



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

**DISEÑO Y CONTRUCCIÓN DE UN DISPOSITIVO**  
**ELECTRÓNICO DE MONITOREO INFRARROJO PARA**  
**SUPERVISIÓN DE TEMPERATURA DE NIÑOS MENORES A 3**  
**AÑOS**

**Trabajo de titulación**

**Tipo:** Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTORA:** SOLEDAD SALOMÉ ARMIJOS SANTANA

Riobamba-Ecuador

2021



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

**DISEÑO Y CONTRUCCIÓN DE UN DISPOSITIVO**  
**ELECTRÓNICO DE MONITOREO INFRARROJO PARA**  
**SUPERVISIÓN DE TEMPERATURA DE NIÑOS MENORES A 3**  
**AÑOS**

**Trabajo de titulación**

**Tipo:** Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTORA:** SOLEDAD SALOMÉ ARMIJOS SANTANA

**DIRECTOR:** ING. PABLO EDUARDO LOZADA YÁNEZ

Riobamba-Ecuador

2021

**©2021, Soledad Salomé Armijos Santana**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, **Soledad Salomé Armijos Santana**, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

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

Riobamba, 24 de junio de 2021.

**Soledad Salomé Armijos Santana**

**172420428-2**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERIA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

El Tribunal de trabajo de titulación certifica que: El trabajo titulación tipo: Proyecto Técnico, **“DISEÑO Y CONTRUCCION DE UN DISPOSITIVO ELECTRÓNICO DE MONITOREO INFRARROJO PARA SUPERVISIÓN DE TEMPERATURA DE NIÑOS MENORES A 3 AÑOS”**, de responsabilidad de la señorita **SOLEDAD SALOMÉ ARMIJOS SANTANA**, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales en tal virtud el tribunal autoriza su presentación.

**FIRMA**

**FECHA**

Ing. Raúl Marcelo Lozada Yáñez  
**PRESIDENTE DEL TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

Ing. Pablo Eduardo Lozada Yáñez  
**DIRECTOR DEL TRABAJO DE**  
**TITULACION**

\_\_\_\_\_

\_\_\_\_\_

Ing. Jorge Luis Paucar Samaniego  
**MIEMBRO DE TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

## **DEDICATORIA**

El presente trabajo lo dedico a mis padres pues gracias a su cariño, apoyo y paciencia me han ayudado a realizarme personal y académicamente, forjando en mí valores y principios para que hoy pueda cumplir con uno de este que es uno de mis objetivos. Gracias por el esfuerzo constante que dedican a sus hijas porque nos demuestran que con amor y perseverancia infinita se puede superar cualquier adversidad que se presente en la vida.

Soledad

## **AGRADECIMIENTO**

Gracias a Dios por permitir disfrutar de la vida y presencia de mi familia. Gracias a mis hermanos Caridad, Rubí, Kevin, Lenin, a mis amigos Andrés Santander y Cristian Pañi quienes me han acompañado en toda esta etapa universitaria, gracias por su gran ayuda y ser luz en mi vida en momentos y situaciones complicadas, por todas las risas y momentos compartidos sin ustedes mi permanencia universitaria no hubiese sido totalmente agradable.

Soledad

## TABLA DE CONTENIDO

INDICE DE TABLAS.....	IX
INDICE DE FIGURAS.....	X
INDICE DE GRÁFICOS.....	XI
INDICE DE ECUACIONES .....	¡ERROR! MARCADOR NO DEFINIDO.
INDICE DE ANEXOS .....	XII
INDICE DE ABREVIATURAS.....	¡ERROR! MARCADOR NO DEFINIDO.
RESUMEN.....	XI
<b>II</b>	
ABSTRACT.....	XI
<b>V</b>	
INTRODUCCION .....	1
<b>CAPITULO I</b>	
1 <b>DIAGNOSTICO DEL PROBLEMA.....</b>	<b>3</b>
1.1 <b>Antecedentes.....</b>	<b>3</b>
1.2 <b>Delimitación del tema .....</b>	<b>4</b>
1.2.1 <i>Realidad del tema a nivel mundial .....</i>	<i>4</i>
1.2.2 <i>Realidad del tema a nivel regional .....</i>	<i>5</i>
1.2.3 <i>Realidad del tema a nivel local.....</i>	<i>5</i>
1.3 <b>Objetivos.....</b>	<b>6</b>
1.3.1 <i>Objetivo General .....</i>	<i>6</i>
1.3.2 <i>Objetivos Específicos .....</i>	<i>6</i>
1.4 <b>Justificación aplicativa .....</b>	<b>7</b>
<b>CAPITULO II</b>	
2 <b>FUNDAMENTO TEORICO.....</b>	<b>8</b>
2.1 <b>Termografía .....</b>	<b>8</b>
2.1.1 <b>Concepto.....</b>	<b>8</b>
2.1.2 <b>Tipos de Termografía .....</b>	<b>8</b>
2.1.2.1 <i>Termografía activa .....</i>	<i>8</i>
2.1.2.2 <i>Termografía pasiva.....</i>	<i>8</i>
2.1.3 <b>Emisividad, Reflexión y Transmisión .....</b>	<b>9</b>
2.2 <b>Espectro Electromagnético .....</b>	<b>10</b>
2.2.1 <b>Definición.....</b>	<b>10</b>



2.3	Onda.....	10
2.4	Calor y Temperatura.....	11
2.5	Cámara Termográfica.....	12
2.5.1	<i>Funcionamiento</i> .....	12
2.6	Imagen Radiométrica .....	12
2.6.1	<i>Resolución Radiométrica</i> .....	13
2.7	Pixel, FOV & IFOV .....	13
2.8	Visión Artificial.....	14
2.8.1	<i>Definición</i> .....	14
2.8.2	<i>Etapas de un sistema de visión artificial</i> .....	14
2.8.3	<i>Aplicaciones</i> .....	15
2.9	Microcontroladores .....	17
2.9.1	<i>Arduino</i> .....	17
2.9.2	<i>Raspberry</i> .....	18
2.10	Cámaras.....	19
2.10.1	<i>Cámara Web Cam</i> .....	19
2.10.2	<i>Cámara Raspberry Pi</i> .....	19
2.10.3	<i>Cámara Flir Lepton 2.5</i> .....	20
2.11	Software.....	22
2.11.1	<i>Phyton</i> .....	22
2.11.1.1	<i>Calidad de Software</i> .....	22
2.11.1.2	<i>Productividad del desarrollador</i> .....	23
2.11.1.3	<i>Portabilidad del programa</i> .....	23
2.11.1.4	<i>Bibliotecas de apoyo</i> .....	23
2.11.2	<i>Matlab</i> .....	23

### CAPITULO III

3	MARCO METODOLOGICO.....	25
3.1	Proceso de Investigación .....	25
3.2	Elección de elementos para el hardware y software a utilizar.....	25
3.3	Diagrama de funcionamiento del Trabajo de Titulación .....	26
3.3.1	<i>Diagrama General</i> .....	26
3.4	Elementos de Hardware del prototipo .....	27
3.4.1	<i>Raspberry Pi</i> .....	27
3.4.2	<i>Elección de la cámara para el prototipo</i> .....	29
3.4.3	<i>Conexiones del prototipo</i> .....	31

<b>3.5</b>	<b>Elementos del software del prototipo.....</b>	<b>32</b>
3.5.1	<i>Análisis de características entre el algoritmo utilizado con los existentes .....</i>	32
3.5.2	<i>Python .....</i>	33
<b>3.6</b>	<b>Desarrollo del Software de Control.....</b>	<b>34</b>
3.6.1	<i>Operación General del Sistema .....</i>	34
<b>3.7</b>	<b>Implementación del Software de Control.....</b>	<b>36</b>
3.7.1	<i>Hilo principal .....</i>	37
3.7.2	<i>Hilo 1 .....</i>	37
3.7.3	<i>Proceso y Subproceso .....</i>	38
<b>3.8</b>	<b>Desarrollo del Algoritmo de Medición de Temperatura .....</b>	<b>38</b>
3.8.1	<i>Método constructor .....</i>	40
3.8.2	<i>Proceso en paralelo.....</i>	42
3.8.3	<i>Análisis del programa principal .....</i>	42
3.8.4	<i>Procesamiento de imágenes y medición de temperatura .....</i>	45
3.8.4.1	<i>Cambio de resolución de imagen .....</i>	46
3.8.4.2	<i>Detección de rostro .....</i>	47
3.8.4.3	<i>Detección de ojos en el rostro .....</i>	48
3.8.4.4	<i>Delimitación de la región de interés en imagen RGB .....</i>	48
3.8.4.5	<i>Determinar la región de interés en una imagen térmica.....</i>	49
3.8.4.6	<i>Medición de Temperatura.....</i>	49

#### **CAPITULO IV**

<b>4</b>	<b>MARCO DE RESULTADOS, ANALISIS Y DISCUSIÓN.....</b>	<b>53</b>
4.1	<b>Pruebas de Funcionamiento del Prototipo .....</b>	<b>53</b>
4.2	<b>Funcionalidad del prototipo.....</b>	<b>55</b>
4.3	<b>Resultados de Evaluación del Prototipo .....</b>	<b>57</b>
4.3.1	<i>Cálculo de MAE Y RMSE .....</i>	61
4.4	<b>El Análisis de costos.....</b>	<b>65</b>
4.5	<b>Análisis del costo del prototipo desarrollado con los productos ya existentes. ....</b>	<b>65</b>

<b>CONCLUSIONES.....</b>	<b>67</b>
--------------------------	-----------

<b>RECOMENDACIONES.....</b>	<b>68</b>
-----------------------------	-----------

#### **GLOSARIO**

#### **BIBLIOGRAFIA**

#### **ANEXOS**

## INDICE DE TABLAS

<b>Tabla 1-2:</b>	Características técnicas de la cámara flir lepton 2.5.....	<b>21</b>
<b>Tabla 1-3:</b>	Comparación entre los atributos de los modelos de raspberry .....	<b>28</b>
<b>Tabla 2-3:</b>	Comparación entre modelos de cámaras compatibles con raspberry pi 3 .....	<b>29</b>
<b>Tabla 3-3:</b>	Conexión cámara flir lepton con raspberry pi. ....	<b>32</b>
<b>Tabla 4-3:</b>	Ventajas y desventajas entre los algoritmos de reconocimiento facial y objetos..	<b>33</b>
<b>Tabla 1-4:</b>	Pruebas de funcionamiento.....	<b>55</b>
<b>Tabla 2-4:</b>	Pruebas de medición de temperatura fase a.....	<b>58</b>
<b>Tabla 3-4:</b>	Pruebas de medición de temperatura fase b.....	<b>59</b>
<b>Tabla 4-4:</b>	Pruebas de medición de temperatura fase c.....	<b>60</b>
<b>Tabla 5-4:</b>	Cálculo de los errores medios absolutos y relativos de cada fase. ....	<b>61</b>
<b>Tabla 6-4:</b>	Costos del prototipo.....	<b>65</b>
<b>Tabla 7-4:</b>	Ventajas y desventajas de los productos existentes en el mercado .....	<b>65</b>

## INDICE DE FIGURAS

<b>Figura 1-1:</b>	Funcionamiento del sistema.....	7
<b>Figura 1-2:</b>	Reflexión, emisividad y transmisión.....	9
<b>Figura 2-2:</b>	Onda electromagnética.....	10
<b>Figura 3-2:</b>	Escala de un termómetro.....	11
<b>Figura 4-2:</b>	Partes de una cámara termográfica. ....	12
<b>Figura 5-2:</b>	Representación de la matriz de un píxel. ....	13
<b>Figura 6-2:</b>	Concepto de funcionamiento de un proceso utilizando visión artificial. ....	16
<b>Figura 7-2:</b>	Tarjeta arduino. ....	17
<b>Figura 8-2:</b>	Esquema de conexión de raspberry pi.....	18
<b>Figura 9-2:</b>	Cámara web modelo logitech b910 hd.....	19
<b>Figura 10-2:</b>	Cámara raspberry pi.....	20
<b>Figura 11-2:</b>	Cámara flir lepton 2.5 .....	20
<b>Figura 1-3:</b>	Raspberry pi 3.....	28
<b>Figura 2-3:</b>	Cámara para raspberry pi 3.....	30
<b>Figura 3-3:</b>	Cámara térmica flir lepton 2.5.....	30
<b>Figura 4-3:</b>	Conexión de los dispositivos del prototipo.....	31
<b>Figura 5-3:</b>	Conexión de la cámara flir lepton a raspberry pi 3.....	32
<b>Figura 6-3:</b>	Código de programación en el software python.....	34
<b>Figura 7-3:</b>	Funcionamiento del sistema.....	34
<b>Figura 8-3:</b>	Estructura de la interfaz gráfica.....	35
<b>Figura 9-3:</b>	Comunicación de los terminales de pipe.....	41
<b>Figura 10-3:</b>	Proyección de colores dispuestos en cmrmap.....	44
<b>Figura 11-3:</b>	Código utilizados para detección de rostros.....	48
<b>Figura 12-3:</b>	Visualización de la imagen rgb y térmica en la gui.....	49
<b>Figura 13-3:</b>	Datos calculados para la obtención de la matriz aplicada en la ecuación.....	52
<b>Figura 1-4:</b>	Prototipo de medición de temperatura.....	53
<b>Figura 2-4:</b>	Termómetro digital infrarrojo modelo ht-668.....	54
<b>Figura 3-4:</b>	Funcionamiento del dispositivo electrónico de temperatura infrarrojo.....	54
<b>Figura 4-4:</b>	Mensaje de alerta febril recibido al número registrado en el programa.....	55
<b>Figura 5-4:</b>	Protocolo experimental.....	57

## INDICE DE GRÁFICOS

<b>Gráfico 1-2:</b>	Diagrama de bloques de las etapas de un sistema de v.a.....	<b>15</b>
<b>Gráfico 1-3:</b>	Diagrama de flujo del funcionamiento general de la interfaz gráfica. ....	<b>27</b>
<b>Gráfico 2-3:</b>	Diagrama de proceso de aplicación del dispositivo.....	<b>37</b>
<b>Gráfico 3-3:</b>	Procesamiento de imágenes y cambio de temperatura. ....	<b>46</b>
<b>Gráfico 4-3:</b>	Resultados de temperatura vs roi, obteniendo valores de r y o. ....	<b>52</b>
<b>Gráfico 1-4:</b>	Funcionamiento del sistema. ....	<b>56</b>
<b>Gráfico 2-4:</b>	Temperatura fase a del termómetro vs prototipo.....	<b>63</b>
<b>Gráfico 3-4:</b>	Temperatura fase b del termómetro vs prototipo.....	<b>63</b>
<b>Gráfico 4-4:</b>	Temperatura fase c del termómetro vs prototipo.....	<b>64</b>
<b>Gráfico 5-4:</b>	Error medio y absoluto del sistema. ....	<b>64</b>

## **INDICE DE ANEXOS**

**Anexo A:** Base inferior en SolidWorks de la estructura del prototipo.

**Anexo B:** Base superior en SolidWorks de la estructura del prototipo.

**Anexo C:** Estructura de la cámara RGB y Térmica del prototipo.

**Anexo D:** Soporte para la cámara RGB del prototipo.

**Anexo E:** Código de adquisición de video de ambas cámaras.

**Anexo F:** Código en Python del mensaje de texto a través de GSM.

**Anexo G:** Código de procesamiento de imágenes.

**Anexo H** Código de detección de puntos de interés en el rostro.

**Anexo I:** Código de medición de temperatura en la ROI de imagen térmica.

## RESUMEN

El objetivo del presente trabajo de titulación fue el diseño y construcción de un dispositivo electrónico de monitoreo infrarrojo para supervisión de temperatura de niños menores a 3 años. Se desarrolló la construcción del prototipo utilizando dos cámaras para la captura de imágenes, una para la imagen RGB y otra para la imagen térmica, el sistema es controlado mediante el microcontrolador Raspberry Pi 3, la temperatura medida del niño se visualiza mediante la interfaz gráfica creada en donde si esta sobrepasa el umbral establecido que es mayor a 37 grados centígrados se enviará un mensaje de texto al celular de la persona encargada del niño indicando que se encuentra en estado febril. Para el funcionamiento del sistema y la creación del código fuente se han utilizado varias técnicas para la ejecución del mismo, entre ellos el histograma de gradientes orientados (HOG) para detección humana, así como también se utilizó métodos como el paralelismo, hilos y multiprocesos los mismos que se basan en el lenguaje de programación Python que es el software elegido ya que permite la creación de códigos legibles, ahorrando recursos y tiempo. Analizados los diferentes dispositivos existentes en el mercado se concluyó que el dispositivo implementado es un aparato tecnológico amigable con el niño, que mide su estado corporal, capaz de alertar a la persona encargada del infante sobre el estado corporal en el que se encuentre. Realizadas las pruebas de funcionamiento y de errores calculados como el error absoluto medio (MAE) y el error cuadrático medio (RMSE) se obtuvo que el dispositivo cumple con los requisitos planteados. Se recomienda el desarrollo de un algoritmo de aprendizaje automático que determine mediante la imagen térmica la región de interés lo cual reduciría el diseño del hardware debido a que la cámara RGB no es esencial.

**Palabras Claves:** <DISPOSITIVO TECNOLÓGICO>, <VISIÓN ARTIFICIAL>, <CÁMARA TÉRMICA >, <TERMOGRAFÍA>, <ESTADO FEBRIL>.

## **ABSTRACT**

The aim of this graduate research project was the design and construction of an electronic infrared monitoring device for monitoring temperature of children under 3 years of age. The prototype was developed using two cameras to capture images, one for RGB images and the other for thermal images. The system is controlled by the Raspberry Pi 3 microcontroller and the child's measured temperature is displayed through the graphical interface created. If a child's temperature exceeds the established threshold of greater than 37 degrees Celsius, a textmessage will be sent to the cell phone of the person responsible for the child, indicating that they have a fever. Several techniques were used to execute system operation and the creation of Source Code, including histogram of oriented gradients (HOG) for human detection, as well as methods such as parallelism, threads and multiprocessing, these techniques are based on Python programming language, which is the chosen software since it allows for the creation ofreadable codes, saving resources and time. After analyzing the different devices on the market,it was concluded that the implemented device is a child-friendly technological device, which measures physical status and is capable of alerting the person responsible for the infant about their body temperature. After the performance tests and the calculated errors such as the meanabsolute error (MAE) and the mean square error (RMSE), it was established that the device meets the necessary requirements. It is recommended that a machine learning algorithm be developed that is capable of determining the region of interest through thermal imaging, as thiswould reduce the hardware design with the RGB camera not essential.

**Keywords:** TECHNOLOGICAL DEVICE> <ARTIFICIAL VISION> <THERMAL CAMERA> <THERMOGRAPHY> <BODY STATE>.



## INTRODUCCION

En los últimos años el desarrollo de la visión artificial se ha incrementado, así como las diferentes aplicaciones en las que puede ser empleada, tratándose básicamente de capturar, procesar, reproducir imágenes tomadas desde una cámara y que con ayuda de un software y un conjunto de algoritmos se realiza el trabajo específico. Desde tiempos remotos cuando se realizaba las primeras tomas de imágenes fotográficas, estas han ayudado al avance dentro de la ciencia, medicina, entre otros campos, pero no fue sino en la década de los años 80 que el campo de visión artificial tomó gran impulso dentro de la sociedad con el adelanto de nuevas cámaras digitales y microprocesadores que son dispositivos básicos para la recolección y el procesamiento de datos dentro de los procedimientos que se requiera. (INFAIMON, 2020)

Herschel un astrónomo británico descubre la luz infrarroja él tenía una hipótesis de que cada color podía transmitir diferentes cantidades de calor, basado en su hipótesis y con ayuda de un termómetro de mercurio en el espectro alcanzando por un prisma de cristal nota que, aunque carecía de luz en la zona roja del espectro generaba más calor, de allí el nombre que en la actualidad se lo conoce como radiación infrarroja. (Global, 2018)

En la década de los años 90 la industria informática se posicionó dentro del mercado dejando así la posibilidad de que la termografía avance a grandes pasos en campos como el área civil, industria automotriz, detección de polución, detección de temperatura, control de procesos, etc. Una de las aplicaciones en la que contribuyó la radiación infrarroja fue la invención de misiles guiados. El avance tecnológico en cuanto a cámaras ha ido avanzando en el transcurso de los años como, por ejemplo, para el año de 1960 los avances tecnológicos en cuanto a cámaras no eran muy buenos, las cámaras eran demasiado grandes y pesadas para la generación de un buen equipo, no fue hasta el año de 1980 que aparecieron en el mercado cámaras más pequeñas y manejables que sirvieron para la creación, generación y avance de las demás aplicaciones ya existentes en la actualidad. (Melgosa, 2011)

Las cámaras termográficas se utilizan en muchas fábricas pesadas como fábricas de reciclaje de metales, obleas fábricas de producción y etc. para controlar la temperatura condiciones de las máquinas. Además, cámara termográfica se puede utilizar para detectar intrusos en entornos con condición de iluminación, por lo que, las cámaras digitales convencionales son menos aplicables. Igual que en el campo de la medicina las cámaras infrarrojas han sido utilizadas para medir la

temperatura corporal humana, puesto que revelan el cambio de la misma en tiempo real y con alta precisión. (Wong, et al., 2009)

El presente trabajo de titulación se refiere a la detección y control del estado febril de un menor de edad, utilizando dos cámaras una RGB y otra térmica capaz de detectar la radiación infrarroja y así determinar si el niño se encuentra en estado febril, basándose en un conjunto de algoritmos utilizando visión artificial, enviando una señal de alerta a la persona encargada del niño en el mismo instante que se haya detectado un aumento de temperatura, lo que le permite detectar a tiempo a sus padres algún caso de emergencia.

## CAPITULO I

### 1 DIAGNOSTICO DEL PROBLEMA

#### 1.1 Antecedentes

El síndrome febril se define como «la elevación térmica del cuerpo como una respuesta específica, mediada por el control central, ante una agresión determinada». (Gómez Toscano, 2018)

Se ha llegado al consenso internacional para considerar fiebre a la temperatura corporal central sobre 38°C. (Diakides, Mary, Bronzino, Joseph D, Peterson, 2018)

La fiebre es considerada una de las enfermedades con más casos de atención hospitalaria, ya sea por diferentes factores que hayan llegado a los centros de atención, dado que los pacientes que ingresan a diario al hospital exclusivamente no son por infección puede ser por tener un fondo inmunológico, alérgico o neoplásico, pero la preocupación de las personas al llegar a centros de salud para tratar estos casos es alarmante y angustiante sobre todo en el caso de los niños. (Cerón, et al., 2010)

Es muy normal la preocupación y malestar que genera en los padres y familiares cuando el niño se encuentra en estado febril, incluso para el personal médico porque tienen que lidiar con conductas inapropiadas que surgen entre los allegados al paciente llamado ocasionalmente “fiebre fobia” (Saad D., et al., 2006), por este motivo es preciso identificar el riesgo y el tipo de infección que tenga el paciente para que no llegue a estado más grave puesto que se considera que las infecciones virales constituyen el grueso de las fiebres infantiles y que los test rápidos disponibles para determinados agentes específicos( virus sincitial, influenza, adenovirus) resultan de gran utilidad en niños de corta edad donde la pobreza sintomática es la norma.(Sanz, F.J, Ramos, J.T, Pelayo, 2009)

Hasta el día de hoy los pediatras luchan constantemente contra la fiebre que es común y difícil de tratar por lo que se debe actuar de manera rápida y segura para bienestar del infante.

Durante el período 1980- 1997 se elaboraron diferentes criterios de puntuación de gravedad ideados para menores de 2-3 meses basados en signos clínicos y valoraciones cualitativas o semicuantitativas. Aunque resultan de cierta utilidad en el manejo decisorio inicial, ninguna de estas escalas ha presentado una validación significativa (Sanz, F.J, Ramos, J.T, Pelayo, 2009).

