

**MONITOREO DE FALLAS EN MAQUINARIA TÉRMICA MEDIANTE  
TERMOGRAFÍA INFRARROJA**

**GONZALO ALBERTO DÍAZ RUIZ  
AUDEMAR DAMIAN VEGA GONZALEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA**

**2012**

**MONITOREO DE FALLAS EN MAQUINARIA TÉRMICA MEDIANTE  
TERMOGRAFÍA INFRARROJA**

**GONZALO ALBERTO DÍAZ RUIZ  
AUDEMAR DAMIAN VEGA GONZALEZ**

**Trabajo de grado para optar al título de  
Ingeniero Mecánico**

**Director  
JABID EDUARDO QUIROGA MENDEZ  
Ingeniero Mecánico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA**

**2012**

*A DIOS porque gracias a él tengo familia, amigos, amigas y mis estudios, que ellos forman mi pilar de vida.*

*A mi familia que sin su apoyo en momentos críticos quizás no hubiese salido adelante, y que gracias a los consejos de mi padre y de mi madre a ellos principalmente les doy las gracias.*

*Quiero de gran manera agradecer al Ingeniero Jabid Quiroga por su paciencia y su colaboración en el transcurso de la elaboración del Proyecto de Grado.*

*Gracias a mis amigos que me apoyaron a lo largo de mi vida universitaria, y que de mi parte siempre tendrán a alguien incondicional.*

*ATTE: Gonzalo Alberto Díaz Ruiz*

*A mí madre Marlen.*

*Por su apoyo incondicional, por la motivación que siempre me  
brindo para seguir adelante, gracias madre por creer en mí, por  
darme consejos durante todo mi período de estudio.*

*A mí padre Audemar.*

*Por su perseverancia y constancia que lo caracterizan, por el  
ejemplo que me dio, de seguir adelante a pesar de las adversidades,  
por su apoyo durante toda mi formación profesional.*

*ATT: AUDEMAR VEGA GONZALEZ*

## CONTENIDO

	<b>Pág.</b>
<b>INTRODUCCIÓN</b>	<b>20</b>
<b>1. CONCEPTOS FUNDAMENTALES</b>	<b>23</b>
<b>1.1 CARACTERISTICAS DEL SISTEMA</b>	<b>23</b>
<b>1.2 MATLAB Y EL TRATAMIENTO DE IMÁGENES</b>	<b>23</b>
1.2.1 Lectura de imágenes a través de archivo	25
1.2.2 Acceso a planos RGB	28
1.2.3 Histograma de una imagen	28
1.2.4 Herramientas de Procesamiento de imágenes	29
1.2.4.1 Extracción de Bordes	30
1.2.4.2 Binarización	31
1.2.4.3 Filtraje	32
1.2.4.4 Dilatación y Erosión	33
1.2.4.5 Llenado de Espacios	34
1.2.4.6 Eliminación de Áreas	35
1.2.4.7 Etiquetado y propiedades de una imagen	36
<b>1.3 TRATAMIENTO DE VIDEO</b>	<b>37</b>
1.3.1 Leer video	37
1.3.2 Propiedades del video	38
1.3.3 Preasignar estructura del video	39
<b>1.4 SENTENCIAS DE CONTROL DE FLUJO</b>	<b>39</b>
1.4.1 Bucles FOR	40
1.4.2 Estructuras IF-ELSE-END	40
1.4.3 Estructura CASE	41
<b>1.5 INTERFAZ GUIDE</b>	<b>41</b>
1.5.1 Componentes de una GUI	44
1.5.2 Propiedades de los Componentes	44
<b>1.6 TERMOGRAFIA INFRARROJA</b>	<b>47</b>
1.6.1 Cámara Termográfica	49
1.6.1.1 Funcionamiento	50
1.6.1.2 Características Principales	50
1.6.1.3 Calibración	51
1.6.1.4 Software de Tratamiento de Imágenes	52
1.6.1.5 Herramientas	53

1.6.2	Imagen Infrarroja	55
1.6.2.1	Ergonomía en el manejo de la Cámara	57
1.6.2.2	Accesibilidad	58
1.6.3	Análisis de la Imagen Térmica	59
1.6.3.1	Análisis Cualitativo	59
1.6.3.2	Análisis cuantitativo	61
1.6.4	Compensación	62
1.6.5	Emisividad	63
1.6.5.1	Factores que Afectan la Emisividad	64
1.6.6	Reflejos	67
<b>2.</b>	<b>BANCO PARA PRUEBAS TERMOGRÁFICAS</b>	<b>69</b>
<b>2.1</b>	<b>MÓDULOS DE PRUEBA</b>	<b>69</b>
2.1.1	Módulo de calentamiento de rodamientos	69
2.1.1.1	Selección del material soporte	70
2.1.1.2	Selección de las chumaceras utilizadas	70
2.1.1.3	Selección del acople flexible	72
2.1.2	Modulo intercambiador de calor por aletas	75
2.1.3	Modulo de profundidad, transferencia de calor por conducción	77
2.1.4	Módulo de tamaño	78
2.1.5	Módulo de forma	81
2.1.6	Módulo de puntos	81
<b>2.2</b>	<b>SISTEMA ELÉCTRICO DEL BANCO</b>	<b>84</b>
<b>3.</b>	<b>PROGRAMA DE ANÁLISIS MATLAB</b>	<b>88</b>
<b>3.1</b>	<b>TRATAMIENTO DE IMÁGENES</b>	<b>88</b>
3.1.1	GUIDE	89
3.1.2	Funcionamiento	90
<b>3.2</b>	<b>TRATAMIENTO DE VIDEO</b>	<b>94</b>
3.2.1	GUIDE	94
3.2.2	Funcionamiento	95
<b>4.</b>	<b>PRUEBAS Y MEDICIONES</b>	<b>99</b>
<b>5.</b>	<b>RESULTADOS</b>	<b>108</b>

<b>6.</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>117</b>
	<b>BIBLIOGRAFÍA</b>	<b>118</b>
	<b>ANEXOS</b>	<b>119</b>

## LISTA DE FIGURAS

Figura 1.	Representación de una imagen a escala de grises en MATLAB	24
Figura 2.	Representación de una imagen a color RGB en MATLAB	25
Figura 3.	Imagen mostrada al utilizar la función imshow	27
Figura 4.	Planos de la imagen a) Rojo, b) Verde, c) Azul	28
Figura 5.	Histograma del plano rojo de una imagen	29
Figura 6.	Imagen resultado de la aplicación del algoritmo de canny	30
Figura 7.	Binarización de una imagen	31
Figura 8.	Filtraje de una imagen a) Con ruido, b) Filtrado	32
Figura 9.	Dilatación y Erosión de una Imagen a) Dilatación, b) Erosión	33
Figura 10.	Aplicación de la función Imfill	34
Figura 11.	Aplicación de bwareaopen	35
Figura 12.	Propiedades de una Imagen a) Imagen, b) Workspace, c) Propiedades	36
Figura 13.	Icono GUIDE	42
Figura 14.	Ventana de inicio de GUI	43
Figura 15.	Opción del Componente	45
Figura 16.	Entorno Property Inspector	46
Figura 17.	Espectro Electromagnético	47
Figura 18.	Imagen Termográfica	48
Figura 19.	Cámara Termográfica FLUKE Ti125	49
Figura 20.	Funcionamiento de la Cámara Termográfica	50
Figura 21.	Certificado de Calibración	52

Figura 22. Software FLUKE SMARTVIEW 3.2	53
Figura 23. Rango, campo y nivel	54
Figura 24. Imagen termográfica con puntos fijos, caliente, frío e isothermas	55
Figura 25. Misma imagen infrarroja, con el mismo rango pero con distinto campo	57
Figura 26. Ergonomía	58
Figura 27. Accesibilidad	59
Figura 28. Imagen infrarroja de la esquina con humedades de una habitación.	60
Figura 29. Imagen Infrarroja de Filtraciones de Aire	61
Figura 30. Variación de la Radiación en función de la Emisividad del cuerpo medido	64
Figura 31. Imagen Infrarroja del Exterior de una Vivienda	68
Figura 32. Vista superior montaje del motor	69
Figura 33. Placa del motor	74
Figura 34. Montaje del motor	74
Figura 35. Módulo Intercambiador de calor por aletas	75
Figura 36. Esquema eléctrico módulo intercambiador	76
Figura 37. Módulo Profundidad	77
Figura 38. Resistencias térmicas	80
Figura 39. Módulo de Tamaño	80
Figura 40. Módulo de Forma	81
Figura 41. Resistencias cilíndricas y sensores	82
Figura 42. Módulo Posición	83
Figura 43. Sistema eléctrico del banco a) Encendido, b) Conectores	84

Figura 44. Conexión Hembra	85
Figura 45. Cables con banana	85
Figura 46. Plano eléctrico del banco	86
Figura 47. Banco de Trabajo	87
Figura 48. GUIDE tratamiento de imágenes	89
Figura 49. Relación ideal entre el área de la figura y el boundingbox	91
Figura 50. Relación entre color, temperatura y valor de pixel en unit8	92
Figura 51. Opciones tratamiento de imágenes	93
Figura 52. GUIDE tratamiento de video	94
Figura 53. Opciones tratamiento de video	98
Figura 54. Revisión cámara termográfica	100
Figura 55. Prueba para definición de parámetros de la cámara	100
Figura 56. Toma de temperatura mediante puntero	102
Figura 57. Curvas de temperatura módulo de forma	102
Figura 58. Curvas de temperatura módulo de tamaño	103
Figura 59. Curvas de temperatura módulo posición	103
Figura 60. Curvas de temperatura módulo profundidad	104
Figura 61. Curva de temperatura módulo de Aletas	104
Figura 62. Curvas de temperatura módulo de rodamientos	105
Figura 63. Administrador de dispositivos	106
Figura 64. Video almacenado	107
Figura 65. Imagen almacenada	107

Figura 66. Binarización a) Temperatura no uniforme, b) Temperatura Uniforme	108
Figura 67. Identificación de figuras geométricas en Imágenes	109
Figura 68. Identificación de figuras geométricas en Video	110
Figura 69. Identificación de puntos calientes en imágenes	111
Figura 70. Identificación de puntos calientes en videos	112
Figura 71. Identificación de tamaño en imágenes	113
Figura 72. Identificación de tamaño en videos	114
Figura 73. Análisis de temperatura en imágenes	115
Figura 74. Análisis de temperatura en video	116

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Formatos y extensiones para imágenes	<b>27</b>
Tabla 2. Formatos y extensiones para videos	<b>38</b>
Tabla 3. Herramientas de una GUI	<b>44</b>
Tabla 4. Emisividad de algunos materiales	<b>65</b>
Tabla 5. Espesores de la lámina de acero	<b>70</b>
Tabla 6. Selección de chumacera	<b>71</b>
Tabla 7. Selección del acople flexible	<b>72</b>
Tabla 8. Selección de acople Interflex	<b>73</b>
Tabla 9. Propiedades físicas y mecánicas del acero inoxidable	<b>79</b>
Tabla 10. Aplicación de Módulos en el programa	<b>88</b>
Tabla 11. Verificación Banco de trabajo	<b>99</b>
Tabla 12. Parámetros de la cámara para módulos	<b>101</b>

## LISTA DE ANEXOS

	<b>Pág.</b>
Anexo A. MANUAL DE USO BANCO DE TRABAJO DE TERMOGRAFÍA INFRARROJA	119
Anexo B. APLICACIÓN MATLAB: MANUAL DE USO	124
Anexo C. MANUAL DE MANTENIMIENTO PREVENTIVO DEL BANCO	130
Anexo D. CÓDIGO MATLAB TRATAMIENTO DE IMÁGENES	133
Anexo E. CÓDIGO MATLAB TRATAMIENTO DE VIDEO	141

## RESUMEN

**TITULO: MONITOREO DE FALLAS EN MAQUINARIA TÉRMICA MEDIANTE TERMOGRAFÍA INFRARROJA\***

**AUTORES: AUDEMAR DAMIAN VEGA GONZALEZ  
GONZALO ALBERTO DIAZ RUIZ\*\***

**PALABRAS CLAVE:** Termografía infrarroja, cámara termográfica, termografía, mantenimiento predictivo, banco de termografía infrarroja.

### **DESCRIPCION:**

Este proyecto fue realizado con el propósito de desarrollar una aplicación en el entorno MATLAB, que permita monitorear fallas en maquinaria por medio de termografía infrarroja y mejore la comprensión de los estudiantes en el tema de mantenimiento predictivo, ya que ha sido un tema poco explorado en la academia, se aportó en el campo de la termografía infrarroja una herramienta para hacer monitoreos de maquinas que se ven expuesto debido a su funcionamiento a posibles fallos.

Se diseñó una aplicación en MATLAB que permite tomar una imagen térmica o un video, este realiza un procesamiento de la imagen y hace una comparación con una imagen de muestra inicial, permitiendo analizar el color rojo como punto de comparación para verificar un cambio en la temperatura, seguido por una alerta específica que indica la respectiva verificación. Los resultados son presentados mediante el entorno gráfico de usuario de MATLAB el cual permite interactuar directamente con el código, ingresando datos continuos para recibir resultados gráficos.

La finalidad principal es la de identificar fallas debidas a aumentos de temperatura y concentración de puntos calientes de temperatura. Para poder hacer simulación de los fenómenos termográficos, se construyó un banco para pruebas térmicas, el banco dispone de 6 módulos, cada modulo sirve para realizar y analizar un comportamiento térmico diferente, comportamientos que son comunes en maquinaria donde se pueden presentar fallas debidas a aumentos súbitos de temperatura como también a comportamientos de aumento de temperatura gradual.

---

\*Trabajo de grado

\*\*Facultad de ingenierías físico-mecánicas. Escuela de ingeniería mecánica. Director Ing. JABID EDUARDO QUIROGA MENDEZ

## SUMMARY

**TITLE: FAULT MONITORING IN THERMAL MACHINERY WITH USE OF THERMOGRAPHY INFRARED\***

**AUTHORS: AUDEMAR DAMIAN VEGA GONZALEZ  
GONZALO ALBERTO DIAZ RUIZ\*\***

**KEYWORDS:** infrared thermography, thermal camera, thermography, predictive maintenance, bank of infrared thermography.

### **DESCRIPTION:**

This project was conducted with the purpose of developing an application in the MATLAB environment, which allows faults in machinery monitoring using infrared thermography and improve student understanding in the field of predictive maintenance, because it has been a relatively unexplored subject at the academy, it contributes in the field of infrared thermography a tool for monitoring of machines that are exposed due to operating faults.

We designed an application in matlab that can take a thermal image or a video, it does image processing and a comparison with an initial sample image, allowing to analyze the color red as a comparison point to verify a change in temperature , followed by a specific alert indicating the respective verification. The results are presented using the graphical user environment which enables MATLAB to interact directly with the code, giving continuous data logging for graphical results.

The main purpose is to identify failures due to increases in temperature and concentration of temperature hot spots. To be able to simulate the thermographic phenomena, we constructed a thermal test bank, the bank has 6 modules, each module is used to implement and analyze different thermal behavior, behaviors that are common in machinery where failures can occur due to increases sudden temperature behavior as well as gradual increase in temperature.

---

\* Degree work

\*\* Physical Mechanical Engineering Faculty. Mechanical Engineering School. Director Eng.JABID EDUARDO QUIROGA MENDEZ

## INTRODUCCIÓN

La gran mayoría de los problemas y averías en el entorno industrial; ya sea de tipo mecánico, eléctrico y de fabricación; están precedidos por cambios de temperatura que pueden ser detectados mediante la monitorización de temperatura con sistema de Termovisión por Infrarrojos. La implementación de programas de inspecciones termográficas en instalaciones, maquinaria, cuadros eléctricos, etc. Hace posible minimizar el riesgo de una falla de equipos y sus consecuencias, a la vez que también ofrece una herramienta para el control de calidad de las reparaciones efectuadas.

Una cámara termográfica es un fiable instrumento a distancia capaz de analizar y visualizar la distribución de temperatura de superficies completas de equipamiento eléctrico y maquinaria con rapidez y precisión. Una de las múltiples ventajas de la termografía es la capacidad para llevar a cabo inspecciones mientras los sistemas eléctricos están en funcionamiento. Al tratarse de un método de diagnóstico sin contacto, la persona puede inspeccionar rápidamente un componente concreto de un equipo a una distancia de seguridad, abandonar la zona de riesgo, regresar a su oficina y analizar los datos sin exponerse a ningún peligro.

La termografía es una herramienta excelente para hacer monitoreos de la condición, en busca de reducir los costos asociados al mantenimiento correctivo de los equipos. La mayoría de los fabricantes de equipos mecánicos han establecido parámetros de temperaturas de operación para sus equipos a plena carga. Ésta es una de las referencias que se toman para la elaboración de diagnósticos. Con termografía infrarroja podemos inspeccionar un sin fin de equipos mecánicos, ejemplo: motores, rodamientos, bombas, bandas, ventiladores, tuberías, etcétera.

También detecta anomalías que suelen ser invisibles a simple vista y que permite realizar correcciones antes de que se produzcan costosos fallos en el sistema, en el proyecto y con el uso de una aplicación en matlab, permite a nivel académico observar comportamientos térmicos en donde con una imagen termografica al banalizarla e identificar sus parámetros se puede definir su tamaño, su forma e identificar sus puntos más calientes.

Para poder realizar pruebas con la cámara termografica se construyo un banco de trabajo en donde se pueden tomar imágenes térmicas de diferentes módulos para observar comportamientos de calentamientos en placas de acero con formas geométricas conocidas, también en sólidos de distintas profundidades, calentamiento de rodamientos y disipadores por aletas.

El banco dispone de 6 módulos, cada modulo sirve para observar diferentes comportamientos de imágenes térmicas con la siguiente denominación: módulo de calentamiento de rodamientos, módulo de profundidad, módulo de forma, módulo de tamaño y disipadores por aletas.

El primer capítulo presenta, en forma general, las características del sistema que definen los elementos constitutivos que sirven para poder tomar imágenes termográficas, también se definen todas las herramientas principales de matlab para el procesamiento y tratamiento de imágenes o video, el funcionamiento de la cámara y de la interfaz de uso con el usuario.

El capitulo dos presenta un banco de pruebas para termografía infrarroja en donde se puede tomar imágenes termográficas o videos, se describe el funcionamiento de cada modulo, algunos aspectos de construcción, conexiones eléctricas y montaje del sistema.

En el capítulo tercero se realizó un programa aplicativo en matlab que permite tener una imagen térmica, leerla y procesarla; compararla con una en estado normal y obtener resultados. También se aplica el mismo concepto con videos. Este programa se basa en la identificación de fallas por medio de cambios de temperatura, identificación de puntos calientes, formas y tamaños de figuras.

Las pruebas y mediciones se exponen en el cuarto capítulo. El cual es reforzado con el quinto capítulo en el cual se sustentan dichas mediciones con los resultados obtenidos.

El libro cierra con las conclusiones del proyecto, recomendaciones y nuevas expectativas de desarrollo e investigación en el área de termográfica infrarroja que tiene muchas aplicaciones a nivel industrial y que se está empezando a explorar a nivel académico en la universidad.

## 1. CONCEPTOS FUNDAMENTALES

### 1.1 CARACTERISTICAS DEL SISTEMA

Al realizar el sistema se encuentra constituido por los siguientes elementos característicos:

**LA CÁMARA TERMOGRÁFICA** toma las imágenes necesarias para su respectivo análisis.

**IMÁGENES TÉRMICAS:** Es la forma en que se van a tomar las imágenes. Cambia si el elemento de inspección tiene características diferentes.

**ANÁLISIS POR INFRARROJOS:** Es el elemento que identifica las características principales de la imagen para dar los respectivos resultados.

**SOFTWARE DE PROGRAMACIÓN** como analizador de información. Es el medio que relaciona las imágenes con los datos.

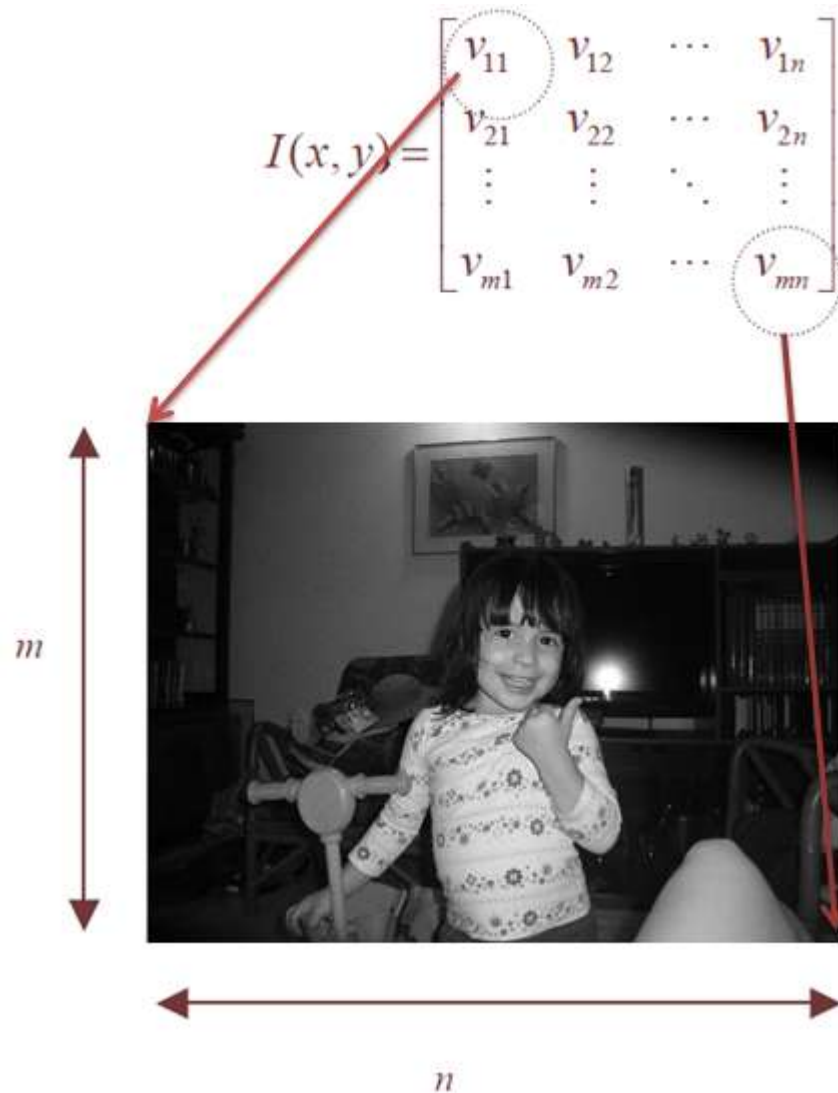
### 1.2 MATLAB Y EL TRATAMIENTO DE IMÁGENES

En matlab una imagen a escala de grises es representada por medio de una matriz bidimensional de  $m \times n$  elementos en donde  $n$  representa el numero de píxeles de ancho y  $m$  el numero de píxeles de largo. El elemento  $v_{11}$  corresponde al elemento de la esquina superior izquierda (ver figura 1), donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco).

Por otro lado una imagen de color RGB (la más usada para la visión computacional, además de ser para matlab la opción default) es representada por

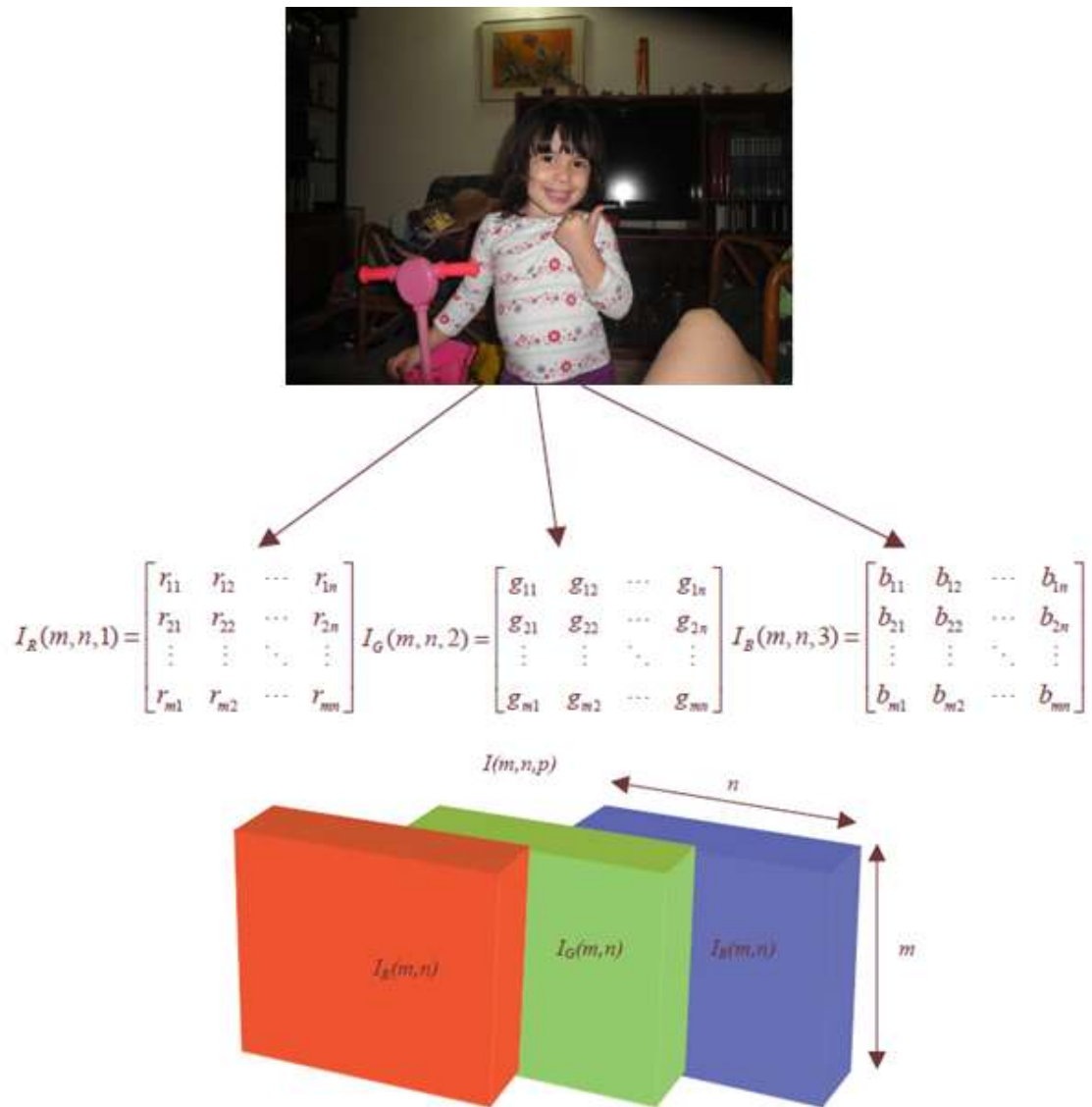
una matriz tridimensional  $m \times n \times p$ , donde  $m$  y  $n$  tienen la misma significación que para el caso de las imágenes de escala de grises mientras  $p$  representa el plano, que para RGB que puede ser 1 para el rojo, 2 para el verde y 3 para el azul. La figura 2 muestra detalles de estos conceptos.

**Figura1. Representación de una imagen a escala de grises en MATLAB**



**Fuente: Autores Proyecto**

Figura 2. Representación de una imagen a color RGB en MATLAB



Fuente: Autores Proyecto

### 1.2.1 Lectura de imágenes a través de archivo

Para leer imágenes contenidas en un archivo al ambiente de matlab se utiliza la función *imread*, cuya sintaxis es *imread* ('nombre del archivo') Donde nombre del archivo es una cadena de caracteres conteniendo el nombre completo de la

imagen con su respectiva extensión, los formatos de imágenes que soporta matlab son los mostrados en la Tabla 1.

Para introducir una imagen guardada en un archivo con alguno de los formatos especificados en la tabla anterior solo tiene que usarse la función *imread* y asignar su resultado a una variable que representará a la imagen (de acuerdo a la estructura, Figura 1 para representar escala de grises y Figura 2 para RGB). De tal forma que si se quisiera introducir la imagen contenida en el archivo *data.jpg* a una variable para su procesamiento en matlab, entonces se tendría que escribir en línea de comandos:

```
> >image = imread ('data.jpg');
```

Con ello la imagen contenida en el archivo *data.jpg* quedará contenida en la variable *image*.

Después de realizar un procesamiento con la imagen, es necesario desplegar el resultado obtenido, la función *imshow* (variable) permite desplegar la imagen en una ventana en el ambiente de trabajo de matlab. Si la variable a desplegar por ejemplo, es *prueba* al escribir en la línea de comandos:

```
> >imshow (prueba);
```

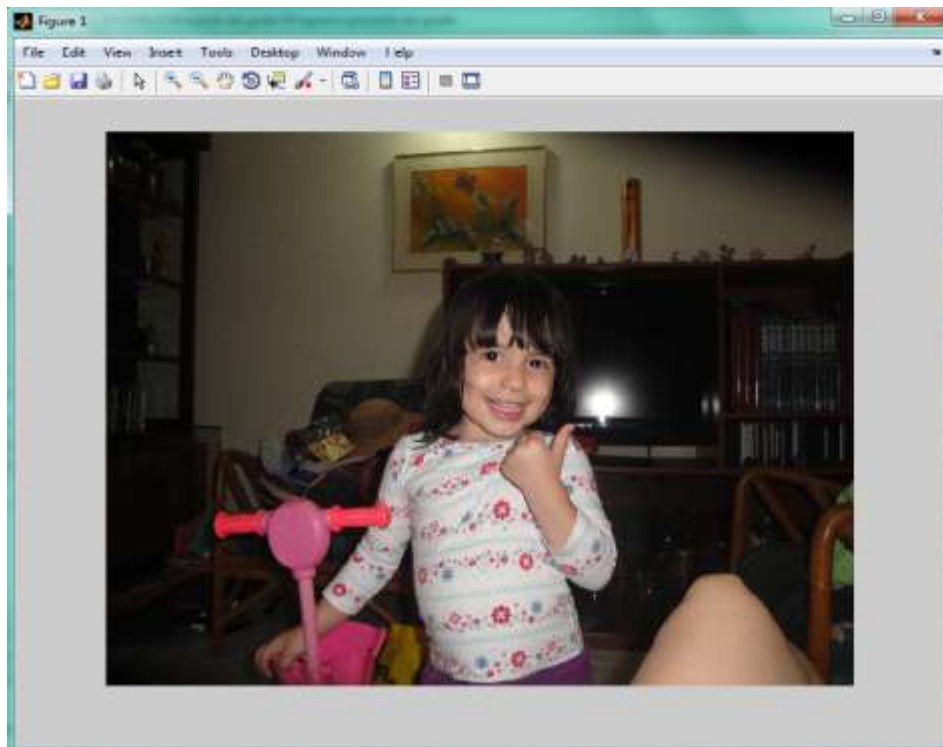
Obtendríamos la imagen de la figura 3.

