



**UNIVERSIDAD ESTATAL  
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

Propuesta Tecnológica, previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN.**

“Implementación de un dispositivo electrónico de monitoreo a través de visión artificial que permita realizar mediciones de frecuencia respiratoria no invasiva”

**AUTOR**

MARIO ENRIQUE SALCEDO TERAN.

**PROFESOR TUTOR**

ING. LUIS CHUQUIMARCA JIMÉNEZ. Msc.

LA LIBERTAD – ECUADOR

2023.

## AGRADECIMIENTO

A Dios por darme fortaleza y sabiduría, para no decaer y así lograr culminar con mis estudios.

A mi madre Rosa Teran, por ser mi motor a seguir, inculcándome buenos valores y enseñanzas, por siempre confiar en mí y apoyarme durante toda mi carrera. Sin ti no hubiera sido posible llegar hasta aquí. Gracias.

A mis hermanos Nelly, Bernardo, Sammy y José, por ser siempre fuente de inspiración, brindándome consejos, amor y paciencia que me impulsaron a concluir con esta anhelada carrera universitaria.

A Camila González, por ser mi compañera incondicional, apoyándome en los momentos más difíciles, para nunca rendirme, este logro también es tuyo.

A mi tía Rosario Teran, por brindarme su apoyo incondicional, y permitirme encontrar en ella otra madre. Gracias.

A mi cuñado Rubén por brindarme su apoyo durante toda mi carrera, de no ser por usted no hubiera podido continuar.

Al resto de mi familia por siempre darme su apoyo y transmitirme seguridad durante todo este tiempo.

A Segundo González y Lourdes Bernardino por abrirme sus brazos y recibirme como un hijo más, son personas dignas de admirar y gracias por creer en mí.

A mis tres amigos Steven Tigrero, María Tomalá y Kevin Malan, con los cuales compartí grandes experiencias enriquecedoras de conocimiento.

A todos mis docentes, en especial a mi tutor de tesis Ing. Luis Chuquimarca, Msc., por la paciencia entregada y brindarme conocimientos para sacar adelante este proyecto.

**Mario Enrique Salcedo Teran.**

## DEDICATORIA

Dedico el presente proyecto de titulación a toda mi familia, quienes con su amor y comprensión me transmitieron la seguridad de seguir adelante.

A mi madre quien, con su esfuerzo y amor, ha dedicado a formarme con buenos principios y valores. Te amo viejita.

A mi hermana Nelly quien, con su apoyo y buenos consejos, me permitieron seguir impulsándome en mis estudios.

A mi hermano Ing. Bernardo por ser mi guía durante toda esta carrera universitaria, todas tus motivaciones me impulsaron a ser mejor.

A mi tía Rosario quien la considero como otra madre, gracias por todo el apoyo brindado, por creer en mí y por ese inmenso amor que me demuestra.

A mis sobrinos María, Doménica, Heinz, Maybelline por ser mi inspiración, por tenerme siempre de ejemplo a seguir, no tienen idea de cuan gratificante es aquello.

A Camila quien con su cariño y apoyo incondicional me motivo a esforzarme en mis estudios, y no decaer en ningún momento, gracias por siempre mantenerme enfocado en mis metas.

A mi amiga María quien me acompañó en múltiples desafíos académicos, sin duda alguna tu carisma, consejos y confianza me motivaron a buscar siempre la solución a todos los problemas, gracias por esa gran amistad, y cito la frase que nos identifica; “Tu tranquila, que yo resuelvo”. Te quiero.

A mi amigo Steven quien se convirtió en un hermano, gracias por siempre estar al pendiente de mí, sin duda alguna en el ámbito académico fuimos y siempre seremos la mejor dupla.

**Mario Enrique Salcedo Teran.**

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado: “Implementación de un dispositivo electrónico de monitoreo a través de visión artificial que permita realizar mediciones de frecuencia respiratoria no invasiva”, elaborado por la estudiante Salcedo Teran Mario Enrique, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicia los trámites legales correspondientes.

La libertad, 07 diciembre del 2023



Firmado electrónicamente por:  
LUIS ENRIQUE  
CHUQUIMARCA JIMENEZ

---

Ing. Luis Enrique Chuquimarca Jiménez. Mgt.

# TRIBUNAL DE SUSTENTACIÓN



---

Ing. Ronald Rovira Jurado, PhD.  
**DIRECTOR DE LA CARRERA DE  
ELECTRÓNICA Y AUTOMATIZACIÓN**

UÜÖÆJÁ  
Y ŠÆÖQ ŪÁ  
ÖUT ÖZÁ  
T UÜÖSÖU

ÖÄ( ää( Ää ää( ^) ÖÄ( ;UÜÖÖJÁ  
Y ŠÆÖQ ŪÄÖUT ÖZÄT UÜÖSÖU  
ÖP(Ä) MUÜÖÆJÁ ŠÆÖQ ŪÄÖUT ÖZÄ  
T UÜÖSÖUÄ) MUÜÖÆJÁ ŠÆÖQ ŪÄ  
ÖUT ÖZÄT UÜÖSÖUÄMÖ& ää( ;ÄMÖ  
T( ää( KÜ( Ä)Ää ä ;Ä^Ä^ÖÄ( & ( ^) ä  
Vä ää ää) K  
Ö&@ÄGEG-ÄGÄÄ ÄHK I ÄÄ KÄÄ

---

Ing. Óscar Gómez Morales, MBA  
**DOCENTE ESPECIALISTA**



Firmado electrónicamente por:  
**LUIS ENRIQUE  
CHUQUIMARCA JIMENEZ**

---

Ing. Luis Chuquimarca Jiménez, MSc.  
**DOCENTE TUTOR  
DOCENTE GUÍA UIC**



---

Ing. Corina Gonzabay De La A, Mgt.  
**SECRETARIA**

## RESUMEN

El presente proyecto se basa en el desarrollo de un circuito electrónico de monitoreo capaz de detectar frecuencia respiratoria de una persona, sin el uso del contacto físico entre algún sensor con el cuerpo humano, y sin la necesidad de contar con personal especializado en el área médica. Esto es posible debido al uso de la visión artificial donde es aplicado el método de magnificación euleriana de video, este método permite obtener, cuantificar y analizar cambios o movimientos que se presentan en un video que a los ojos humanos no son notorios, esto en conjunto con el uso de técnicas de análisis de flujo óptico (optical Flow) en secuencia de imágenes, las cuales permiten estimar la frecuencia respiratoria de una persona en estado de reposo de una manera eficiente.

El monitoreo es completamente no invasivo, por lo cual, permite que la respiración sea monitoreada sin cambios o alteraciones por parte de la persona, ya que al tratarse de la frecuencia respiratoria es una variable que presenta cambios a voluntad, es decir, que puede variar si la persona está en conocimiento de que se le va a realizar un análisis.

El circuito puede realizar monitoreos de manera constante y a un muy bajo costo, donde para poder realizar el análisis solo basta con tener una Raspberry pi 4 y un dispositivo que permita realizar la adquisición del video en el caso de este proyecto se cuenta con una cámara de Raspberry pi OV5647. Cabe destacar que también cuenta con una interfaz intuitiva para ser visualizada a través de un monitor por lo que cualquier persona puede accionar el circuito para poder realizar el monitoreo.

**Palabras clave:** Circuito de monitoreo, frecuencia respiratoria, magnificación euleriana de video, flujo óptico.

## ABSTRACT

This project is based on the development of an electronic monitoring circuit capable of detecting respiratory rate of a person, without the use of physical contact between any sensor with the human body, and without the need for specialized personnel in the medical area. This is possible due to the use of artificial vision where the method of Eulerian magnification of video is applied, this method allows to obtain, quantify and analyze changes or movements that occur in a video that are not noticeable to human eyes, this in conjunction with the use of optical flow analysis techniques (optical Flow) in sequence of images, which allow to estimate the respiratory rate of a person at rest in an efficient manner.

The monitoring is completely non-invasive, which allows breathing to be monitored without changes or alterations on the part of the person, since the respiratory frequency is a variable that changes at will, i.e., it can vary if the person is aware that an analysis is going to be performed.

The circuit can perform monitoring constantly and at a very low cost, where to perform the analysis just enough to have a Raspberry pi 4 and a device that allows the acquisition of video in the case of this project has a Raspberry pi camera OV5647. It is worth noting that it also has an intuitive interface to be used for the analysis.

**Key words:** Monitoring circuit, respiratory rate, video Eulerian magnification, optical flow.

## ÍNDICE GENERAL

<b>AGRADECIMIENTO</b> .....	i
<b>DEDICATORIA</b> .....	ii
<b>APROBACIÓN DEL TUTOR</b> .....	iii
<b>TRIBUNAL DE GRADO</b> .....	iv
<b>RESUMEN</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>ÍNDICE GENERAL</b> .....	vii
<b>ÍNDICE DE FIGURAS</b> .....	ix
<b>ÍNDICE DE TABLAS</b> .....	xii
<b>ÍNDICE DE ANEXOS</b> .....	xiii
<b>ÍNDICE DE ECUACIONES</b> .....	xiv
<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I</b> .....	2
<b>1. FUNDAMENTACIÓN.</b> .....	2
<b>1.1. Antecedentes.</b> .....	2
<b>1.2. Descripción del Proyecto.</b> .....	4
<b>1.3. Objetivos del proyecto.</b> .....	6
<b>1.3.1. Objetivo General.</b> .....	6
<b>1.3.2. Objetivos Específicos.</b> .....	6
<b>1.4. Justificación.</b> .....	7
<b>1.5. Alcance del Proyecto.</b> .....	8
<b>1.6. Metodología.</b> .....	9
<b>CAPÍTULO II</b> .....	11
<b>2.1. Marco Contextual.</b> .....	11
<b>2.2. Marco Conceptual.</b> .....	12
<b>2.2.1 Visión artificial.</b> .....	12
<b>2.2.2 Flujo Óptico.</b> .....	12
<b>2.2.3 MEV.</b> .....	12
<b>2.2.4 Signos vitales.</b> .....	14
<b>2.2.5 Temperatura corporal.</b> .....	14
<b>2.2.6 Frecuencia Respiratoria.</b> .....	14
<b>2.2.6 Algoritmo.</b> .....	15
<b>2.2.7 Frame.</b> .....	15



2.2.8 Microprocesador.....	15
2.2.9 Computador monoplaca.....	15
2.3. Marco Teórico.....	16
2.3.1 Signos vitales.....	16
2.3.2 Frecuencia respiratoria.....	17
2.3.3 Alteraciones de la frecuencia respiratoria.....	18
2.3.4 Visión por computador.....	18
2.3.5 Procesos de la visión por computador.....	19
2.3.7 Magnificación Euleriana de video.....	21
2.3.8 Método por aproximación Lineal.....	22
2.3.9 Método basado en fase usando una pirámide directiva compleja.....	23
2.3.10 Método basado en fase rápida usando la pirámide de Riesz.....	24
2.3.11. Optical Flow.....	25
<b>CAPÍTULO III.....</b>	<b>29</b>
3.1. Componentes de la propuesta.....	29
3.1.1 Componentes Físicos.....	29
3.1.2 Componentes Lógicos.....	34
3.2. Diseño de la propuesta.....	39
3.2.1 Circuito monitor de frecuencia respiratoria con Arduino mega 2560 y termistor LM35.....	40
3.2.2 Dispositivo electrónico aplicando el algoritmo de magnificación euleriana para medir frecuencia respiratoria no invasiva.....	48
3.2.3 Interfaz gráfica.....	73
3.3. Estudio de Factibilidad.....	75
3.4. Resultados.....	76
3.4.1 Pruebas empleadas.....	76
<b>BIBLIOGRAFÍA.....</b>	<b>104</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Funcionamiento simple de todo el sistema.....	9
<b>Figura 2.</b> Diagrama de algoritmo de magnificación euleriana.....	22
<b>Figura 3.</b> Método por aproximación lineal. ....	22
<b>Figura 4.</b> Método basado en fase usando una pirámide directiva compleja.....	23
<b>Figura 5.</b> Método basado en fase rápida usando la pirámide de Riesz. ....	24
<b>Figura 6.</b> Líneas de movimiento en flujo óptico. ....	26
<b>Figura 7.</b> Pirámide del método de FarneBack.....	28
<b>Figura 8.</b> Raspberry pi 4 model B.....	30
<b>Figura 9.</b> Cámara Raspberry PI NOIR OV5647 visión nocturna 1080P 5MPX.....	31
<b>Figura 10.</b> Arduino Mega 2560.....	32
<b>Figura 11.</b> Sensor LM35 .....	33
<b>Figura 12.</b> Amplificador operacional TL084CN.....	34
<b>Figura 13.</b> Software Python. ....	34
<b>Figura 14.</b> Librería OpenCV. ....	35
<b>Figura 15.</b> Librería NumPy .....	35
<b>Figura 16.</b> Librería SciPy.....	36
<b>Figura 17.</b> Librería Matplotlib .....	37
<b>Figura 18.</b> Tkinter .....	37
<b>Figura 19.</b> Interfaz Pi Imager.....	38
<b>Figura 20.</b> RealVnc .....	38
<b>Figura 21.</b> Identificativo MobaXterm.....	39
<b>Figura 22.</b> Multisim .....	39
<b>Figura 23.</b> Bloque de funcionamiento del circuito.....	40
<b>Figura 24.</b> Tipos de conexión LM35.....	41
<b>Figura 25.</b> LM35 tipo encapsulado (T0-92).....	42
<b>Figura 26.</b> Amplificador de instrumentación. ....	42
<b>Figura 27.</b> Circuito amplificador desarrollado en multisim.....	44
<b>Figura 28.</b> Circuito amplificador y osciloscopio.....	45
<b>Figura 29.</b> Osciloscopio con la salida de voltaje.....	45
<b>Figura 30.</b> Circuito final hacia el Arduino. ....	46
<b>Figura 31.</b> Simulación del circuito realizado en tinkercad.....	47
<b>Figura 32.</b> Circuito físico de la etapa de amplificación, sensor LM35 y Arduino. ....	47
<b>Figura 33.</b> Procesamiento de las etapas del algoritmo. ....	48
<b>Figura 34.</b> Método de aproximación lineal orientado a la magnificación euleriana de video.....	49
<b>Figura 35.</b> Diagrama de proceso de la función principal. ....	50
<b>Figura 36.</b> Diagrama de proceso de la magnificación euleriana de video. ....	51
<b>Figura 37.</b> Aplicación de magnificación euleriana de video.....	54
<b>Figura 38.</b> Diagrama de flujo del método optical flow.....	55
<b>Figura 39.</b> Evaluación de flujo óptico.....	56
<b>Figura 40.</b> Identificación de dirección de vectores. ....	57
<b>Figura 41.</b> Diagrama de flujo de conteo de respiraciones.....	58
<b>Figura 42.</b> Interfaz Raspberry Pi Imager.....	59
<b>Figura 43.</b> Sistema operativo. ....	59
<b>Figura 44.</b> Configuración inicial de Raspberry.....	59
<b>Figura 45.</b> Selección de país, idioma, zona horaria en la Raspberry. ....	60
<b>Figura 46.</b> Usuario y contraseña de Raspberry. ....	60