## **1.2 Delimitación del tema**

### **1.2.1 *Realidad del tema a nivel mundial***

A nivel mundial existen dispositivos comerciales para monitoreo de temperatura del niño y otros que incluyen cámaras móviles de video vigilancia panorámicas y receptoras de imagen y sonido, para ver y oír al bebé en cualquier lugar y en la oscuridad, entre los más comerciales están:

iTherm: Es un brazalete con sensor de temperatura que se enlaza a cualquier smartphone que cuente con iOS o Android por medio de Bluetooth 4.0 de baja energía. Utiliza una batería CR2032 que promete una vida útil de entre 12 y 18 meses.(Rosas, 2014)

TedCare: Es un dispositivo que se coloca en el tobillo del niño y permite controlar el ritmo cardiaco, oxigenación de sangre y temperatura corporal. Los datos recolectados son enviados mediante una aplicación que simula un semáforo y mediante el tono de color enviado se puede saber el estado del bebé. (El Empresario, 2018)

Existen varias investigaciones realizadas a nivel mundial para monitorizar la temperatura corporal de un niño. Un ejemplo de este tipo de sistemas es el denominado “Diseño de un sistema no invasivo de medición de la temperatura corporal interna”, el cual se llevó a cabo en la Universidad politécnica de Cataluña y tuvo como objetivo el crear un dispositivo de medición de temperatura de tipo no invasivo, que tomara las mediciones a través de un brazalete ubicado en la muñeca del paciente. (Lozano, 2015)

En este mismo sentido docentes de la Universidad de Londres desarrollaron el proyecto denominado “Sistema de detección temprana no invasiva y portátil de la fiebre para niños pequeños”, el cual tuvo como el monitorear la fiebre en niños menores de 2 años bajo un entorno no clínico, para ello se usaron varios sensores ubicados en diferentes áreas corporales.(Alarcón, A , Arias, G, Llanos, 2018)

### **1.2.2 Realidad del tema a nivel regional**

Con respecto al continente americano sobre todo en la región de América Latina las personas utilizan con mayor frecuencia para realizar mediciones de temperatura corporal el termómetro digital o de mercurio, en el mercado son altamente comercializados. (PETER MCCOLL C, KAREN COHEN SCH, FRANCISCA SOTO-AGUILAR B, 2013)

Entre los métodos más utilizados se encuentran el rectal, bucal, axilar y ótica cada uno de ellos es un procedimiento práctico y preciso pero el utilizado con mayor frecuencia para medición de temperatura es el rectal por su eficiencia. Se han empleado en los últimos años los termómetros digitales y electrónicos por facilidad, rapidez y exactitud de lectura e incluso por comodidad del paciente, la termometría timpánica es otra de las técnicas utilizadas la cual consiste en medir el grado de temperatura por el canal auditivo del paciente este procedimiento es muy acertado siempre que la técnica utilizada sea la correcta. (Cerón, et al., 2010)

### **1.2.3 Realidad del tema a nivel local**

A nivel nacional, la investigación “Diseño y construcción de un módulo inalámbrico para la adquisición y procesamiento en tiempo real, de las variables fisiológicas requeridas en atención prehospitalaria” realizada en la Escuela Politécnica Nacional, comprende la adquisición, procesamiento y análisis de las variables fisiológicas: temperatura, presión, frecuencia cardíaca y electrocardiograma, haciendo uso de sensores para luego registrar cada una de estas variables. (Erazo Chulde, y Vicente Romero, 2009)

En los estudios realizados y dispositivos comerciales revisados se encuentra el problema que todos deben estar en contacto con el infante para poder medir la temperatura, esto implica un riesgo para la salud del niño ya que estos dispositivos pueden tener alteraciones en su funcionamiento y causar algún tipo de daño al infante, por ejemplo, la batería de estos dispositivos se puede recalentar lo que provoca quemaduras, o puede fallar la lectura de la temperatura. También, es conocido que los niños a una temprana edad no tienen el hábito de utilizar cosas en su cuerpo como pulseras o parches, ya que estos afectarían la comodidad del niño al momento de dormir. Es por esto que el presente trabajo plantea diseñar y construir un dispositivo de monitoreo infrarrojo para supervisión de temperatura en niños menores a 3 años. El diseño del dispositivo permitirá monitorear la temperatura de un infante. Existen cámaras termográficas disponibles en el mercado, pero su alto precio no permite que estén al alcance de todos los usuarios, por eso, este dispositivo tratará de emular una. Para cumplir con este objetivo, se necesita utilizar equipos y materiales del menor tamaño posible que sean de fácil acceso y que no sobrepasen el precio de

una cámara térmica existente en el mercado. El dispositivo propuesto detectará cuando el niño tenga una temperatura elevada, y se enviará una alerta a los padres o adultos responsables a cargo del infante.

### **1.3 Objetivos**

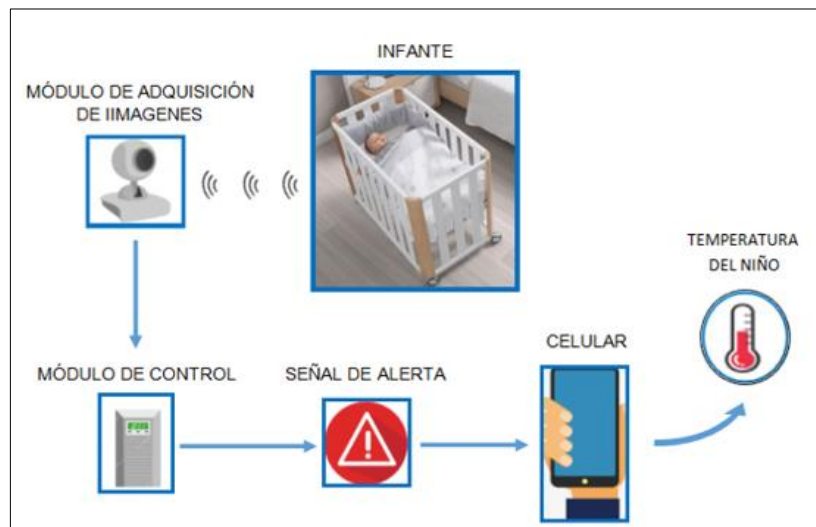
#### **1.3.1 *Objetivo General***

- Diseño y construcción de un dispositivo electrónico de monitoreo infrarrojo para supervisión de temperatura de niños menores a 3 años.

#### **1.3.2 *Objetivos Específicos***

- Investigar los tipos de dispositivos electrónicos de monitoreo infrarrojo para supervisión de temperatura que se han construido.
- Establecer los requerimientos que debe satisfacer el dispositivo electrónico de monitoreo infrarrojo.
- Seleccionar el hardware y el software más adecuados para diseñar el dispositivo electrónico de monitoreo infrarrojo
- Diseñar y construir el dispositivo electrónico que cumpla con los requerimientos establecidos.
- Comprobar si el dispositivo electrónico implementado cumple con todos los requerimientos planteados en esta investigación.

## 1.4 Justificación aplicativa



**Figura 1-1:** Funcionamiento del sistema.

Realizado por: Armijos, Soledad, 2020

El dispositivo monitorea la temperatura de un niño que se encuentra en una habitación. Con la ayuda de una cámara infrarroja la cual capta las imágenes para luego enviar estos datos a un controlador. Este realiza un procesamiento de imágenes empleando técnicas de visión artificial para detectar al niño en la habitación, así como también ejecuta la lectura de la temperatura para la supervisión del estado del infante. Los datos obtenidos son enviados mediante un mensaje al celular de los padres o del adulto responsable, dando así una señal de alerta.

Cabe recalcar que una señal de alerta es de suma importancia hoy en día, ya que, por diferentes factores como la falta de tiempo, trabajo excesivo, estrés por el ritmo de vida tan vida de la sociedad actual los padres de familia requieren un indicador de prevención para que sus hijos no sufran alguna molestia cada día y esto es precisamente uno de los objetivos que cumple el mecanismo presentado.

## CAPITULO II

### 2 FUNDAMENTOS TEORICO

#### 2.1 Termografía

##### 2.1.1 *Concepto*

La termografía es una técnica bidimensional utilizada en algunos campos de la investigación, ciencia y desarrollo, puesto que la industria informática ha evolucionada a pasos agigantados dentro del mercado industrial, la termografía infrarroja o también conocida como termografía pudo lograr la fácil detección y obtención de datos requeridos, proporcionados por una cámara termográfica en la que se puede reconocer el espectro de la imagen térmica llamada también termograma.

Se define también a la termografía como “*método de medición pasivo, sin contacto en la que cualquier objeto cuya temperatura sea superior al cero absoluto emite radiación infrarroja, invisible al ojo humano*”.(Testo, 2008)

##### 2.1.2 *Tipos de Termografía*

###### 2.1.2.1 *Termografía activa*

Mediante una fuente de generación infrarroja externa el calor emitido por la fuente alcanza el objeto en el que se presentan perturbaciones de calor sobre su superficie en la que alguna alteración interna puede provocar cambios bruscos en la temperatura del mismo, las mismas que serán analizadas para comprobar la condición en la que se encuentra el objeto expuesto al calor.(DUARTE, y PEÑA RODRÍGUEZ, 2011)

###### 2.1.2.2 *Termografía pasiva*

Al contrario de la termografía activa, esta no necesita de alguna estimulación externa el propio objeto se encarga de generar calor en el entorno donde se encuentre, permitiendo conductas en las que se pueda analizar el comportamiento térmico del objeto por la que cualquier defecto



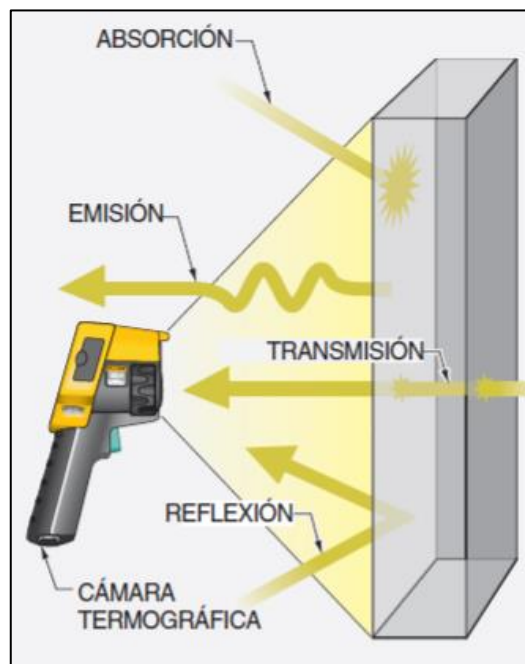
presentado se determina con una distribución anormal de temperaturas.(DUARTE, y PEÑA RODRÍGUEZ, 2011)

### 2.1.3 *Emisividad, Reflexión y Transmisión*

**Emisividad ( $\epsilon$ ):** El cuerpo en su superficie va a ser afectada su temperatura siempre en su superficie por este tipo de energía, esta se trata de toda la descarga de energía radiante. (Camacho Bedoya, et al., 2017)

**Reflexión ( $\rho$ ):** Es la medida de la capacidad de un objeto de reflejar radiación infrarroja, este tipo de energía dependerá siempre de algunos factores como la temperatura del objeto y el componente del que este hecho. (Testo, 2008, p.7)

**Transmisión ( $\tau$ ):** es la medida de la capacidad de un material de permitir el paso de la radiación infrarroja, este tipo de energía dependerá del modelo del material y que tan sea el mismo. (Testo, 2008, p.8)



**Figura 1-2:** Reflexión, Emisividad y Transmisión.

Fuente: Camacho, Bedoya; et al., 2017

## 2.2 Espectro Electromagnético

### 2.2.1 Definición

La guía de la radiación infrarroja define al espectro electromagnético como “*el rango de todo los tipos de radiación electromagnética clasificada por longitud de onda*”.(Melgosa, 2011)

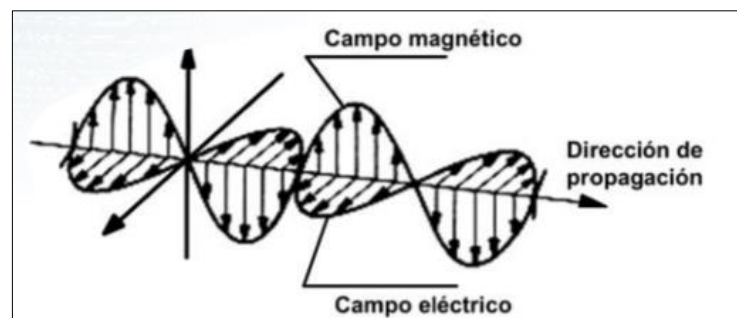
La radiación puede presentarse en distintas formas ya que es una fuente de energía saliente por ende el espectro electromagnético es el número de repeticiones de la radiación electromagnética de todas sus frecuencias (cantidad de repeticiones de un fenómeno por unidad de tiempo). (Luque Ordóñez, 2012)

### 2.3 Onda

La periodicidad de las perturbaciones a través del espacio en función del tiempo se denomina ondas electromagnéticas.

Las características que mas importantes de las ondas electromagnéticas son que su propagación puede cualquier medio es decir que no requieren un medio para difundirse incluso puede propagarse en el vacío.

Otra de las características principales es que se trasladan en tres dimensiones y que las perturbaciones se producen de forma perpendicular a la dirección en que se propaga. (Luque Ordóñez, 2012)



**Figura 2-2:** Onda Electromagnética

**Fuente:** Luque,Ordóñez. 2012

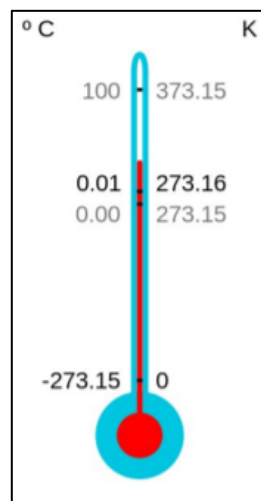
## 2.4 Calor y Temperatura

Aunque la mayor parte del tiempo las personas solemos relacionar la temperatura con el calor esto no es así ya que la percepción que sentimos normalmente en nuestro cuerpo u objetos se relaciona más bien con una sensación térmica que precisamente con una temperatura exacta.

**Calor:** es la cantidad de energía emitida por un cuerpo que siempre será transmitido por un cuerpo que contenga mayor cantidad de temperatura a otra con temperatura más baja hasta que los dos se encuentren en un perfecto equilibrio térmico. (Fernández Rodríguez, 2004)

**Temperatura:** *“Es una magnitud que nos da el "nivel" térmico de los cuerpos. Coincide con la sensación fisiológica de "caliente" o "frío". Cuando tocamos un cuerpo podemos hacer una estimación aproximada de su temperatura, de forma análoga a cómo podemos conocer el valor aproximado de una fuerza mediante el esfuerzo muscular”.* (Sauma, 2010)

Para realizar la medición de temperatura existen varios instrumentos que nos ayudan a detectar la variación de la mismo y poder analizar los datos para cualquier tipo de estudio requerido. Entre los muchos aparatos existentes está el termómetro como el de mercurio y el de alcohol los mismos en el que su funcionamiento se apoya en propiedades como la luz que emite el cuerpo o su conductividad eléctrica.



**Figura 3-2:** Escala de un termómetro.

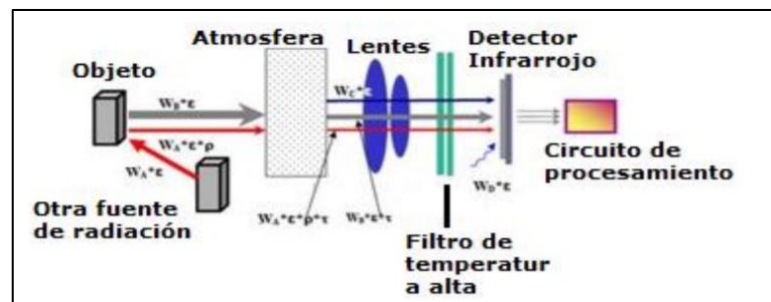
**Fuente:** Fernández, Rodríguez. 2004

## 2.5 Cámara Termográfica

### 2.5.1 Funcionamiento

El manejo de la cámara termográfica tiene que ser utilizada con cuidado ya que al igual que muchas cámaras comerciales estas poseen de botones y menús para ingresar a las distintas opciones que tiene para realizar un trabajo adecuado, así como también se derivan las distintas aplicaciones en las que la industria de la tecnología y la ciencia las está utilizando.

El funcionamiento se refiere a que la cámara termográfica primero detecta la radiación infrarroja emitida por el objeto, adquiriendo una respuesta (cambio de tensión o resistencia eléctrica) que será recibida por los elementos electrónicos que la cámara posee. La señal electrónica se convierte en una imagen electrónica la misma se verá manifestada como una imagen con distintos tonos de colores y dependiendo de ellos se analiza la magnitud de radiación infrarroja del objeto procedente (Melgosa, 2011).



**Figura 4-2:** Partes de una cámara termográfica.

Fuente: Neita; Peña. 2011

## 2.6 Imagen Radiométrica

la imagen radiométrica es una imagen térmica la cual contiene datos de temperatura de la imagen tomadas en todos los puntos de la figura correspondiente a un patrón.

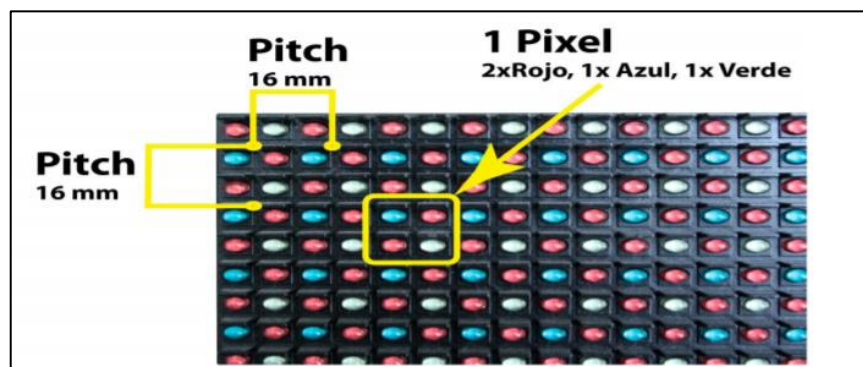
La imagen radiométrica se dice que es puntual con respecto a su temperatura porque no se considera el objeto a medir como un objeto aislado, al contrario estará bajo condiciones cambiantes, rodeado de otros objetos que le influyan, unas actuaciones, etc.(La Suma De Todos, 2011,p.20)

### 2.6.1 Resolución Radiométrica

La resolución radiométrica es considerada como el número de energía que se puede medir en escala radiométrica en el sensor. El sensor de la resolución recopila los datos de la imagen mediante un nivel en escala de grises que contiene la imagen y este indica la resolución de la misma. El parámetro “bit number” es el que se encarga de determinar la calidad radiométrica de la imagen. (IDEAM, 2008,p.19)

### 2.7 Pixel, FOV & IFOV

**Píxel:** Se define como la unidad mínima de color que contiene una imagen digital. La información que se obtiene en una imagen digitalizada está constituida por una matriz de píxeles las mismas que vemos en la pantalla de nuestro ordenados o dispositivo electrónico y quedan guardados en los sistemas de almacenamiento de estos. (Destruels, 2015, p.2)



**Figura 5-2:** Representación de la matriz de un píxel.

**Fuente:** Destruels, 2015

**FOV:** FOV o también denominado campo de visión, *ángulo horizontal visible a través de una lente de infrarrojos.* (Melgosa, 2011, p.38)

**IFOV:** IFOV o también llamado campo de visión instantánea *Medida de resolución geométrica de una cámara de infrarrojo.* (Melgosa, 2011,p.38)

## **2.8 Visión Artificial**

### **2.8.1 Definición**

El objetivo principal de la visión por computadora (también llamada visión de máquina) es reconstruir e interpretar escenas naturales basándose en el contenido de imágenes capturadas por varias cámaras. Los sistemas de visión por computadora incluyen cosas como satélites de levantamiento, sistemas de navegación robóticos, escáneres inteligentes y sistemas de detección remota. La visión por computadora, se centra en extraer información útil de las imágenes.(Peters, 2017)

Los sistemas de visión por computadora suelen emular la percepción visual humana. El hardware de elección en los sistemas de visión por computadora es alguna forma de cámara digital, programada para aproximarse a la percepción visual. Por tanto, existen estrechos vínculos entre la visión por ordenador, el procesamiento de imágenes digitales, la óptica, la fotometría y la fotónica.(Peters, 2017)

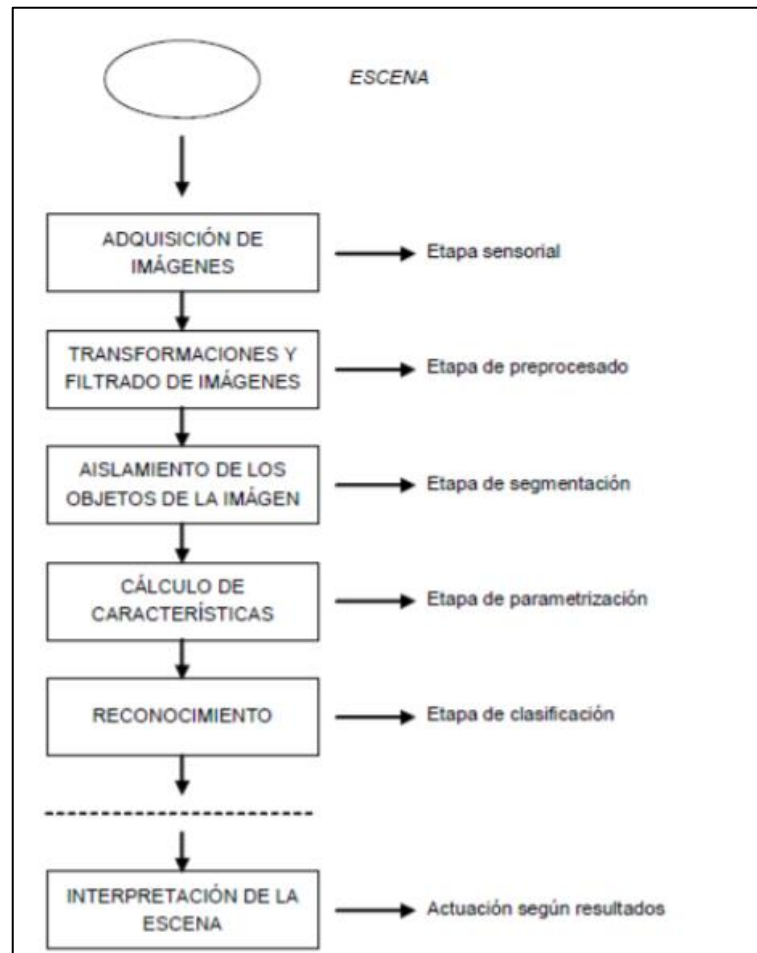
### **2.8.2 Etapas de un sistema de visión artificial**

**Captura:** Es la etapa en que se adquiere la imagen, es una etapa decisiva para el resto de las demás fases puesto que si no se tiene una imagen lo suficientemente bien calibrada por la cámara el proceso para la obtención de los datos será complicado e inestable.

**Preprocesamiento:** Normalmente las imágenes digitales suelen estar acompañadas por factores que no ayudan a tener una visualización o a que la obtención de la misma sea la adecuada por ejemplo el ruido, pérdida de definición de la imagen o por motivos que tienen que ver también con el enfoque de la cámara que se esté usando, entonces el preprocesado lo que hace es solucionar las fallas que son producidas por el dispositivo que se esté utilizando, empleando algoritmos que permitan que la imagen sea lo más exacta posible.(Alejandro, et al., 2009)

**Segmentación:** Se encarga de dividir una imagen en objetos, separando los elementos que contiene cada una de ellas para poder saber la posición fija correspondiente con el fondo que tiene la imagen.

