



# INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA

UNIDAD ZACATENCO

*“CARACTERIZACION DEL NIVEL DE LIQUIDO EN  
RECIPIENTES DE VIDRIO USANDO EL METODO DE  
ENERGIA Y DISTANCIAS EUCLIDIANAS EN IMÁGENES”*

## TESIS

QUE PARA OBTENER EL GRADO DE INGENIERO EN  
COMUNICACIONES Y ELECTRONICA

PRESENTA

PEDRO DE LA PEÑA OROZCO

ASESORES

DR. MARIO JIMENEZ HERNANDEZ

ING. GONZALO PEÑA LOPEZ

MEXICO D.F. 2018



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”**

**T E M A D E T E S I S**

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMUNICACIONES Y ELECTRÓNICA  
POR LA OPCIÓN DE TITULACIÓN TESIS Y EXAMEN ORAL INDIVIDUAL  
DEBERA (N) DESARROLLAR C. PEDRO DE LA PEÑA OROZCO

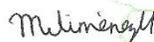
**“CARACTERIZACIÓN DEL NIVEL DE LÍQUIDO EN RECIPIENTES DE VIDRIO USANDO EL  
MÉTODO DE ENERGÍA Y DISTANCIAS EUCLIDIANAS EN IMÁGENES”**

IMPLEMENTACIÓN DE UNA INTERFAZ GRÁFICA PARA EL ANÁLISIS DEL LLENADO DE ENVASES DE  
CRISTAL UTILIZANDO EL MÉTODO DE ENERGÍA Y DISTANCIAS EUCLIDIANAS EN EL ANÁLISIS DE  
IMÁGENES CON FORMATO BMP.

- ❖ INTRODUCCIÓN
- ❖ ESTADO DEL ARTE
- ❖ DESARROLLO E IMPLEMENTACIÓN DE HARDWARE Y SOFTWARE
- ❖ ANÁLISIS DE RESULTADOS
- ❖ CONCLUSIONES
- ❖ APÉNDICES
- ❖ BIBLIOGRAFÍA

CIUDAD DE MÉXICO, A 25 DE ENERO DEL 2018.

**ASESORES**

  
DR. MARIO JIMÉNEZ HERNÁNDEZ

  
ING. GONZALO PEÑA LÓPEZ

  
R. SALVADOR RICARDO MENESES GONZÁLEZ  
JEFE DEL DEPARTAMENTO DE  
INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA

Instituto Politécnico Nacional

P r e s e n t e

Bajo protesta de decir la verdad el que suscribe **Pedro De La Peña Orozco**, manifiesto ser autor y titular de los derechos morales y patrimoniales de la obra titulada **“Caracterización del nivel de líquido en recipientes de vidrio usando el método de energía y distancias euclidianas en imágenes”**, en adelante **“La Tesis”** y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la ley federal del derecho de autor, otorgo a el **Instituto Politécnico Nacional**, en adelante **El IPN**, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales de un impreso y un disco únicamente para consulta de **“La Tesis”** por un periodo de **1 año** contando a partir de la fecha de la presente autorización, dicho periodo se renovara automáticamente en caso de no dar aviso a **El IPN** de su terminación.

En virtud de lo anterior, **EL IPN** deberá reconocer en todo momento mi calidad de autor de **“La Tesis”**.

Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de **“La Tesis”**, manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de la **“La Tesis”**, por lo que deslindo de toda responsabilidad a **El IPN** en caso de que el contenido de la **“La Tesis”** o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

México, Ciudad de México, a 18 de abril de 2018.

Atentamente

  
Pedro De La Peña Orozco

# INDICE

INDICE .....	4
<b>INDICE DE FIGURAS</b> .....	6
<b>INDICE DE TABLAS</b> .....	8
Resumen .....	9
Capítulo 1  Introducción.....	11
1.1 Planteamiento del problema .....	11
1.2 Propuesta de solución .....	12
1.3 Justificación.....	12
1.4 Hipótesis .....	13
1.5 Objetivo General.....	13
1.6 Objetivos Específicos .....	13
1.7 Alcances y Limitaciones .....	14
Capítulo 2 Antecedentes Y Estado Del Arte .....	15
2.1 Fundamentos de la Imagen digital .....	15
2.2 Compresión de Imágenes .....	19
2.4 Formato BMP .....	25
2.5 Energía de una imagen .....	30
2.6 Distancias Euclidianas .....	31
2.7 Métodos alternos para el procesamiento digital de imágenes (artículos) .....	35
2.7.1- <i>“Método de umbralado por determinación         de pixeles para la segmentación de elementos en</i>	

<i>imágenes aplicado al análisis de problemas en la industria</i> .....	35
2.7.2 “Método de Binarización de Imágenes” .....	37
2.7.3 Equipos de inspección de envases.....	38
2.7.4 “Sistemas de inspección por rayos-x de envases de vidrio” .....	38
Capítulo 3 Implementación y Desarrollo de Software Y Hardware .....	40
3.1 Dispositivo de captura de las imágenes de envases de vidrio.....	40
3.2 Comunicación de la Cámara Web con la PC.....	42
3.3 Desarrollo del prototipo .....	42
3.4 Desarrollo del hardware .....	43
3.4.1 Diseño de la banda transportadora.....	43
3.5 Diseño de la Interfaz Gráfica.....	53
3.6 Programación del sistema. ....	59
Capítulo 4 Análisis de Resultados .....	68
4.1 Resultados para una línea de 10 envases llenos y 10 envases vacíos .....	68
4.2 Interpretación de los resultados.....	73
Capítulo 5 Conclusiones.....	79
Capítulo 6 Apéndices .....	80
Capítulo 7 Bibliografía.....	93

## INDICE DE FIGURAS

Figura 2.1 Representación de una imagen digital [1]	15
Figura 2.2 Digitalizando una imagen [4]	17
Figura 2.3 Ejemplo de una imagen digital [5]	17
Figura 2.4 Componentes básicos del procesamiento de imágenes [6]	18
<b>Figura 2.5</b> Proceso de adquisición de la imagen [6]	19
Figura 2.6 Representación de formatos de imagen [8]	23
Figura 2.7 Intensidad en escala de grises para 8 bits [2]	23
Figura 2.8 Modelo de cubo unitario para representar formato RGB [10]	24
Figura 2.9. La resolución de una imagen nos indica el número de píxeles que caben en cada unidad de longitud. Es decir que, a mayor resolución e igual tamaño, mayor será la cantidad de píxeles y menor el tamaño de éstos y, por lo tanto, la imagen tendrá mayor nitidez. Al comparar las imágenes vemos que ambas tienen el mismo tamaño, pero el número de píxeles difiere una de la otra. La primera imagen tiene menor resolución que la segunda. [10]	27
En la <b>Figura 2.10</b> se muestra un ejemplo de un bitmap	30
<b>Figura 2.10</b> Ejemplo de un bitmap[12]	30
Figura 2.11 Distancia Euclidiana en dos dimensiones	33
<b>Figura 2.12</b> Distancia Euclidiana en 3 dimensiones.	33
Figura 2.13 Histograma ideal	34
<b>Figura 2.14</b> A split-beam (100W / 400W) x-ray system designed to inspect metal, glass and ceramic containers. [Mettler Toledo].	39
<b>Figura 3.1.</b> Cámara Web True Basix CW-1000	40
Figura 3.2. Imagen de un envase de vidrio vacío.	41
Figura 3.3. Imagen de un envase de vidrio lleno.	41
<b>Figura 3.4</b> Base de madera plana sujeta al soporte con tornillo	44
Figura 3.5 Soporte vertical con perforación de 0.635 cm	45
<b>Figura 3.6</b> Soporte vertical con los dos orificios de 0.635 cm y 0.833 cm de diámetro.	45
Figura 3.7 Montaje de los 4 soportes verticales a la base de madera plana	46
Figura 3.8 Rodillos ensamblados	46
<b>Figura 3.9</b> Rodillos ensamblados a los soportes verticales mediante los orificios de 0.635cm	47
<b>Figura 3.10</b> Motorreductor sujeto al soporte vertical con los engranes en sincronía para girar	47
<b>Figura 3.11.</b> Motorreductor B01 1:220, [ Robodacta]	48

<b>Figura 3.12</b> B01 1:220, [Robodacta]	49
<b>Figura 3.13</b> Cinta transportadora montada en los rodillos	50
<b>Figura 3.14</b> Banda transportadora con soportes para reforzar y soportar el peso.	50
<b>Figura 3.15</b> Piezas de acrílico montadas en el ángulo de aluminio	51
Figura 3.16 Acrílico montado en la banda transportadora con vista frontal y posterior	51
Figura 3.17 Soporte de metal sujetado a la banda transportadora	52
<b>Figura 3.18</b> Cámara montada al soporte de metal	52
<b>Figura 3.19</b> Diagrama eléctrico de la banda transportadora	53
<b>Figura 3.20</b> Prototipo terminado	53
<b>Figura 3.21</b> Diseño de la interfaz gráfica de usuario GUI *.fig.	54
<b>Figura 3.22</b> Interfaz gráfica de usuario GUI.	57
<b>Figura 3.23.</b> Diagrama de flujo del funcionamiento del sistema.	58
<b>Figura 3.24</b> Diagrama a bloques código para leer y guardar las matrices de las imágenes.	59
<b>Figura 3.25</b> Creación de un archivo de Excel que contiene en una hoja de cálculo los valores de las matrices de cada una de las imágenes.	60
<b>Figura 3.26</b> Lectura de las matrices de las imágenes en formato BMP.	60
<b>Figura 3.27</b> Diagrama a bloques y código de programa que lee la matriz de la imagen de envases llenos y saca la energía de la misma.	61
Figura 3.28 Diagrama a bloques y código de programa que lee la matriz de la imágenes de envases vacíos y saca la energía de la misma.	62
Figura 3.29. Captura de imagen	63
3.7 Proceso de Evaluación del Sistema	66
<b>Figura 4.1</b> Comportamiento de las energías de envases	69
Figura 4.2 Envase #1 lleno      Figura 4.3 Envase #2 lleno      Figura 4.4 Envase #3 lleno	70
Figura 4.5 Envase #4 lleno	70
Figura 4.8 Envase #7 lleno	70
Figura 4.11 Envase #10 lleno	70
Figura 4.12 Envase #1 vacío	71
Figura 4.15 Envase #4 vacío	72
Figura 4.18 Envase #7 vacío	72
<b>Figura 4.21</b> Envase #10 vacío	72
<b>Figura 4.23</b> Comportamiento de las energías de envases de prueba llenos	75
<b>Figura 4.24</b> Comportamiento de las energías de envases de prueba vacíos	76
<b>Figura 4.26</b> Energías muestra VS Energías pruebas en envases vacíos.	77

## **INDICE DE TABLAS**

Tabla 2.1 Contenido del bloque de la cabecera [12]_____	26
Tabla 2.2 Contenido del bloque de la cabecera de mapa de bits [12]_____	26
Tabla 2.3 Contenido de la paleta de colores de mapa de bits [12]_____	27
Tabla 2.4 Umbrales globales para cada método formato RGB_____	27
Tabla 3.1 Especificaciones técnicas del motorreductor B01 1:220, [Robodacta]_____	34
Tabla 4.1 Niveles de energía de 10 envases llenos_____	47
Tabla 4.2 Niveles de energía de 10 envases vacíos_____	67
Tabla 4.3 Niveles de energía de 10 envases de prueba llenos_____	70
Tabla 4.4 Niveles de energía de 10 envases de prueba vacíos_____	73
Tabla 4.5 Resultados prueba 1_____	77

# Resumen

La identificación del nivel de llenado en la industria es uno de los procesos más importantes, ya que forma una de las etapas en las que se tienen mayor inspección y en las que se tiene que cumplir un control de calidad total.

En este trabajo terminal, se presenta un prototipo de un sistema de verificación de nivel de llenado en envases de vidrio, este sistema su función principal es detectar si el envase está totalmente lleno o totalmente vacío, lo que nos permitirá descartar envases vacíos o que no estén al nivel establecido, se toman muestras que se guardan en una base de datos con las cuales el sistema comparará las muestras entrantes y determinará el nivel de llenado de cada una de éstas.

El hardware es una cámara web como elemento de captura de las imágenes de los frascos que están colocados en una banda transportadora, mientras que el software es un algoritmo desarrollado en la plataforma Matlab, el cual permite el procesamiento de las imágenes y la comparación de las mismas, así como el control de la banda para sincronizarlo con la captura de la cámara.

Los sistemas para evaluar y verificar el nivel de llenado de envases de vidrio, actualmente se basan en un método en el cual una persona encargada los observa a través de una pantalla con una luz intensa y el frasco lleva marcado un punto hasta el que se debe llenar, lo que se observa es que líquido este exactamente hasta ese punto.

El algoritmo para la verificación de nivel de llenado en envases, presentado en este trabajo, se basa en uno de los métodos de comparación de niveles de energía entre imágenes llamada Método de la Energía. Por lo que la propuesta en este trabajo es utilizar este

concepto para verificar el nivel de llenado en un envase por medio de la energía total que obtenemos de cada imagen. La medición se basa en la obtención de una matriz por pixel Y

una matriz total de cada imagen que nos da la energía total de la misma, la cual está en escala de grises en formato de ocho bits.

La matriz que se obtiene por cada pixel contiene valores de 0 a 255 que son los permitidos para ese formato, al obtener la matriz total de cada imagen, esta es comparada con el nivel de energía de la base de datos máximo y mínimo que determina si el envase está totalmente lleno o totalmente vacío aplicando el algoritmo matemático de la energía.

La etapa de verificación consistió en la comparación de 20 frascos con distintos niveles de llenado los cuales fueron colocados y transportados en la banda la cual lleva una sincronización con la cámara y al tomar la imagen esta es procesada y comparada con una base de datos previamente realizada, el algoritmo desarrollado compara el nivel de energía total de la muestra tomada y la compara con los niveles de energía establecidos y dependiendo del resultado lo evalúa en los rangos y determina el estado en el que se encuentra cada frasco entrante.

# Capítulo 1 Introducción

Actualmente, existen diferentes métodos de procesamiento y análisis de imagen como el método de segmentación, método de diferencia de energía, entre otros.

En este capítulo se describen las generalidades del método de diferencia de energía y el método de distancias euclidianas el cual se va a utilizar para los fines del proyecto. Dentro del mismo capítulo se muestra la justificación, hipótesis y objetivos general y específicos del presente proyecto.

## 1.1 Planteamiento del problema

El proceso de llenado de botellas con líquido, se lleva a cabo en una estación en la cual el líquido es bombeado al recipiente en cuestión desde el depósito, las botellas se llenan dependiendo del tamaño de la misma y el control se regula a través de una bomba de regulación continua y sensores de nivel, el nivel de llenado permanece constante para cada botella con el objetivo de optimizar la calidad de dicho proceso. El nivel de llenado se puede modificar manual o automáticamente dependiendo del sistema utilizado, a pesar del desarrollo tecnológico que se ha suscitado a través de los años, una maquina necesita de un mantenimiento periódico para que su funcionamiento siga siendo el adecuado, si no se le da el necesario aumenta el rango de falla que pudiera tener, tomando en cuenta que en las industrias es indispensable el mantenimiento a los equipos dado a que no se pueden permitir perdidas en la producción y aun cuando un equipo está en las mejores condiciones pueden existir diversas fallas, por diversas razones, para el objetivo de este estudio se enfocara en la falla en el nivel de llenado, durante este procedimiento es necesario que todos los envases de la línea de producción estén a un nivel de llenado óptimo para que se considere apto para el siguiente paso a seguir en proceso de producción, para ello es necesario que se verifique el nivel de llenado en cada envase, o en su defecto si el envase

está vacío, para esto es necesario tener un “inspeccionador” que realice este trabajo, con el objetivo de rechazar los envases que no cumplan con las especificaciones requeridas y que no se aumente la merma en la producción.

## 1.2 Propuesta de solución

En este trabajo se desarrolla un dispositivo que permite la verificación del nivel de llenado de envases de vidrio a partir de un algoritmo desarrollado basándonos en el método de la energía como medio de comparación.

A diferencia de otro tipo de sistemas que todavía utilizan operadores humanos para verificar el nivel de llenado, en este proyecto se utiliza un software que verifica automáticamente el nivel de cada envase entrante por medio de una cámara web como elemento de captura, obteniendo imágenes bidimensionales en escala de grises para su mejor análisis.

## 1.3 Justificación

Como se puede observar en los puntos anteriores, la industria de llenado de envases tiene a un empleado encargado de la supervisión de los envases y si el llenado de los mismos es óptimo para que se pueda conservar dentro de la línea de producción. Lo que sin duda es adecuado ya que a gran escala esto genera pérdidas económicas para la empresa y el control de calidad no cumpliría con las especificaciones requeridas.

Tomando en cuenta lo anterior, el sistema propuesto en esta investigación es una herramienta que permite una automatización dentro de este proceso, por lo que sin desplazar a la persona encargada de realizar esta operación, se haría más eficaz la revisión y la supervisión del mismo presentando resultados favorables.

En este proyecto se desarrolla un algoritmo que nos permite comparar la energía de envases entrantes en la línea de producción con niveles de energía previamente calculados.

#### 1.4 Hipótesis

Al tener diferentes niveles de energía en los pixeles, es posible apreciar cambios en la intensidad de cada uno de la imagen capturada mediante la cámara web, lo cual permite que cada uno de los envases sea identificado por medio de algoritmos matemáticos desarrollados para comparar las características con una base de datos y así se identifique el nivel de llenado de cada uno correctamente.

#### 1.5 Objetivo General

Implementación de una interfaz gráfica para el análisis del llenado de envases de cristal utilizando el método de energía y distancias euclidianas en el análisis de imágenes con formato BMP.

#### 1.6 Objetivos Específicos

- Diseño de un sistema para la adquisición de imágenes de frascos de vidrio Que permita su desplazamiento en el eje horizontal.
- Desarrollo de una interfaz gráfica de usuario para obtener imágenes en formato BMP a partir de web cámara.
- Desarrollo de los algoritmos de cálculo de energía bidimensional y de distancias euclidianas para la evaluación de las imágenes.
- Determinar los valores de umbral máximo y mínimo de una base de datos de imágenes de 20 frascos de vidrio diferentes.

- A partir del método de distancias euclidianas desarrollado caracterizar el llenado de frascos de vidrio.
- Realizar el análisis estadístico de los resultados.

### 1.7 Alcances y Limitaciones

#### Alcances:

- Desarrollo de un sistema para procesamiento de imagen que permita extraer características suficientes para tomar la decisión de descartar el envase.

#### Limitaciones:

- Funcionamiento en tiempo real
- Tiempo de análisis de cada una de las regiones de estudio
- Bmp 640 x480 a 8 bits en escala de grises

# Capítulo 2 Antecedentes Y Estado Del Arte

En este capítulo, se mencionan de forma general los métodos más utilizados para el procesamiento de imagen. Se hace hincapié en el método que se va a utilizar para la realización del proyecto para lograr la clasificación de los envases conforme a las condiciones de llenado.