**Tabla 1. Formatos y extensiones para imágenes**

Formato	Extensión
TIFF	.tiff
JPEG	.jpg
GIF	.gif
BMP	.bmp
PNG	.png
XWD	.xwd

**Fuente: [www.mathworks.com](http://www.mathworks.com)**

**Figura 3. Imagen mostrada al utilizar la función imshow**



**Fuente: Autores Proyecto**

### 1.2.2 Acceso a planos RGB

En el caso de imágenes RGB estas cuentan con tres planos uno para cada color que representa. Al considerar ahora que la imagen RGB contenida en la variable *image* es la mostrada en la figura 3, y se desea obtener cada uno de los planos que la componen. Entonces:

```
> >planoR=image( :, :,1) ;  
> >planoG=image( :, :,2) ;  
> >planoB=image( :, :,3) ;
```

Los planos resultantes por los anteriores comandos son mostrados en la siguiente figura.

**Figura 4. Planos de la imagen a) Rojo, b) Verde, c) Azul**



**Fuente: Autores Proyecto**

### 1.2.3 Histograma de una imagen

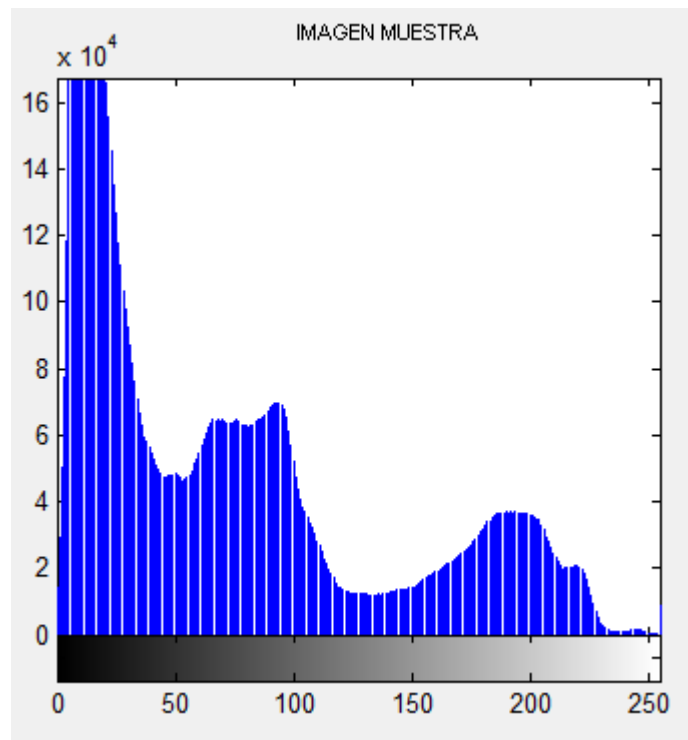
Es un diagrama que muestra la cantidad de píxeles de una matriz bidimensional que tiene la misma información de color (ver figura 5).

Para imágenes a escala de grises y planos RGB se utiliza el siguiente comando, si se considera el plano rojo de la figura 4.

```
>>figure, imhist (planoR)
```

La función *figure* crea gráficos de objetos en ventanas individuales en la pantalla donde MATLAB muestra salidas gráficas.

**Figura 5. Histograma del plano rojo de una imagen**



**Fuente: Autores Proyecto**

#### **1.2.4 Herramientas de Procesamiento de imágenes**

El número de funciones que implementa el toolbox para el procesamiento de imagen es muy diverso, sin contar la múltiple oferta de funciones ya generada por otros usuarios y disponibles a través del Internet, sin embargo, serán tratadas algunas consideradas como las más usadas y útiles para la visión computacional.

#### 1.2.4.1 Extracción de Bordes

En visión computacional es de utilidad para hacer reconocimiento de objetos o bien para segmentar regiones, extraer los bordes de objetos (que en teoría delimitan sus tamaños y regiones). La función *edge* da la posibilidad de obtener los bordes de la imagen. La función permite encontrar los bordes a partir de dos diferentes algoritmos que pueden ser elegidos, *canny* y *sobel*. El formato de esta función es:

**Figura 6. Imagen resultado de la aplicación del algoritmo de canny**



**Fuente: Autores Proyecto**

**ImageT=edge(ImageS, algoritmo);**

Donde ImageT es la imagen obtenida con los bordes extraídos, ImageS es la variable que contiene la imagen en escala de grises a la cual se pretende recuperar sus bordes, mientras que algoritmo puede ser uno de los dos canny o sobel. De tal forma que si a la imagen de la Fig. 1 contenida en la variable

imagegray se le quieren recuperar sus bordes utilizando en algoritmo canny se escribiría en línea de comandos:

```
> >ImageR=edge(imagegray,canny);
```

Si se desea convertir una imagen RGB a escala de grises se utiliza:

```
>>im_g=rgb2gray(img);
```

La figura 6 muestra un ejemplo del uso de estas funciones.

#### 1.2.4.2 Binarización

Las operaciones morfológicas son operaciones realizadas sobre imágenes binarias basadas en formas. Una imagen binaria es una en la cual cada píxel puede tener solo uno de dos valores posibles 1 o 0. Para binarizar una imagen se debe escribir:

**Figura 7. Binarización de una imagen**



**Fuente: Autores Proyecto**

```
>>Im_bw=im2bw(img,umbral);
```

Donde Im\_bw es la imagen binarizada, img es la variable que contiene la imagen a escala de grises y umbral es el respectivo porcentaje de captación de los pixeles el cual está entre 0 y 1 y puede ser colocado manualmente o calculado por medio de:

```
>>umbral=graythresh(img);
```

La Fig. 7 muestra un ejemplo del uso de esta función.

### 1.2.4.3 Filtraje

Se utiliza para disminuir o eliminar el ruido de una imagen. Su comando es:

**Figura 8. Filtraje de una imagen a) Con ruido, b) Filtrado**



(a)

(b)

**Fuente: Autores Proyecto**

Esta función filtra la imagen im\_g (tipo doublé, unit 8 no acepta) con el filtro multidimensional h, el resultado se almacena en Im\_fil.

La función de Matlab que me permite generar el filtro h será **fspecial**.

La función fspecial crea filtros bidimensionales del tipo especificado por type.

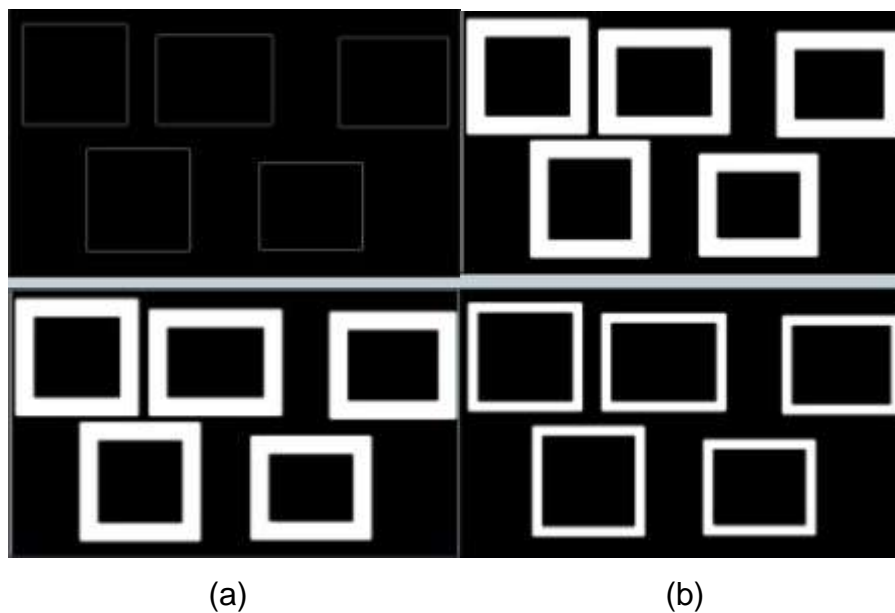
El valor de type puede ser un filtro pasa baja gaussiano, un detector de bordes de sobel o prewitt, un operador laplaciano, un filtro de la media, etc.

La figura 8 muestra un ejemplo de esta función.

#### 1.2.4.4 Dilatación y Erosión

Son las operaciones morfológicas más utilizadas y se aplican a imágenes binarizadas. La primera adiciona pixeles en las fronteras de la imagen y la segunda los remueve. Los comandos son:

**Figura 9. Dilatación y Erosión de una Imagen a) Dilatación, b) Erosión**



**Fuente: Autores Proyecto**

```
>>Re=imdilate(Img,SE);
```

```
>>Re=imerode(Img,SE);
```

Donde Re es la variable que guarda el resultado dilatado (imdilate) o erosionado (imerode) de la Imagen (Img) previamente binarizada con una estructura de arreglo SE; su comando es:

```
>>SE=strel(arreglo,d);
```

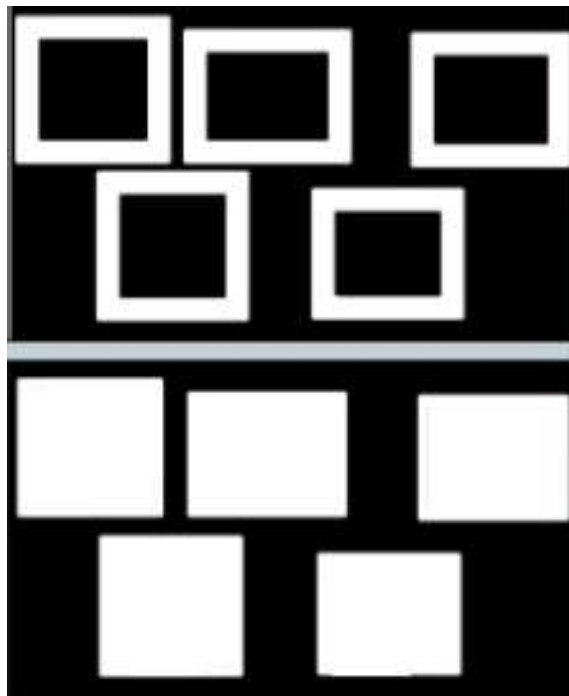
El arreglo puede ser cuadrado (square), circulo (disk), entre otros y la constante d es el espesor de dicho arreglo.

La figura 9 muestra ejemplos de estas funciones.

#### 1.2.4.5 Llenado de Espacios

Se utiliza para rellenar aquellos espacios que no fueron alcanzados por la función imdilate (dilatación) y se requiera que así sea. El comando es:

**Figura 10. Aplicación de la función Imfill**



**Fuente: Autores Proyecto**

```
>> fill=imfill(img,'holes');
```

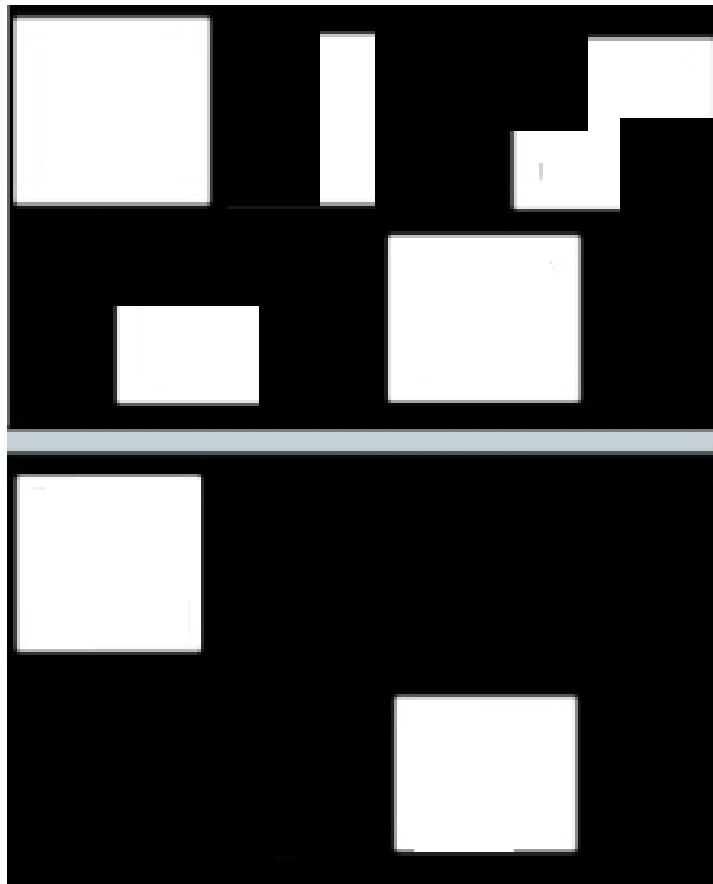
El resultado de la imagen (img) previamente binarizada se guardará en la variable fill; el parámetro “holes” indica el llenado de los espacios entre puntos.

La figura 10 muestra un ejemplo de esta función.

#### 1.2.4.6 Eliminación de Áreas

El ruido afecta a cualquier tipo de imágenes presentándose como pequeñas áreas o puntos que distorsionan la lectura de estas. Para eliminar este problema, se debe aplicar la función bwareaopen y su comando es:

**Figura 11. Aplicación de bwareaopen**



**Fuente: Autores Proyecto**

```
>> bw=bwareaopen(img,A);
```

Donde bw es la variable que guarda el resultado de procesar la imagen (img) previamente binarizada y A es el tamaño del área a eliminar.

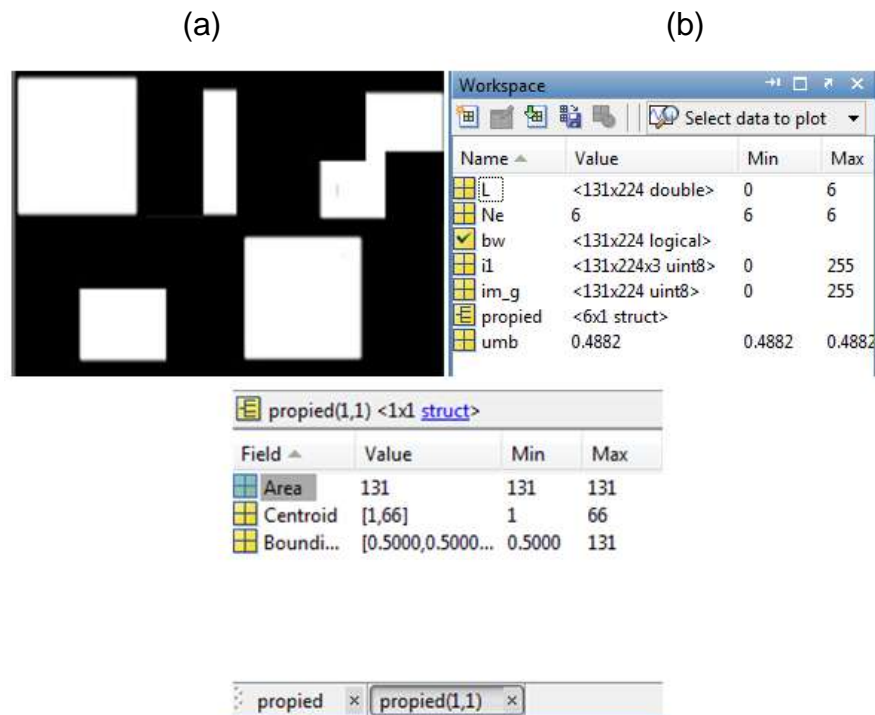
Un ejemplo de esta función se da en la figura 11.

#### 1.2.4.7 Etiquetado y propiedades de una imagen

Ciertas imágenes están compuestas por varias figuras con diferentes propiedades, de las cuales se requiere saber cierto tipo de información de gran ayuda para su procesamiento como su área, centroide, entre otros.

Para etiquetarlas (distinguir las unas de otras) se utiliza esta función:

**Figura 12. Propiedades de una Imagen a) Imagen, b) Workspace, c) Propiedades**



(c)

**Fuente: Autores Proyecto**

```
>> [L Ne]=bwlabel(img);
```

Donde se crea una matriz etiquetada de los elementos (L) de la imagen (Img) previamente binarizada y el número de ellos (Ne).

Para hallar sus propiedades se utiliza la siguiente función:

```
>> propied=regionprops(L);
```

La cual guarda el resultado de cada elemento en la variable propied.

La figura 12 muestra un ejemplo de esta función.

### **1.3 TRATAMIENTO DE VIDEO**

El video es la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción de una secuencia de imágenes que representan escenas en movimiento. Estas escenas son conocidas como “frames” y como tal cada uno de ellos puede ser tratado como una imagen.

Las herramientas utilizadas para el video son las mismas que se utilizan para imágenes pero hay algunas que caracterizan a este como tal y generan nuevas aplicaciones.

#### **1.3.1 Leer video**

Con el uso de la instrucción de matlab mmreader es posible leer archivos de video desde varios formatos (ver tabla 2), su comando es:

```
>> obj = mmreader(nombre.extension)
```

**Tabla 2. Formatos y extensiones para videos**

Formato	Extensión
AVI	(.avi)
MPEG	(.mpg)
Motion JPEG	(.mj2)
Windows Media	(.wmv)
Apple quick time	(.mov)

**Fuente:** [www.mathworks.com](http://www.mathworks.com)

Donde obj lee los datos de vídeo desde el archivo llamado nombre. El constructor mmreader busca el archivo en la ruta de MATLAB. Si no se puede construir el objeto por cualquier razón, mmreader genera un error.

### **1.3.2 Propiedades del video**

El video como la imagen tiene ciertas características que lo identifican como tal y estas pueden ser guardadas en variables para el uso que se desee dar. Para saber el número de frames que lo componen tenemos:

```
>>nFrames = obj.NumberOfFrames;
```

Donde nFrames es la variable que guarda el resultado del objeto “obj”.

También existen características de forma como lo son el alto y el ancho del video en el reproductor y sus comandos son, respectivamente:

```
>>vidHeight = obj.Height;
```

```
>>vidWidth = obj.Width;
```

### 1.3.3 Preasignar estructura del video

Cada video tiene propiedades únicas que lo identifican, entre ellas el número de frames, alto, ancho, mapa de colores, planos y la información detallada de la matriz que conforma; todo esto es llamado “estructura” y para usarla, se debe crear antes de llenarla con las diferentes imágenes.

El típico error que se comete al crear un video es no darle el suficiente espacio que necesita al iniciar un bucle que guarde cada frame en una variable cambiante, creando así errores como: “Index exceeds matrix dimentions”, esto traduce que se está incorporando más información de la que la estructura del video puede soportar.

Este problema se soluciona de esta manera:

```
>>mov(1:nFrames) = ...  
struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...  
      'colormap', []);
```

Donde la variable mov guarda los frames del 1 al total de ellos en una estructura “vacía” que se puede llenar con la información necesaria y suficiente.

## 1.4 SENTENCIAS DE CONTROL DE FLUJO

Se utilizan como en cualquier otro lenguaje de programación. Suelen escribirse en archivos de Matlab y no directamente en la línea de comandos. Permiten también estructuras anidadas.

Es importante tener en cuenta las sentencias que terminan esos controles (end), para colocarlas cuando son necesarias y en el orden adecuado.

### 1.4.1 Bucles FOR

Los bucles *for* permiten que un grupo de órdenes se repitan un número fijo y predeterminado de veces. La forma general de un bucle *for* es:

```
for x= array,  
órdenes  
end ;
```

Las *órdenes* entre las sentencias *for* y *end* se ejecutan una vez para cada columna del *array*. Es decir, en cada iteración la variable *x* valdrá  $x=array(:,i)$  variando *i* entre 1 y la última columna.

### 1.4.2 Estructuras IF-ELSE-END

En muchas ocasiones es necesario evaluar una expresión dependiendo de que una condición se cumpla o no. Esta estructura es:

```
if expresión,  
órdenes  
end ;
```

En algunos casos existen dos alternativas, dependiendo de si es cierta o no dicha expresión. Entonces:

```
if expresión,  
órdenes_si_expresión_verdadera  
else  
órdenes_si_expresión_falsa  
end ;
```

Cuando hay tres o más opciones la forma de la estructura *if-else-end* será:

```

if expresión1,
órdenes_si_expresión1_verdadera
elseif expresión2,
órdenes_si_expresión2_verdadera
elseif expresión3,
órdenes_si_expresión3_verdadera
...
else
órdenes_si_ninguna_expresión_verdadera
end ;

```

En esta última forma sólo se evalúan las órdenes asociadas con la primera expresión verdadera; las expresiones condicionales siguientes se saltan hasta el *end* y no se evalúan. La orden *else* es opcional y podría no aparecer.

### 1.4.3 Estructura CASE

Esta sentencia sirve para agrupar varias sentencias IF en una sola, en el caso particular en el que una variable es comparada a diferentes valores, todos ellos constantes, y que realiza acciones si coincide con ellos. Su sintaxis es:

```

SELECT CASE  expresión_control
          CASE  lista_expresiones_1
                bloque_sentencias_1
          CASE  lista_expresiones_2
                bloque_sentencias_2
          CASE ELSE
                bloque_sentencias_n
END

```

## 1.5 INTERFAZ GUIDE

GUIDE es un entorno de programación visual disponible en MATLAB para realizar y ejecutar programas que necesiten ingreso continuo de datos. Tiene las

características básicas de todos los programas visuales como Visual Basic o VisualC++.

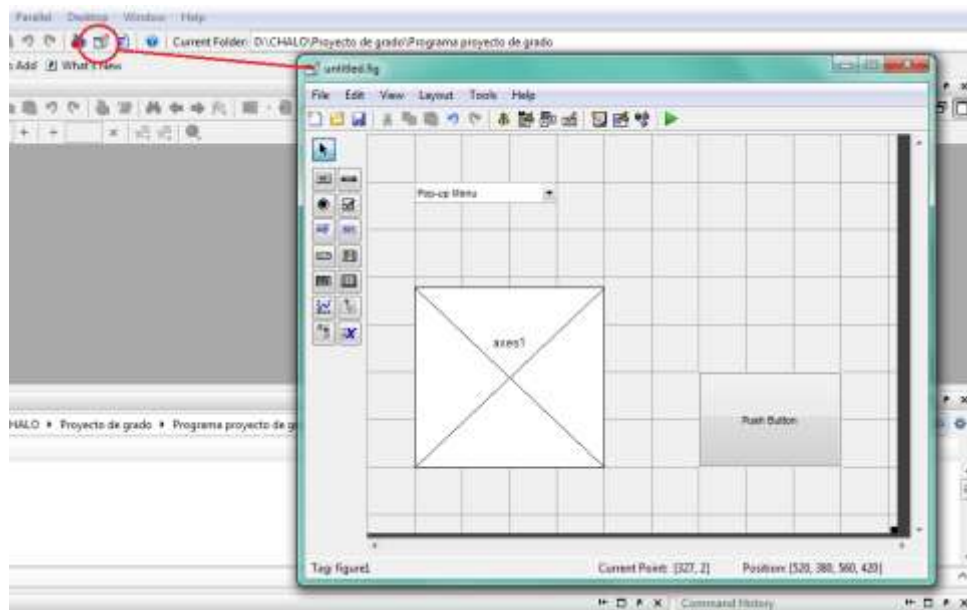
Para iniciar un proyecto, se puede hacer de dos maneras:

- Ejecutando la siguiente instrucción en la ventana de comandos:

**>> guide**

- Haciendo un click en el ícono que muestra la figura

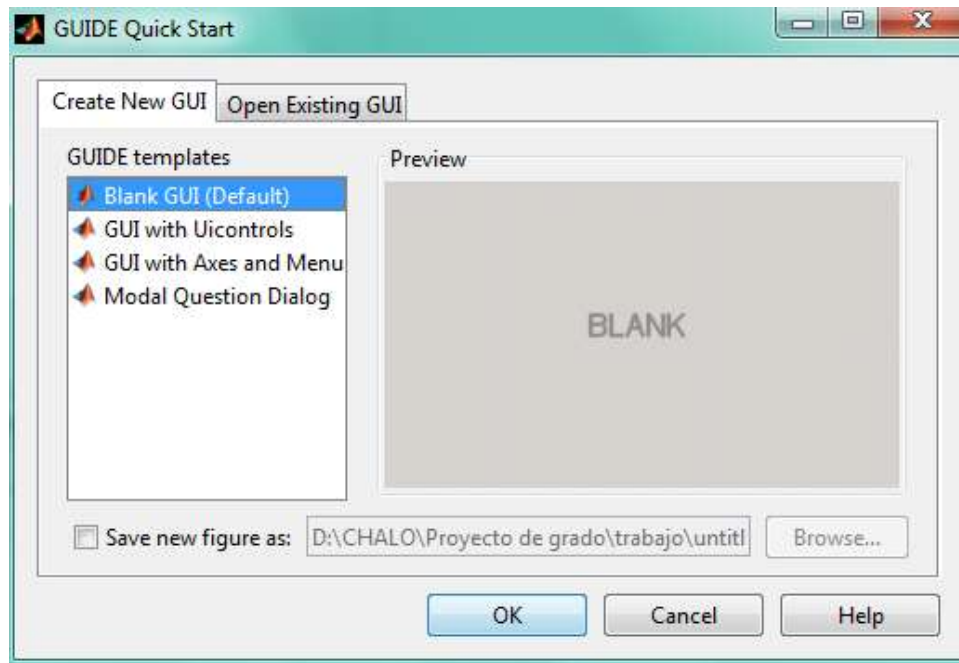
**Figura 13. Icono GUIDE**



**Fuente: Autores Proyecto**

Se presenta el siguiente cuadro de dialogo:

**Figura 14. Ventana de inicio de GUI**



**Fuente: Autores Proyecto**

- **Blank GUI (Default)**

La opción de interfaz gráfica de usuario en blanco (predeterminada), presenta un formulario nuevo, en el cual se diseña el programa.

- **GUI with Uicontrols**

Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades. Se puede ejecutar este ejemplo y obtener resultados.

- **GUI with Axes and Menu**

Esta opción es otro ejemplo el cual contiene el menú File con las opciones Open, Print y Close. En el formulario tiene un Popup menu, un push button y un objeto Axes, se puede ejecutar el programa eligiendo alguna de las

seis opciones que se encuentran en el menú despegable y haciendo click en el botón de comando.

- **Modal Question Dialog**

Con esta opción se muestra en la pantalla un cuadro de diálogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No dependiendo del botón que se presione, el GUI retorna el texto seleccionado (la cadena de caracteres 'Yes' o 'No').

### 1.5.1 Componentes de una GUI

Son todas las herramientas que permiten al usuario interactuar con el entorno gráfico.

**Tabla 3. Herramientas de una GUI**

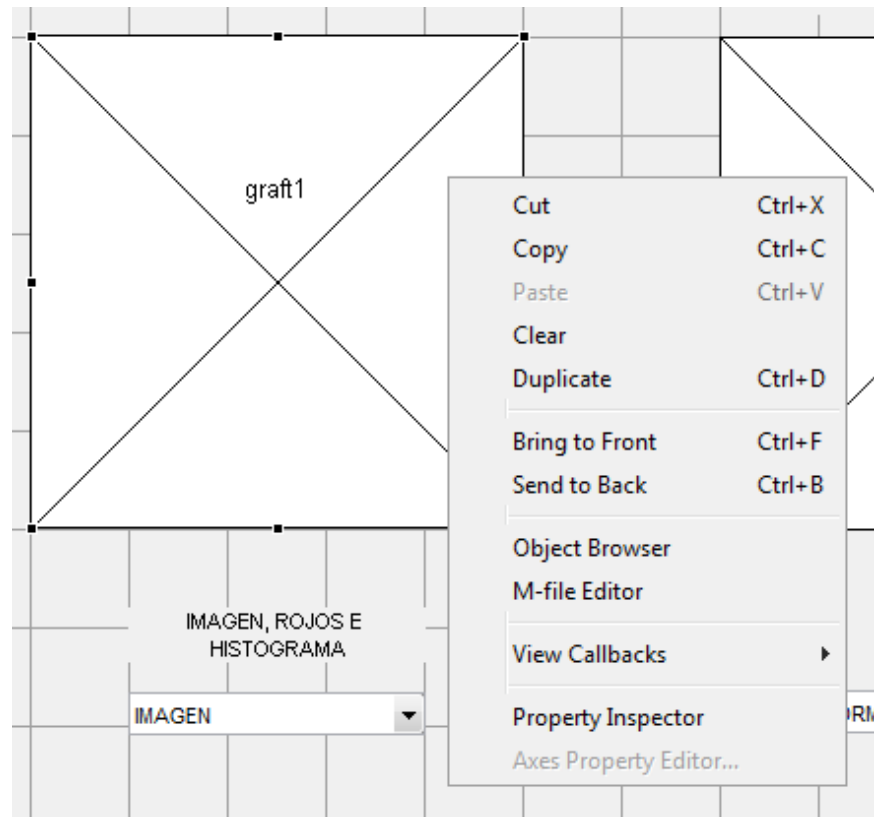
<i>Control</i>	<i>Valor de estilo</i>	<i>Descripción</i>
Check box	'checkbox'	Indica el estado de una opción o atributo
Editable Text	'edit'	Caja para editar texto
Pop-up menu	'popupmenu'	Provee una lista de opciones
List Box	'listbox'	Muestra una lista deslizable
Push Button	'pushbutton'	Invoca un evento inmediatamente
Radio Button	'radio'	Indica una opción que puede ser seleccionada
Toggle Button	'togglebutton'	Solo dos estados, "on" o "off"
Slider	'slider'	Usado para representar un rango de valores
Static Text	'text'	Muestra un string de texto en una caja
Panel button		Agrupar botones como un grupo
Button Group		Permite exclusividad de selección con los radio button

Fuente: [www.mathworks.com](http://www.mathworks.com)

### 1.5.2 Propiedades de los Componentes

Cada uno de los elementos de GUI, tiene un conjunto de opciones a las cuales se accede con click derecho.

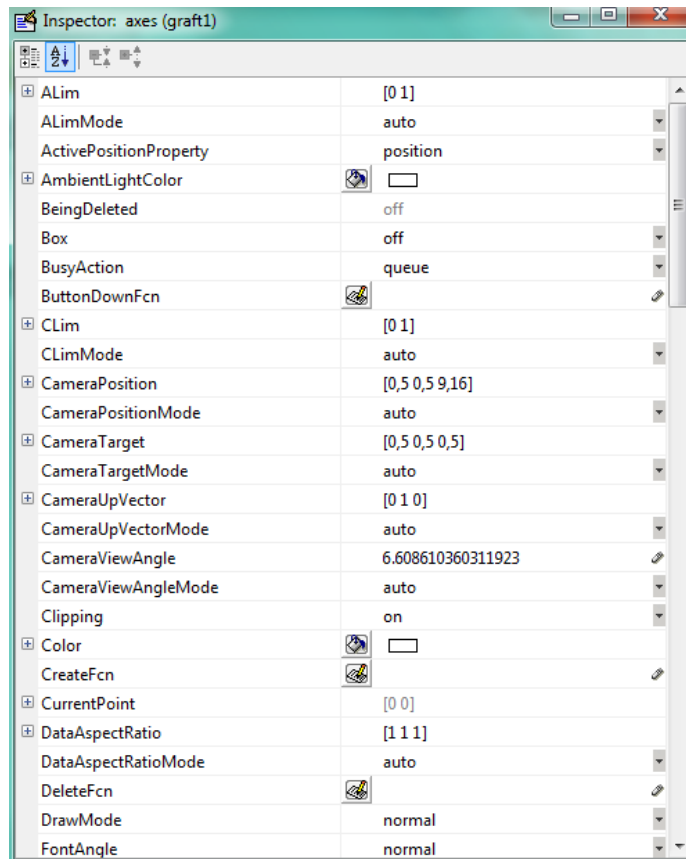
**Figura 15. Opción del Componente**



**Fuente: Autores Proyecto**

La opción Property Inspector permite personalizar cada elemento.