**Reconocimiento:** Resueltas cada una de las etapas anteriores se procede a la etapa de reconocimiento que consiste en determinar ciertos parámetros que definen al objeto como: color, textura, grosor, etc. y se decide si la imagen tratada es la correcta. (Alejandro, et al., 2009)



**Gráfico 1-2:** Diagrama de bloques de las etapas de un sistema de V.A.

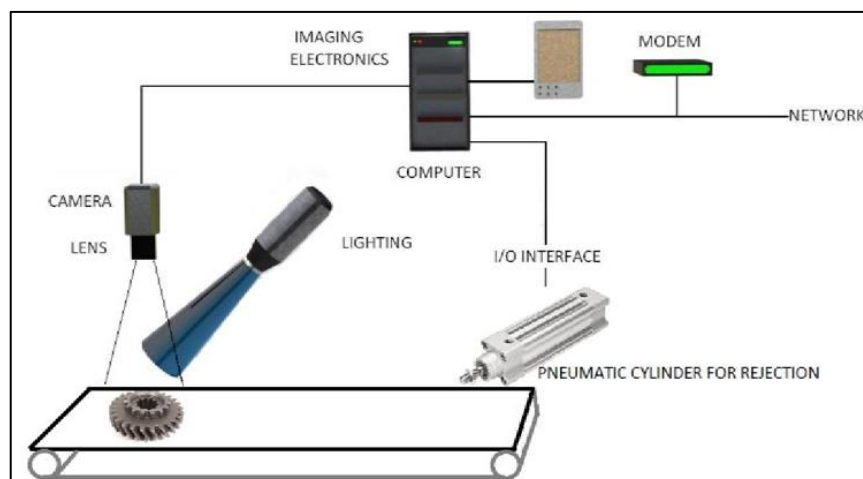
**Fuente:** Alejandro, et al., 2009

### 2.8.3 Aplicaciones

La visión juega un papel fundamental para los seres vivos al permitirles interactuar con el medio ambiente de forma eficaz y eficiente. El objetivo de la visión artificial es dotar a los dispositivos de computación, y más generalmente a los sistemas artificiales, con capacidades para hacer frente a situaciones predeterminadas no a priori. Para tal fin, tenemos que tener en cuenta las limitaciones informáticas de las arquitecturas de alojamiento y las especificaciones de las tareas a realizar. Estos elementos conducen a adaptación y optimización continua de las técnicas habituales de procesamiento visual, como los desarrollados en Visión por Computador y Procesamiento de Imágenes. El desarrollo rápido de procesadores y dispositivos informáticos disponibles para el público requieren una potencia computacional grande y de bajo costo. (Solari, Fabio, Chessa, Manuela, Sabatini, 2012)

Los estudios de visión artificial abren una gran oportunidad para diferentes dominios como manufactura, agricultura, acuicultura, investigación médica, también estudios de investigación y aplicaciones para una mejor comprensión de sus procesos y operaciones. Como científicos, los esfuerzos se han dirigido y siguen dirigidos hacia la comprensión materiales, sistemas y clases específicas de tipos de frutas, o diagnóstico de pacientes a través de clasificación de imágenes y análisis, y también detección e inspección en tiempo real de piezas defectuosas.(Zohdy., Mahmood, Mahmood A., Ramadan, Nagy, y Hefny, 2018)

Los investigadores en ciencias de los materiales lograron hallazgos muy importantes en la comprensión y analizar imágenes microestructurales, ya que el objetivo es producir un método común para extraer características relativas a las micrografías. Además, el objetivo principal del análisis de imágenes es extraer datos significativos incluidos en las imágenes que se van a analizar, el análisis de imágenes contiene muchos procesos, incluido el registro de imágenes y fusión de imágenes, el registro de imágenes utiliza dos imágenes, la primera se llama imagen de referencia, mientras que lo segundo, llamado imagen detectada, la imagen detectada se alinea con la imagen de referencia, el resultado o la salida de este proceso se utiliza en otro proceso que se llama fusión de imágenes, la fusión de imágenes es el proceso de combinando los datos contenidos en dos imágenes en una imagen, ya que la imagen resultante será más informativa que las imágenes de entrada, los procesos mencionados, el registro de imágenes y la fusión de imágenes son muy útiles en muchos dominios prácticos y de aplicación, como industrial, médico, acuicultura y agricultura y otros áreas que utilizan tecnologías y aplicaciones de visión artificial.(Zohdy., Mahmood, Mahmood A., Ramadan, Nagy, y Hefny, 2018)



**Figura 6-2:** Concepto de funcionamiento de un proceso utilizando visión artificial.

Fuente: Sathiyamoorthy, 2014



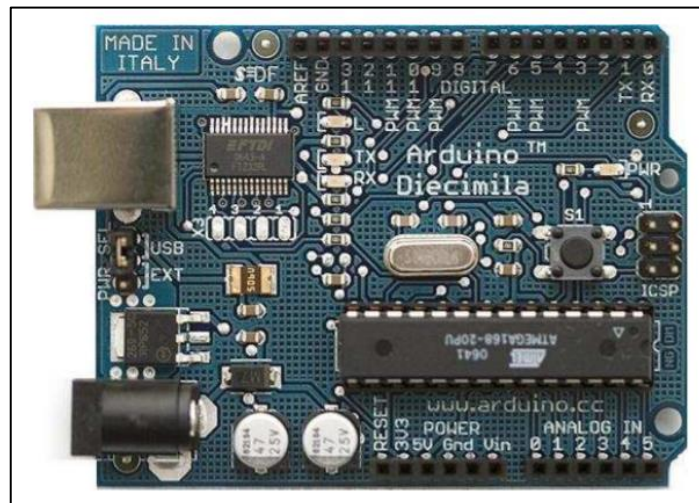
## 2.9 Microcontroladores

Un microcontrolador contiene los componentes principales de la computadora: procesador, memoria de programas y datos, interfaces de entrada y salida. Por lo tanto, puede denominarse computadora de un solo chip. El término “Microcontrolador” dice que este dispositivo está desarrollado para controlar objetos y procesos.(Baškys, 2012)

### 2.9.1 Arduino

Arduino es una plataforma prototipo (de código abierto) basada en un hardware y software. Consiste en una placa de circuito, que se puede programar (denominada un microcontrolador) y un software listo para usar llamado Arduino IDE (Integrated Development Environment), que se utiliza para escribir y cargar el código de la computadora en la placa física.(Tutorialspoint, 2019)

Esta pequeña placa de Arduino se puede conectar a la computadora y una serie de enchufes de conexión que se pueden conectar a electrónica como motores, relés, sensores de luz, diodos láser, altavoces, micrófonos, y más. Se alimenta a través de conexión USB desde la computadora, de una batería de 9V, o de una fuente de alimentación. Pueden controlarse desde la computadora o programarlo desde la computadora y luego desconectarlo permitiendo que funcione de forma independiente.(Monk, 2011)



**Figura 7-2:** Tarjeta Arduino.

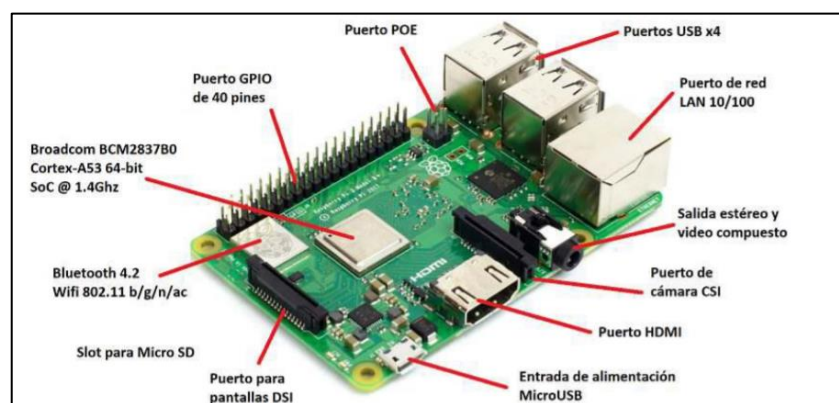
**Fuente:** Ruiz Guti, 2007

## 2.9.2 Raspberry

La Raspberry Pi se conoce como una computadora de placa única, lo que significa exactamente lo que suena como: es una computadora, como una computadora de escritorio, una computadora portátil o un teléfono inteligente, pero construida en una sola placa de circuito. Como la mayoría de las computadoras de placa única, la Raspberry Pi es pequeña, aproximadamente lo mismo huella como una tarjeta de crédito, pero eso no significa que no sea potente: una Raspberry Pi puede hacer cualquier cosa que pueda hacer una computadora más grande y que consume más energía, aunque no necesariamente tan rápido. (Halfacree, 2018)

La placa Raspberry Pi contiene un procesador ARM basado en Broadcom, un chip gráfico, RAM, GPIO y otros conectores para dispositivos externos. El procedimiento operativo de Raspberry Pi es muy similar en comparación con la PC y requiere más hardware como teclado, mouse, unidad de visualización, fuente de alimentación, tarjeta SD con sistema operativo instalado (actuando como disco duro) para operación. Raspberry Pi también facilita puertos USB, Ethernet para Internet / Red-Conectividad de igual a igual. (Nayyar, y Puri, 2015)

Como cualquier otra computadora, donde el sistema operativo actúa como columna vertebral para el funcionamiento. Raspberry Pi, facilita el código abierto sistema operativo basado en Linux. Hasta la fecha, se están utilizando más de 30 sistemas operativos basados en diferentes versiones de Linux lanzado. La fundación Raspberry Pi también ha lanzado varios accesorios como Camera, Gertboard y Compute Model Kit para implementar módulos de hardware complementarios. (Nayyar, y Puri, 2015)



**Figura 8-2:** Esquema de conexión de Raspberry pi.

Fuente: Agora, 2009

## 2.10 Cámaras

Una cámara es un dispositivo que consta de una apertura provista de una lente y un obturador a través del cual la imagen de un objeto se proyecta sobre una superficie para grabar (como en una película fotosensible o un sensor electrónico) o para traducirla en impulsos eléctricos (Merriam-Webster Dictionary, 2018)

### 2.10.1 Cámara Web Cam

Una cámara web es una pequeña cámara de video digital que se conecta a una computadora. También se conoce como cámara web que puede capturar imágenes o videos en movimiento. Estas cámaras vienen con un software que debe instalarse en la computadora y que ayuda a transmitir su video en Internet en tiempo real. Tiene la capacidad de tomar fotografías, incluidos videos HD, pero su calidad de video puede ser menor en comparación con otros modelos de cámaras (JavaTPoint, 2011).



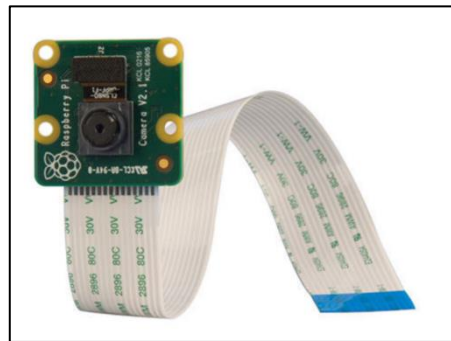
**Figura 9-2:** Cámara Web Modelo Logitech B910 HD.

Fuente: Logitech, 2016

### 2.10.2 Cámara Raspberry Pi

La versión actual del módulo de cámara Raspberry Pi, conocido como módulo "v2" o "Versión 2.1", se basa en un sensor de imagen Sony IMX219 de alta calidad, el mismo tipo de sensor que puede encontrar en la parte posterior de su teléfono inteligente o tableta. Este es un sensor de 8 megapíxeles, que significa que puede tomar fotografías con hasta 8 millones de píxeles. Lo hace

capturando imágenes hasta 3280 píxeles de ancho por 2464 de alto. Además de imágenes fijas, el módulo de la cámara puede capturar secuencias de video con resolución Full HD la misma resolución que la mayoría de los televisores - a una velocidad de 30 cuadros por segundo (30 fps). Para más suave movimiento o incluso para crear un efecto de cámara lenta, la cámara se puede configurar para capturar a una mayor velocidad de fotogramas al reducir la resolución: 60 fps para secuencias de video de 720p y hasta 90 fps para 480p o resolución "VGA". (Halfacree, 2018)

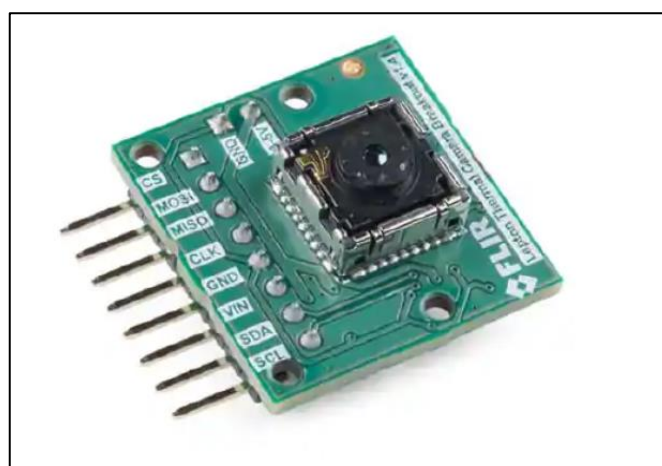


**Figura 10-2:** Cámara Raspberry Pi.

Fuente: Matt, 2016

### 2.10.3 Cámara Flir Lepton 2.5

La cámara Flir Lepton es un infrarrojo de onda larga completo (LWIR) módulo de cámara diseñado para interfaz fácilmente en interfaces nativas de dispositivos móviles y otros productos electrónicos de consumo. Captura infrarrojos entrada de radiación en su respuesta nominal banda de longitud de onda (de 8 a 14 micrones) y emite una imagen térmica uniforme con radiométría para proporcionar una imagen de temperatura con mediciones. (FLIR, 2018)



**Figura 11-2:** Cámara Flir Lepton 2.5

Fuente: SparkFun Electronics, 2018

Las características técnicas de la cámara térmica Flir Lepton 2.5 se detallan a continuación en la tabla 1-2.

**Tabla 1-2:** Características técnicas de la cámara Flir Lepton 2.5

<b>Visión de conjunto</b>	
Tecnología de sensores	Microbolómetro VOx no refrigerado
Rango espectral	Infrarrojo de onda larga, 8 $\mu\text{m}$ a 14 $\mu\text{m}$
Formato de matriz	80 $\times$ 60, escaneo progresivo
Tamaño de píxel	17 $\mu\text{m}$
Velocidad de fotogramas efectiva	8,6 Hz (aplicación comercial exportable)
Sensibilidad térmica	<50 mK (0.050 ° C)
Compensación de temperatura	Automático. Imagen de salida independiente de la temperatura de la cámara.
Rango dinámico de escena	0° a 120°C
Precisión radiométrica	N / A
FOV - horizontal	51 °
FOV - diagonal	63.5 °
Formato de salida	Seleccionable por el usuario de 14 bits, 8 bits (AGC aplicado) o 24 bits RGB (AGC y color aplicado)
Protección solar	Integral
<b>Eléctricas</b>	
Reloj de entrada	Niveles de tensión IOS CMOS de 25 MHz nominales
Interfaz de datos de video	Video sobre SPI
Puerto de control	CCI (I2C-like), CMOS IO Voltage Levels
Voltaje de suministro de entrada (nominal)	2.8 V, 1.2 V, 2.5 V a 3.1 V IO
Disipación de potencia (Típico, Ta)	(Típico, Ta) 150 mW (funcionamiento), 650 mW (durante el evento del obturador), 4 mW (modo de espera)
<b>Mecánicas</b>	
Dimensiones del paquete: versión de socket (w x l x h)	10.5 x 11.7 x 6.4 mm

Peso	0.9 gramos
<b>Ambiental</b>	
Rango de temperatura para un óptimo funcionamiento	-10 ° C a +65 ° C
Rango de temperatura no operativo	-40 ° C a +80 ° C
Shock	1500 G @ 0,4 ms

Fuente: Lepton, 2017

## 2.11 Software

Es un término genérico para organizar colecciones de datos e instrucciones de computadora, a menudo dividido en dos categorías principales: software del sistema que proporciona las funciones básicas no específicas de la tarea de la computadora, y software de aplicación que es utilizado por los usuarios para realizar tareas específicas (Eddy Tukamushaba Kurobuza, y Godfrey Ahumuza Ikaanga, 2008).

### 2.11.1 *Phyton*

Python es un potente lenguaje de programación fácil de aprender. Tiene estructuras de datos eficientes de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos. La elegante sintaxis y dinámica de Python la mecanografía, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para la creación de scripts y la aplicación rápida desarrollo en muchas áreas en la mayoría de las plataformas. (van Rossum, y Python Development Team, 2018)

A continuación, se describe características importantes del lenguaje de programación Python.

#### 2.11.1.1 *Calidad de Software*

Para muchos, el enfoque de Python en la legibilidad, la coherencia y la calidad del software en general lo distingue de otras herramientas en el mundo de las secuencias de comandos. El código Python está diseñado para ser legible y, por tanto, reutilizable y mantenible, mucho más que los tradicionales lenguajes de secuencias de comandos. Python tiene un soporte profundo para más avanzados mecanismos de reutilización de software, como la programación orientada a objetos. (Mark, 2009)

### 2.11.1.2 *Productividad del desarrollador*

Python aumenta la productividad del desarrollador muchas veces más allá de compilado o estáticamente lenguajes escritos como C, C ++ y Java. El código Python suele ser de un tercio a un quinto del tamaño del código C ++ o Java equivalente. Eso significa que hay menos para escribir, 3 menos para depurar y menos para mantener después del hecho. Los programas de Python también se ejecutan de inmediato, sin los largos pasos de compilación y enlace requeridos por otras herramientas, aumentando aún más la velocidad del programador(Mark, 2009).

### 2.11.1.3 *Portabilidad del programa*

La mayoría de los programas de Python se ejecutan sin cambios en las principales plataformas informáticas. Portabilidad el código Python entre Linux y Windows, por ejemplo, suele ser solo una cuestión de copiar el código de un script entre máquinas. Además, Python ofrece múltiples opciones para codificar interfaces de usuario gráficas portátiles, programas de acceso a bases de datos, sistemas basados en web y más. Incluso las interfaces del sistema operativo, incluido el programa lanzamientos y procesamiento de directorios, son tan portátiles en Python como sea posible ser (Mark, 2009).

### 2.11.1.4 *Bibliotecas de apoyo*

Python viene con una gran colección de funcionalidades portátiles y prediseñadas, conocidas como biblioteca estándar. Esta biblioteca admite una variedad de tareas de programación a nivel de aplicación, desde la coincidencia de patrones de texto hasta la creación de scripts de red. Además, Python se puede ampliar tanto con bibliotecas propias como con una vasta colección de software de soporte de aplicaciones de terceros. El dominio de terceros de Python ofrece herramientas para la construcción de sitios web, programación numérica, acceso a puerto serie, desarrollo de juegos y mucho más. La extensión NumPy, por ejemplo, se ha descrito como un equivalente gratuito y más potente de la programación numérica de Matlab.(Mark, 2009)

## 2.11.2 *Matlab*

Matlab es un lenguaje de programación desarrollado por MathWorks. Comenzó como un lenguaje de programación matricial donde la programación de álgebra lineal era simple(Colgren, 2007).

Es una programación de alto nivel de cuarta generación lenguaje y entorno interactivo para cálculo numérico, visualización y programación. Permite manipulaciones matriciales; trazado de funciones y datos; implementación de algoritmos; creación de interfaces de usuario; interactuar con programas escritos en otros lenguajes, incluidos C, C ++, Java y Fortran; analizar datos; desarrollar algoritmos; y crear modelos y aplicaciones. Tiene numerosos comandos integrados y funciones matemáticas que lo ayudan a cálculos matemáticos, generación de gráficos y ejecución de métodos numéricos(Colgren, 2007).



## **CAPITULO III**

### **3 MARCO METODOLOGICO**

Dentro del marco metodológico se presenta los métodos que se utilizó para efectuar el trabajo de titulación mencionado, también las distintas técnicas empleadas para el desarrollo del prototipo.

#### **3.1 Proceso de Investigación**

La metodología que se utilizó para la investigación del proyecto de titulación planteado es la experimental ya que se trabajó en cada objetivo propuesto, partiendo primero de la problemática que se analizó en cada uno de ellos y así llegar a cumplir con los requerimientos planteados. El modo bibliográfico se utilizó para la recopilación de toda la información necesaria para tratar temas como: termografía, espectro electromagnético, calor y temperatura, imagen radiométrica, microcontroladores, etc., donde toda esa información se obtuvo de fuentes como tesis, artículos científicos, revistas, etc.

#### **3.2 Elección de elementos para el hardware y software a utilizar**

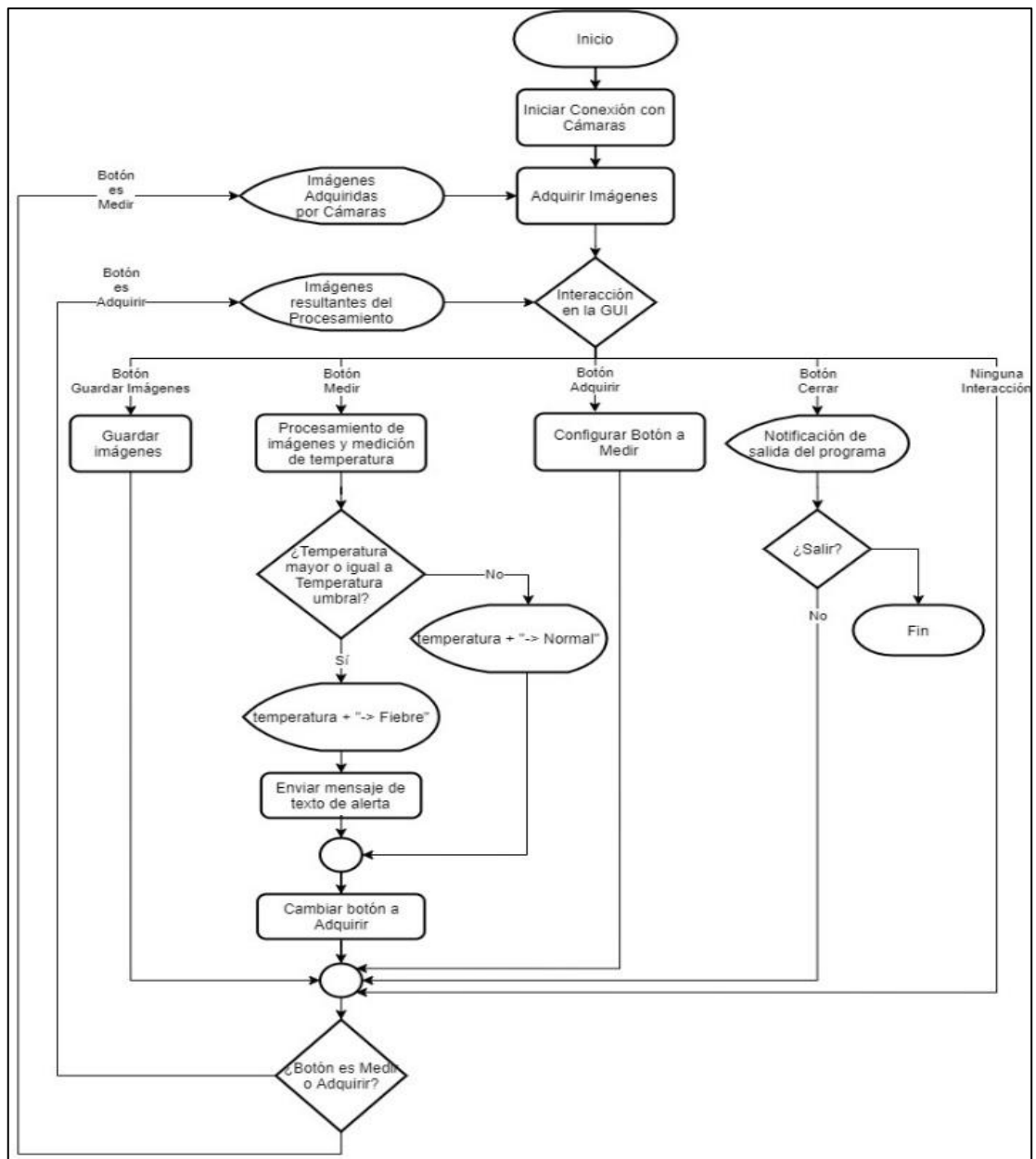
De acuerdo a los requerimientos planteados y los datos obtenidos se decidió elegir algunos elementos para la construcción y el diseño del dispositivo tomando como principales características: la compatibilidad que existen entre ellos, el costo y la información que tienen dentro del mercado y la capacidad de procesamiento.

Para ello en los elementos del hardware se utilizó el microcontrolador raspberry pi 3+, así como también dos cámaras para la adquisición de imagen que corresponden a la cámara de la raspberry pi y la segunda cámara que es la térmica Flir Lepton 2.5, en cuanto al software se realizó la programación con el lenguaje Python 3.7, en el que también es programado el mensaje de texto que realiza la función de señal de alerta a las personas encargadas del niño.

### **3.3 Diagrama de funcionamiento del Trabajo de Titulación**

#### **3.3.1 *Diagrama General***

El gráfico 1-3 muestra el funcionamiento general del dispositivo electrónico del trabajo de titulación mediante un diagrama de su funcionamiento general, el cual inició con el encendido de las dos cámaras para la adquisición de imágenes, al mismo tiempo que se puede visualizar en la pantalla del computador la interfaz gráfica del usuario que tiene los botones de medir o adquirir temperatura, pues debido a la programación que se realizó dentro de cada uno de ellos, se detectó si la temperatura es normal o mayor al umbral propuesto, y si es mayor este envía un mensaje de texto al número de la persona encargada del infante como mensaje de alerta. En la GUI se programó también un botón de guardar imágenes y el botón salir, el primero será muy útil para cualquier otro tipo de análisis que se requiera hacer después y el segundo al momento de seleccionarlo este indica una notificación de salida de emergencia al usuario y dependiendo de su respuesta sigue dentro del programa o lo da por concluido. Todo el funcionamiento del sistema se detalla con más precisión en el tema de desarrollo del software de control propuesto dentro de este capítulo.



**Gráfico 1-3:** Diagrama de flujo del funcionamiento general de la interfaz gráfica.

Realizado por: Armijos Soledad,2020

### 3.4 Elementos de Hardware del prototipo

#### 3.4.1 Raspberry Pi

Este microcontrolador se encargó del control y procesamiento del software y hardware del dispositivo propuesto, por este motivo se realizó una tabla comparativa de los distintos modelos que existen hoy en día dentro del mercado industrial como se muestra en la tabla 1-3.

**Tabla 1-3:** Comparación entre los atributos de los modelos de Raspberry

ATRIBUTOS	MODELOS			
	Raspberry Pi 1	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi 3+
<b>Velocidad</b>	700 MHz	900 MHz	1200MHz	1.4-GHz, 4-core
<b>SOC</b>	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837B0
<b>CPU</b>	ARM1176JZF-S	Cortex-A7	ARM cortex A-53	(Cortex-A53)
<b>RAM</b>	512 MB	1GB		1GB
<b>Conectividad Inalámbrica</b>	No	No	Bluetooth 4.1	802.11ac, Bluetooth 4.2, Ethernet
<b>Video</b>	HDMI 1.4	HDMI 1.4	HDMI 1.4	HDMI 1.4
<b>Red</b>	No	10/100 Ethernet vía hub USB	WiFi 802.11n	WiFi 802.11n
<b>Almacenamiento</b>	Micro SD	Micro SD	Micro SD	Micro SD
<b>Alimentación</b>	5 V a 2A micro USB	5 V a 2A micro USB	5 V a 3A microUSB	5 V a 2.5A (#) microUSB

Realizado por: Armijos Soledad,2020

Considerando todas las características mencionadas se decidió elegir el modelo Raspberry Pi 3 ya que debido a sus atributos es el más apto para realizar la etapa de control del dispositivo electrónico, como se observa en la tabla 1-3.

Se utilizó la Raspberry Pi 3 porque posee los recursos necesarios que nos brinda la facilidad para desarrollar un entorno gráfico donde se tenga una mejor interacción por parte del usuario para el uso del prototipo como se muestra en la figura 1-3, ya que sus especificaciones técnicas son suficientes para que nos permita obtener una imagen clara del niño.



**Figura 1-3:** Raspberry Pi 3.

Realizado por: Armijos Soledad,2020

### 3.4.2 Elección de la cámara para el prototipo

En la tabla 2-3 se presenta las características de las diferentes cámaras existentes en el mercado compatibles con el microcontrolador a utilizar.