## 2.1 Fundamentos de la Imagen digital

- Procesamiento digital de imágenes

El proceso digital de imágenes se puede ver como una función de 2 dimensiones, en donde  $x$  y  $y$  son coordenadas espaciales, en donde la amplitud de  $f$  para cualquier par de coordenadas  $(x, y)$  que son coordenadas espaciales y se refieren a la función que representa la intensidad o nivel de grises en ese punto. Cuando las coordenadas y la intensidad son valores finitos, cantidades discretas podemos llamar a la imagen una imagen digital. [1]

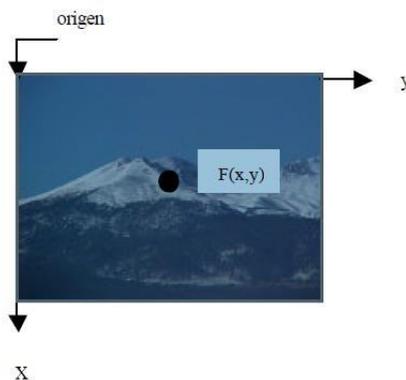


Figura 2.1 Representación de una imagen digital [1]

El campo del procesamiento digital de imagen se refiere a procesar imágenes digitales desde un computador. Nótese que una imagen digital está compuesta por un número finito de elementos, cada uno de estos tiene una localización y valor particular. Estos elementos son llamados elementos de imagen o píxeles.

Algunas de las operaciones principales en el procesamiento digital de imágenes son el escalamiento, la codificación, extracción de características, reconocimiento de patrones, entre otras. [3]

- Fundamentos de la imagen digital

La visión es el sentido más desarrollado que tenemos los seres humanos, no es sorprendente que las imágenes jueguen el rol más importante en la percepción humana. Sin embargo, tienen una visión limitada en la banda de espectro electromagnético, a diferencia de las máquinas que cubren todo ese espectro, desde el rango de gamma a las ondas de radio. Estas pueden operar en imágenes generadas por fuentes que los humanos no están acostumbrados a asociar con imágenes. Estas incluyen el ultrasonido, electrón microscopy, y el generador de imágenes por computadora.

Hay dos grandes grupos de imágenes digitales: las imágenes vectoriales y las imágenes de mapas de bits también llamadas imágenes pixelares, ráster o bitmap.

- Pixel

Un pixel es un término usado en la mayoría de widely para denotar los elementos de una imagen digital.

Una imagen digital se puede considerar como una matriz en cuyos renglones y columnas se identifica un punto de la imagen y los valores correspondientes de la matriz que identifican el nivel de gris en ese punto, como vimos anteriormente, son los *pixeles* [1].

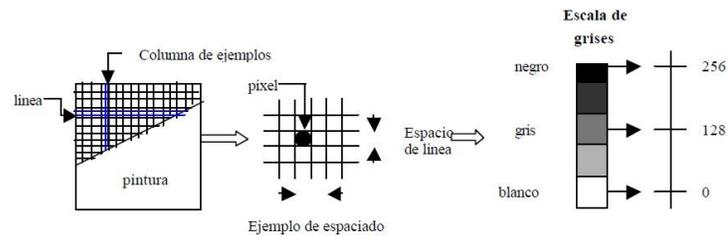


Figura 2.2 Digitalizando una imagen [4]

Matemáticamente, el *pixel* representa la intensidad o nivel de gris que se ha asociado a la imagen en la coordenada  $(m, n)$ . Por lo tanto, una representación matricial bidimensional de la intensidad de una imagen de  $m \times n$  elementos, donde  $m$  es el número de filas y  $n$  representa el número de columnas en una imagen está dada por,

$$f = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad (2.1)$$

Un sencillo ejemplo de una imagen digital la podemos observar en la **Figura 2.3**

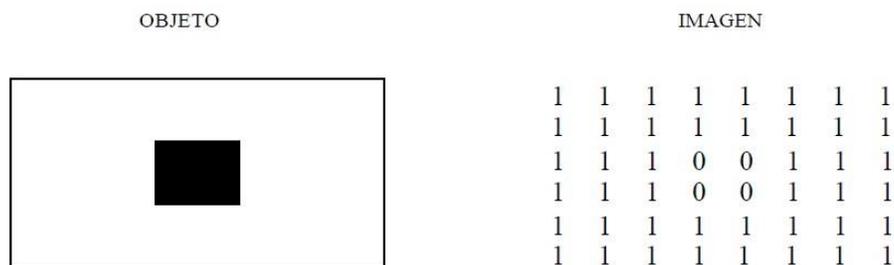
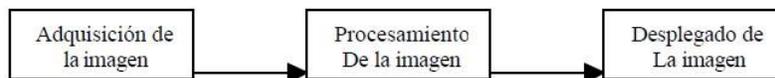


Figura 2.3 Ejemplo de una imagen digital [5]

El procesamiento de imágenes se puede representar en tres pasos [6]:

- **Adquisición de una imagen:** Es la conversión de una escena a la representación digital que puede ser procesada por una computadora. La adquisición de una imagen se puede llevar a cabo por medio de un sensor designado para ver una escena y proporcionar una representación digital de la misma.
- **Procesamiento de la imagen:** Proporciona procesamiento digital de una o más imágenes para producir un resultado esperado. El procesamiento puede extenderse del simple incremento de una imagen para mejorar el despliegue de los detalles de la escena, a un procesamiento más complejo que envuelve diferentes componentes de la imagen.
- **Despliegue de la imagen.** Permite que el observador pueda ver el producto resultante. Puede desplegarse, en el caso de las computadoras, en la pantalla

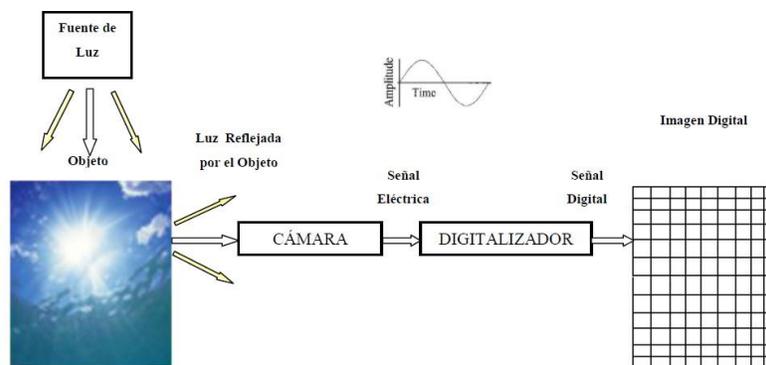


**Figura 2.4** Componentes básicos del procesamiento de imágenes [6]

En la adquisición de imágenes deben existir dos elementos básicos. El primero es algún dispositivo básico para captar la imagen, que sea sensible a una determinada banda del espectro de energía electromagnético como son las bandas de rayos-x, ultravioleta, visible o el infrarrojo, y que producen una señal eléctrica proporcional al nivel de energía sentido.

El segundo es el digitalizador que convierte la salida del dispositivo físico de sensado a forma digital. En esta categoría se agrupan las cámaras CCDs (Charge-Coupled Devices) que tienen la ventaja de la velocidad de captura de hasta 1/10,000 seg.

Las imágenes son representaciones de objetos, los cuales son sensados a través de su energía radiante, por ejemplo, la luz. La formación de una imagen requiere de una fuente de radiación, un objeto y un sistema de formación de imagen, como se muestra en la **Figura 2.5**.



**Figura 2.5** Proceso de adquisición de la imagen [6]

## 2.2 Compresión de Imágenes

La compresión de datos se refiere al proceso de reducir la cantidad de datos requeridos para representar una cantidad de información dada [1]. La compresión de imágenes se vuelve necesaria cuando deseamos calcular el número de bits por imagen que resulta de una estimación de muestras y conteo de algún esquema. Uno de los aspectos más importantes del almacenaje de imágenes es obtener una compresión eficiente.

La portabilidad de plataformas y ejecución son igualmente importantes para la selección de técnicas de compresión y descompresión de imágenes a ser empleadas. Las soluciones de compresión son más abundantes, debido a la implementación de estándares internacionales. En este contexto, **JPEG** ha evolucionado como la técnica para la compresión estándar. JPEG es más que un algoritmo para compresión de imágenes ya que contiene una arquitectura para un grupo de funciones de compresión de imágenes además de muchas capacidades que permiten que sea útil para un rango de aplicaciones que se refieran a compresión de imágenes. El grupo de datos de JPEG depende de lo que el *decoder* necesita para descomprimir el conjunto de datos. JPEG está relacionado con imágenes que contienen dos dimensiones, escala de grises y color [7].

Existen dos categorías de algoritmos de compresión de datos: *lossless* y *lossy*.

La técnica *lossless* también es conocida como compresión inversa. En esta técnica la imagen reconstruida después de comprimirla es numéricamente idéntica a la imagen original en una base de píxel por píxel. Mientras en la técnica *lossy*, también conocida como compresión irreversible, la imagen reconstruida contiene cierta degradación a la imagen original. Las técnicas *lossy*, causan una degradación de la imagen en cada paso de compresión y descompresión. La degradación es a menudo irreconocible, y depende de la razón de compresión que se elija. En general, las técnicas *lossy* proporcionan una compresión muy lejos de la ideal que las técnicas *lossless*. [7,9]

Otros métodos que podemos mencionar para comprimir imágenes, son MPEG, GIF, Wavelets entre otros, así como otras alternativas de reducir imágenes se basan en la segmentación como es el caso de *quadrees* y *octrees*.

La segmentación subdivide una imagen en partes u objetos. La segmentación se debe detener cuando los objetos de interés en la aplicación han sido identificados, es decir, hay que parar de segmentar de acuerdo a las necesidades del problema.

Aunque habrá que considerar que mientras más segmentada se encuentre una imagen, menos pérdidas de datos habrá al momento de tratarla. Si llegamos a segmentar una

imagen hasta su máxima resolución, dicho en otras palabras, hasta que la unidad mínima sea el píxel, más exacta será la representación de mi imagen.

### 2.3 Formatos de Imagen

- Formato de imágenes digitales

Las imágenes digitales se pueden guardar en distintos formatos. Cada uno se corresponde con una extensión específica del archivo que lo contiene. Los más utilizados en la actualidad son: BMP, GIF, JPG, TIF y PNG.[8]

#### **BMP (Bitmap = Mapa de bits)**

- Ha sido muy utilizado porque fue desarrollado para aplicaciones Windows.
- La imagen se forma mediante una parrilla de píxeles.
- El formato BMP no sufre pérdidas de calidad y por tanto resulta adecuado para guardar imágenes que se desean manipular posteriormente.
- Ventaja: Guarda gran cantidad de información de la imagen.
- Inconveniente: El archivo tiene un tamaño muy grande.

#### **GIF (Graphics Interchange Format = Formato de Intercambio Gráfico)**

- Ha sido diseñado específicamente para comprimir imágenes digitales.
- Reduce la paleta de colores a 256 colores como máximo (profundidad de color de 8 bits)
- Admite gamas de menor número de colores y esto permite optimizar el tamaño del archivo que contiene la imagen.
- Ventaja: Es un formato idóneo para publicar dibujos en la web.
- Inconveniente: No es recomendable para fotografías de cierta calidad ni originales ya que el color real o verdadero utiliza una paleta de más de 256 colores.

### **JPG-JPEG (Joint Photographic Experts Group = Grupo de Expertos Fotográficos Unidos)**

- A diferencia del formato GIF, admite una paleta de hasta 16 millones de colores.
- Es el formato más común junto con el GIF para publicar imágenes en la web.
- La compresión JPEG puede suponer cierta pérdida de calidad en la imagen. En la mayoría de los casos esta pérdida se puede asumir porque permite reducir el tamaño del archivo y su visualización es aceptable. Es recomendable utilizar una calidad del 60-90 % del original.
- Cada vez que se modifica y guarda un archivo JPEG, se puede perder algo de su calidad si se define cierto factor de compresión.
- Las cámaras digitales suelen almacenar directamente las imágenes en formato JPEG con máxima calidad y sin compresión.
- Ventaja: Es ideal para publicar fotografías en la web siempre y cuando se configuren adecuadamente dimensiones y compresión.
- Inconveniente: Si se define un factor de compresión se pierde calidad. Por este motivo no es recomendable para archivar originales.

### **TIF-TIFF (Tagged Image File Format = Formato de Archivo de Imagen Etiquetada)**

- Almacena imágenes de una calidad excelente.
- Es el formato ideal para editar o imprimir una imagen.
- Ventaja: Es ideal para archivar archivos originales.
- Inconveniente: Produce archivos muy grandes.

### **PNG (Portable Network Graphic = Gráfico portable para la red)**

- Es un formato de reciente difusión alternativo al GIF.
- Tiene una tasa de compresión superior al formato GIF (+10%)

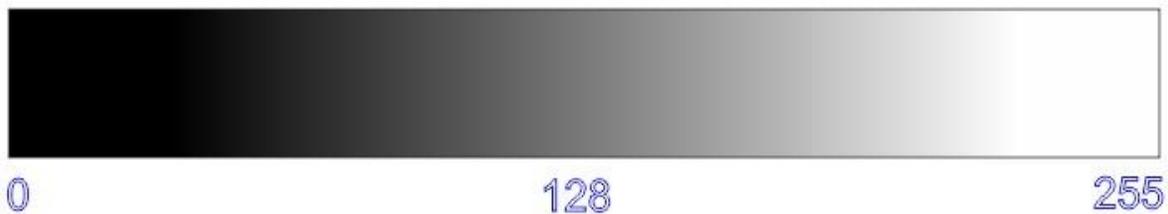
- Admite la posibilidad de emplear un número de colores superior a los 256 que impone el GIF.



Figura 2.6 Representación de formatos de imagen [8]

- Modo Escala de Grises

Este modo se refiere a las imágenes formadas por píxeles que pueden adoptar distintos valores que va de 0 (negro) a 255 (blanco), tal como se ilustra en la **Figura 2.7**. Este espectro de tonalidades se gradúa en una escala de 256 niveles. Por este motivo, el modo Escala de Grises es el más apropiado para la impresión de imágenes en blanco y negro. [2]



**Figura 2.7** Intensidad en escala de grises para 8 bits [2]

- Modelo de colores RGB

Siglas de *red*, *green*, y *blue*, (rojo, verde y azul). RGB es un modelo de color utilizado normalmente para presentar color en los sistemas de video, cámaras, y monitores de ordenadores. Representa todos los colores como combinaciones de rojo, verde y azul. RGB es el sistema de color más utilizado por la mayoría de los formatos actuales. [10]

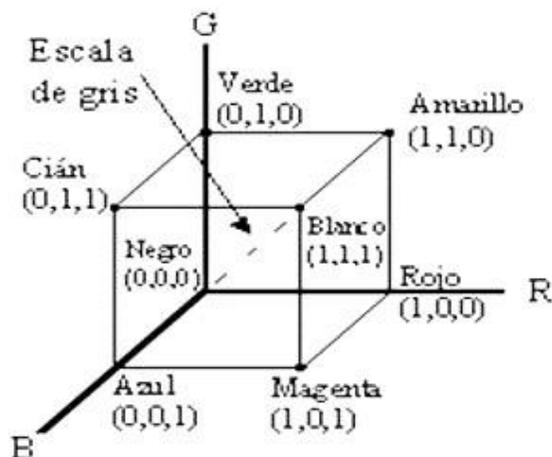
Los colores primarios del modelo RGB son aditivos, es decir, para producir el resultado se suman las contribuciones individuales de cada primario. Como todos sabemos los colores básicos son el **R** RED (ROJO) **G** GREEN (VERDE) Y EL **B** BLUE (AZUL) y de la mezcla de estos y

su aplicación con mayor o menor intensidad obtenemos el resto de colores; encontrándonos en los extremos con el blanco y el negro; es decir un color cuyo valor RGB es 0,0,0 es el color negro, y el color cuyo valor es 255, 255, 255 es el color blanco, en este punto cabe señalar que los valores **RGB** son expresados en una combinación de 3 cifras cuyos valores mínimo y máximo son 0 y 255 respectivamente. Mezclando los diferentes valores de un modo u otro obtendremos los diferentes colores como se describe gráficamente a continuación:

**255,0,0**   **0,255,0**   **0,0,255**   **255,128,0**   **125,23,46** ... y así todas las combinaciones posibles que podamos realizar con los 3 valores.

Representando el modelo en un cubo unitario, cada componente de la triplete toma valores en el rango de 0 a 1. Así un color  $C\lambda$  se expresa:

$$C\lambda = RR + GG + BB \quad \dots (2.2)$$



**Figura 2.8** Modelo de cubo unitario para representar formato RGB [10]

Asumiendo que utilizamos números enteros en el rango 0.255 (un byte), es decir 256 valores posibles para cada color primario, entonces con tres bytes podemos representar 256 x 256

x 256 combinaciones diferentes de los colores primarios, es decir una gama de más de 16 millones de colores (frecuentemente referida como "color verdadero").

## 2.4 Formato BMP

- **BMP**

El formato BMP (Microsoft Windows Bitmap Format) es un formato de propósito general para grabar mapas de bits, los archivos con este formato tienen la extensión BMP y son comúnmente utilizados para grabar imágenes generadas por medio de un digitalizador de imágenes [10].

Un mapa de bits implica una representación física de la imagen con su paleta de colores, este formato soporta un plano o mapa de bits de 1, 4, 8 o 24 bits por pixel.

El bit es la unidad fundamental de memoria de una computadora, cuyo valor es de 0 ó 1. Un byte contiene 8 bits y es la cantidad de información estándar necesario para almacenar un carácter del teclado. Asumiendo que un bit puede contener dos valores y que un byte contiene 8 bits podemos ver que =256 valores en un byte [2].

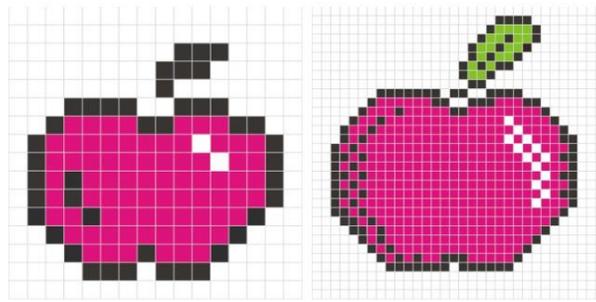
La imagen de mapa de bits o bitmap consiste en una cuadrícula —también llamada ráster, grilla o matriz— donde los píxeles se ordenan en las correspondientes celdas con valores correspondientes a los ejes x e y, horizontal y vertical.

La representación de una imagen cualquiera se obtiene asignando y almacenando un único color en cada uno de los píxeles, es decir que cada uno tendrá un color y una ubicación dentro de la cuadrícula. [9] A diferencia de la imagen vectorial, la imagen bitmap no puede escalarse sin consecuencias que alteren su aspecto y calidad. A la hora de escalar una de estas imágenes, habrá que tener en cuenta la resolución, el modo de color, y el formato de compresión entre otros factores. La nitidez o la calidad visual en una imagen de mapa de bits está dada por su resolución.

