_24()
```

A grandes rasgos las operaciones que se realizan en esas rutinas son parecidas. Para imágenes binarias el valor de un píxel de coordenadas “x”, “y” puede estar en “0” o “255”, negro o blanco respectivamente. En cambio, en imágenes mayores a 1 BPP se tendrán valores desde “0” hasta “255” y como se tienen tres planos de colores, surge por lo tanto, una coordenada adicional, que llamaremos “z” y que está asociada con cada una de las componentes rojo, verde y azul.

Entonces a grosso modo, independientemente de cual sea la etiqueta: “1” o “24”, en las rutina de pintar se crea una ventana con las características de la imagen de origen, ésta ventana es copia de un formulario-ventana pre establecido en el programa. El Form1 es el patrón de referencias para las imágenes de 1 bit y el Form2 por su parte es la referencia para imágenes de 24 bits. Estas ventanas son creadas con ciertos atributos de la imagen, para que al procesar la imagen sea mucho más sencilla y coherente la manipulación de la información.

Rutina: “pintar\_1”

En esta rutina se crea una ventana como copia del formulario Form1, ésta copia se crea de tal manera que sea visible, con el nombre de la imagen y en ella se hacen llamados a unas subrutinas que guardan información sobre la matriz que posee el color y el encabezado de la imagen, por último se establecen como ancho y alto de esta ventana iguales a los de la imagen.

Rutina: “pintar\_24”

En esta rutina se realizan las mismas operaciones que en la rutina Pintar\_1, la diferencia es que la ventana que se crea al principio de la rutina es copia del formulario Form2, y ésta a su vez como se comentó anteriormente posee características de las imágenes de 24 bits.

## **2.3 Almacenamiento de la Imagen**

Es conveniente poder recuperar los efectos producidos por las operaciones morfológicas sobre las imágenes que se opera, en otras palabras, guardar los cambios realizados. La forma en que se guarde una imagen dependerá si es una imagen binaria o no.

La rutina de “*Guardar Imagen*” lee el encabezado de la imagen y luego discrimina la profundidad de bits de dicha imagen. Todas las imágenes que no sean de 1 bit son guardadas como imágenes de 24 bits, ésta operación es realizada de esta manera para facilitar su manipulación, y ser consecuente con el patrón de operaciones morfológicas creadas, unas operan sobre imágenes binarias y otras sobre imágenes de 24 bits.

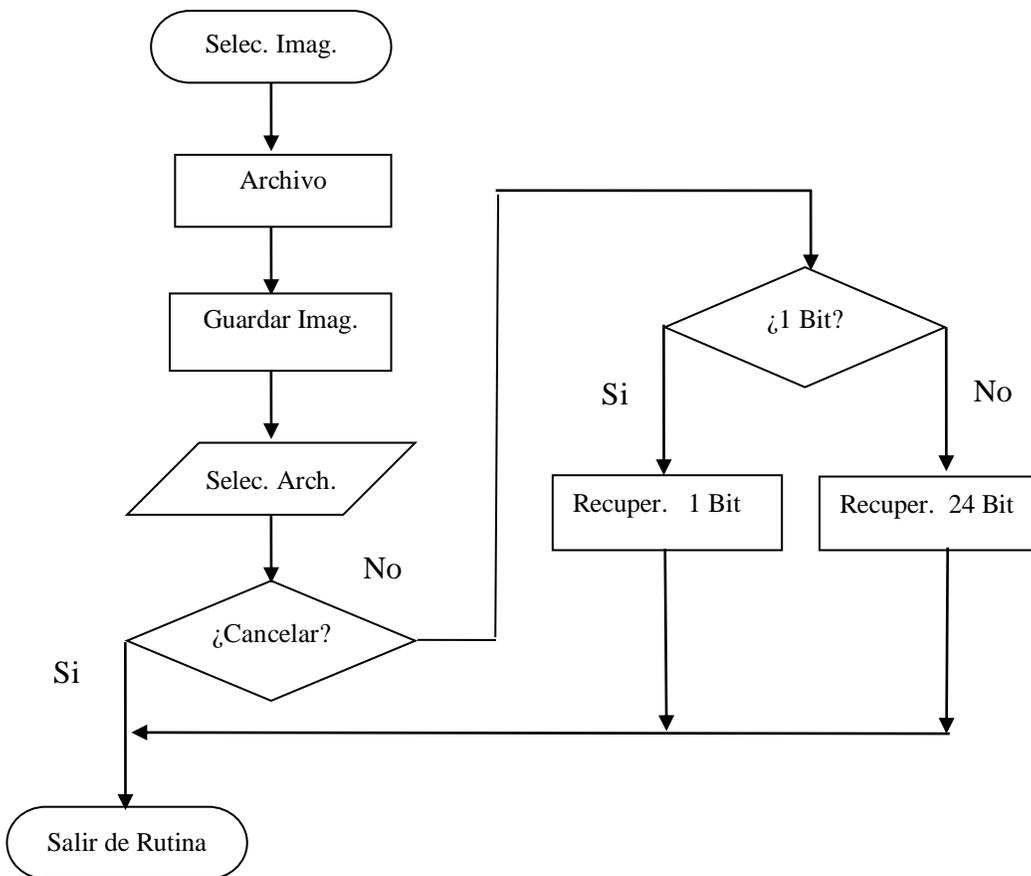


Figura 2.5 Diagrama de Flujo: Guardar Imagen

## 2.4 Propiedades

Cada imagen posee la información de sus atributos en su encabezado, esa información es importante para poder abrir y procesar dicha imagen. El usuario puede obtenerla seleccionando una imagen que esté abierta y presionando la opción de *Propiedades* allí se mostrará un cuadro con la información presente en el encabezado. En la Figura 2.4 se muestra un cuadro con las propiedades de la imagen 3x13.

La rutina de propiedades recupera el valor de los atributos presentes en cada imagen (el encabezado), ellos fueron guardados en la imagen justo antes de ser mostrada en el monitor.

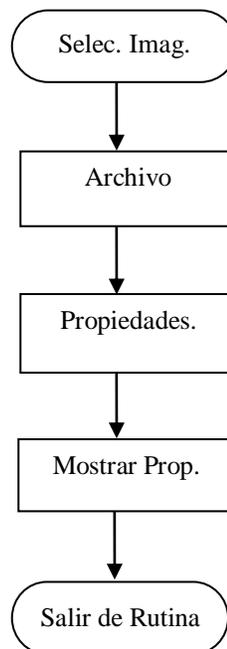


Figura 2.6 Diagrama de Flujo: Propiedades