**Tabla 2-3:** Comparación entre modelos de cámaras compatibles con Raspberry Pi 3

<b>CÁMARAS</b>		
<b>MODELOS</b>	<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
<b>MÓDULO DE CÁMARA RASPBERRY PI</b>	Compatibilidad con Raspberry Pi  Resolución de imagen y captura de video de 1080p30, 720p60 y 640x480p90	La distancia que se puede utilizar el módulo es máximo 30 cm esto se debe al tamaño de su bus de conexión.
<b>BRIO ULTRA HD PRO WEBCAM</b>	Videoconferencias 4K Ultra HD (hasta 4096 x 2160 píxeles a 30 fps)  Compatible con raspberry	Trabaja a resoluciones que no son compatibles con raspberry  Es necesario redimensionar las imágenes que genera para utilizarlas.
<b>MLX90640</b>	se comunica a través de la interfaz I2C  Es compatible con voltaje de funcionamiento de 3.3 V/5 V, soporta plataformas host como Raspberry Pi/Arduino (ESP32)/STM32	55° de campo de visión, de imagen de arreglo de infrarrojos cámara de imagen térmica de 32 x 24 píxeles
<b>MÓDULO CÁMARA TERMICA FLIR LEPTON 2.5</b>	Interfaces de vídeo SPI Utiliza fuentes de alimentación estándar compatibles con teléfonos móviles  Mayor sensibilidad que las matrices de termopila comunes  Bajo consumo operativo: 150 mW típica, 650 mW durante la obturación	

Realizado por: Armijos Soledad,2020

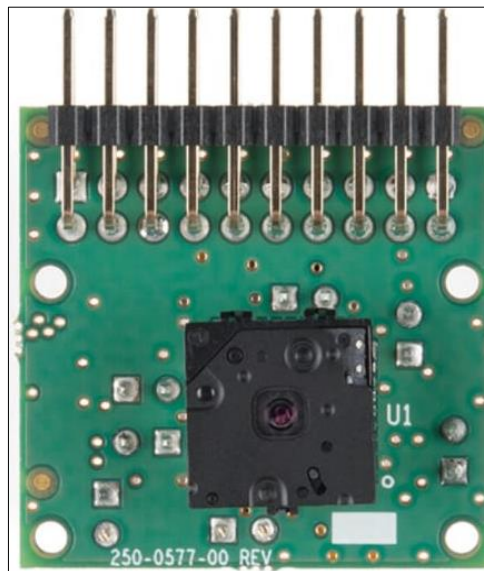
Fundamentándose en la tabla 2-3 se optó por utilizar dos cámaras, que son el módulo de cámara raspberry para hacer capturar la imagen RGB, por su compatibilidad con la raspberry y por el bajo costo dentro del mercado y el módulo de cámara térmica Flir Lepton 2.5 que se encarga de

realizar la toma de imagen térmica, ya que esta capta un amplio espectro del infrarrojo capturando una radiación infrarroja cuya longitud de onda varía entre los 8 y 14 microns, además de que usa dos buses de comunicación serial, una para el control de la cámara implementado por el bus I<sup>2</sup>C a través del cual se puede activar o desactivar algunas características y el segundo bus es el SPI que es el de salida, estos componentes son elegidos de acuerdo a las características ofrecidas al momento de trabajar con la Raspberry Pi, la cámara del módulo de la raspberry Pi como la cámara Flir Lepton 2.5 se muestran en la figura 2-3 y figura 3-3 respectivamente.



**Figura 2-3:** Cámara para Raspberry Pi 3.

**Realizado por:** Armijos Soledad,2020

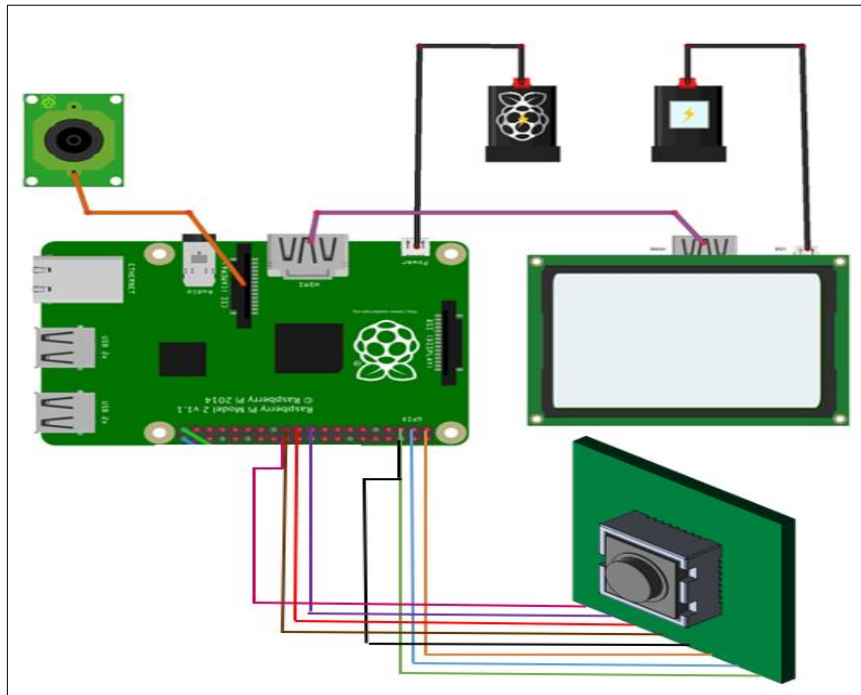


**Figura 3-3:** Cámara térmica Flir Lepton 2.5.

**Realizado por:** Armijos Soledad,2020

### 3.4.3 Conexiones del prototipo

En la figura 4-3 se muestra la conexión de los elementos electrónicos que conformaron el dispositivo final.



**Figura 4-3:** Conexión de los dispositivos del prototipo.

**Realizado por:** Armijos Soledad,2020

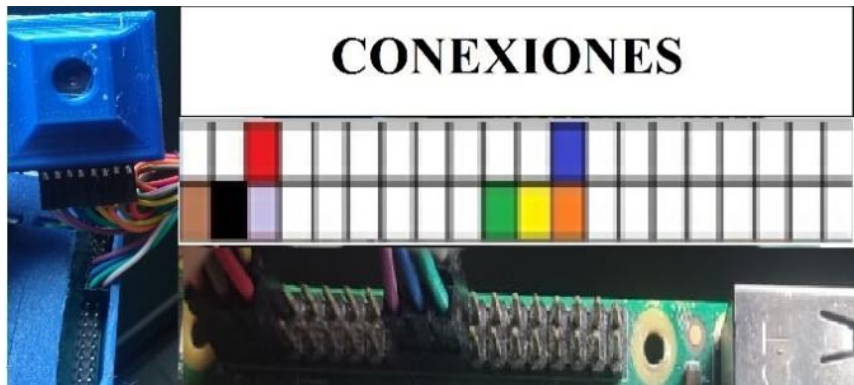
El circuito se formó por la conexión de las dos cámaras, la cámara RGB y la cámara Flir Lepton 2.5 hacia la raspberry puesto que estos elementos electrónicos poseen las características de optimización y compatibilidad, conectando cada uno de los pines de las cámaras a la raspberry para luego desarrollar el algoritmo adecuado y así el dispositivo logre cumplir con los requerimientos propuestos.

Para el funcionamiento del prototipo se realizó la conexión de sus dos cámaras. La cámara de la raspberry pi 3 que trabaja en conjunto de sus interfaces que son I<sup>2</sup>C y SPI, también se realizó la conexión de la cámara térmica Flir Lepton 2.5 directamente a los pines de la raspberry pi 3 como se indica en la figura 5-3 y la tabla 3-3.

**Tabla 3-3:** Conexión Cámara Flir Lepton con Raspberry Pi.

NOMBRE PIN	PIN RASPBERRY	PIN CÁMARA
SCL	5	SCL
SDA	3	SDA
3V3	1	VIN
GND	6	GND
CLK	23	CLK
MISO	21	MISO
MOSI	19	MOSI
CE00	24	CS

Realizado por: Armijos, Soledad, 2020



**Figura 5-3:** Conexión de la Cámara Flir Lepton a Raspberry Pi 3.

Realizado por: Armijos Soledad,2020

### 3.5 Elementos del software del prototipo

#### 3.5.1 *Análisis de características entre el algoritmo utilizado para reconocimiento facial con los existentes*

En la tabla 4-3 se muestra los diferentes algoritmos investigados existentes para la detección de rostros y objeto, realizando un enfoque entre las ventajas y desventajas de cada uno para obtener la elección del algoritmo que cumpla satisfactoriamente con el objetivo del dispositivo.



**Tabla 4-3:** Ventajas y desventajas entre los algoritmos de reconocimiento facial y objetos.

<b>ALGORITMOS PARA DETECCIÓN DE ROSTROS Y OBJETOS</b>	
<b>ALGORITMOS</b>	<b>CARACTERÍSTICAS</b>
<b>SIFT (SCALE INVARIANT FEATURE TRANSFORM)</b>	Las características de este algoritmo son invariantes ante cambios de rotación, escala, corrimiento, y parcialmente invariantes a cambios de iluminación y perspectiva 3D
<b>HISTOGRAMA DE GRADIENTES ORIENTADOS HOG</b>	Se basa en la afirmación de que la apariencia local de un objeto puede caracterizarse bastante bien mediante la distribución de los gradientes locales de intensidad o las direcciones de los bordes en la imagen.
<b>SURF (SPEEDED-UP ROBUST FEATURES)</b>	El porcentaje en acierto en reconocimiento es muy elevado. Se puede tomar como partida un fichero de pesos ya creado. Al ser un tipo de red neuronal convolucional presenta las mismas desventajas que el anterior
<b>LBP (LOCAL BINARY PATTERN)</b>	El operador describe cada pixel respecto a los niveles de grises relativos de los pixeles contiguos; si el nivel de gris del pixel vecino es menor que el del pixel de referencia, el valor se establece en 0, mientras que, si es mayor o igual, el valor será de 1.

Fuente: Alcántara-Montiel, et al., 2019

Al analizar las características de cada algoritmo de detección de rostros y objetos se optó por desarrollar el programa con el algoritmo basado en el histograma de gradientes orientados puesto que tiene un bajo costo en procesamiento computacional y tiene un muy bajo porcentaje de errores de clasificación, detectando consistentemente la zona específica del rostro que se requiere.

### **3.5.2 Python**

La programación para el control y detección de temperatura y sistema de alerta se realizó en el lenguaje de programación Python, instalando las librerías necesarias para la ejecución del programa, entre ellas las más importantes que son para la detección de imágenes utilizando las cámaras, denominada Opencv, y la librería Tkinter que se utilizó para la creación de la respectiva GUI del sistema, como se muestra en la figura 6-3.

```

25 from PIL import ImageTk
26 import multiprocessing
27 import threading
28 from pylepton import Lepton
29 import cmapy
30 import nexmo
31
32 client = nexmo.Client(key='d679d248', secret='EwrFk1y69Sjct8dMq')
33
34 libc.conectarFLIR()
35 libc.desactivarTLInear()
36 libc.desactivarAGC()
37
38 B = 1428
39 F = 1
40 R = 364201.85
41 O = -29.07
42 # ~ np.set_printoptions(threshold=sys.maxsize)
43 -----
44 #Creación de clases
45 -----
46 class InterfazGrafica():
47
48     # Constructor:
49     def __init__(self):
50         self.imagenAncho = 400

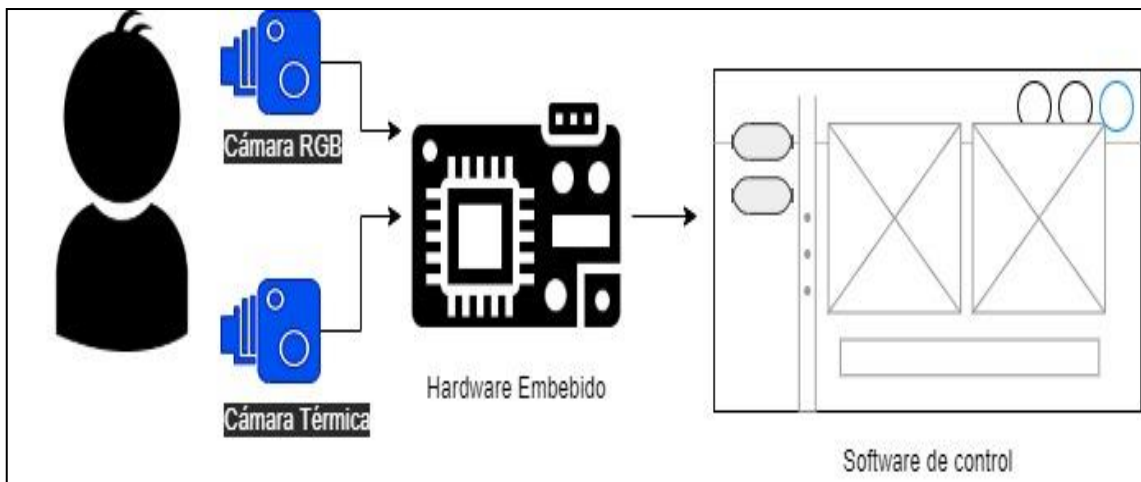
```

**Figura 6-3:** Código de programación en el software Python.

Realizado por: Armijos Soledad,2020

### 3.6 Desarrollo del Software de Control

#### 3.6.1 Operación General del Sistema



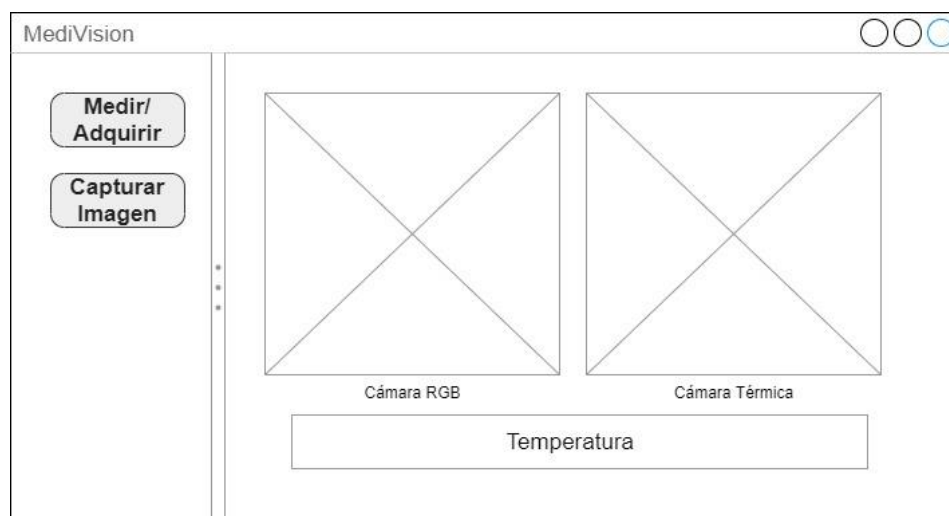
**Figura 7-3:** Funcionamiento del sistema.

Realizado por: Armijos Soledad,2020

La figura 7-3 representa gráficamente el funcionamiento del prototipo de monitoreo infrarrojo para supervisión de temperatura. El mismo que constó de una etapa de adquisición de imágenes junto con el hardware que permite el control del dispositivo, se utilizó dos cámaras una RGB y otra térmica Flir Lepton 2.5, mediante estos dos dispositivos las imágenes son adquiridas por el hardware embebido encargado de procesarlas para determinar la temperatura en el niño, midiendo la temperatura en una región concreta del niño en este caso en la frente y en el canto interior del niño. Los resultados de procesamiento y temperatura se visualizan en el software de control que

se diseñó en el lenguaje de programación Python para que la interacción con el usuario sea más sencilla y rápida.

En la figura 8-3 se visualiza la estructura de la interfaz gráfica que se desarrolló para el prototipo, utilizando el lenguaje de programación Python, se diseñó una ventana, en la misma que se observan dos botones que es Medir/ Adquirir y otro de capturar imagen estos botones se muestran al inicio de la GUI, pues cuando el dispositivo empieza su funcionamiento la imagen de la cámara RGB y térmica se muestran simultáneamente. En el botón medir temperatura se ejecutan varios cálculos intermedios como reconocimiento facial, determinación de la región de interés de la imagen RGB y la imagen térmica, una vez visualizada la imagen del niño frente a la cámara y mediante el botón medir, se procesan los cálculos y se obtiene la lectura de la temperatura y el estado febril del niño ubicada en la parte inferior de la ventana de la interfaz, si es menor a 37 grados centígrados se reconoce como temperatura corporal normal y si es mayor a esta se detectará como estado febril en el niño, este rango de temperatura se tomó en cuenta según Diakides, Bronzino y Peterson que es el lumbral escogido para este proyecto de titulación.



**Figura 8-3:** Estructura de la interfaz gráfica.

**Realizado por:** Armijos Soledad,2020

El comportamiento general de la aplicación se lo describe en el diagrama de flujos de la figura 1-3, primero se inició con la conexión de la cámara, luego se procedió a una etapa de adquisición de imágenes mientras que la interfaz espera a una interacción en la GUI, esta interacción se puede realizar en los distintos procesos propuestos como: “Guardar Imágenes”, “Procesamiento de imágenes y medición”, “Configurar botón a medir”, para ejecutarse las acciones determinadas de cada caso. Cuando no se ha realizado ninguna interacción en el proceso se puede visualizar en la interfaz el botón con la opción Medir/ Adquirir e inmediatamente se visualiza en la pantalla las imágenes por las dos cámaras. Al momento que se ejecuta la opción del botón “medir” se procesan

las imágenes y se mide la temperatura realizando el cálculo de detección de rostro, determinación de los puntos de interés en el rostro y a partir de estos puntos se determina la región de interés en la imagen RGB para determinar los puntos que se requiere en la imagen térmica y realizar los cálculos correspondientes de medición de temperatura. Si la temperatura obtenida es mayor a 37 grados centígrados que es la temperatura umbral escogida, se visualiza un mensaje en la GUI que indica el grado de temperatura corporal con un mensaje indicando que el niño esta con fiebre, incluyendo un mensaje de alerta al número celular registrado como responsable del niño, indicando el estado febril del mismo, en caso que la temperatura sea menor a 37 grados centígrados se indicará con un mensaje en la interfaz gráfica indicando que el niño se encuentra en estado normal. Cuando el botón cambia su estado a adquirir lo que se muestra en pantalla son las imágenes resultantes del procesamiento, luego se espera nuevamente una interacción con el usuario y se repite el ciclo.

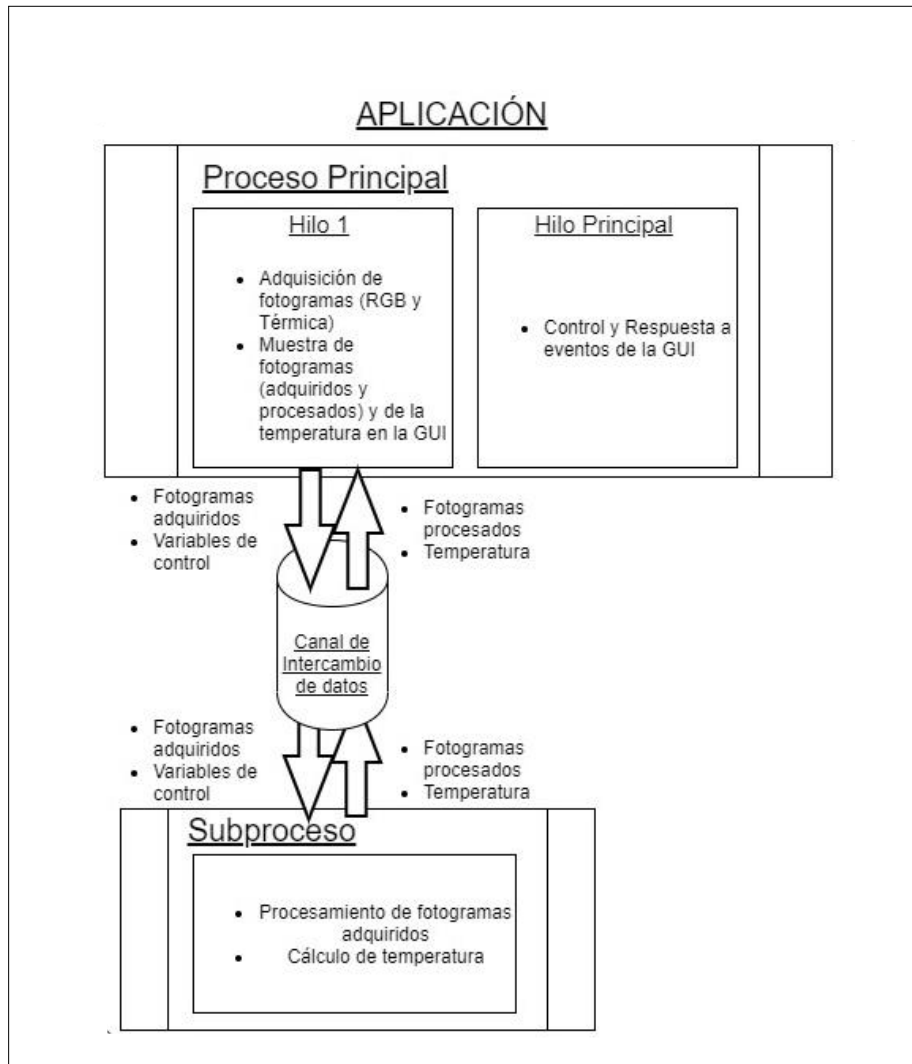
En los dos ciclos sea de adquisición o medición de imágenes se puede presionar el botón de “guardar imágenes” que guarda las imágenes reales captadas con la cámara térmica de 80x60 cuyos pixeles varían de 0 a 4000, esto por si se requiere algún tipo de análisis en otro software.

Cuando se presiona el botón “cerrar” se visualiza un mensaje de salida del programa donde se pregunta si en realidad desea terminar el proceso, si este es el caso termina su proceso el programa y se cierra y si es que no, se mantiene en el mismo bucle.

### **3.7 Implementación del Software de Control**

A nivel de programación la aplicación está constituida como se muestra en el gráfico 2-3, para garantizar una mejor respuesta al usuario se creó una aplicación multi hilo y multiproceso.

Hacerlo de la forma multi hilo o multi proceso aporta un nivel de paralelismo, significa que pueden ejecutarse varias instrucciones a la vez y no una a una, gracias a que los procesadores poseen varios núcleos y mediante software se puede indicar que las instrucciones se ejecutan al mismo instante, y el resultado será la suma de los productos parciales. Todo este conjunto de procesos se realizó con el lenguaje Python, constando de un proceso principal divididos en dos hilos.



**Gráfico 2-3:** Diagrama de proceso de aplicación del dispositivo.

Realizado por: Armijos Soledad,2020

### 3.7.1 *Hilo principal*

El hilo principal se encargó del control y de la respuesta a eventos de la interfaz de usuario es decir que se encarga de verificar las interacciones de la GUI.

### 3.7.2 *Hilo 1*

Este hilo se encargó de la etapa de adquisición de imágenes y la muestra de las fotografías adquiridas y procesadas y de la temperatura de la GUI. Gracias a este tipo de procedimiento el procesamiento es rápido porque así la aplicación no se ocupa de adquirir fotogramas mientras se intente ejecutar otro proceso.

### **3.7.3 Proceso y Subproceso**

Una vez que el proceso principal se haya realizado, se desarrolló la fase de comunicación con un proceso en paralelo que fue creado para que se ocupe del cálculo de la detección de rostro, detección de los puntos de interés en el rostro, medición de temperatura, etc. Para realizar estos cálculos es necesario un canal de comunicación o intercambio de datos. Existen algunos métodos como colas, pilas, árboles, pero el que se ha escogido para trabajar en este proyecto de titulación es el llamado pipe o tubería ya que es un canal bidireccional y brinda facilidad al trabajar con el lenguaje Python, este permite transportar las imágenes adquiridas en el hilo al momento de presionar el botón “medir”, enviadas hasta el subproceso para realizar el cálculo respectivo de la temperatura. Cuando se haya realizado el subproceso este envía las imágenes procesadas y la temperatura hacia el hilo 1 del proceso principal para que se encargue de mostrar los resultados.

Al realizar el proceso de esta manera el sistema mostró los fotogramas y reaccionó a los eventos de la interfaz de usuario.

### **3.8 Desarrollo del Algoritmo de Medición de Temperatura**

Para el desarrollo del algoritmo del dispositivo de medición de temperatura se importó un conjunto de librerías necesarias para la ejecución del programa, las mismas que serán detalladas a continuación:

Time: se usó principalmente para tratar con algunas métricas que tiene el programa, el tiempo que tarde en procesar los datos entre otros.

Numpy: se utilizó para el cálculo numérico en matrices.

Imutils: Se empleó para que implemente funciones que son exclusivamente de procesamiento de imágenes y visión artificial.

Dlib: Se empleó para la detección facial y reconocimiento de los puntos de interés del rostro del niño.

Tkinter: Se utilizó para el desarrollo de la interfaz gráfica.

PIL: Se empleó para mostrar las imágenes en la interfaz gráfica porque no se pueden presentar directamente si no que se necesita hacer operaciones intermedias con la librería PIL.

Multiprocessing y threading: Destinadas para el desarrollo del proceso principal usando la técnica de multi hilo y multiproceso.

La librería usada para el funcionamiento de la cámara térmica Flir Lepton 2.5 es pylepton.

Para el control de algunos aspectos de la cámara Flir Lepton 2.5 se realizó un script en el lenguaje C, compilando el kit del desarrollo de software de la cámara térmica para poder ejecutar configuraciones de la cámara y realizarlos directamente desde Python.

El uso del SDK de Lepton para comunicarse con la cámara Lepton requiere abrir un puerto de comunicación antes de realizar cualquier otra llamada. La operación de puerto abierto especifica la deseada velocidad en baudios para el puerto y devuelve un descriptor de puerto para usar con todas las demás API de SDK. Un anfitrión necesita abrir un puerto para cada conexión puerto-cámara que admiten. Por lo general, esto es solo una vez, pero el SDK no impone ninguna limitación. La operación de apertura de puerto también identifica la ID del dispositivo, liberando automáticamente la aplicación Host sin tener que especificar el ID del dispositivo. El descriptor de puerto devuelve el ID de dispositivo seleccionado.

Para el funcionamiento de la cámara térmica Flir Lepton 2.5 se utilizó librerías disponibles en la página oficial de la Flir Lepton que permiten la comunicación con las interfaces I<sup>2</sup>C y SPI de la Raspberry Pi.