<b>Figura 47.</b> Configuración de la red Wifi en la Raspberry.....	61
<b>Figura 48.</b> Pantalla de inicio de la Raspberry. ....	61
<b>Figura 49.</b> Ip de la Raspberry.....	62
<b>Figura 50.</b> Activación de opciones para conexión remota.....	62
<b>Figura 51.</b> Nueva sesión en MobaXterm .....	63
<b>Figura 52.</b> Nueva sesión ya iniciada en MobaXterm.....	63
<b>Figura 53.</b> Configuración de SSH con la dirección Ip.....	64
<b>Figura 54.</b> Conexión correcta de MobaXterm .....	64
<b>Figura 55.</b> Interfaz RealVNC viewer.....	65
<b>Figura 56.</b> Usuario y contraseña para RealVNC.....	65
<b>Figura 57.</b> Permiso de conexión RealVNC.....	66
<b>Figura 58.</b> Instalación de Python .....	66
<b>Figura 59.</b> Python instalado .....	67
<b>Figura 60.</b> Instalación Python IDLE.....	67
<b>Figura 61.</b> Instalación de librería Open CV.....	68
<b>Figura 62.</b> Instalación de librería Numpy .....	68
<b>Figura 63.</b> Ejecución de comando para buscar librerías automáticamente.....	69
<b>Figura 64.</b> Instalación de librería Scipy.....	69
<b>Figura 65.</b> Actualización de todas las librerías.....	70
<b>Figura 66.</b> Diseño 3D del dispositivo electrónico.....	70
<b>Figura 67.</b> Base del diseño.....	71
<b>Figura 68.</b> Tapa de la base.....	71
<b>Figura 69.</b> Base para la cámara.....	72
<b>Figura 70.</b> Tapa de la cámara.....	72
<b>Figura 71.</b> Impresión 3D de la estructura.....	73
<b>Figura 72.</b> Interfaz gráfica.....	73
<b>Figura 73.</b> Bloqueo de botón durante procesamiento.....	74
<b>Figura 74.</b> Ventana para análisis en tiempo real.....	74
<b>Figura 75.</b> Valor de respiración impreso en la interfaz gráfica.....	75
<b>Figura 76.</b> Iluminaciones usadas en las pruebas respectivas.....	77
<b>Figura 77.</b> Posición frontal, lateral y a 45 grados.....	77
<b>Figura 78.</b> Resultados obtenidos de la primera persona en posición lateral.....	79
<b>Figura 79.</b> Resultados obtenidos de la primera persona en posición frontal.....	79
<b>Figura 80.</b> Resultados obtenidos de la primera persona en posición 45 grados.....	80
<b>Figura 81.</b> Resultados obtenidos de la primera persona agitada en posición lateral.....	81
<b>Figura 82.</b> Resultados obtenidos de la primera persona agitada en posición frontal.....	82
<b>Figura 83.</b> Resultados obtenidos de la primera persona agitada en posición 45 grados.....	82
<b>Figura 84.</b> Resultados obtenidos de la segunda persona en posición lateral.....	83
<b>Figura 85.</b> Resultados obtenidos de la segunda persona en posición frontal.....	84
<b>Figura 86.</b> Resultados obtenidos de la segunda persona en posición frontal.....	85
<b>Figura 87.</b> Resultados obtenidos de la tercera persona en posición lateral.....	86
<b>Figura 88.</b> Resultados obtenidos de la tercera persona en posición frontal.....	86
<b>Figura 89.</b> Resultados obtenidos de la tercera persona en posición 45 grados.....	87
<b>Figura 90.</b> Resultados obtenidos de la cuarta persona en posición lateral.....	88
<b>Figura 91.</b> Resultados obtenidos de la cuarta persona en posición frontal.....	89
<b>Figura 92.</b> Resultados obtenidos de la cuarta persona en posición 45 grados.....	89
<b>Figura 93.</b> Resultados obtenidos de la quinta persona en posición lateral.....	90
<b>Figura 94.</b> Resultados obtenidos de la quinta persona en posición frontal.....	91
<b>Figura 95.</b> Resultados obtenidos de la quinta persona en posición 45 grados.....	91

<b>Figura 96.</b> Resultados obtenidos de la sexta persona en posición lateral.....	92
<b>Figura 97.</b> Resultados obtenidos de la sexta persona en posición frontal.....	93
<b>Figura 98.</b> Resultados obtenidos de la sexta persona en posición 45 grados.....	94
<b>Figura 99.</b> Vectores de movimiento en pruebas.....	95
<b>Figura 100.</b> Magnitud de vectores en pruebas. ....	95
<b>Figura 101.</b> Respiraciones por minuto de pruebas.....	96
<b>Figura 102.</b> Error en la sección analizar. ....	98

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Referencia de las respiraciones de una persona con respecto a su edad. ....	17
<b>Tabla 2.</b> Comparativa entre los diversos métodos de magnificación euleriana [39]......	25
<b>Tabla 3.</b> Datos técnicos de Raspberry pi 4B [49].....	29
<b>Tabla 4.</b> Datos técnicos de Cámara Raspberry Pi visión nocturna OV5647 [51]. ....	30
<b>Tabla 5.</b> Características de Arduino mega 2560 [54]......	31
<b>Tabla 6.</b> Datos técnicos del sensor LM35 [59]. ....	33
<b>Tabla 7.</b> Datos técnicos del amplificador operacional TL084CN [61]. ....	33
<b>Tabla 8.</b> Valores para la implementación de la propuesta.....	75
<b>Tabla 9.</b> Datos obtenidos de la primera persona. ....	80
<b>Tabla 10.</b> Datos obtenidos de la primera persona con agitación. ....	83
<b>Tabla 11.</b> Datos obtenidos de la segunda persona.....	85
<b>Tabla 12.</b> Datos obtenidos de la tercera persona.....	87
<b>Tabla 13.</b> Datos obtenidos de la cuarta persona.....	90
<b>Tabla 14.</b> Datos obtenidos de la quinta persona.....	92
<b>Tabla 15.</b> Datos obtenidos de la sexta persona. ....	94
<b>Tabla 16.</b> Precisión del algoritmo en posición frontal. ....	96
<b>Tabla 17.</b> Precisión del algoritmo en una posición lateral. ....	97
<b>Tabla 18.</b> Precisión del algoritmo en una posición de 45 grados. ....	97
<b>Tabla 19.</b> Porcentaje total de precisión con respecto a la posición. ....	98
<b>Tabla 20.</b> Precisión del circuito con Arduino y LM35.....	99
<b>Tabla 21.</b> Comparativa de precisión entre ambos circuitos. ....	99

## ÍNDICE DE ANEXOS

<b>Anexo 1.</b> Código para el dispositivo electrónico de monitoreo a través de visión artificial para medir frecuencia respiratoria no invasiva. ....	112
<b>Anexo 2.</b> Código para el circuito monitor de frecuencia respiratorio con Arduino 2560. ....	125

## ÍNDICE DE ECUACIONES

<b>Ecuación 1.</b> Ecuación para amplificar el movimiento.....	13
<b>Ecuación 2.</b> Aproximación mediante serie de Taylor. ....	13
<b>Ecuación 3.</b> Resultado de aplicar un filtro paso banda. ....	13
<b>Ecuación 4.</b> Amplificación de la señal filtrada. ....	13
<b>Ecuación 5.</b> Combinación de ecuación de amplificación y aplicación de filtro paso banda. .....	13
<b>Ecuación 6.</b> Ecuación de amplificación euleriana.....	14
<b>Ecuación 7.</b> Técnica de suposición del valor gris constante. ....	26
<b>Ecuación 8.</b> Ecuación para hallar valor de resistencia. ....	41
<b>Ecuación 9.</b> Valor de resistencia. ....	41
<b>Ecuación 10.</b> Fórmula para hallar corriente en la resistencia de ganancia. ....	43
<b>Ecuación 11.</b> Expresión de voltaje en R1 y RG ..... 43	43
<b>Ecuación 12.</b> Simplificación de la ecuación de voltaje. ....	43
<b>Ecuación 13.</b> Etapa donde $R_2=R_3$ ..... 43	43
<b>Ecuación 14.</b> Simplificación de ecuación en caso de resistencias iguales. ....	44
<b>Ecuación 15.</b> Ecuación para calcular voltaje de salida.....	44
<b>Ecuación 16.</b> Filtro temporal.....	53
<b>Ecuación 17.</b> Fórmula para determinar el porcentaje de error. ....	96

## INTRODUCCIÓN

La creciente importancia de la tecnología en el área médica ha promovido el desarrollo de dispositivos electrónicos modernos para mejorar la calidad de atención y proveer soluciones eficientes en el monitoreo de la salud. Lo que conlleva a la implementación de un dispositivo electrónico de monitoreo a través de visión artificial, el cual permita realizar mediciones de frecuencia respiratoria de manera no invasiva.

La necesidad de una medición precisa y continua de la frecuencia respiratoria es esencial en la atención médica, principalmente en entornos clínicos y de cuidado domiciliario. Tradicionalmente, la obtención de aquellos datos ha estado vinculado a métodos intrusivos o a dispositivos que pueden resultar incómodos para los pacientes.

Obtener la frecuencia respiratoria analizando movimientos en un video es un método en el que se centra esta propuesta. Una de sus principales ventajas de este método es que cualquier cámara con una resolución adecuada, podría ser utilizada para realizar una monitorización de la frecuencia respiratoria, destacando que la mayoría de cámaras que existen en el mercado se conectan a través de USB, el cual el microprocesador a utilizar cuenta con este puerto.

En concreto, este trabajo se plantea con el fin de proporcionar una solución tecnológica la cual permita realizar el monitoreo de la frecuencia respiratoria de manera precisa, teniendo en cuenta que no se debe afectar el confort del paciente. Es por esto que se empleará un enfoque metodológico que integre principios de visión artificial para el procesamiento de imágenes, extracción de características relevantes y la aplicación de algoritmos específicos para el análisis de patrones respiratorios, a su vez se van a explorar varios fundamentos teóricos y prácticos que deben respaldar con veracidad la implementación del dispositivo, de la misma manera realizar un análisis de los resultados obtenidos mediante pruebas experimentales que abarquen todos los objetivos planteados y finalmente llevar a cabo una comparación con un método convencional.



# CAPÍTULO I

## 1. FUNDAMENTACIÓN.

### 1.1. Antecedentes.

El continuo desarrollo de nuevos algoritmos y aplicaciones hacen de la visión artificial una tecnología en constante evolución y ha experimentado un rápido avance en las últimas décadas, así lo demuestran las numerosas investigaciones y publicaciones existentes en la comunidad científica. Esto podría deberse a la gran cantidad de contenido visual como imágenes y video que se genera en la actualidad, así como la capacidad de procesamiento y almacenamiento que tienen los dispositivos electrónicos y la disponibilidad de herramientas, librerías y lenguajes de programación para dicho fin [1].