**Figura 16. Entorno Property Inspector**



**Fuente: Autores Proyecto**

Al hacer click derecho en el elemento ubicado en el área de diseño, una de las opciones más importantes es View Callbacks, la cual, al ejecutarla, abre el archivo .m asociado al diseño y posiciona en la parte del programa que corresponde a la subrutina que se ejecutará cuando se realice una determinada acción sobre el elemento que estamos editando.

Por ejemplo, al ejecutar View Callbacks>>Callbacks en el Push Button, nos ubicaremos en la parte del programa:

```
functionpushbutton1_Callback(hObject, eventdata, handles)
```

% hObject handle to pushbutton1

% eventdata reserved-to be defined in a future version of MATLAB

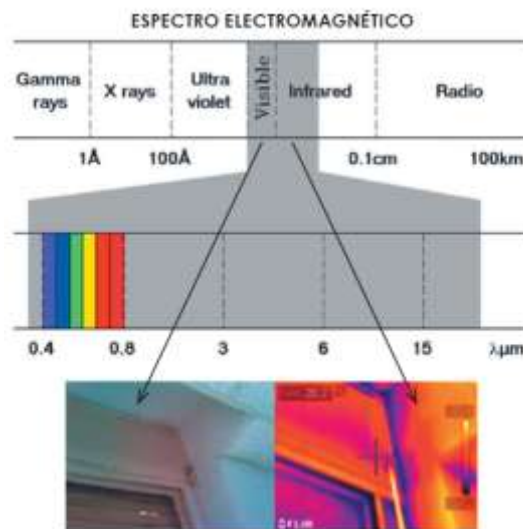
% handles structure with handles and user data (see GUIDATA)

## 1.6 TERMOGRAFIA INFRARROJA

La Termografía Infrarroja es la técnica de producir una imagen visible a partir de radiación infrarroja invisible (para el ojo humano, ver figura 17) emitida por objetos de acuerdo a su temperatura superficial. La cámara termográfica es la herramienta que realiza esta transformación (ver figura 18).

Esta técnica permite detectar, sin contacto físico con el elemento bajo análisis, cualquier falla que se manifieste en un cambio de la temperatura sobre la base de medir los niveles de radiación dentro del espectro infrarrojo.

**Figura 17. Espectro Electromagnético**



**Fuente: FLIR systems**

**Las unidades del espectro y longitud de onda son descritas en la imagen.**

En general, una falla electromecánica antes de producirse se manifiesta generando e intercambiando calor. Este calor se traduce habitualmente en una elevación de temperatura que puede ser súbita, pero, por lo general y dependiendo del objeto, la temperatura comienza a manifestar pequeñas variaciones.

**Figura 18. Imagen Termográfica**



**Fuente: eBuilding**

Si es posible detectar, comparar y determinar dicha variación, entonces se pueden detectar fallas que comienzan a gestarse y que pueden producir en el futuro cercano o a mediano plazo una parada de planta y/o un siniestro afectando personas e instalaciones. Esto permite la reducción de los tiempos de parada al minimizar la probabilidad de salidas de servicio imprevistas, no programadas, gracias a su aporte en cuanto a la planificación de las reparaciones y del mantenimiento. Los beneficios de reducción de costos incluyen ahorros de energía, protección de los equipos, velocidad de inspección y diagnóstico, verificación rápida y sencilla de la reparación, etc.

### 1.6.1 Cámara Termográfica

Como con cualquier otro equipo, para manejar una cámara termográfica es necesaria cierta habilidad y práctica, podríamos decir que diaria. Ocurre con frecuencia que con muchos de los equipos y sistemas con los que se trabaja en el día a día, existen menús, botones y utilidades que no se manejan habitualmente y se trabaja con lo mismo casi de manera automática.

**Figura 19. Cámara Termográfica FLUKE Ti125**



**Fuente: [www.fluke.com](http://www.fluke.com)**

Con la cámara termográfica puede pasar lo mismo, nos habituamos a usarla de una determinada manera y de ahí no nos movemos, echando a perder numerosas aplicaciones y no sacando el máximo partido a un equipo que ha costado una buena suma de dinero y tiempo dedicado a formarnos. Existen cámaras más o menos complejas y costosas, pero todas tienen el mismo principio.

Es importante tener cierta paciencia y dedicarle tiempo a familiarizarse con el equipo y aprender todo acerca de él. Los distintos fabricantes emplean distintos botones, menús y submenús, pero hay elementos comunes que sí son necesarios conocer.

### 1.6.1.1 Funcionamiento

Básicamente, la óptica de la cámara hace converger sobre su detector la radiación infrarroja que emite el objeto bajo estudio, obtiene una respuesta (cambio de tensión o resistencia eléctrica) que será leída por los elementos electrónicos (la placa electrónica) de la cámara. Esa señal electrónica es convertida en una imagen electrónica en la pantalla, donde los distintos tonos de color se corresponden con distintas señales de radiación infrarroja procedentes del objeto de estudio.

**Figura 20. Funcionamiento de la Cámara Termográfica**



**Fuente: FLIR systems**

### 1.6.1.2 Características Principales

Las principales características que hay que comprobar en una cámara termográfica son:

- Resolución: Número de píxeles o de puntos de medida.
- Amplios rangos de medida de temperatura.
- Precisión y exactitud.
- Capacidad de diferenciación de los incrementos de temperatura.
- Distancia a la que es capaz de medir.

- Puntero laser y linterna.
- Duración y facilidad de cambio de la batería.
- Tamaño de la pantalla.
- Capacidad de almacenamiento y tipo de soporte en que lo almacena (estándar .JPG).
- Tamaño, ergonomía, maniobrabilidad de la cámara y resistencia a caídas.
- Posibilidad de toma simultánea de fotografías y grabación de video para facilitar la emisión de informes.
- Software que acompaña a la cámara.
- Que pueda ser actualizada.
- Fabricante con servicio post-venta, soporte técnico y formación certificada.

### **1.6.1.3 Calibración**

Cuando se adquiere una cámara termográfica, ésta se entrega perfectamente calibrada y preparada para la inspección. De no ser así, se deberá devolver al fabricante para su reparación, siempre en garantía.

Se debe exigir el **Certificado de Calibración** de la cámara, con su modelo de cámara, número de serie y fecha de calibración. Es recomendable escanearlo o tenerlo bien guardado y localizable pues es habitual que para algunas inspecciones el cliente lo exija.

Es recomendable calibrar la cámara una vez al año si su uso es muy habitual. Este proceso se realiza en laboratorios acreditados, en condiciones controladas de temperatura y humedad y con lo que se llaman, simuladores de cuerpo negro.

Cada cámara tiene en su electrónica una curva en la que se relaciona la radiación recibida con una temperatura dada. Si esta curva se desajusta, es necesario reparar el equipo.

**Figura 21. Certificado de Calibración**



**Fuente: FLUKE TI125**

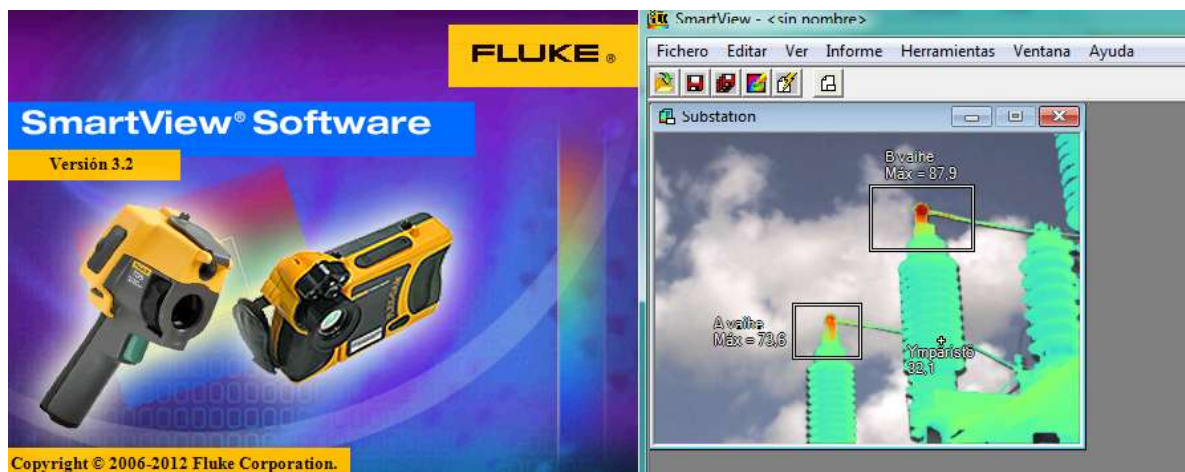
#### **1.6.1.4 Software de Tratamiento de Imágenes**

El termógrafo no se encuentra solo, con sus conocimientos y su cámara termográfica. Generalmente al adquirir la cámara termográfica, el fabricante proporcionará un software de tipo básico. Este software en muchos casos será suficiente para trabajar con las imágenes térmicas y descubrir más información de la que se pensaba que tenía la termografía cuando se toma en el lugar de la inspección. Además permitirá poder enfatizar y subrayar el problema para que el cliente tome más conciencia de él.

Además del software que viene con la cámara, muchos fabricantes desarrollan otros más complejos y con mayores y más potentes herramientas de diagnóstico para el termógrafo y no es raro que con el tiempo, el usuario de la cámara adquiera otro que le permita hacer trabajos cada vez más complejos.

Todos permitirán generar un informe, con los datos de la imagen, el logo de la empresa, etc. Pero también se puede optar por simplemente copiar la imagen (una vez exprimida la información al máximo), pegarla en el informe y escribir las conclusiones que se consideren oportunas.

**Figura 22. Software FLUKE SMARTVIEW 3.2**



Fuente: [www.fluke.com](http://www.fluke.com)

#### 1.6.1.5 Herramientas

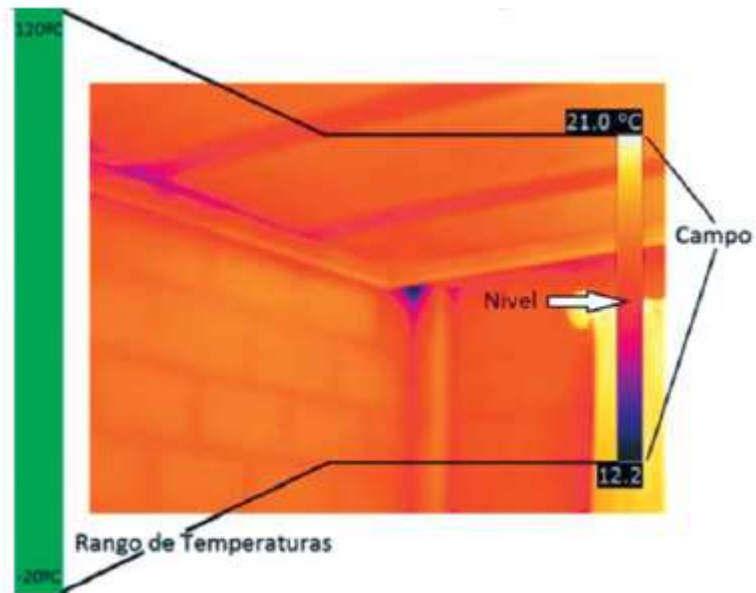
La cámara nada más encenderla ofrece ya datos relacionados con la imagen que muestra, por tanto conviene definir algunos conceptos iniciales.

**El rango**, el primer ajuste, fijará las temperaturas máxima y mínima a partir de las cuales se puede medir con la cámara (ver figura 25).

**El campo** será el intervalo de temperaturas que se ven durante la inspección. También se denomina contraste térmico y la cámara lo ajusta automáticamente.

**El nivel** es el punto medio del campo y si éste varía, también lo hace el nivel.

**Figura 23. Rango, campo y nivel**

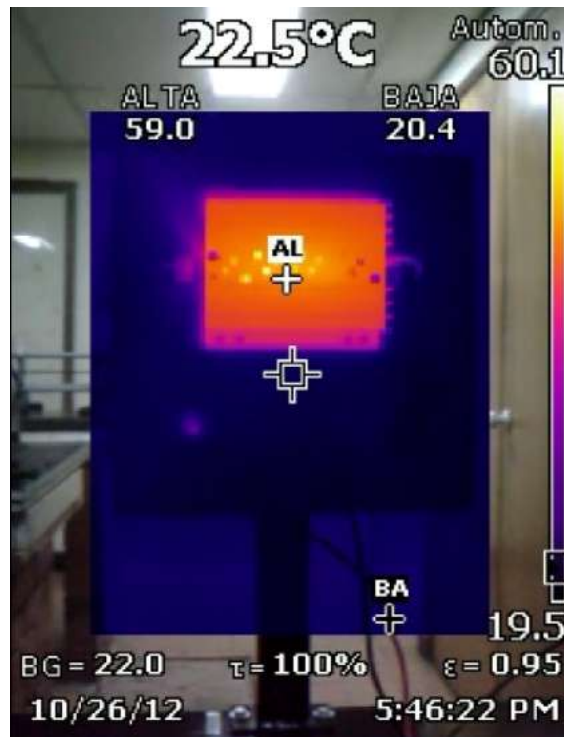


**Fuente: FLIR systems**

Además de estos parámetros, las cámaras termográficas suelen tener otras funciones útiles para el termógrafo, estas son:

- **El puntero de medida:** marca una zona concreta de la imagen. En función de la cámara, éste podrá ser fijo o móvil e incluso se podrán colocar varios puntos de medida.
- **El punto frío y punto caliente:** mostrará en la imagen, en tiempo real, el punto más frío y el punto más caliente que se esté viendo.
- **La Isotherma:** de gran utilidad para resaltar partes de la imagen con la misma temperatura, asignándoles un mismo color.
- **Distintas alarmas** que se pueden crear y definir para que avise de una ausencia de aislamiento o detecte humedades.
- **El video:** útil para evaluar en continuo un objeto.

**Figura 24. Imagen termográfica con puntos fijos, caliente, frío e isothermas**



**Fuente: Autores Proyecto**

### **1.6.2 Imagen Infrarroja**

Capturar la imagen, guardarla en la tarjeta de memoria de la cámara y posteriormente usarla para trabajar en el informe termográfico es el proceso habitual de trabajo del termógrafo.

Se podrá ver el problema, solucionarlo o no, avisar, pero no habrá constancia de ello. Si se toma la imagen pero no se guarda o no se encuentra la carpeta donde está, se tiene el mismo problema. Y por último, si no nos paramos a pensar, delante del ordenador y con el software de análisis de las imágenes térmicas, es probable que se pase por alto algún detalle que pueda ser de vital importancia para el informe.

Existen **cuatro grandes reglas** para capturar una imagen, no se deben olvidar pues nunca se podrá modificar una imagen si se ha cometido un error en alguno de estos puntos:

**Encuadre** de la imagen: en el fondo se es casi fotógrafo, pero no creativo, es decir, si se quiere medir un objeto, se debe presentar bien visible y en la posición que dé más información. ¡Acercándose todo lo que se pueda!, manteniendo una distancia segura.

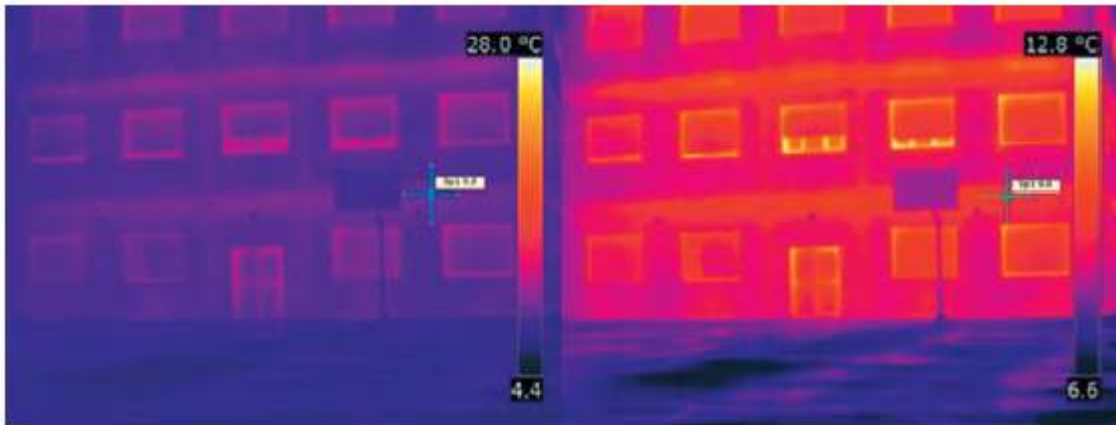
**Enfoque:** una imagen desenfocada dará medidas erróneas y además, hablará muy mal del termógrafo. Por supuesto se debe tratar de mantener la pantalla en todo momento limpia.

**Ajuste del rango** de temperatura: se debe fijar el rango de medida que incluya lo que se pretende medir. Si se fija un rango muy ancho, se tendrá menos precisión en la medida. Es como si se quieren medir milivoltios fijando el rango en kilovoltios.

Cuidado con las **condiciones** meteorológicas: la temperatura exterior, cambiante, y la hora a la que se realiza la inspección termográfica pueden hacer que ésta y sus resultados varíen de un día a otro, por lo que elegir el momento adecuado para hacer el trabajo es un factor a tener en cuenta. Obviamente esto no tiene sentido si la inspección es interior y el ambiente no afecta.

Si se olvida alguno de estos puntos, se debe volver al lugar de la inspección a realizar de nuevo el trabajo, con todo lo que ello conlleva de sobre costo para el informe.

**Figura 25. Misma imagen infrarroja, con el mismo rango pero con distinto campo**



**Fuente: FLIR systems**

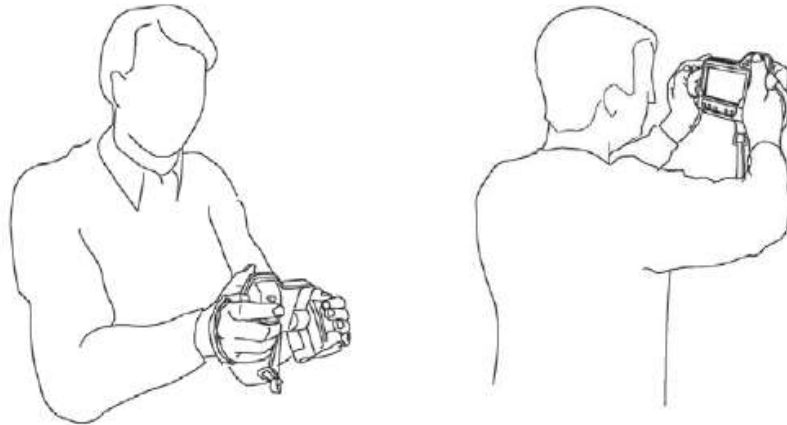
### **1.6.2.1 Ergonomía en el manejo de la Cámara**

Otro aspecto importante tiene que ver con la ergonomía. Hay que tener en cuenta que se puede estar tomando imágenes termográficas durante horas, o viendo un proceso de trabajo o fabricación, etc. Saber coger la cámara es importante, elegir una cámara que sea cómoda para la aplicación que se tenga será importante, incluso se llegará a usar un trípode para determinadas prácticas.

Por último, si se está trabajando con la cámara frente al objetivo, será útil mantener la pantalla despejada de menús. El termógrafo poco experimentado suele llenar la pantalla de información poco útil para la inspección, mostrando datos de emisividad, con el punto frío y punto caliente activos (lo cual va variando a medida que se mueve la cámara), etc. Interesa ver el objeto en infrarrojos, nada más, la capacidad de interpretación de lo que se ve es la herramienta más útil que se posee.

La ergonomía en el manejo de la cámara es un aspecto a tener en cuenta (ver figura 26).

**Figura 26. Ergonomía**



**Fuente: FLIR systems**

### **1.6.2.2 Accesibilidad**

La accesibilidad de los distintos botones es también un aspecto importante para poder trabajar de manera cómoda y rápida. A veces se tendrá la otra mano ocupada (ver figura 27).

**Figura 27. Accesibilidad**



**Fuente: FLIR systems**

### **1.6.3 Análisis de la Imagen Térmica**

Como termógrafo, se dispone de una sola herramienta de trabajo, la cámara, y con las imágenes obtenidas se realizarán los informes y se sacarán las conclusiones. Pero esto no es apuntar y disparar, no se trabaja en el visual sino en el infrarrojo, y esto tiene grandes diferencias que llevan a tener que entrenar al cerebro a interpretar las imágenes térmicas.

Existen dos métodos de análisis de las imágenes térmicas, los **análisis cualitativos** y los **cuantitativos**.

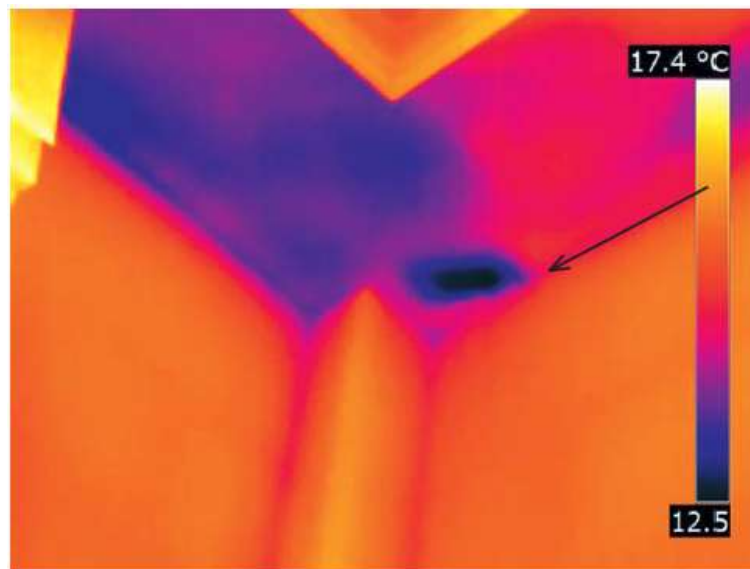
#### **1.6.3.1 Análisis Cualitativo**

La imagen térmica es analizada para poner de manifiesto anomalías de distinta magnitud, localizarlas y evaluar el nivel de gravedad. Es el primer análisis que hace el termógrafo, aunque se quiera hacer otro tipo de análisis, cuantitativo, inevitablemente al comienzo se tratará de rápidamente buscar aquello que en

nuestra cabeza no encaja o se ve que se aleja de lo normal para el objeto bajo estudio.

El 90% de las inspecciones serán de este tipo, lo cual no quiere decir que sea ni más sencillo ni más complejo, simplemente es un tipo de análisis.

**Figura 28. Imagen infrarroja de la esquina con humedades de una habitación**



**Fuente: FLIR systems**

La mancha azul oscura de la esquina de la habitación de la Fig. 28 revela la presencia de humedad. ¿Es necesario cuantificar cuánto agua entra? Más bien interesará saber por dónde entra y cómo evitar que entre.

Este análisis se basa en la temperatura aparente, sin compensación.

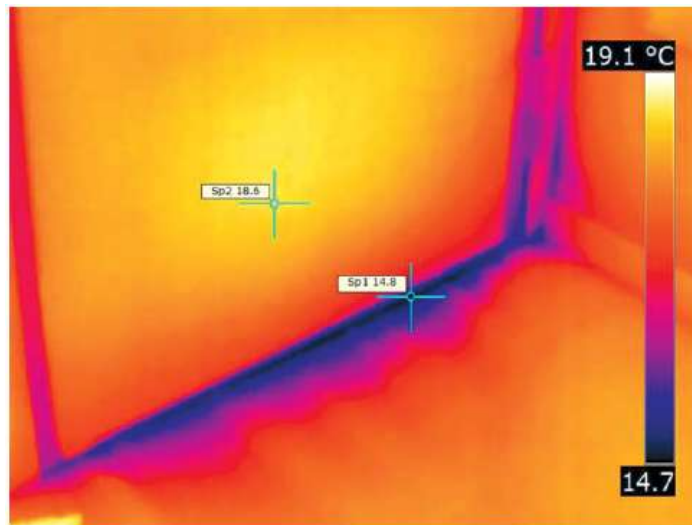
Es habitual que las empresas que realizan inspecciones termográficas tengan definidos unos niveles de gravedad, por ejemplo:

- Grado 1: Reparación Urgente
- Grado 2: Programar Reparación
- Grado 3: Revisar en la siguiente inspección

### 1.6.3.2 Análisis cuantitativo

Determina la temperatura o temperaturas de las partes de la imagen térmica que interesan para, a partir de éstas, extraer las conclusiones sobre las anomalías detectadas y las soluciones a adoptar.

**Figura 29. Imagen Infrarroja de Filtraciones de Aire**



**Fuente: FLIR systems**

En la Fig. 29 se podría hacer un análisis simplemente cualitativo, poniendo de manifiesto que existen infiltraciones de aire. Sin embargo, el cliente ha pedido que se cuantifiquen esas pérdidas, para lo se necesitan conocer algunos datos como la temperatura exterior e interior y la superficie por donde entra aire.

Este análisis se realiza con temperaturas reales, con compensación de los parámetros de objeto de emisividad y temperatura aparente reflejada.

De los resultados obtenidos se podrán sacar distintas conclusiones y saber si es relevante o no, si hay que actuar o no, en función de los criterios de clasificación de fallos antes mencionados, propios o publicados por algún organismo o asociación relevante.

También hay que tener y usar datos de referencia, es decir, datos históricos que permiten saber si una temperatura es elevada o baja para un mismo componente, y cuánto lo es respecto al mismo componente en estado normal.

#### **1.6.4 Compensación**

La compensación de la imagen se puede hacer, bien durante la inspección o con los software de tratamiento de imágenes en la oficina.

Todas las cámaras infrarrojas van a permitir modificar los “parámetros de objeto”, temperatura aparente reflejada, emisividad (ver tabla 4), distancia y temperatura exterior. Esto permite pasar de temperatura aparente a temperatura real.

Conviene en este punto, explicar los efectos que tiene la emisividad sobre la imagen térmica:

- Si se mide un cuerpo con alta emisividad, la temperatura aparente del cuerpo es cercana a su temperatura real. Se ve la realidad.
- Si se mide un cuerpo de baja emisividad, la temperatura aparente del cuerpo es cercana a las temperaturas aparentes de los cuerpos cercanos. Se ve la realidad y seguramente no se pueda medir o será francamente difícil.

Además de poder modificar en los parámetros de objeto la emisividad, se puede usar un truco para medir sin compensar, se puede pintar o cubrir la superficie que se quiere medir con algo de emisividad conocida, alta y constante. Por ejemplo, se puede colocar un trozo de cinta aislante (emisividad de 0,95) sobre la superficie de una tubería de cobre, la cinta se calentará a la temperatura del cobre y del líquido que circula por él, entonces se podrán hacer mediciones de temperatura sobre esta cinta.

### 1.6.5 Emisividad

Una vez que se sabe la radiación que viene de otras fuentes distintas al objetivo, ahora compensar la emisividad es ya el último paso y obstáculo, para tener el dato de la temperatura a partir de la radiación recibida.

Emisividad es la relación entre la radiación que emite un cuerpo real y la emitida por un cuerpo negro, para una misma temperatura y longitud de onda.

Traducido a una fórmula se tiene:

$$\epsilon = \text{WCR} / \text{WCN}$$

Donde  $\epsilon$  es la emisividad del objeto, **WCR** en la radiación del cuerpo real y **WCN** la radiación del cuerpo como si fuera totalmente negro.

O dicho de otra manera,  $\text{WCN} = \text{WCR} / \epsilon$ , y este es el cálculo que hace la cámara. Ésta conoce la radiación que recibe del objeto, si se introduce la emisividad, se divide y es como si la radiación la hubiera emitido un cuerpo negro y su temperatura fuera real. Se ha transformado un cuerpo real en un cuerpo negro en el que se puede leer la temperatura.

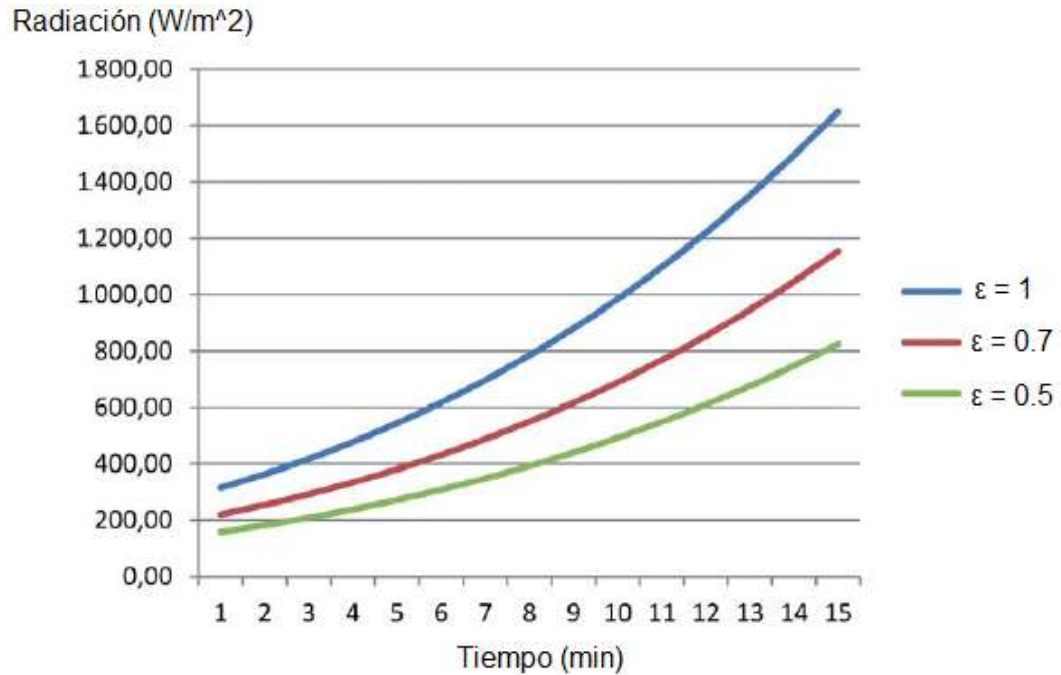
En la gráfica de la Fig. 30 que se muestra a continuación, igual a la mostrada para el cuerpo negro ( $\epsilon = 1$ ), se han añadido dos series más modificando el valor de emisividad.

La serie 2, para  $\epsilon = 0,7$

La serie 3, para  $\epsilon = 0,5$

Se ve que cuanto más nos alejamos del cuerpo negro, menos radiación percibe la cámara para la misma temperatura (eje de las Y).

**Figura 30. Variación de la Radiación en función de la Emisividad del cuerpo medido**



**Fuente: FLIR systems**

Como se puede ver, una misma imagen puede dar información completamente distinta y por tanto, resultados y conclusiones muy distintas. Hay que tener cuidado con esto, la emisividad del cuerpo, siempre desconocida, puede llevar a errores brutales y a diagnósticos desacertados.

#### **1.6.5.1 Factores que Afectan la Emisividad**

- El material del cuerpo: distintos materiales poseen distintas emisividades. Incluso un mismo material puede poseer distinta emisividad, esto se encuentra con frecuencia en materiales expuestos a determinadas condiciones ambientales, de manera que un material nuevo tiene una emisividad distinta pasados algunos años.