Todo este procedimiento se ejecutó desde Python pues lo que se hizo prácticamente con el código de lenguaje C es definir algunas funciones para implementar en las configuraciones de la cámara térmica, y se utilizó estas funciones para no implementar otros parámetros ya que desde Python es más complejo debido a que cada una de ellas utilizan algunos tipos y clases definidas en la SDK de la Flir Lepton.

Una vez compilado el código en el lenguaje C se transfirió a Python a través de la librería “ctypes”, con esta librería se crea un objeto con la clase LoadLibrary que contiene como parámetro la ruta de nuestra librería compilada.

Se inició la conexión de la cámara a través de la interfaz I<sup>2</sup>C para poder modificar algunas de sus características y desactivar la librería TLinear, el cual convierte el valor de pixeles del video a un valor de temperatura en grados Kelvin multiplicados por 100 para poder obtener exclusivamente el valor de pixeles sin procesar. Luego se desactivó así mismo el AGC el cual ejecuta un algoritmo

de mejora de contraste para que la salida que es de 0 a 4000 pase a ser de 0 a 255 y así permite distinguir un punto en la imagen que tenga una temperatura inferior respecto al resto del ambiente.

### 3.8.1 *Método constructor*

Se creó una clase de interfaz gráfica en el que se define algunos métodos entre ellos el método constructor, el mismo que se encarga de realizar la creación de la interfaz gráfica, para ello se determinó algunos atributos como la imagen de ancho y la imagen de alto, para definir el tamaño de la imagen que se muestra en la interfaz gráfica, las mismas que tienen una medida de 400 y 430 respectivamente. Se escogió estos valores mediante pruebas realizadas en el dispositivo para que el usuario obtenga una visualización adecuada de la imagen proyectada de las cámaras.

Dentro del método constructor se desarrolló el proceso de crear interfaz, mediante el cual se elaboró una interfaz de usuario usando la librería tkinter, creando una ventana utilizando el constructor Tk de la librería mencionada, luego se configuró el título y se definió algunas partes de la interfaz gráfica o widgets que forman parte de los controles, decoraciones, indicadores, etc.

LabelFrame es un marco el cual contiene los controles proporcionados al manejo del proceso del sistema de la interfaz gráfica diferenciando controles e indicadores para conocer la función que cumple cada uno. Se utilizó el gestor de acabados grid, mismo que proporciona una posición a la ventana gráfica principal haciendo que se ubique en una determinada área, de acuerdo a los parámetros que se asignó como: la posición de fila cero, columna cero, relleno horizontal de cinco, relleno vertical 4 y que ocupe toda la posición de la celda. (00: row=0; column=0; padx=5; pady=4; sticky= NSEW).

La creación de botones en la GUI se obtuvo después de la ejecución de parámetros como el FrameControles. El texto que se muestra desde el inicio del proceso “medir temperatura” y el parámetro command que es una función que se crea al inicio del procesamiento, esta no es una variable si no una referencia a una función definida por el usuario, es decir indica lo que el botón realiza cada vez que sea presionado. Se dan también posiciones relativas al FrameControles asignadas como: row=1; column =0; padx=10; pady=10; sticky= SEW. El botón capturar imágenes lo que hace es guardar las imágenes mostradas en la interfaz gráfica de usuario.

Para el marco que contiene a las imágenes adquiridas por la cámara se creó otro LabelFrame que contiene indicadores que muestran las imágenes que son adquiridas por las cámaras o las que devuelva el subprocesso que se encarga del procesamiento de las imágenes, dando así las



posiciones relativas al contenido de la ventana principal, configurando sus parámetros como el alto de la imagen con el comando `rowconfigure`, de la misma manera se configuró el parámetro de altura de la imagen con el comando `columnconfigure`, usando el mismo proceso de configuración para cada uno de los marcos contenidos dentro de la interfaz gráfica.

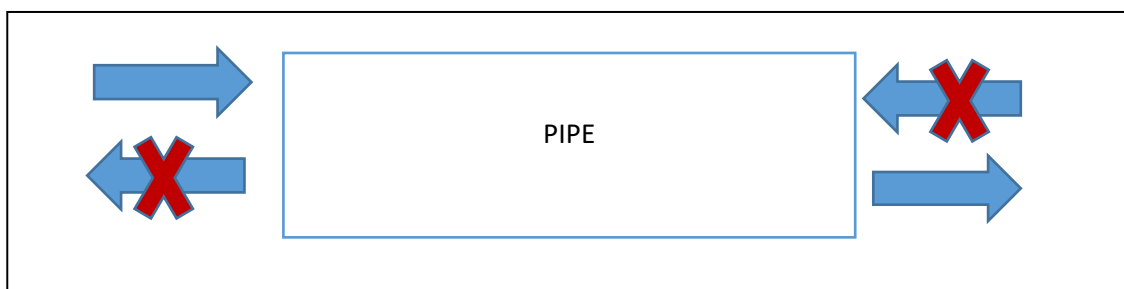
Dentro de los `labelframe` se utilizó `widgets` que sirven para mostrar imágenes o textos, estos se denominan `label` en `tkinter`, como parámetro obligatorio es el contenedor y opcional es el texto, así mismo se definió otro `labelframe` utilizado para medir la temperatura, el mismo que muestra el texto que dice “presione el botón medir para obtener lectura”.

En la ventana principal se configuró un atributo llamado “resizable” que tiene como finalidad desactivar la posibilidad de configurar el tamaño de la ventana tanto en ancho como alto de la imagen, de esa forma se evita que los `widgets` se desalíneen por opciones como maximizar, minimizar, etc.

El comportamiento que la ventana tiene cuando se pulsa el botón “cerrar” se definió con la instrucción `self.ventanaprincipal.protocol` y como parámetro se obtiene lo que se va a cambiar que es el protocolo de cerrado, dando como segundo parámetro una referencia a una función que contiene el código que va a ejecutarse cuando se presiona el botón “cerrar”, para que al realizar esta opción muestre la notificación al usuario y le pregunte si “¿Desea salir de la aplicación?”, luego libere recursos y cierre el programa.

Para el control del comportamiento del proceso se estableció variables precisas para la ejecución del mismo como se observa en el diagrama de flujo del gráfico 1-3.

Se usó el método constructor `pipe` de la librería `multiprocessing` que define el canal de comunicación, creando así dos terminales el terminal uno y el terminal dos, estos pueden comunicarse a través de cada extremo del `pipe` y en la misma dirección, pero no al mismo tiempo porque se puede perder datos de lectura o escritura, tal como se indica en la figura 9-3.



**Figura 9-3:** Comunicación de los terminales de pipe.

Realizado por: Armijos Soledad,2020

### **3.8.2 *Proceso en paralelo***

Este proceso se utilizó principalmente para depuración de códigos en la etapa de pruebas, y así ordenar lo que se realiza en el proceso, en este proceso se da nombre a la variable target que realizó la función de procesar video, ejecutando la transmisión de algunos parámetros como la del extremo terminal 1 del pipe.

Luego del subprocesso se asignó el parámetro `self.procesamiento.daemon = True`, este controla la posibilidad de si se ejecute el cierre del programa no siga actuando el proceso en un segundo plano y no ocupe más recursos.

### **3.8.3 *Análisis del programa principal***

Se creó un thread asignando como parámetro “Actualización de video en GUI” y como argumento de esta función se asigna el otro extremo del pipe es decir el terminal 2, de la misma manera al atributo Daemon se asignó un “true” para que se cierre todo cuando finalice el proceso principal.

Se inició un hilo para adquirir imágenes y también para recibir las imágenes del proceso en paralelo, ambos resultados de adquisición y procesamiento son mostrados en la GUI por este hilo, utilizando la función “start” para que el hilo llamado videoGUI comience a ejecutarse en paralelo.

Seguido este paso, se inició el proceso en paralelo encargado del procesamiento de imágenes. Este proceso envió los frames procesados al hilo para mostrarlos en la GUI junto con la temperatura, mientras que el control de la interfaz de usuario se ejecutó mediante el método “mainloop” del objeto de entrada principal que es de la clase Tk, iniciando así el bucle de la interfaz gráfica que corresponde a hilo principal en el proceso.

Para realizar la medición de temperatura primero se ejecutó mediante el botón “medir temperatura” asignado por la línea de código de inicio de procesamiento que comienza a ejecutarse cuando se presiona el botón “medir temperatura”.

Al iniciar el procesamiento el atributo medición se asignó un “true”, este funciona controlando el proceso que tiene el hilo de actualización de video es decir el hilo 1, enviando los fotogramas y el valor de “true” a través del pipe, haciendo el cálculo en el subprocesso y devolviéndolo al proceso principal.

La variable asignada como “botonMedir” se configuró para que su texto indique “adquirir” y su atributo command cambia a la función adquirir, esto quiere decir que en el botón “medir” se asignara la función “adquirir” y en lugar de iniciar el procesamiento se realiza la adquisición de los datos capturados de la imagen.

El proceso que realiza la función adquirir es cambiar la variable de resultado a “falso”, modificando el texto nuevamente a “medir” y el atributo command se modifica a “iniciar procesamiento”, ejecutando dos funciones en el mismo botón.

Dentro del cálculo de la función iniciar procesamiento, la variable self.medicion= True se enlazó con el hilo VideoGUI, el mismo que está unido al método “actualizarVideoGUI”, este se conecta con la cámara RGB a través de la librería OpenCv que creó un objeto constructor de la clase VideoCapture. Luego se inició un bucle while donde se ejecutó la función real que adquiere la imagen, devolviendo dos valores, un valor lógico y la imagen. El valor lógico es verdadero si no se obtuvo problemas al momento de la captura de la foto y es falso si ocurre lo contrario, en caso de ser verdadero se realiza la comunicación con la cámara térmica mediante la librería pylepton, por lo tanto se usa el módulo lepton y se asigna la variable camaraTermica, teniendo esta variable se emplea la comunicación para adquirir los fotogramas que se envíen por la interfaz SPI, este proceso se realizó utilizando la función capture que devuelve la imagen térmica que tiene dimensiones de 60,80,1 siendo un arreglo tridimensional pero como los datos están como matriz no como arreglo tridimensional se extrajo las filas y columnas de la primera matriz del arreglo tridimensional para transformarlas en un arreglo bidimensional. La imagen que devuelve el método capture de la librería lepton contiene pixeles cuyos valores oscilan entre 0 y 4000, estos valores sirven para propósitos de análisis de la temperatura por este motivo se necesita una copia de dicha imagen para esto se usó el método “copy” que devuelve una copia de la matriz, asignándole la variable frameT, esta variable se envía al proceso en paralelo para el cálculo de temperatura respectivo.

Para el proceso de visualización en la interfaz gráfica se normalizó los valores obtenidos del método capture de la imagen térmica, asignando los valores mínimos a ceros y los máximos a 65535 para extender el contraste, pero para que la imagen sea la adecuada se ejecutó un desplazamiento de bits hacia la derecha para que los datos se encuentren dentro de los 8 bits, esto quiere decir que en este proceso la imagen ya toma valores de entre 0 y 255 donde los ceros toman un color más oscuro y los de 255 se aproximan a tonos claros como se indica en la figura 10-3, sin embargo para transformarlos a un entero de 8 bits se utilizó el comando unit8 de la librería numpy, realizado este procedimiento la imagen térmica ya se ha transformó en escala de grises y

para que no se muestre en escala de grises se aplicó un mapa de color mediante la función “applyColorMap” que se encuentra dentro de la librería cmapy (CMRmap)



**Figura 10-3:** Proyección de colores dispuestos en CMRmap.

Fuente: Vejarano, 2019

Una vez adquiridas las imágenes térmicas y de la cámara RGB se ejecutó la sentencia “if” que tiene como condición la variable medición, esta toma valores de verdadero o falso según se presione o no el botón “medir”, por lo tanto cuando se presiona dicho botón la variable toma el valor de “true” y se ejecuta el proceso, ya que los hilos de un proceso si comparten el mismo espacio de datos, entonces si se puede leer una variable que ha sido modificado dentro del hilo principal, mientras la variable sea verdadera se muestra en la interfaz gráfica un mensaje de “procesando” hasta que se visualice la temperatura calculada.

Al terminar la etiqueta de “procesando” se construyó una lista que es un tipo de dato de Python donde se almacenan valores de diferente tipo, en este caso particular se almacenó una matriz de 640 x 480, también se guardó la imagen térmica con valores originales de 0 a 4000 y un valor lógico que servirá para que el subprocesso reconozca si se debe calcular la temperatura. Estos tres parámetros se envían a través del canal de comunicación pipe para luego enviar una lista que se acaba de construir con la función send del terminal 1, posteriormente se ejecutó la función recv del terminal 1, la misma que espera que haya un dato disponible en el pipe para aceptarlo y almacenarlo en otra variable, pasando directamente al subprocesso.

El proceso en paralelo se enlazó con la función de procesar video donde se definen los parámetros utilizados en la ecuación para obtener la temperatura, proporcionando constantes que permiten detectar los puntos donde se mide la temperatura corporal, en este caso los ojos, donde a partir de estos puntos se obtuvo la región de interés. En esta fase se utilizó un detector de rostros y un clasificador de formas para encontrar los puntos de medición en el rostro del niño, luego se ejecutó un bucle “while” dentro del subprocesso en paralelo, efectuando la función recv del terminal 2, una vez recibidos los datos, se almacenó estos datos en la variable “listaRecibida” que contiene el fotograma adquirido de la cámara RGB y de la térmica obteniendo un valor lógico que se asigna

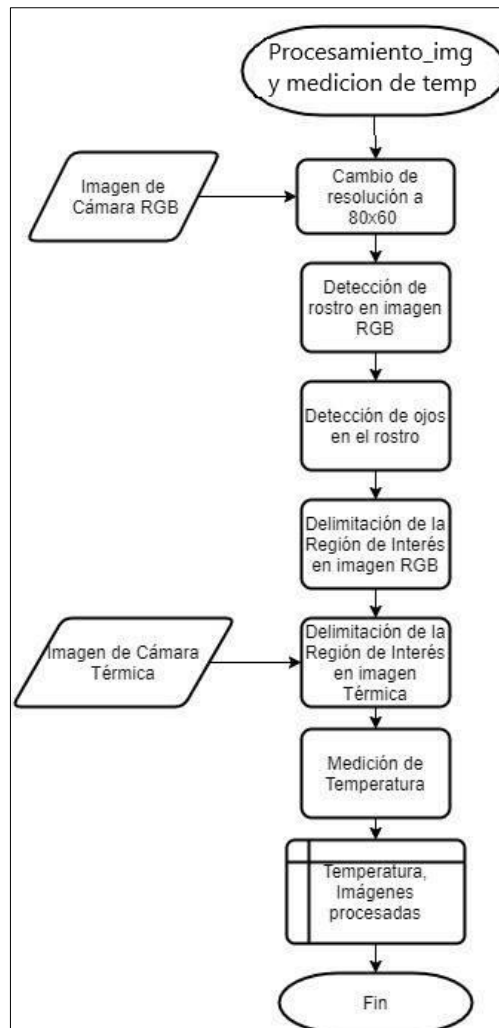
a la variable “calcular” y si esta es verdadera se continúa con la fase de procesamiento de las imágenes.

Por lo tanto, se recibió la imagen de la cámara RGB, la cual tiene una resolución de 80 x 60 puesto que ese mismo tamaño tiene la imagen térmica, esta imagen se envió al proceso de detección de rostros, y a partir del rostro detectado, utilizando el clasificador de forma se detectó los ojos en el rostro del niño. Seguido este paso se delimitó la región de interés en la imagen RGB para realizar una transformación de coordenadas y obtener la misma zona delimitada en la imagen térmica, dado que el dispositivo se encuentra fijo se puede analizar la distancia de las imágenes a procesar, realizando una translación o escalamiento de coordenadas. Una vez calculada la región de interés y al realizar la transformación de coordenadas, obteniendo la imagen térmica, se definió la zona requerida en la misma y se realizó la medición de temperatura que promedia el valor de los píxeles dentro de la región de interés por medio de una ecuación, que relaciona el promedio de píxeles con la temperatura resultante de la calibración de la cámara, por lo tanto, devuelve la temperatura y las imágenes procesadas.

Dentro de la estructura “if” se encuentran las variables “factorEscalaX” y “factorEscalaY” que permite visualizar los recuadros de las posiciones de la imagen de la cámara RGB y térmica, siendo un cociente entre el ancho de la imagen original de 640 dividido para el ancho de la imagen térmica de 80, todo esto convertido en un valor entero, para el factor de escala en horizontal y vertical es el mismo procedimiento pero con valores tomados del alto de la imagen que es la división de 480 que corresponde a la imagen RGB entre 60 de la imagen térmica.

#### **3.8.4 *Procesamiento de imágenes y medición de temperatura***

El algoritmo que se utilizó para realizar el procesamiento de las imágenes capturadas por el dispositivo se detalla paso a paso en el diagrama de flujo del gráfico 3-3, por lo tanto, se explicará a detalle el procedimiento de cada uno de ellos.



**Gráfico 3-3:** Procesamiento de imágenes y cambio de temperatura.

**Realizado por:** Armijos Soledad,2020

#### 3.8.4.1 *Cambio de resolución de imagen*

Para la reducción del tamaño de imagen RGB 80 x 60 donde el código `frameTermica.shape[1]` arroja el número de columnas de la imagen térmica que es 80 y el `frameTermica.shape[0]` el número de filas que es 60, se utilizó la función `resize` de `openCv` la misma que reduce el tamaño de la imagen, para luego realizar la copia del código `frameTermica` a una variable llamada `frameT` y así obtener la visualización y análisis de las imágenes respectivamente, luego de convertir las imágenes en escala de grises se redimensionó la imagen al tamaño de la fotografía en RGB. A esta imagen se le aplicó un mapa de color para que se observe en GUI como una imagen con colores asociados a una escala térmica y no en escala de grises.

#### 3.8.4.2 *Detección de rostro*

Para el procesamiento de imágenes donde se realizó la detección del rostro para la medición de temperatura, se hizo uso de la librería “dlib” que tiene implementado un detector de rostro compuesto por un algoritmo de programas llamado HOG o “histograma de gradientes orientados” que sirve para extraer las características del rostro y se clasifica a través de una máquina de soporte de vectores, que son algoritmos de aprendizaje automático, donde el rostro detectado pasa a través de un clasificador de formas implementados mediante árboles de regresión ocupándose de distinguir los puntos de interés en el rostro que se dé como parámetro. La variable detector y predictor se encargó de localizar el rostro y los puntos de interés respectivamente.

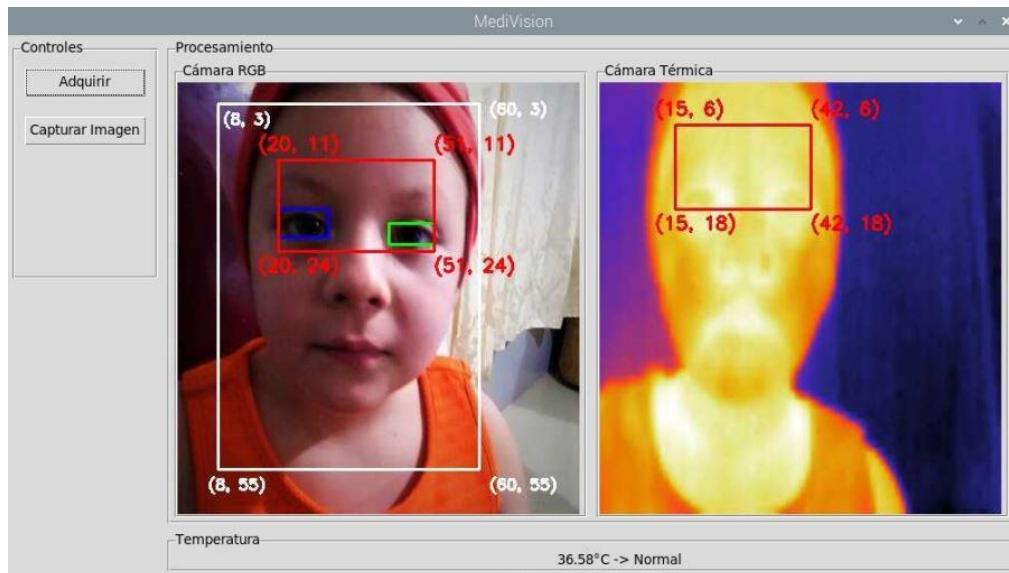
El método de histogramas de gradientes orientados usa pirámides de imágenes y el número de pirámides usadas está determinado por el parámetro (imagenRGB, 1), el número de imágenes que se requirió detectar y el número de pirámides de imágenes. Para esto se definió una lista donde se guardan las coordenadas de la región de interés del niño. En la secuencia “for” se encuentra la función “enumerate”, lo que hace esta función es recoger los valores de la variable de la lista de detecciones del rostro, esta variable debe ser un objeto iterable pudiendo ser un arreglo o matriz, esta devuelve el valor del índice iterable y del índice mismo que se extraen con las funciones propuestas como se indica en la figura 14-3, se utilizó el código “. left” que arroja las coordenadas de la izquierda en “x” y dibuja un rectángulo en la imagen RGB para poder realizar el cálculo del parámetro que contiene el punto inicial del rectángulo visualizado para la detección del rostro, es decir contiene los ejes de la esquina superior izquierda en valores X e Y, y lo mismo para los puntos X e Y finales del rectángulo. Para la selección del color en donde se requiere dibujar el rectángulo se realizó mediante una tupla de tres valores donde el primer valor está asociado con el azul, el segundo con el verde y el tercero con el rojo, siendo el grosor de la línea el último parámetro de ejecutar.





### 3.8.4.5 Determinar la región de interés en una imagen térmica

Mediante las pruebas realizadas durante la ejecución del dispositivo se obtuvo que multiplicando la coordenada  $X_i$  de la región de interés en la imagen RGB por el factor 1.2, el resultado será la misma coordenada, pero mapeada en la imagen térmica, este valor se colocó con respecto a la ubicación de las cámaras en el dispositivo, debido a que la cámara térmica siempre se posiciona más cerca del objetivo en relación con la cámara RGB como se puede visualizar en la figura 12-3. Para la coordenada  $Y_i$  se multiplicó por el factor 1.2 pero restando el valor de 5 ya que la imagen térmica se muestra ligeramente más abajo con respecto a la imagen RGB debido al posicionamiento del dispositivo, esto se realizó con la finalidad de que se otorgue al usuario una correcta visualización en la GUI, el mismo procedimiento será para las coordenadas finales, de esta manera se obtuvo las coordenadas de interés en la imagen térmica.



**Figura 12-3:** Visualización de la imagen RGB y térmica en la GUI.

**Realizado por:** Armijos Soledad, 2020

### 3.8.4.6 Medición de Temperatura

Para la medición de temperatura se hizo un recorte de la imagen térmica según la región de interés analizada, los pixeles que están dentro de la imagen se promedian usando el código de línea `ROI_termica.flatten().mean()` guardado en la variable `pixelesPromedio`, utilizando la ecuación 1-3 de temperatura que devolverá la temperatura en grados centígrados, la misma que se muestra a continuación:

**Ecuación 1-3:** Ecuación de medición de temperatura.

$$T = \frac{B}{np} \times \log\left(\frac{R}{(\text{pixelesPromedio} - 0) + F}\right) - 273.15$$

Realizado este procedimiento se envió los datos aplicados al proceso principal para mostrar resultados, para ello se construyó una lista que contiene tres elementos, que es la imagen RGB y térmicas procesadas cada una y un valor numérico que es la temperatura T, calculada por la ecuación (1) mencionada anteriormente para enviarlas a través del pipe del terminal 2 y que el hilo reciba estos datos para así poder configurar el texto del label Temperatura procesado. Realizado este proceso se extrajo los valores recibidos mientras que el código `end=time.perf_counter` verifica el tiempo que tarda todo el proceso dado y se ejecutan acciones dependiendo del valor calculado de la temperatura.

Para el resultado final se utilizó “stringTemperatura” con el valor que se obtuvo de temperatura y dependiendo de ese valor si es  $>37^{\circ}\text{C}$  que es el umbral establecido, se indicó en la pantalla la palabra ‘Fiebre’ o ‘Normal’ dependiendo de cuál sea el caso en el que el niño se encuentre.

Mediante la instrucción `cliente.send_message` se envió el mensaje de texto a los padres o persona responsable del infante (fiebre registrada. Temperatura: ‘stringTemperatura’) una señal de alerta y precaución del estado en el que se encuentra su hijo. En caso de que la temperatura no haya sido mayor al umbral escogido se muestra la temperatura con el indicativo de la palabra “Normal” en lugar de “fiebre”, para esto la variable resultados se cambiará a TRUE y mediación a FALSE para que el proceso que está ejecutándose en paralelo evite seguir calculando la misma imagen.

Una vez terminado el proceso anterior la variable `frameRGB` cambia, igual que la variable `frameTermica` para que muestre la imagen ya procesada de cada una en la interfaz gráfica, vistas en el `panelVideoRGB` ubicadas dentro de la función denominada `mostrarEnGUI`, la misma que se encargó de redimensionar la imagen que se recibe a 400 de ancho x 430 de alto convirtiéndola de BGR a RGB, debido a que estas imágenes no se pueden mostrar directamente en la interfaz gráfica y hay que procesarlas mediante librerías como `pill`, por este motivo se utilizó el módulo “image” que obtuvo una imagen a partir de una matriz que se guardó para luego ser procesada mediante el comando `ImageTk.PhotoImage(imageGUI)`, para así ya ser visualizada la imagen en la GUI donde el parámetro `panel` que es el panel video RGB, configurando su atributo `image` para entregar la imagen requerida.

Para el proceso del cierre del programa y de acuerdo al método constructor ya mencionado, se cambió el comportamiento del cierre del programa y se asoció a una función que muestra el

mensaje al usuario donde pregunta si ¿desea salir de la aplicación?, en caso de que la respuesta sea afirmativa se libera los recursos usados por la cámara RGB y termina el programa con la instrucción `self.ventranaPrincipal.destroy()`.