Las dimensiones —alto y ancho— de la superficie que ocupa una imagen se miden en píxeles. Por lo tanto, las dimensiones de la imagen estarán ligadas a la resolución y al peso, es decir, al espacio que ocupa la imagen en nuestro ordenador. Debemos tomar en cuenta, sin embargo, que el píxel, a pesar de ser una unidad de medida, es un concepto inmaterial que no tiene una medida concreta. Sería, por lo tanto, imposible determinar si un píxel mide un milímetro o un centímetro. En principio es solamente una unidad de medida variable por medio de la cual se dividen las celdas de una rejilla. Es así que si tenemos una imagen de  $200 \times 100$  píxeles sin saber su tamaño físico real, lo único que sabemos es que la hemos dividido en 20.000 celdillas. Sin embargo, cuando a esa imagen le asignemos una resolución, entonces sí sabremos el tamaño del píxel para esa imagen y para esa resolución dada. Por ejemplo, si decimos que la imagen consta de 100 píxeles por pulgada, tendremos 100 píxeles cada 2,54 centímetros.

Es decir que habrá 100 celdillas por cada 2,54 centímetros, con lo cual cada píxel medirá 0,254 milímetros. Pero si dijéramos que esa imagen tiene una resolución de 1 píxel por pulgada, entonces cada celda o píxel tomará el valor de 2,54 centímetros.

Observemos la imagen siguiente:



**Figura 2.9.** La resolución de una imagen nos indica el número de píxeles que caben en cada unidad de longitud. Es decir que, a mayor resolución e igual tamaño, mayor será la cantidad de píxeles y menor el tamaño de éstos y, por lo tanto, la imagen tendrá mayor nitidez. Al comparar las imágenes vemos que ambas tienen el mismo tamaño, pero el número de píxeles difiere una de la otra. La primera imagen tiene menor resolución que la segunda. [10]

Un aspecto importante en este formato es que las líneas exploradas de una imagen comienzan a grabarse desde la línea inferior hasta la línea superior de ésta, lo cual no ocurre en la mayoría de los formatos como JPEG, GIF, PNG entre otros.

Un archivo BMP consta de cuatro estructuras o bloques diferentes [12]. El primer bloque lo conforma la cabecera del mapa de bits (xBITMAPFILEHEADER) que se encuentra al principio de cada archivo y solo sirve para validar el formato, conteniendo la información que se muestra en la Tabla 2.1.

**Tabla 2.1** Contenido del bloque de la cabecera [12]

Variables	Longitud en bytes	Definición
<b>bfType</b>	2 bytes	Igual a 4D42h: Corresponde al código ASCII de las letras BM
<b>bfSize</b>	4 bytes	Indica el tamaño del archivo BMP en bytes
<b>bfReserved1</b>	2 bytes	Contienen el valor reservado que es igual a 0
<b>bfReserved2</b>	2 bytes	
<b>bfOffBits</b>	4 bytes	Indica en donde comienza el mapa de bits a partir del inicio del archivo

- Cabecera del archivo bitmap (BITMAPFILEHEADER)

Proporciona la información acerca del archivo el tipo, tamaño y diseño. Esta cabecera del archivo comprende cuatro campos.

- Encabezado de información del mapa de bits (BITMAPINFOHEADER)

Esta parte de la estructura proporciona la información acerca de la imagen en especial las dimensiones y colores,

Esta información está comprendida en la **Tabla 2.2**.

**Tabla 2.2** Contenido del bloque de la cabecera de mapa de bits [12]

Variables	Longitud en bytes	Definición
biSize	4 bytes.	Tamaño de la estructura.
biWidth	4 bytes.	Ancho del mapa de bits en pixeles
biHeight	4 bytes.	Largo del mapa de bits en pixeles
biPlanes	2 bytes.	Número de planos en el mapa de bits
biBitCount	2 bytes.	Cantidad de bits por pixel
biCompression	4 bytes.	Compresión de la imagen.
biSizeImage	4 bytes.	Tamaño del mapa de bits
biXPelsPerMeter	4 bytes.	Tamaño horizontal del mapa de bits en pixeles por metro.
biYPelsPerMeter	4 bytes.	Tamaño vertical del mapa de bits en pixeles por metro.
biClrUsed	4 bytes.	Número de colores usados en el mapa de bits.
biClrImportant	4 bytes.	Indica el número de colores en la paleta de la imagen que son considerados importantes.

○ Paleta de imágenes (RGBQUAD)

Esta paleta es opcional. Contiene cuatro bytes de forma sucesivas para cada una de las entradas. Esta información se muestra en la **Tabla 2.3**

**Tabla 2.3** Contenido de la paleta de colores de mapa de bits [12]

Variables	Longitud en bytes	Definición
rgbBlue	1 byte.	Intensidad del color azul, verde y rojo primarios respectivamente. Cada uno de ellos tiene rango de 0 a 255.
rgbGreen	1 byte.	
rgbRed	1 byte.	
rgbReserved	1 byte.	Indica la intensidad de un color extra, el cual no se usa.

○ Codificación de imágenes

Esta se realiza escribiendo los bits de forma sucesiva que correspondan a cada pixel, línea por línea, comenzando por el pixel del extremo inferior izquierdo.

- Las imágenes de 2 colores usan 1 bit por pixel.
- Las imágenes de 16 colores usan 4 bit por pixel
- Las imágenes de 256 colores usan 8 bit por pixel.

Cada línea de la imagen debe tener un número total de bits que sea múltiplo de 4, sino se debe completar con todos los 0 necesarios para respetar el criterio.

En la **Figura 2.10** se muestra un ejemplo de un bitmap

<u>BitmapFileHeader</u>		<u>Win3ColorTable</u>			
<u>Type</u>	19778	<u>Blue</u>	<u>Green</u>	<u>Red</u>	<u>Unused</u>
<u>Size</u>	3118	[00000000]	84	252	84 0
<u>Reserved1</u>	0	[00000001]	252	252	84 0
<u>Reserved2</u>	0	[00000002]	84	84	252 0
<u>OffsetBits</u>	118	[00000003]	252	84	252 0
<u>BitmapInfoHeader</u>		[00000004]	84	252	252 0
<u>Size</u>	40	[00000005]	252	252	252 0
<u>Width</u>	80	[00000006]	0	0	0 0
<u>Height</u>	75	[00000007]	168	0	0 0
<u>Planes</u>	1	[00000008]	0	168	0 0
<u>BitCount</u>	4	[00000009]	168	168	0 0
<u>Compression</u>	0	[0000000A]	0	0	168 0
<u>SizeImage</u>	3000	[0000000B]	168	0	168 0
<u>XPelsPerMeter</u>	0	[0000000C]	0	168	168 0
<u>YPelsPerMeter</u>	0	[0000000D]	168	168	168 0
<u>ColorsUsed</u>	16	[0000000E]	84	84	84 0
<u>ColorsImportant</u>	16	[0000000F]	252	84	84 0

**Figura 2.10** Ejemplo de un bitmap[12]

## 2.5 Energía de una imagen

La función de energía de una señal representa la energía disipada por una resistencia de 1 ohm cuando se le aplica un voltaje  $x(t)$ . En una señal continua, la Energía total  $E$  en el intervalo de tiempo  $t_1$  a  $t_2$  está definida como:

$$E = \int_{t_1}^{t_2} |x(t)|^2 dt \quad (2.3)$$

Para el caso de las señales discretas donde  $N$  es el número de muestras de la señal, la energía se define por:

$$E = \sum_{m=0}^{N-1} x(m)^2 \quad (2.4)$$

## 2.6 Distancias Euclidianas

Para lograr el método de verificación de la condición de llenado en el presente trabajo, se necesita obtener los valores de umbral, máximo y mínimo para crear una base de datos, con las que se pueda trabajar para la obtención de los resultados deseados. Dichos umbrales serán el rango de clasificación para determinar si un envase está lleno o vacío.

Teniendo identificadas las variables a utilizar para el análisis de cada envase, en este caso como se mencionó en los fundamentos de la imagen digital, el pixel es la unidad mínima y es nuestra base de estudio, por tanto, cada envase tendrá un valor determinado para cada variable y partiendo de esto se realizará la comparación y clasificación.

Para verificar adecuadamente la condición de llenado de cada envase es necesario determinar los niveles de umbral de cada envase, calculando la distancia que existe entre las diferentes matrices que conforman las imágenes de cada envase de prueba.

Una distancia mayor nos indica que la condición de llenado comparado con las imágenes en la base de datos para cada caso estará más alejada del umbral promedio obtenido y una distancia menor nos indica que la condición de llenado comparado con las imágenes en la base de datos para cada caso estará más cercana del umbral promedio obtenido, de forma que cuando mayor sea su valor, mayor serán las diferencias entre los pixeles de la imagen y será menor la probabilidad de que los métodos de clasificación los coloquen en un mismo grupo.

La manera más directa de comparar la imagen original con una colección de imágenes, es comparar cada pixel del cuadro de la imagen correspondiente con su correspondiente pixel en la imagen de la colección de imágenes, y acumular la distancia entre cada pareja de pixeles para determinar la distancia general entre las dos imágenes. [8]

Las funciones de distancia más comunes utilizadas son la distancia euclidiana, la distancia de Manhattan, la distancia de Chebychev, y la distancia Minkowski. Estas distancias están definidas de la siguiente manera:

Sea  $X$  un objeto arbitrariamente descrito por el vector de características:

$$a_1(x); a_2(x), \dots, a_3(x) \quad (2.5)$$

Donde denota el valor de  $r$ -ésimo atributo del objeto; entonces la distancia entre dos objetos  $x_i$  y  $x_j$ , es definida por para cada una de las siguientes distancias

- Distancia euclidiana

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (2.6)$$

- Distancia de Manhattan

$$d(x_i, x_j) = \sum_{i=1}^n |a_r(x_i) - a_r(x_j)| \quad (2.7)$$

- Distancia de Chebychev

$$d(x_i, x_j) = \max_{i=1} |a_r(x_i) - a_r(x_j)| \quad (2.8)$$

- Distancia de Minkowski

$$d(x_i, x_j) = (\sum_{i=1}^n (a_r(x_i) - a_r(x_j))^p)^{\frac{1}{p}} \quad (2.9)$$

- Distancia de Mahalanobis

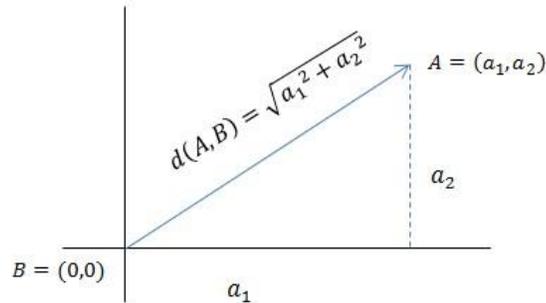
$$d(x_i, x_j) = ((a_r(x_i) - a_r(x_j))^{ts-1} (a_r(x_i) - a_r(x_j)))^{\frac{1}{2}} \quad (2.10)$$

La Distancia Euclidiana está dada por la siguiente expresión:

$$d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \quad (2.11)$$

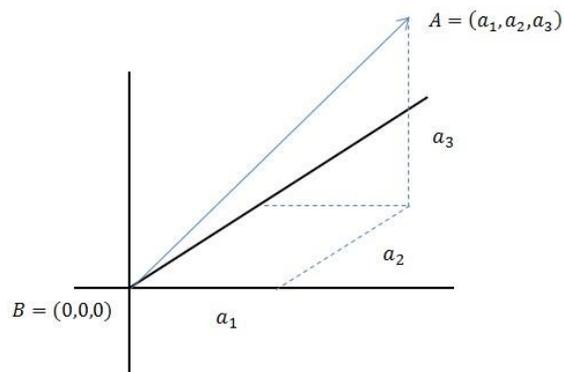
Cuando  $n=2$ , la Distancia Euclidiana tiene su razón de ser en el célebre *Teorema de Pitágoras*. De hecho, si y al graficar los puntos en un sistema cartesiano de dos dimensiones,

se ve que la distancia de A a B,  $d(A, B)$ , es la hipotenusa de un triángulo rectángulo de catetos y como se muestra en la **Figura 2.11**.



**Figura 2.11** Distancia Euclidiana en dos dimensiones

Cuando  $n=3$ , la Distancia Euclidiana tiene su fundamento en una generalización del Teorema de Pitágoras como se muestra en la **Figura 2.12**.



**Figura 2.12** Distancia Euclidiana en 3 dimensiones.

- NIVELES DE UMBRAL

Los **métodos del valor umbral** son un grupo de algoritmos cuya finalidad es segmentar gráficos rasterizados, es decir separar los objetos de una imagen que nos interesen del resto. Con la ayuda de los métodos de valor umbral en las situaciones más sencillas se puede

decidir qué píxeles conforman los objetos que buscamos y qué píxeles son sólo el entorno de estos objetos.

Las técnicas de umbralización buscan obtener un valor de umbral que permita binarizar a la imagen separando adecuadamente el fondo (background) y el objeto a separar (foreground). Muchas de las técnicas de umbralización están basadas en la información estadística que brinda el histograma, sobre todo en aquellas imágenes donde los objetos tienen una superficie o textura homogénea y el fondo es más o menos uniforme.

Si el histograma de una imagen posee dos picos o lóbulos (**Figura 2.13**) podemos separar dos zonas o regiones y el umbral será aquel valor que se encuentra en el valle entre ambas. Este es un caso ideal, el histograma de intensidad de una imagen (niveles de gris) tendría bien marcado los dos picos para objeto y fondo, y diríamos que el umbral óptimo es aquel valor  $T$  que separa ambas regiones.

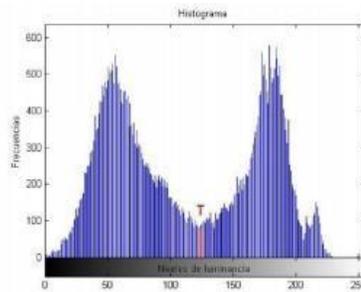


Figura 1. Histograma ideal

*Figura 2.13 Histograma ideal*

Global en gris: Los métodos se emplearán en imágenes convertidas en escalas de grises a través el comando `rgb2gray` de MatLab.

Global en Capas de Colores: Los algoritmos de cada método se emplearán en las capas de color (Rojo, Verde y Azul)

**Tabla 2.4** Umbrales globales para cada método formato RGB

Metodo	Umbrales globales		
	Rojo	Verde	Azul
Yen	114	109	100
Pun	45	48	40
Kapur	120	116	115
Kittler	118	110	94

## 2.7 Métodos alternos para el procesamiento digital de imágenes (artículos)

Existen distintos métodos que pueden ser utilizados para los mismos fines, sin embargo, no son viables por distintas razones.

A continuación, se mencionan algunos métodos que son utilizables en el procesamiento digital de imágenes y algunos otros que ya han sido aplicados en la inspección de nivel de llenado de envases.

### 2.7.1- “Método de umbralado por determinación de píxeles para la segmentación de elementos en imágenes aplicado al análisis de problemas en la industria”.

**Procesamiento por Histograma:** El histograma de una imagen es una representación gráfica de la frecuencia de ocurrencia de intensidad de pixel en la imagen. En una imagen de escala de grises de 8 bits, como un documento examinado, 256 valores de la escala de grises son posibles. Otro método de adquisición de imagen, por ejemplo, una cámara digital, podría producir las imágenes de 16 bits que resulta en 65,536 posibles valores de escala de grises. Un pico grande en un histograma de la imagen identifica un grupo de valores de escala de grises común, mientras que un valle indica que uno o varios valores de escala de grises adyacentes son menos comunes en la imagen. En una imagen a color, los componentes de

color individuales, rojo, verde y azul (RGB) pueden verse similarmente como histogramas. Aunque el histograma representa cada pixel dentro de una imagen, la información espacial sobre cada pixel se pierde, es decir, el número de pixeles a un valor de escala de grises específico se representa en el histograma, pero el histograma no proporciona la localización de esos pixeles dentro de la imagen original.

También, porque la información de situación de pixel se pierde, una imagen no puede regenerarse de su histograma. El eje vertical del histograma representa la frecuencia de ocurrencia del pixel; el eje horizontal representa los niveles de escala de grises.

En el histograma de una imagen monocromática, se localizan los pixeles negros en 0, y se localizan los pixeles blancos en 255. Su histograma presenta un rango grande de niveles de escala de grises cerca de cero hasta 255. Al más gran número de pixeles en la imagen está en el nivel de la mitad de grises. Aunque el histograma no proporciona las localizaciones del pixel, puede ser valioso conocer la forma del histograma. En algunos casos, manipulando el histograma pueden proporcionar una imagen más agradable visualmente. Por ejemplo, en la igualación del histograma, una función de transferencia se aplica al histograma que rellena algunos de los valles de algunos de los picos para utilizar más de los valores de la escala de grises disponibles. Sin embargo, desde que la igualación del histograma asigna nuevos niveles de escala de grises a algunos pixeles, el análisis cuantitativo no es más grande posible. La igualación del histograma también falla en varias situaciones; su efectividad disminuye cuando la variación en el contraste ocurre a una imagen, como podría ser el caso con la cubierta parcial de nubes. La igualación del histograma adaptable considera el problema de cambios de contraste aplicando las funciones de transferencia similares al histograma de cada vecino del pixel. Aunque este método ejecuta igualación tradicional del histograma para imágenes de contraste bajo, hay por lo menos dos desventajas asociadas con la igualación del histograma adaptable: el proceso de tiempo de computadora es intenso, y el resultado es una representación artificial de la imagen original .

### 2.7.2 “Método de Binarización de Imágenes”

Las imágenes binarias siempre se obtienen a partir de imágenes de niveles de gris. En la actualidad no existen cámaras comerciales que proporcionen imágenes binarias.

El proceso de conversión de una imagen de nivel de gris a una imagen formada solo por dos valores o etiquetas (0 para el negro y 1 para el blanco) se conoce como **binarización**. [14]

La mayor parte de las aplicaciones no precisan de tantos niveles de gris sino más bien lo contrario; pueden utilizar pocos niveles debido a que trabajan con escenas de muy alto contraste. Tanto es así que en muchas aplicaciones industriales se llega al extremo de utilizar únicamente dos niveles de gris. De esta forma, se obtiene lo que se conoce como imagen binaria. Trabajar con imágenes binarias resulta muy interesante por dos motivos:

En primer lugar, porque se reduce al mínimo los datos necesarios para representar la imagen y ello permite un máximo aprovechamiento de la potencia computacional.

En segundo lugar, porque las propiedades geométricas y topológicas de los objetos presentes en la imagen, en las que se basan un gran número de aplicaciones industriales, puede obtenerse rápida y fácilmente a partir de las imágenes binarias. Desde el punto de vista computacional, las imágenes binarias se procesan mucho más rápidamente. Por este motivo, las primeras aplicaciones de visión artificial en línea emplearon este tipo de imágenes casi exclusivamente.

En definitiva, la binarización consiste en, a partir de un nivel de gris predeterminado denominado umbral de binarización, etiquetar como 0 todos los píxeles con niveles de gris inferior a ese umbral y como 1 los que tengan un nivel de gris igual o superior. Si la imagen está bien contrastada, la pérdida de información es mínima.