En el ámbito de la medicina la visión artificial es ampliamente utilizada para realizar diagnósticos de los pacientes, donde trabaja con imágenes provenientes de procesos como ecografías, resonancias magnéticas, tomografías, radiografías, entre otros, lo que permite la detección de diversas enfermedades con respecto a lo que se está analizando [1]. Es por esto, que la detección de signos vitales a través de un procesamiento de imágenes de un video en el que se logra apreciar el movimiento de un sujeto, esto tiene mucha relevancia, ya que varios investigadores se han interesado en este tema en específico, debido a que una de sus principales ventajas es el uso práctico y la comodidad del mismo y a su vez la versatilidad que le puede ofrecer a el paciente, esto es aplicado por varios métodos los cuales pueden ser los análisis de componentes principales [2]. En cuanto al método aplicable de la magnificación con los principios de funcionamiento que puede ser la magnificación de movimiento basado en fase [3] o magnificación euleriana lineal de movimiento, si se aplica de la forma correcta estos métodos permiten llegar a la correcta medida de signos vitales que es lo que se busca dentro de esta investigación.

La visión artificial consiste en la extracción automatizada de información de las imágenes, por información podemos entender desde modelos 3D, posición de la

cámara, reconocimiento de objetos, y agrupación y búsqueda de contenido, también puede incluir la deformación de las imágenes, la eliminación de ruidos y la realidad aumentada. La visión artificial es una rama de la IA donde permite que computadoras puedan extraer información de diversos videos o entre otros tipos de entradas [4]. Muchas veces la visión artificial trata de imitar a la visión humana, pero otras veces la geometría o un enfoque más estadístico puede ser fundamental para resolver un problema [5].

Para garantizar que el sistema sea estable es necesario acondicionar las imágenes y para esto es completamente necesario obtener constantemente un alto contraste a pesar de los diferentes cambios y condiciones exteriores tal como, ruido de luces externas, las variaciones de materiales y tipos de sistemas [6].

En cuanto a la Magnificación Euleriana de video consiste que al aplicarla en un video hace que se amplifique los sutiles cambios del color de cada pixel, revelándose todos los movimientos que no son evidentes, como nuestro propio pulso [7], además si se realiza un poco más de procesamiento se logra observar el video originalmente grabado, pero haciendo visibles los cambios antes mencionados dentro del mismo, permitiendo obtener información como la magnitud del movimiento o el tiempo en el que ocurre [8].

Para poder hacer un análisis se realiza una reconstrucción tridimensional del objeto y su seguimiento dinámico en dirección lateral, vertical, rotación y de amplificación a través del análisis de imágenes de vídeo de alta velocidad. Las imágenes se analizan mediante los algoritmos de flujo óptico y de visión estereo [9].

Según [10], para la adquisición de señales biomédicas hay que realizar un acondicionamiento, que va a depender de la aplicación que se requiera realizar, las señales al ser obtenidas deben ser codificadas para ser transmitidas y decodificadas para ser visualizadas por un especialista.

Según [11], para la adquisición de señales biomédicas se puede emplear un sistema para el monitoreo de señales analógicas a digitales utilizando protocolo de comunicación RS – 232, además de poder utilizar NetBeans IDE la cual puede servir para la transmisión

de video, el desarrollo de la interfaz y la obtención de datos y video del módulo, esto con el fin de poder detectar estas señales y poder realizar una adquisición de datos precisa.

En esta investigación se plantea la forma que se pueden realizar mediciones a través de un video, en el cual se lo realiza mediante un procesamiento de imágenes o también conocido como visión artificial, el hecho de que exista una posibilidad de no acudir al médico y realizar mediciones en distintos lugares y a cualquier hora con el simple hecho de poseer una cámara y una Raspberry pi 4. Los beneficiarios de esto son las personas que lo adquieran, como ya sabemos los avances tecnológicos y la medicina siempre van de la mano, debido a que estas son complementarias ya sea para tratamientos de alguna enfermedad, entre otros, Lo que se busca con la investigación es ver la forma de aportar ciertas características que aún no se ven hoy en día, apoyándose de magnificación euleriana, recalando que está ya antes mencionada es usada en otros ámbitos, y otra característica es el flujo óptico que es el encargado de ayudar a la detección del mismo.

## **1.2. Descripción del Proyecto.**

El proyecto se basa en el desarrollo de un dispositivo electrónico que sea capaz de realizar mediciones respiratorias, en el cual se ve influido por varios métodos y fundamentos teóricos, para esto es necesario conocer los diferentes programas donde se puede realizar la programación del mismo, donde la interfaz funcione de manera intuitiva y de fácil acceso, para realizar este procedimiento se usa una cámara digital, que esta será la encargada de grabar un video y una vez se obtiene se procede a realizar un estudio mediante el flujo óptico de imagen [12] que se va a observar en el video, apoyándose también de la magnificación euleriana de video donde el algoritmo se encargara de amplificar los movimientos del video [6]. Los movimientos que se logre observar serán los encargados de determinar la medición precisa de la respiración y estos resultados serán observados por visión artificial [13] en un ordenador monoplaca.

La cámara que se usará debe poseer una resolución buena, mínimo de “1280×720” debido a que esto es muy importante para que se pueda apreciar todos los movimientos realizados por el pecho, ya que la estimación debe cumplir con la

mayor precisión posible, debido que al tratarse de la salud de una persona es un factor muy importante.

Para poder desarrollar el código es importante adaptar el algoritmo sobre magnificación Euleriana de video el cual fue desarrollado y publicado por el MIT [17] cabe destacar que este algoritmo será usado como código base de implementación a su vez se le deben realizar varias optimizaciones para lograr una estimación de la frecuencia respiratoria de un sujeto dentro de un video, lo que permitirá una perfecta visualización del movimiento que transcurre durante el tiempo que dure el video.

Una vez teniendo el algoritmo desarrollado para que se puedan realizar las mediciones respiratorias, el computador monoplaca debe tener las características necesarias para poder ejecutar el programa. Cabe destacar que este dispositivo debe permitir movilizarlo de una manera sencilla y al computador monoplaca se le deberá conectar la cámara para que se desarrolle todo el proceso.

Para el proceso de pruebas es importante destacar varias situaciones a tomar en cuenta, una fundamental es la distancia que se encontrara la cámara y la persona, donde se debe tomar una distancia especifica permitiendo que el video capte la región de interés (pecho) y que ocupe en gran parte esta área, esto conlleva a que el video que se requiera analizar cuente con todas las características para un correcto procesamiento [17]. En cuanto a las pruebas cada persona que realice el monitoreo será grabada en una posición de reposo que consta de un minuto, destacando que al estar utilizando una Raspberry pi 4 la duración del video recae en una duración de 7 segundos, este proceso se repetirá con unas cuantas variaciones en la indumentaria del sujeto con accesorios que puedan alterar la estimación de la frecuencia respiratoria. Una vez se obtiene el vídeo en tiempo real el sistema utiliza métodos basados en magnificación Euleriana para obtener la información requerida, donde es importante que amplifique los movimientos del área de interés, entregando un video procesador y posteriormente utilizar el flujo óptico el cual permitirá identificar varios parámetros y proceder a determinar su estimación.

### **1.3. Objetivos del proyecto.**

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

Implementar un dispositivo electrónico de monitoreo a través de visión artificial que permita realizar mediciones de frecuencia respiratoria no invasiva, mediante el método de magnificación euleriana para identificar enfermedades en el sistema cardiorrespiratorio de una persona.

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

- Desarrollar un algoritmo mediante visión artificial, considerando el método de magnificación euleriana que mida la frecuencia respiratoria de una persona de manera no invasiva.
- Implementar un dispositivo electrónico con tecnología de visión artificial considerando la selección de elementos idóneos para las mediciones de frecuencia respiratoria no invasiva.
- Crear una interfaz por computadora amigable para el usuario que realiza las mediciones de frecuencia respiratoria.
- Utilizar un termistor nasal para verificar la exactitud del dispositivo no invasivo desarrollado, comparando los resultados obtenidos de las frecuencias respiratorias entre ambos sistemas.

#### **1.4. Justificación.**

Durante mucho tiempo se ha considerado que la manera más eficiente de poder conocer el diagnóstico de un paciente en lo que respecta su salud siempre se conocía a través de métodos totalmente invasivos, los mismos que se realizaban introduciendo instrumentos en el cuerpo para poder así encaminar tratamientos ante enfermedades, lo que sin duda resultaba molesto y generaba incomodidad o incluso malestar en los pacientes.

Sin embargo, con el pasar de los años, se han implementado y creado nuevos sistemas de detección que han permitido comprobar diagnósticos, siendo menos invasivos y más eficaces, inclusive son más inteligentes y no necesitan de mucho tiempo para determinar ya sea la frecuencia o ritmo respiratorio, la temperatura, la presión, entre otros.

La frecuencia respiratoria es el número de veces que respiras por minuto. Al considerarse como uno de los signos vitales que tiene una persona permite que a través de las cuantificaciones se conozca el estado de salud de un paciente dependiendo de las variaciones que se dan (acelerado o normal).

Tiene un importante alcance investigativo y social en el ámbito de la salud puesto que se ha establecido crear un sistema empleado con la magnificación euleriana que permita estimar la frecuencia respiratoria de una persona mediante un microprocesador, para así obtener valores de respiración en una interfaz.

La realización de este sistema tiene un impacto social, debido a que al ser altamente innovador permite con facilidad la utilidad del mismo y permite conocer rápidamente el estado del paciente. Básicamente los principales beneficiarios son toda una población, categorizada desde la mínima y máxima edad de una persona, es decir niños, jóvenes, adultos y adultos mayores. De esta manera ayudará a detectar síntomas y posteriormente las enfermedades que tiene como sintomatología principal cambios en la frecuencia respiratoria se pueden encontrar lesiones estructurales, trastornos neurodegenerativos y trastornos del desarrollo.

Las investigaciones realizadas han demostrado el desarrollo y funcionamiento del mismo es por lo que se prevé que es un sistema confiable por sus componentes y su funcionamiento inteligente capaz de encaminarse a un avance posteriormente con un margen de error mínimo. Los resultados son muy favorables dados a partir de la recolección de datos captados del ritmo respiratorio de una población.

El presente proyecto a realizar se justifica con el fin de realizar monitores de parámetros fisiológicos de un ser humano, como previamente se ha tratado el signo vital a monitorear es la frecuencia respiratoria, empleando un método no invasivo, es decir, que tiene la finalidad de evitar alterar la medición y a su vez no causar incomodidad sobre la persona.

### **1.5. Alcance del Proyecto.**

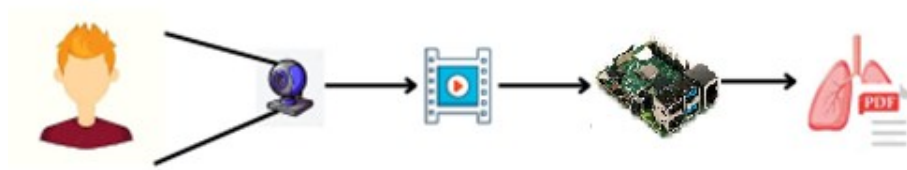
El proyecto que se plantea a ser investigado tiene como finalidad el desarrollo de un sistema en el que tras la obtención de un video en el que se muestra la persona mediante una cámara digital, pueda realizar una amplificación y reconocimiento del pecho, logrando magnificar los movimientos que son causados por la respiración. La información recopilada pertenece a la frecuencia respiratoria de la persona que se le está realizando el estudio, los resultados serán analizados en un computador monoplaca, el cual debe permitir el reconocimiento de una manera fácil, en la cual se aplica la magnificación euleriana de video, donde los cambios o anomalías que se pueden presentar en la respiración del sujeto sean perceptibles, luego del análisis se presentan los resultados mediante una interfaz por computadora.

Para lograr adquirir el video es necesario contar con una cámara RGB que al menos tenga una resolución de 720p (1280x720), los estudios realizados con respecto a la magnificación euleriana de video tienen la certeza y aseguran que mientras más píxeles y resolución tenga la cámara que se está usando, mejor será la señal que es extraída en el análisis.

Como se mencionó anteriormente, la iluminación es importante debido a que debe garantizar la distinción de colores y sombras que se presenten en el transcurso

del video, es decir, que debe verse lo máximo posible, por lo que puede ser natural o artificial.

Se busca limitar para luego poder analizar los resultados que se obtienen por el sistema que se va a desarrollar, la población con la que se va a trabajar está entre el rango de edad de 1 a 32 años, y pacientes que se encuentren en cama (sin importar la edad), además de contar con un ambiente controlado, se sabe que el valor de frecuencia respiratoria normal se encuentre entre 15 a 20 ciclos por minuto. Como dato adicional, la persona a la que se le va a realizar el análisis debe permanecer en reposo, es decir, que no puede realizar ningún movimiento adicional. En la Figura 1 se puede observar el diagrama de funcionamiento simple de todo el sistema, en el que como primera instancia esta la obtención del video donde para adquirirlo se usa la Cámara Raspberry Pi OV5647 y una Raspberry pi 4, y esta a su vez le aplica el algoritmo de magnificación euleriana de video, y finalmente la presentación de datos (resultados).



**Figura 1.** Funcionamiento simple de todo el sistema.

Fuente. Autoría propia.