Tabla 4. Emisividad de algunos materiales

Material	Emissivity*	Material	Emissivity*
Aluminum, polished	0.05	Iron, wrought, polished	0.28
Aluminum, rough surface	0.07	Lacquer, Bakelite	0.93
Aluminum, strongly oxidized	0.25	Lacquer, black, dull	0.97
Asbestos board	0.96	Lacquer, black, shiny	0.87
Asbestos fabric	0.78	Lacquer, white	0.87
Asbestos paper	0.94	Lampblack	0.96
Asbestos slate	0.96	Lead, gray	0.28
Brass, dull, tarnished	0.22	Lead, oxidized	0.63
Brass, polished	0.03	Lead, red, powdered	0.93
Brick, common	0.85	Lead, shiny	0.08
Brick, glazed, rough	0.85	Mercury, pure	0.10
Brick, refractory, rough	0.94	Nickel, on cast iron	0.05
Bronze, porous, rough	0.55	Nickel, pure polished	0.05
Bronze, polished	0.10	Paint, silver finish**	0.31
Carbon, purified	0.80	Paint, oil, average	0.94
Cast iron, rough casting	0.81	Paper, black, shiny	0.90
Cast iron, polished	0.21	Paper, black, dull	0.94
Charcoal, powdered	0.96	Paper, white	0.90
Chromium, polished	0.10	Platinum, pure, polished	0.08
Clay, fired	0.91	Porcelain, glazed	0.92
Concrete	0.92	Quartz	0.93
Copper, polished,	0.01	Rubber	0.93
Copper, commercial burnished	0.07	Shellac, black, dull	0.91
Copper, oxidized	0.65	Shellac, black, shiny	0.82
Copper, oxidized to black	0.88	Snow	0.80
Electrical tape, black plastic	0.95	Steel, galvanized	0.28
Enamel **	0.90	Steel, oxidized strongly	0.88
Formica	0.93	Steel, rolled freshly	0.24
Frozen soil	0.93	Steel, rough surface	0.96
Glass	0.92	Steel, rusty red	0.69
Glass, frosted	0.96	Steel, sheet, nickelplated	0.11
Gold, polished	0.02	Steel, sheet, rolled	0.56
Ice	0.97	Tar paper	0.92
Iron, hot rolled	0.77	Tin, burnished	0.05
Iron, oxidized	0.74	Tungsten	0.05
Iron, sheet galvanized, burnished	0.23	Water	0.98
Iron, sheet, galvanized, oxidized	0.28	Zinc, sheet	0.20
Iron, shiny, etched	0.16		

Fuente: [www.fluke.com](http://www.fluke.com)

- Geometría: si es redondo, agujereado, liso, alargado, etc., influirá sobre su emisividad.
- Estructura superficial: las superficies lisas, pulidas, brillantes reflejan más que las rugosas. Sobre una superficie rugosa se obtendrá un mejor valor de la temperatura pues su emisividad es mayor.
- El ángulo de la imagen: si nos colocamos perpendiculares al objetivo, nos reflejaremos sobre él, si nos esquinamos mucho, serán otros objetos los que se reflejen en la imagen.
- La temperatura: por suerte esto sólo ocurrirá a temperaturas muy altas y en los metales. Algunos metales, a temperaturas cercanas a los 1.000 °C pueden incrementar su emisividad.
- El color del objeto: ¡ERROR! El color es algo del espectro visual, no del infrarrojo, las cámaras no saben de qué color son los objetos que enfocan, el color visual no afecta a la emisividad. Sí es cierto que los colores absorben luz visible de una u otra forma, la radiación solar es más alta en la franja visual del espectro que en la del infrarrojo, así algunos colores absorben más radiación que otros y por tanto se calientan más (e irradian más a nuestra cámara), pero no tiene que ver con la emisividad.

Estas cosas conviene recordarlas e ir experimentando con ellas, aprendiendo incluso a base de errores. Muchas veces el usuario de la cámara termográfica simplemente se dedica a tomar imágenes para luego centrarse en complejos cálculos, sin saber que el dato de partida, la imagen, es errónea por cosas tan simples como el ángulo, un reflejo, etc.

### 1.6.6 Reflejos

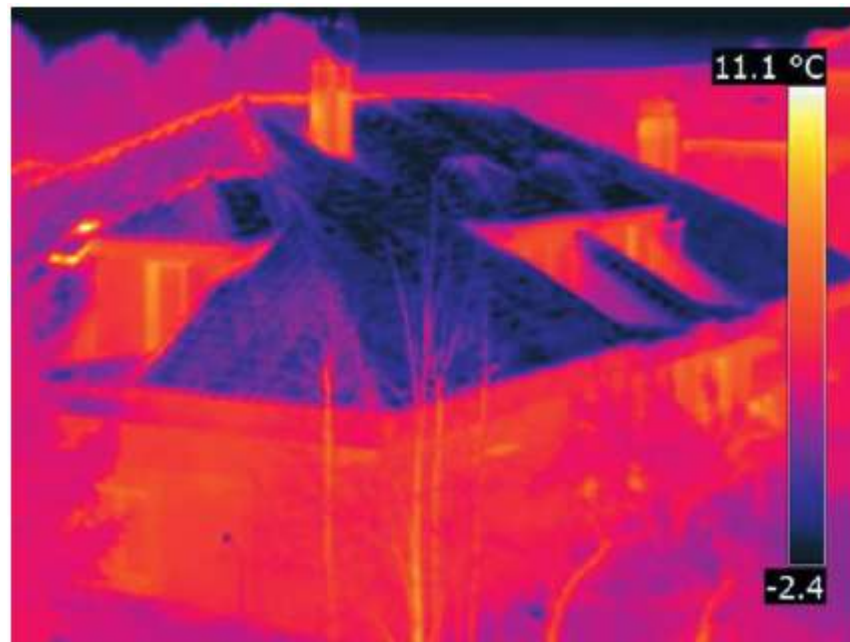
No sólo la luz visible se refleja, en un espejo por ejemplo, la radiación infrarroja también, confundiendo al termógrafo más experimentado, por ejemplo los metales pulidos son casi espejos térmicos. No se trata de una herramienta del termógrafo sino de algo de lo que hay que huir a la hora de analizar las imágenes térmicas. El objeto de medida siempre estará influenciado por distintas fuentes de reflexión y éstas incidirán sobre nuestra cámara. Descubrirlos y evitarlos es fundamental.

En muchas ocasiones el termógrafo se ve reflejado en el objeto que está inspeccionando. Se puede descubrir fácilmente, si se mueve y el posible punto caliente se mueve, será su reflejo, o si cambia el ángulo de enfoque de la cámara y el reflejo desaparece, ya sabe de qué se trata.

¿Cómo evitarlos?

- Evitar ser uno mismo el reflejo, no situarse delante del objeto de medida.
- Si es inevitable, moverse y si el punto caliente se mueve, al menos se sabrá que es uno mismo.
- En los reflejos, el ángulo de incidencia es igual al de salida, sabiendo esto se puede evitar la fuente de reflejo.
- Podemos apantallar la fuente de reflejo, con cartón, etc.
- Intentar medir en las partes del objeto que tienen alta emisividad, estas no engañarán, su temperatura será más real.

**Figura 31. Imagen Infrarroja del Exterior de una Vivienda**



**Fuente: FLIR systems**

En termografías exteriores, el cielo y el sol son quizás las fuentes de reflexión más importantes. En la imagen de la Fig. 31 se podría pensar que el techo de esta vivienda tiene un problema de humedad. Más bien se trata del reflejo del cielo en la teja. El ángulo del tejado respecto a la cámara facilita esa reflexión. Se sabe que el techo no está a una temperatura tan baja.

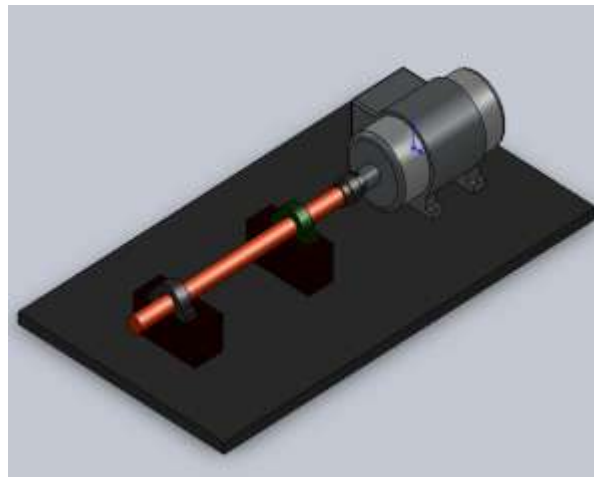
## 2. BANCO PARA PRUEBAS TERMOGRÁFICAS

### 2.1 MÓDULOS DE PRUEBA

El banco de pruebas para tomografía infrarroja consta de 6 módulos los cuales permiten observar diferentes aplicaciones para analizar con cámara termográfica, a continuación se va presentar una breve descripción de cada modulo y su funcionamiento.

#### 2.1.1 Módulo de calentamiento de rodamientos

**Figura 32. Vista superior montaje del motor**



**Fuente: Autores Proyecto**

Cuando los componentes mecánicos se desgastan y pierden eficiencia suelen disipar más calor. Como resultado en este caso, de un montaje defectuoso, las chumaceras aumentarán su temperatura antes de averiarse ya que se incurrió a propósito en fallos oxidando una de ellas en el montaje.

Este consta de un motor eléctrico monofásico el cual permite revisar los signos de desgaste en el contacto de las escobillas y los cortocircuitos en los armazones

donde se produce un calor excesivo antes del fallo, pero son imposibles de detectar mediante un análisis de vibraciones ya que generan poca o ninguna. La termografía ofrece una visión completa y, en este caso, permite comparar las temperaturas de funcionamiento normal del motor sin carga con alteraciones en el sistema.

### 2.1.1.1 Selección del material soporte

La base donde va montado el sistema fué seleccionada según la siguiente tabla:

**Tabla 5. Espesores de la lámina de acero**

Calibre	Espesores			Peso por hoja					
	mm.	pulg.	Límite	kg/m <sup>2</sup>	3' x 6'	3' x 8'	3' x 10'	4' x 8'	4' x 10'
10	3.42	0.1345	0.1405 0.1285	27.471	45.92	61.23	76.54	81.65	102.06
11	3.04	0.1196	0.1256 0.1136	24.420	40.82	54.43	68.04	72.57	90.72
12	2.66	0.1046	0.1106 0.0987	21.365	35.30	47.06	58.78	62.70	78.31
13	2.28	0.0897	0.0947 0.0847	18.315	30.62	40.82	51.03	54.43	68.04
14	1.90	0.0747	0.0797 0.0697	15.262	25.515	34.020	42.525	45.36	56.70
16	1.52	0.0598	0.0648 0.0548	12.210	20.412	27.216	34.020	36.29	45.36

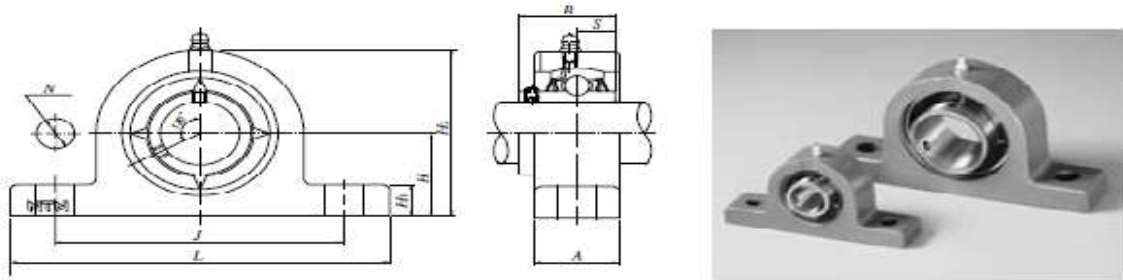
Fuente: [www.bideco.com.mx](http://www.bideco.com.mx)

### 2.1.1.2 Selección de las chumaceras utilizadas

Las chumaceras utilizadas en el montaje se seleccionaron según la tabla ofrecida por NTN catalogo CAT .No 3902/E.

Tabla 6. Selección de chumacera

**Pillow Block Unit; UCPG2 series**  
Cylindrical bore with set screw



Bore dia. mm	Part number	Boundary dimensions mm									Nominal hole dia.	Insert bearing number	Basic dynamic rated load kN Cr	Basic static rated load kN Cor	Housing part number	Max. kg (lb.)
		H	L	J	A	N	H1	H2	B	S						
12	UCPC201D1	30.2	127	05	25	12	14	62	34	12.7	M10	UC201D1	12.8	6.65	PC203D1	0.7
16	UCPC202D1							65				UC202D1				
17	UCPC203D1							65				UC203D1				
20	UCPC204D1	33.3	65	UC204D1	PC204D1											
25	UCPC205D1	36.5	140	105	30	15	71	34	14.3	UC205D1	14.0	7.85	PC205D1	0.8		
30	UCPC206D1	42.0	165	121	35	17	83	38.1	15.0	UC206D1	10.5	11.3	PC206D1	1.3		

Fuente: NTN, CAT No.3902/E

Para el acople del eje con las chumaceras, se utilizó un acople flexible por varias ventajas mecánicas que nos ofrece, como:

- Facilidad de montaje y conexión axial.
- Buena capacidad de absorción de vibraciones.
- Libres de mantenimiento (no requieren lubricación).
- A prueba de fallas (siguen trabajando aún si el elastómero se daña).
- Resistentes al aceite, polvo, arena, grasa, humedad y muchos solventes.
- Por la casi inexistente “holgura o juego” entre sus componentes ofrecen gran precisión de posicionamiento.

### 2.1.1.3 Selección del acople flexible

Primero se debe calcular el torque nominal con la siguiente fórmula:

$$In-lbs = T_n = (HP \times 63025)/RPM$$

HP=0.5 RPM=3440

Tn= 9.16061 in-lbs.

Se estima el factor de servicio.

**Tabla 7. Selección del acople flexible**

TIPO DE TRABAJO O MÁQUINA	Motor Eléctrico		Motores de combustión			
	Torque estándar	Alto Torque	4 o más cilindros	3 cilindros	2 cilindros	1 cilindro
Operación uniforme Ejemplo: Bombas hidráulicas y centrífugas, generadores livianos, ventiladores, sopladores, transportadores de banda o tornillo.	1.0	1.25	1.5	1.7	2.0	2.7
Operación uniforme con fluctuación moderada Máquinas para el trabajo de la madera, molinos, maquinaria textil, mezcladores.	1.6	1.8	2.0	2.3	2.5	3.0
Operación fluctuante Hornos rotativos, máquinas litográficas y de impresión, generadores, bombas para líquidos viscosos.	1.8	1.9	2.2	2.5	2.7	3.1
Operación fluctuante con choques Mezcladoras de concreto, martinetes, molinos para papel, bombas de compresión, bombas de propela, entorchadoras de cable, centrífugas.	1.8	2.0	2.5	2.7	3.0	3.4
Operación muy fluctuante con choques Excavadoras, molinos de bolas, bombas de pistón, prensas de forja y estampado.	2.1	2.3	2.7	3.0	3.4	3.8
Trabajo pesado, muy fluctuante con choques fuertes Compresores y bombas de pistón, movimiento de rodillos pesados, estrusoras de ladrillo, prensas de mandíbula para moler piedra.	2.5	3.1	3.3	3.6	4.0	4.5

Fuente: [www.Intermec.com](http://www.Intermec.com)

Se toma un factor de servicio de 1 debido a que no está trabajando bajo carga en sistema y no tiene operaciones fluctuantes.

El fabricante intermec ofrece una tabla de selección para acoples flexibles (interflex) según su torque nominal.

**Tabla 8. Selección de acople Interflex**



INTERFLEX No.	Torque nominal (Tn) in-lbs	Torque de diseño (Td) in-lbs	Hueco máximo con manzana: Estandar	Hueco máximo con manzana: Extra -Grande	RPM Máximas permisibles	Dimensiones en mm		
						A	B	L
GE 14	66	133	16		19,000	30	30	34
GE 19	89	177	19	24	14,000	40	40	64
GE 24	310	620	24	32	10,600	55	55	77
GE 28	841	1,682	28	38	8,500	65	65	88
GE 38	1,882	3,363	38	45	7,100	80	66	111
GE 42	2,345	4,691	42	55	6,000	95	75	125

Fuente: [www.Intermec.com](http://www.Intermec.com)

Se utilizó un eje de acero 10-20 de 36 cm de largo y diámetro de 19 mm; tiene un proceso de galvanizado superficial para evitar que se presente corrosión del mismo.

A continuación, una imagen de la descripción del motor:

Figura 33. Placa del motor



Fuente: Autores Proyecto / SIEMENS

Figura 34. Montaje del motor



Fuente: Autores Proyecto

### 2.1.2 Modulo intercambiador de calor por aletas

El modulo consiste en un intercambiador Aleteado según la siguiente disposición:

**Figura 35. Módulo Intercambiador de calor por aletas**



**Fuente: Autores Proyecto**

En este modulo se observará el comportamiento de la disipación de energía de un intercambiador por aletas. También, se pintará la mitad de las aletas de un color brillante para modificar la emisividad con la que la cámara trabaja y la otra mitad seguirá de color negro; esto influirá en la toma de datos y se demostrará la importancia de este parámetro al mostrar una temperatura diferente a la verdadera. El modulo dispone de un disipador, este tiene en su centro acoplada una resistencia térmica.

La resistencia opera con:

$$V=125.5v$$

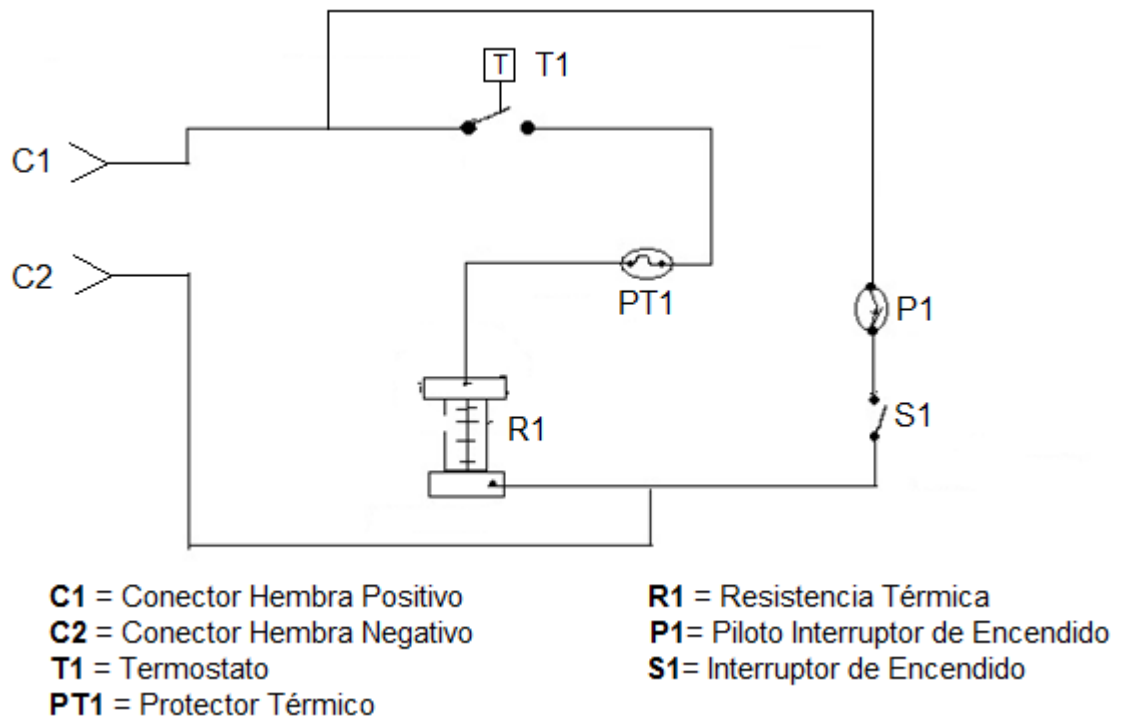
$$R=90.6 \Omega$$

$$\text{Potencia}= 173,8 \text{ Watts}$$

El sistema también dispone de un interruptor con piloto para encendido, también tiene un termostato que permite abrir o cerrar el circuito en función de la temperatura de la resistencia eléctrica.

El disipador esta sujetado en una caja de madera donde internamente se encuentran dispuestas las conexiones eléctricas .El disipador está aislado térmicamente de la madera mediante una lamina de caucho para impedir el calentamiento de esta , a continuación se presenta un breve esquema eléctrico.

**Figura 36. Esquema eléctrico módulo intercambiador**

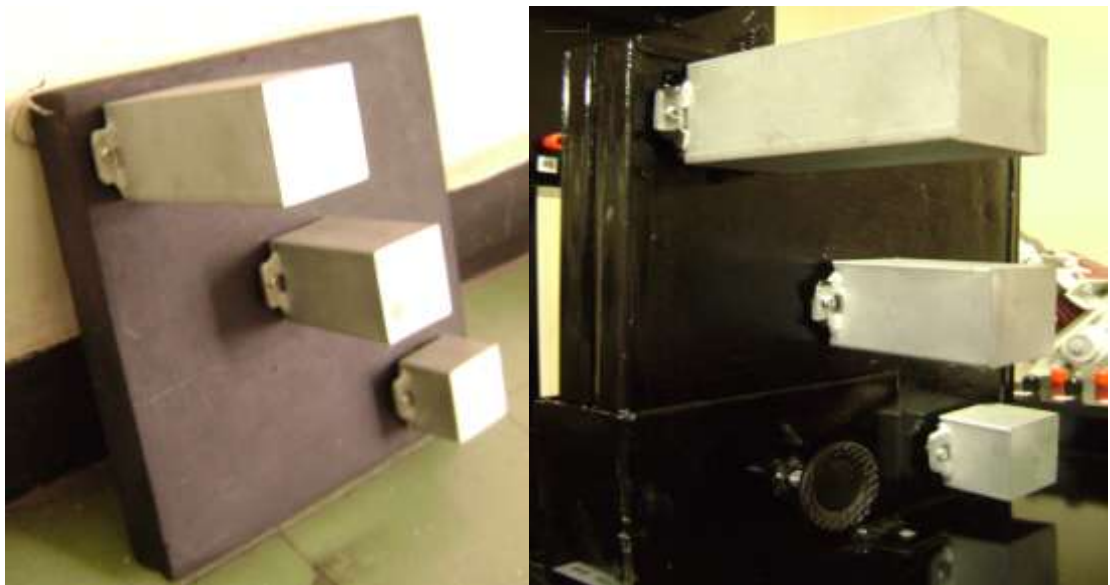


**Fuente: Autores Proyecto**

### 2.1.3 Módulo de profundidad, transferencia de calor por conducción

En este módulo se busca observar la transferencia de calor por conducción y verificar si esta propiedad (profundidad) altera la toma de datos. Está construido según la siguiente disposición.

**Figura 37. Módulo Profundidad**



**Fuente: Autores Proyecto**

Este modulo dispone de tres barras de acero 10-20 de perfil cuadrado de 4.5 cm de lado empotradas a una placa de concreto de 25 X 25 cm, atornilladas en sus extremos con chazo; las barras tienen respectivamente 4, 8 y 12 cm de largo y tiene un galvanizado superficial para evitar la oxidación.

Cada barra tiene en su extremo una resistencia térmica de 161.727 watts.

#### **2.1.4 Módulo de tamaño**

Consiste en dos placas de acero inoxidable cuadradas de diferentes tamaños (ver figura 39). Lo que se busca en este modulo es observar y analizar la progresión del calentamiento para áreas diferentes a iguales tiempos. También se verificará si esta propiedad (tamaño) altera la toma de datos en un cuerpo del mismo material.

Se utilizó acero inoxidable de la categoría de los ferríticos, serie 400 AISI (American Iron & Steel Institute) los cuales mantienen una estructura ferrítica estable desde la temperatura ambiente hasta el Punto de fusión, sus características son:

- Resistencia a la corrosión de moderada a buena, la cual se incrementa con el contenido de cromo y algunas aleaciones de molibdeno.
- Endurecidos moderadamente por trabajo en frío: no pueden ser endurecidos por tratamiento térmico.
- Son magnéticos.
- Su soldabilidad es pobre por lo que generalmente se eliminan las uniones por soldadura a calibres delgados.
- Usualmente se les aplica un tratamiento de recocido con lo que obtienen mayor suavidad, ductilidad y resistencia a la corrosión.
- Debido a su pobre dureza, el uso se limita generalmente a procesos de formado en frío.

La tabla 9 presenta algunas propiedades térmicas y mecánicas del acero serie AISI 400.

Las placas de acero cuadrado están dispuestas en 2 tamaños diferentes, están greteadas y lijadas, en sus extremos tienen perforaciones para poder ajustarlas a las tabletas de cerámica gres, perforadas también, y se ensamblan en madera.

Se utilizaron tabletas de cerámica ya que es un buen aislante térmico que soporta grandes temperaturas.

**Tabla 9. Propiedades físicas y mecánicas del acero inoxidable**

TIPO DE ACERO		403	405	409	410	416 *
ANÁLISIS - % VALOR QUÍMICO MÁXIMO EXCEPTO DONDE MUESTRA EL MÍNIMO	Cromo	11.5-13.0	11.5-14.5	10.5-11.75	11.5-13.5	12-14
	Níquel	--	--	--	--	--
	Carbón	.15	.08	.08	.15	.15
	Manganeso	1.0 Min	1.0	1.0	1.0	1.25
	Silicio	.5	1.0	1.0	1.0	1.0
	Fósforo	.040	.040	.040	.040	.060
	Azufre	.30	.30	.30	.30	*
	Otros	--	Al .1 - .3	Ti 6xC Min 0.75 Max	--	Mo .60 (OPT)
PROPIEDADES MECÁNICAS (TEMPLADO):	Esfuerzo de fluencia psi (0.2% Compensación)	40.000	40.000	35.000	45.000	83.000
	Esfuerzo último psi	75.000	65.000	65.000	70.000	105.000
	Elongación % en 2" (100mm)	35	25	25	25	20
	Dureza: Brinell BHN	153	150	150	150	225
	Rockwell B	82	75	75	80	97
	Impacto Izod. Ft. - lbs	90	20	20	90	25
PROPIEDAD ELECTRICAS	Creep - 1% flujo en 10.000 hrs	12.000	8.000	10.500	12.000	9.000
	Módulo elasticidad en tensión psi x10 <sup>6</sup>	29	29	29	29	29
RESISTENCIA TÉRMICA	Resistividad eléctrica en microhm a 68 ° F	57	60	59	57	57
	Permeabilidad magnética a 200H	--	--	--	--	--
EXPANSION TÉRMICA	Max. temperatura de operac. - servicio intermitente ° F	1450	1450	1425	1450	1400
	Servicio continuo ° F	1300	1400	1400	1300	1250
CONDUCTIVIDAD TÉRMICA	(In./In./°F x 10 <sup>-3</sup> ) 32 - 212 °F	5.5	6.0	6.5	5.5	5.5
	32 - 1200 ° F	6.5	7.0	7.2	6.5	6.5
CONDUCTIVIDAD TÉRMICA	(B.T.U./Ft. /Hr)° F/ft.) y 212 °F	14.4	14.6	14.4	14.4	14.4
	932 ° F	16.6	16.4	16.6	16.6	16.6

Fuente: [www.wesco.com.co](http://www.wesco.com.co)

Este modulo tiene resistencias térmicas según la siguiente disposición de ajuste a la placa de acero, que también se utilizó en el modulo de forma.

**Figura 38. Resistencias térmicas**



**Fuente: Autores Proyecto**

El módulo de tamaño se puede apreciar en la siguiente imagen.

**Figura 39. Módulo de Tamaño**



**Fuente: Autores Proyecto**

### 2.1.5 Módulo de forma

En este modulo, lo que se busca es analizar con termografía infrarroja y reconocer la forma de las figuras por medio de matlab, su configuración igual al modulo de tamaño. Dispone de 4 placas de acero inoxidable de formas simétricas conocidas y aisladas.

**Figura 40. Módulo de Forma**



**Fuente: Autores Proyecto**

Las resistencias térmicas para este modulo y el de tamaño consumen aproximadamente 25 watts.

### 2.1.6 Módulo de puntos

El modulo de puntos me permite identificar con termografía infrarroja y con matlab la localización de áreas a diferentes temperaturas, contiene 4 resistencias cilíndricas cada una con un sensor de temperatura.

**Figura 41. Resistencias cilíndricas y sensores**



**Fuente: Autores Proyecto**

Cada resistencia tiene una potencia de consumo de 62.45 watts , para cada una se acopló un interruptor térmico que básicamente detecta la temperatura a la cual se encuentra ajustado y dependiendo de un aumento de esta o disminución, abre o cierra el circuito eléctrico respectivamente para el paso de corriente eléctrica.

Se utilizaron 4 sensores de diferentes temperaturas, 150,105, 90 y 70 grados centígrados respectivamente. Se utilizan para el control automático de la temperatura y para la protección contra el sobrecalentamiento de aparatos para uso doméstico y otros aparatos eléctricos.

**Especificaciones de los sensores accionados por temperatura (Termistor).**

- Tipo: normalmente cerrado.
- Exceso de temperatura: 5-10°C / minuto.
- Rigidez dieléctrica: 1500Vac / 1 minuto.
- Rango de temperatura: 60° - 160°C.
- Tolerancia de operación:  $\pm 5\%$ .

- Potencia de entrada: 240 Vac.
- Reinicialización automática (reset).
- Duración de vida de los contactos: 240Vac / 6A - 30000 ciclos.
- rigidez dieléctrica: 1500Vac / 1 minuto.

Cada resistencia está dispuesta y alojada en una estructura de cerámica tipo baldosa.