### 2.7.3 Equipos de inspección de envases

De hecho, hasta los envases transparentes presentan retos de inspección de control de calidad únicos, afirma Rick Reardon, gerente de la línea de productos, de Teledyne TapTone, una compañía de sistemas de inspección de empaques en North Falmouth, Massachusetts. Y dice, “Los empaquetadores están encarados con nuevas, y veces desafiantes aplicaciones, tales como la dosificación con nitrógeno líquido [LN2] a las altas velocidades de las líneas de producción. Llenar envases con fluidos calientes o el introducir una nueva tapa o un nuevo cierre también pueden presentar desafíos especiales para los fabricantes de bebidas, quienes tradicionalmente han manejado envases de vidrio”. Y continúa, “El control de calidad se ha transformado en un problema que han estado lidiando los envasadores de bebidas, y, por ende, para nosotros, la detección de fugas y derrames se ha transformado en una prioridad”.

Las series de sistemas de inspección de envase de la compañía Teledyne TapTone son diseñadas específicamente para aplicarse al problema de la detección de fugas de envases - los cerrados a presión y los no cerrados a presión - durante la producción y a su alta velocidad.

“Es buena idea pescar un problema antes de que le llegue al consumidor, pero es mejor aún resolver el problema permanentemente, para que no vuelva a suceder”, dice Dechman.

Para eso, el sistema de MTC de cámaras de alta velocidad, 20/20 Hindsight (gigE) hace de la video grabación de alta velocidad un dispositivo industrial del tipo enchufa-y-juega para efectuar el control de calidad. La tecnología gigabyte Ethernet (gigE) permite la video grabación de alta velocidad que va de 2,000 a 3,000 cuadros por segundo.

### 2.7.4 “Sistemas de inspección por rayos-x de envases de vidrio”

Seguridad del producto y garantía de calidad para botellas y tarros de vidrio

Los sistemas de inspección de vidrio de Safeline X-ray se han ganado la confianza de fabricantes de alimentos, bebidas y productos farmacéuticos de todo el mundo para garantizar la seguridad e integridad de sus productos. Nuestros sistemas de inspección por rayos-x son ideales para detectar contaminantes en tarros y botellas de vidrio utilizados como envases para una amplia gama de productos alimenticios, como condimentos y salsas, pescado y marisco, frutas y verduras, productos de alimentos infantiles, comida preparada, carne y aves, y productos lácteos, además de bebidas y medicamentos presentados en envases de vidrio.

Las máquinas de inspección de vidrio pueden detectar una amplia gama de contaminantes densos, como metal, piedra mineral, hueso calcificado, plástico de alta densidad, compuestos de caucho y vidrio en vidrio, independientemente de su forma, tamaño o ubicación dentro de la botella o del tarro. Al garantizar los máximos niveles de seguridad y calidad de los productos, los sistemas de inspección de Safeline X-ray garantizan el bienestar del consumidor, evitan que el cliente no quede satisfecho, minimizan el riesgo de que sea necesario realizar retiradas de productos y ayudan a los fabricantes a preservar la integridad de marca.



**Figura 2.14** A split-beam (100W / 400W) x-ray system designed to inspect metal, glass and ceramic containers. [Mettler Toledo].

# Capítulo 3 Implementación y Desarrollo de Software Y Hardware

## 3.1 Dispositivo de captura de las imágenes de envases de vidrio.

El dispositivo para captura de envases de vidrio, es una cámara web mostrada en la **Figura 3.1.**



**Figura 3.1.** Cámara Web True Basix CW-1000

La cámara web tiene las siguientes especificaciones técnicas:

- El nombre de la cámara web es True Basix Cámara Web Estándar modelo CW-1000
- Las interfaces que utiliza son USB 1.1 y USB 2.0.
- El tamaño de la imagen capturada es de 640\*480 pixeles.
- La cámara web cuenta con un cable USB de 1.3 metros que utilizaremos para conectarla con la computadora.

Para obtener el umbral máximo y mínimo de energía tanto de los envases de vidrio llenos como de los envases de vidrio vacíos, capturamos 40 imágenes tanto de envases llenos como de envases vacíos.

Las imágenes se tomaron a escalas de grises como se muestra en la **Figura 3.2** y la **Figura 3.3** respectivamente.



*Figura 3.2. Imagen de un envase de vidrio vacío.*



*Figura 3.3. Imagen de un envase de vidrio lleno.*

### 3.2 Comunicación de la Cámara Web con la PC

Para lograr una conexión exitosa entre la cámara web y la PC el fabricante nos menciona los siguientes requerimientos:

- Sistema operativo Windows XP, Windows Vista o Windows 7.
- OS (Sistema Operativo): Tarjeta de sonido y altavoces.
- Tecnología USB: 1.1 o 2.0.

Utilizaremos la tecnología USB 2.0 (Universal Serie Bus) que es una interfaz plug and play entre la PC y la cámara web la cual trabaja a velocidades en promedio de 12Mbps, trabaja como una interfaz de transmisión de datos y de distribución de energía, el cual consta de tres componentes los cuales son el controlador, hubs o concentradores y periféricos.

La tecnología Plug and Play ayuda a que los controladores se instalen automáticamente sin necesidad de apagar el equipo ni que el sistema busque el nuevo hardware ya que el sistema automáticamente reconoce el dispositivo conectado.

### 3.3 Desarrollo del prototipo

Dentro del desarrollo del prototipo se hará referencia al desarrollo del hardware; que describe cómo se montó la cámara web para poder capturar la imagen de envases llenos y vacíos, en primera instancia el diseño y realización de una banda transportadora que nos permitirá simular una línea de producción.

También se menciona el desarrollo del software, en el cual se llevaron a cabo los algoritmos de verificación y la interfaz gráfica.

## 3.4 Desarrollo del hardware

### 3.4.1 Diseño de la banda transportadora

Para el desarrollo del hardware se utilizó el material mencionado a continuación para simular la línea de producción y capturar la imagen de los envases.

- Cámara web modelo *True Basix CW-1000*
- Motorreductor *B01 1:220*
- 1 LED de 3mm (Diodo emisor de luz) luz de color rojo.
- 1 eliminador a 5.25 v.
- 1 interruptor de 1 posición a xxxx volts
- 100 cm de alambre para protoboard.
- 1 engrane de 3cm de diámetro y 36 dientes.
- 1 engrane de 6cm de diámetro y 60 dientes.
- 6 cinturones de plástico de 10x0.25cm.
- 2 rodillos de madera de 3.3 cm de diámetro x 14 cm de largo.
- 2 rodillos de madera de 0.6 cm de diámetro x 19 cm de largo.
- 2 soportes de madera (base) 60 x 5x2cm.
- Base de madera plana de 60x25 x0.9 cm.
- 4 soportes de madera vertical de 30x 5x2 cm.
- 1 soporte de madera horizontal de 34x5x2cm
- Base de triplay plana de 34x18x0.4cm.
- 1 pieza de acrílico de 34x30cm.
- 1 pieza de acrílico de 34x10cm.
- 1 ángulo de aluminio de 1.8x1.8cm
- Soporte de metal (tipo gusano) para cámara de 28.5cm.

- 6 remaches de 2cm de largo
- Tornillos para madera de 3.8cm de largo.
- Pedazo de tela de 90x 10cm.
- Pegamento 5000 0 silicón.
- Pintura de aceite (color blanco y negro)

### Procedimiento

- ✦ Con la base de madera plana de 60x25x0.9cm y los soportes de madera (base) de 60x5x2cm se armó la base colocando 4 tornillos para madera de 3.8cm de largo, en cada uno de los vértices de la madera plana para sujetar los soportes. Posteriormente se pintó de color negro. Como se muestra en la **Figura 3.4**.



**Figura 3.4** Base de madera plana sujeta al soporte con tornillo

- ✦ Para los 4 soportes verticales de madera de 30x 5x2 cm, se hizo una perforación con una broca para madera de 0.635 cm de diámetro, a una altura de 22.5 cm desde la base a 2.5 cm del ancho de cada uno, como se muestra en la **Figura 3.5**.



*Figura 3.5 Soporte vertical con perforación de 0.635 cm*

- ✦ A uno de los soportes verticales de madera de 30x5x2cm, se le hizo una perforación 3 cm arriba del orificio anterior con una broca para madera de 0.833cm de diámetro. Como se muestra en la **Figura 3.6**. Esto para colocar el soporte en donde se va a hacer el montaje de la cámara web.



*Figura 3.6 Soporte vertical con los dos orificios de 0.635 cm y 0.833 cm de diámetro.*

- ✦ Se fijaron los 4 soportes de madera a la base con 2 tornillos de 3.8cm de largo cada uno, a 18 cm de largo x 3 cm de ancho con respecto a cada una de las esquinas de la misma. Como se muestra en la **Figura 3.7**.



*Figura 3.7 Montaje de los 4 soportes verticales a la base de madera plana*

- ✦ Se hizo una perforación en el centro de los rodillos de madera de 3.3 cm de extremo a extremo, con una broca de 0.635cm de diámetro. Se introdujo el rodillo de 0.6cm dentro del orificio del rodillo de 3.3cm. Como se muestra en la **Figura 3.8**. Con la finalidad de que el cilindro pequeño este sujeto a los soportes de madera antes mencionados.



*Figura 3.8 Rodillos ensamblados*

- ✦ Se pegó con silicón el engrane de 3cm a un extremo de uno de los rodillos. Se colocaron los rodillos en los soportes verticales tomando como base los orificios que se realizaron previamente. **Figura 3.9**.



**Figura 3.9** Rodillos ensamblados a los soportes verticales mediante los orificios de 0.635cm

- ✦ Tomando como referencia el engrane pegado a uno de los rodillos se colocó debajo de éste el engrane de 6cm previamente ensamblado al motorreductor (verificar que los engranes se puedan mover libremente), posteriormente fue sujetado el motorreductor a uno de los soportes verticales con los 6 cinturones plásticos. Como se muestra en la **Figura 3.10**.



**Figura 3.10** Motorreductor sujeto al soporte vertical con los engranes en sincronía para girar

✦ Para el movimiento de la banda se utilizó el motorreductor **B01 1:220**, el cual se muestra en la **Figura 3.11**



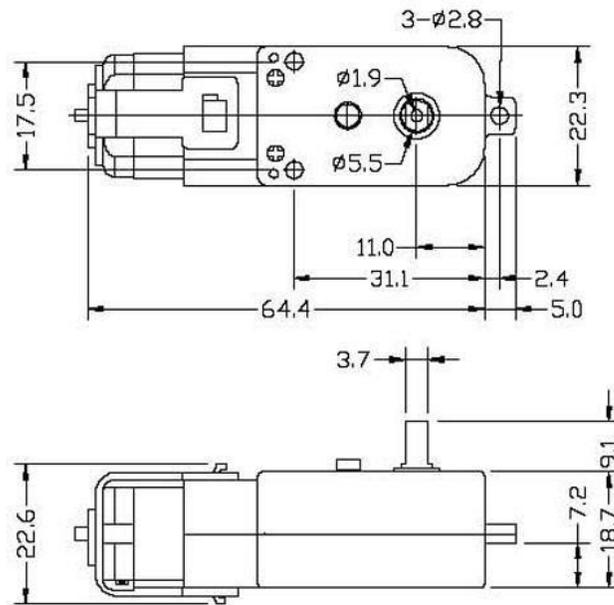
**Figura 3.11.** Motorreductor B01 1:220, [Robodacta]

En la **Tabla 3.1** se muestran las especificaciones técnicas del motorreductor B01 1:220

**Tabla 3.1** Especificaciones técnicas del motorreductor B01 1:220,  
[Robodacta]

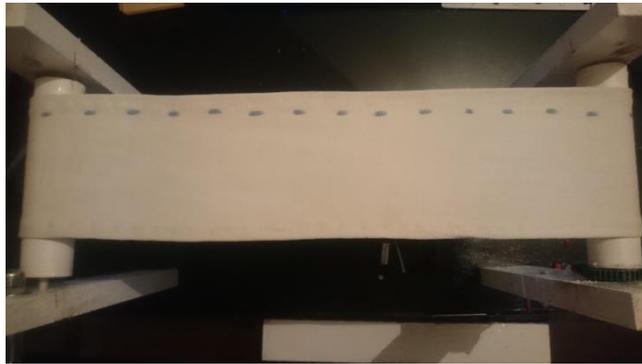
Especificaciones	
Voltaje	5 V
Corriente (sin torque)	75 mA
Corriente máxima (con torque)	670 mA
Velocidad (RPM)	38 RPM
Torque	3.6 Kg-cm
Características adicionales	-relación 1:220

En la **Figura 3.12** se muestra el diagrama interno del motorreductor *B01 1:220* antes mencionado, [Robodacta]



**Figura 3.12** *B01 1:220*, [Robodacta]

- ✦ Se colocó el pedazo de tela sobre los rodillos formando una banda transportadora, considerando que la tensión sea suficiente para hacer girar los rodillos. **Figura 3.13**



**Figura 3.13** Cinta transportadora montada en los rodillos

- ✦ Para darle soporte a la banda transportadora y que el peso de los envases no venza la banda, se pegó con silicón el soporte horizontal de 34x5x2cm a una altura de 19.5cm desde la base tomando como referencia el lado de los engranes. Se fijó con silicón la base de triplay de 34x18x0.4cm al soporte horizontal quedando en medio de la banda transportadora para así reforzar el soporte. Como se muestra en la **Figura 3.14**.



**Figura 3.14** Banda transportadora con soportes para reforzar y soportar el peso.

- ✦ Con las dos piezas de acrílico y el ángulo de aluminio se formo un ángulo de 90 grados entre las piezas que se fijaron con los 6 remaches de 2cm de largo. **Figura 3.15**



**Figura 3.15** Piezas de acrílico montadas en el ángulo de aluminio

- ✦ Habiendo formado una sola pieza de aluminio la pegamos a 10 cm de la base plana de madera en la parte contraria al soporte de 34x5x2cm (**Figura 3.14**), para simular una pared y un techo blanco para los envases de vidrio que pasaran sobre la banda transportadora, a fin de que aparezcan irregularidades. Como se observa en la **Figura 3.16**



**Figura 3.16** Acrílico montado en la banda transportadora con vista frontal y posterior

- ✦ Tomando el soporte de madera con las dos perforaciones de la **Figura 3.6**, sujetamos el soporte de metal (tipo gusano) para cámara de 28.5cm. Tal como se muestra en la **Figura 3.17**.



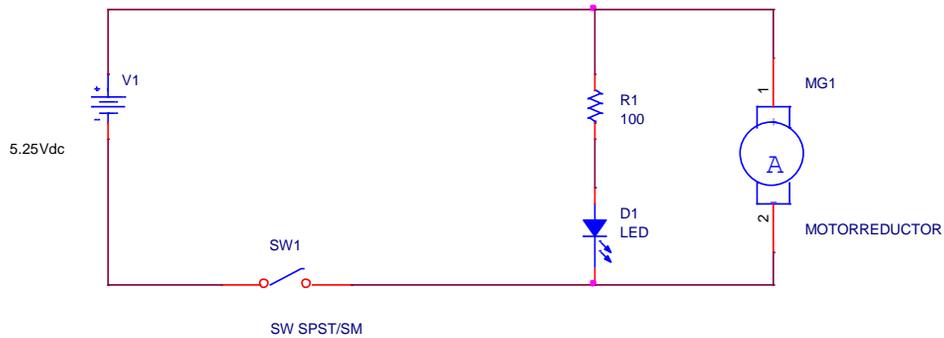
*Figura 3.17 Soporte de metal sujetado a la banda transportadora*

- ✦ Se pegó la cámara al extremo libre del soporte de metal para así poder hacer el escaneo de los envases sobre la banda transportadora. **Figura 3.18**



*Figura 3.18 Cámara montada al soporte de metal*

- ✦ Se montó el circuito mostrado en la **Figura 3.19**, el cual controla el giro del motorreductor mediante el interruptor, así mismo el torque hace que giren los engranes y por consecuencia los rodillos para el funcionamiento de la banda.



**Figura 3.19** Diagrama eléctrico de la banda transportadora

✦ En la **Figura 3.20** se muestra el prototipo terminado.



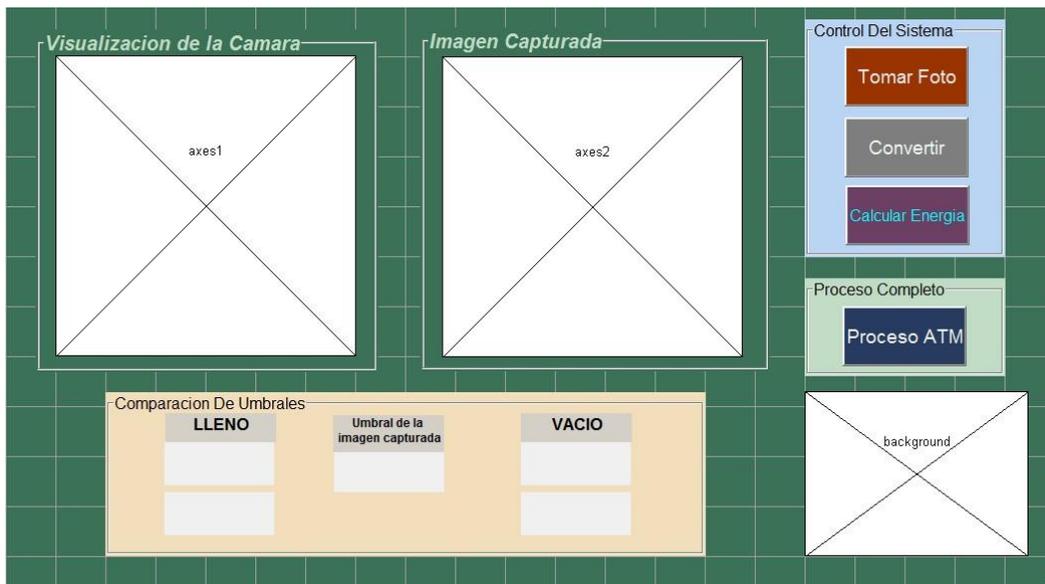
**Figura 3.20** Prototipo terminado

### 3.5 Diseño de la Interfaz Gráfica.

Para este trabajo, utilizando MATLAB, se diseñó una interfaz de usuario (GUI) para lograr capturar las imágenes de los envases de vidrio y de esta forma poder determinar si están vacíos o llenos y de esta forma poder clasificarlos.

Una GUI contiene diferentes elementos gráficos, tales como botones, campos de texto, menús, gráficas, etc. La forma de implementar las GUIs con Matlab es crear objetos y definir qué acciones se desean.

Para nuestro trabajo, la GUI mostrada en la **Figura 3.21**, estará conformada por tres pantallas (Axes), una en la cual se podrá apreciar en tiempo real la imagen de los envases de vidrio que se capturará, otra en la cual se presentara la imagen en formato BMP a escala de grises y otra para visualizar el escudo de ESIME, cuatro botones, uno para realizar la captura de la imagen, otro para convertir la imagen capturada a escala de grises, otro para calcular la energía de la imagen y determinar si el envase está vacío o lleno y el último realiza el proceso completo, ocho static text los cuales están divididos en tres secciones, la primera nos muestra los umbrales previamente adquiridos de los envases llenos, la segunda nos muestra los umbrales previamente adquiridos de los envases vacíos y la última nos muestra el umbral que está calculando la interfaz de la imagen capturada.