## 2.5 Salir del Programa

La opción *Salir* permite al usuario cerrar el programa automáticamente. En esta opción se llama a una rutina que muestra en pantalla el siguiente mensaje: "*¿Desea cerrar Software PDI?*", de ser afirmativa la respuesta se ejecuta la instrucción "END" que se encarga de cerrar el programa.

## **2.6 Ajuste de la Imagen**

Los ajustes son las operaciones de pre procesamiento que acondicionan la imagen antes de aplicarle las operaciones de morfología matemática y ellos son brillo y contraste. En ocasiones es conveniente aumentar o disminuir el brillo-contraste de una imagen para resaltar aspectos que no se logran apreciar. Las imágenes de 1 bit presentan brillo y contraste máximo por poseer valores restringidos a los extremos “0” y “255”.

### **2.6.1 Brillo de la Imagen**

Para modificar el brillo de una imagen se añade una constante al valor de cada píxel, si se desea aumentar el brillo el valor de la constante será positivo, si en vez se desea disminuirlo el valor será negativo. Cuando se aumenta el brillo la imagen se aclara porque el valor de cada píxel tiende a “255”, si por el contrario disminuimos el brillo la imagen se oscurece porque el valor de cada pixel tiende a “0”.

### **2.6.2 Contraste de la Imagen**

Para modificar el contraste de una imagen, primero se promedia el valor de cada pixel en cada uno de los planos RGB, obteniéndose así un promedio por plano. En cada plano se sustrae el valor de su promedio a cada pixel respectivamente y luego se multiplica cada pixel por una constante, si la constante es mayor que cero el contraste aumentará si por el contrario es menor que cero el contraste disminuirá, esta acción producirá que los valores más altos tiendan a ser más claros y los valores más bajos tiendan a ser más oscuros. Al finalizar se suma el promedio sustraído anteriormente a cada uno de los planos. Realizar este proceso de esta manera permite modificar el contraste sin afectar el brillo de la imagen.

## **2.7 Convertir Imagen**

En algunos casos es necesario hacer unas modificar la imagen, si ella está a color transfórmala a escala de grises, si esta en gris o color transformarla a blanco y negro, o invertir (complementar) una imagen binaria blanco y negro.

### **2.7.1 Imagen: Color a Escala Grises.**

En una imagen a color, un píxel está compuesto por la superposición de los tres colores elementales: rojo, verde y azul, si los promediamos obtendremos un número, ese valor debe ser colocado en el pixel en cuestión. Si se repite este proceso para cada pixel de la imagen el resultado será una imagen en escala de grises.

### **2.7.2 Imagen: Escala de Grises a Blanco y Negro**

La transformación de una imagen escala de grises o color<sup>6</sup> a una imagen blanco y negro es por medio a un parámetro que fijamos previamente llamado treshold. Ese parámetro de referencia nos permite determinar si un píxel se convertirá en parte del objeto o del fondo en la imagen blanco y negro. En el programa, el treshold empieza en 128, si el valor de un píxel supera el treshold, ese pixel será colocado en “255” de lo contrario en “0”. Por último, el resultado es guardado como una imagen binaria.

### **2.7.3 Imagen: Invertir Blanco y Negro**

Invertir una imagen blanco negro es un proceso simple, se lee la información del color de cada píxel, si el valor del píxel es “0” es intercambiado por “255” y viceversa.