**Figura 42. Módulo Posición**

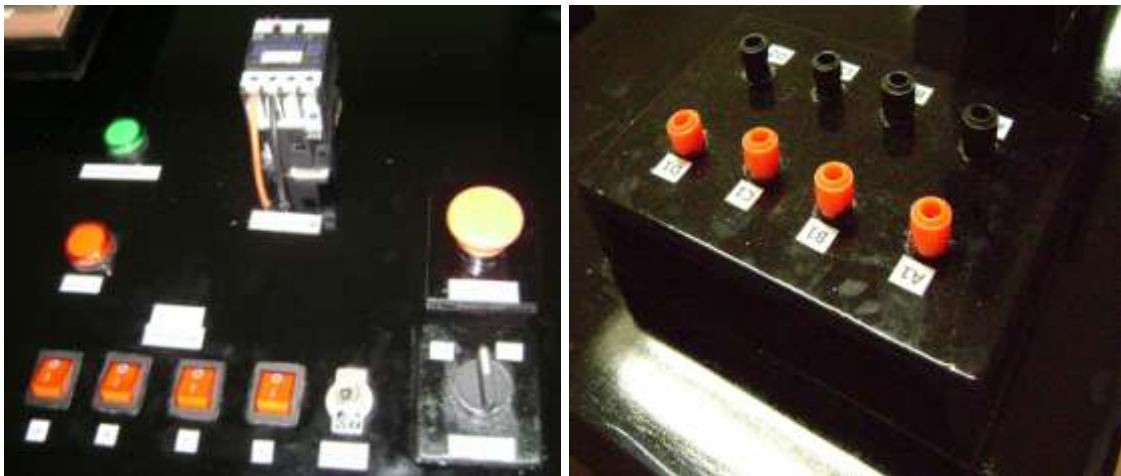


**Fuente: Autores Proyecto**

## 2.2 SISTEMA ELÉCTRICO DEL BANCO

El sistema eléctrico del banco es simple, consta de 4 interruptores con piloto que permiten el paso de corriente eléctrica a cada módulo que se esté usando; un guarda motor para proteger el motor de sobrecargas de corriente en el arranque; dos pilotos, uno de ellos indica el encendido y energía del banco y el otro la puesta en marcha del motor eléctrico y dos interruptores, uno para el encendido general del banco y otro para la parada de emergencia

**Figura 43. Sistema eléctrico del banco a) Encendido, b) Conectores**



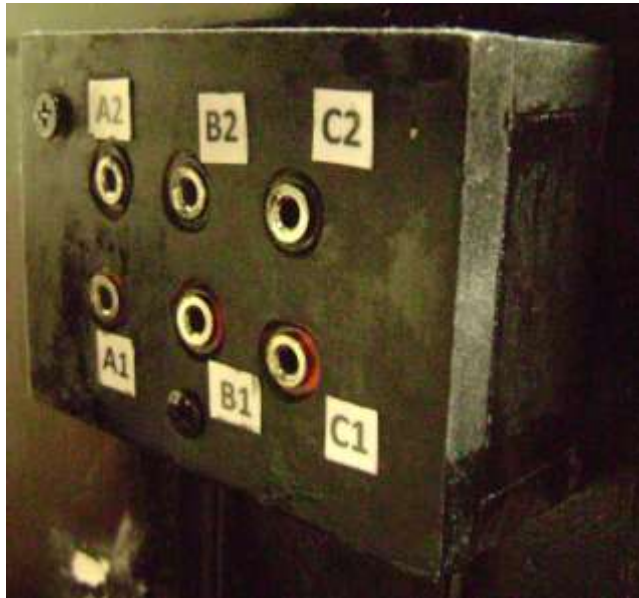
(a)

(b)

**Fuente: Autores Proyecto**

Cada interruptor energiza un par de conectores hembra de banana para poder energizar los módulos. Los módulos en su parte posterior tienen conectores hembra de banana para poder hacer la conexión eléctrica.

**Figura 44. Conexión Hembra**



**Fuente: Autores Proyecto**

La conexión para cada modulo se hace por medio de cable con banana.

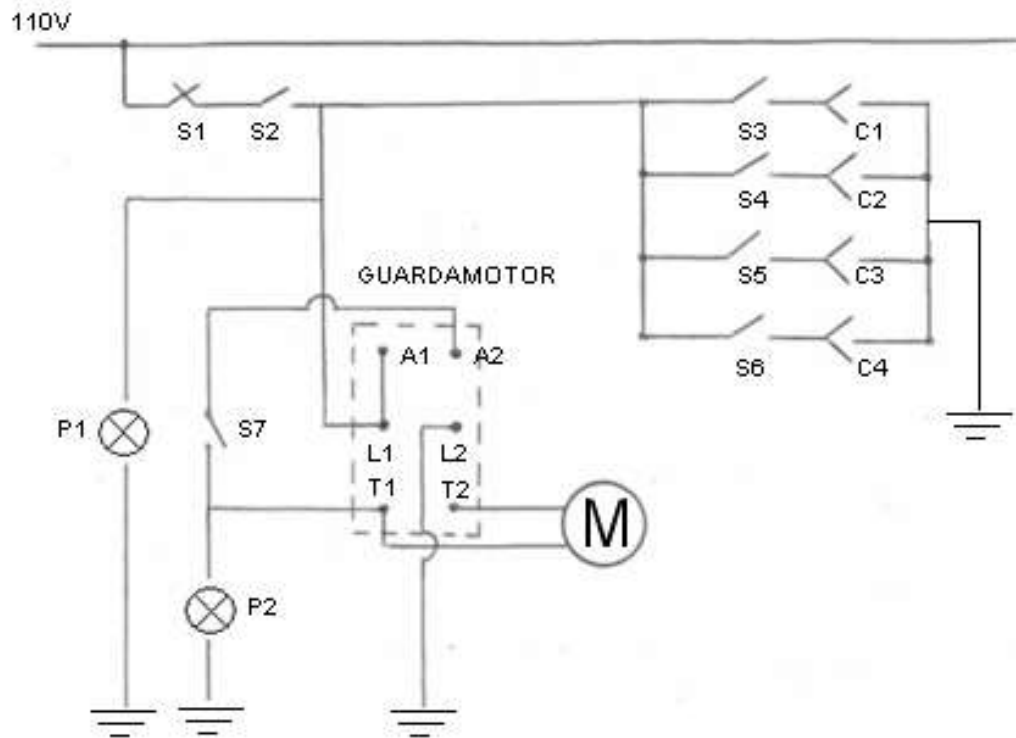
**Figura 45. Cables con banana**



**Fuente: Autores Proyecto**

A continuación se presenta el plano eléctrico de las conexiones del banco y la imagen del ensamble final del banco, la estructura metálica del banco se realizó bajo condiciones de ergonomía y funcionalidad.

**Figura 46. Plano eléctrico del banco**



S1=Parada de emergencia	P1=Piloto encendido del banco
S2=Encendido del banco	P2=Piloto encendido del motor
S3,S4,S5,S6=Interruptores Módulos	C1,C2,C3,C4=Conectores hembra para banana
S7=Encendido Motor	

**Fuente: Autores Proyecto**

**Figura 47. Banco de Trabajo**



**Fuente: Autores Proyecto**

### 3. PROGRAMA DE ANÁLISIS MATLAB

Los módulos mostrados en el capítulo anterior no solo prestan una ayuda para la transferencia de calor, también sirven como herramientas prueba para la detección de fallas en maquinaria y tratamiento de imágenes.

El programa sirve como medio de inspección al momento de ejecutar la aplicación que se desee, dando a conocer diferentes resultados ligados a forma, tamaño, puntos críticos o temperatura. Esto se da por medio de imágenes o videos que se procesarán y brindarán los datos necesarios para la verificación de las funciones a realizar.

#### 3.1 TRATAMIENTO DE IMÁGENES

Esta aplicación permite analizar imágenes según el toolbox de tratamiento de matlab con sus respectivas funciones mostradas en el capítulo 1. Cada módulo, ya descrito, brinda una utilidad y resultados específicos para este programa. El código pertinente se encuentra en el anexo D.

**Tabla 10. Aplicación de Módulos en el programa**

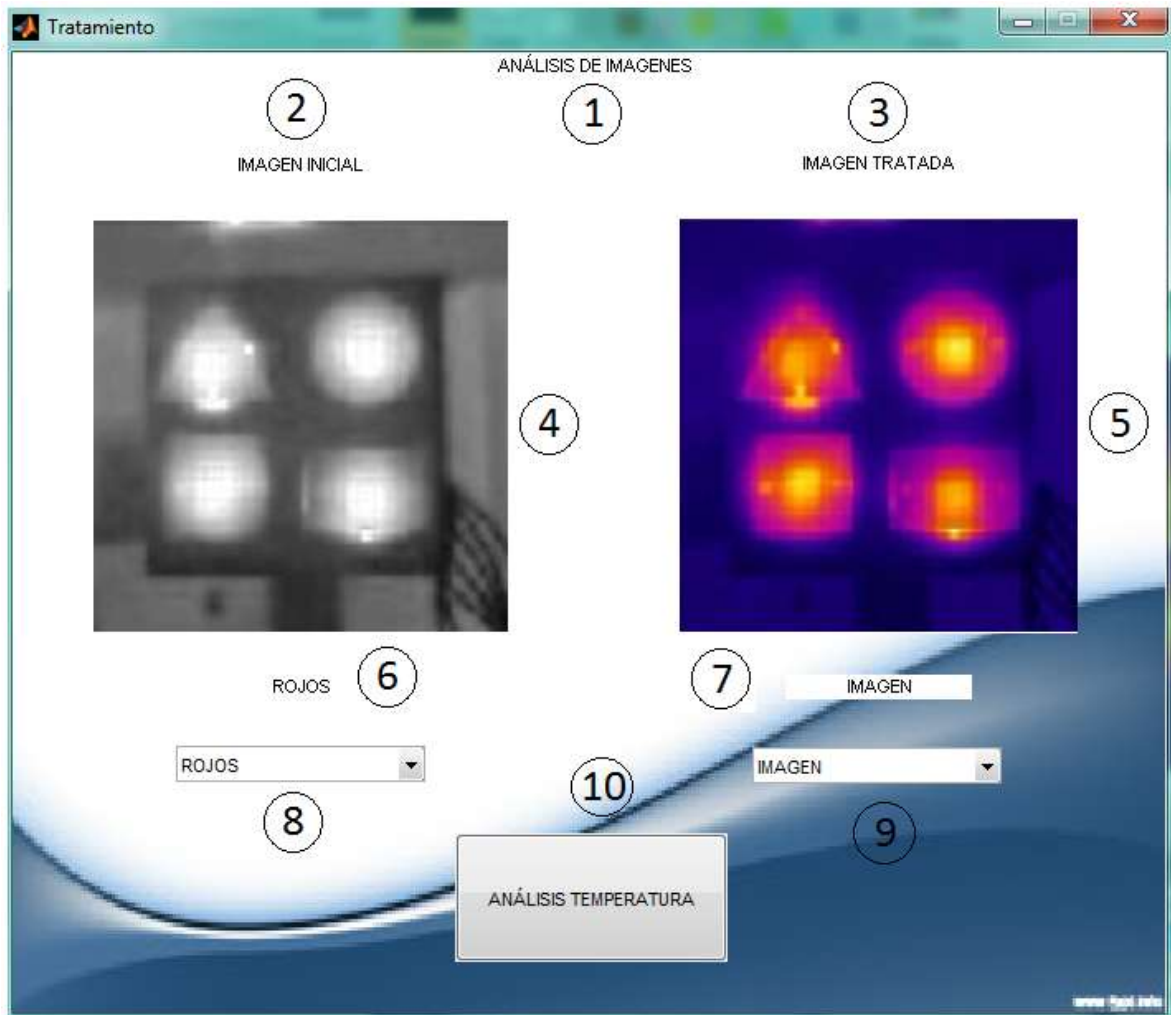
<b>MÓDULO</b>	<b>APLICACIÓN</b>
MOTOR	FALLAS POR TEMPERATURA EN RODAMIENTOS
ALETAS	TRANSFERENCIA DE CALOR
TAMAÑO PROFUNDIDAD	IDENTIFICACIÓN DE ÁREAS
FORMA	IDENTIFICACIÓN DE FIGURAS GEOMÉTRICAS
POSICIÓN	IDENTIFICACIÓN DE PUNTOS CALIENTES

**Fuente: Autores Proyecto**

### 3.1.1 GUIDE

La interfaz que se aplica contiene cinco (5) cuadros de texto estático, dos (2) popup menú, dos (2) axes y un (1) pushbutton, a los cuales se les ha asignado un número en la siguiente figura para ser explicados.

**Figura 48. GUIDE tratamiento de imágenes**



**Fuente: Autores Proyecto**

1. **Texto estático 1:** Título de la aplicación.
2. **Texto estático 2:** Nombre del axes1 (4).
3. **Texto estático 3:** Nombre del axes2 (5).
4. **Axes 1:** Espacio que muestra la imagen inicial y su respectivo tratamiento.
5. **Axes 2:** Espacio que muestra la imagen a comparar con la inicial y su respectivo tratamiento.
6. **Texto estático 4:** Nombre de la opción escogida de la lista de la imagen inicial (8).
7. **Texto estático 5:** Nombre de la opción escogida de la lista de la imagen a comparar con la inicial (9).
8. **Popup menú 1:** Lista de opciones (imagen, rojos e histograma) de la imagen inicial.
9. **Popup menú 2:** Lista de opciones (imagen, rojos, histograma, tamaño, forma y puntos calientes) de la imagen a comparar con la inicial.
10. **Pushbutton 1:** Botón accionador del análisis de temperatura y su respectiva alerta.

### 3.1.2 Funcionamiento

Se divide básicamente en siete acciones (ver figura 51):

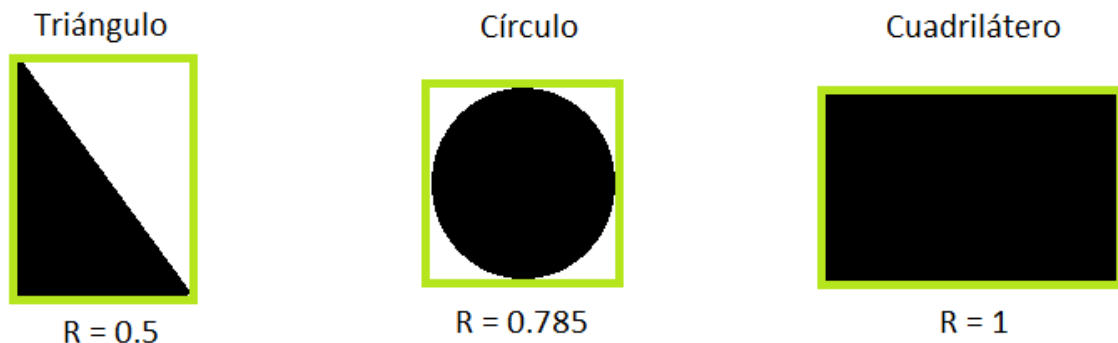
1. **Imagen:** Es la acción más sencilla, lee la imagen guardada mediante el comando `Imread` y la muestra en el axes deseado por medio del comando `imshow`.
2. **Rojos:** Lee la imagen guardada mediante `Imread`, la convierte al plano ROJO y la muestra en el axes deseado con `imshow`.
3. **Histograma:** Lee la imagen guardada por medio de `Imread`, la convierte al plano ROJO y muestra su histograma en el axes deseado con `Imhist`.

**4. Forma:** Lee la imagen guardada mediante `Imread`, aumenta su brillo para captar mejor los píxeles, la convierte al plano ROJO (color más sobresaliente en una imagen térmica), se extraen los bordes mediante el logaritmo de canny, dilata estos bordes por medio de una rejilla cuadrada de 4 píxeles de espesor, rellena los espacios entre puntos para crear, aproximadamente, el perfil de la figura, elimina las áreas menores consideradas como ruido, etiqueta cada una de las figuras resultantes y guarda sus propiedades específicas como área y boundingbox, grafica un rectángulo (boundingbox) sobre cada figura, crea una relación del área de la figura resultante del proceso con el área abarcada por este último (ver figura 49), según esta relación se define que tipo de figura geométrica es y la muestra en el axes deseado mediante `imshow`.

El **boundingbox** es una propiedad que el toolbox de tratamiento de imágenes de MATLAB permite utilizar. Consiste en un rectángulo que abarca toda la figura y se extiende hasta los límites de esta.

Las figuras obtenidas mediante imágenes térmicas no son ideales, por lo tanto, las relaciones establecidas no serán iguales ( $=$ ) sino mayores ( $>$ ) o menores ( $<$ ) con valores aproximados, establecidos por pruebas en el laboratorio.

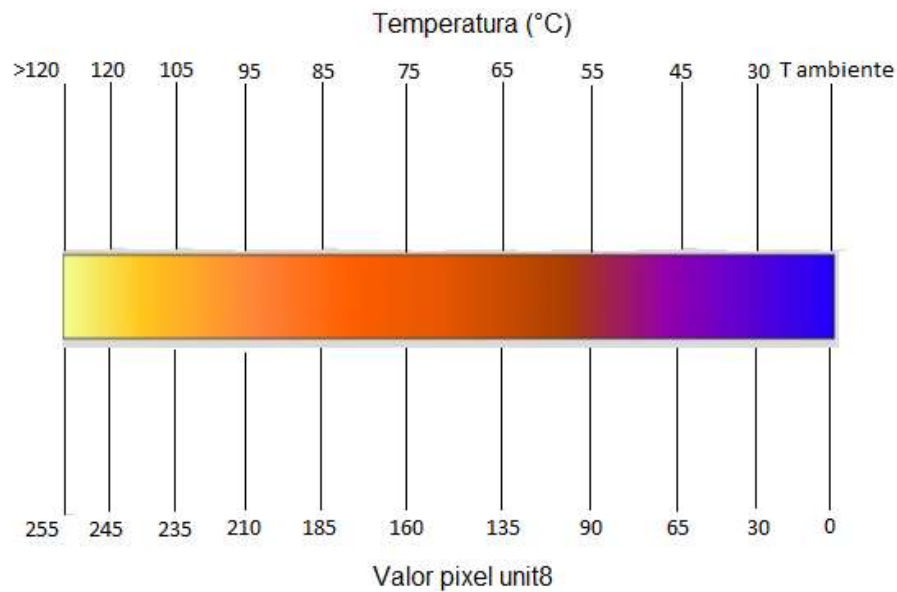
**Figura 49. Relación ideal entre el área de la figura y el boundingbox**



**Fuente: Autores Proyecto**

**5. Puntos calientes:** Lee la imagen guardada con `Imread`, la convierte al plano ROJO, obtiene su número de filas y columnas mediante el comando `size`, crea un bucle para recorrer todas las filas seguido de otro para todas las columnas, guarda el valor del pixel en `unit8` (0 - 255) de su respectiva coordenada, asigna un color relacionado a temperatura según el resultado obtenido (ver figura 50) y muestra la imagen en el axes deseado por medio de `Imshow`.

**Figura 50. Relación entre color, temperatura y valor de pixel en `unit8`**



**Fuente: Autores Proyecto**

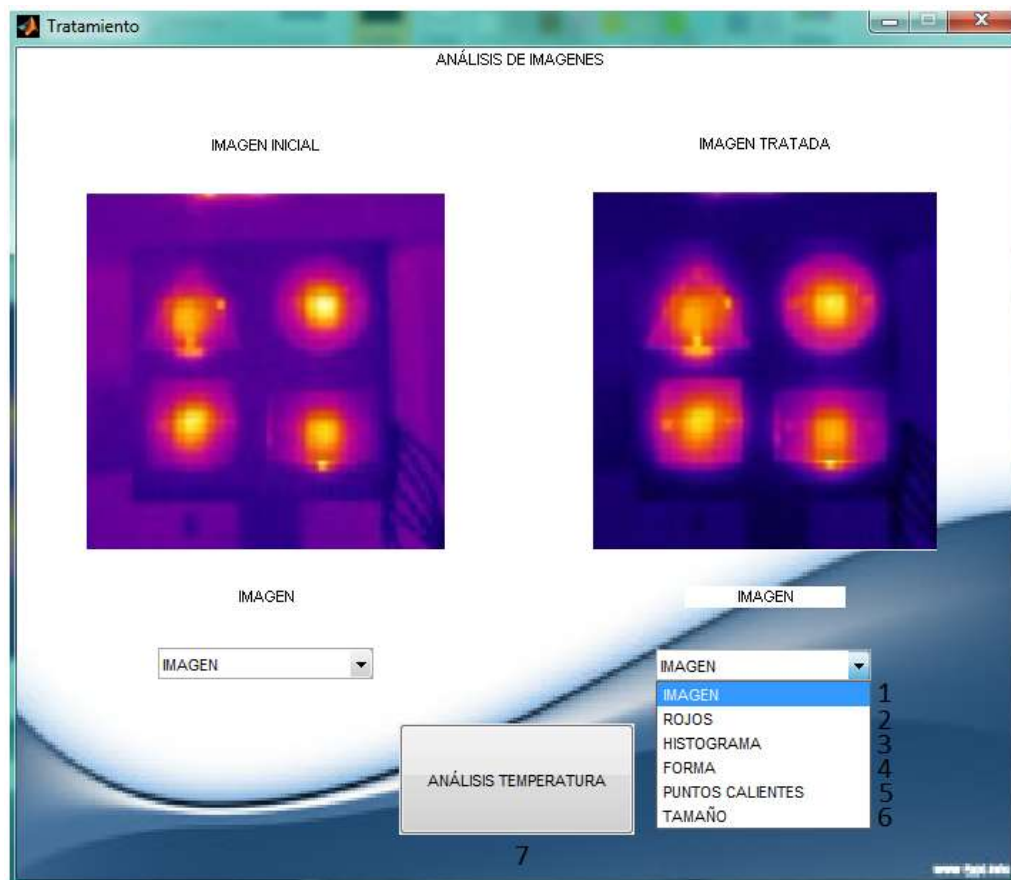
**6. Tamaño:** Lee la imagen mediante `Imread`, aumenta el brillo, convierte al plano ROJO, guarda el umbral y con este la binariza, elimina áreas pequeñas consideradas como ruido, etiqueta cada una de las figuras resultantes y guarda sus propiedades principales como área y `boundingbox`, dibuja un rectángulo (`boundingbox`) sobre cada figura, relaciona las áreas resultantes con la mayor y la menor mediante los comandos `max` y `min` respectivamente, según los resultados anteriores, nombra las figuras y las muestra en el axes deseado con `Imshow`.

**7. Análisis de temperatura:** Para realizar esta acción en la aplicación, se necesitan dos imágenes: La “inicial” o estado “normal” de funcionamiento y la “prueba” o estado “modificado” de funcionamiento. Estas dos se pueden adquirir según el manual de uso del programa en el anexo B.

El programa, básicamente, compara el histograma de cada imagen detectando cambios drásticos en el valor (0 – 255) de sus píxeles.

Lee las dos imágenes, las convierte al plano ROJO, guarda los histogramas, aplica un valor absoluto a la resta de estos dos y si el valor supera un cambio específico en el número de píxeles; mostrará un mensaje con una alerta definida.

**Figura 51. Opciones tratamiento de imágenes**



**Fuente: Autores Proyecto**

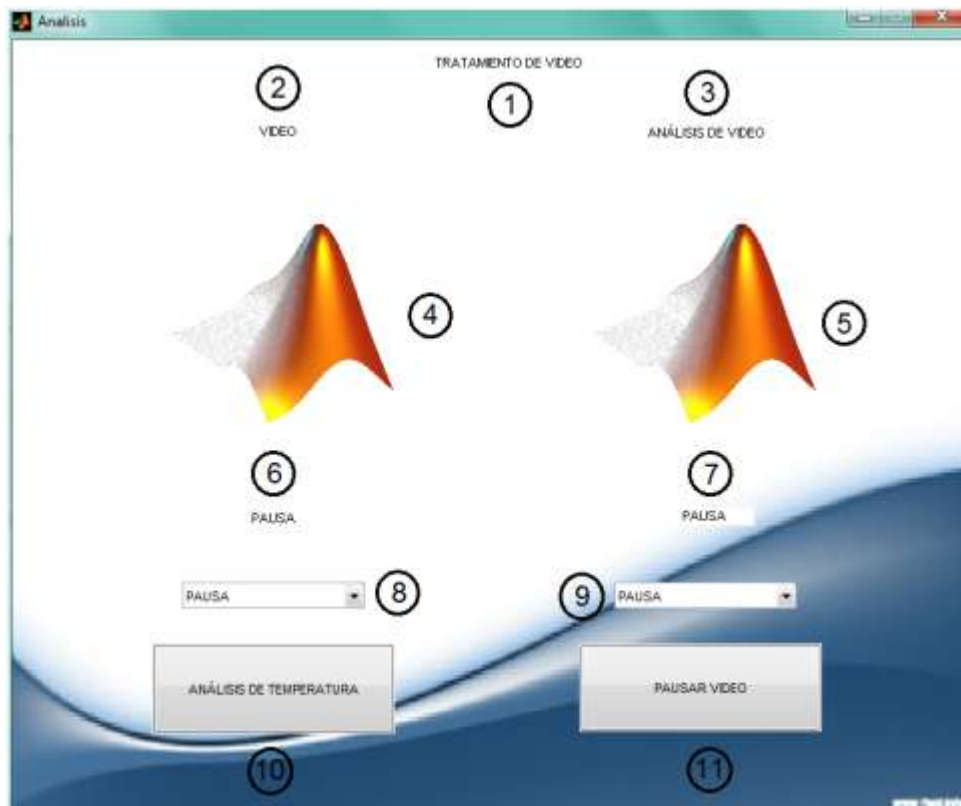
## 3.2 TRATAMIENTO DE VIDEO

Esta aplicación permite analizar videos según el toolbox de tratamiento de matlab con sus respectivas funciones mostradas en el capítulo 1. Cada módulo, brinda una utilidad y resultados específicos para este programa (ver tabla 9). El código pertinente se encuentra en el anexo E.

### 3.2.1 GUIDE

La interfaz que se aplica contiene cinco (5) cuadros de texto estático, dos (2) popup menú, dos (2) axes y dos (2) pushbutton, a los cuales se les ha asignado un número en la siguiente figura para ser explicados.

**Figura 52. GUIDE tratamiento de video**



**Fuente: Autores Proyecto**

1. **Texto estático 1:** Título de la aplicación.
2. **Texto estático 2:** Nombre del axes1 (4).
3. **Texto estático 3:** Nombre del axes2 (5).
4. **Axes 1:** Espacio que muestra el video, su plano rojo y su histograma.
5. **Axes 2:** Espacio que muestra el video y su respectivo tratamiento.
6. **Texto estático 4:** Nombre de la opción escogida de la lista de video (8).
7. **Texto estático 5:** Nombre de la opción escogida de la lista de tratamiento de video (9).
8. **Popup menú 1:** Lista de opciones de video (video normal, rojos e histograma).
9. **Popup menú 2:** Lista de opciones de tratamiento de video (tamaño, forma y puntos calientes).
10. **Pushbutton 1:** Botón accionador del análisis de temperatura y su respectiva alerta.
11. **Pushbutton 2:** Botón pausa de video.

### 3.2.2 Funcionamiento

Se divide básicamente en 9 acciones (ver figura 53); el video prueba del que se hablará se puede conseguir según el manual de uso de la cámara en el anexo.....

1. **Pausa:** Es la acción más sencilla, lee la imagen de pausa guardada mediante el comando `Imread` y la muestra en el axes deseado por medio del comando `imshow`.
2. **Video:** Lee y guarda el video mediante `mmreader`; obtiene el número de frames, alto y ancho; pre asigna la estructura, crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; la muestra en el axes deseado con `imshow` y pausa durante un breve periodo de tiempo para que sea apreciada.

**3. Rojos:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura, crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; obtiene su plano ROJO; la muestra en el axes deseado con imshow y pausa durante un breve periodo de tiempo para que sea apreciado.

**4. Histograma:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura, crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; obtiene su plano ROJO; muestra el histograma en el axes deseado con Imhist y pausa durante un breve periodo de tiempo para que sea apreciado.

**5. Forma:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura; crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; aumenta su brillo para captar mejor los pixeles; la convierte al plano ROJO (color más sobresaliente en una imagen térmica); se extraen los bordes mediante el logaritmo de canny; dilata estos bordes por medio de una rejilla cuadrada de 4 pixeles de espesor; rellena los espacios entre puntos para crear, aproximadamente, el perfil de la figura; elimina las áreas menores consideradas como ruido; etiqueta cada una de las figuras resultantes y guarda sus propiedades específicas como área y boundingbox; grafica un rectángulo (boundingbox) sobre cada figura; crea una relación del área de la figura resultante del proceso con el área abarcada por este último (ver figura 49); según esta relación, se define que tipo de figura geométrica es; muestra en el axes deseado mediante imshow y pausa durante un breve periodo de tiempo para que sea apreciado.

**6. Puntos calientes:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura; crea un bucle que lee frame por

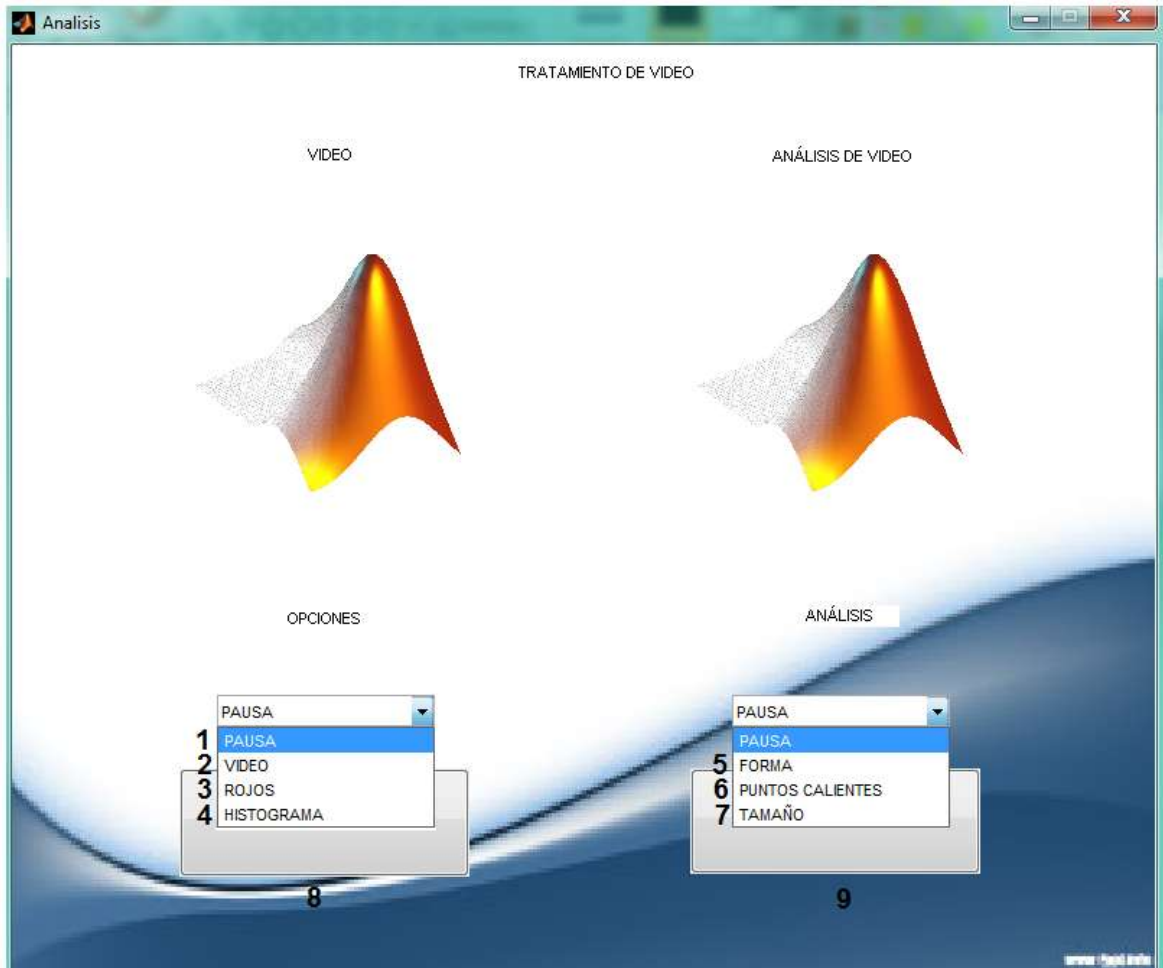
frame y lo guarda en una nueva variable; la convierte en una imagen; aumenta el brillo; convierte al plano ROJO; guarda el umbral y con este la binariza; elimina áreas pequeñas consideradas como ruido; obtiene el perímetro de las figuras resultantes; crea un bucle que recorre todas estas; guarda los perímetros en un vector y los dibuja en las figuras; muestra en el axes deseado mediante imshow y pausa durante un breve periodo de tiempo para que sea apreciado.