## **1.6. Metodología.**

### **Etapas para la realización del proyecto.**

Como instancia base, se debe hacer una búsqueda de bibliografías con técnicas y algoritmos que son usados para el procesamiento de video, esto debe incluir la revisión de otros trabajos ya realizados con una temática similar, que tengan temas a tratar en cuanto a detección de frecuencia respiratoria por magnificación euleriana, que permita dar a conocer las limitaciones y parámetros a considerar en el método que se va a emplear [14], de la misma manera buscar el que sea acorde a lo que se quiere realizar [15], con el único fin de desarrollarlo de una manera más eficiente.



Una vez hallado y analizado varias aplicaciones del metodo, se procederá a estudiar el algoritmo de magnificación Euleriana de video que está publicado por el MIT [16] el estudio de este algoritmo tiene la finalidad de usarse como código base, el cual se le debe realizar varias modificaciones y optimizaciones para que pueda medir la frecuencia respiratoria, con esto se busca la clara visualización del movimiento, así como la información necesaria para desarrollar el proceso.

Una vez obtenido el código y esté listo para usarse con las respectivas adaptaciones, se procederá a desarrollar una interfaz de computadora con las características que debe ser de uso fácil e intuitivo y esta deberá ser desarrollada y adaptada para que su ejecución sea dada por computador monoplaca.

Como última etapa se procederá a realizar pruebas de funcionamiento donde los resultados que se obtuvo por el dispositivo permiten ser comparados con los obtenidos por otro dispositivo a usarse, el cual tendrá que proporcionar una medida confiable. La forma de verificar la medida o precisión del mismo es el de usar un sensor de temperatura de muy bajo costo, que tienen las características de medir la frecuencia respiratoria en base a los cambios de temperatura del aire que se está respirando.

Hay que tener en cuenta que para realizar el protocolo de pruebas la cámara a utilizarse deberá estar situada a una distancia prudente del sujeto al que se le va a realizar el análisis y enfocada al área de interés (el pecho) que ocupe casi en su totalidad parte del frame, esto permitirá obtener mejores resultados. Las pruebas a las personas que se las vaya a realizar deben permanecer en reposo durante un minuto, en el cual la indumentaria debe ser cambiada con el único fin de poder verificar si las prendas pueden cambiar la estimación del mismo. Con el video ya listo el sistema debe encargarse de usar los métodos de magnificación Euleriana para así lograr obtener toda la información para cumplir los requerimientos para su respectivo análisis

## CAPÍTULO II

### 2.1. Marco Contextual.

El proyecto permite desarrollar un algoritmo de visión artificial para realizar mediciones de frecuencia respiratoria no invasiva para la provincia de Santa Elena, teniendo en cuenta otros dispositivos que son los tradicionalmente usados para esta acción, y se parte de una instancia base de un algoritmo publicado por el MIT.

Para el desarrollo del algoritmo se buscará proyectos relacionados con magnificación euleriana, ya que es el punto de donde se toma como referencia para toda la investigación, además de conocer diferentes enfoques y aplicaciones que le han dado a dicho algoritmo, este algoritmo permite magnificar los movimientos realizados de una persona.

La intención de la investigación es desarrollar un dispositivo electrónico que permita realizar mediciones de carácter no invasivo, esto quiere decir que no es necesario contar con un sensor o un aparato en específico para poder realizarlo, y destacando que el dispositivo tiene que tener precisión al realizar las mediciones, además de contar con una interfaz interactiva donde cualquier persona pueda ejecutar un monitoreo.

Cabe destacar que las mediciones realizadas serán en un rango de edad de 1 a 32 años y en pacientes que se encuentren hospitalizados, esto con el fin de asegurar que los videos receptados cumplan con todas las características para un correcto funcionamiento del algoritmo, pero teniendo en cuenta que tambien se realizarán mediciones a jóvenes donde se deben cumplir con todas las indicaciones para que el algoritmo funcione de una manera correcta y no presente problemas en cuanto al conteo de respiraciones, además se contar con un espacio controlado, donde las características de este espacio debe contar con buena iluminación y un buen posicionamiento de la cámara.

## **2.2. Marco Conceptual.**

### **2.2.1 Visión artificial.**

La visión artificial es un campo de la IA la cual permite que las computadoras y los sistemas puedan obtener información a través de imágenes digitales, videos y otras entradas visuales, lo que permite que consideren acciones o hagan recomendaciones basadas en esa información. Si la IA lo que permite realizar es que las computadoras piensen, la visión artificial tiene las características que permite ver, observar y comprender [17].

### **2.2.2 Flujo Óptico.**

El flujo óptico “optical flow” se define como el movimiento de los pixeles individuales en el plano de la imagen, y a su vez permite analizar el movimiento que se realiza. Cabe destacar que sirve como una buena aproximación del verdadero movimiento físico proyectado en el plano de la imagen. Los métodos más usados para el cálculo de flujo óptico asumen que el color/intensidad de un pixel es invariante bajo un desplazamiento [18].

### **2.2.3 MEV.**

El método de magnificación euleriana de video es un método que se encarga de revelar dentro de un video variaciones temporales y/o variaciones que son imposibles o casi imperceptibles de distinguir a simple vista, este método es tan centralizado que combina filtros espaciales que es la computarización de pirámides gaussianas estas pirámides se tratan de una descomposición espacial de la imagen de n niveles producto de la combinación de dos técnicas bastante comunes del procesamiento de imágenes, el cambio de resolución; también se aplican filtros temporales a cada banda de frecuencia espacial, para lograr revelar estos sutiles cambios dentro de los videos, donde la entrada simplemente es una secuencia de video común, el video se descompone espacialmente y se le aplican filtros temporales, la señal que resulta luego del método es amplificada utilizando algunos parámetros los cuales permiten visualizar una vez finalizado los pequeños

movimientos. Cabe destacar que el filtrado temporal es homogéneo en todas las bandas espaciales, de la misma forma para todos los niveles de la pirámide de imagen. En cuanto al filtrado espacial una pirámide gaussiana se obtiene con un filtrado binominal y un factor de submuestreo de 2.

Para lograr amplificar el movimiento se busca obtener la expresión que se muestra en la ecuación 1.

$$\hat{I}(x, t) = f(x + (1 + \alpha)\delta(t))$$

**Ecuación 1.** Ecuación para amplificar el movimiento.

Se supone que la imagen se puede aproximar a una serie de Taylor de primer orden, y esto se realiza escribiendo la imagen en el tiempo como  $f(x + \delta(t))$ , esto quiere decir que la serie en  $x$  podría expresarse tal cual como se encuentra en la Ecuación 2.

$$f(c, t) \approx f(x) + \delta(t) \frac{\partial f(x)}{\partial x}$$

**Ecuación 2.** Aproximación mediante serie de Taylor.

Sea  $B(x, t)$  el resultado de aplicar un filtro paso banda en el tiempo a la imagen  $I(x, t)$ , para el primer análisis se considera que la señal  $\delta(t)$  se encuentra por dentro de la señal filtrada, donde queda expresada en la ecuación 3.

$$B(x, t) = \delta(t) \frac{\partial f(x)}{\partial x}$$

**Ecuación 3.** Resultado de aplicar un filtro paso banda.

Para este proceso se amplifica la señal que esta filtrada por un factor  $\alpha$ , posteriormente sumada a la original  $I(x, t)$ , por lo que se obtiene la ecuación 4

$$\hat{I} = I(x, t) + \alpha B(x, t)$$

**Ecuación 4.** Amplificación de la señal filtrada.

La combinación de la ecuación 2 y la ecuación 3, se muestra en la ecuación 5.

$$\hat{I}(x, t) \approx f(x) + (1 + \alpha)\delta(t) \frac{\partial f(x)}{\partial x}$$

**Ecuación 5.** Combinación de ecuación de amplificación y aplicación de filtro paso banda.

Donde se toma en cuenta que en cuanto a la serie de Taylor de primer orden mantiene una amplitud en cuanto a grandes perturbaciones  $(1 + \alpha)\delta(t)$ , por lo que se puede relacionar la amplificación de la señal filtrada con la magnificación de movimiento. Finalmente, la ecuación de la señal procesada se expresa como se muestra en la ecuación 6.

$$\hat{I}(x, t) \approx (x + (1 + \alpha)\delta(t))$$

**Ecuación 6.** Ecuación de amplificación euleriana.

#### **2.2.4 Signos vitales.**

Los signos vitales son un conjunto de variables, señales o parámetros que se presentan en un ser humano, la cual tiene la característica de vislumbrar funciones básicas de los organismos que se encuentran en el ser humano, los signos vitales son la frecuencia cardíaca, respiración, temperatura y presión arterial [19].

#### **2.2.5 Temperatura corporal.**

La temperatura corporal es una medida de la capacidad del cuerpo de producir y eliminar calor, la temperatura corporal es controlada a través del hipotálamo, ocasiona que si se eleva demasiado el organismo se encarga de activar mecanismos de vasodilatación como la sudoración, esto se hace para promover la pérdida de calor, mientras que si la temperatura disminuye se aumenta el metabolismo y esto hace que se produzcan contracciones espasmódicas para retener el calor [20].

#### **2.2.6 Frecuencia Respiratoria.**

La frecuencia respiratoria es el proceso en el que ingresa oxígeno al cuerpo y este es expulsado como anhídrido carbónico, para poder completar el ciclo debe existir una etapa de inspiración (inhalación) cuando el aire ingresa al cuerpo, mientras que la otra etapa es de espiración (exhalación) donde el aire es expulsado del organismo. La frecuencia respiratoria es el número de respiraciones que una persona realiza por un minuto [21].

### **2.2.6 Algoritmo.**

Un algoritmo es un conjunto de acciones que permiten resolver un problema, también pueden ser detallados como instrucciones secuenciales, es decir que va uno detrás del otro, su avance es parte sustancial para lo que conlleva la visión artificial [22].

### **2.2.7 Frame.**

Un frame o también conocido como fotogramas es una imagen concreta dentro de una sucesión de imágenes que se encuentran en movimiento ya sea un video o una animación, cabe destacar que cuantos más frames por segundo tenga, la animación o video tendrá más dinámica, la cual permite que no existan parpadeos en la animación [23].

### **2.2.8 Microprocesador.**

Un microprocesador es un circuito electrónico que se encarga de procesar la energía necesaria para que el dispositivo electrónico en que se encuentra pueda funcionar, teniendo en cuenta la ejecución de los comandos y los programas adecuadamente, también es conocido como la unidad central de procesos, cabe destacar que actualmente es considerado el componente electrónico más influyente en la vida del ser humano [24].

### **2.2.9 Computador monoplaca.**

Un computador monoplaca como su mismo nombre indica es una computadora completa en una sola placa diseñada y ensamblada en un circuito impreso. Tiene como característica principal que su diseño se centra en un solo microprocesador con RAM, E/S de un tamaño muy reducido, lo cual permite que contenga todas las características necesarias en una placa base [25].

## **2.3. Marco Teórico.**

### **2.3.1 Signos vitales.**

Son los que reflejan funciones necesarias del cuerpo humano, donde se incluye la frecuencia cardíaca, respiratoria, temperatura corporal y la presión arterial, estos pueden ser observados, medidos y ser vigilados por ciertos aparatos construidos específicamente para realizar la acción, esto permite determinar el funcionamiento físico de una persona. [26]

#### **Variables que afectan a los signos vitales.**

Son parametrizadas para lograr estandarizar las condiciones basales del ser humano, cabe destacar que la medida siempre será diferente ya que trata de individuos diferentes y existen una variedad de variables que modificaran los signos vitales directamente [27].

##### **2.3.1.1 Edad.**

Los signos vitales se muestran desde que nacen hasta su deceso y los años de vida pueden realizar variaciones en la mayoría de ellos, cabe destacar que la frecuencia cardíaca es mayor en la edad infantil y a medida que envejece va disminuyendo debido a que la frecuencia cardíaca toma mucho más tiempo para acelerar mientras se realiza actividad física y disminuye cuando la persona se encuentra en estado de reposo, así mismo ocurre con la frecuencia respiratoria disminuye para las personas adultas y la temperatura corporal se vuelve un tanto complicada de controlar en personas adultas [28].

##### **2.3.1.2 Genero.**

Luego de la etapa de pubertad la frecuencia cardíaca y respiratoria que presentan las mujeres es mayor a la que presentan los hombres, teniendo en cuenta que ambos poseen el mismo rango de edad, cabe destacar que una vez en la etapa adulta los hombres tienen la presión arterial más alta que las mujeres, en todo caso esto se invierte a una edad de 50 años [29].

##### **2.3.1.3 Actividad física.**

Al mantenerse en actividades físicas una persona puede hacer que el cuerpo entre en un estado de bradicardia, esto se debe a la fuerza de contracción del corazón, esta actividad produce un gran aumento en el pulso cardíaco, la

temperatura corporal y la frecuencia respiratoria, destacando que aumenta el musculo cardiaco [30].

### 2.3.2 Frecuencia respiratoria.

El proceso respiratorio consta en dos etapas, las cuales es la inhalación y la exhalación, la inhalación es la etapa donde la persona ingresa el oxígeno a su cuerpo, mientras que la etapa de exhalación es cuando la persona expulsa anhídrido carbónico. Cabe destacar que la frecuencia respiratoria se determina como el número de veces que una persona respira en el transcurso de un minuto.