---

<sup>6</sup> La imagen es convertida automáticamente a una en escala de grises.

## 2.8 Operaciones Morfológicas

Las operaciones morfológicas son la parte esencial del programa PDI, ellas fueron implementadas comenzando con las más básicas: erosión y dilatación, y a partir de estas se hicieron otras operaciones más complejas como: apertura, cierre, filtros morfológicos y detección de bordes.

Recordemos que anteriormente se explicó que a la hora de mostrar las imágenes con el programa PDI tendremos las binarias y las de 24 bits, de igual manera las operaciones morfológicas se aplicarán tanto a imágenes binarias como a las de 24 bits. A continuación se explica cómo se implementaron las operaciones básicas.

### 2.8.1 Erosión

Para implementar la operación de erosión disponemos de dos rutinas, sobre imágenes binarias utilizamos la rutina: “erosionar\_1” y sobre imágenes en escala de grises o color utilizamos la rutina: “erosionar\_24”.

Rutina: “erosionar \_1”

Esta rutina utiliza tres matrices, una contiene la información de la imagen a erosionar, otra la del EE y la última contendrá la información de la imagen una vez erosionada. Como se explicó en el capítulo anterior el origen del EE recorre cada píxel de la imagen, determinando en cada paso si todos sus elementados están incluidos en ella, si se satisface esta condición el pixel formará parte de la imagen erosionada, de lo contrario el pixel pasará a formar parte del fondo.

Partiendo de lo mencionado anteriormente, asumiremos que el resultado de la erosión sobre un pixel se mantendrá en “1”, mientras no se consiga un punto del EE que no se encuentre contenido en la imagen, en cuyo caso tomará entonces el valor “0”. Para ello haremos uso de una variable que denominaremos “ero” y que inicialmente valdrá “1”, su valor final dependerá de la relación entre el EE y la imagen. El planteamiento anterior se resume en la siguiente tabla:

<i>Imagen</i>	<i>EE</i>	<i>Res.</i>
0	0	1
0	1	0
1	0	1
1	1	1

**Imagen:** “0” forma parte del fondo. “1” forma parte de la imagen.

**EE:** “0” no forma parte del EE. “1” forma parte del EE.

**Res:** Resultado.

Tabla 2 Relación entre un pixel de la Imagen y un punto del EE para determinar la erosión binaria.

Como lo mencionamos en la teoría, hacemos uso de una matriz para guardar la información del EE, y recordemos además que sólo los puntos que tienen valor “1” forman parte de él. El origen del EE recorre secuencialmente todos los puntos de la imagen, en cada ubicación se chequea cada uno de los puntos del EE con el pixel correspondiente de la imagen. Como podemos apreciar en la Tabla 2 el único caso en el cual el resultado es “0” es porque el punto del EE vale “1” y el pixel de la imagen que se corresponde con él vale “0”, es decir, el punto del EE no está incluido en la imagen. En cualquier otro caso el resultado valdrá “1”, ya sea porque tanto el punto del EE como la imagen valen “1” o porque el punto del EE vale “0” y no es considerado.

El comportamiento definido en la Tabla 2 lo implementaremos como una función lógica, cuyo resultado es el siguiente:

$$\text{Res} = \overline{EE} + \text{Imagen}$$

A continuación se muestra como se utilizó este resultado para implementar la operación de erosión binaria en el programa PDI:

```

For Y = 0 To BMP_encabezado.BMP_Height - 1
  For X = 0 To BMP_encabezado.BMP_Width - 1

    ero = 1

    For ye = 0 To ee_alto - 1
      For xe = 0 To ee_ancho - 1

        Xn = (X + xe - Round((ee_ancho - 1) / CDb(2) + 0.1))
        Yn = (Y + ye - Round((ee_alto - 1) / CDb(2) + 0.1))

        If (((Xn >= 0) And (Xn <= (BMP_encabezado.BMP_Width - 1))) And ((Yn >= 0)...
          ... And (Yn <= (BMP_encabezado.BMP_Height - 1)))) Then

          ero = ero And ((Not (ee(xe, ye)) Or Matrix_Neo(Xn, Yn)))

        End If

      Next xe
    Next ye

    Imagen_1(X, Y) = ero

  Next X
Next Y

```

**ee():** Matriz que contiene el Elemento Estructurante.

**Matrix\_Neo():** Matriz que posee la información de la imagen original.

**Imagen\_1():** Matriz que posee la información de la imagen erosionada.

Una vez finalizado el bucle “For” de la variable “ye”, el valor de “ero” se guarda en la matriz imagen, en las mismas coordenadas del pixel de la imagen original donde se ha ubicado el origen del EE. Este procedimiento se realiza para cada uno de los pixeles de la imagen original.

## Rutina: “erosionar \_24”

Al igual que en la erosión binaria en esta rutina se utilizan tres matrices, una contiene la información de la imagen a erosionar, otra la del EE y la última contendrá la información de la imagen una vez erosionada. Recordemos que al utilizar el modelo RGB tendremos tres planos (rojo, verde y azul) y por lo tanto, la operación de erosión se debe aplicar a cada uno de ellos.

Para imágenes en escala de grises o color, al realizar la operación de erosión sobre un pixel, el resultado será un valor entre “0” y “255”, ese número es el mínimo valor proveniente de restar punto a punto el EE a la imagen en la ubicación establecida. De igual manera que en la erosión binaria, el origen del EE recorre secuencialmente todos los puntos de la imagen, en cada ubicación se resta cada uno de los puntos del EE a los pixeles correspondientes de la imagen, quedándonos al final con el valor más pequeño de todas las sustracciones. El resultado se coloca en la matriz que contendrá la información de la imagen erosionada, en las mismas coordenadas del punto seleccionado en la imagen original.

Partiendo de lo mencionado anteriormente, asumiremos que inicialmente el resultado de la resta es “255”, para ello usaremos una variable “ero” para cada uno de los planos, el procedimiento consiste en hacer una comparación entre el valor de “ero” y el de cada sustracción, cada vez que se obtenga un valor más pequeño que el contenido en “ero”, éste reemplazará el valor existente en dicha variable. Finalmente el valor de “ero” se coloca en la matriz de la imagen erosionada, en las coordenadas (x,y) establecidas.

A continuación se muestra como se implementó la erosión en escala de grises o color en el programa PDI:

```

For Y = 0 To BMP_encabezado.BMP_Height - 1
  For X = 0 To BMP_encabezado.BMP_Width - 1

    ero_0 = 255
    ero_1 = 255
    ero_2 = 255

    For ye = 0 To ee_alto - 1
      For xe = 0 To ee_ancho - 1

        Xn = (X + xe - Round((ee_ancho - 1) / CDBl(2) + 0.1))
        Yn = (Y + ye - Round((ee_alto - 1) / CDBl(2) + 0.1))

        If (((Xn >= 0) And (Xn <= (BMP_encabezado.BMP_Width - 1))) And ((Yn >= 0)...
          ...And (Yn <= (BMP_encabezado.BMP_Height - 1)))) Then

          D0 = Matrix_Neo(Xn, Yn, 0) - CInt(ee(xe, ye))
          D1 = Matrix_Neo(Xn, Yn, 1) - CInt(ee(xe, ye))
          D2 = Matrix_Neo(Xn, Yn, 2) - CInt(ee(xe, ye))

          If D0 < 0 Then D0 = 0
          If D1 < 0 Then D1 = 0
          If D2 < 0 Then D2 = 0

          If (((ee(xe, ye)) > 0) And (D0 < ero_0)) Then ero_0 = D0
          If (((ee(xe, ye)) > 0) And (D1 < ero_1)) Then ero_1 = D1
          If (((ee(xe, ye)) > 0) And (D2 < ero_2)) Then ero_2 = D2

        End If

      Next xe
    Next ye

    Imagen_24(X, Y, 0) = ero_0
    Imagen_24(X, Y, 1) = ero_1
    Imagen_24(X, Y, 2) = ero_2