**7. Tamaño:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura; crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; aumenta el brillo; convierte al plano ROJO; guarda el umbral y con este la binariza; elimina áreas pequeñas consideradas como ruido; etiqueta cada una de las figuras resultantes y guarda sus propiedades principales como área y boundingbox; dibuja un rectángulo (boundingbox) sobre cada figura; relaciona las áreas resultantes con la mayor y la menor mediante los comandos max y min respectivamente; según los resultados anteriores, nombra las figuras; muestra en el axes deseado con imshow y pausa durante un breve periodo de tiempo para que sea apreciado.

**8. Análisis de Temperatura:** Lee el video mediante mmreader; obtiene el número de frames, alto y ancho; pre asigna la estructura, crea un bucle que lee frame por frame y lo guarda en una nueva variable; la convierte en una imagen; obtiene su plano ROJO; guarda el histograma del primer frame (estado inicial) y lo compara con los histogramas de los siguientes; si el valor absoluto de la resta de estos dos supera un cambio en el número de pixeles definidos, se ejecutará una alerta de este cambio y terminará el análisis, si no ocurre, el análisis continuará.

**9. Pausa:** Detiene el video durante un breve periodo de tiempo.

Figura 53. Opciones tratamiento de video



Fuente: Autores Proyecto

#### 4. PRUEBAS Y MEDICIONES

Las pruebas y mediciones realizadas al banco de trabajo para pruebas termográficas fueron hechas en el laboratorio de vibraciones mecánicas con la cámara termográfica Fluke TI125 (ver especificaciones anexo.....) de la siguiente manera:

Se revisó el buen funcionamiento del banco de trabajo accionando cada uno de los interruptores para dar energía a los módulos, se revisa si la temperatura aumenta en cada resistencia y se da el visto bueno. Para el motor, se revisa que no haya interrupciones al inicio de su encendido o anomalías en su trabajo. Por último, se acciona la parada de emergencia para corroborar su acción.

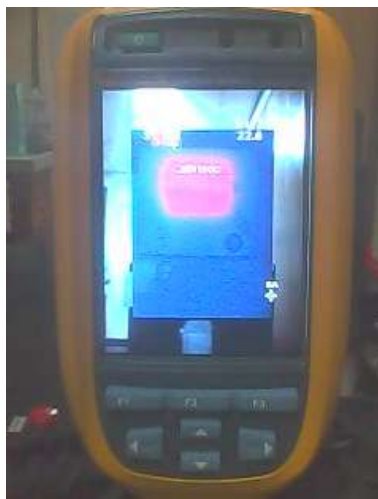
**Tabla 11. Verificación Banco de trabajo**

<b>INTERRUPTOR</b>	<b>FUNCIONAMIENTO</b>	<b>ANORMALIDAD</b>
Encendido	Ok	Ninguna
A	Ok	Ninguna
B	Ok	Ninguna
C	Ok	Ninguna
D	Ok	Ninguna
Motor	Ok	Ninguna
Emergencia	Ok	Ninguna
Piloto Encendido	Ok	Ninguna
Piloto Motor	Ok	Ninguna

**Fuente: Autores / Prueba**

Después de revisar el banco de trabajo se prosiguió a verificar el buen estado de la cámara, observando que cada una de sus funciones y diferentes botones de uso estuvieran bien. La anterior tabla fué hecha para verificar el funcionamiento.

**Figura 54. Revisión cámara termográfica**



**Fuente: Autores / Prueba**

Se establecieron los diferentes parámetros de la cámara como el rango, emisividad, marcadores, temperatura de fondo, etc.; los cuales fueron definidos en diferentes pruebas específicas para cada módulo y así generar óptimos resultados en la toma de datos. Para una mejor información ver el anexo.....

**Figura 55. Prueba para definición de parámetros de la cámara**



**Fuente: Autores / Prueba**

**Tabla 12. Parámetros de la cámara para módulos**

Parámetro \ Módulo	Forma	Tamaño	Profundidad	Posición	Aletas	Rodamientos
Rango	Automático					
Temperatura fondo	Temperatura ambiente					
Emisividad	0,55	0,55	0,90	0,95	0,95	0,95
Paleta de color	Arco de Hierro					
Marcadores	Todos					
Formato	Avi, jpg, is2, is3					

**Fuente: Autores / Prueba**

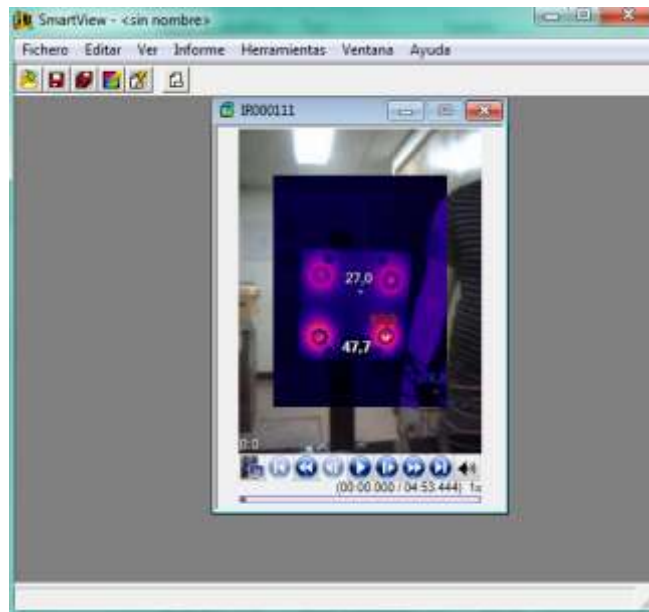
Habiendo verificado ya dos de los elementos esenciales del proyecto, se pudo empezar a tomar los diferentes datos de las resistencias y de los módulos.

Cada una de las resistencias manejan diferentes propiedades dependiendo de su aplicación, entre ellas, el aumento de temperatura en el tiempo, donde influyen características como tamaño, forma y espesor que definen un comportamiento único. La temperatura fué tomada en el extremo más lejano del centro de la resistencia y se utilizó el software smartview 3.2 (ver anexo.....), el cual viene con la cámara.

La cámara TI125 puede generar dos tipos de formato de imágenes o de video, uno es el “.avi” o “.jpg” que puede ser visto en cualquier visor de imágenes o reproductor de video, y el otro es el “.is2/3” que sólo puede ser leído por el smartview. Este último permite revisar cada pixel y el valor de la temperatura en el (en grados Fahrenheit) para generar gráficas de comportamiento en el tiempo.

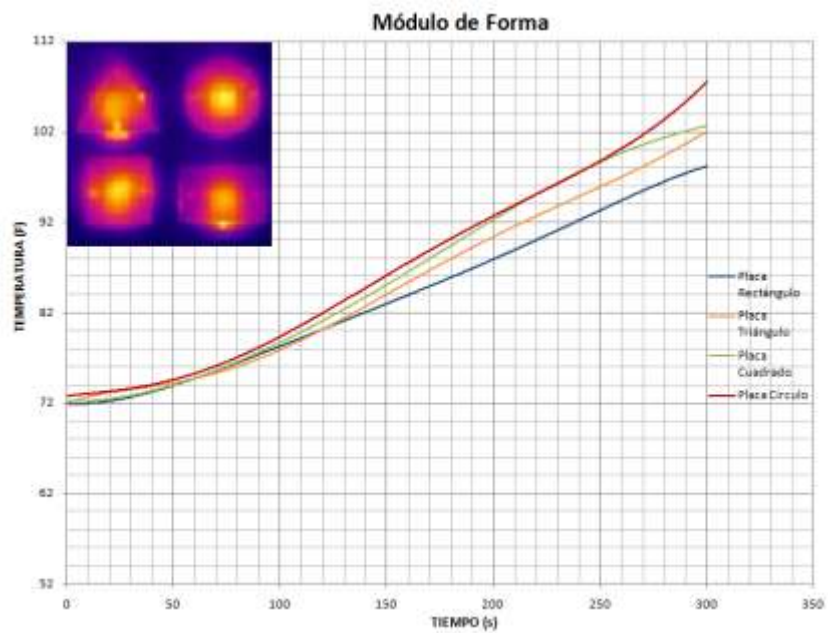
Las siguientes gráficas se obtuvieron tomando datos de temperatura en el pixel más lejano del centro de la resistencia cada 10 segundos en un video de 5 minutos.

Figura 56. Toma de temperatura mediante puntero



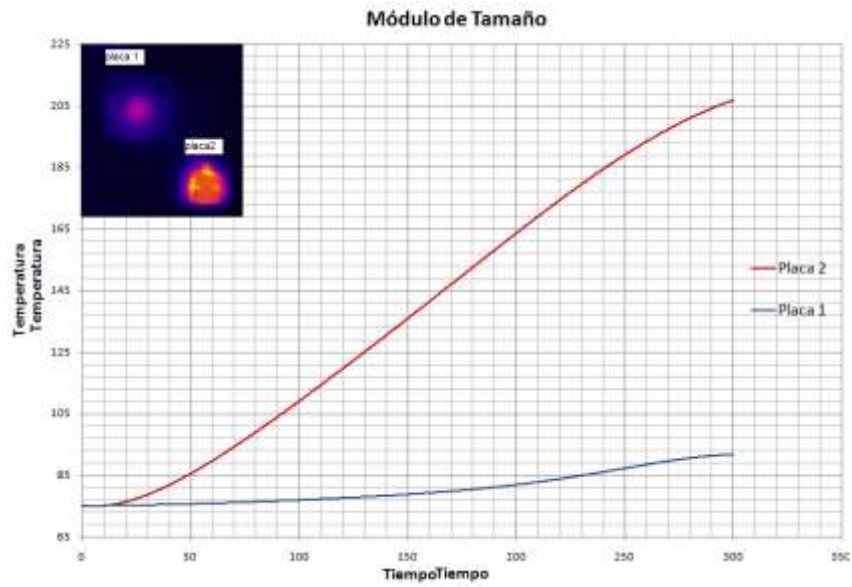
Fuente: Autores / Medición

Figura 57. Curvas de temperatura módulo de forma



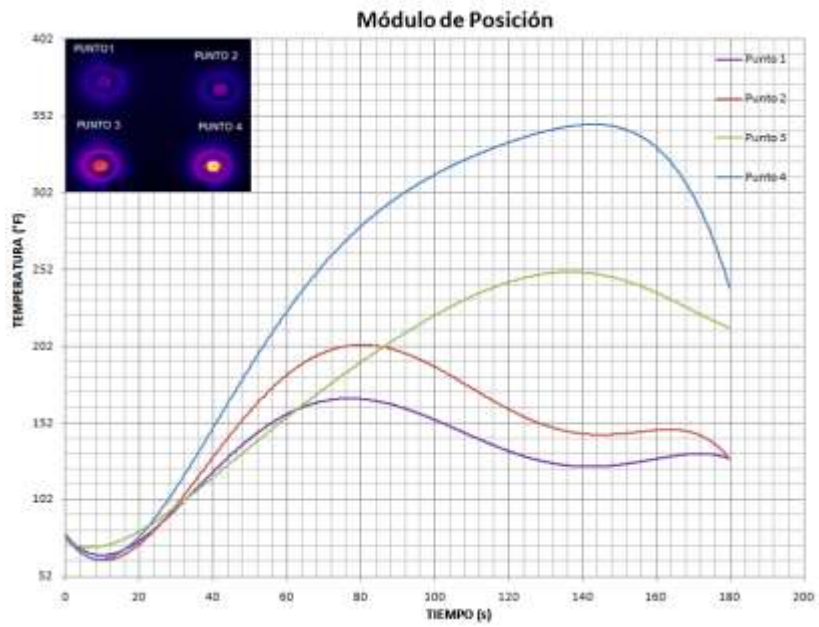
Fuente: Autores / Medición

Figura 58. Curvas de temperatura módulo de tamaño



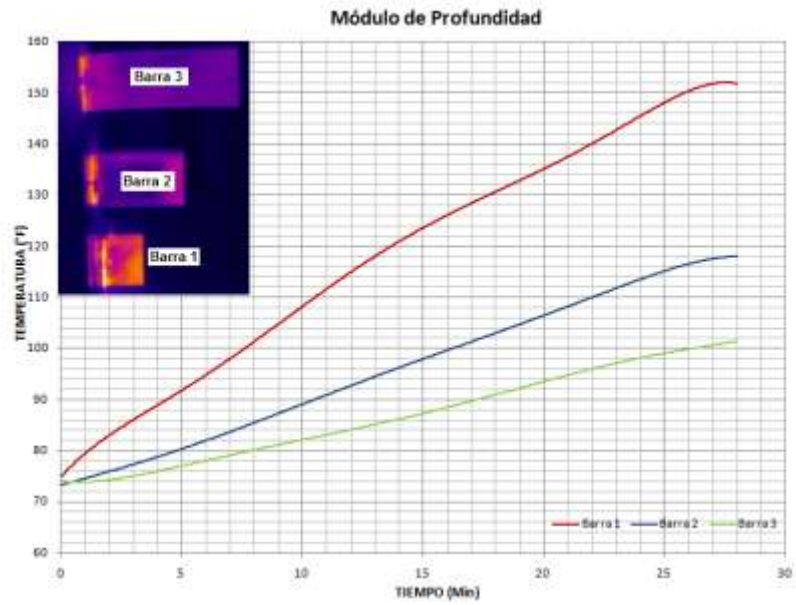
Fuente: Autores / Medición

Figura 59. Curvas de temperatura módulo posición



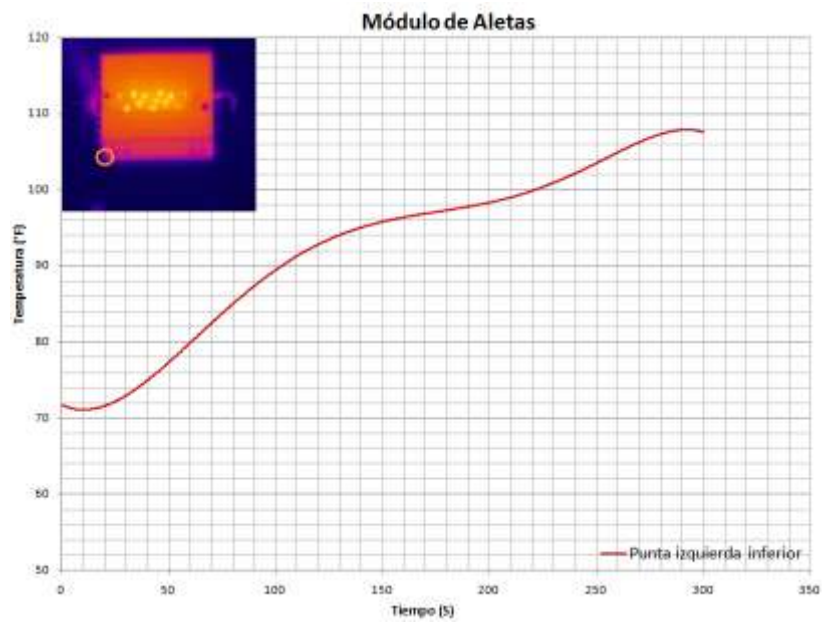
Fuente: Autores / Medición

Figura 60. Curvas de temperatura módulo profundidad



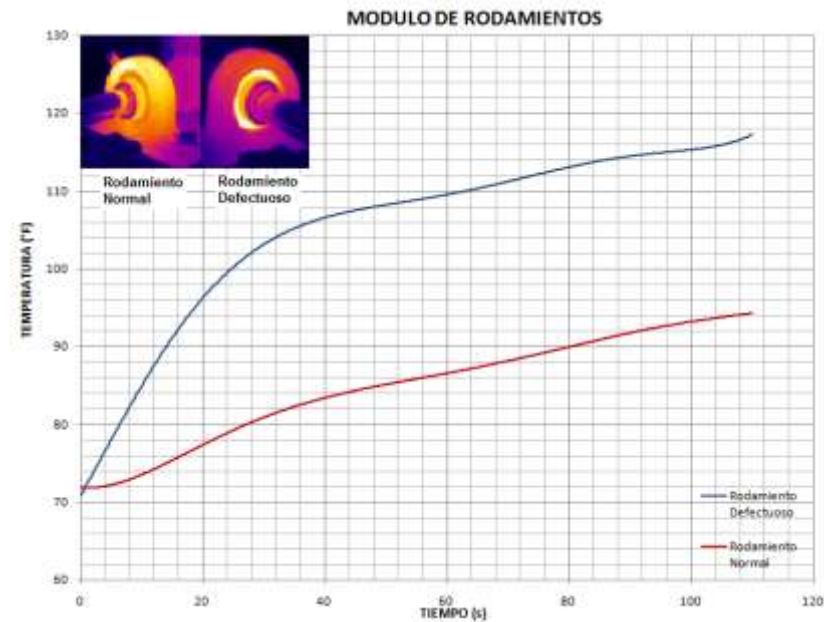
Fuente: Autores / Medición

Figura 61. Curva de temperatura módulo de Aletas



Fuente: Autores / Medición

**Figura 62. Curvas de temperatura módulo de rodamientos**



**Fuente: Autores / Medición**

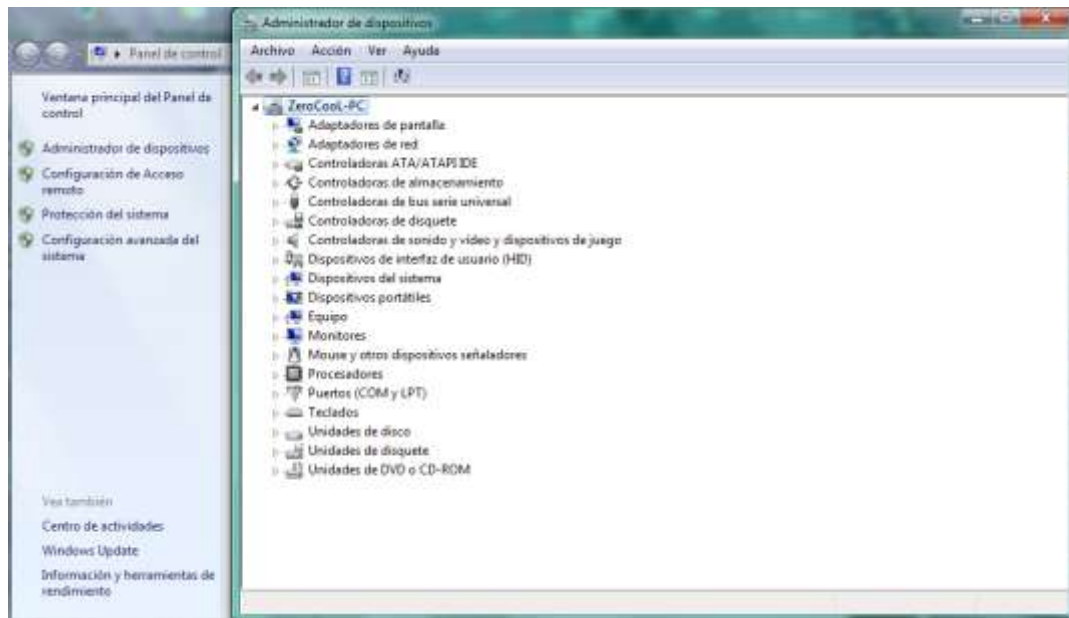
Cada una de estas gráficas muestra el comportamiento de la temperatura de uno de los extremos más alejados de las resistencias (colocadas en el centro de la placa) para verificar su respectiva transferencia de calor hacia este punto.

También esta información sirve para mantener una relación entre el cambio de temperatura y el tiempo para el código de matlab.

Para la captación y tratamiento de videos e imágenes se conectó la cámara termográfica al computador para su respectivo reconocimiento.

MATLAB reconoce dispositivos de imágenes en puertos del computador como lo son las webcam o los celulares, pero al tratar de reconocer la cámara termográfica esta no es uno de ellos sino un dispositivo "térmico de imágenes" en un puerto totalmente diferente y no se reconoció (ver figura 63). También se trató de convertir a puerto COM (serial) y puerto LP (impresora) pero tampoco fue reconocido.

**Figura 63. Administrador de dispositivos**



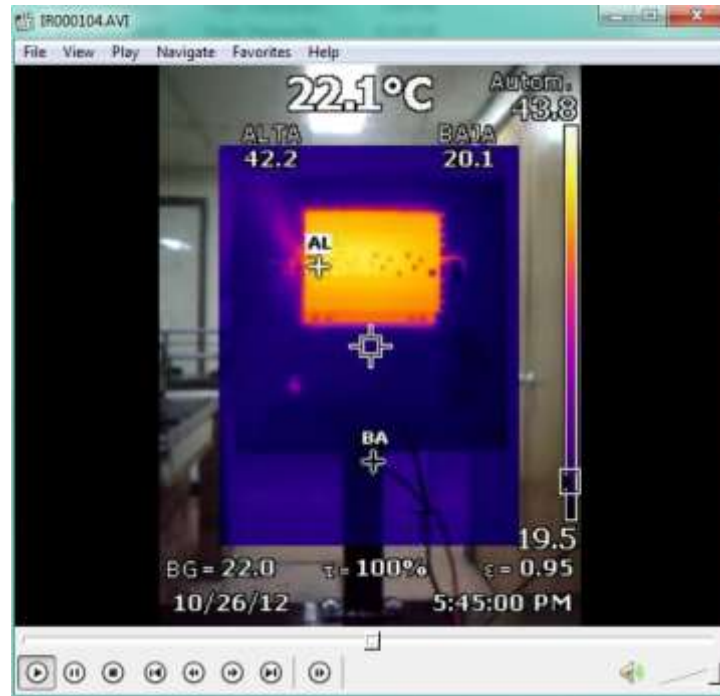
**Fuente: Autores Proyecto**

Para el análisis en tiempo real, el software smartview nos permite ver el video que se está grabando aunque no permite su respectivo tratamiento, así que primero se tuvieron que grabar para luego ser tratados.

Las imágenes si se pueden grabar del smartview y tratarlas directamente (ver anexo B).

Para la implementación del código se grabaron los diferentes videos necesarios de cada uno de los módulos en funcionamiento en formato “.avi” y se tomaron las imágenes necesarias en formato “.jpg” para luego ser introducidos al código y hacer el respectivo análisis.

**Figura 64. Video almacenado**



**Fuente: Autores / Medición**

**Figura 65. Imagen almacenada**



**Fuente: Autores / Medición**

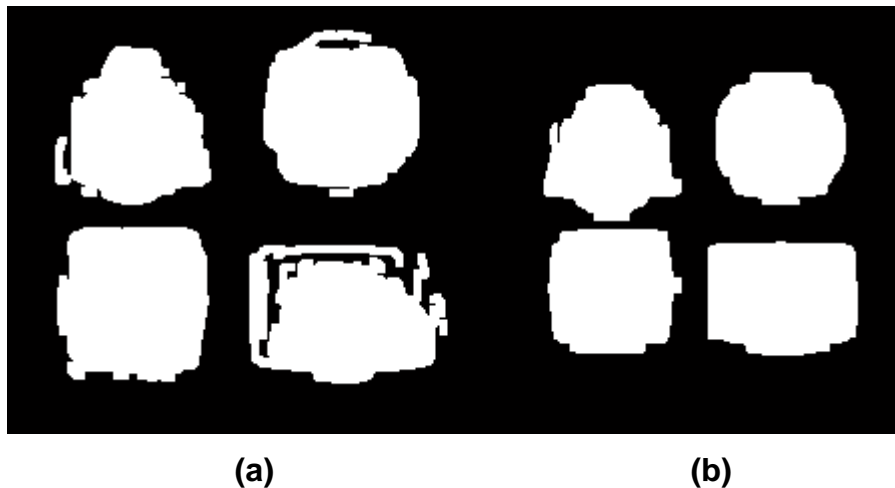
## 5. RESULTADOS

### IDENTIFICACIÓN DE FORMA

El módulo de forma se utilizó para el estudio de figuras geométricas mediante la aplicación en MATLAB. Se tomaron videos de 5 minutos e imágenes en formatos “.avi” y “.jpg”, respectivamente, con la cámara termográfica para su análisis e interpretación de resultados.

Para la correcta identificación de las figuras, las resistencias térmicas deben distribuir el calor por la mayor cantidad de área de las placas, así se dará una binarización correcta (ver figura 66). El programa trabaja con el plano rojo de las imágenes así que entre más intenso sea el color de la temperatura, mejor se podrán extraer los bordes de la figura resultante. Para ello, se debe dar un tiempo para que las resistencias transmitan calor en el mayor espacio posible.

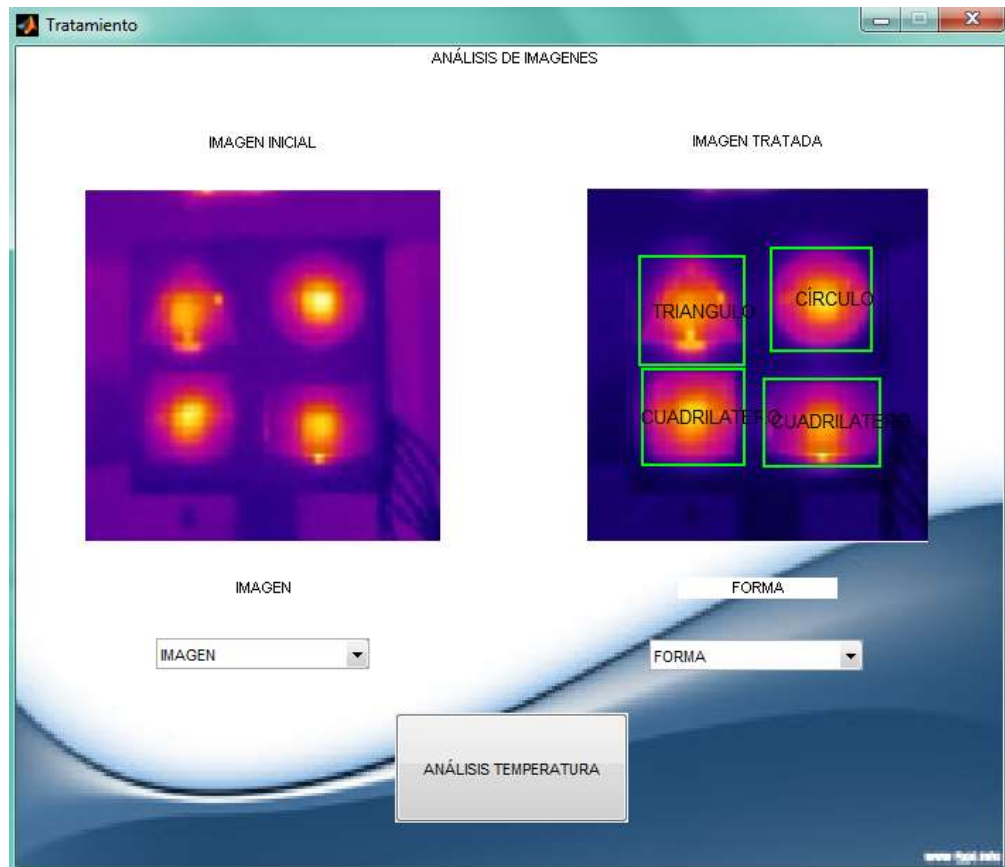
**Figura 66. Binarización a) Temperatura no uniforme, b) Temperatura Uniforme**



**Fuente: Autores Proyecto**

Imagen térmica del módulo de forma con tratamiento.

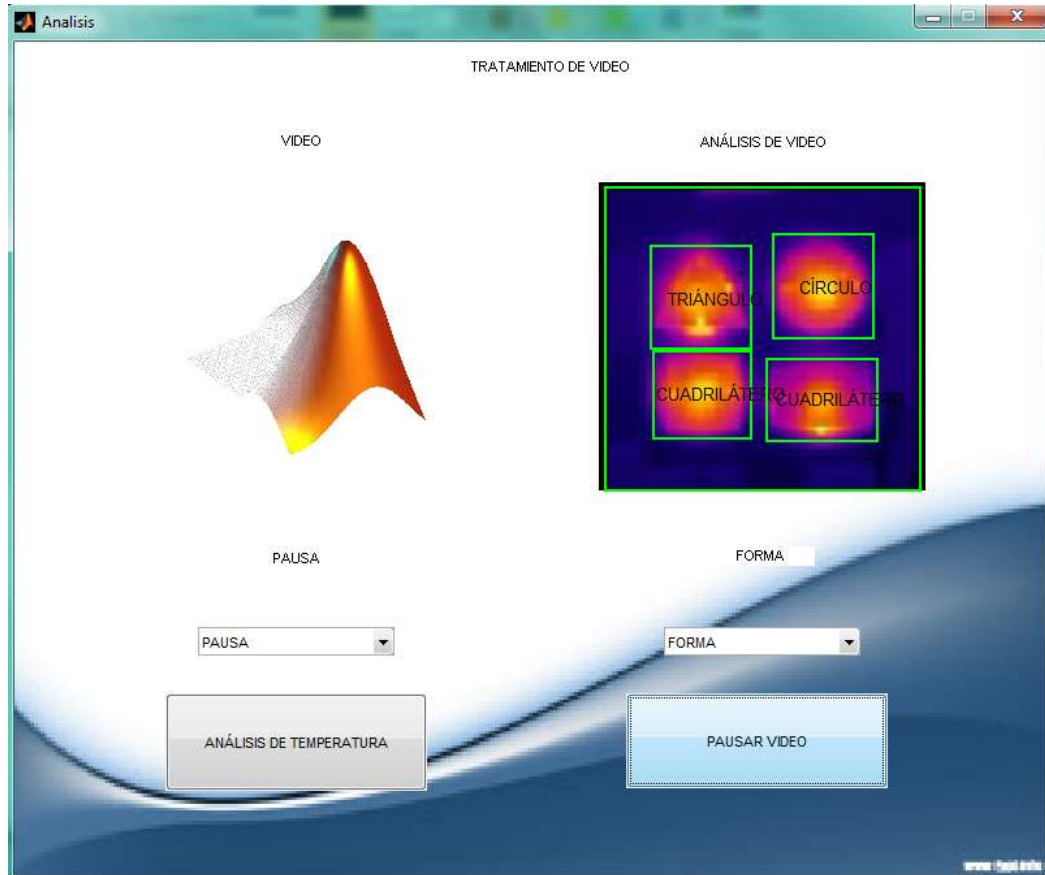
**Figura 67. Identificación de figuras geométricas en Imágenes**



**Fuente: Autores / Resultados**

Video térmico del módulo de forma con tratamiento.