**Figura 3.21** Diseño de la interfaz gráfica de usuario GUI \*.fig.

Las funciones que realiza cada uno de los componentes con los que cuenta la GUI se explican a continuación:

- Axes 1:

El axes 1 genera una pantalla en la cual el usuario puede observar en tiempo real la imagen que la webcam capta.

- Axes 2:

El axes 2 muestra al usuario la imagen capturada y después convertida a escala de grises y en formato BMP.

- Axes 3:

El Axes 3 lo utilizamos para añadir una imagen en formato JPG con el objetivo darle presentación a la interfaz, en este caso, el escudo de ESIME.

- Botón Tomar Foto:

Éste, realiza la función de capturar la imagen de los envases de vidrio y mostrarlos en el Axes 2. en escala de grises que se muestra en el axes 2. Al mismo tiempo hace la función de compararla con los umbrales de energía previamente obtenidos y determinar si el envase está lleno o está vacío.

- Botón Convertir:

Este botón hace la función de convertir la imagen previamente capturada a escala de grises en formato BMP y mostrarla en el axes 2.

- Botón Calcular Energía:

Este botón hace la función de calcular la energía en la imagen previamente convertida a escala de grises para de esta manera compararla con los umbrales previamente adquiridos y así determinar si el envase está lleno o está vacío.

En la **Figura 3.5** se muestra la interfaz gráfica de usuario en funcionamiento.

- Boton Proceso ATM:

Este botón realiza el proceso completo, es decir, captura la imagen, la convierte a escala de grises, calcula su energía y la compara con los umbrales previamente adquiridos para determinar si el envase está lleno o está vacío.

- Primera sección de Static text:

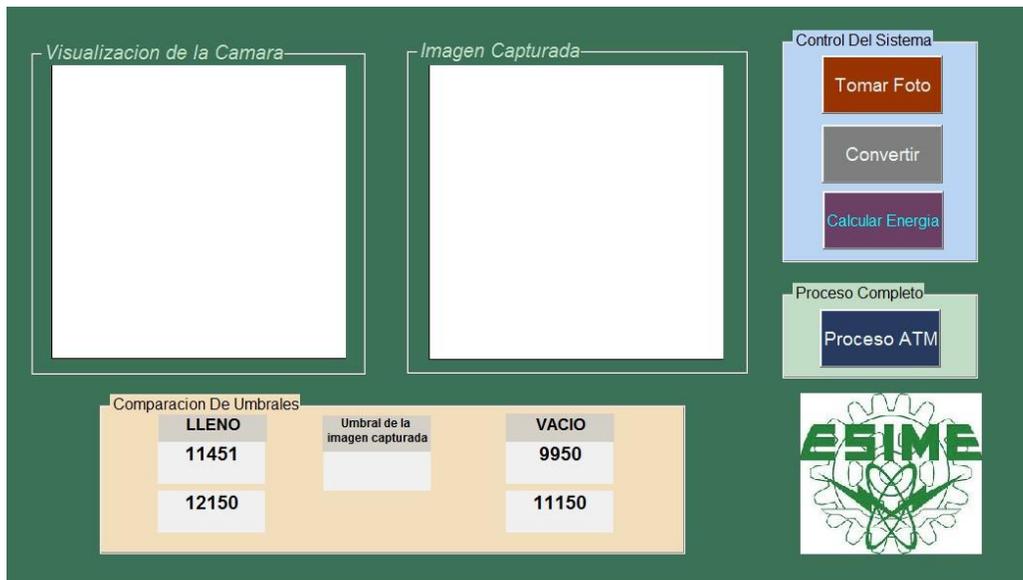
En esta sección tenemos primero el título "LLENO" y nos muestra los umbrales previamente adquiridos de envases llenos.

- Segunda sección de Static text:

En esta sección tenemos primero el título “VACIO” y nos muestra los umbrales previamente adquiridos de envases vacíos.

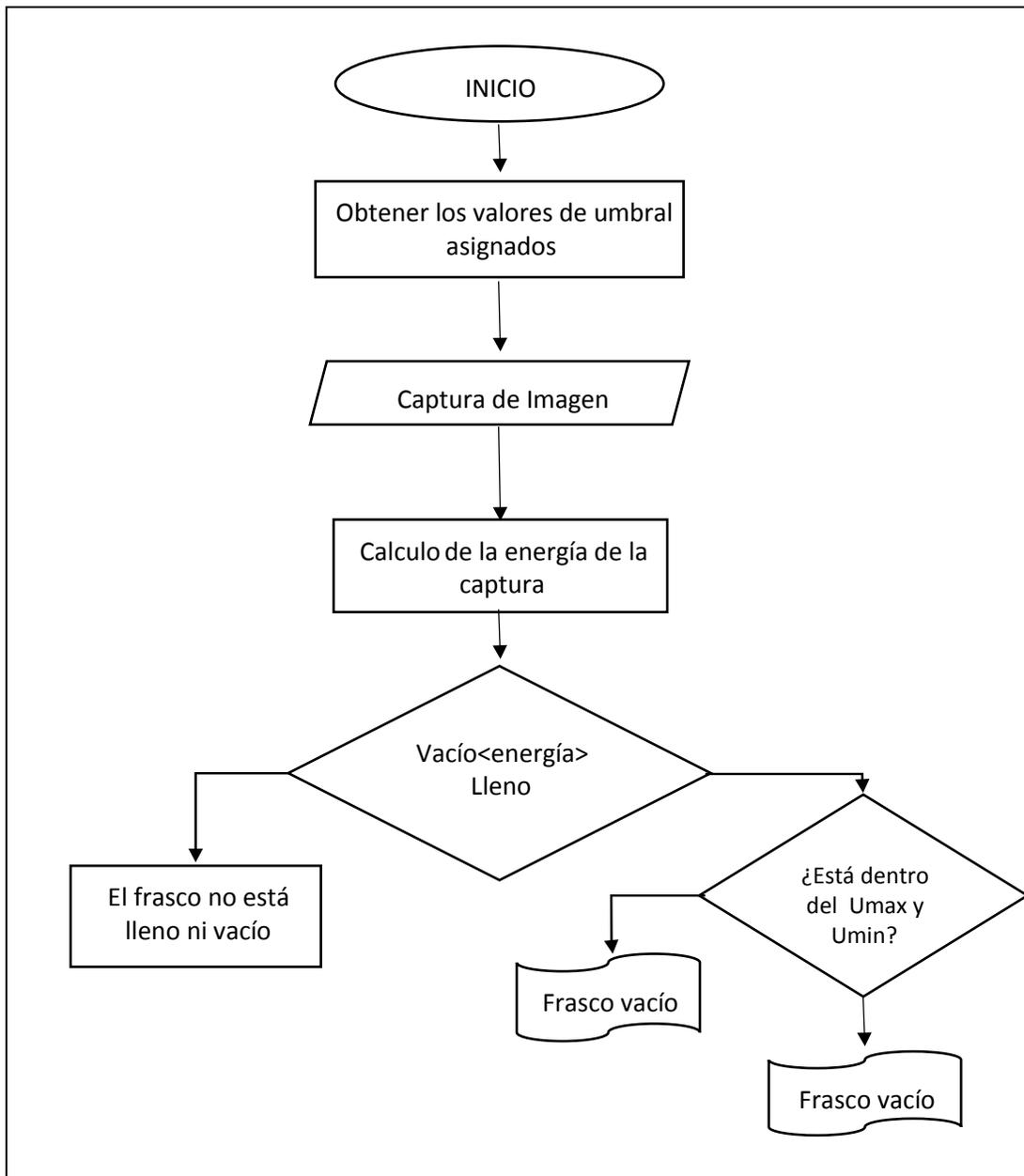
- Tercera sección de Static text:

En esta sección tenemos primero el título “Umbral de la Imagen Capturada” y nos muestra la energía calculada a tiempo real de la imagen previamente calculada. **Figura 3.22**



**Figura 3.22** Interfaz gráfica de usuario GUI.

En la **Figura 3.23** se muestra el funcionamiento del sistema:



**Figura 3.23.** Diagrama de flujo del funcionamiento del sistema.

### 3.6 Programación del sistema.

Utilizando MATLAB leemos cada uno de los 10 envases llenos y vacíos las cuales se almacenan en matrices, de cada una de esas matrices se saca el umbral de energía y se delimita el umbral máximo y el umbral mínimo tanto de envases llenos como envases vacíos. En la **Figura 3.24** se muestra el diagrama a bloques y el código que se utilizó para leer y guardar las matrices de las imágenes.

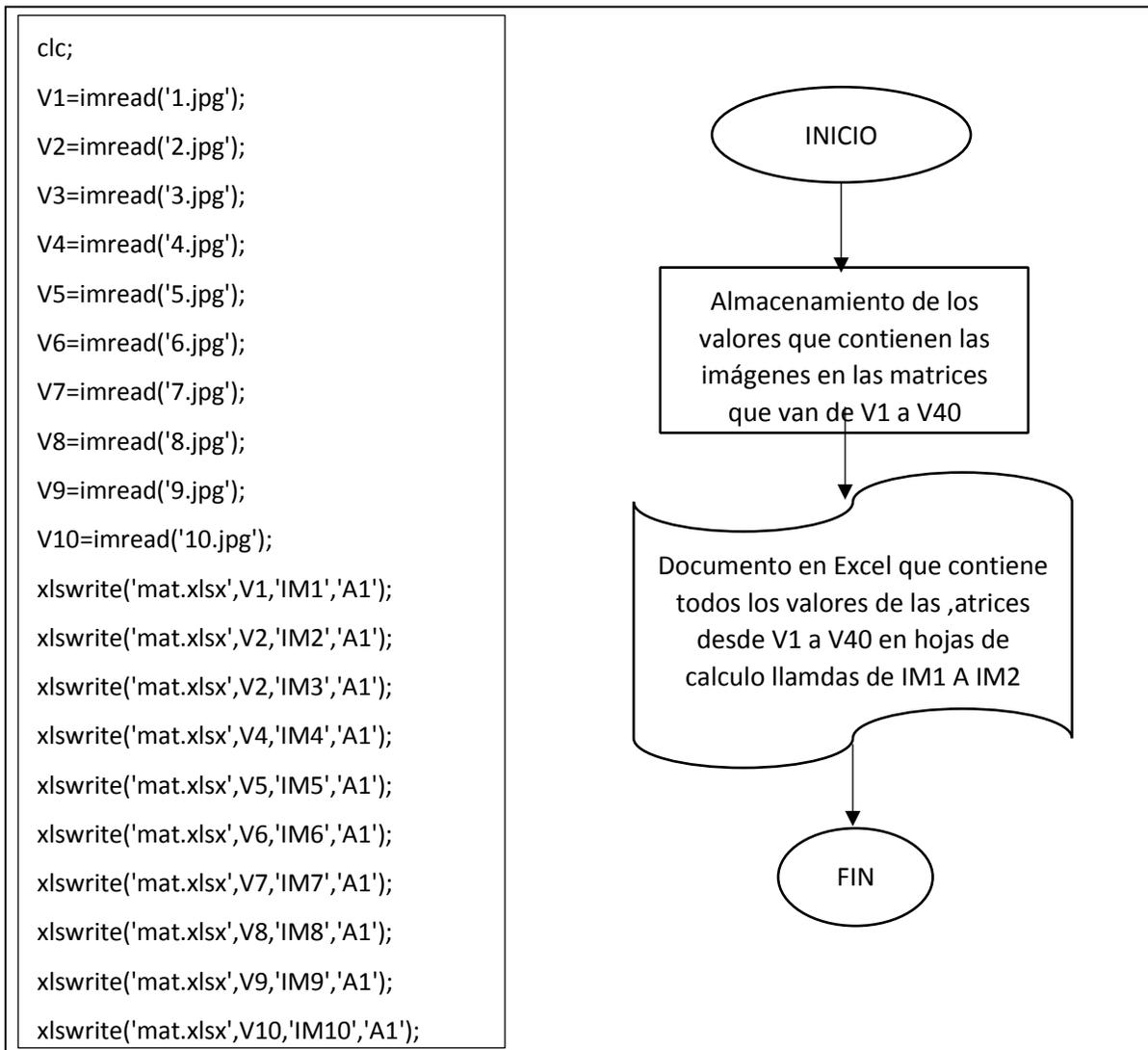
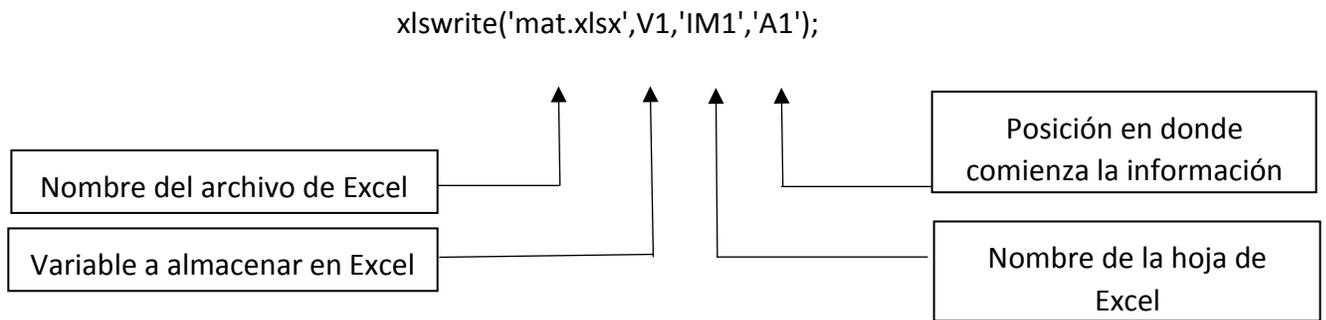


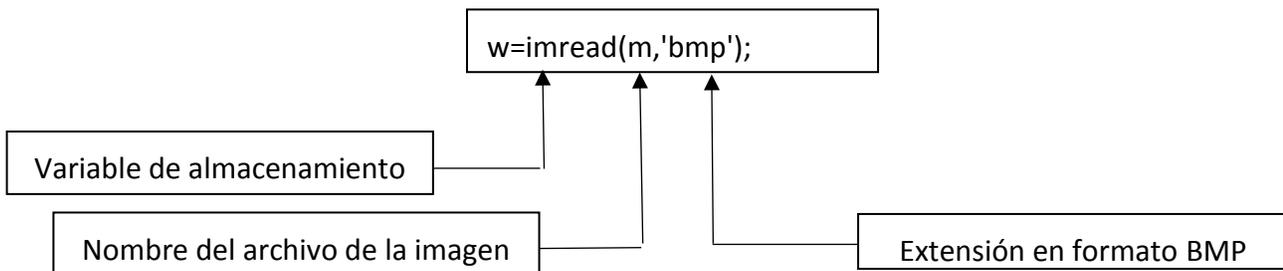
Figura 3.24 Diagrama a bloques código para leer y guardar las matrices de las imágenes.

Para poder crear un archivo de Excel se utiliza la función “xlswrite” con la cual se almacenan los valores que contiene la matriz de cada una de las 10 imágenes utilizadas para el entrenamiento del sistema tanto en envases llenos y 10 imágenes de envases vacíos, en la **Figura 3.25** se muestra el uso de dicha función.



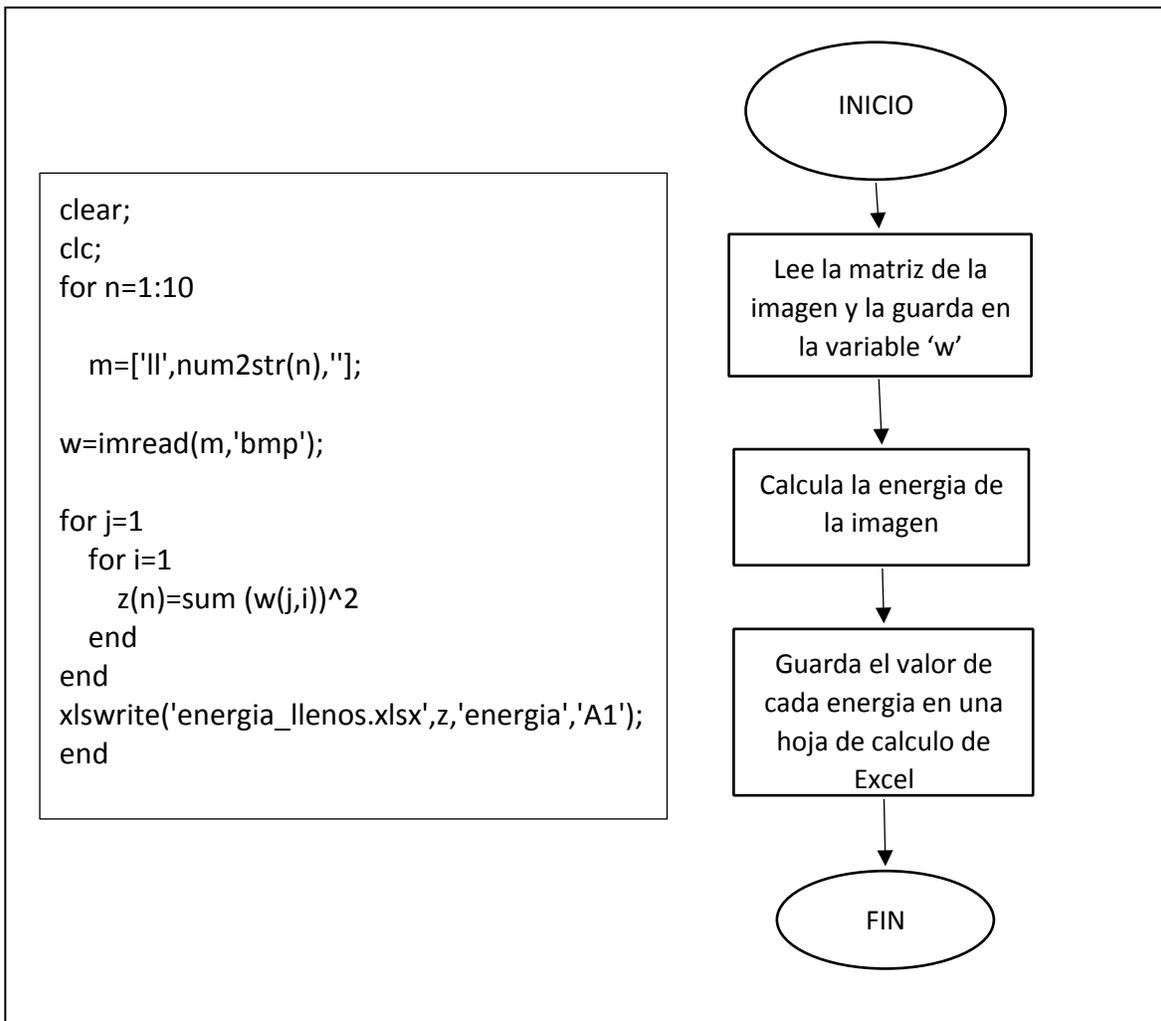
**Figura 3.25** Creación de un archivo de Excel que contiene en una hoja de cálculo los valores de las matrices de cada una de las imágenes.