Durante la etapa de respiración el organismo toma oxígeno de aire y se encarga de llevarlo a los alveolos, el proceso da inicio con la contracción del diafragma y a su vez los músculos intercostales que son los encargados de realizar la inspiración y espiración [31].

Para poder valorar la respiración de una persona se tiene en cuenta la observación, palpación, la percusión y la auscultación, además se toma muy en cuenta la valoración del paciente [32].

En personas adultas la frecuencia respiratoria normal cuando una persona se encuentra en reposo es de 15 a 20 revoluciones por minuto, si la persona se encuentra fuera de este rango es debido a una anomalía que presenta.

Los valores de frecuencia respiratoria según la edad del individuo se pueden observar en la Tabla 1 [33].

**Tabla 1.** Referencia de las respiraciones de una persona con respecto a su edad.

<b>Edad</b>	<b>Respiraciones por minuto</b>
<b>Recién nacido</b>	30-40
<b>Primer año</b>	26-30
<b>Segundo año</b>	25
<b>Tercer año</b>	25
<b>4-8 años</b>	20-25
<b>8-15 años</b>	18-20



<b>Edad adulta</b>	16-20
<b>Vejez</b>	14-16

### **2.3.3 Alteraciones de la frecuencia respiratoria.**

Estas alteraciones aparecen cuando los ciclos de respiración están fuera de un rango específico o muchas veces el ser humano hace que varíe inconscientemente.

#### **2.3.3.1 Bradipnea.**

Cuando se presenta una frecuencia de respiración menor a 12 respiraciones por minuto, puede ser provocada por pleuritis, una infección, alteraciones neurológicas o alteraciones electrolíticas.

#### **2.3.3.2 Taquipnea.**

Cuando la frecuencia respiratoria que presenta una persona es mayor a las 20 respiraciones por minuto son ocasionada por ansiedad, respuesta al dolor y actividad física.

#### **2.3.3.3 Apnea.**

Es la falta de movimiento respiratorio, esta se puede presentar mientras una persona duerme y se encuentra soñando, y es considerada como una condición grave.

#### **2.3.3.4 Disnea.**

Es la que se presenta en forma de sobreesfuerzo para poder respirar, mayormente sucede durante la inspiración por obstrucción parcial de la vía aérea superior.

#### **2.3.3.5 Tirajes.**

Se presenta cuando existe obstrucciones en la inspiración, por lo que durante los tirajes los músculos reaccionan creando una mayor tracción en la pared torácica.

#### **2.3.3.6 Ortopnea.**

Se determina como la incapacidad de poder respirar cuando la persona se encuentra en una posición horizontal.

### **2.3.4 Visión por computador.**

Es una herramienta o técnica la cual permite realizar una interpretación de objetos tridimensionales del mundo real en base a imágenes que son capturadas con el único fin de procesarlas y de esta forma generar información a través de ella. Este

proceso de captación y procesamiento ayuda a realizar un análisis en específico que va a depender de la aplicación que se le quiera dar [34] [35].

La visión por computador está en constante crecimiento y así como aumenta el número de algoritmos y las aplicaciones que se desarrollan aumenta de la misma forma su demanda, el campo que se ve más beneficiado por la visión por computador es cada vez más extenso, la cual puede desarrollar cualquier actividad sin la necesidad de la vista, por esta razón son usadas en ámbitos como la biología, robótica, industria, seguridad, entre otros [35].

### **2.3.5 Procesos de la visión por computador.**

Al realizar una visión por computador tiende a renacer un patrón de funcionamiento, en el cual se define una serie de procesos que se llevan a cabo desde la adquisición de la imagen hasta donde ya se obtiene el resultado.

#### **2.3.5.1 Captura.**

La primera etapa de la visión por computador o visión artificial es donde se obtiene la imagen digital del objeto, con el fin de realizarle un análisis, para esto es necesario tener un dispositivo que sea capaz de capturar la imagen o video.

#### **2.3.5.2 Preprocesamiento.**

Cuando se llega a la etapa de procesamiento la imagen que ya fue capturada entra a ser sometida a ciertas técnicas de procesamiento de imagen de uso general y previamente establecidas las cuales son orientadas ya sea a resaltar o eliminar ciertas características para que el posterior procesamiento sea sencillo [35].

#### **2.3.5.3 Segmentación.**

En la etapa de segmentación representa un gran desafío, ya que es uno de los problemas que presenta la visión artificial. Consiste en el proceso donde la imagen es dividida en áreas o regiones para lograr separar las zonas que son de interés para el estudio a realizar, la afinidad que tenga en este proceso puede determinar el éxito o fracaso del algoritmo, debido a que clasifica cada uno de las píxeles y les asigna una categoría [36].

#### **2.3.5.4 Descripción.**

En esta etapa el algoritmo obtendrá unas características diferenciadoras del objeto al que se le está realizando el estudio. Para así lograr separarlo del resto de

la imagen, las características que permiten diferenciar se pueden encontrar de manera interna o externa [37].

#### **2.3.5.5 Reconocimiento.**

La etapa de reconocimiento es la encargada de clasificar todos los objetos ya identificados en la imagen tomando en cuenta todas las características previamente obtenidas en la etapa anterior. Por lo tanto, esta etapa es la encargada de la creación de los grupos de los objetos con características parecidas [38].

#### **2.3.5.6 Interpretación.**

En la etapa de interpretación el algoritmo toma todas las características y grupos ya definidos por las etapas anteriores, realizando una interpretación de toda la escena que ya ha sido previamente capturada, el proceso que realiza es en base a los parámetros que se definan para poder resolver el problema, por lo que intentara imitar a la visión humana con el fin de reaccionar a estímulos que presenta la imagen [39].

#### **2.3.6. Iluminación.**

Para procesos donde se utilice la visión artificial o también conocida como visión por computadora la iluminación juega un papel crucial, las iluminaciones de tipo natural son bastante escasas o en algunos casos exageradas por lo que al tener una iluminación de este modo tiende a generar una cantidad inmensa de sombra o reflejos, por lo que reduce en un cierto porcentaje la capacidad de procesamiento de cualquier algoritmo o en casos obliga al programador a desarrollar algoritmos más robustos y complejos. Por esta razón es recomendable que se invierta en un sistema de buena iluminación para evitar estos problemas [40].

Hay diversos tipos de iluminación que corresponden a que se requiere analizar, es decir, que cada una tiene una aplicación especial a lo que se requiera, se toma en cuenta en qué dirección va a golpear a un objeto para poder iluminar cierta característica esencial que se requiera, entre los diversos tipos de iluminación se encuentran:

- Iluminación a contraluz.
- Iluminación puntual.
- Iluminación difusa.
- Iluminación decorativa.
- Iluminación direccional.

- Iluminación coaxial.
- Iluminación directa.
- Iluminación indirecta.
- Iluminación polarizada.

También es importante considerar cuál es la fuente de la iluminación, esto se refiere a que se debe determinar el tipo de luz para lograr iluminar el objeto, entre las fuentes más destacadas, se encuentran:

- Diodos led.
- Laser.
- Fluorescente.
- Incandescente.

### **2.3.7 Magnificación Euleriana de video.**

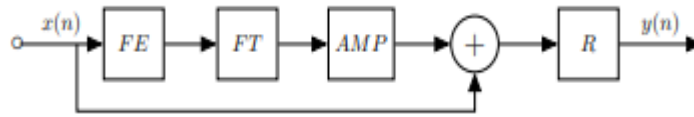
La magnificación euleriana de video para su creación fue basándose en el enfoque Euleriano, en el cual analiza propiedades de fluidos en movimiento como la presión y la velocidad a través del tiempo y espacio [41].

Este es un método en el cual revela cosas que son imperceptibles al ojo humano, lo que conlleva a que este método use filtros espaciales que es la computarización de pirámides gaussianas, estas pirámides se tratan de una descomposición espacial de la imagen de n niveles producto de la combinación de dos técnicas bastante comunes del procesamiento de imágenes, el cambio de resolución y filtros temporales para poder ver estas variaciones que se presentan, la señal logra ser amplificada aplicando parámetros que permitan visualizar estos pequeños movimientos [41].

El método para desarrollarse toma una secuencia del video de entrada y aplica una descomposición fácil, posterior aquello aplica un filtro temporal a todos los fotogramas, lo que resulta que la señal dada ya sean permisibles todos los movimientos pequeños que se dieron en el transcurso del video [42].

El método asume que la imagen entera está cambiando, exceptuando la frecuencia, amplitud y fase de los componentes de la escena son diferentes y que la señal de interés este presente [43].

En la Figura 2 se muestra un diagrama del fundamento matemático que sienta las bases para el algoritmo de magnificación euleriana de video.

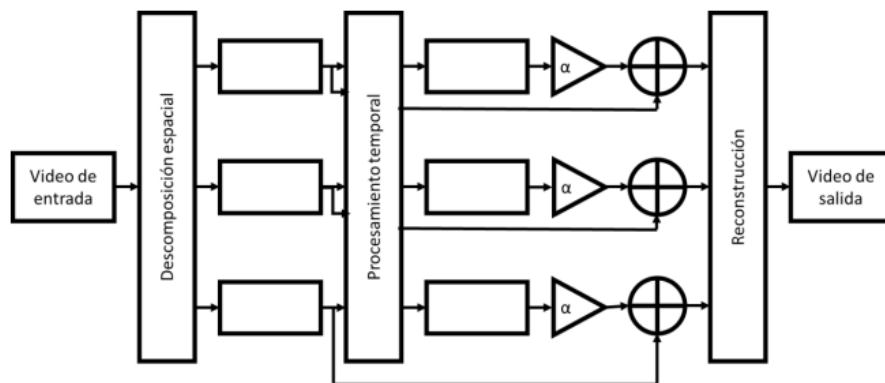


**Figura 2.** Diagrama de algoritmo de magnificación euleriana.

Fuente: [42]

### 2.3.8 Método por aproximación Lineal.

Este método mezcla el procesamiento espacial y temporal para distinguir variaciones temporales o de movimiento en un video, cabe destacar que este método es simple por lo que su funcionamiento se delimita para factores bastante pequeños de amplificación, tener en cuenta que con este método el ruido también es amplificado [41], el proceso con el que lo realiza se aprecia en la siguiente Figura 3.



**Figura 3.** Método por aproximación lineal.

Fuente: Adaptado de “Eulerian Video Motion Magnification” (P.2386) [41]

Este método realiza su proceso siguiendo los siguientes pasos:

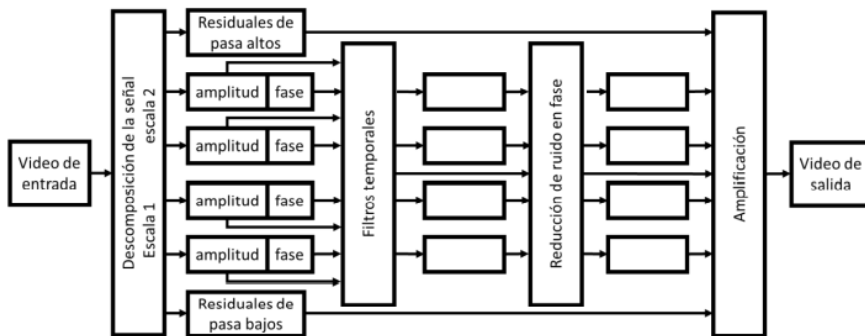
- Como primera instancia se descompone la secuencia en diversas bandas de frecuencia aplicando una pirámide gaussiana que por lo general es de tercer

o cuarto nivel, al fotograma se le aplica un filtro pasa baja esto conlleva a que se reduzcan las muestras para la construcción de la pirámide laplaciana.

- Se realiza el proceso temporal en el cual aplica un filtro pasa banda que permite extraer la frecuencia que se requiere [44].
- Al extraer todas las señales serán amplificadas por un factor  $\alpha$ .
- Finalmente, para poder realizar la reconstrucción se agrega la señal que fue magnificada a la señal que no presenta cambios (original) y se suprime la pirámide espacial.

### 2.3.9 Método basado en fase usando una pirámide direccional compleja.

El método está basado en técnicas de flujo óptico que son basadas en fase, lo principal que realiza es direccionar la amplificación, de tal forma que todos los movimientos que no sean perceptibles puedan ser magnificados sin importar el ruido de la imagen que los rodea, este proceso se logra debido a que obtiene ancho de bandas específicos con la pirámide direccional compleja, todo el proceso que realiza este método puede apreciarse en la Figura 4.