**Figura 68. Identificación de figuras geométricas en Video**



**Fuente: Autores / Resultados**

## IDENTIFICACIÓN DE PUNTOS CALIENTES

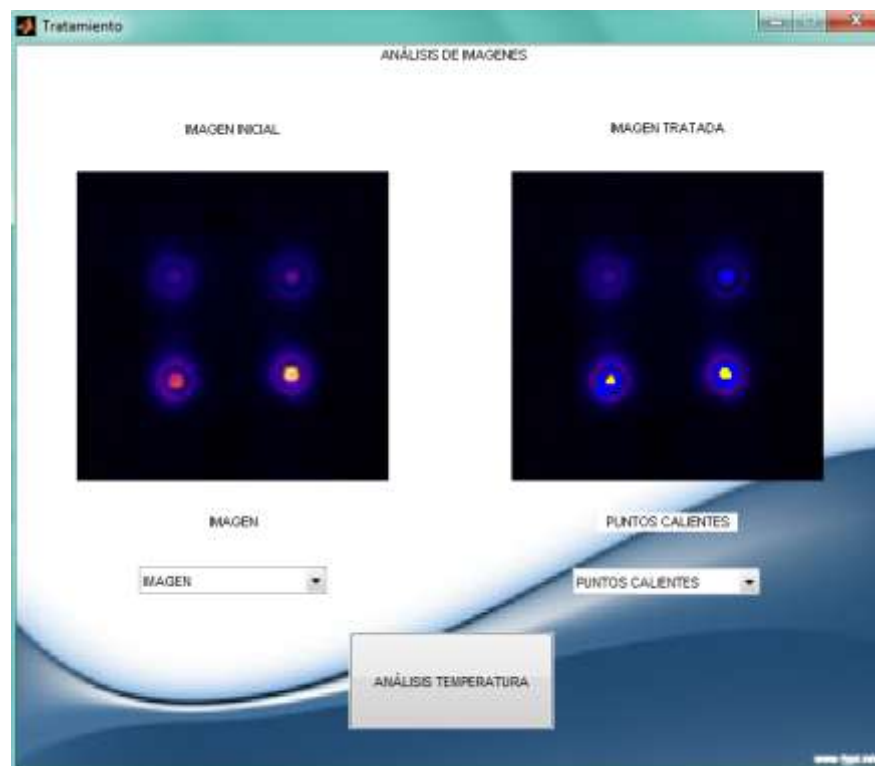
El módulo de posición se utilizó para el estudio de puntos calientes mediante la aplicación en MATLAB. Se tomaron videos de 5 minutos e imágenes en formatos “.avi” y “.jpg”, respectivamente, con la cámara termográfica para su análisis e interpretación de resultados.

La identificación de puntos en imágenes es diferente a la de video por la demora en el muestreo de datos, mientras en la primera se da un color específico para cierto valor del pixel, en la segunda resaltamos el perímetro. Esto se debe al tiempo en que el bucle recorre cada pixel y asigna el color, si es en una sola imagen este tiempo es insignificante comparado al que se necesitaría para hacerlo en un video que contiene muchas más de estas; en el mismo instante en que el programa debería resaltar los colores de los pixeles en un frame, ya se estaría cambiando a uno nuevo.

El efecto de resaltar el perímetro hace la misma función de colorear el pixel, ya que el área dentro de estos será la misma.

Imagen térmica del módulo de posición con tratamiento

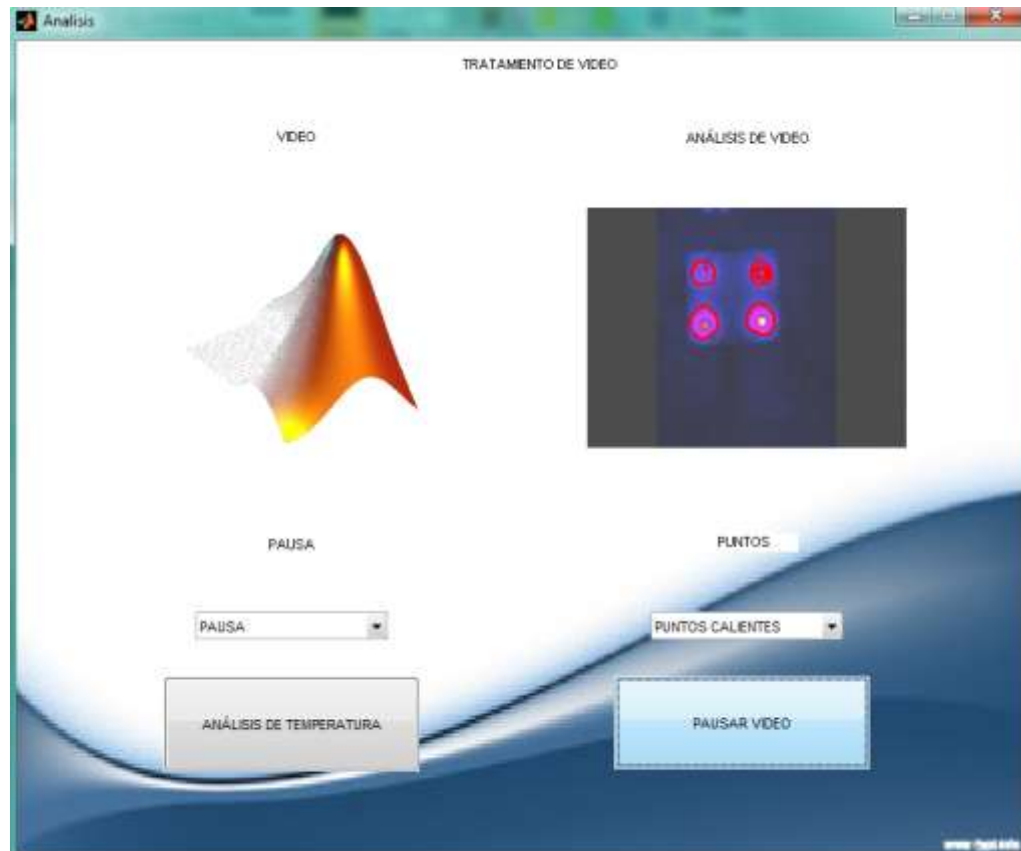
**Figura 69. Identificación de puntos calientes en imágenes**



**Fuente: Autores / Resultados**

Video térmico del módulo de posición con tratamiento

**Figura 70. Identificación de puntos calientes en videos**



**Fuente: Autores / Resultados**

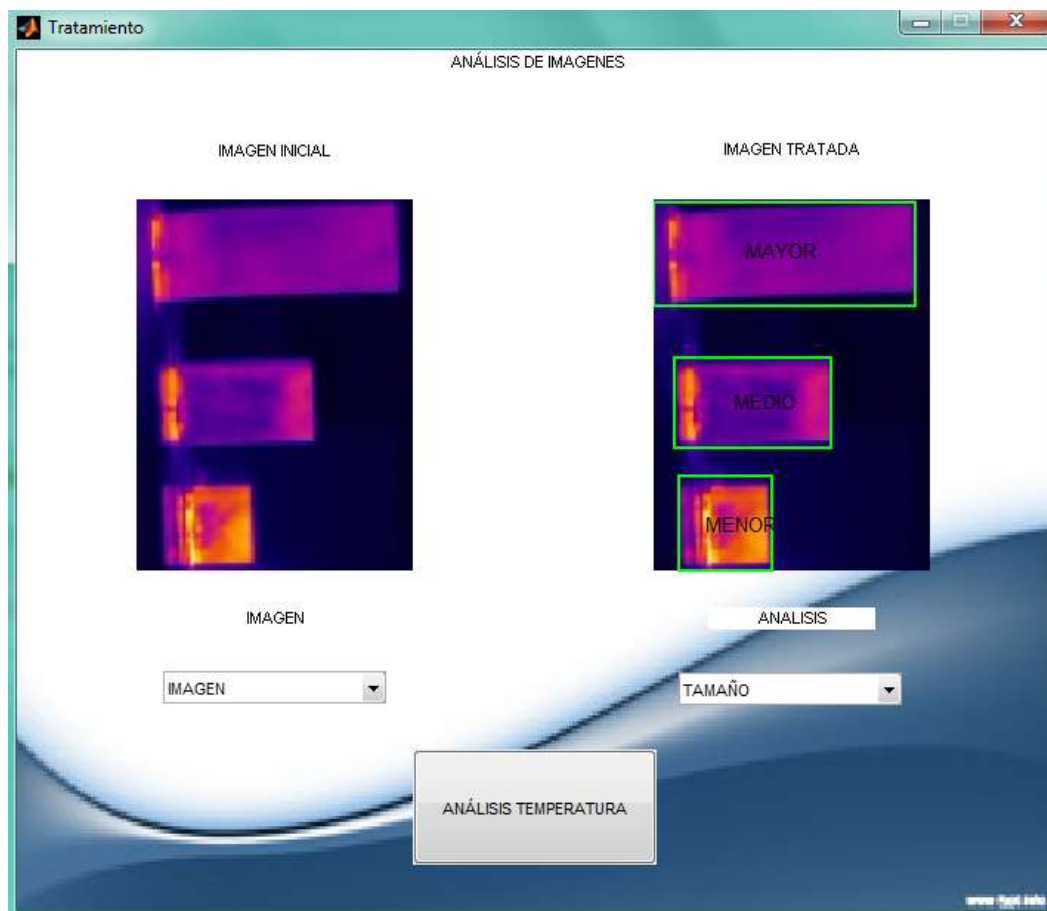
## **IDENTIFICACIÓN DE TAMAÑO**

Los módulos de profundidad y tamaño se utilizaron para el estudio de área mediante la aplicación en MATLAB. Se tomaron videos de 5 minutos e imágenes en formatos “.avi” y “.jpg”, respectivamente, con la cámara termográfica para su análisis e interpretación de resultados.

El área que identifica el programa está relacionada con la temperatura de la placa, para un mejor tratamiento, esta última debe ser uniforme y no concentrada en puntos exactos (ver figura 66) para generar una binarización aproximada. Debe darse un cierto tiempo para que las resistencias transmitan el calor en el mayor espacio posible.

Imagen térmica del módulo de profundidad con tratamiento

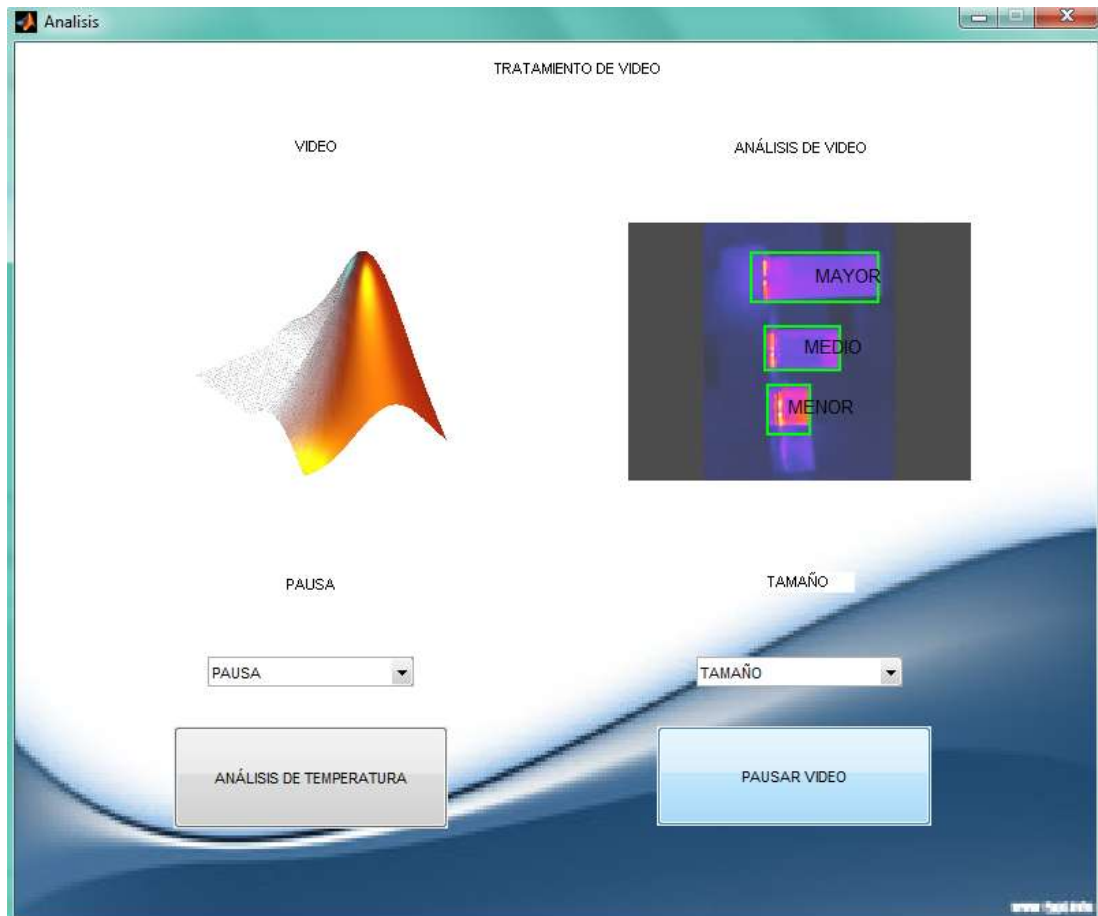
**Figura 71. Identificación de tamaño en imágenes**



**Fuente: Autores / Resultados**

Video térmico del módulo de profundidad con tratamiento

**Figura 72. Identificación de tamaño en videos**



**Fuente: Autores / Resultados**

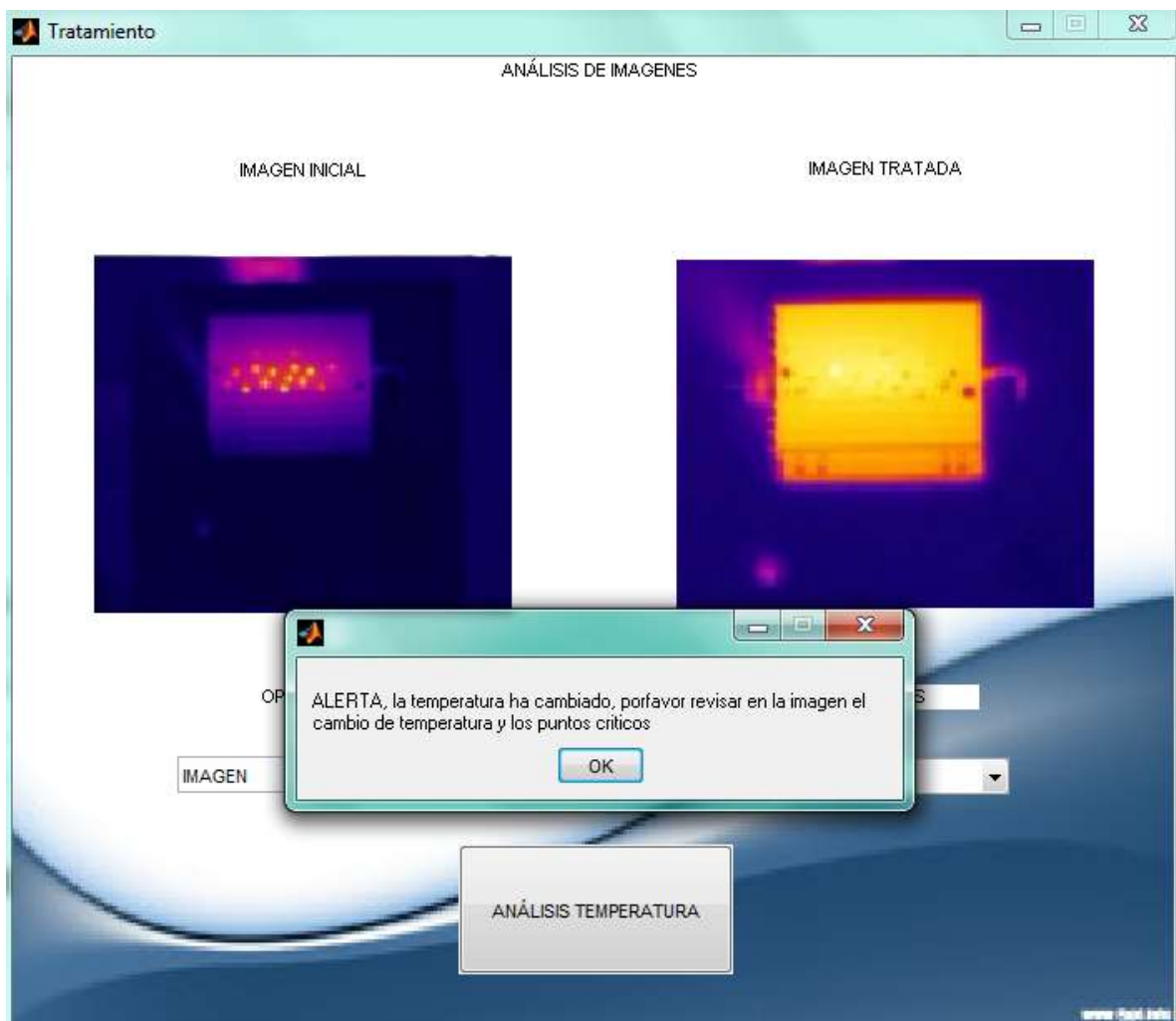
## **ANÁLISIS DE TEMPERATURA**

Todos los módulos se usaron para el estudio de temperatura mediante la aplicación en MATLAB. Se tomaron videos de 5 minutos e imágenes en formatos “.avi” y “.jpg”, respectivamente, con la cámara termográfica para su análisis e interpretación de resultados.

El único requisito que debe haber para la ejecución de esta función es un cambio en los rojos de una imagen, así por medio de su histograma se puede diferenciar y aplicar la respectiva alerta.

Imagen térmica del módulo de aletas con tratamiento

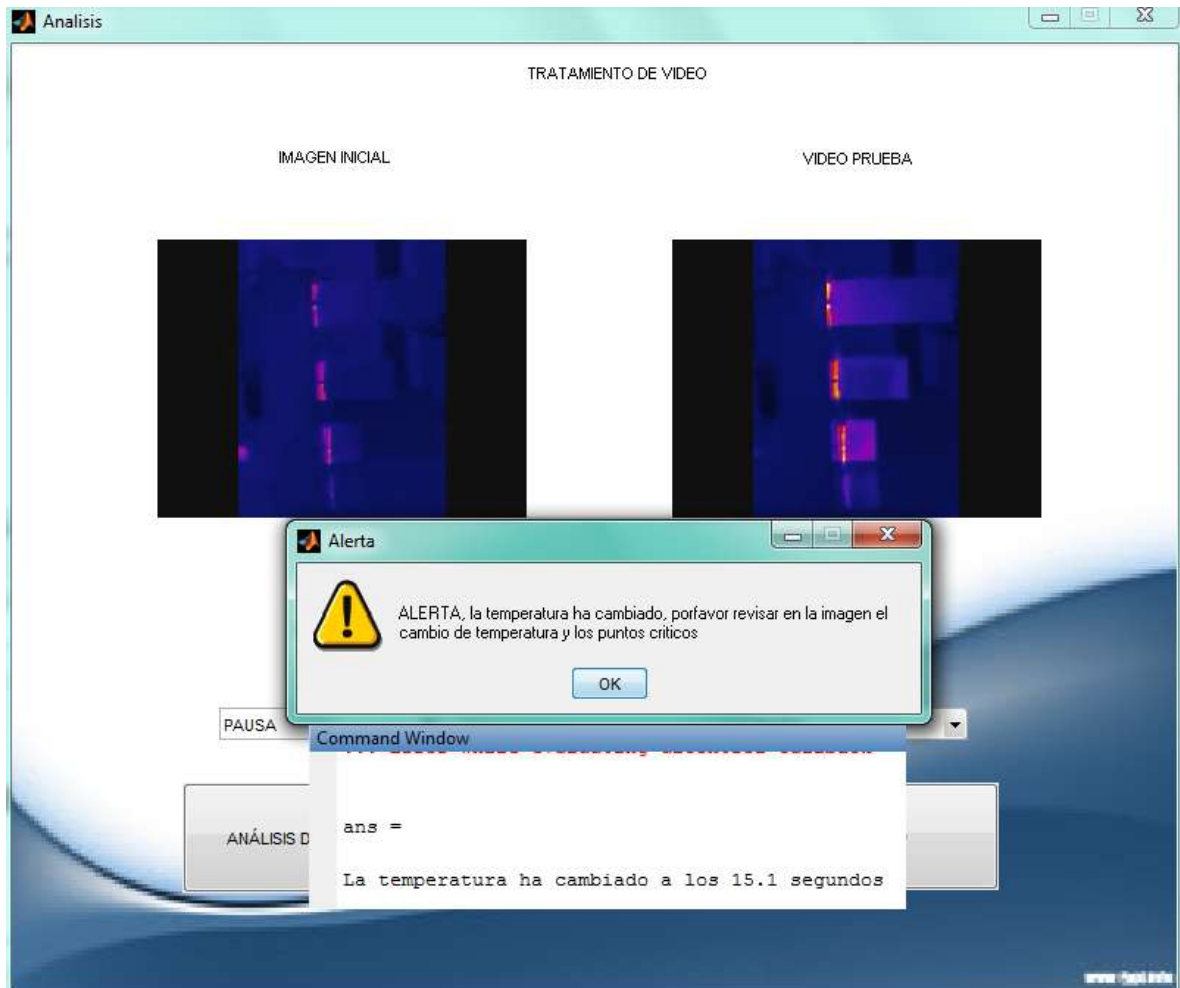
**Figura 73. Análisis de temperatura en imágenes**



**Fuente: Autores / Resultados**

Video térmico del módulo de profundidad con tratamiento

**Figura 74. Análisis de temperatura en video**



**Fuente: Autores / Resultados**

## **PROFUNDIDAD Y TAMAÑO**

Estas dos propiedades no alteran de ninguna forma la toma de datos debido a que en cualquier parte de la imagen o del video que se encuentre un punto caliente; este lejos, cerca, más grande o más pequeño; la temperatura será la verdadera y no se alterarán los resultados (ver figuras 69 – 72).

## 6. CONCLUSIONES Y RECOMENDACIONES

- Se lograron cumplir los objetivos propuestos inicialmente, los cuales consistían en el desarrollo de una aplicación en el entorno MATLAB para monitoreo de fallas en maquinaria mediante termografía infrarroja.
- El análisis consiste en identificar 'fallas térmicas simuladas', mediante cambios de temperatura abruptos.
- El banco construido permite el uso de: diferentes módulos térmicos con propiedades físicas únicas y un motor eléctrico con sus respectivos rodamientos.
- Los módulos térmicos trabajan propiedades físicas como tamaño, forma, profundidad, posición, temperatura y disipación de calor.
- La obtención de imágenes y videos termográficos se hace mediante la cámara termográfica y el banco de trabajo. Además, la experiencia se realizó con base en el tratamiento morfológico de estos.
- En la experiencia se induce calentamiento a las resistencias para hacer un análisis del cambio de temperatura en relación con el tiempo.
- Se logra visualizar en los resultados el comportamiento del campo de infrarrojos en la obtención de datos morfológicos y de temperatura.
- Los módulos y el banco de trabajo, fueron diseñados para soportar temperaturas menores a 250°C.
- Tanto el funcionamiento del banco y la cámara como su mantenimiento fueron recopilados en manuales específicos para su fácil aprendizaje.
- Resulta necesaria la adquisición de una cámara termográfica que se conecte directamente como dispositivo de imágenes al computador, para generar tratamiento en tiempo real sin necesidad de un software.

## BIBLIOGRAFÍA

- BENITEZ, Hernán y LOAIZA, Humberto. Termografía activa pulsada en inspección de materiales: Técnicas avanzadas de Procesado. Colombia, Cali: 2011.
- COMUNIDAD de Madrid. Guía de la termografía infrarroja. España, Madrid: Gráficas Arias Montano, 2011. ISBN: 36-312-2011.
- FLUKE EDUCACIÓN Y FORMACIÓN. [en línea]. <<http://www.fluke.com/fluke/coes/formación/>> [citado en 10 de noviembre de 2012].
- GARCIA, Santiago. La Contratación del Mantenimiento Industrial. Volumen 3. España, Madrid: Edígrafos, 2009. ISBN: 978-84-7978-962-6.
- MANUAL de Uso de la Cámara termográfica Fluke TI125, 1ra Ed. Fluke, Colombia, Bogotá: 2011.
- TERMOGRAFÍA FLIR. [en línea]. <<http://www.flir.com/thermography/americas/>> [citado en 10 de noviembre de 2012].

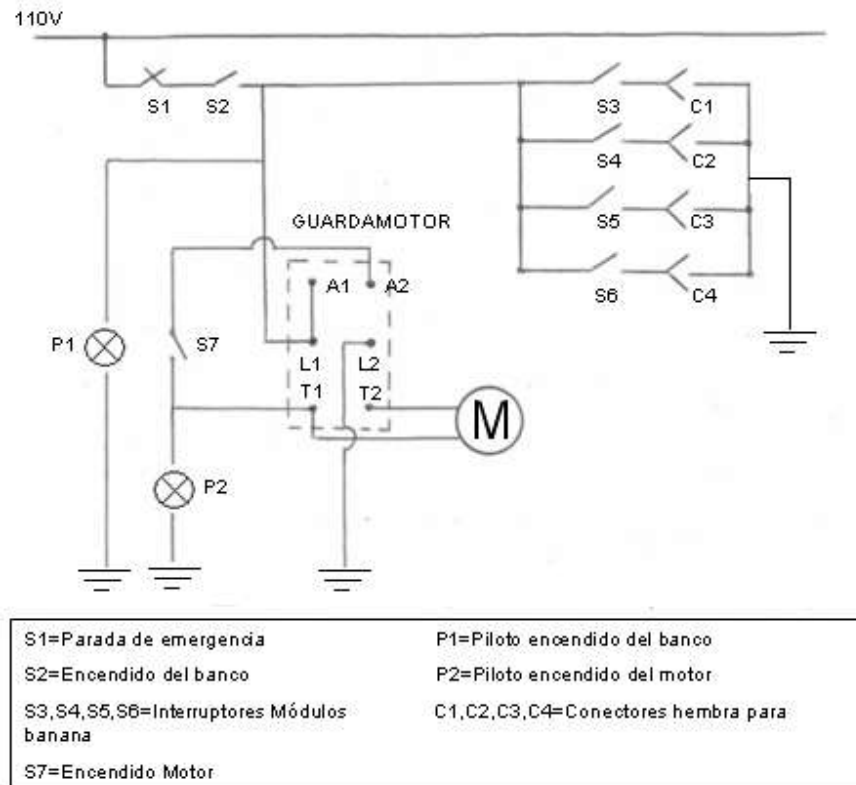
**ANEXO A**  
**MANUAL DE USO BANCO DE TRABAJO DE TERMOGRAFIA INFRARROJA**  
**TABLERO DE CONTROL**



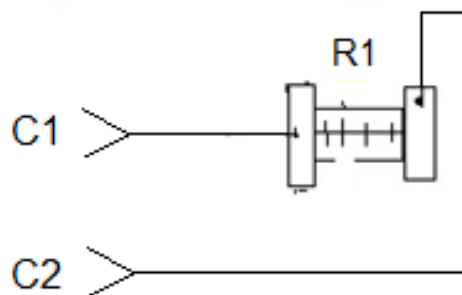
**El tablero de control del banco dispone de:**

- a.** Interruptor de la resistencia A
- b.** Interruptor de la resistencia B
- c.** Interruptor de la resistencia C
- d.** Interruptor de la resistencia D
- e.** Interruptor de encendido del motor eléctrico.
- f.** Interruptor de encendido del banco.
- g.** Interruptor de parada de emergencia.
- 1.** Piloto encendido del motor eléctrico.
- 2.** Piloto encendido del banco.

Estas son las conexiones eléctricas pertinentes del banco, en caso que se desee su modificación o arreglo.

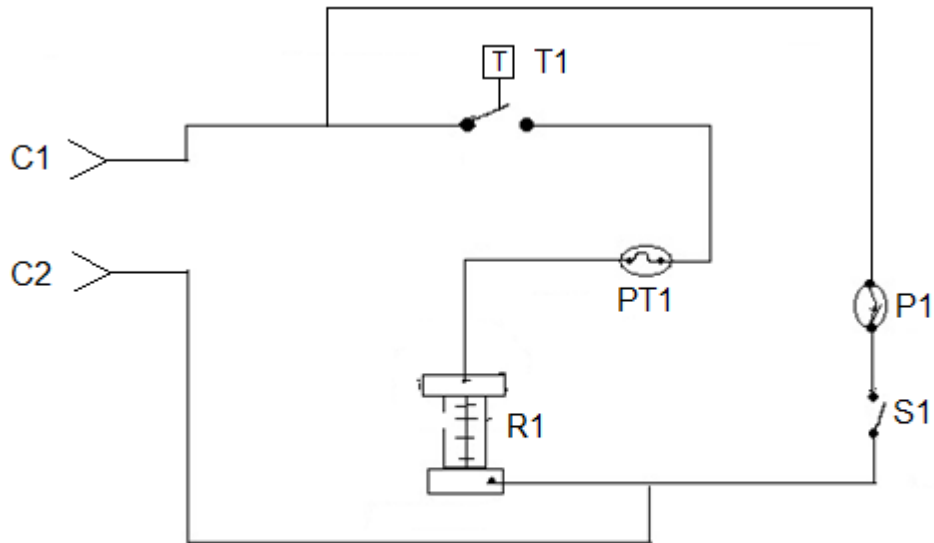


Las resistencias de los módulos (exceptuando el de aletas) obedecen el siguiente esquema eléctrico.



- C1** = Conector Hembra Positivo
- C2** = Conector Hembra Negativo
- R1** = Resistencia Térmica

El diagrama eléctrico del módulo de aletas es el siguiente.



**C1** = Conector Hembra Positivo  
**C2** = Conector Hembra Negativo  
**T1** = Termostato  
**PT1** = Protector Térmico

**R1** = Resistencia Térmica  
**P1** = Piloto Interruptor de Encendido  
**S1** = Interruptor de Encendido

### Encendido del banco

Para encender el banco se dispone de un interruptor **(f)**, al ser accionado se encenderá el piloto **(2)** el cual indica que el sistema está energizado.

### Encendido del motor eléctrico

Una vez energizado el banco se puede encender el módulo de calentamiento de rodamientos utilizando el interruptor **(e)**, este módulo puede permanecer accionado por un tiempo no mayor a 5 minutos ya que se va a observar el funcionamiento de un rodamiento en condiciones de operación no aceptables; se provocará un sobrecalentamiento en la chumacera; esta no está lubricada y tiene oxidación con sales permitiendo generar un calentamiento muy alto en poco tiempo.

Al momento de accionar el interruptor **(e)** se encenderá el **piloto 1**, el cual indicará energía del motor eléctrico.

### **Accionamiento de los interruptores de energizado de conectores tipo banana**

Los interruptores **(a)** ,**(b)**, **(c)** y **(d)** permiten energizar cada uno de los conectores tipo banana hembra los cuales se utilizan para encender cualquier módulo que se esté usando en el momento de la prueba termográfica .