Para obtener la energía de cada imagen capturada tanto de envases llenos como de envases vacíos utilizamos los valores de la matriz de cada una de las imágenes las cuales son leídas por Matlab, son almacenados en una variable y utilizando la fórmula para calcular la energía de una imagen obtenemos la cantidad de energía contenida por la imagen a calcular, el comando utilizado para leer las matrices de cada imagen es “imread” el cual se asigna a una variable para poder ser procesada en Matlab como se muestra en la **Figura 3.26**.

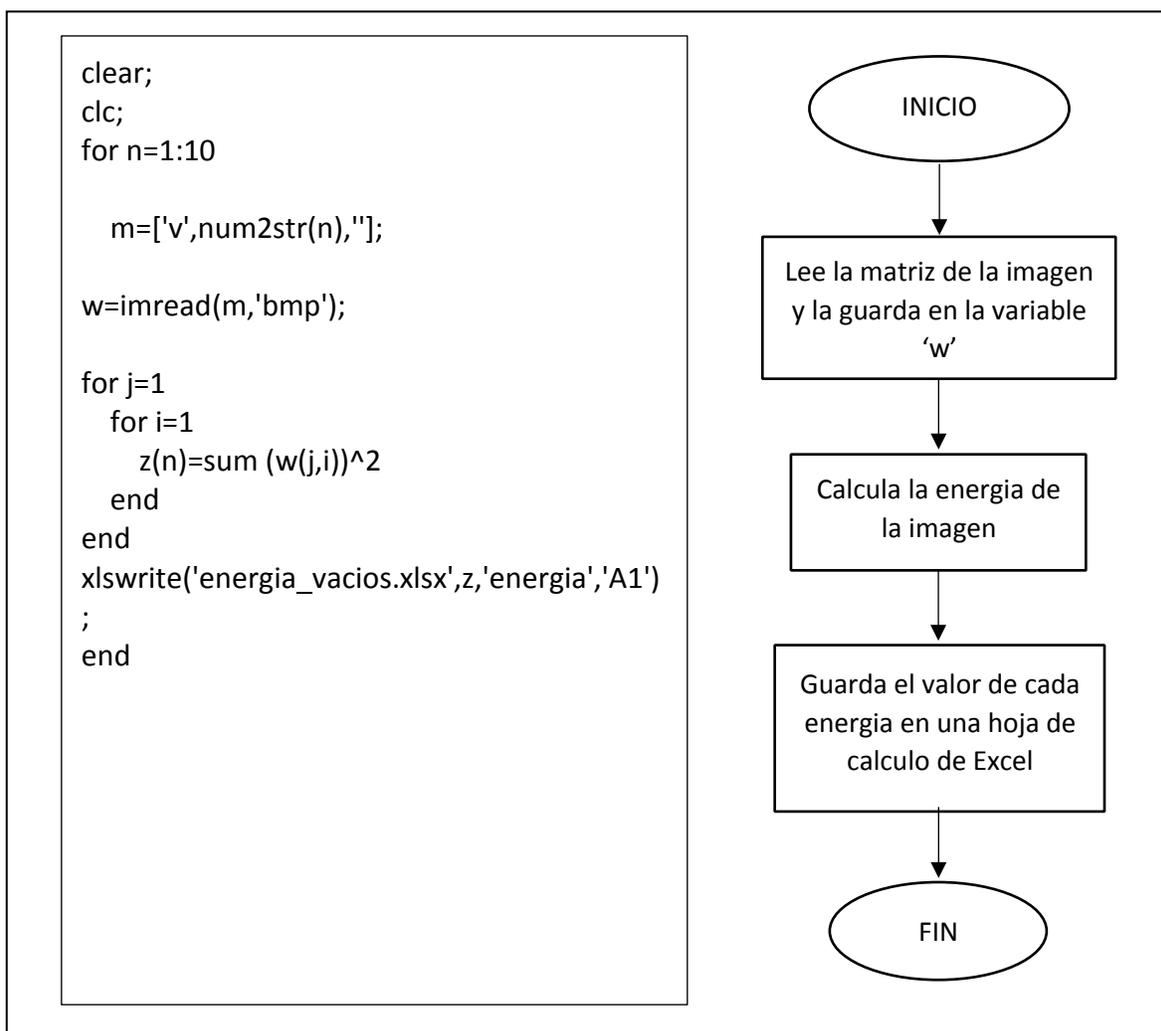


**Figura 3.26** Lectura de las matrices de las imágenes en formato BMP.

En la **Figura 3.27** y en la **Figura 3.28** se muestra el diagrama a bloques y el código del programa que lee la matriz de la imagen y calcula la de energía de la misma tanto en envases llenos como en envases vacíos.



**Figura 3.27** Diagrama a bloques y código de programa que lee la matriz de la imagen de envases llenos y saca la energía de la misma.



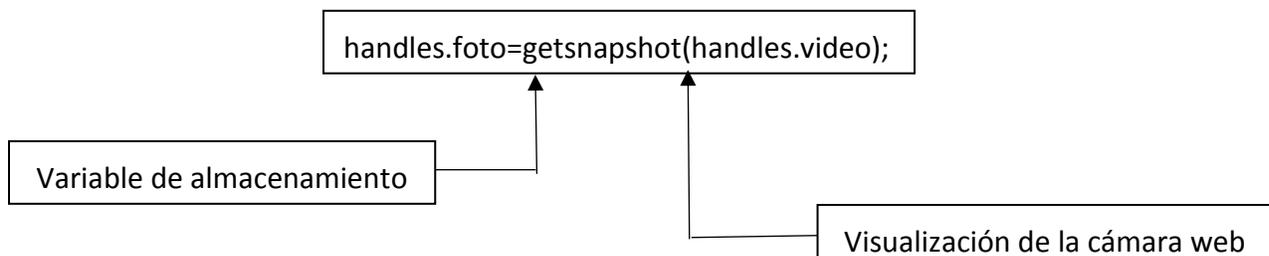
*Figura 3.28 Diagrama a bloques y código de programa que lee la matriz de la imágenes de envases vacíos y saca la energía de la misma.*

El resultado obtenido se guarda en una variable y de esta forma sacamos el umbral máximo y el umbral mínimo de cada uno de las 10 imágenes tomadas de envases vacíos como el

umbral máximo y el umbral mínimo de cada una de las 10 imágenes tomadas de envases llenos.

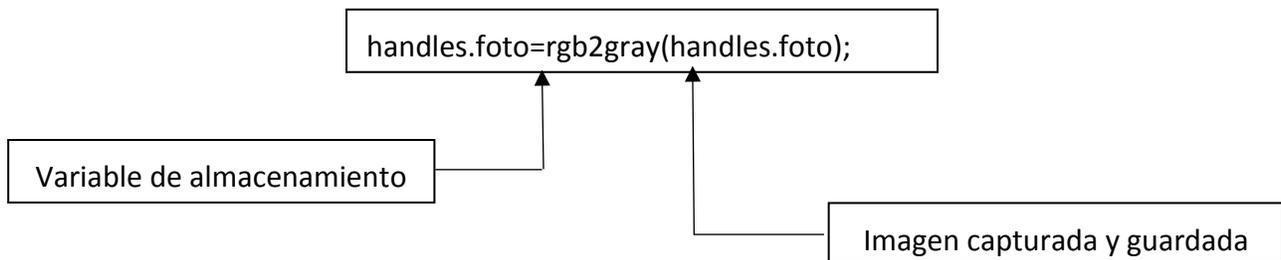
La etapa de verificación consiste en capturar una imagen a tiempo real, convertir la imagen capturada a escala de grises y comparar la imagen capturada contra el umbral de energía, para después verificar si el resultado se encuentra dentro del umbral máximo y el umbral mínimo ya sea de los envases llenos o de los envases vacíos.

Utilizamos la banda transportadora para transportar los envases de vidrio y en nuestra interfaz mediante la cámara web podremos observar como el envase se pone frente a la cámara y de esta manera con el botón “Tomar Foto” capturamos la imagen a procesar, para capturar la imagen utilizamos el comando “getsnapshot” el realiza la función de capturar lo que se está viendo en la web cam y lo guarda en una variable como se muestra en la **Figura 3.29.**



*Figura 3.29. Captura de imagen*

Utilizamos el botón “Convertir” para convertir la imagen a escala de grises y se utiliza el comando “rgb2gray” el cual lee la imagen guardada previamente en una variable, la convierte a escala de grises y la guarda en una nueva variable. El funcionamiento de este comando se muestra en la **Figura 3.30.**



*Figura 3.30 Convertir a escala de grises*

Utilizamos el botón de “Calcular Energía” para calcular la energía de la imagen capturada a tiempo real y con este mismo botón se compara con los umbrales obtenidos y determina si el envase está lleno o está vacío, al determinarlo la interfaz nos reproducirá un archivo con extensión .WAV que nos dirá si el envase está lleno o está vacío. Para reproducir el archivo con extensión .WAV se utiliza el comando “wavread” el cual lee un archivo con extensión .WAV y lo reproduce.

En la Figura 3.31. se muestra el diagrama a bloques y parte del código que nos permite calcular la energía de la imagen capturada a tiempo real y compararla con los umbrales de energía previamente adquiridos.

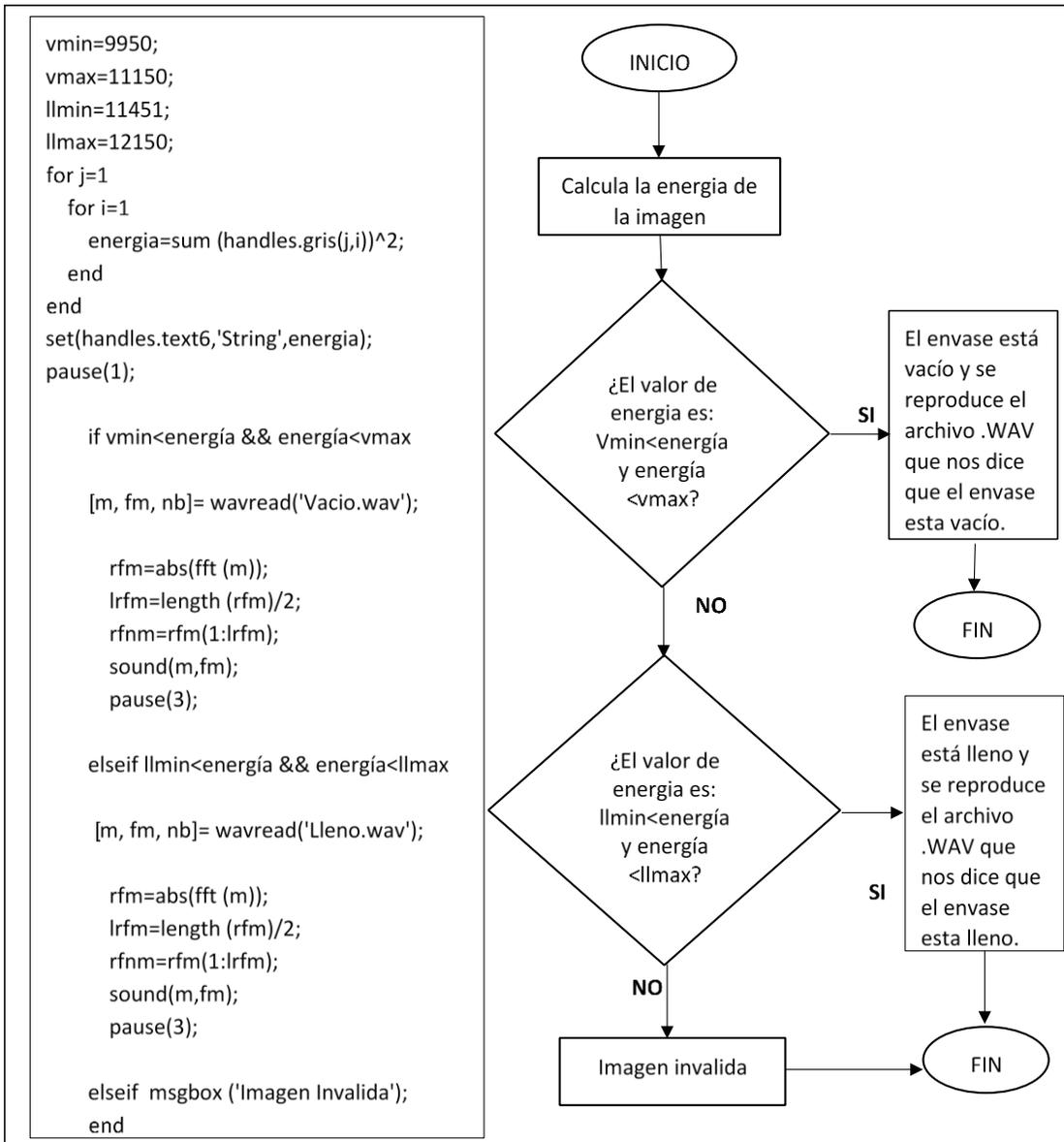


Figura 3.31 Diagrama a bloques y código del cálculo de la energía de la imagen capturada a tiempo real y comparada con los umbrales de energía para determinar si el envase está lleno o vacío.

### *3.7 Proceso de Evaluación del Sistema*

Como ya se ha mencionado, las imágenes tienen características que las representan, como es la intensidad en cada uno de sus píxeles, los cuales son representados en bits, así como la resolución espacial, tipo de imagen, paleta de colores, tamaño, la estructura en bytes, el largo y ancho de la imagen entre otras. Partiendo de estas características, podemos comparar las diferentes intensidades de cada píxel entre dos o más imágenes. En nuestro caso se comparan las imágenes obtenidas de los envases de vidrio tanto llenos como vacíos capturados por una cámara web.

El método propuesto en esta investigación es utilizando la energía de cada imagen, con el cual compararemos la energía de una imagen capturada a tiempo real con los umbrales previamente calculados y determinar si el envase está lleno o está vacío.

Para el uso del prototipo propuesto, es necesaria una etapa de entrenamiento, la cual consiste en los siguientes pasos:

1. Crear una base de datos con las matrices de 10 imágenes tanto de envases llenos como de envases vacíos.
2. Calcular la energía de cada una de las imágenes de los 10 envases llenos como de las imágenes de los 10 envases vacíos.
3. Determinar los umbrales máximos y mínimos de las 10 imágenes de envases llenos y las 10 imágenes de envases vacíos.

Una vez entrenado el sistema, se puede proceder al uso del mismo, los pasos a realizar para la verificación del llenado de los envases son los siguientes:

1. Transportar por la banda transportadora un envase ya sea lleno o vacío hasta que se encuentre en la visión de la web cam.
2. Adquirir la imagen de un envase ya sea lleno o vacío en tiempo real.
3. Convertir la imagen capturada a escala de grises.
3. Calcular la energía de la imagen capturada.
4. Mostrar la energía de la imagen capturada.

# Capítulo 4 Análisis de Resultados

En este capítulo se muestran los resultados obtenidos del sistema de verificación de llenado de envases propuesto. Inicialmente se muestran los resultados obtenidos del sistema de 10 envases en línea, comparando los umbrales de energía obtenidos mediante el algoritmo de entrenamiento del sistema para la verificación del nivel de llenado de envases de vidrio mediante distancias euclidianas y energía previamente desarrollado, el objetivo de éste es que sea utilizable en cualquier recinto con el menor número de fallas.

Posteriormente,

## 4.1 Resultados para una línea de 10 envases llenos y 10 envases vacíos

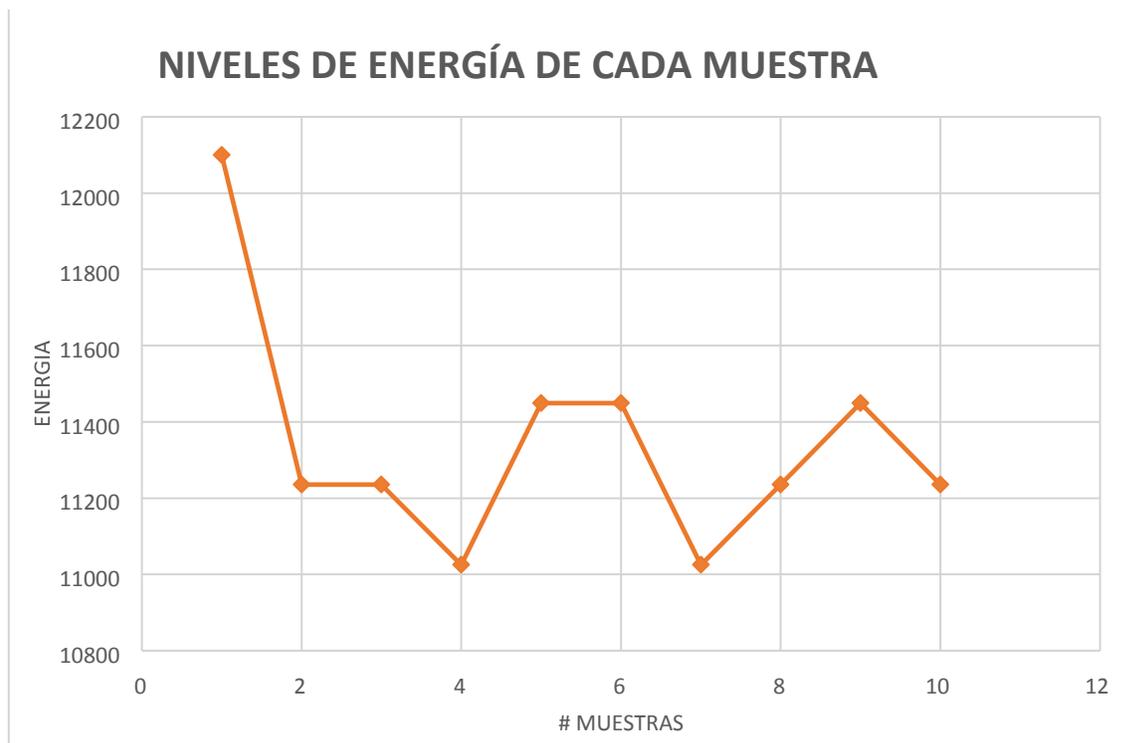
Para el correcto funcionamiento del sistema, fue necesaria una etapa de entrenamiento, en la cual se obtuvieron los valores de energía para delimitar los umbrales máximos y mínimos tanto para envases llenos como para envases vacíos; Estos valores de umbral se muestran en las siguientes tablas.

*Tabla 4.1 Niveles de energía de 10 envases llenos*

# de muestra	Niveles de energía de cada muestra
1	12100
2	11236
3	11236
4	11025
5	11449
6	11449

<b>7</b>	11025
<b>8</b>	11236
<b>9</b>	11449
<b>10</b>	11236

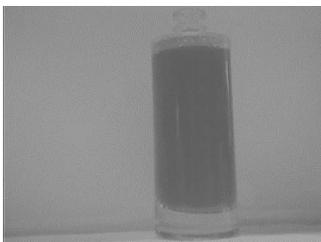
En la **Figura 4.1** se observa el comportamiento de la energía para cada muestra utilizada para el entrenamiento del sistema en el apartado de envases llenos.