Para la calibración de la cámara se requirió obtener dentro de una región de interés el promedio de pixeles correspondientes a un objeto que se encuentre dentro de dicha región y cuya temperatura se conozca, para esto se realizó pruebas orientas a medir la temperatura del niño con síntomas de estado febril en función de la región de interés que en este caso se definió en los ojos del infante, tomando fotos y obteniendo datos que se relaciona mediante la ecuación 2-3 no lineal mostrada a continuación : (Lin., Lu, y Lin, 2019)

**Ecuación 2-3:** Ecuación para calcular  $T_R$

$$T_R = \frac{B}{\ln\left(\frac{R}{S-O} + F\right)} - 273.15$$

Esta ecuación devolvió la temperatura en grados kelvin a partir de una entra 'S' que es el promedio de pixeles en la región de interés, dicha ecuación se obtuvo a partir de los datos de aplicación de la cámara leptón para obtener sus parámetros B, R, O y F y que se ajusten a nuestro datos, obteniendo una toma de valores manteniendo un promedio de pixeles permaneciendo en el modelo no lineal de la ecuación 2-3, se encontró el valor de los parámetros que mejor se ajuste a los datos, los mismo que pueden variar entre los valores siguientes  $R=(10000, 1000000)$ ;  $B=(1200,1700)$ ;  $F=(0.5,3)$ ;  $O=(-16384, 16383)$ , donde a F generalmente toma un valor de 1 y B corresponde a 1428, ajustando los parámetros de O y R. Para obtener los valores de O y R usando el lenguaje de programación Python, se utilizó las librerías numpy y scipy que se utilizó también para el ajuste de curvas y matplotlib para la obtención de las gráficas. Se definió luego la función del modelo no lineal dado por la ecuación 3-3:

**Ecuación 3-3:** Ecuación para encontrar el valor promedio de pixeles.

$$S = \frac{R}{\exp\left(\frac{B}{T_K}\right) - F} + O$$

Obteniendo así los pixeles en función de la temperatura, a la misma vez que se obtuvo los datos definidos como matriz como se muestra en la figura 13-3, se ejecutó una columna de temperatura y de pixeles y se usó la función `curve_fit` de las librerías scipy para adecuar la función de ajuste a los datos previamente obtenidos.

```

import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

def funcionAjuste(temperaturas, R, O):
    B = 1428
    F = 1
    temperaturas = temperaturas + 273.15 #convierte temperaturas de °C a K
    return R/(np.exp(B/temperaturas) - F) + O

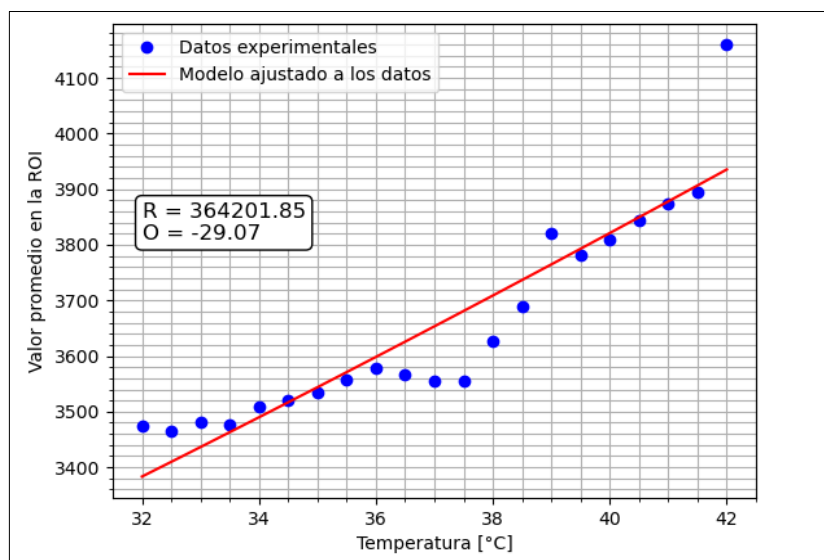
datos = np.array([
    [32, 3473.6],
    [32.5, 3465.69],
    [33, 3480.94],
    [33.5, 3475.69],
    [34, 3508.0],
    [34.5, 3519.99],
    [35, 3533.3],
    [35.5, 3556.9],
    [36, 3577.9],
    [36.5, 3566.5],
    [37, 3554.0],
    [37.5, 3555.61],
    [38, 3625.81],
    [38.5, 3688.84],
    [39, 3820.07],
    [39.5, 3781.45],
    [40, 3809],
    [40.5, 3843.9],
    [41, 3873.06],
    [41.5, 3895.07],
    [42, 4159.33],
])

```

**Figura 13-3:** Datos calculados para la obtención de la matriz aplicada en la ecuación.

Realizado por: Armijos Soledad,2020

Entre los parámetros mencionados se calculó los valores óptimos, es decir O y R para el ajuste de datos y continuar con el proceso de la obtención de la gráfica. Se graficó los datos experimentales que son las medidas tomadas, luego se graficó el modelo y se obtuvo una gráfica de la ROI vs Temperatura como se observa en el gráfico 4-3.



**Gráfico 4-3:** Resultados de Temperatura vs ROI, obteniendo valores de R y O.

Realizado por: Armijos Soledad,2020

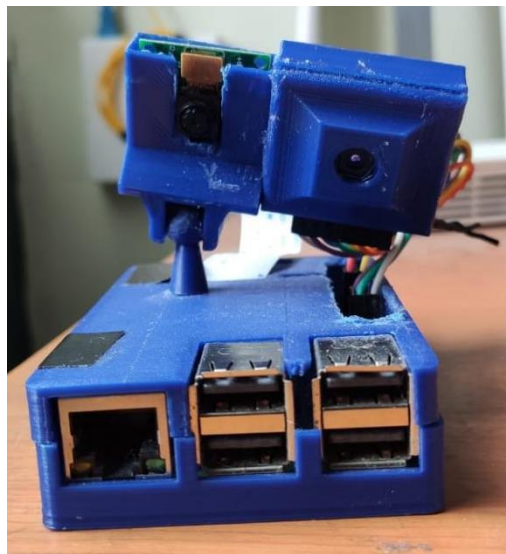
## CAPITULO IV

### 4 MARCO DE RESULTADOS, ANALISIS Y DISCUSIÓN

#### 4.1 Pruebas de Funcionamiento del Prototipo

El proceso para medir el estado febril del niño del prototipo radicó en la captura de imágenes con dos cámaras instaladas en el dispositivo que son la cámara RGB y la cámara térmica Flir Lepton 2.5, y por medio de una interfaz gráfica se visualizó el rostro del niño, indicando la región de interés y las coordenadas de la imagen donde se detectó la temperatura que indica si el infante se encuentra en estado normal o en estado febril, si el caso es favorable para el estado de fiebre del niño se envió un mensaje al celular de la persona responsable del niño, indicando la temperatura corporal del mismo que será considerada de acuerdo al umbral seleccionado, en este caso es de 37° centígrados, como se indicó en el capítulo tres del escrito.

Se realizó las pruebas con 2 niños, tomando 15 datos de temperatura de cada uno de ellos. Durante las pruebas se usó el dispositivo electrónico de monitoreo infrarrojo para supervisión de temperatura implementado para este trabajo de titulación el mismo que se observa en la figura 1-4, se realizó una comparación con otros dispositivos ya existentes en el mercado, de ellos se escogió el más común utilizado que es el termómetro infrarrojo digital, el modelo de este termómetro con el que se ha trabajado es el HT-668, indicado en la figura 2-4.



**Figura 1-4:** Prototipo de medición de temperatura.

**Realizado por:** Armijos Soledad, 2021



**Figura 2-4:** Termómetro digital infrarrojo modelo HT-668.

**Realizado por:** Armijos Soledad,2021

En el interfaz del dispositivo se visualizó el procesamiento del programa, donde permitió ver la sección de la región de interés que se registró al momento de pulsar el botón medir ubicado en el recuadro de controles, en la figura 3-4 se observa que la temperatura medida alcanzó un valor de 37.4, la misma que se encontró por encima del umbral establecido por lo tanto en el label registrado como temperatura se mostró el mensaje de “fiebre” ya que el grado de temperatura ha sobrepasado los 37 grados centígrados establecido como temperatura normal.



**Figura 3-4:** Funcionamiento del dispositivo electrónico de temperatura infrarrojo.

**Realizado por:** Armijos Soledad,2021

Cuando en el programa se detectó que existe una temperatura que sobrepase el umbral establecido previamente, el programa se encargó de enviar un mensaje de alerta mediante GSM al número colocado dentro del código fuente como se indica en la figura 4-4, para que este indique a la persona que esté a cargo del niño el estado en el que se encuentra, se indicó la temperatura corporal del niño y así previene algún incidente mayor que pueda traer problemas mayores.



**Figura 4-4:** Mensaje de alerta febril recibido al número registrado en el programa.

**Realizado por:** Armijos Soledad,2021

## 4.2 Funcionalidad del prototipo

La implementación del prototipo se lo realizó midiendo el cambio de temperatura en niños menores a tres años de edad de la parroquia Eloy Alfaro Esmeraldas.

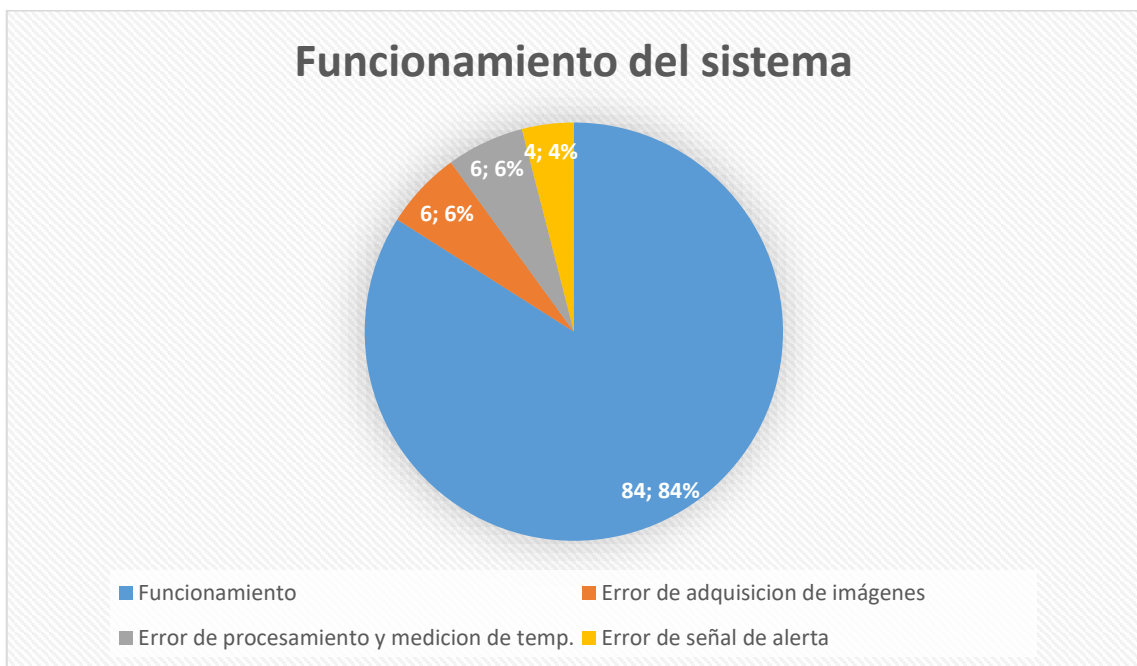
El proceso se realizó tomando en cuenta los distintos análisis que se realizó dentro de cada sucesión del prototipo como son: adquisición de imágenes, interacción con la GUI, el procesamiento y medición de temperatura y el envío de la señal de alerta a sus padres. Los resultados se muestran en la tabla 1-4.

**Tabla 1-4:** Pruebas de Funcionamiento

Nº Prueba	Adquisición de imágenes	Interacción con la GUI	Procesamiento y medición de temperatura	Señal de alerta
1	X	X	X	X
2	X	X	X	X
3	X	X	X	
4	X	X	X	X
5	X	X	X	X
6	X	X	X	X
7		X	X	X
8	X	X	X	X

9	X	X	X	X
10	X	X	X	X
11		X	X	X
12	X	X		X
13	X	X	X	X
14	X	X	X	X
15	X	X		X
16	X	X	X	X
17	X	X	X	X
18	X	X	X	X
19	X	X	X	X
20	X	X	X	X

Realizado por: Armijos Soledad,2021



**Gráfico 1-4: Funcionamiento del sistema.**

Realizado por: Armijos Soledad,2021

En la gráfica 1-4 se observa que una vez realizadas las pruebas del prototipo de acuerdo a cada fase del proceso el porcentaje de funcionamiento del Sistema fue del 84%, mientras que el porcentaje del error de procesamiento y medición de temperatura es del 6% al igual que el porcentaje de error de adquisición de imágenes, el error que se obtuvo de la señal de alerta es del 4%, mientras que la interacción con la GUI no tuvo errores y funcionó correctamente.



### 4.3 Resultados de Evaluación del Prototipo

Durante las pruebas el dispositivo electrónico de supervisión de temperatura se colocó a una distancia de 40 a 80 cm frente al niño para capturar las imágenes RGB y térmicas respectivamente. Primero se midió la temperatura corporal estable durante un determinado tiempo esta sería la Fase A de la prueba, luego se colocó una bolsa de agua fría en la frente del niño durante dos minutos para tomar nuevamente la temperatura que sería la Fase B de las pruebas, una vez realizado la fase A y B se colocó una bolsa de agua caliente durante dos minutos más y se realizó la medición de temperatura durante cinco minutos que será la última Fase, la Fase C como se detalla en la figura 5-4.

	Temperatura Fase A	Aplica bolsa con hielo	Temperatura Fase B	Aplica bolsa de agua caliente	Temperatura Fase C
Tiempo(minutos)	0-5	5-7	7-12	12-14	14-19

**Figura 5-4:** Protocolo experimental.

**Realizado por:** Armijos Soledad,2021

Para realizar las pruebas se hizo una comparación entre el termómetro digital HT-668 y el dispositivo electrónico infrarrojo implementado, calculando una estadística tomando la media del valor absoluto del error, es decir de las medidas tomadas de cada fase se calculó un promedio, y se obtuvo también un error medio cuadrático para expresar la diferencia entre el dispositivo y el termómetro digital.

Para obtener los datos de la temperatura en la fase A, se realizó con el cálculo del valor de la temperatura corporal cuando el niño se encontró en un estado de temperatura normal, es decir que no haya sido sometido a ningún cambio de temperatura por motivo externo, como se visualiza en la tabla 2-4.

En la tabla 2-4 se visualiza la Fase de las pruebas que se realizó tomando en cuenta la temperatura y tiempo del prototipo implementado y el termómetro digital.

**Tabla 2-4:** Pruebas de medición de temperatura Fase A

Fase A		
Tiempo [s]	Temperatura [°C]	
	Prototipo	Termómetro
0	36,4	36,7
10	36,2	36,3
20	36,3	36,3
30	36,4	36,5
40	36,5	36,5
50	36,6	36,5
60	36,4	36,6
70	36,3	36,5
80	36,2	36,5
90	36,4	36,7
100	36,5	36,7
110	36,6	36,7
120	36,2	36,4
130	36,5	36,5
140	36,4	36,6
150	36,1	36,3
160	36,7	36,7
170	36,3	36,4
180	36,2	36,3
190	36,5	36,7
200	35,6	35,7
210	36,4	36,5
220	36,7	36,8
230	36,4	36,5
240	36,3	36,5
250	36,2	36,4
260	36,1	36,4
270	36,2	36,3
280	36,3	36,5
290	36,5	36,7
300	36,3	36,4

Realizado por: Armijos Soledad,2021

Para obtener los datos de la temperatura en la fase B se realizó obteniendo el valor de la temperatura corporal cuando el niño haya realizado la prueba mediante la colocación de la bolsa de agua fría y así observar y digitar los datos respectivos como se visualiza en la tabla 3-4.

La tabla 3-4 muestra la Fase B de las pruebas que se realizó tomando en cuenta la temperatura y tiempo del prototipo implementado y el termómetro digital.

**Tabla 3-4:** Pruebas de medición de temperatura Fase B

Fase B		
Tiempo [s]	Temperatura [°C]	
	Prototipo	Termómetro
0	35,2	35,3
10	35,4	35,5
20	35,1	35,2
30	35,3	35,5
40	35,3	35,4
50	35,4	35,6
60	35,2	35,5
70	35,3	35,5
80	35,1	35,3
90	35,6	35,7
100	35,2	35,5
110	35,1	35,4
120	35,2	35,3
130	35,2	35,4
140	35,2	35,3
150	35,4	35,6
160	35,3	35,4
170	35,3	35,5
180	35,2	35,4
190	35,6	35,8
200	35,4	35,5
210	35,6	35,8
220	35,4	35,5
230	35,3	35,5
240	35,2	35,4
250	35,2	35,3
260	35,1	35,5
270	35,3	35,4
280	35,5	35,7
290	35,2	35,3
300	35,1	35,2

Realizado por: Armijos Soledad,2021

Para obtener los datos de la temperatura en la fase C se realizó obteniendo el valor de la temperatura corporal cuando el niño haya realizado la prueba mediante la colocación de la bolsa de agua caliente y así observar y digitar los datos respectivos como se visualiza en la tabla 4-4.

La tabla 4-4 muestra la Fase C de las pruebas que se realizó tomando en cuenta la temperatura y tiempo del prototipo implementado y el termómetro digital.

**Tabla 4-4:** Pruebas de medición de temperatura Fase C

Fase C		
Tiempo [s]	Temperatura [°C]	
	Prototipo	Termómetro
0	37,4	37,7
10	37,3	37,4
20	37,3	37,3
30	37,4	37,5
40	37,5	37,5
50	37,6	37,5
60	37,4	37,6
70	37,3	37,5
80	37,2	37,5
90	37,4	37,7
100	37,5	37,7
110	37,6	37,7
120	37,3	37,4
130	37,2	37,4
140	37,5	37,7
150	37,4	37,5
160	37,3	37,6
170	37,5	37,7
180	37,7	37,5
190	37,4	37,6
200	37,6	37,8
210	37,2	37,4
220	37,5	37,6
230	37,4	37,5
240	37,3	37,5
250	37,6	37,7
260	37,4	37,5
270	37,3	37,2
280	37,5	37,4
290	37,2	37,4
300	37,2	37,3

Realizado por: Armijos Soledad,2021

### 4.3.1 Cálculo de MAE Y RMSE

Para obtener el cálculo del error medio absoluto se realizó mediante la siguiente ecuación 1-4:

**Ecuación 1-4:** Error medio absoluto.

$$MAE = \frac{1}{N} \sum_{n=1}^N |T_n - REF_n|$$

Que es la diferencia de la medición del prototipo con el termómetro, tomando todas esas mediciones realizadas se suman y se dividen para el número de mediciones evaluadas, se tomó estas muestras en espacio de tiempo de 10 segundos, durante 300 segundos.

Una vez que se realizó el cálculo anterior, se analizó el RMSE mediante la ecuación 2-4 indicada a continuación:

**Ecuación 2-4:** Error medio relativo cuadrático.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (T_n - REF_n)^2}$$

La misma que es la diferencia de los valores tomados elevadas al cuadrado, se dividen para el número de mediciones realizadas en este caso se divide para 30, y el resultado será la raíz cuadrada de este valor, para ellos se realizó el cálculo del MAE Y RMSE de cada fase con las ecuaciones mencionadas, se evaluó los valores en la unidad de grados centígrados cada uno. Estos cálculos se demuestran en la tabla 5-4.

En la tabla 5-4 se muestra el cálculo de los errores medios absolutos y relativos tomados en cada fase de prueba realizada anteriormente.

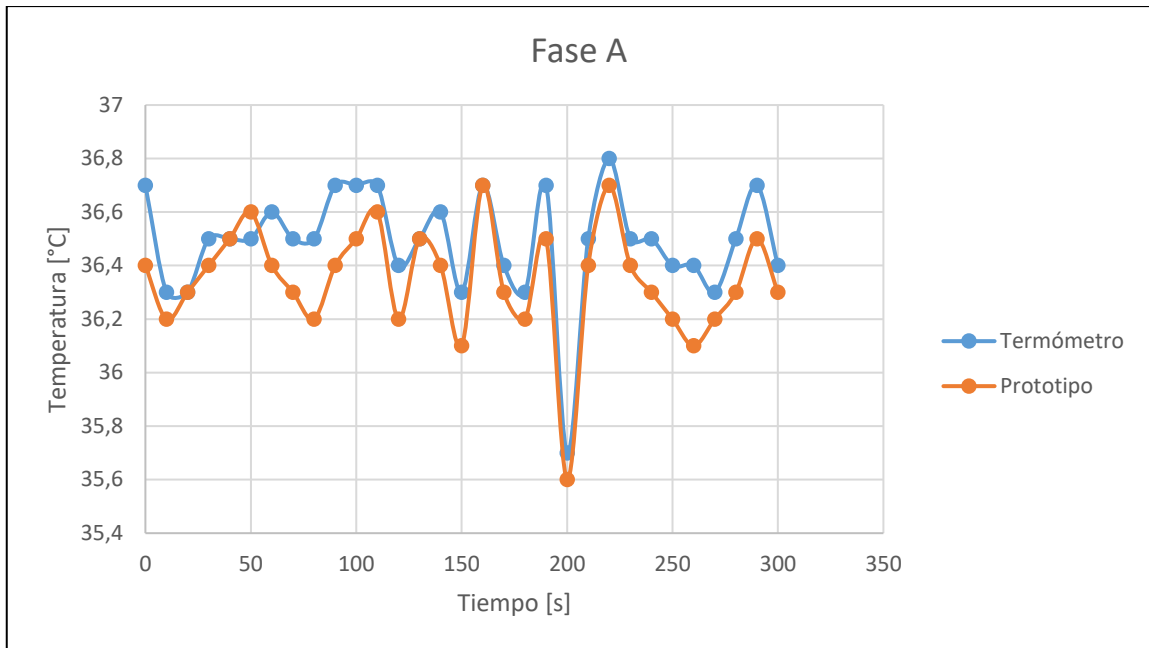
**Tabla 5-4:** Cálculo de los errores medios absolutos y relativos de cada fase.

MUESTRAS	Fase A		Fase B		Fase C	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
1	0,3	0,09	0,1	0,01	0,3	0,09
2	0,1	0,01	0,1	0,01	0,1	0,01
3	0	0	0,1	0,01	0	0
4	0,1	0,01	0,2	0,04	0,1	0,01
5	0	0	0,1	0,01	0	0

6	0,1	0,01	0,2	0,04	0,1	0,01
7	0,2	0,04	0,3	0,09	0,2	0,04
8	0,2	0,04	0,2	0,04	0,2	0,04
9	0,3	0,09	0,2	0,04	0,3	0,09
10	0,3	0,09	0,1	0,01	0,3	0,09
11	0,2	0,04	0,3	0,09	0,2	0,04
12	0,1	0,01	0,3	0,09	0,1	0,01
13	0,2	0,04	0,1	0,01	0,1	0,01
14	0	0	0,2	0,04	0,2	0,04
15	0,2	0,04	0,1	0,01	0,2	0,04
16	0,2	0,04	0,2	0,04	0,1	0,01
17	0	0	0,1	0,01	0,3	0,09
18	0,1	0,01	0,2	0,04	0,2	0,04
19	0,1	0,01	0,2	0,04	0,2	0,04
20	0,2	0,04	0,2	0,04	0,2	0,04
21	0,1	0,01	0,1	0,01	0,2	0,04
22	0,1	0,01	0,2	0,04	0,2	0,04
23	0,1	0,01	0,1	0,01	0,1	0,01
24	0,1	0,01	0,2	0,04	0,1	0,01
25	0,2	0,04	0,2	0,04	0,2	0,04
26	0,2	0,04	0,1	0,01	0,1	0,01
27	0,3	0,09	0,4	0,16	0,1	0,01
28	0,1	0,01	0,1	0,01	0,1	0,01
29	0,2	0,04	0,2	0,04	0,1	0,01
30	0,2	0,04	0,1	0,01	0,2	0,04
PROMEDIO	0,15	0,17416467	0,17333333	0,18973666	0,16	0,17888544

Realizado por: Armijos Soledad,2021

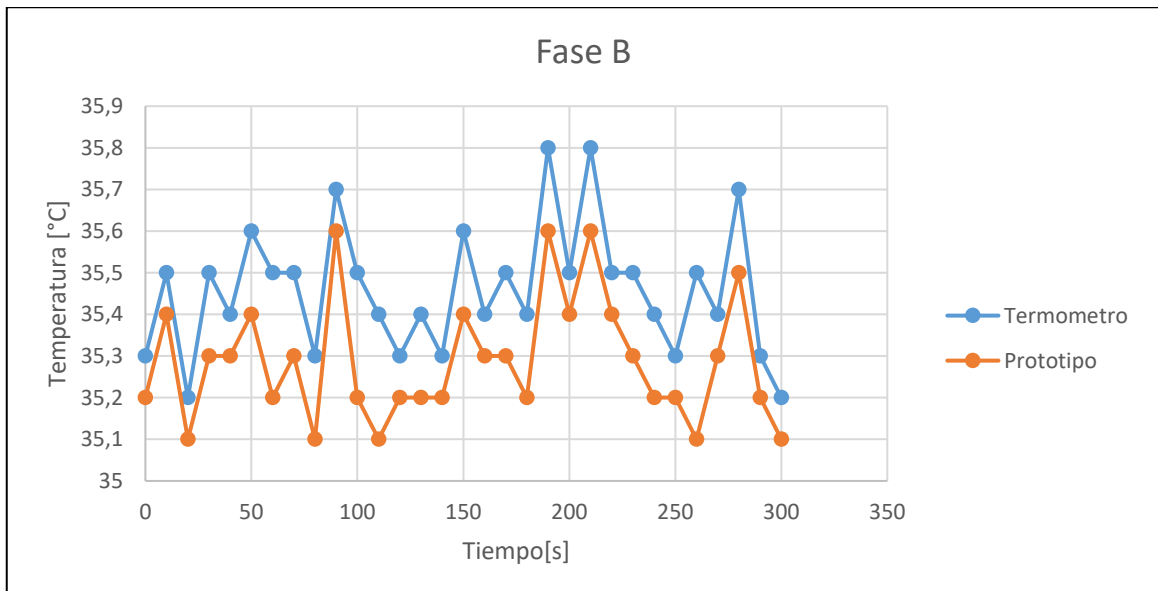
La representación del cambio de temperatura de cada fase se muestra en la gráfica 1-4 en un intervalo específico, puesto que en cada fase se tomó las medidas del niño cuando se encuentra en una temperatura corporal normal por eso se obtuvo que el cambio de temperatura se mantiene estable dentro del tiempo establecido, en la gráfica 2-4 se observa que los valores tomados del termómetro con respecto al prototipo implementado son similares por lo tanto el error calculado será mínimo.