**Figura 4.** Método basado en fase usando una pirámide direccional compleja

Fuente: Adaptado de “Eulerian Video Motion Magnification” [41]

Los pasos que realiza este método se describen a continuación:

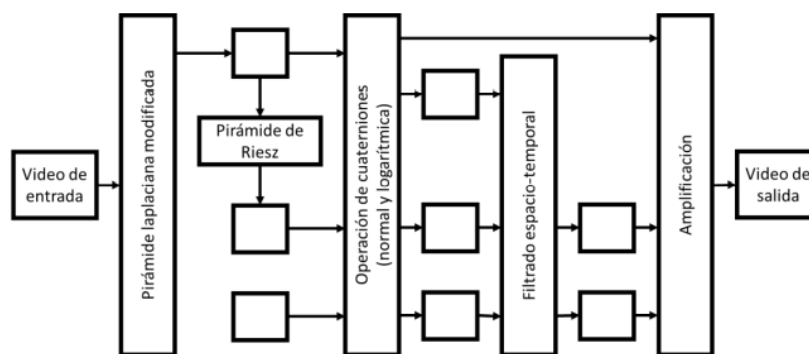
- Como primera instancia realiza una descomposición de la secuencia del video que se ingresa, en las diversas bandas de frecuencia con la separación de la magnitud y la fase.
- Posteriormente, usa un filtro temporal a cada una de las ubicaciones, además de orientación y le realiza un escalamiento a la pirámide direccional.

- Realiza un proceso de eliminación de ruido en fase, para esto aplica filtros pasa bandas temporales en cada fase, es decir que actúa en la instancia donde se filtran los movimientos [44].
- Ahora hace uno de los procesos más importantes, donde realiza la amplificación de las bandas de frecuencia que se requieran.
- Finalmente, realiza la reconstrucción de todo el video.

### 2.3.10 Método basado en fase rápida usando la pirámide de Riesz.

Este método tiene la capacidad de realizar magnificaciones idénticas en un cuarto de tiempo que el método previamente visto que es el de la pirámide direccional compleja, esto es posible ya que todo el proceso lo realiza en el dominio espacial, esto se realiza con el fin de evitar dar arrastres de ruido que se presentan en el dominio de la frecuencia.

La función del método se basa en que permite descomponer la imagen de interés en varias bandas (sub-bandas) sin orientación alguna, para la construcción de una pirámide invertida con amplia respuesta impulsiva se usan filtros pasa baja y a su vez filtros pasa altos, el funcionamiento del método puede apreciarse en la Figura 5.



**Figura 5.** Método basado en fase rápida usando la pirámide de Riesz.

Fuente: Adaptado de “Eulerian Video Motion Magnification” [41]

Los pasos que realiza todo el proceso se detallan a continuación:

- Como primera instancia realiza una descomposición total del video en las diversas bandas de frecuencia, usando una pirámide muy parecida a la laplaciana.

- Luego se usa la transformada de Riesz la cual permite generar la pirámide de Riesz.
- Usa operaciones cuaternarias para hallar la amplitud y la fase.
- Posteriormente, la fase cuaternaria realiza su filtración en tiempo y espacio con lo que permite aislar todos los movimientos de interés, dichos valores son los usados para lograr cambiar la fase de la entrada de la pirámide de Riesz y ocasionar que se produzca una sub-banda que esta amplificada del movimiento.
- Finalmente, las bandas son utilizadas para lograr generar la salida del video magnificado.

En la Tabla 2 se puede observar una comparativa entre los diferentes métodos de magnificación euleriana de video.

**Tabla 2.** Comparativa entre los diversos métodos de magnificación euleriana [41].

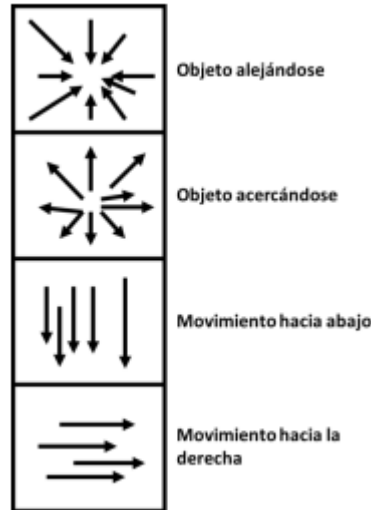
Parámetro	Aproximación lineal	Basa en fase	Basada en fase rápida.
Descomposición	Pirámide laplaciana	Pirámide direccional compleja	Pirámide de Riesz
Límite para $\alpha$	$(1+\alpha)\delta(t) < \frac{\lambda}{8}$	$\alpha \delta(t) \frac{\lambda n}{4}$	Mucha similitud al de la pirámide direccional
Ruido	Amplificado	Minimizado	Minimizado.

### 2.3.11. Optical Flow.

Optical Flow se define como un conjunto de técnicas las cuales sirven para identificar y analizar el flujo de movimientos dentro de unas imágenes que se presentan de manera secuencial, cabe destacar que este método está ligado a la iluminación, por ende, la exactitud de este método depende de ella, ya que funciona a medida que intenta determinar el desplazamiento de los pixeles entre distintas imágenes, este desplazamiento debe volverse reconocible y cuantificable [45].



El movimiento que se presenta en la secuencia de imágenes es identificado mediante flechas y la magnitud es la que está ligada directamente con la magnitud de movimiento, en la Figura 6 se observan las líneas de movimiento las cuales representan los desplazamientos de los píxeles.



**Figura 6.** Líneas de movimiento en flujo óptico.

Para lograr obtener el flujo óptico en un video existe una técnica llamada suposición del valor gris constante, en este caso el método consiste en que al menos en dos imágenes consecutivas de una secuencia los píxeles en una escala de grises se mantengan constantes y en caso de que se considere como un grupo de píxeles en movimiento se obtiene el resultado de la ecuación 7 [46]:

$$I_x u + I_y v + I_t = 0$$

**Ecuación 7.** Técnica de suposición del valor gris constante.

Para esta ecuación tanto la  $I_x$  y  $I_y$  corresponde a gradientes espaciales de una primera imagen. Por otro lado, la  $I_t$  corresponde a la gradiente temporal que esta entre una y otra imagen [46].

### 2.3.11.1 Métodos.

Se pueden hallar varios métodos de flujo óptico y son clasificados con respecto al tipo de flujo que se requiera procesar [47].

#### **2.3.11.1.1 Metodo de flujo denso.**

El método de flujo denso cuenta con una característica especial, en la que su forma de realizar el análisis es a toda la imagen, esto se refiere a que analiza en su totalidad los pixeles, es decir, frame a frame para lograr procesar el flujo de movimiento, este método tiende a ser más efectivo, pero su desventaja es que requiere de algoritmos bastantes robustos [47].

#### **2.3.11.1.2 Método de Horn-Schunck.**

El método de Horn-Schunck usa directamente la ecuación de flujo óptico para poder realizar un análisis de los pixeles dentro del frame, pero tiene una simplificación de su proceso realizando una consideración baja, es decir, que todos los pixeles que están entre los frames consecutivos tendrán movimientos similares. Para poder realizar este procesamiento se debe utilizar un filtro Sobel el cual permite obtener los gradientes espaciales y un filtro diferencial para poder obtener gradientes temporales [48].

#### **2.3.11.1.3. Metodo de Farneback.**

El método de Farneback se basa al igual que el anterior en la ecuación general del flujo óptico y para poder realizar su procesamiento construye una pirámide imaginaria con un numero de niveles, donde el numero puede ser agregado por el usuario, este método aplica que a cada nivel de la pirámide tiende a tener una resolución más baja a comparación del anterior, el algoritmo puede rastrear puntos específicos dentro de diferentes niveles de resolución sin importar el nivel de la pirámide, pero empezando del nivel más bajo [49].

En caso de que se requiera identificar desplazamientos con magnitudes bastante amplias se va a requerir aumentar el número de niveles en la pirámide, aunque si se aumentan los niveles el algoritmo aumentaría su robustez. En la Figura 7 se logra observar el orden jerárquico dentro de la pirámide y como la resolución va a depender del nivel en que se realice el análisis, cabe destacar, que la distancia entre los pixeles varía dependiendo del nivel que se analice.



**Figura 7.** Pirámide del método de FarneBack

Autor: Riascos y Cardona [49].

#### **2.3.11.1.4. Métodos de flujo escaso.**

Para los métodos de flujo escaso trabajan sobre una parte específica de la imagen, en la cual identifican un grupo de píxeles de interés, estos métodos suelen ser menos efectivos, pero su ventaja es que presentan algoritmos bastante ligeros por lo cual su procesamiento será mucho más rápido [47].

#### **2.3.11.1.5. Método de Lucas-Kanade.**

Este método tiene similitud con el de Horn-Schunck, aparece a partir de la ecuación general del flujo óptico, solo que usa dos filtros, un filtro totalmente personalizado el cual obtiene los gradientes espaciales y un filtro diferencial el cual permite obtener los gradientes temporales. Cabe destacar, que cuando se obtienen los gradientes el algoritmo produce una división en secciones a las imágenes más pequeñas, para de esta forma obtener los vectores de flujo óptico, por lo que al realizar este procedimiento parcial proporciona que la respuesta del sistema sea más rápida [49].

## CAPÍTULO III

### 3.1. Componentes de la propuesta

#### 3.1.1 Componentes Físicos.

Los componentes físicos son los encargados de llevar a cabo la implementación física, a continuación, se presentan todos los componentes físicos usados para desarrollar la propuesta tecnológica:

##### 3.1.1.1 Raspberry pi 4.

Una Raspberry pi 4 es un microordenador que tiene un tamaño bastante pequeño, suele compararse su tamaño con el de una tarjeta de crédito, su estructura esta realizada en una placa base que soporta varios componentes que normalmente usa un ordenador, cuenta con un procesador ARM de hasta 1500 MHz, un chip gráfico y una memoria ram que puede ser hasta de 8 GB, permite la conexión a una red mediante puerto ethernet, bluetooth y Wifi, además cuenta con una tarjeta SD que permite instalar sistemas operativos libres [50], en la Tabla 3 se logra apreciar los datos técnicos de la Raspberry pi 4:

**Tabla 3.** Datos técnicos de Raspberry pi 4B [51]

DATOS TECNICOS.	
Modelo.	Raspberry pi 4B
CPU	4 núcleos a 1.5 GHZ con brazo cortex-A72
MEMORIA	4GB LPDDR4 RAM
GPU	VideoCore VI
PUERTOS	2 puertos USB 3, 5v vía cabezal GPIO
Conectividad	.1 lac Wi-fi/Bluetooth 5., Gigabit Ethernet
ALIMENTACION	5V/3 <sup>a</sup> vía USB-C, 5V vía cabezal GPIO



**Figura 8.** Raspberry pi 4 model B.

Fuente. Raspberry [52]

### 3.1.1.2 Cámara Raspberry Pi OV5647.

La cámara Raspberry Pi de visión nocturna OV5647 de 5MP posee una resolución 1080p y enfoque ajustable, tiene incorporado dos sensores con leds infrarrojos los cuales permiten que la cámara tenga buena visión nocturna, a su vez dichos sensores también incorporan una fotorresistencia la cual les permite detectar la intensidad de luz ambiente y de esta forma lograr ajustar el umbral de la luz infrarroja automáticamente [53].

**Tabla 4.** Datos técnicos de Cámara Raspberry Pi visión nocturna OV5647 [53].

Datos técnicos	
Sensor	OV5647 5MP 2592×1944 píxeles
Resolución video	1080p
Apertura (F)	1.8.
Lente	1/4 5M
CCD tamaño	1/4 pulg
Longitud Focal	1.85 mm.
FOV D	70 grados
Dimensión	25mm x 24mm
Alimentación	3.3 V
Fotorresistor.	Detección de intensidad de luz ambiental



**Figura 9.** Cámara Raspberry PI NOIR OV5647 visión nocturna 1080P 5MPX

Fuente: [54]

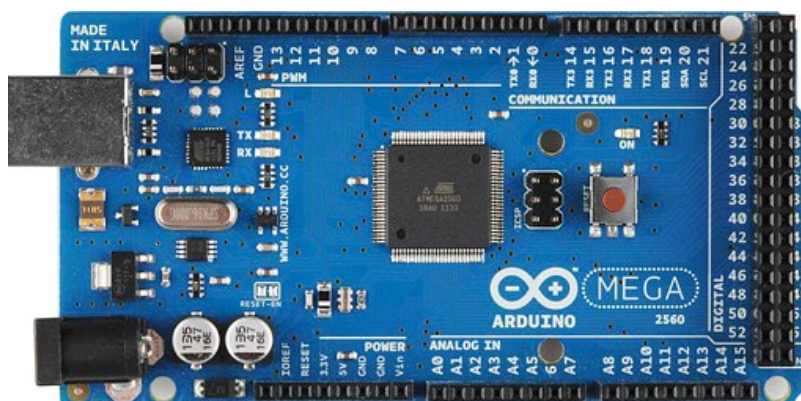
### 3.1.1.3 Arduino mega

Arduino Mega es una placa de desarrollo de código abierto construida utilizando el microcontrolador modelo Atmega2560 con pines de entrada y salida (E/S) analógicas y digitales. La tarjeta se programa en un entorno de desarrollo que implementa un lenguaje de procesamiento/cableado [55].