**Figura 4.1** Comportamiento de las energías de envases

En la **Tabla 4.1** se muestran los niveles de energía para cada muestra, en este caso son envases llenos, la energía es calculada por el algoritmo previamente desarrollado. Con estos resultados podemos determinar que el valor máximo es de 12100 y el mínimo de 11025 y con esto tenemos nuestro umbral máximo y umbral mínimo.

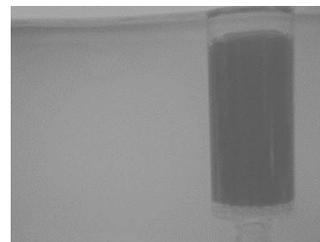
A continuación, se muestran las imágenes de envases llenos obtenidas por el sistema correspondiente a las energías de la **Tabla 4.1**.



*Figura 4.2 Envase #1 lleno*



*Figura 4.3 Envase #2 lleno*



*Figura 4.4 Envase #3 lleno*



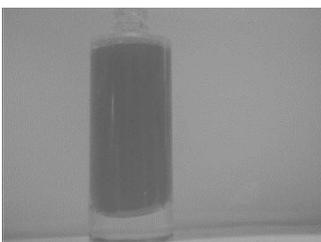
*Figura 4.5 Envase #4 lleno*



*Figura 4.6 Envase #5 lleno*



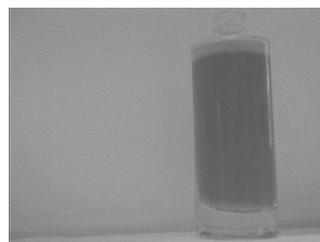
*Figura 4.7 Envase #6 lleno*



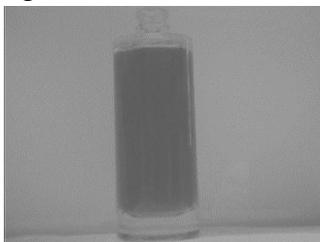
*Figura 4.8 Envase #7 lleno*



*Figura 4.9 Envase #8 lleno*



*Figura 4.10 Envase #9 lleno*

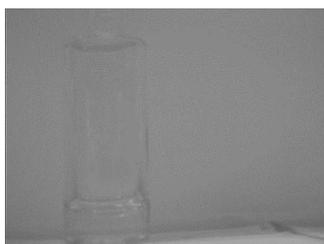


*Figura 4.11 Envase #10 lleno*

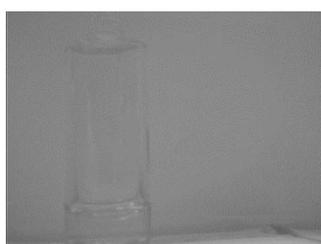
**Tabla 4.2** Niveles de energía de 10 envases vacíos

# de muestra	Niveles de energía de cada muestra
1	10404
2	10201
3	10201
4	10000
5	10201
6	10404
7	10609
8	10201
9	10000
10	10201

A continuación, se muestran las imágenes de envases vacíos obtenidas por el sistema correspondiente a las energías de la tabla 4.2.



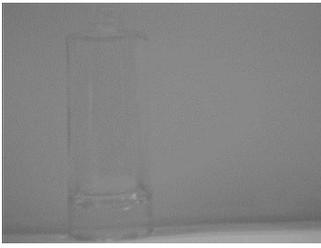
*Figura 4.12* Envase #1 vacío



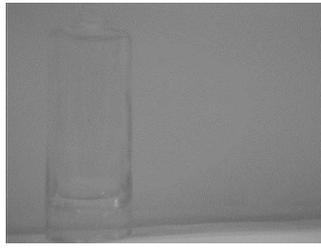
*Figura 4.13* Envase #2 vacío



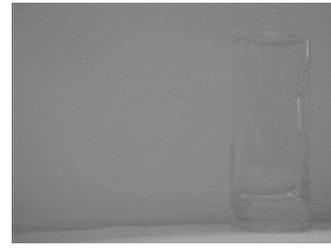
*Figura 4.14* Envase #3 vacío



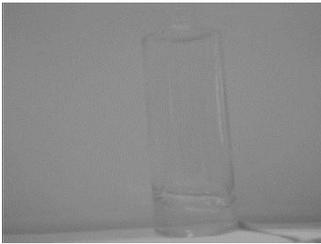
*Figura 4.15 Envase #4 vacío*



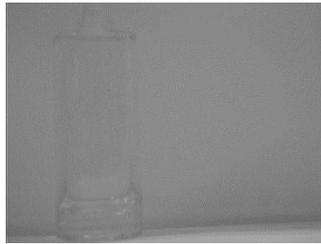
*Figura 4.16 Envase #5 vacío*



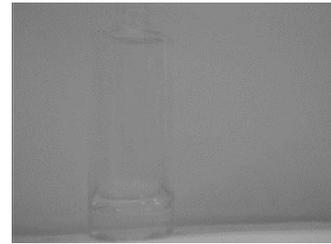
*Figura 4.17 Envase #6 vacío*



*Figura 4.18 Envase #7 vacío*



*Figura 4.19 Envase #8 vacío*

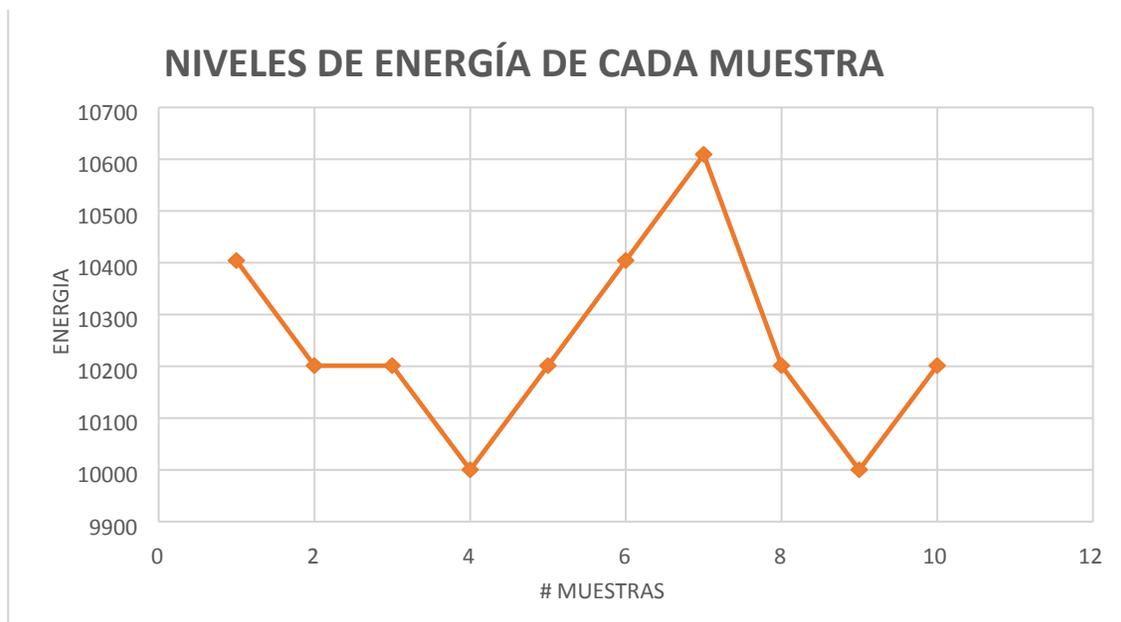


*Figura 4.20 Envase #9 vacío*



*Figura 4.21 Envase #10 vacío*

En la **Figura 4.22** se observa el comportamiento de la energía para cada muestra utilizada para el entrenamiento del sistema en el apartado de envases vacíos.



*Figura 4.22 Comportamiento de las energías de envases vacíos*

En la **Tabla 4.2** se muestran los niveles de energía para cada muestra, en este caso son envases vacíos, la energía es calculada por el algoritmo previamente desarrollado. Con estos resultados podemos determinar que el valor máximo es de 10404 y el mínimo de 10000 y con esto tenemos nuestro umbral máximo y umbral mínimo.

#### 4.2 Interpretación de los resultados

De los resultados de las tablas anteriores, se puede observar que las energías tanto de las imágenes de envases llenos como las imágenes de envases vacíos tienen un amplio margen unas de las otras, podemos observar que el umbral máximo de envases vacíos (10404) y el umbral mínimo de envases llenos (11025) tienen un amplio margen de distancia por lo cual podemos concluir que nuestro sistema no detectaría un envase lleno cuando está vacío o un envase vacío cuando está lleno pero es probable que en algunos casos la cantidad de energía de la imagen no esté entre los umbrales máximo y mínimo de los envases llenos ni tampoco entre los umbrales máximo y mínimo de los envases vacíos. Bajo tal característica,

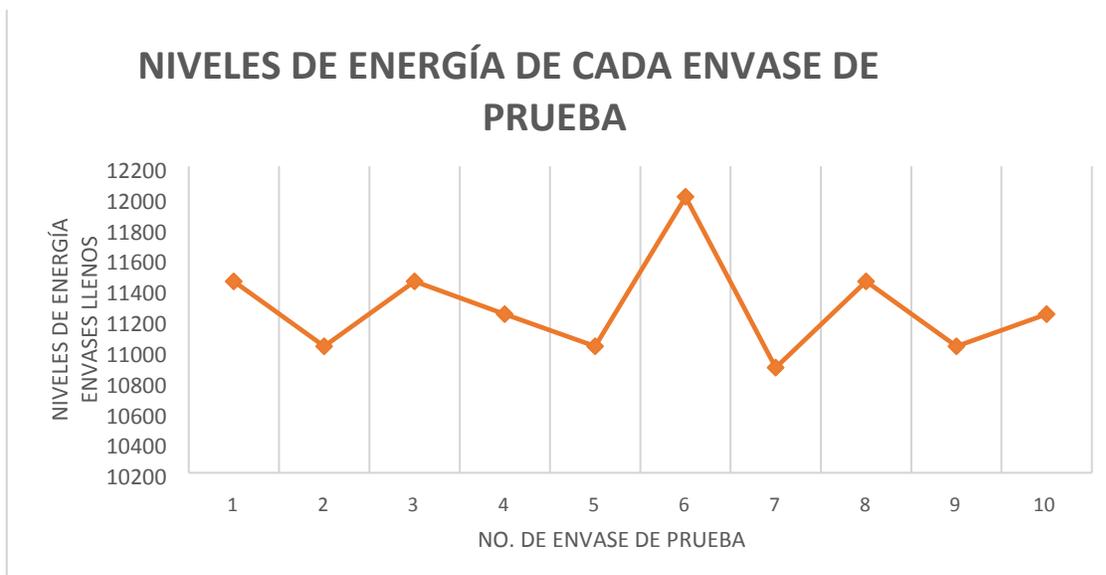
el sistema podría no detectar en algunas ocasiones si el envase está lleno o está vacío. Para verificar lo anterior, se realizaron pruebas a 10 diferentes envases llenos y 10 diferentes envases vacíos, evaluando a través del rango de umbrales de energía tanto de envases llenos como de envases vacíos. Se capturaron con ayuda de la banda transportadora y la cámara web las 10 imágenes de envases llenos, así como las 10 imágenes de envases vacíos las cuales una por una a tiempo real se les calculo la cantidad de energía y fue comparada tanto con los umbrales de envases llenos como de envases vacíos. Los resultados de la cantidad de energía calculada en cada envase tanto llenos como vacíos se muestran en las **Tablas 4.3 y 4.4** respectivamente.

La **Tabla 4.3** representa la cantidad de energía obtenida en envases llenos para cada envase de prueba lleno.

*Tabla 4.3 Niveles de energía de 10 envases de prueba llenos*

# de envase de prueba	Niveles de energía de cada envase de prueba
1	11449
2	11025
3	11449
4	11236
5	11025
6	12000
7	10886
8	11449
9	11025
10	11236

En la **Figura 4.23** se muestra un gráfico del comportamiento de los niveles de energía para cada uno de los envases de prueba llenos.



**Figura 4.23** Comportamiento de las energías de envases de prueba llenos

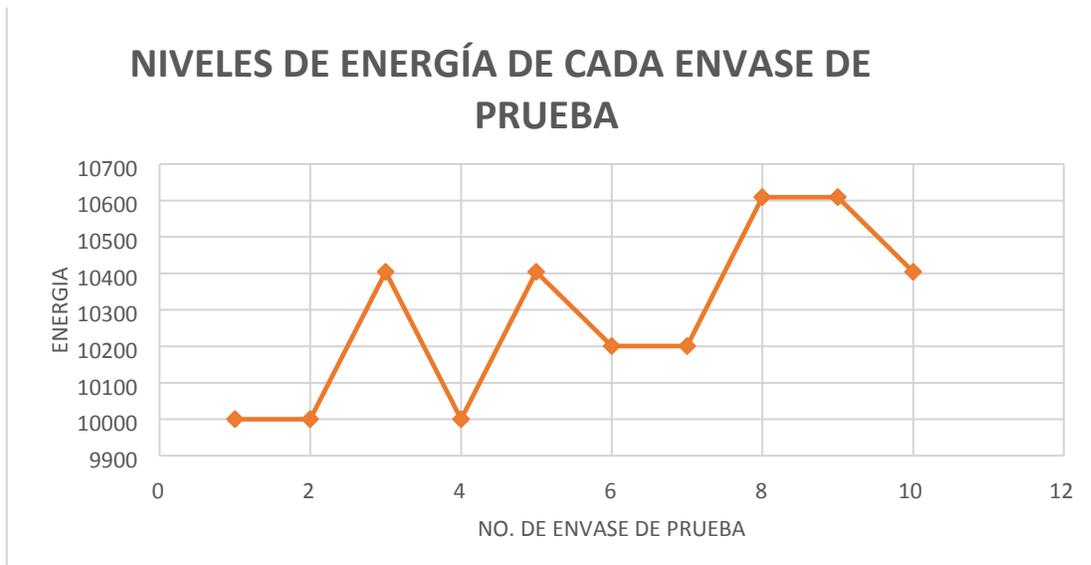
La **Tabla 4.4** representa la cantidad de energía obtenida en envases vacíos para cada envase de prueba vacío.

**Tabla 4.4** Niveles de energía de 10 envases de prueba vacíos

# de envase de prueba	Niveles de energía de cada envase de prueba
1	10000
2	10000
3	10404
4	10000
5	10404
6	10201
7	10201
8	10609

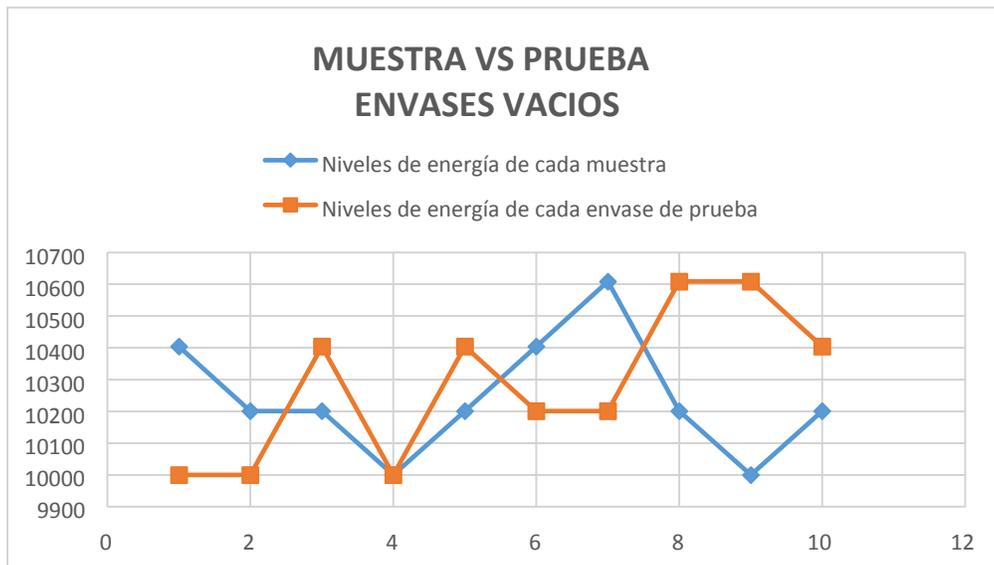
9	10609
10	10404

En la **Figura 4.24** se muestra un gráfico del comportamiento de los niveles de energía para cada uno de los envases de prueba vacíos.



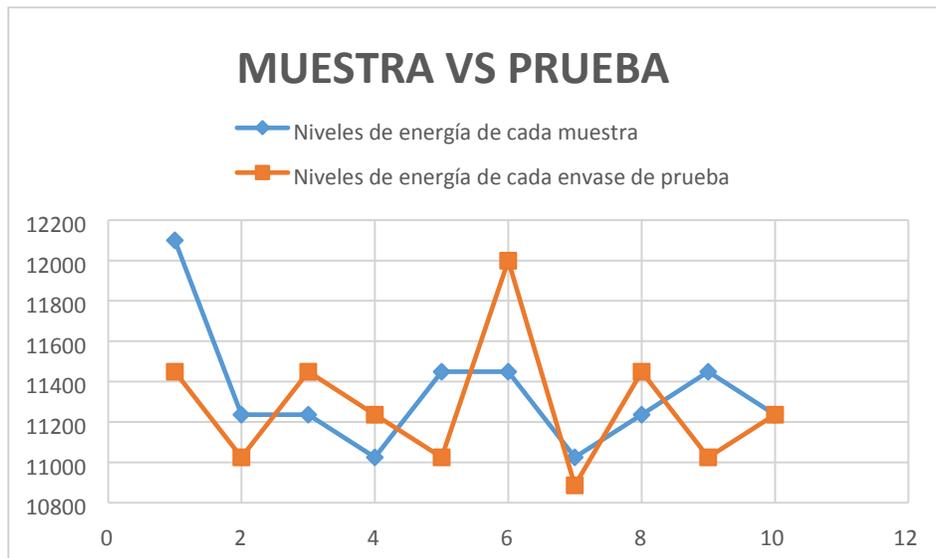
**Figura 4.24** Comportamiento de las energías de envases de prueba vacíos

En la **Figura 4.25** se muestra un gráfico comparativo entre los niveles de energía de las muestras y los niveles de energía de los envases de prueba vacíos.



*Figura 4.25 Energías muestra VS Energías pruebas en envases vacíos.*

En la **Figura 4.26** se muestra un gráfico comparativo entre los niveles de energía de las muestras y los niveles de energía de los envases de prueba llenos.



*Figura 4.26 Energías muestra VS Energías pruebas en envases vacíos.*

Se aprecia que el comportamiento esperado en los envases llenos no se cumplió al 100% como se observa con el envase no. 7 la cantidad de energía obtenida no se encuentra dentro del

umbral calculado previamente. No obstante, con los envases vacíos obtuvimos el 100% de los envases dentro del umbral de energía. Sin embargo, en ambos casos podemos observar que la cantidad de energía obtenida varía de un envase a otro.

En la Tabla 4.5 se muestran los resultados obtenidos de la Prueba No.1 tanto de envases llenos como de envases vacíos, obteniendo 20 resultados.

En la Figura 4.27 se muestra el gráfico de distribución de resultados de prueba 1.