  Next X
Next Y

```

**ee()**: Matriz que contiene el Elemento Estructurante

**Matrix\_Neo()**: Matriz que posee la información de la imagen original.

**Imagen\_24()**: Matriz que posee la información de la imagen erosionada.

Una vez finalizado el bucle “For” de la variable “ye”, los valores de “ero” se guardan en la matriz imagen en su plano correspondiente, en las mismas coordenadas del pixel de la imagen original donde se ha ubicado el origen del EE. Este procedimiento se realiza para cada uno de los pixeles de la imagen original.

## 2.8.2 Dilatación

Para implementar la operación de dilatación disponemos de dos rutinas, sobre imágenes binarias utilizamos la rutina: “dilatar\_1” y sobre imágenes en escala de grises o color utilizamos la rutina: “dilatar\_24”.

Rutina: “dilatar \_1”

Esta rutina utiliza tres matrices, una contiene la información de la imagen a dilatar, otra la del EE y la última contendrá la información de la imagen una vez dilatada. Como se explicó en el capítulo anterior el origen del EE reflejado recorre cada píxel de la imagen, determinando en cada paso si al menos un punto del EE está contenido en ella, si se satisface esta condición el píxel formará parte de la imagen dilatada, de lo contrario el píxel pasará a formar parte del fondo.

Al reflejar un EE definido en una matriz con “n” filas y “m” columnas (n,m), el número de fila y columnas de la matriz para EE reflejado serán los mismos que posea la matriz original. Para reflejar un EE, se realizará una inversión en los valores de los índices de la matriz, tanto para las filas como las columnas, es decir, el primer elemento de la matriz del EE reflejado  $EE_R(0,0)$  se corresponde con el último elemento de la matriz del EE, es decir,  $EE_R(0,0) = EE(n,m)$  y en general  $EE_R(i, j) = EE(n - i, m - j)$  con  $0 \leq i \leq n$ ,  $0 \leq j \leq m$ .

Volviendo con la implementación de la dilatación binaria, asumiremos que el resultado sobre un píxel cualquiera se mantendrá en “0”, mientras no se consiga un punto del EE que se encuentre contenido en la imagen, en cuyo caso tomará entonces el valor “1”. Para ello haremos uso de una variable que denominaremos “dila” y que inicialmente valdrá “0”, su valor final dependerá de la relación entre el EE reflejado y la imagen. El planteamiento anterior se resume en la siguiente tabla:

<i>Imagen</i>	<i>EE</i>	<i>Res.</i>
0	0	0
0	1	0
1	0	0
1	1	1

**Imagen:** “0” forma parte del fondo. “1” forma parte de la imagen.

**EE:** “0” no forma parte del EE. “1” forma parte del EE.

**Res:** Resultado.

Tabla 3 Relación entre un pixel de la Imagen y un punto del EE para determinar la dilatación binaria.

Al igual que en la erosión binaria, el origen del EE (reflejado) recorre secuencialmente todos los puntos de la imagen, en cada ubicación se chequea cada uno de los puntos del EE con el pixel correspondiente de la imagen. Como podemos apreciar en la Tabla 3 el único caso en el cual el resultado es “1” es porque el punto del EE vale “1” y el pixel de la imagen que se corresponde con el también vale “1”, es decir, el punto del EE está incluido en la imagen. En cualquier otro caso el resultado valdrá “0”, ya sea porque imagen vale “0” o porque el punto del EE vale “0” y no es considerado.

El comportamiento definido en la Tabla 3 lo implementaremos como una función lógica, cuyo resultado es el siguiente:

$$\text{Res} = EE \cdot \text{Imagen}$$

A continuación se muestra como se utilizó este resultado para implementar la operación dilatación binaria en el programa PDI:

```

For Y = 0 To BMP_encabezado.BMP_Height - 1
  For X = 0 To BMP_encabezado.BMP_Width - 1

    dila = 1

    For ye = 0 To ee_alto - 1
      For xe = 0 To ee_ancho - 1

        Xn = (X + xe - Round((ee_ancho - 1) / CDbI(2) + 0.1))
        Yn = (Y + ye - Round((ee_alto - 1) / CDbI(2) + 0.1))

        If (((Xn >= 0) And (Xn <= (BMP_encabezado.BMP_Width - 1))) And ((Yn >= 0)...
        ... And (Yn <= (BMP_encabezado.BMP_Height - 1)))) Then

          dila = dila Or ((ee(xe, ye)) And Matrix_Neo(Xn, Yn))

        End If

      Next xe
    Next ye

    Imagen_1(X, Y) = dila

  Next X
Next Y

```

**ee():** Matriz que contiene el Elemento Estructurante.

**Matrix\_Neo():** Matriz que posee la información de la imagen original.

**Imagen\_1():** Matriz que posee la información de la imagen dilatada.

Una vez finalizado el bucle “For” de la variable “ye”, el valor de “dila” se guarda en la matriz imagen, en las mismas coordenadas del pixel de la imagen original donde se ha ubicado el origen del EE. Este procedimiento se realiza para cada uno de los pixeles de la imagen original.

## Rutina: “dilatar \_24”

Igual que en las operaciones anteriores, en ésta rutina se utilizan tres matrices, una contiene la información de la imagen a dilatar, otra la del EE y la última contendrá la información de la imagen una vez dilatada. Recordemos nuevamente que al utilizar el modelo RGB tendremos tres planos (rojo, verde y azul) y por lo tanto, la operación de dilatación se debe aplicar a cada uno de ellos.

Para imágenes en escala de grises o color, al realizar la operación de dilatación sobre un pixel, el resultado será un valor entre “0” y “255”, ese número es el máximo valor proveniente de sumar punto a punto el EE y la imagen en la ubicación establecida. De igual manera que en la dilatación binaria el origen del EE recorre secuencialmente todos los puntos de la imagen, en cada ubicación se suma cada uno de los puntos del EE a los pixeles correspondientes de la imagen, quedándonos al final con el valor más grande de todas las adiciones. El resultado se coloca en la matriz que contendrá la información de la imagen dilatada, en las mismas coordenadas del punto seleccionado en la imagen original.

Partiendo de lo mencionado anteriormente, asumiremos que inicialmente el resultado de la suma es “0”, para ello usaremos una variable “dila” para cada uno de los planos, el procedimiento consiste en hacer una comparación entre el valor de “dila” y el de cada adición, cada vez que se obtenga un valor más grande que el contenido en “dila”, éste reemplazará el valor existente en dicha variable. Finalmente el valor de “dila” se coloca en la matriz de la imagen dilatada, en las coordenadas (x,y) establecidas.

A continuación se muestra como se implementó la dilatación en escala de grises o color en el programa PDI:

```

For Y = 0 To BMP_encabezado.BMP_Height - 1
  For X = 0 To BMP_encabezado.BMP_Width - 1

    dila_0 = 255
    dila_1 = 255
    dila_2 = 255

    For ye = 0 To ee_alto - 1
      For xe = 0 To ee_ancho - 1

        Xn = (X + xe - Round((ee_ancho - 1) / CDbl(2) + 0.1))
        Yn = (Y + ye - Round((ee_alto - 1) / CDbl(2) + 0.1))

        If (((Xn >= 0) And (Xn <= (BMP_encabezado.BMP_Width - 1))) And ((Yn >= 0)...
          ...And (Yn <= (BMP_encabezado.BMP_Height - 1)))) Then

          D0 = Matrix_Neo(Xn, Yn, 0) + CInt(ee(xe, ye))
          D1 = Matrix_Neo(Xn, Yn, 1) + CInt(ee(xe, ye))
          D2 = Matrix_Neo(Xn, Yn, 2) + CInt(ee(xe, ye))

          If D0 > 255 Then D0 = 255
          If D1 > 255 Then D1 = 255
          If D2 > 255 Then D2 = 255

          If (((ee(xe, ye)) > 0) And (D0 > dila_0)) Then dila_0 = D0
          If (((ee(xe, ye)) > 0) And (D1 > dila_1)) Then dila_1 = D1
          If (((ee(xe, ye)) > 0) And (D2 > dila_2)) Then dila_2 = D2

        End If

      Next xe
    Next ye

    Imagen_24(X, Y, 0) = dila_0
    Imagen_24(X, Y, 1) = dila_1
    Imagen_24(X, Y, 2) = dila_2