**Tabla 5.** Características de Arduino mega 2560 [56].

Datos técnicos	
Modelo	Arduino mega 2560
Microcontrolador	ATmega2560
Voltaje de operación	5v
Voltaje de entrada	7-12 V
Voltaje de entrada máximo.	6-20 V
Pines E/S digital	54 (15 salidas PWM)
Pines de entrada analógica	16
Corriente CC para el pin 3.3 V	50mA
Corriente por cada pin E/S	20mA
Memoria flash	256 KB (8kB usado para arranque bootloader)
SRAM	8 KB
EEPROM	4KB

Frecuencia del reloj	16 MHz
Longitud	101.52 mm
Ancho	53.3 mm
Peso	37g



**Figura 10.** Arduino Mega 2560

Fuente: Arduino [57]

#### 3.1.1.4 Sensor de temperatura LM35.

El sensor de temperatura LM35 posee una característica esencial la cual es su alta precisión, cuya tensión de salida se representa con linealidad que es proporcional a la temperatura medida en grados Celsius (centígrados), además consume sólo 60  $\mu$ A de la fuente de alimentación [58].

El LM35 tiene la capacidad de funcionar en un rango de temperatura el cual es entre  $-55^{\circ}$  a  $+150^{\circ}$  C. En el LM35 posee una tensión de salida de 10 mV por cada grado centígrado, es decir, 10 mV/ $^{\circ}$ C en lo cual se obtiene un rango de -0.55 V para el caso de  $-55^{\circ}$  C hasta los 1.5 V para el caso de los  $150^{\circ}$  C [59].



**Figura 11.** Sensor LM35

Fuente. [60]

Los datos técnicos de este sensor se pueden observar en la Tabla 6.

**Tabla 6.** Datos técnicos del sensor LM35 [61].

Datos técnicos	
Resolución	10 mV por cada grado centígrado
Alimentación	Rango de 4 Vdc hasta 20 Vdc
Tipo de medición	Salida analógica
Numero de pines	GND. VCC Y VSalida
Consumo de corriente	60 uA

### 3.1.4.5 Amplificador operacional TL084CN.

El Amplificador Operacional TL084CN DIP-14 es un circuito integrado (CI) de 14 pines que internamente está compuesto por cuatro amplificadores operacionales de entrada JFET de alta velocidad que incorporan JFET de alto voltaje y transistores bipolares en un circuito integrado monolítico [62], los datos técnicos se pueden observar en la Tabla 7.

**Tabla 7.** Datos técnicos del amplificador operacional TL084CN [63].

Datos Técnicos	
Encapsulado	DIP-14
Pines	8
Voltaje de suministro max-min	18v-6v
Corriente de suministro operativo	1.4 mA
Corriente de salida por canal	40 mA
Numero de Amp Ops	4
Disipación de potencia	680 mW
Ganancia de producto	4 Mhz





**Figura 12.** Amplificador operacional TL084CN.

Fuente: [64]

### 3.1.2 Componentes Lógicos

#### 3.1.2.1 Python.

Python es ampliamente usado en aplicaciones web, en desarrollos de software, ciencia de datos y machine Learning. La razón de usar Python se debe a que es eficiente y su aprendizaje es bastante sencillo, destacando su amplia ejecución de plataformas diferentes, posee una capacidad de adaptación a diversos sistemas aumentando su velocidad cuando se refiere a desarrollo [65].



**Figura 13.** Software Python.

Fuente: [66]

#### 3.1.2.2 OPENCV.

Por sus siglas en inglés Open Computer Vision (Vision Artificial Abierta), es una librería de visión artificial de código abierto, la cual se enfoca en el tratamiento de imagen, escrita en C y C++, puede ser ejecutado en los sistemas operativos como Linux, Windows, Mac OS X, tiene interfaces en Matlab, Java y Python. La librería Open Cv es usada para programar un sistema de visión artificial en tiempo real.



**Figura 14.** Librería OpenCV.

Fuente. [67]

### **3.1.2.3 NUMPY NP.**

La palabra NumPy significa "Numerical Python", es una biblioteca de código abierto [68] propiamente a ser usada en python, la cual es utilizada para programación científica, especialmente ciencia de datos, ingeniería, matemáticas y programación científica.

La plataforma consta de objetos multidimensionales en "arrays" y una agrupación de herramientas integradas para implementación en Python [68], es decir que NumPy es un híbrido de C y Python la cual es utilizada como una opción a la programación tradicional que se realiza en MATLAB [68]. Muchas técnicas de ciencia de datos requieren grandes marcos y matrices y cálculos complejos con el fin de separar información valiosa de los datos. NumPy tiene la ventaja de lograr simplificar este proceso mediante un conjunto de diferentes funciones matemáticas.



**Figura 15.** Librería NumPy

Fuente. [69]

#### 3.1.2.4 SciPy

La biblioteca SciPy es gratuita y de código abierto de Python la cual es incluida en un conjunto de herramientas donde también incluye varios recursos de algoritmos los cuales permiten a los ingenieros adentrarse en un proceso de desarrollo de algoritmo en Python.

Existen diversas maneras de usar la biblioteca SciPy entre los que más se destaca es en el uso de diseñar y refinar algoritmos para el aprendizaje autónomo, posee un enfoque modular, con módulos los cuales son relacionados con una optimización de algoritmos, integración, álgebra lineal y procesamiento de señales, esta biblioteca también puede ser usada en diversas herramientas como matplotlib. [70]

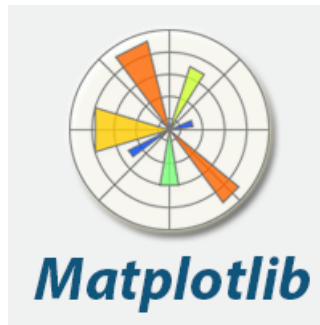


**Figura 16.** Librería SciPy

Fuente. [71]

#### 3.1.2.5. Matplotlib.

Es una herramienta desarrollada para lograr realizar trazados, histogramas, diagramas de barra o cualquier tipo de graficas que se requieran, también es considerada como una alternativa a Matlab, es muy completa, y su gran beneficio es que genera visualizaciones de datos muy detalladas [72].



**Figura 17.** Librería Matplotlib

Fuente: [73]

### **3.1.2.6 Tkinter.**

Es una biblioteca propiamente de Python la cual su uso es para el desarrollo de aplicaciones de escritorio, tiene la ventaja de que facilita el posicionamiento y propiamente el desarrollo de una interfaz gráfica [74].



**Figura 18.** Tkinter

Fuente: [75]

### **3.1.2.7 RASPBERRY PI IMAGER**

Es una herramienta que facilita la instalación de Raspberry Pi OS y otros sistemas operativos en una tarjeta microSD. Prepara la tarjeta para su uso con una Raspberry Pi.



**Figura 19.** Interfaz Pi Imager.

Fuente. [76]

### **3.1.2.8 RealVNC Viewer.**

Este programa consta de un servidor y una aplicación para el cliente para lograr realizar el Protocolo Virtual Network Computing (VNC) y de esta forma poder controlar la pantalla de otro equipo de forma remota, se logra observar su identificador en la Figura 14.



**Figura 20.** RealVnc

Fuente: [77]

### **3.1.2.9 MobaXterm.**

MobaXterm es un software de escritorio de uso remoto para Windows, en el cual incluye un servidor X11 con herramientas de red, como comandos SSH, VNC, MOSH o FTP y Unix, en un único archivo EXE portátil, lo que ayuda a personas

naturales o empresas a conectarse y manipular tareas o aplicativos desde computadoras remotas [78].



**Figura 21.** Identificativo MobaXterm.

Fuente: [79]

### **3.1.2.10 Multisim.**

Multisim es un software estándar en la industria, el cual se usa para el diseño de circuitos y simulación SPICE para electrónica de potencia, analógica y digital en la educación y la investigación [80].



**Figura 22.** Multisim

Fuente: [81]

## **3.2. Diseño de la propuesta**

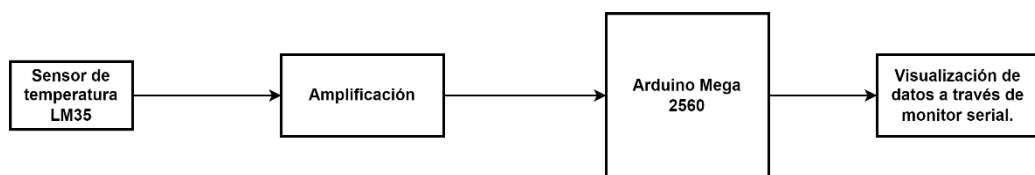
A continuación, se presentará todo lo realizado para poder cumplir con los objetivos planteados:

### 3.2.1 Circuito monitor de frecuencia respiratoria con Arduino mega 2560 y termistor LM35.

La razón fundamental de porque se va a desarrollar un circuito de medición de frecuencia respiratorio es para poder verificar el grado de confiabilidad del algoritmo desarrollado con visión artificial, esto conlleva a realizar una búsqueda de componentes necesarios y artículos científicos relacionados a la medición de frecuencia respiratoria de una persona, en la cual la mayoría de artículos científicos fueron realizados a partir del uso sensores térmicos.

El desarrollo de un circuito electrónico en base a un sensor de temperatura LM35 es ideal en esta aplicación debido a que la temperatura que se obtiene cerca de las fosas nasales tiene una relación directa con la acción de respirar, también se toma en cuenta que los componentes empleados en estas aplicaciones son fáciles de adquirir y tienen un costo muy bajo.

Cabe destacar que este circuito es realizado debido a que se necesita una fuente confiable de medidas de respiración para posteriormente ser comparado con el valor obtenido mediante magnificación, el proceso para poder obtener el valor de respiración puede observarse en la Figura 23 la cual está constituida por un Arduino mega 2560, la etapa de amplificación que es esencial debido a que se requiere observar con más claridad los cambios de voltaje y de esta forma usarlo como un sensor de frecuencia respiratoria.

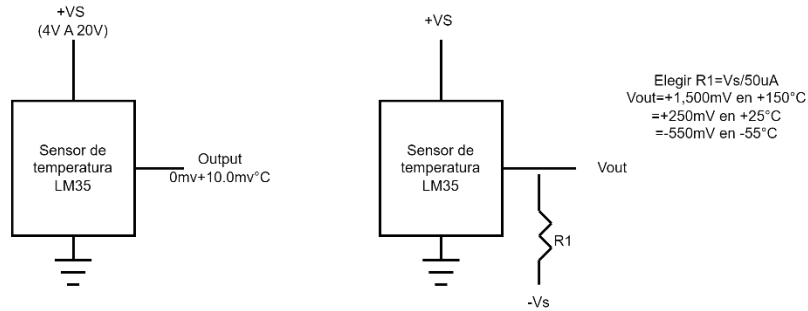


**Figura 23.** Bloque de funcionamiento del circuito.

Fuente. Autoría propia.

Para lograr desarrollar el circuito amplificado es necesario tomar en cuenta que el LM35 obtiene una tensión de salida de 10 mV por cada grado centígrado, es decir, 10 mV/°C lo que da un rango de -0.55 V para los -55 °C hasta los 1.5 V para los 150 °C.

Para poder conectar el sensor existen dos posibilidades, una es la conexión básica y la otra opción que es la más factible es la conexión recomendada, se puede observar en la Figura 24.



**Figura 24.** Tipos de conexión LM35

Fuente: Autoría propia.

Según se muestra en la Figura 23, el voltaje de alimentación  $V_S$  del sensor LM35 puede tener un rango de variación entre 4 V a 20 V. El valor de la resistencia  $R_1$  se elige en función de la ecuación 8.

$$R_1 = \frac{V_S}{50\mu A}$$

**Ecuación 8.** Ecuación para hallar valor de resistencia.

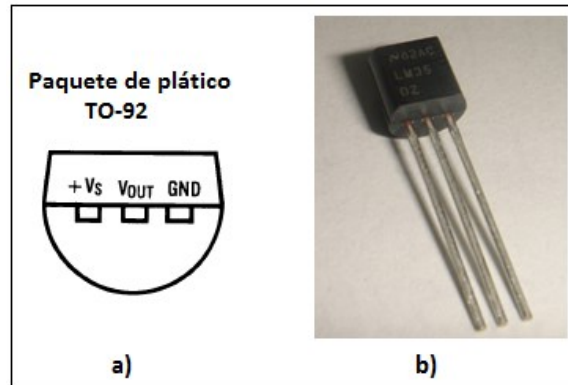
Por ejemplo, para una fuente de alimentación de 5 V se tendría el valor que se muestra en la ecuación 9.

$$R_1 = \frac{5V}{50\mu A} = 100\text{ k}\Omega$$

**Ecuación 9.** Valor de resistencia.

El sensor LM35 está disponible en varios tipos de encapsulados herméticos, uno de los más utilizado es el paquete TO-92 que se muestra en la Figura 25.



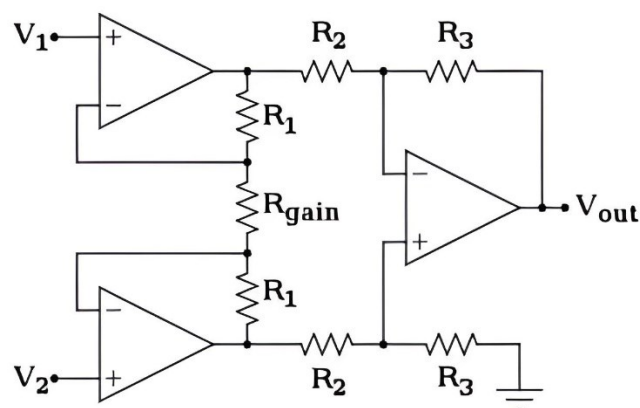


**Figura 25.** LM35 tipo encapsulado (T0-92)

Como el LM35 posee una tensión de salida muy baja (en el orden de los mV) es importante realizar una de amplificación, esto se realiza para lograr observar con mejor claridad los cambios de voltaje y poder utilizarlo como un sensor de exhalaciones.