### **Parada de emergencia**

La parada de emergencia, interruptor normalmente cerrado **(g)**, será accionado en caso de emergencia, si se percibe alguna anomalía, en tal caso, se tendrán las siguientes recomendaciones de uso:

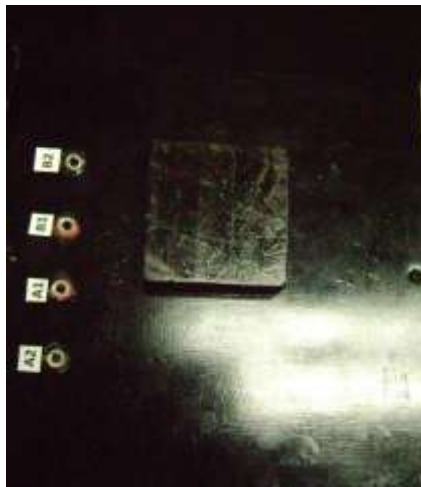
1. Accionar la parada de emergencia en caso de mal funcionamiento del sistema.
2. Si es accionada la parada de emergencia se interrumpirá la energía total del banco, lo cual permitirá protección del sistema ante posibles sobrecargas térmicas o eléctricas.
3. Después de presionar la parada de emergencia hay que tener en cuenta que no se puede desenchavar el interruptor inmediatamente ya que el sistema seguirá funcionando.
4. Una vez enclavado el interruptor de emergencia se deberá cambiar al estado "OFF" los interruptores de encendido principal y del motor (si este está en uso).

## CONECTORES PARA BANANA TIPO HEMBRA



Cada uno de los conectores para banana tipo hembra es energizado por los interruptores **(a)**, **(b)**, **(c)** y **(d)** respectivamente, cada conector está debidamente señalado con su correspondiente interruptor, se utilizan cables tipo banana macho para energizar los módulos a usar.

## SUJECIÓN DE MÓDULOS



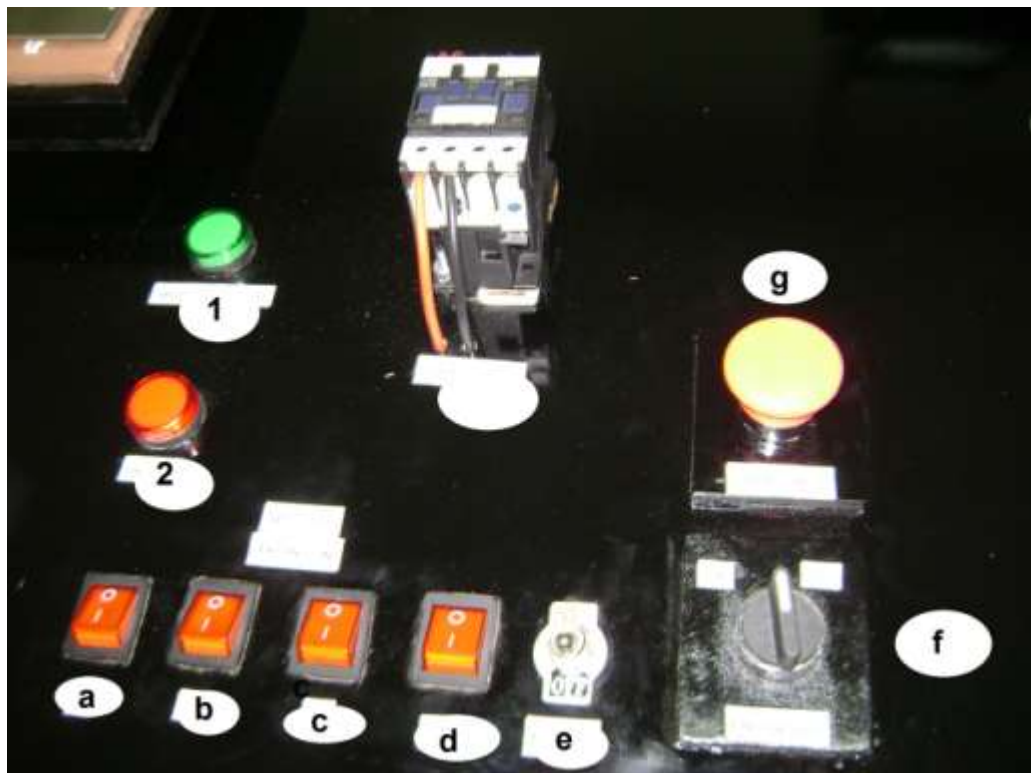
Cada módulo dispone en su parte posterior un soporte hembra para ajustar a una altura apropiada y así poder tomar las fotografías y hacer las pruebas de una forma cómoda y sencilla

## ANEXO B

### APLICACIÓN MATLAB: MANUAL DE USO

Este manual contiene los pasos necesarios para la respectiva toma de imágenes o videos térmicos y su tratamiento. Serán necesarios 3 elementos: Cámara termográfica (en este caso la TI125 de industrias FLUKE y su respectivo manual), banco de trabajo para pruebas termográficas (ubicado en el laboratorio de vibraciones mecánicas en el edificio de laboratorios pesados) y un portátil o computador de mesa con el software MATLAB y la aplicación (la cual se encuentra en el CD anexo al libro de trabajo de grado). **Para realizar el procedimiento, se debe tener en cuenta que, anteriormente, el manual de la cámara como el del banco de trabajo han sido estudiados y entendidos.**

#### PARTE A: Encendido del Banco de Trabajo



Conectar el banco de trabajo, accionar el interruptor **(f)** “ON”, verificar que el piloto **(2)** esté encendido en rojo (esto indica que el banco está energizado). Si se desea trabajar con los módulos; montar el módulo que se desee analizar en la barra soporte, conectar los cables banana macho con sus respectivos conectores hembra (debidamente indicados en la parte posterior de los módulos y en la mesa), para energizar las resistencias requeridas, revisar el encendido de cada interruptor de resistencia en rojo **(a-b-c-d)** prendiendo y apagando una vez para verificación. Si se desea trabajar con el motor y los rodamientos, encender y apagar una vez el interruptor **(e)** y ver su respectiva energización mediante el piloto **(1)**, en verde, para verificación.

## PARTE B: Preparación Cámara Termográfica

La cámara termográfica debe ser estudiada mediante la lectura de su respectivo manual y seleccionar los siguientes parámetros de medición según el módulo a analizar.

Parámetro \ Módulo	Forma	Tamaño	Profundidad	Posición	Aletas	Rodamientos
<b>Rango</b>	Automático					
<b>Temperatura fondo</b>	Temperatura ambiente					
<b>Emisividad</b>	0,55	0,55	0,90	0,95	0,95	0,95
<b>Paleta de color</b>	Arco de Hierro					
<b>Marcadores</b>	Ninguno					
<b>Formato</b>	.AVI y .JPG					

Cuando todos los parámetros correspondientes estén activados, se procede a capturar la imagen o el video de la siguiente manera, **teniendo en cuenta los consejos del capítulo 1, sección 1.6:**

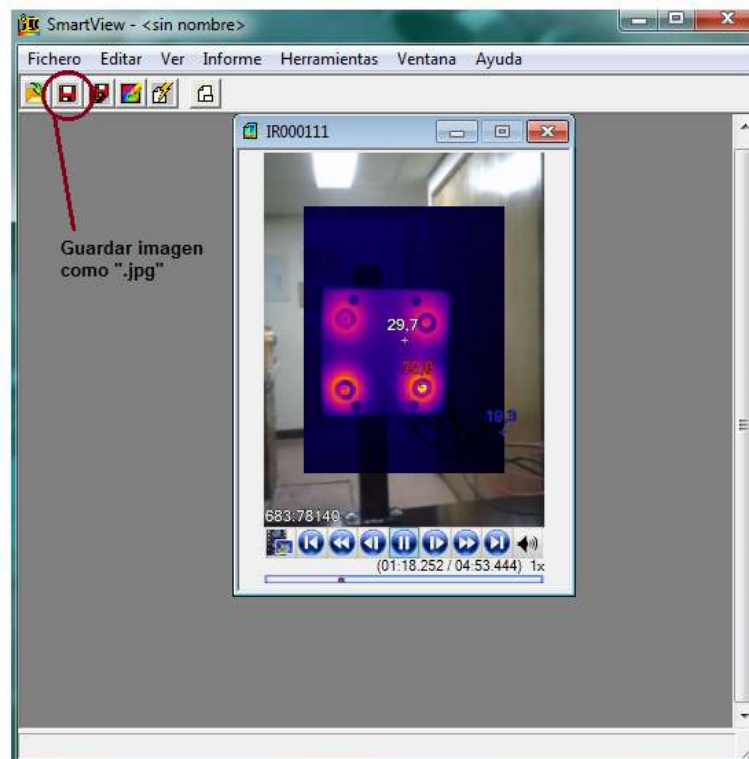
## IMAGEN

Encender las resistencias del módulo; esperar 40 a 50 segundos, aproximadamente, para que la temperatura se estabilice y sea uniforme; presionar el botón de toma de imágenes de la cámara, la cual, automáticamente, guardará en la memoria SD; conectar esta al computador y copiar el archivo (según la parte C).

Si se desea obtener una imagen mediante el **software smartview 3.2** (únicamente para productos FLUKE), se procede de la siguiente manera:

Instalar el software mediante el CD que viene con la cámara (este permite ver la grabación en tiempo real); conectar la cámara al computador por medio del cable USB; abrir el programa; encender las resistencias del módulo; esperar 40 a 50 segundos, aproximadamente, para que la temperatura se estabilice y sea uniforme; grabar la imagen (según la parte C) directamente como la siguiente figura:

### Entorno Smartview 3.2



## **VIDEO**

Entrar al menú de la cámara con **F2**, escoger la opción “**Cámara**”, escoger la opción “**Video**”, escoger la opción “**Sólo video**”, click en **F1** para aceptar, encender las resistencias del módulo; esperar 40 a 50 segundos, aproximadamente, para que la temperatura se estabilice y sea uniforme, presionar el botón de toma de imágenes para iniciar el video, esperar 5min de grabación máxima, dar click en la opción “**Guardar**” y automáticamente estará en la memoria SD, conectar esta al computador y copiar el archivo (según la parte C).

## **PARTE C: Tratamiento de la imagen o del video**

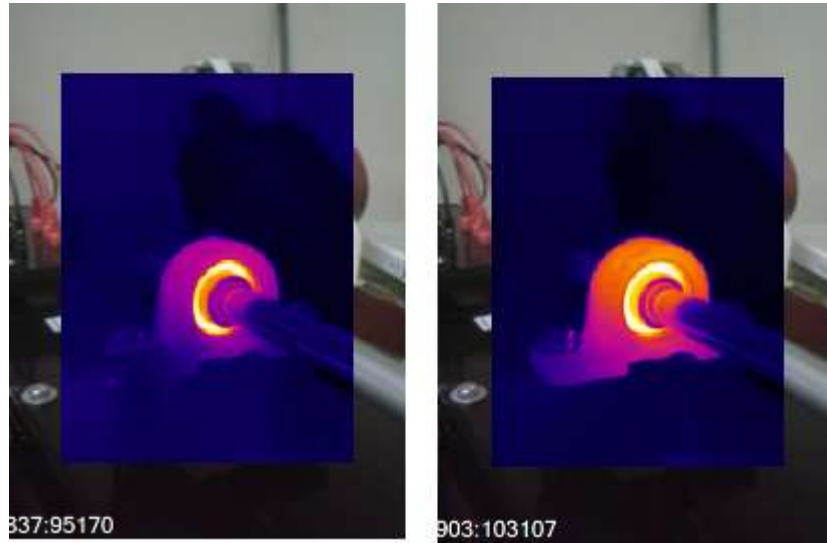
- **Tratamiento de Imágenes**

La aplicación “*Tratamiento*” para imágenes se encuentra dentro de la carpeta “*Análisis de Temperatura*” en el CD principal del proyecto.

***Toda imagen que se quiera analizar debe estar dentro de esta carpeta en el Workspace de MATLAB.***

Esta aplicación trabaja gracias a dos imágenes, la primera debe ser aquella que muestre un comportamiento de temperatura estable del artefacto que se quiera analizar, debido a que será nuestro punto de comparación en estado “normal” de la segunda, la cual debe ser este mismo artefacto pero con un cambio en la temperatura visible para poder realizar la tarea.

## Rodamiento en estado normal de operación y con calentamiento



(a)

(b)

A la imagen (a) la llamaremos “*inicial.jpg*” y la guardaremos en la carpeta anteriormente dicha, mientras que a la (b) la llamaremos “*prueba.jpg*” y la guardaremos en el mismo lugar. Si los nombres no son los sugeridos se debe proceder al código directamente para cambiar la fuente (ver anexo D).

Al tener las dos imágenes necesarias se abrirá el archivo “*Tratamiento.m*”, se dará play y se procederá según las opciones descritas en el capítulo 3, sección 3.1.

- **Tratamiento de Video**

La aplicación “*Análisis*” para videos se encuentra dentro de la carpeta “*Análisis de Temperatura*” en el CD principal del proyecto.

***Todo video que se quiera analizar debe estar dentro de esta carpeta en el Workspace de MATLAB.***

Esta aplicación trabaja gracias a un video que muestre el cambio de la temperatura en el tiempo. A este lo llamaremos “**prueba.avi**” y lo guardaremos en la carpeta anteriormente dicha. Si el nombre no es el sugerido, se debe proceder al código y cambiar la fuente (ver anexo E).

Al tener el video necesario se abrirá el archivo “**Análisis.m**”, se dará play y se procederá según las opciones descritas en el capítulo 3, sección 3.2.

## **ANEXO C**

### **MANUAL DE MANTENIMIENTO PREVENTIVO DEL BANCO**

#### **RECOMENDACIONES DE MANTENIMIENTO**

La labor de Mantenimiento propuesta consiste básicamente en limpieza, revisión, ajustes, correcciones y reparaciones menores que deben cumplirse con una frecuencia predeterminada.

#### **ASPECTOS GENERALES**

Los objetivos propuestos se alcanzan por medio de un programa de actividades básicas donde se realizan limpiezas generales, revisión evaluación y mejoramiento de desempeño , eliminación de ruidos ajenos al funcionamiento habitual, toma y evaluación de datos físicos que comparados con datos determinados para condiciones normales de operación dan la pauta para establecer la condición real del equipo y determinar así, los pasos a seguir.

#### **INSPECCIONES PERIODICAS PROGRAMADAS**

Consiste en:

- a) Lectura de amperajes y voltajes de todos los módulos que tiene el sistema para comprobar el funcionamiento normal de todos los componentes.
- b) Revisar funcionamiento adecuado del motor eléctrico.
- c) Lubricación de rodamientos.
- d) Revisar funcionamiento y protección de la instalación eléctrica.
- e) Realizar proceso de limpieza y protección de pintura.

#### **FRECUENCIA DE LAS REVISIONES**

Teniendo en cuenta la novedad de los equipos, la calidad de su instalación, el grado de exposición a agentes contaminantes y corrosivos, la frecuencia de su uso, y el grado de compromiso con los procesos de la UIS, el mantenimiento se hará en períodos de cada 15 días y se deben hacer según el siguiente orden de revisión.

## **RECOMENDACIONES DE TAREAS MINIMAS A REALIZAR PARA EL MOTOR**

Cuando el motor no se ha colocado en funcionamiento por un periodo igual o superior a 6 meses no es necesario ningún tipo de control. Gire el rotor mensualmente para una u otra posición después de 6 meses de no uso, antes de poner en uso el rodamiento más próximo al motor debe ser lubricado.

En un mantenimiento de motores eléctricos, adecuadamente aplicado, se debe inspeccionar periódicamente niveles de aislamiento, la elevación de temperatura (bobinas y soportes), desgastes, lubricación de los rodamientos, vida útil de los soportes, examinar eventualmente el ventilador, cuanto al correcto flujo de aire, niveles de vibraciones, desgastes de escobas y anillas colectoras.

El descaso de uno de los ítems anteriores puede significar paradas no deseadas del equipo. La frecuencia con que deben ser hechas las inspecciones, depende del tipo del motor y de las condiciones locales de aplicación.

La carcasa debe ser mantenida limpia, sin acumulo de aceite o polvo en su parte externa para facilitar el intercambio de calor con el medio.

### **LIMPIEZA**

El motor debe estar limpio, exento de polvacera, detritos y aceites. Para limpiarlo, se debe utilizar escobas o trapos limpios de algodón. Si el polvo no es abrasivo, se debe emplear un soplete de aire comprimido, soplando la suciedad de la tapa deflectora y eliminando todo el acumulo de polvo contenido en las aletas del ventilador y en las aletas de refrigeración.

## **REVISION COMPLETA**

Limpie las bobinas sucias con un pincel o escobilla. Use un trapo humedecido con alcohol o con solventes adecuados para remover grasa, aceite y otras suciedades que estén adheridos sobre las bobinas. Seque con aire seco.

Pase aire comprimido por entre los canales de ventilación en el paquete de chapas del estator, rotor y soportes.

Drene el agua condensada, limpie el interior de las cajas de conexión y de las anillas colectoras.

## ANEXO D

### CÓDIGO MATLAB TRATAMIENTO DE IMÁGENES

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

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

% Last Modified by GUIDE v2.5 16-Nov-2012 12:52:03

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

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

```

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

% --- Executes just before Tratamiento is made visible.
function Tratamiento_OpeningFcn(hObject, eventdata, handles, varargin)
%Al iniciar el programa se coloca el fondo
a=imread('fondo2.jpg');
image(a)
%desaparece los demas axes
axis off
% This function has no output args, see OutputFcn.
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Tratamiento (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
%Guarda el número de la opción del popupmenú en la variable opc
opc=get(handles.popupmenu1, 'value');
%Escoge la opción
switch opc
    case 1
        %Coloca la palabra IMAGEN en el texto estático 3
        set(handles.text3, 'string', 'IMAGEN')
        %Grafica en el axes1
        axes(handles.graft1);
        %lee la imagen inicial

```

```

        il=imread('inicial.jpg');
        %Muestra la imagen inicial
        imshow(il)

    case 2
        set(handles.text3,'string','ROJOS')
        axes(handles.graft1);
        il=imread('inicial.jpg');
        %convierte la imagen al plano rojo
        il_r=il(:,:,1);
        imshow(il_r)

    case 3
        set(handles.text3,'string','HISTOGRAMA')
        axes(handles.graft1);
        il=imread('inicial.jpg');
        il_r=il(:,:,1);
        %Muestra el histograma de la imagen
        imhist(il_r)

end
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
%           contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
%           popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

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

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
%Guarda el número de la opción del popupmenú2 en la variable opc
opc=get(handles.popupmenu2,'value');
switch opc

```

```

case 1
    set(handles.text5,'string','IMAGEN')
    img=imread('prueba.jpg');
    axes(handles.graft2);
    imshow(img)
case 2
    set(handles.text5,'string','ROJOS')
    img=imread('prueba.jpg');
    img_r=img(:,:,1);
    axes(handles.graft2);
    imshow(img_r)
case 3
    set(handles.text5,'string','HISTOGRAMA')
    img=imread('prueba.jpg');
    img_r=img(:,:,1);
    axes(handles.graft2);
    imhist(img_r)
case 4
    set(handles.text5,'string','FORMA')
    img=imread('prueba.jpg');
    %aumenta el brillo de la imagen
    imgb=img+60;
    im_r=imgb(:,:,1);
    %muestra los bordes de la imagen segun el método de canny y la
    %binariza
    im_borde=edge(im_r,'canny');
    %crea una rejilla cuadrada de 4 pixeles de espesor
    SE=strel('square',4);
    %diala los bordes mediante la rejilla
    bn=imdilate(im_borde,SE);
    %llena los espacios entre lineas
    bn_f=imfill(bn,'holes');
    %elimina las áreas menores a 3000
    bw=bwareaopen(bn_f,3000);

    axes(handles.graft2);
    imshow(img)
    %Guarda y etiqueta los componentes en L y el número de ellos en
Ne

    [L Ne]=bwlabel(bw);
    %Guarda las propiedades de los componentes
    propied=regionprops(L);

%crea un bucle de conteo para los elementos etiquetados
for n=1:size(propied,1)
    %grafica un rectángulo verde de espesor 2 según la propiedad boindung
box del elemento

    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2
)
    %Obtiene la medida horizontal del centroide de cada elemento
    x=propied(n).Centroid(1);
    %Obtiene la medida vertical del centroide de cada elemento
    y=propied(n).Centroid(2);

```

```

%obtiene la medida horizontal del bounding box de cada elemento
w=propied(n).BoundingBox(3);
%obtiene la medida vertical del bounding box de cada elemento
z=propied(n).BoundingBox(4);
%Obtiene el área de cada elemento
a=propied(n).Area;
    %Si la relación entre el área del elemento y la del bounding box
es
    %mayor a 0.9 es un cuadrilátero
    if (a/(w*z))>0.87
    %Escribe la palabara CUADRILÁTERO en las medidas especificadas
    text(x-40,y,'CUADRILATERO')
    else
    %Si la relación entre el área del elemento y la del bounding
box es
    %mayor a 0.75 es un Círculo
    if (a/(w*z))>0.75
    text(x-20,y,'CÍRCULO')
    else
    %Si la relación entre el área del elemento y la del
bounding box es
    %menor a 0.75 es un Triángulo
    if (a/(w*z))<0.75
    text(x-30,y,'TRIANGULO')
    else
    %si no es nada de lo anterior, no se reconoce la figura
    text(x-20,y,'NO SE RECONOCE')
    end
    end
    end
end
case 5
    set(handles.text5,'string','PUNTOS CALIENTES')
    img=imread('prueba.jpg');
    img_r=img(:,:,1);
    %Obtiene en f el número de filas y en c el número de columnas
    [f,c]=size(img_r);
    axes(handles.graft2)
    imshow(img)
    %mantiene los datos actuales
    hold on
    %crea un bucle que recorre todas las filas
    for i=1:f
    %crea un bucle que recorre todas las columnas
    for j=1:c
    %guarda el valor en unit8 (0 - 255)
    val=img_r(i,j);
    %si el valor supera 200 (Temperatura mayor a 90°C)
    if val>200
    %asigne color amarillo en la coordenada (j,i)
    plot(j,i,'y')
    else
    %si el valor supera 150 (Temperatura entre 70 y 90°C)
    if val>150

```

```

        %asigne color rojo en la coordenada (j,i)
        plot(j,i,'r')
    else
        %si el valor supera 100 (Temperatura entre 57 y 70°C)
        if val>80
            %asigne color azul en la coordenada (j,i)
            plot(j,i, '.')
        end
    end
end
end
end
end
%No mantiene los datos actuales
hold off
case 6
    img=imread('prueba.jpg');
    imgb=img+60;
    img_r=imgb(:,:,1);
    %Obtiene el umbral de img_r
    umb=graythresh(img_r);
    %Binariza según el umbral a img_r
    bn=im2bw(img_r,umb);
    %Elimina las áreas menores a 1000
    bw=bwareaopen(bn,1000);
    axes(handles.graft2)
    imshow(img)
    [L Ne]=bwlabel(bw);
    propied=regionprops(L);

for n=1:size(propied,1)

rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2
)
x=propied(n).Centroid(1);
y=propied(n).Centroid(2);
a=[propied.Area];
    %Guarda el área máxima en b
    b=max(a);
    %Guarda el área mínima en c
    c=min(a);
    %Si el área es igual a la máxima
    if propied(n).Area==b;
        %Escriba MAYOR en las coordenadas específicas
        text(x-40,y,'MAYOR')
    else
        %Si el área es igual a la mínima
        if propied(n).Area==c;
            %Escriba MENOR en las coordenadas específicas
            text(x-20,y,'MENOR')
        else
            %Si no es nada de lo anterior, escriba MEDIO
            text(x-20,y,'MEDIO')
        end
    end
end
end

```

```

end

end
% hObject      handle to popupmenu2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%CARGAMOS, MOSTRAMOS Y LEEMOS LA IMAGEN INICIAL
img1=imread('inicial.jpg');
axes(handles.graft1)
imshow(img1)
img1_r=img1(:,:,1);
x=imhist(img1_r);

%CARGAMOS, MOSTRAMOS Y LEEMOS LA IMAGEN A COMPARAR
img2=imread('prueba.jpg');
axes(handles.graft2)
imshow(img2)
img2_r=img2(:,:,1);
y=imhist(img2_r);

%REALIZAMOS LA CONDICION
%Si el valor absoluto de la resta de la cantidad de rojos en la imagen
%inicial con los de la imagen prueba es mayor a 20 (cambio aproximado de
%20°C)
for j=1:50:250
if abs(x(j) - y(j)) > 20;
    %Mostar mensaje alerta cambio

```

```

        msgbox('ALERTA, la temperatura ha cambiado, porfavor revisar
en la imagen el cambio de temperatura y los puntos criticos')
        dbstop
else
end
end
msgbox('funcionamiento normal')
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

## ANEXO E

### CÓDIGO MATLAB TRATAMIENTO DE VIDEO

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

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

% Last Modified by GUIDE v2.5 11-Nov-2012 22:48:31

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

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

```

% --- Executes just before Analisis is made visible.
function Analisis_OpeningFcn(hObject, eventdata, handles, varargin)
%Al iniciar el programa se coloca el fondo
a=imread('fondo2.jpg');
image(a)
%desaparece los demas axes
axis off
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Analisis (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
%Guarda el número de la opción del popupmenú en la variable val
val=get(handles.popupmenu1, 'value');
%Escoge la opción
switch val
    case 1
        %Grafica en el axes1
        axes(handles.graft1);
        %Coloca la palabra PAUSA en el texto estático 1
        set(handles.text1, 'string', 'PAUSA')
        %lee y guarda la imagen que se mostrará cuando el video esté en
        %pausa
        i1=imread('opciones.jpg');
        %Muestra la imagen de pausa
        imshow(i1)

```

```

case 2
set(handles.text1,'string','VIDEO')
%Lee y guarda el video para analizar
obj = mmreader('prueba.avi');
%Guarda el número de frames del video
nFrames = obj.NumberOfFrames;
%Guarda el alto del frame
vidHeight = obj.Height;
%Guarda el ancho del frame
vidWidth = obj.Width;

% Preasigna la estructura del video
mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

% Crea un bucle para leer frame por frame
for k = 1:5:nFrames-1
%Lee y guarda la estructura del nuevo video
mov(k).cdata = read(obj, k);
%Convierte cada frame del video en una imagen
frame=mov(k).cdata;
axes(handles.graft1);
imshow(frame)
%Pausa la imagen para poder ser apreciada
pause(0.0001)
end

case 3
set(handles.text1,'string','ROJOS')
obj = mmreader('prueba.avi');
nFrames = obj.NumberOfFrames;
vidHeight = obj.Height;
vidWidth = obj.Width;

mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

for k = 1:5:nFrames-1
mov(k).cdata = read(obj, k);
frame=mov(k).cdata;
%Convierte la imagen al plano ROJO
frame_r=frame(:,:,1);
axes(handles.graft1);
imshow(frame_r)
pause(0.0001)
end

case 4
set(handles.text1,'string','HISTOGRAMA ROJOS')
obj = mmreader('prueba.avi');

```

```

nFrames = obj.NumberOfFrames;
vidHeight = obj.Height;
vidWidth = obj.Width;

mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

for k = 1:5:nFrames-1
mov(k).cdata = read(obj, k);
frame=mov(k).cdata;
frame_r=frame(:,:,1);
axes(handles.graft1);
%Obtiene el histograma de la imagen
imhist(frame_r)
pause(0.0001)
end

end

% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

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

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
opc=get(handles.popupmenu2,'value');
switch opc

    case 1
        axes(handles.graft2);

```

```

set(handles.text4,'string','PAUSA')
il=imread('opciones.jpg');
imshow(il)

case 2
set(handles.text4,'string','FORMA')
obj = mmreader('prueba.avi');
nFrames = obj.NumberOfFrames;
vidHeight = obj.Height;
vidWidth = obj.Width;

mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

for k = 1:5:nFrames-1
mov(k).cdata = read(obj, k);
frame=mov(k).cdata;
%Aumenta el brillo de la imagen
frame1=frame+60;
imgb=frame1(:,:,1);
%Obtiene los bordes de la imagen mediante el logaritmo de canny
im_borde=edge(imgb,'canny');
%Crea una estructura cuadrada de 4 pixeles de espesor
SE=strel('square',4);
%Dilata los límites de la imagen con la estructura anterior
bn=imdilate(im_borde,SE);
%Llena los espacios vacios dentro de la figura
bn_f=imfill(bn,'holes');
%Elimina áreas menores consideradas como ruido
bw=bwareaopen(bn_f,3000);
%Guarda y etiqueta las figuras y el número de ellas
[L Ne]=bwlabel(bw);
%Obtiene las propiedades principales de las figuras
prop=regionprops(L);
axes(handles.graft2);
imshow(frame)
pause(0.0001)

%Crea un bucle que recorra todas las figuras resultantes
for n=1:size(prop,1);
%Grafica un rectángulo según el boundingbox alrededor de la
figura
rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
%obtiene centroides
x=prop(n).Centroid(1);
y=prop(n).Centroid(2);
%Obtiene medidas del boundingbox
w=prop(n).BoundingBox(3);
z=prop(n).BoundingBox(4);
%Obtiene áreas
a=prop(n).Area;
%Si la relación entre el área de la figura y la del

```



```

    %Obtiene el perímetro de las figuras resultantes
    B=bwboundaries(bn);
    %Crea un bucle que recorre todas las figuras
    for n=1:length(B);
    %Crea un vector que guarda los perímetros
    boundary=B{n};
    %Dibuja los perímetros de cada figura
    plot(boundary(:,2),boundary(:,1),'r','LineWidth',2)
    pause(0.0001)
    end
    hold off
end

case 4
set(handles.text4,'string','TAMAÑO')
obj = mmreader('prueba.avi');
nFrames = obj.NumberOfFrames;
vidHeight = obj.Height;
vidWidth = obj.Width;

mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

for k = 1:5:nFrames-1
mov(k).cdata = read(obj, k);
frame=mov(k).cdata;
frame1=frame+60;
imgr=frame1(:,:,1);
umb=graythresh(imgr);
bn=im2bw(imgr,umb);
bw=bwareaopen(bn,1000);
axes(handles.graft2)
imshow(frame1)
pause(0.001)
[L Ne]=bwlabel(bw);
propied=regionprops(L);

for n=1:size(propied,1)
rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2
)
x=propied(n).Centroid(1);
y=propied(n).Centroid(2);
a=[propied.Area];
b=max(a);
c=min(a);
if propied(n).Area==b;
text(x,y,'MAYOR')
else
    if propied(n).Area==c;
text(x,y,'MENOR')
    else
text(x,y,'MEDIO')
    end
end
end

```

```

        end
    end
    end
    pause(0.001)
    end
end

```

```

% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu2

```

```

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

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

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
    set(handles.text2,'string','IMAGEN INICIAL')
    set(handles.text3,'string','VIDEO PRUEBA')
    obj = mmreader('prueba.avi');
    nFrames = obj.NumberOfFrames;
    vidHeight = obj.Height;
    vidWidth = obj.Width;

    % Preallocate movie structure.
    mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);

    % Read one frame at a time.
    for k = 1:nFrames-1
        mov(k).cdata = read(obj, k);
    end

```

```

frame1=mov(1).cdata;
axes(handles.graft1)
imshow(frame1)
frame1_r=frame1(:,:,1);
x=imhist(frame1_r);

frame2=mov(k).cdata;
axes(handles.graft2)
imshow(frame2)
pause(0.001)
frame2_r=frame2(:,:,1);
y=imhist(frame2_r);

for m=1:50:250
    if abs(x(m) - y(m)) > 50;
        msgbox('ALERTA, la temperatura ha cambiado, porfavor revisar
en la imagen el cambio de temperatura y los puntos
criticos', 'Alerta', 'warn')
        ['La temperatura ha cambiado a los ', num2str(k/10), '
segundos']
        dbstop
    end
end
end
msgbox('funcionamiento normal')
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
pause(5)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```