**Gráfico 2-4:** Temperatura Fase A del termómetro vs prototipo.

Realizado por: Armijos Soledad, 2021

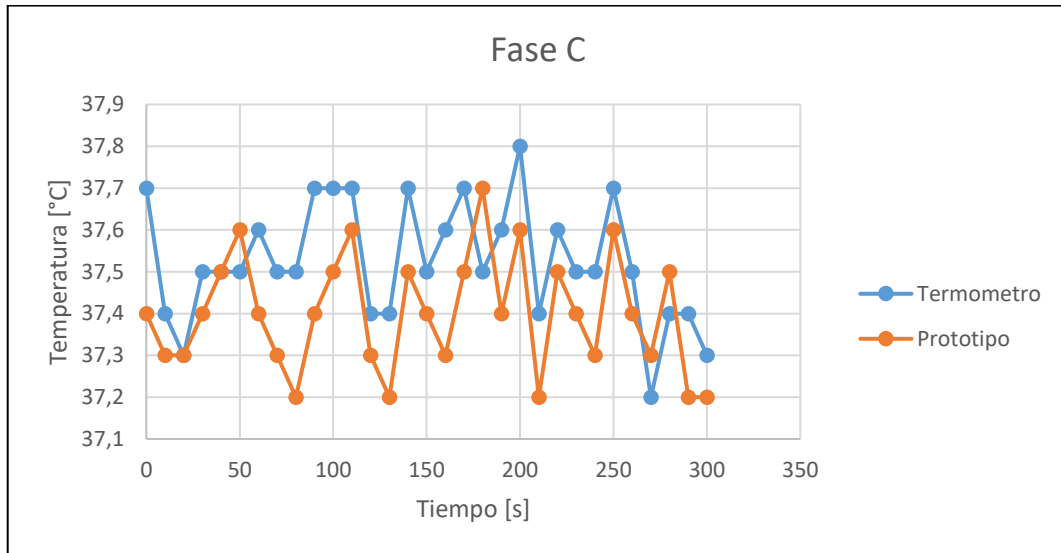
En la fase B la temperatura aumenta progresivamente ya que se bajó la temperatura corporal, se utilizó una bolsa de hielo, por lo tanto, la temperatura creció a medida que transcurrió más tiempo la misma que se hizo durante cinco minutos cada diez segundos como se muestra en la gráfica 3-4.



**Gráfico 3-4:** Temperatura Fase B del Termómetro vs Prototipo.

Realizado por: Armijos Soledad, 2021

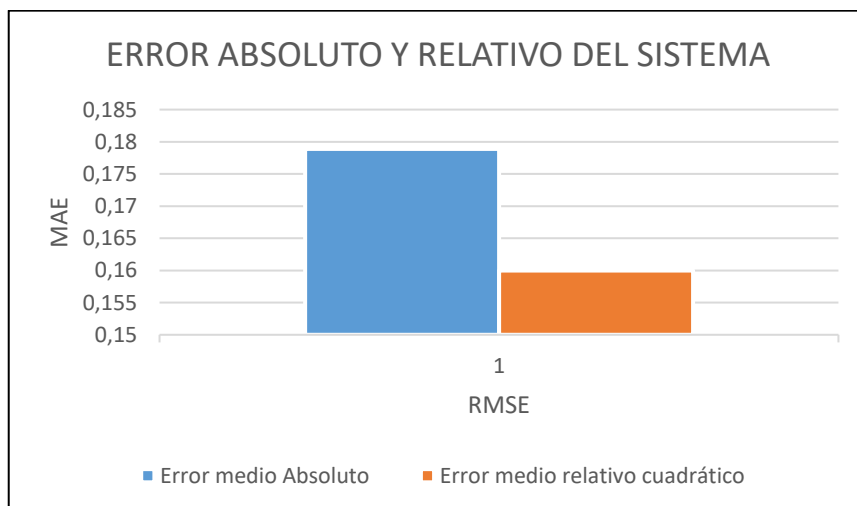
En la fase C debido a que previamente se hizo un cambio de aumento de la temperatura colocando una bolsa de agua caliente en el niño, la temperatura disminuyó con respecto al tiempo ya establecido previamente.



**Gráfico 4-4:** Temperatura Fase C del Termómetro vs Prototipo.

Realizado por: Armijos Soledad,2021

Tomando en cuenta las pruebas y cálculos realizadas del dispositivo de medición infrarroja de supervisión para niños menores a tres años con respecto al termómetro digital elegido, se obtuvo que el error medio absoluto del sistema es de 0.17889 respecto a 0.16 que es el error medio relativo cuadrático como se observa en el gráfico 5-4.



**Gráfico 5-4:** Error medio y absoluto del sistema.

Realizado por: Armijos Soledad,2021



#### 4.4 El Análisis de costos

La tabla 6-4 contiene el análisis de costos de los diferentes elementos que conforman el dispositivo se tomó en cuenta cada uno de ellos sin considerarse los elementos dañados durante la implementación.

**Tabla 6-4:** Costos del Prototipo

MATERIALES	CANTIDAD	PRECIO UNITARIO (\$)	PRECIO TOTAL (\$)
Raspberry pi 3 b, carcasa, disipador y cargador	1	125	100
Módulo de cámara raspberry pi 5MP	1	15	11
Camara Flir Lepton 2.5	1	234	234
Componentes electrónicos	-	10	10
Impresión 3D	1	20	20
<b>TOTAL</b>			<b>375</b>

Realizado por: Armijos Soledad, 2021

#### 4.5 Análisis del costo del prototipo desarrollado con los productos existentes en el mercado.

En la tabla 7-4 se muestra las distintas características de los dispositivos de control de temperatura de un niño.

**Tabla 7-4:** Ventajas y desventajas de los productos existentes en el mercado

PRODUCTOS EXISTENTES EN EL MERCADO			
PRODUCTOS	VENTAJAS	DESVENTAJAS	COSTO
<b>iTherm</b>	<p>Monitorea la temperatura del niño.</p> <p>Incluye programación de alarmas en el celular.</p> <p>Envía estadísticas a los familiares de los datos obtenidos de la temperatura.</p>	<p>Para niños mayores a cinco años de edad el dispositivo no funciona</p> <p>Para que el termómetro funcione se requiere mínimo un Android de 4.4 y el bluetooth se apaga periódicamente.</p> <p>La alarma sobre exceder la temperatura establecida no siempre funciona.</p>	150

<b>TedCare</b>	<p>Monitorea signos vitales como oxigenación en la sangre, temperatura y ritmo cardiaco.</p> <p>Se complementa con una aplicación para alertar a los padres del niño</p>	<p>Solo funciona en niños hasta 12 meses de nacido.</p> <p>La aplicación de alerta solo está disponible para Android.</p>	130
----------------	--	---	-----

**Realizado por:** Armijos Soledad,2021

Al realizar la comparativa entre los dispositivos existentes en el mercado y el prototipo implementado para el trabajo de titulación, se evidenció que el prototipo implementado se encuentra en un rango de costo mayor, pero con la ventaja que el dispositivo no se coloca en el cuerpo del niño por este motivo el infante no es propenso a sufrir cualquier tipo de molestia o inclusive algún tipo de accidente.

El prototipo implementado al colocarse a cierta distancia del niño también posee la ventaja de registrar la temperatura cuando el infante se encuentre en estado febril y dar una alerta de mensaje de texto al celular de la persona responsable del niño.

## CONCLUSIONES

Dentro de los dispositivos de monitoreo infrarrojo existentes en el mercado se encuentran algunos con características similares como el TedCare, termómetros digitales o de mercurio que, aunque el rango de precios de estos dispositivos es menor con respecto al implementado estos poseen desventajas como: incomodidad en el uso diario, posibilidad de algún accidente y falta de señal de alarma.

El dispositivo electrónico de monitoreo infrarrojo implementado en este proyecto cumple con los requerimientos establecidos previamente: la detección de temperatura utilizando algoritmos de visión artificial, el envío de una señal de alarma mediante un mensaje de texto a la persona encargada del niño, el tamaño y la facilidad del prototipo para ser ubicado en diferentes zonas donde se encuentre el menor de edad.

El dispositivo se construyó principalmente por dos cámaras, la cámara del módulo Raspberry Pi y la Cámara Flir Lepton 2.5 que permiten la captura de imagen RGB y térmica respectivamente, junto con el microcontrolador Raspberry Pi 3 que controla el sistema mediante el software de lenguaje de programación Python donde se realizó la programación de los datos analizados para lograr la detección de temperatura cuando el niño se encuentre en estado febril.

Al analizar los datos calculados en las pruebas se deduce que el prototipo cumple con los requerimientos que se propuso en el diseño, puesto que se ejecutaron las pruebas sin contratiempos debido a que trabajar con imágenes térmicas brinda beneficios porque estas no se ven afectadas por la variación de iluminación y en el cálculo final de las pruebas muestra que el error absoluto medio general (MAE) y el error cuadrático del sistema (RMSE) son  $0.17886^{\circ}\text{C}$  y  $0.16^{\circ}\text{C}$  respectivamente, es decir el error calculado es mínimo.

## **RECOMENDACIONES**

Se recomienda que el dispositivo se encuentre a una distancia máxima de 30 centímetros del menor de edad para el correcto análisis de datos y funcionamiento del prototipo.

Se recomienda el desarrollo de un algoritmo de aprendizaje automático que determine la región de interés únicamente a partir de la imagen de la cámara térmica. Esto simplificaría el diseño del hardware haciendo que la cámara RGB no sea necesaria.

Es importante el desarrollo del prototipo en FPGA para poder mejorar la velocidad de procesamiento y que de esta forma el prototipo pueda medir en tiempo real. Esto abriría aplicaciones para su uso en entornos donde se requiera monitorear temperatura constantemente, como en aeropuertos u hospitales.

Se recomienda la expansión del prototipo para determinar otros signos vitales para poder implementar un sistema de triaje automático para hospitales. Esto puede mejorar la atención en centros de salud y hospitales haciéndola más eficiente

## GLOSARIO

**Algoritmo:** Conjunto detallado y lógico de pasos para alcanzar un objetivo o resolver un problema.(Sataloff,, Johns, y Kost, 2016)

**Arquitectura:** Diseño y la estructura de un sistema.(Oxford University Press, 2021a)

**Gradiente:** La velocidad a la que la temperatura, la presión, etc. cambia, o aumenta y disminuye, entre una región y otra.(Oxford University Press, 2021b)

**Histograma:** Diagrama que consta de rectángulos cuya área es proporcional a la frecuencia de una variable y cuyo ancho es igual al intervalo de clase.(Oxford University Press, 2021c)

**Interfaz Gráfica:** También llamada GUI. Es una forma de dar instrucciones a una computadora usando cosas que se pueden ver en la pantalla, como símbolos y menús.(Oxford University Press, 2021d)

**Infrarrojo:** Tener o usar ondas electromagnéticas que son más largas que las de la luz roja en el espectro y que no se pueden ver.(Oxford University Press, 2021e)

**Protocolo:** Un conjunto de reglas que controlan la forma en que se envían los datos entre computadoras. (Oxford University Press, 2021f)

**Prototipo:** El primer diseño de algo a partir del cual se copian o desarrollan otras formas. (Oxford University Press, 2021g)

**String:** Serie de caracteres.(Oxford University Press, 2021h)

## BIBLIOGRAFIA

**AGORA, T.**, *Manual de usuario. Business* [en línea]. Monterrey-México: 2009 , [Consulta: 10 febrero 2021]. Disponible en: [https://issuu.com/kherian.com/docs/manual\\_agora\\_2009\\_esp](https://issuu.com/kherian.com/docs/manual_agora_2009_esp)

**ALARCÓN, A , ARIAS, G, LLANOS, J..**, *Sistema de medición de temperatura corporal a través de dispositivos móviles para monitorear niveles de fiebre en niños entre los 0 y 5 años en la ciudad de Neiva.* [blog]. [Consulta: 10 febrero 2021]. Disponible en: <https://revistas.utp.ac.pa/index.php/memoutp/article/download/1838/html?inline=1>.

**ALCÁNTARA-MONTIEL, C.V., PEDRAZA-ORTEGA, J.C., RAMOS-ARREGUÍN, J.M., GORROSTIETA-HURTADO, E., TOVAR-ARRIAGA, S. y VARGAS-SOTO, J.E.**, "Detección efectiva de rostros en imágenes utilizando descriptores basados en HOG". *Research in Computing Science* [en línea], 2019, (México) 148(8), pp. 371-385. [Consulta: 20 de septiembre 2020]. ISSN 1870-4069. DOI 10.13053/rcs-148-8-28. Disponible en: [https://www.rcs.cic.ipn.mx/2019\\_148\\_8/Deteccion%20efectiva%20de%20rostros%20en%20imágenes%20utilizando%20descriptores%20basados%20en%20HOG.pdf](https://www.rcs.cic.ipn.mx/2019_148_8/Deteccion%20efectiva%20de%20rostros%20en%20imágenes%20utilizando%20descriptores%20basados%20en%20HOG.pdf)

**BAŠKYS, A.**, *Microcontrollers* [blog]. [Consulta: 12 de agosto 2020]. Disponible en: [http://dspace.vgtu.lt/bitstream/1/1381/1/1339\\_Baskys\\_Microcontrllers\\_Microcontrollers\\_web.pdf](http://dspace.vgtu.lt/bitstream/1/1381/1/1339_Baskys_Microcontrllers_Microcontrollers_web.pdf)

**CAMACHO BEDOYA, Calet, FORERO SIERRA, Diego, SARMIENTO DIAZ, Hector**, 2017. *La termografía como herramienta de diagnostico predictivo para los elementos electricos conectados a la red energia.* [En línea] (Trabajo de titulación). (Tecnología). Universidad Tecnológica de Pereira, Colombia. 2017. pp.44-50. [Consulta: 2020-11-23]. Disponible en: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/7467/62136C172.pdf?sequence=1&isAllowed=y>

**CERÓN, M., GONZÁLEZ, L.R., SEGUR, F.G., CERVANTES, G.V., ALBERTO, J., VILLARREAL, E. y URBINA-MEDINA, H.**, 2010. *Fiebre en pediatría* [en línea], 201, (México) 77(1), pp. 3-8. [Consulta: 17 octubre 2020]. Disponible en: <https://www.medigraphic.com/pdfs/pediat/sp-2010/sps101b.pdf>

**COLGREN, R.**, *Introduction to MATLAB. Basic MATLAB®, Simulink®, and Stateflow®.*, Restos, Estados Unidos: American Institute of Aeronautics & Astronautics. pp. 1-42.

**DIAKIDES, Mary; et al.**, *Medical infrared imaging: principles and practices* [en línea]. Florida-USA: TCR Press, Tylor& Francis Group, 2018. [Consulta: 18 de diciembre 2020]. Disponible en: <https://books.google.com.ec/books?hl=es&lr=&id=FzLOBQAAQBAJ&oi=fnd&pg=PP1&dq=Medical+Infrared+Imaging:+Principles+and+Practices&ots=KWxTU4Shdk&sig=ckMdxoeuZ4EksL7rr8GMoBoVVrQ#v=onepage&q&f=false>

**NEITA DUARTE, Lidia Yaneth, & PEÑA RODRÍGUEZ, Erick Omar**, principios básicos de la termografía infrarroja y su utilización como técnica para mantenimiento predictivo. [En línea] (Trabajo de titulación)(Ingeniería). Uniersidad Pontificia Bolivariana, Medellín - Colombia. 2011. *Repositorio Institucional UPB* [en línea], vol. 126, no. 3, pp. 221. [Consulta: 10 febrero 2021]. ISSN 01852698. Disponible en: [http://biblioteca.upbbga.edu.co/docs/digital\\_20999.pdf](http://biblioteca.upbbga.edu.co/docs/digital_20999.pdf)

**EL EMPRESARIO**. *TedCare, pulsera para monitorear los signos vitales del bebé*. [blog]. [Consulta: 10 febrero 2021]. Disponible en: <https://elempresario.mx/emprendedores/tedcare-pulsera-monitorear-los-signos-vitales-bebe>.

**ERAZO CHULDE, Juan Carlos & VICENTE ROMERO, Kleber Reynaldo**. Diseño Y Construcción De Un Módulo Inalámbrico Para La Adquisición Y Procesamiento, Entiempo Real, De Las Variables Fisiológicas Requeridas En Atención Prehospitalaria. (Trabajo de titulación) (Ingeniería). escuela Superior Politécnica de Chimborazo, Riobamba - Ecuador. 2009. pp. 1-96.

**FERNÁNDEZ RODRÍGUEZ, Rubén**. Tema 9: Calor y Temperatura. [blog]. [Consulta: 13 febrero 2021]. Disponible en: [https://www.apuntesmareaverde.org.es/grupos/cn/Temas\\_2/T9\\_2ESO\\_Calor\\_Temperatura\\_v2016.pdf](https://www.apuntesmareaverde.org.es/grupos/cn/Temas_2/T9_2ESO_Calor_Temperatura_v2016.pdf).

**FLIR**, 2018. FLIR LEPTON Engineering Datasheet [blog]. Document Number: 500-0659-00-09. Rev: 201. , no. August, pp. 1-118. Disponible en: <https://www.cornestech.co.jp/tech/wp-content/uploads/sites/2/2018/06/500-0659-00-09-Lepton-Engineering-Datasheet-Rev201.pdf>

**IDEAM**. Protocolo de procesamiento digital de imágenes para la cuantificación DE LA deforestación en Colombia nivel subnacional-escala gruesa y fina-26 5.1.2 calibración radiométrica y corrección atmosférica [blog]. [Consulta: 8 febrero 2021]. Disponible en: [http://www.ideam.gov.co/documents/13257/13817/Protocolo\\_para\\_la\\_cuantificaci%C3%B3n\\_Deforestaci%C3%B3n\\_Nivel\\_Nacional.pdf](http://www.ideam.gov.co/documents/13257/13817/Protocolo_para_la_cuantificaci%C3%B3n_Deforestaci%C3%B3n_Nivel_Nacional.pdf)

**INFAIMON.** Historia de la visión artificial: así ha evolucionado esta tecnología. [blog]. [Consulta: 8 abril 2021]. Disponible en: <https://blog.infaimon.com/historia-vision-artificial/>.

**JAVATPOINT,** 2011. Webcam | What is a webcam - javatpoint. [blog]. [Consulta: 12 abril 2021]. Disponible en: <https://www.javatpoint.com/webcam>.

**LIN, Jia-Wei; et al.** A thermal camera based continuous body temperature measurement system. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pp. 1681-1687. DOI 10.1109/ICCVW.2019.00208. Disponible en: [https://openaccess.thecvf.com/content\\_ICCVW\\_2019/papers/CVPM/Lin\\_A\\_Thermal\\_Camera\\_Based\\_Continuous\\_Body\\_Temperature\\_Measurement\\_System\\_ICCVW\\_2019\\_paper.pdf](https://openaccess.thecvf.com/content_ICCVW_2019/papers/CVPM/Lin_A_Thermal_Camera_Based_Continuous_Body_Temperature_Measurement_System_ICCVW_2019_paper.pdf)

**LLAMAS, Luis.** Modelos y características de Raspberry Pi [blog]. [Consulta: 14 abril 2021]. Disponible en: <https://www.luisllamas.es/modelos-de-raspberry-pi/>.

**LOGITECH.** Beyond Built-in : Why a Better Webcam Matters How to Uplevel Your Ability to Connect , Communicate and Collaborate Using Your Laptop or PC Beyond Built-in : Why a Better Webcam Matters. [blog]. [Consulta: 14 enero 2021]. Disponible en: <http://www.logitech.com/assets/41349/logitech--why-a-better-webcam-matters.ENG.pdf>.

**LOZANO, Karem.** Diseño de un sistema no invasivo de medición de la temperatura corporal interna. [blog]. [Consulta: 30 de enero 2021]. Disponible en: <https://upcommons.upc.edu/handle/2099.1/25792?show=full>.

**LUQUE ORDÓÑEZ, Javier.** Espectro electromagnético y espectro radioeléctrico [blog]. [Consulta: 30 de noviembre 2020]. Disponible en: [https://www.acta.es/medios/articulos/ciencias\\_y\\_tecnologia/062017.pdf](https://www.acta.es/medios/articulos/ciencias_y_tecnologia/062017.pdf)

**MARK, L.** *Learning Python* [en línea]. California - USA. O'Reilly Media, Inc, 2009. [Consulta: 25 de noviembre 2020]. Disponible en: [https://cfm.ehu.es/ricardo/docs/python/Learning\\_Python.pdf](https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf)

**MATT.** 8-megapixel Raspberry Pi Camera Module v2 - Raspberry Pi Spy. [blog]. [Consulta: 15 febrero 2021]. Disponible en: <https://www.raspberrypi-spy.co.uk/2016/04/8-megapixel-raspberry-pi-camera-module-v2/>.

**MELGOSA, S.,** 2011. Guia Termografía, Aplicaciones En Ahorro Energético [bog]. [Consulta:



8 de diciembre 2020]. Disponible en: <https://www.fenercom.com/wp-content/uploads/2011/11/Guia-de-la-Termografia-Infrarroja-Aplicaciones-en-Ahorro-y-Eficiencia-Energetica-fenercom-2011.pdf>

**MERRIAM-WEBSTER DICTIONARY.** Camera | Definition of Camera by Merriam-Webster. [blog]. [Consulta: 12 abril 2021]. Disponible en: <https://www.merriam-webster.com/dictionary/camera>.

**MONK, S.** *Monk Simon-Programming Arduino\_ Getting Started With Sketches-Tab Books (2011)*. S.l.: s.n. ISBN 9780071784238.

**NAYYAR, A. y PURI, V.** "Raspberry Pi-A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review International Journal of Advanced Research in Raspberry Pi- A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review". *International Journal of Advanced Research in Computer Science and Software Engineering 5(12)*, vol. 5, no. 12, pp. 720-737.

**OXFORD UNIVERSITY PRESS.** Abreviatura de gui - Definición, imágenes, pronunciación y notas de uso | Diccionario americano avanzado de Oxford en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: [https://www.oxfordlearnersdictionaries.com/us/definition/american\\_english/gui#gui\\_\\_17](https://www.oxfordlearnersdictionaries.com/us/definition/american_english/gui#gui__17).

**OXFORD UNIVERSITY PRESS.** Arquitectura sustantivo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/architecture?q=architecture>.

**OXFORD UNIVERSITY PRESS.** Gradiente sustantivo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/gradient?q=gradient>.

**OXFORD UNIVERSITY PRESS.** Histograma sustantivo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/histogram?q=histogram>.

**OXFORD UNIVERSITY PRESS.** Infrared adjective - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/infrared?q=infrared>.

**OXFORD UNIVERSITY PRESS.** Protocol sustantivo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/protocol?q=protocol>.

**OXFORD UNIVERSITY PRESS.** String\_1 sustantivo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: [https://www.oxfordlearnersdictionaries.com/us/definition/english/string\\_1?q=string](https://www.oxfordlearnersdictionaries.com/us/definition/english/string_1?q=string).

**OXFORD UNIVERSITY PRESS.** Sustantivo prototipo - Definición, imágenes, pronunciación y notas de uso | Diccionario Oxford Advanced Learner's en OxfordLearnersDictionaries.com. [blog]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.oxfordlearnersdictionaries.com/us/definition/english/prototype?q=prototype>.

**MCCOLL, P.; et al.** "Comparación de valores de temperatura obtenidos con termómetro óptico digital y con termómetros axilar y rectal de mercurio en menores de 5 años". Revista chilena de pediatría [en línea], 2013, (Chile). 84 (3), pp. 293-299. [Consulta: 11 de diciembre 2020]. ISSN 0370-4106. Disponible en: [https://scielo.conicyt.cl/scielo.php?script=sci\\_abstract&pid=S0370-41062013000300007&lng=es&nrm=iso](https://scielo.conicyt.cl/scielo.php?script=sci_abstract&pid=S0370-41062013000300007&lng=es&nrm=iso)

**PETERS, J.F.,** 2017. Foundations of Computer Vision. , no. March 2017, pp. 431. DOI 10.1007/978-3-319-52483-2.

**ROSAS, Israel.** iTherm, un brazalete para seguir la temperatura de niños. [blog]. [Consulta: 10 febrero 2021]. Disponible en: <https://www.fayerwayer.com/2014/04/itherm-brazalete-temperatura-ninos/>.