*Tabla 4.5 Resultados prueba 1*

Resultados Prueba 1	
Envases vacíos	10
Envases llenos	9
Imagen inválida	1



**Figura 4.27** Resultados prueba 1.

## Capítulo 5 Conclusiones

De acuerdo a las estadísticas obtenidas en este trabajo se puede concluir que el sistema utilizado para la detección de envases llenos y vacíos podría ser una solución para la mejora de procesos de calidad que la industria requiera. Así como disminuir el riesgo laboral del trabajador al estar expuesto durante periodos prolongados a los sistemas actuales de los departamentos de calidad.

El sistema puede ser funcional no solo para las industrias que utilicen envases de vidrio, si no que puede extenderse a otras ya que se cuenta con un sistema de entrenamiento, lo que nos permite poder tener resultados exitosos, aun cuando los factores externos sean diferentes en cada industria.

# Capítulo 6 Apéndices

## Apéndice A. MATLAB

El nombre MATLAB proviene de la contracción de MATrix LABoratory. MATLAB es un lenguaje de alto nivel orientado al desarrollo de cálculos técnicos.

Integra cálculo, visualización y programación en un entorno interactivo de fácil manejo donde los problemas y las soluciones se expresan en la notación matemática habitual. El elemento de información básico en MATLAB es una tabla a la que no hace falta asignar dimensión con antelación. Esto permite abordar problemas que requieren una formulación vectorial o matricial en mucho menos tiempo de lo que se tardaría con un lenguaje escalar no interactivo como C o FORTRAN.

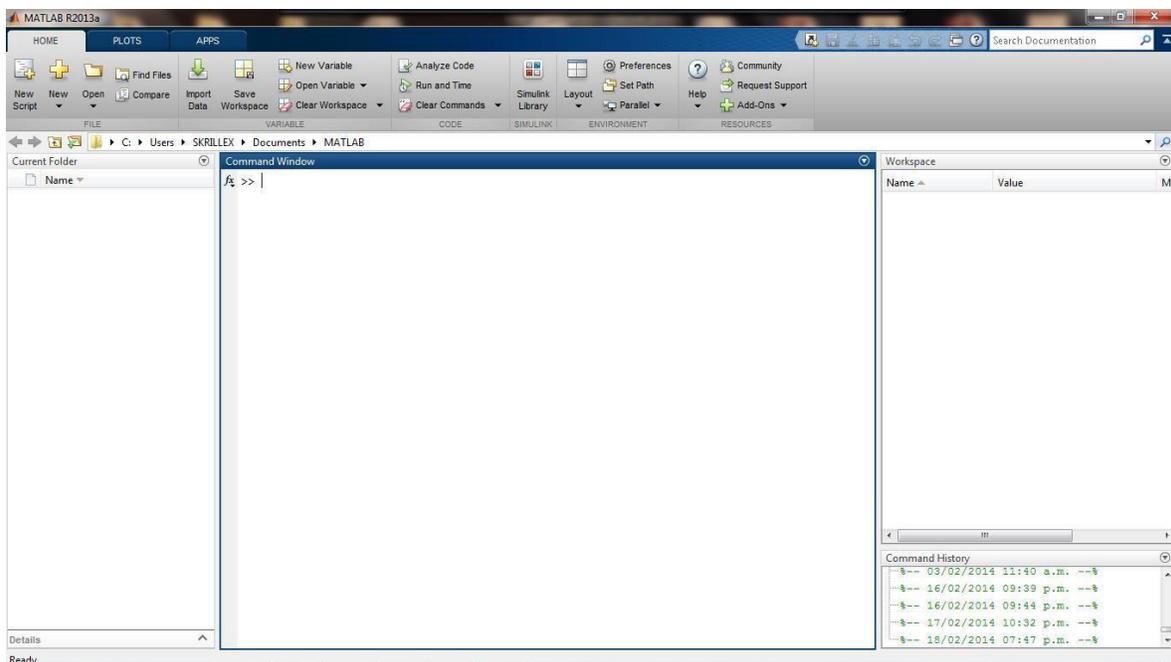
Se pueden resolver problemas numéricos relativamente complejos sin necesidad de escribir un programa para ello. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. Tiene gran capacidad de expansión ya que permite que el usuario defina sus propias funciones, es por ello que es considerado como un lenguaje de programación para cálculos técnicos y científicos donde se encuentran implicados elevados cálculos matemáticos y la visualización gráfica.

MATLAB trabaja esencialmente con un solo tipo de objetos: matrices, con entradas eventualmente complejas. Esto significa que todas las variables representan matrices, de este modo los escalares quedan caracterizados por matrices de  $1 \times 1$  y los vectores, por matrices de una fila o una columna.

Para trabajar con MATLAB se comienza como con cualquier otra aplicación de Windows, haciendo doble clic en el icono correspondiente.

La ventana principal, denominada ventana de orden, es el lugar donde se interacciona con MATLAB. En dicha venta contiene 4 ventanas.

1. Ventana de Command Window en su interior, aparece el prompt característico de MATLAB (>>) que es el indicativo de petición de órdenes.
2. Ventana de Workspace en su interior, aparecerán todas las variables representadas por matrices.
3. Ventana de Command History en su interior, aparecerán todas las instrucciones que se han utilizado desde el inicio y hasta el final del programa.
4. Ventana de Current Folder en su interior, se mostraran todos los archivos que estén en la carpeta donde su ubicara MATLAB y guardan las funciones.



**Figura A.1:** Ventana principal de MATLAB.

Existe una ventana auxiliar que es la ventana de ayuda, que provee enlaces con otras cuestiones relacionadas con la información que se haya solicitado.

## Código fuente

- Código para la captura de imágenes de envases llenos y vacíos para el entrenamiento del sistema:

```
function Interfaz_final_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
handles.energia=[];
handles.foto=[];
handles.noback=[];
handles.gris=[];
guidata(hObject,handles);
vmin=9950;
vmax=11150;
lmin=11441;
lmax=12150;
set(handles.text1,'String',lmin);
set(handles.text2,'String',lmax);
set(handles.text4,'String',vmin);
set(handles.text3,'String',vmax);
try
    handles.video=videoinput('winvideo',2,'YUY2_640x480');
    guidata(hObject, handles);
    VidRes=get(handles.video, 'VideoResolution');
    nBands=get(handles.video, 'NumberOfBands');
    hImage=image(zeros(VidRes(2),VidRes(1),nBands),'Parent',handles.axes1);
    preview(handles.video,hImage);
```

```
catch
```

```
end
```

```
background = imread('ESIME.jpg'); %Leer imagen
```

```
axes(handles.background); %Carga la imagen en background
```

```
axis off;
```

```
imshow(background); %Presenta la imagen
```

```
handles.output = hObject;
```

```
guidata(hObject, handles);
```

```
function varargout = Interfaz_final_OutputFcn(hObject, eventdata, handles)
```

```
varargout{1} = handles.output;
```

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

```
handles.foto=getsnapshot(handles.video);
```

```
guidata(hObject, handles);
```

```
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises
```

```
guidata(hObject, handles);
```

```
imshow(handles.gris,'parent',handles.axes2);
```

```
axes(handles.axes1)
```

```
for j=1
```

```
    for i=1
```

```
        energia2=sum (handles.gris(j,i))^2
```

```
    end
```

```
end
```

```
    set(handles.text10,'String',energia2);
```

***function LLENO\_Callback(hObject, eventdata, handles)***

```
handles.foto=getsnapshot(handles.video);
guidata(hObject, handles);
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises
guidata(hObject, handles);
imshow(handles.gris,'parent',handles.axes2);
axes(handles.axes1)
for j=1
    for i=1
        energia1=sum (handles.gris(j,i))^2
    end
end
set(handles.text9,'String',energia1);
```

***function FOTO\_Callback(hObject, eventdata, handles)***

```
clc;
[m, fm, nb]= wavread('Tomarfoto.wav');
rfm=abs(fft (m));
lrfm=length (rfm)/2;
rfnm=rfm(1:lrfm);
sound(m,fm);
pause(3);
handles.foto=getsnapshot(handles.video);
guidata(hObject, handles);
imshow(handles.foto,'parent',handles.axes2);
axes(handles.axes1)
```

```

function GRIS_Callback(hObject, eventdata, handles)
[m, fm, nb]= wavread('Gris.wav');
rfm=abs(fft (m));
lrfm=length (rfm)/2;
rfnm=rfm(1:lrfm);
sound(m,fm);
pause(3);
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises
guidata(hObject, handles);
imshow(handles.gris,'parent',handles.axes2);
axes(handles.axes1)

```

***function ENERGIA\_Callback(hObject, eventdata, handles)***

```

[m, fm, nb]= wavread('Energia.wav');
rfm=abs(fft (m));
lrfm=length (rfm)/2;
rfnm=rfm(1:lrfm);
sound(m,fm);
pause(3);
vmin=9950;
vmax=11150;
llmin=11441;
llmax=12150;
set(handles.text1,'String',llmin);
set(handles.text2,'String',llmax);
set(handles.text4,'String',vmin);
set(handles.text3,'String',vmax);

```

```

for j=1
    for i=1
        energia=sum (handles.gris(j,i))^2;
    end
end
set(handles.text5,'String',energia);
pause(1);
if vmin<energia && energia<vmax
    [m, fm, nb]= wavread('Vacio.wav');
    rfm=abs(fft (m));
    lrfm=length (rfm)/2;
    rfnm=rfm(1:lrfm);
    sound(m,fm);
    pause(3);
    %r=('envase vacio');
    %msgbox('EL ENVASE ESTA VACIO');
elseif lmin<energia && energia<llmax
    [m, fm, nb]= wavread('Lleno.wav');
    rfm=abs(fft (m));
    lrfm=length (rfm)/2;
    rfnm=rfm(1:lrfm);
    sound(m,fm);
    pause(3);
    %r=('envase lleno');
    %msgbox('EL ENVASE ESTA LLENO');
elseif vmax<energia && energia<lmin

```

```
msgbox('EL ENVASE NO ESTA NI LLENO NI VACIO');  
elseif msgbox ('Imagen Invalida');  
end
```

***function FOTO\_1\_Callback(hObject, eventdata, handles)***

```
clc;  
[m, fm, nb]= wavread('Tomarfoto.wav');  
rfm=abs(fft (m));  
lrfm=length (rfm)/2;  
rfnm=rfm(1:lrfm);  
sound(m,fm);  
pause(3);  
handles.foto=getsnapshot(handles.video);  
guidata(hObject, handles);  
imshow(handles.foto,'parent',handles.axes2);  
axes(handles.axes1)
```

***function CONVERTIR\_1\_Callback(hObject, eventdata, handles)***

```
[m, fm, nb]= wavread('Gris.wav');  
rfm=abs(fft (m));  
lrfm=length (rfm)/2;  
rfnm=rfm(1:lrfm);  
sound(m,fm);  
pause(3);  
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises  
guidata(hObject, handles);
```

```
imshow(handles.gris,'parent',handles.axes2);  
axes(handles.axes1)
```

```
function ENERGA_1_Callback(hObject, eventdata, handles)
```

```
[m, fm, nb]= wavread('Energia.wav');  
rfm=abs(fft (m));  
lrfm=length (rfm)/2;  
rfnm=rfm(1:lrfm);  
sound(m,fm);  
pause(3);  
for j=1  
    for i=1  
        energia=sum (handles.gris(j,i))^2;  
    end  
end  
set(handles.text5,'String',energia);  
llenno=str2double(get(handles.text9,'String'));  
vacio=str2double(get(handles.text10,'String'));  
a=abs(lleno-energia)  
b=abs(vacio-energia)  
c=energia-llenno  
d=energia-vacio  
pause(1);  
    if b<a  
        [m, fm, nb]= wavread('Vacio.wav');
```

```

    rfm=abs(fft (m));
    lrfm=length (rfm)/2;
    rfnm=rfm(1:lrfm);
    sound(m,fm);
    pause(3);
    %r=('envase vacio');
    %msgbox('EL ENVASE ESTA VACIO');
else
    [m, fm, nb]= wavread('Lleno.wav');
    rfm=abs(fft (m));
    lrfm=length (rfm)/2;
    rfnm=rfm(1:lrfm);
    sound(m,fm);
    pause(3);
    %r=('envase lleno');
    %msgbox('EL ENVASE ESTA LLENO');
end

```

***function ATM\_1\_Callback(hObject, eventdata, handles)***

```

handles.foto=getsnapshot(handles.video);
guidata(hObject, handles);
imshow(handles.foto,'parent',handles.axes2);
axes(handles.axes1)
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises
guidata(hObject, handles);
imshow(handles.gris,'parent',handles.axes2);
axes(handles.axes1)

```

```

for j=1
    for i=1
        energia=sum (handles.gris(j,i))^2;
    end
end
set(handles.text5,'String',energia);
lleno=str2double(get(handles.text9,'String'));
vacio=str2double(get(handles.text10,'String'));
a=abs(lleno-energia)
b=abs(vacio-energia)
c=energia-llenno
d=energia-vacio
pause(1);
    if b<a
        [m, fm, nb]= wavread('Vacio.wav');
        rfm=abs(fft (m));
        lrfm=length (rfm)/2;
        rfnm=rfm(1:lrfm);
        sound(m,fm);
        pause(3);
        %r=('envase vacio');
        %msgbox('EL ENVASE ESTA VACIO');
    else
        [m, fm, nb]= wavread('Lleno.wav');
        rfm=abs(fft (m));
        lrfm=length (rfm)/2;
        rfnm=rfm(1:lrfm);

```

```

    sound(m,fm);
    pause(3);
    %r=('envase lleno');
    %msgbox('EL ENVASE ESTA LLENO');
end

```

***function ATM\_Callback(hObject, eventdata, handles)***

```

handles.foto=getsnapshot(handles.video);
guidata(hObject, handles);
imshow(handles.foto,'parent',handles.axes2);
axes(handles.axes1)
handles.gris=rgb2gray(handles.foto);% convierte la imagen a escala de grises
guidata(hObject, handles);
imshow(handles.gris,'parent',handles.axes2);
axes(handles.axes1)
vmin=9950;
vmax=11150;
lmin=11441;
lmax=12150;
set(handles.text1,'String',lmin);
set(handles.text2,'String',lmax);
set(handles.text4,'String',vmin);
set(handles.text3,'String',vmax);
for j=1
    for i=1

```

```

        energia=sum (handles.gris(j,i))^2;
    end
end
set(handles.text5,'String',energia);
pause(1);
    if vmin<energia && energia<vmax
        [m, fm, nb]= wavread('Vacio.wav');
        rfm=abs(fft (m));
        lrfm=length (rfm)/2;
        rfnm=rfm(1:lrfm);
        sound(m,fm);
        pause(3);
        %r=('envase vacio');
        %msgbox('EL ENVASE ESTA VACIO');
    elseif lmin<energia && energia<llmax
        [m, fm, nb]= wavread('Lleno.wav');
        rfm=abs(fft (m));
        lrfm=length (rfm)/2;
        rfnm=rfm(1:lrfm);
        sound(m,fm);
        pause(3);
        %r=('envase lleno');
        %msgbox('EL ENVASE ESTA LLENO');
    elseif vmax<energia && energia<lmin
        msgbox('EL ENVASE NO ESTA NI LLENO NI VACIO');
    elseif msgbox ('Imagen Invalida');
    end

```

# Capítulo 7 Bibliografía

- [1] Rafael C. González, *Digital Image Processing Third edition*, Pearson International Edition, pp. 58-80, 2007.
- [2] Patrick J. Van Fleet, Wiley- Interscience, *Discrete Wavelet Transformations An elementary approach with applications*, pp 50-54, 2008.
- [3] Havard Rue, Leonhard Held Gaussian, *Markov Random Fields: Theory and Applications*, Chapman & Hall/ CRC
- [4] Castleman, Kenneth R. *Digital image processing*. Prentice-Hall. Englewood Cliffs., N.J., 1979.
- [5] Don Pearson, *Image processing*. Mc Graw Hill. London -New York, 1991
- [6] Green, William B. *Digital image processing : a systems approach*. Van Nostrand Reinhold Co .New York , 1989.
- [7] Rabbani, Majid; Jones, Paul W.. *Digital Image Compression Techniques*. Spie Optical Engineering Press. Bellingham, Wash. , 1991
- [8] José Jaime Esqueda Elizondo, *Fundamentos de procesamiento de imágenes*, Universidad Autónoma de Baja California, 2005.
- [9] *Formatos de imagen*, 2008, Artículo recuperado en: [http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/image\\_n0105.html](http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/image_n0105.html)

Fecha de consulta: 06/01/2016

- [10] Barbeira, María Pilar, *Fundamentos de la Imagen Digital* ,2011, Artículo recuperado en [http://valijas.ceibal.edu.uy/files/imagen\\_digital\\_distribucion.pdf](http://valijas.ceibal.edu.uy/files/imagen_digital_distribucion.pdf) Fecha de consulta: 11/Marzo/2016

- [11] El formato BMP, Marzo 2016, Artículo recuperado en:  
<http://es.kioskea.net/contents/719-el-formato-bmp>  
Fecha de consulta: 18/03/2016
- [12] Estructura de un archivo BMP, Julio 2012, Artículo recuperado en:  
<http://imagenesupb.blogspot.mx/2012/07/estructura-de-un-archivo-bmp.html> Fecha de Consulta: Abril 2016
- [13] El formato BMP, Marzo 2016, Artículo recuperado en:  
<http://es.ccm.net/contents/719-el-formato-bmp> Fecha de consulta:  
18/Marzo/2016
- [14] Eusebio de la Fuente López y Félix Miguel Trespaderne, VISION ARTIFICIAL INDUSTRIAL. *Procesamiento de Imágenes para Inspección Industrial y Robótica*.  
Valladolid - Secretariado de Publicaciones e Intercambio Editorial, Universidad de Valladolid (2012), Capítulo 4, Artículo recuperado en:  
<http://www.librovision.eii.uva.es/pdf/cap4.pdf> Fecha  
de consulta: Abril/2016
- [15] [http://rd.udb.edu.sv:8080/jspui/bitstream/11715/283/1/034512\\_tesis.pdf](http://rd.udb.edu.sv:8080/jspui/bitstream/11715/283/1/034512_tesis.pdf)
- [16] Jorge Hernández-Constante, Josué Álvarez-Borrego, Universidad de Guadalajara, División de Electrónica y Computación, *Inspección de envases vacíos mediante técnicas de histogramas*, Octubre 2010, Artículo recuperado en:  
<http://usuario.cicese.mx/~josue/pdf/Paginas%20desdeMemoriasSenie2010.pdf> Fecha de consulta: 20/Abril/2016
- [17] Thermo Fisher Scientific S.A, *Equipos de inspección de envase*, Octubre 2008, Artículo recuperado en:  
<http://www.industriaalimenticia.com/articles/83519equipos-de-inspeccion-de-envase> Fecha de consulta:

[18] Mettler Toledo ,Sistemas de inspección de rayos X SideChek, 2016, Artículo recuperado en : [http://mx.mt.com/mx/es/home/products/Product-Inspection\\_1/safeline-x-ray-inspection/glass-x-ray-inspection.html](http://mx.mt.com/mx/es/home/products/Product-Inspection_1/safeline-x-ray-inspection/glass-x-ray-inspection.html)

Fecha de consulta: 24/Abril/2016