  Next X
Next Y

```

**ee()**: Matriz que contiene el Elemento Estructurante

**Matrix\_Neo()**: Matriz que posee la información de la imagen original.

**Imagen\_24()**: Matriz que posee la información de la imagen dilatada.

Una vez finalizado el bucle “For” de la variable “ye”, los valores de “dila” se guardan en la matriz imagen en su plano correspondiente, en las mismas coordenadas del pixel de la imagen original donde se ha ubicado el origen del EE. Este procedimiento se realiza para cada uno de los pixeles de la imagen original.

## **2.9 Sistema de Ayuda**

Uno de los Objetivos de este trabajo es proporcionar suficiente información de ayuda que permita al usuario del programa PDI aprender a utilizarlo, contando además con información teórica sobre imágenes digitales y sobre como fueron implementadas las rutinas en el programa. La ayuda está dividida en tres secciones: *Ayuda Teórica*, *Ayuda Sobre el Programa* y el *Código Fuente Comentado*.

### **2.9.1 Ayuda Teórica**

En la *Ayuda Teórica* se encuentra los conceptos y definiciones en los que se fundamentan las rutinas y operaciones que se realizan en el programa. La *Ayuda Teórica* es prácticamente un resumen del MARCO TEORICO contenido en éste documento. La *Ayuda Teórica* representa una herramienta o un soporte a la mano del usuario para consultar rápidamente sobre:

- Imágenes con formato BMP
- Pre Procesamiento de Imágenes
- Morfología Matemática
- Aplicaciones.

### **2.9.2 Ayuda Sobre el Programa**

Para crear el programa PDI es necesario poner en práctica, toda la información referente a la teoría explicada en el MARCO TEORICO, el mecanismo para realizar el programa de PDI depende muchísimo de cómo está estructurado y de cómo funciona el Visual Basic 6.0, ya que hay varias rutinas que poseen peculiaridades o detalles que surgen debido a las características del entorno de programación. En la

*Ayuda Sobre el Programa* se explica su utilización y funcionamiento a nivel de programación.

### **2.9.3 Código fuente**

Un recurso importante para el usuario es el código fuente del programa, éste cuenta con comentarios pertinentes en las rutinas, que complementan la información de cómo se implementaron las diferentes funciones y operaciones del programa PDI, aquí puede apreciarse como se implementaron a partir de las rutinas de erosión y dilatación, las operaciones más complejas como: apertura, cierre, filtros morfológicos, detección de bordes y granulometría.

## Resultados y Conclusiones

Se elaboró un programa para procesar Imágenes Digitales, el cual permite abrir imágenes de cualquier profundidad en formato BMP, las procesa y las guarda como imágenes digitales en blanco y negro (1 bit) o imágenes en escala de grises o color (24 bits). El programa puede hacer ajustes al brillo y contraste de las imágenes, transformar una imagen a color a escala de grises, transformar una imagen en escala de grises a blanco y negro, e invertir (complementar) una imagen blanco y negro. El programa PDI realiza operaciones morfológicas tales como: erosión, dilatación, apertura, cierre, filtros morfológicos, detección de bordes y granulometría. El programa PDI es un recurso importante que será utilizado como complemento en la formación académica de los alumnos en la Orientación de Instrumentación.

Las rutinas de las operaciones morfológicas del programa PDI, fueron escritas en un módulo externo al programa principal, con la intención de facilitar el proceso de creación de una Librería de Enlace Dinámico (Dynamic Link Library, "dll") que permitiría el uso de las operaciones morfológicas desde cualquier otro programa o aplicación.

La operación de granulometría contabiliza objetos de diferentes tamaños presentes en una imagen binaria y muestra el resultado a través de un gráfico de barras. Se comprobó su buen funcionamiento hasta con EE tan pequeños como 3 píxeles de diámetro.

El programa PDI cuenta además con información de ayuda que ofrece al usuario una guía explicativa de las rutinas y conceptos teóricos sobre imágenes digitales y Morfología Matemática. En la ayuda también se encuentra disponible el código fuente del programa debidamente comentado.

## Bibliografía

Ortiz, Francisco. (2002) *“Procesamiento morfológico de imágenes a color. Aplicación a la reconstrucción geodésica”*. Universidad de Alicante.

Gonzalez R.C., Woods R.E. (1996) *“Tratamiento Digital de Imágenes”*. Adison-Wealwey Publishing Co.

Wouters, Win. (2000) *“BMP Format”*. Versión V1.1.

Myler, Harley R. (1998) *“UCFImage”*. University of Central Florida.

Dougherty, Edgard R. (1992) *“An Introduction to Morphological Image Processing”*. SPI – The International Society for Optical Engineering.

Myler, Harley R., Weeks Arthur R. (1993) *“Computer Imaging Recipes in C”*. P T R Prentice Hall (Englewood Cliffs, N.J).

Universidad Nacional de Quilmes (2005) *“Ing. en Automatización y Control Industrial”*. Cátedra: Visión Artificial.

Gamino, Arturo. (2005) *“Operaciones morfológicas rápidas por descomposición del elemento de estructura mediante discos”* Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional.

Sbert, Catalina. *“cap5.pdf”* dmi.uib.es 6/11/2008. 1 de mayo de 2014  
<http://dmi.uib.es/~catalina/docencia/PDS/cap5.pdf>