**RUIZ GUTI, M.** Manual de Programación Arduino Arduino : Manual de Programación [en línea].California - USA: Creative Commons, 2007 *Arduino Notebook*, vol. 1, pp. 3-70. [Consulta: 12 de noviembre 2020]. Disponible en: <https://arduinoobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>

**SAAD D, Muris; et al.** "Conocimientos sobre fiebre y conductas de los residentes del Postgrado de pediatría ante un paciente febril". *Archivos Venezolanos de Puericultura y Pediatría*, vol. 69, no. 3, pp. 96-102. ISSN 0004-0649.

**SANZ, F.J, RAMOS, J.T, PELAYO, F.,** 2009. Síndrome febril sin foco en pediatría. *ECG Fácil*, vol. 7, no. 4, pp. 135-145. DOI 10.1016/b978-84-8086-476-3.50006-5.

**SATHIYAMOORTHY, S.** "Industrial Application of Machine Vision". *International Journal of Research in Engineering and Technology*, vol. 03, no. 19, pp. 678-682. ISSN 23217308. DOI 10.15623/ijret.2014.0319120.

**SOLARI, F. ; et al.** *Machine Vision - Applications and Systems* [en línea]. Rijeka - Croatia: Intech, 2012. [Consulta: 05 de diciembre 2020].S.l.: s.n. ISBN 9789535103738. Disponible en: [http://www.eyeshots.it/papers/eyeshots\\_res\\_pub/SolariA12.pdf](http://www.eyeshots.it/papers/eyeshots_res_pub/SolariA12.pdf)

**SPARKFUN ELECTRONICS.** Hoja de datos de la guía de conexión de FLIR Lepton de SparkFun Electronics | Electrónica Digi-Key. [blog]. [Consulta: 17 febrero 2021]. Disponible en: <https://www.digikey.be/htmldatasheets/production/3447286/0/0/1/flir-lepton-hookup-guide.html>.

**TESTO.** Guía de bolsillo Teoría – Aplicación Práctica – Consejos y Trucos. , vol. 1, pp. 7,8,9.

**TUTORIALSPPOINT,** 2019. About the Tutorial Copyright & Disclaimer. *Tutorials Point (I) Pvt. Ltd.*, pp. 1-13.

**VAN ROSSUM, G. & PYTHON DEVELOPMENT TEAM.** Python Tutorial Release 3.7.0. *Python Software Foundation*, pp. 1-155.

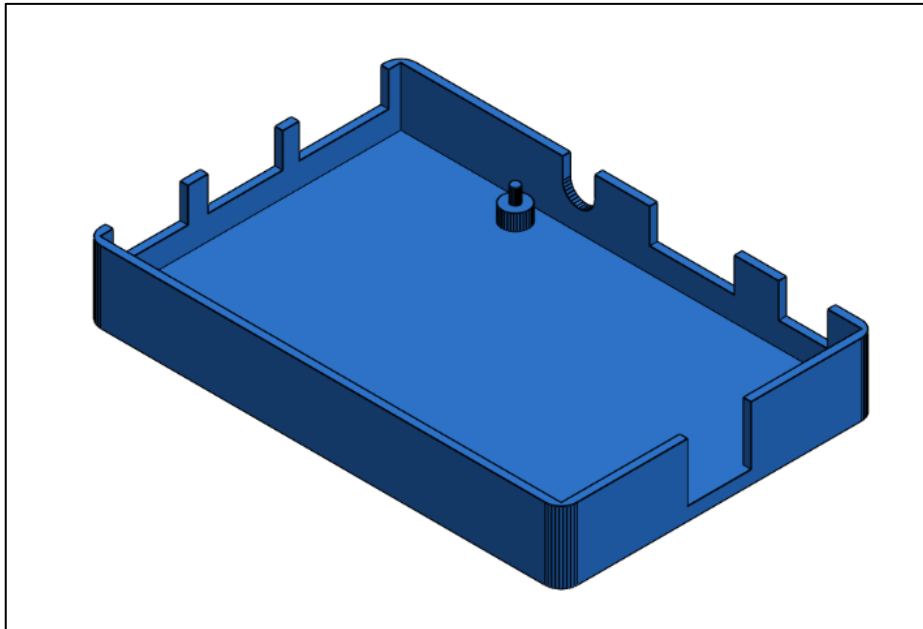
**VEJARANO, C.,** 2019. docs / colorize\_all\_examples.md · master · Código abierto / cmapy · GitLab. [blog]. [Consulta: 19 febrero 2021]. Disponible en: [https://gitlab.com/cvejarano-oss/cmapy/blob/master/docs/colorize\\_all\\_examples.md](https://gitlab.com/cvejarano-oss/cmapy/blob/master/docs/colorize_all_examples.md).

**WONG, Wai K.; et al.** "An effective surveillance system using thermal camera". *International Conference on Signal Acquisition and Processing, ICSAP 2009*, no. April, pp. 13-17. DOI 10.1109/ICSAP.2009.12.

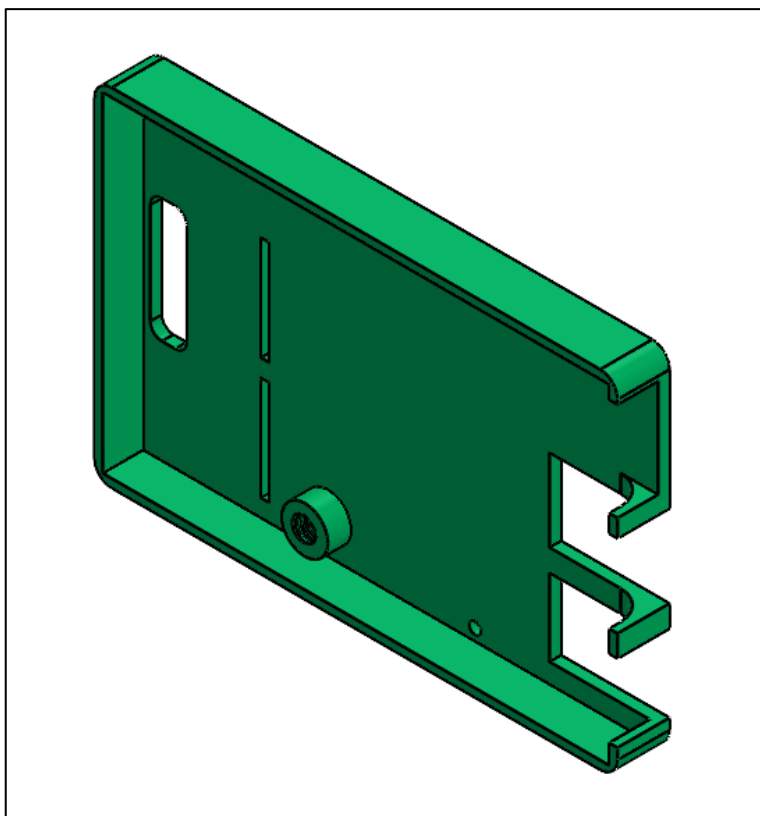
**ZOHDY, Bassem S.; et al.** "Machine Vision Application on Science and Industry".  
*Optoelectronics in Machine Vision-Based Theories and Applications*no. August, pp. 233-254.  
DOI 10.4018/978-1-5225-5751-7.ch008.

## ANEXOS

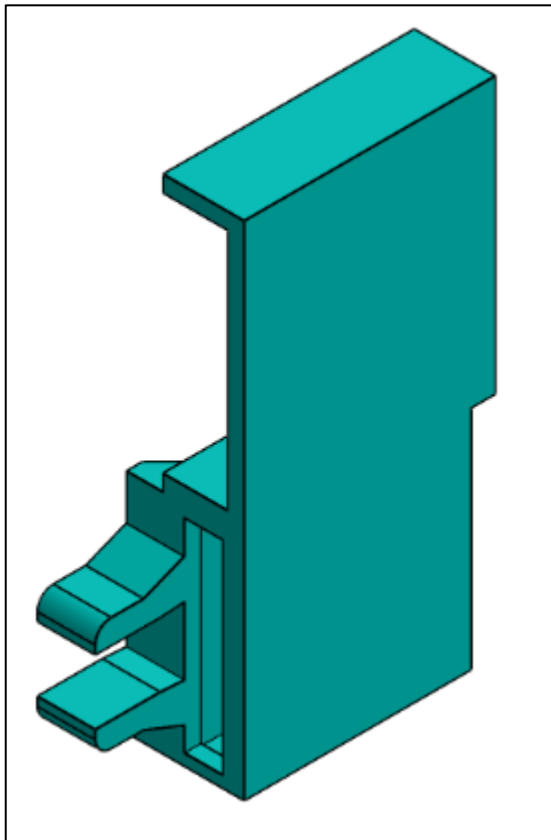
**Anexo A:** Base inferior en SolidWorks de la estructura del prototipo.



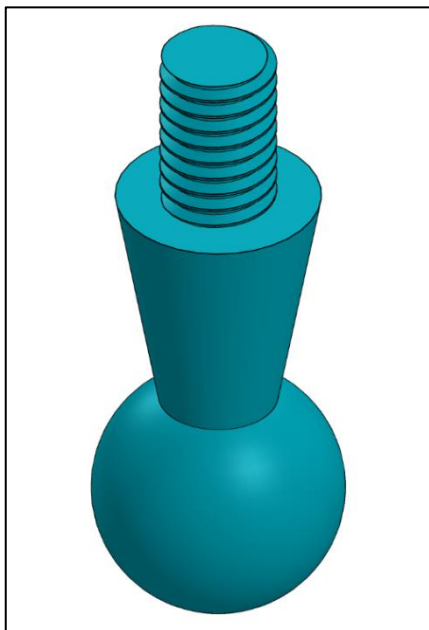
**Anexo B:** Base superior en solidworks de la estructura del prototipo.



**Anexo C:** Estructura de la cámara RGB y Térmica del prototipo.



**ANEXO D:** Soporte para la cámara RGB del prototipo.

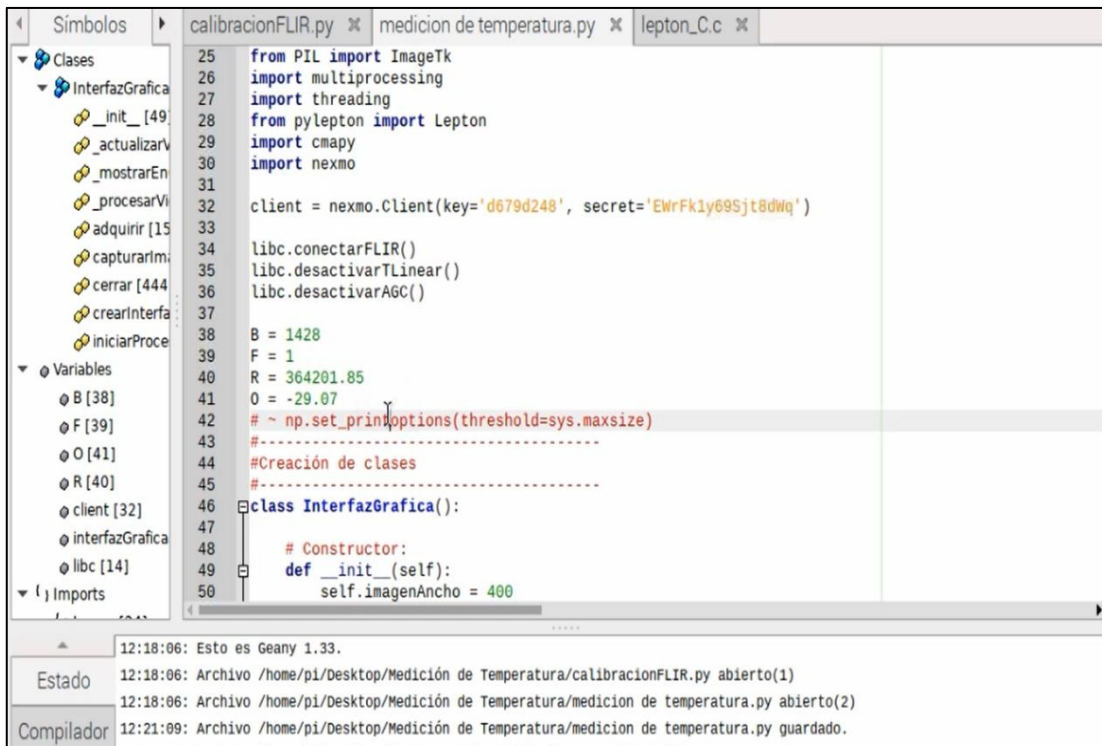


## Anexo E: Código de adquisición de video de ambas cámaras.

```
calibracionFLIR.py x medicion de temperatura.py x lepton_C.c x
342     img = cv2.hconcat([cv2.cvtColor(self.imgRGB, cv2.COLOR_RGB2BGR),
343                       cv2.cvtColor(self.imgTermica, cv2.COLOR_RGB2BGR)])
344     cv2.imwrite(nombreArchivo, img)
345     msg.showinfo('Captura', f'Imagen guardada como {nombreArchivo}')
346
347
348     # Método para iniciar la adquisición de video de ambas cámaras:
349     def _actualizarVideoGUI(self, terminal1):
350         """ Inicia un hilo para adquirir los frames. También se encarga
351             de mostrar los frames adquiridos en la GUI o de mostrar los
352             frames procesados """
353         print("-----THREAD STARTED-----")
354         self.camaraRGB = cv2.VideoCapture(0)
355
356         while True:
357             ret, frameRGB = self.camaraRGB.read()
358             if ret:
359                 with Lepton() as camaraTermica:
360                     # ~ print('THREAD: TRUE')
361                     frameTermica, _ = camaraTermica.capture() #(60, 80, 1)
362                     frameTermica = frameTermica[:,0] #(60, 80)
363                     frameT = frameTermica.copy() #para análisis
364
365                     cv2.normalize(frameTermica, frameTermica, 0, 65535, cv2.NORM_MINMAX) # extend
366                     frameTermica = np.right_shift(frameTermica, 8, frameTermica) # fit data into 8
367                     frameTermica = np.uint8(frameTermica)
```

```
calibracionFLIR.py x medicion de temperatura.py x lepton_C.c x
368         frameTermica = cv2.applyColorMap(frameTermica, cmapy.cmap('CMRmap'))
369
370         i = i+1
371
372         if self.medicion == True:
373             self.labelTemperatura.configure(text = 'Procesando...')
374             # ~ time.sleep(5)
375             listaParaEnviar = [frameRGB, frameT, True]
376             start = time.perf_counter()
377             terminal1.send(listaParaEnviar)
378             # ~ terminal1.close()
379             listaRecibida = terminal1.recv()
380             # Código para obtener los frames procesados
381             #---
382             self.labelTemperatura.configure(text = 'Procesado')
383             frameRGB = listaRecibida[0]
384             frameTermica = listaRecibida[1]
385             temperatura = listaRecibida[2]
386             end = time.perf_counter()
387             print('Tiempo transcurrido: {:.4f}s'.format(end-start))
388             """
389             Solo para los resultados
390             """
391             # ~ frameRGB = cv2.imread('pruebas/resultadoRGB2.jpg')
392             # ~ frameTermica = cv2.imread('pruebas/resultadoTermica2.jpg')
393             temperatura = 36.8 + np.random.rand()
```

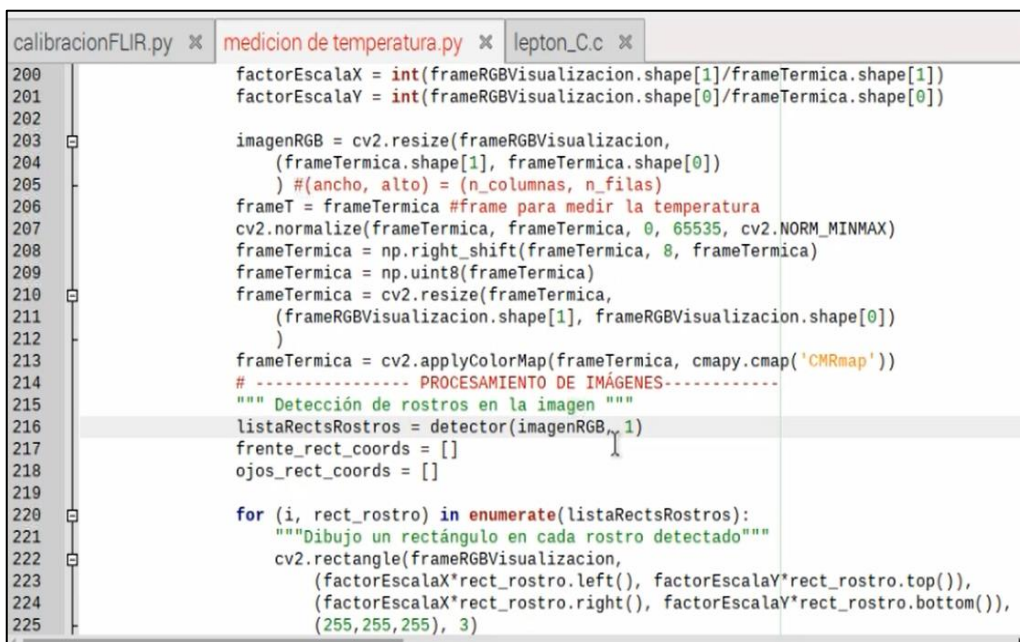
## Anexo F: Código en Python del mensaje de texto a través de GSM.



```
25 from PIL import ImageTk
26 import multiprocessing
27 import threading
28 from pylepton import Lepton
29 import cmaps
30 import nexmo
31
32 client = nexmo.Client(key='d679d248', secret='EwrFk1y69Sjt8dwq')
33
34 libc.conectarFLIR()
35 libc.desactivarTLinear()
36 libc.desactivarAGC()
37
38 B = 1428
39 F = 1
40 R = 364201.85
41 O = -29.07
42 # ~ np.set_printoptions(threshold=sys.maxsize)
43 #-----
44 #Creación de clases
45 #-----
46 class InterfazGrafica():
47
48     # Constructor:
49     def __init__(self):
50         self.imagenAncho = 400
```

12:18:06: Esto es Geany 1.33.  
Estado 12:18:06: Archivo /home/pi/Desktop/Medición de Temperatura/calibracionFLIR.py abierto(1)  
12:18:06: Archivo /home/pi/Desktop/Medición de Temperatura/medicion de temperatura.py abierto(2)  
Compilador 12:21:09: Archivo /home/pi/Desktop/Medición de Temperatura/medicion de temperatura.py guardado.

## Anexo G: Código de procesamiento de imágenes.



```
200 factorEscalaX = int(frameRGBVisualizacion.shape[1]/frameTermica.shape[1])
201 factorEscalaY = int(frameRGBVisualizacion.shape[0]/frameTermica.shape[0])
202
203 imagenRGB = cv2.resize(frameRGBVisualizacion,
204                       (frameTermica.shape[1], frameTermica.shape[0])
205                       ) #(ancho, alto) = (n_columnas, n_filas)
206 frameT = frameTermica #frame para medir la temperatura
207 cv2.normalize(frameTermica, frameTermica, 0, 65535, cv2.NORM_MINMAX)
208 frameTermica = np.right_shift(frameTermica, 8, frameTermica)
209 frameTermica = np.uint8(frameTermica)
210 frameTermica = cv2.resize(frameTermica,
211                          (frameRGBVisualizacion.shape[1], frameRGBVisualizacion.shape[0])
212                          )
213 frameTermica = cv2.applyColorMap(frameTermica, cmaps.cmap('CMRmap'))
214 # ----- PROCESAMIENTO DE IMÁGENES-----
215 """ Detección de rostros en la imagen """
216 listaRectsRostros = detector(imagenRGB, 1)
217 frente_rect_coords = []
218 ojos_rect_coords = []
219
220 for (i, rect_rostro) in enumerate(listaRectsRostros):
221     """Dibujo un rectángulo en cada rostro detectado"""
222     cv2.rectangle(frameRGBVisualizacion,
223                  (factorEscalaX*rect_rostro.left(), factorEscalaY*rect_rostro.top()),
224                  (factorEscalaX*rect_rostro.right(), factorEscalaY*rect_rostro.bottom()),
225                  (255,255,255), 3)
```



```

220     for (i, rect_rostro) in enumerate(listaRectsRostros):
221         """Dibujo un rectángulo en cada rostro detectado"""
222         cv2.rectangle(frameRGBVisualizacion,
223                       (factorEscalaX*rect_rostro.left(), factorEscalaY*rect_rostro.top()),
224                       (factorEscalaX*rect_rostro.right(), factorEscalaY*rect_rostro.bottom()),
225                       (255,255,255), 3)
226         cv2.putText(frameRGBVisualizacion,
227                    f'{{(rect_rostro.left(), rect_rostro.top())}}',
228                    (factorEscalaX*rect_rostro.left()-80, factorEscalaY*rect_rostro.top()+10),
229                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
230         cv2.putText(frameRGBVisualizacion,
231                    f'{{(rect_rostro.right(), rect_rostro.top())}}',
232                    (factorEscalaX*rect_rostro.right()+10, factorEscalaY*rect_rostro.top()+10),
233                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
234         cv2.putText(frameRGBVisualizacion,
235                    f'{{(rect_rostro.left(), rect_rostro.bottom())}}',
236                    (factorEscalaX*rect_rostro.left()-80, factorEscalaY*rect_rostro.bottom()+10),
237                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
238         cv2.putText(frameRGBVisualizacion,
239                    f'{{(rect_rostro.right(), rect_rostro.bottom())}}',
240                    (factorEscalaX*rect_rostro.right()+10, factorEscalaY*rect_rostro.bottom()+10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)

```

: Archivo /home/pi/Desktop/Medición de Temperatura/medicion de temperatura.py guardado.  
: Archivo /home/pi/Desktop/Medición de Temperatura/medicion de temperatura.py guardado.

## Anexo H: Código de detección de puntos de interés en el rostro.

```

calibracionFLIR.py x medicion de temperatura.py x lepton.C.c x
240     (factorEscalaX*rect_rostro.right()+10, factorEscalaY*rect_rostro.bottom()+10),
241     cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
242
243     """ Detección de puntos de interés (ojos) en cada rostro
244     de la imagen RGB
245     """
246     puntosDeInteres = predictor(imagenRGB, rect_rostro)
247     matrizPuntosDeInteres = face_utils.shape_to_np(puntosDeInteres)
248     """Para cada conjunto de punto de interés en el diccionario "ojos" """
249     for (name, (m, n)) in ojos.items():
250         """
251         Se itera en la matrizPuntoDeInteres para obtener unicamente
252         las coordenadas de puntos de interes que contiene el diccionario
253         de ojos
254         """
255         (x, y, ancho, alto) = cv2.boundingRect(
256             np.array([matrizPuntosDeInteres[m:n]])
257         )
258         if name == 'right_eye':
259             color = (255, 0, 0)
260             frente_rect_coords.append((x, y+alto))
261         if name == 'left_eye':
262             color = (0, 255, 0)
263             frente_rect_coords.append((x+ancho, frente_rect_coords[0][1]))
264
265     cv2.rectangle(frameRGBVisualizacion, (x*factorEscalaX, y*factorEscalaY),

```

```

265         cv2.rectangle(frameRGBVisualizacion, (x*factorEscalaX, y*factorEscalaY),
266                     (factorEscalaX*(x+ancho), factorEscalaX*(y+alto)), color, 2)
267         """Coordenadas de la ROI en la imagen RGB"""
268         xi = frente_rect_coords[0][0]
269         yi = int(frente_rect_coords[0][1]*0.5)
270         xf = frente_rect_coords[1][0]
271         yf = int(frente_rect_coords[1][1]*1.1)
272         """Dibujo de la ROI en la imagen RGB para visualización"""
273         cv2.rectangle(frameRGBVisualizacion,
274                     (factorEscalaX*xi, factorEscalaY*yi),
275                     (factorEscalaX*xf, factorEscalaY*yf),
276                     (0, 0, 255), 2)
277         cv2.putText(frameRGBVisualizacion, f'{{xi, yi}}',
278                   (factorEscalaX*xi -20, factorEscalaY*yi -10),
279                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
280         cv2.putText(frameRGBVisualizacion, f'{{xf, yi}}',
281                   (factorEscalaX*xf, factorEscalaY*yi-10),
282                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
283         cv2.putText(frameRGBVisualizacion, f'{{xi, yf}}',
284                   (factorEscalaX*xi-20, factorEscalaY*yf+20),
285                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
286         cv2.putText(frameRGBVisualizacion, f'{{xf, yf}}',
287                   (factorEscalaX*xf, factorEscalaY*yf+20),
288                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
289         """Coordenadas de la ROI en la imagen Térmica"""
290         xi_T = int(xi*1.2)

```

### Anexo I: Código de medición de temperatura en la ROI de imagen térmica.

```

calibracionFLIR.py x medicion de temperatura.py x lepton_C.c x
299         cv2.putText(frameTermica, f'{{xi_T, yi_T}}',
300                   (factorEscalaX*xi_T -20, factorEscalaY*yi_T -10),
301                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
302         cv2.putText(frameTermica, f'{{xf_T, yi_T}}',
303                   (factorEscalaX*xf_T, factorEscalaY*yi_T-10),
304                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
305         cv2.putText(frameTermica, f'{{xi_T, yf_T}}',
306                   (factorEscalaX*xi_T-20, factorEscalaY*yf_T+20),
307                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
308         cv2.putText(frameTermica, f'{{xf_T, yf_T}}',
309                   (factorEscalaX*xf_T, factorEscalaY*yf_T+20),
310                   cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
311
312         """Medición de la temperatura en la ROI de la imagen térmica"""
313         ROI_termica = frameT[yi_T:yf_T, xi_T:xf_T]
314         pixelesPromedio = ROI_termica.flatten().mean()
315
316         T = B/np.log(R/(pixelesPromedio - 0) + F) - 273.15
317         # ~ print('{:.2f}°C'.format(T))
318
319         #-----
320
321         """
322         Se envía los frames procesados y la temperatura al
323         proceso principal para mostrar en la GUI
324         """

```