Para amplificar la tensión de salida del LM35 se propone utilizar amplificadores operacionales en configuración de amplificador de instrumentación, se selecciona debido a que con ella sólo se requiere cambiar el valor de una sola resistencia para regular la ganancia de tensión del circuito, entre sus características es que es capaz de aumentar la tensión de entrada hasta 100 veces, además de que elimina ruidos que posiblemente causen problemas en el medidor de frecuencia respiratoria.

En la Figura 26 se logra observar el esquema de un amplificador de instrumentación diseñado por OPAMP's.



**Figura 26.** Amplificador de instrumentación.

Fuente: Autoría propia

Para el amplificador operacional los voltajes de entrada se igualan, es decir que, en el amplificador de instrumentación, el voltaje que existe en la entrada inversora es el mismo que se tiene en la entrada no inversora.

Con lo anterior se permite calcular la corriente que pasa por la resistencia de ganancia  $R_G$  con la diferencia de estos voltajes sobre el valor de dicha resistencia, en la ecuación 10 se muestra la fórmula para hallar la corriente.

$$I_G = \frac{V_1 - V_2}{R_G}$$

**Ecuación 10.** Fórmula para hallar corriente en la resistencia de ganancia.

La corriente es la misma que circula por la resistencia  $R_1$ , esto se debe al alto nivel de impedancia de entrada del amplificador operacional, el voltaje que cae a través de esta rama intermedia  $R_1$  y  $R_G$ , se expresa en la ecuación 11.

$$V = (R_G + 2R_1) \left( \frac{V_2 - V_1}{R_G} \right) = (V_2 - V_1) \left( \frac{R_G}{R_G} + \frac{2R_1}{R_G} \right)$$

**Ecuación 11.** Expresión de voltaje en  $R_1$  y  $R_G$

También se puede obtener una simplificación la cual es mostrada en la ecuación 12.

$$V = (V_2 - V_1) \left( 1 - \frac{2R_1}{R_G} \right)$$

**Ecuación 12.** Simplificación de la ecuación de voltaje.

La expresión de la ecuación 12, define la diferencia de tensión que existe entre la salida inmediata de la primera etapa de amplificadores operacionales, ya que el resto del circuito es un restador de ganancia.

En la etapa donde  $R_2 = R_3$ , la salida que se obtendrá será la diferencia de tensión de su entrada sin ganancia, por lo tanto, se va a expresar de la misma forma que la caída de tensión que se tiene entre las resistencias  $R_1$  y  $R_G$

$$V = (V_2 - V_1) \left( 1 - \frac{2R_1}{R_G} \right)$$

**Ecuación 13.** Etapa donde  $R_2=R_3$

La simplificación mencionada se aplica cuando  $R_2 = R_3$ , en este caso como se encuentra que todas las resistencias son iguales, a excepción de  $R_G$ , se puede simplificar la expresión, tomando en cuenta que  $R_G$  será la resistencia encargada de variar la ganancia, la expresión que se obtiene se puede observar en la ecuación 14.

$$a = \frac{R_1}{R_X}$$

$$R_X = R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = R_7$$

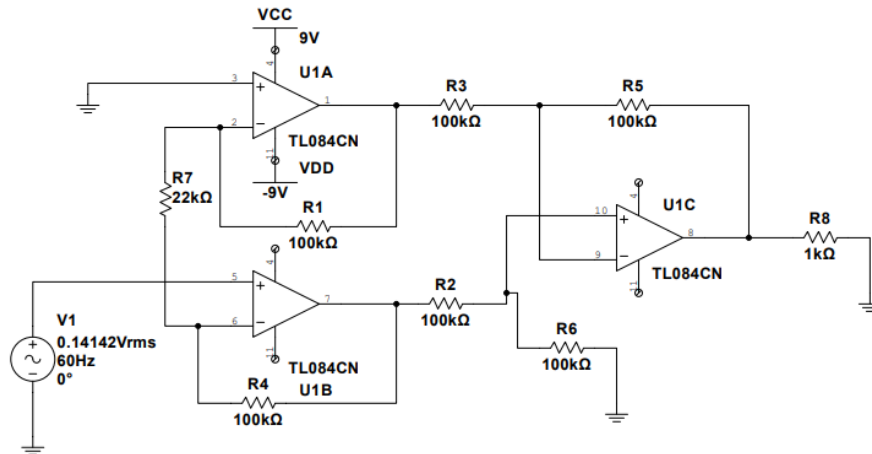
**Ecuación 14.** Simplificación de ecuación en caso de resistencias iguales.

La expresión para lograr calcular el voltaje de salida queda definida como se muestra en la ecuación 15.

$$V_o = (V_2 - V_1) \left( 1 + \frac{2}{a} \right)$$

**Ecuación 15.** Ecuación para calcular voltaje de salida.

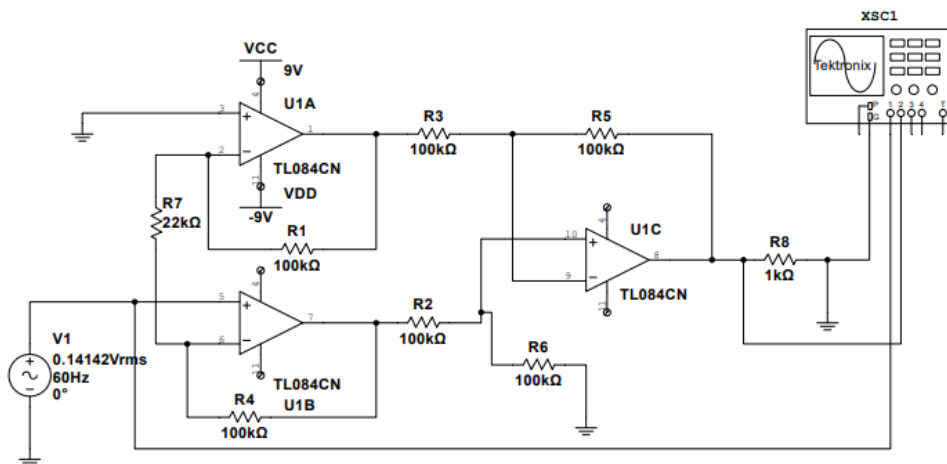
Para poder realizar una simulación de cuanto será la amplificación que realiza dicho amplificador, se usa multisim, y construimos el circuito como se muestra en la Figura 27.



**Figura 27.** Circuito amplificador desarrollado en multisim.

Fuente: Autoría propia.

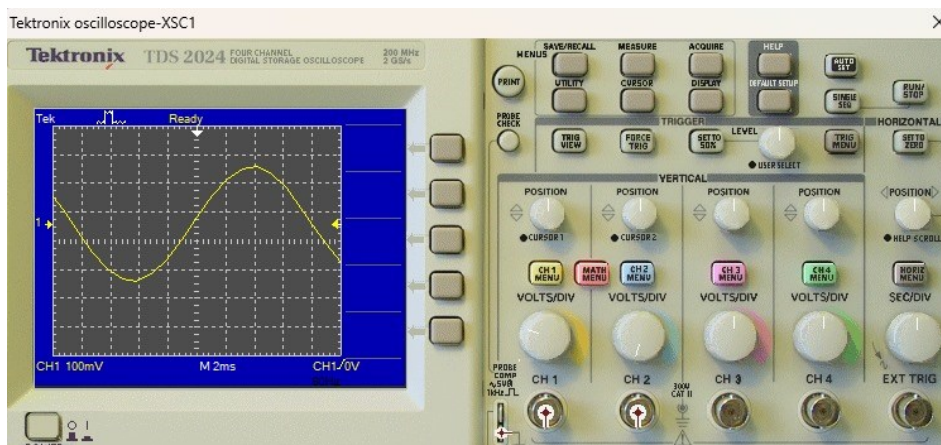
Para poder comprobar es necesario realizar una modificación e incluir un osciloscopio en el circuito como se muestra en la Figura 28.



**Figura 28.** Circuito amplificador y osciloscopio.

Fuente: Autoría propia.

Multisim permite mostrar cómo se comporta el circuito, cabe destacar que el circuito cumple con las características requeridas para que realice el conteo de respiraciones, la gráfica de salida se logra observar en la Figura 29.



**Figura 29.** Osciloscopio con la salida de voltaje

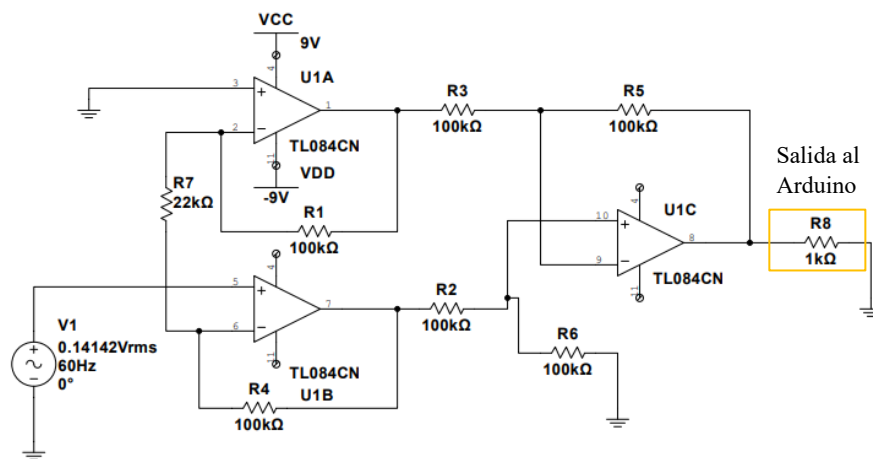
Fuente: Autoría propia.

Una vez realizada la comprobación del funcionamiento de la amplificación mediante la simulación, se debe reemplazar la fuente de voltaje por el sensor LM35.

Cabe destacar que para poder realizar comprobaciones se toma en cuenta la temperatura ambiente, durante la tarde la medida es de 2.2V, mientras que, en las mañanas, dicho valor está aproximadamente en 2V.

Para lograr comprobar la medida se coloca el sensor de temperatura LM35 en una fosa nasal y se procede a respirar para poder verificar algún cambio de tensión en la resistencia R8, independientemente de la temperatura ambiente, se observa que al respirar en el sensor la temperatura varía entre + 0.5 a + 0.9 V, por lo que se determina un umbral de tensión de 2.6 V.

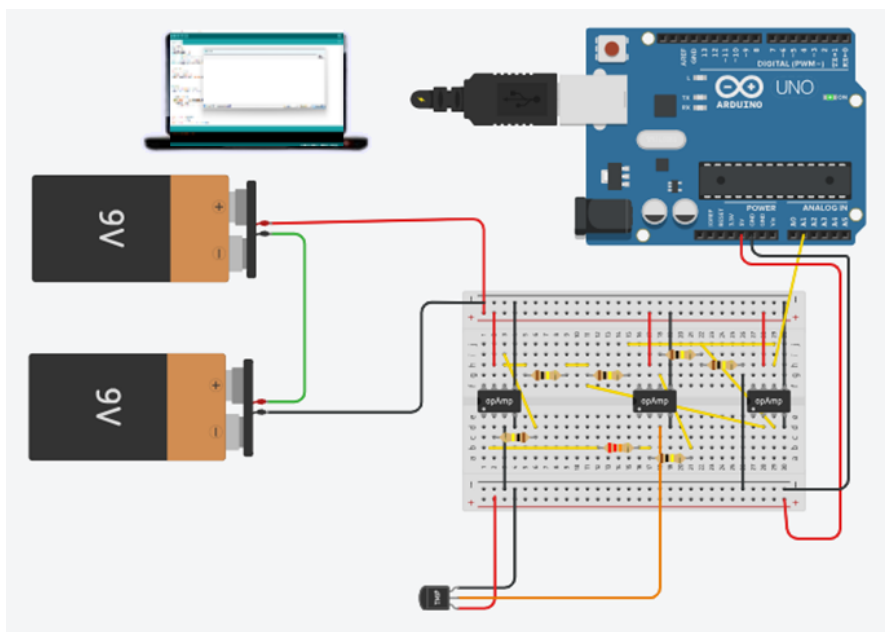
Para lograr conectar el amplificador es necesario quitar la resistencia (R8) esto se debe a que esta resistencia solo es utilizada para observar los cambios en tensión al realizar una exhalación y finalmente se conecta un cable hacia una entrada analógica del Arduino mega, la entrada analógica elegida es la A0, una vez colocado, también hay que conectar una tierra del Arduino utilizado a la tierra común del circuito, como se muestra en la Figura 30.



**Figura 30.** Circuito final hacia el Arduino.

Fuente: Autoría propia.

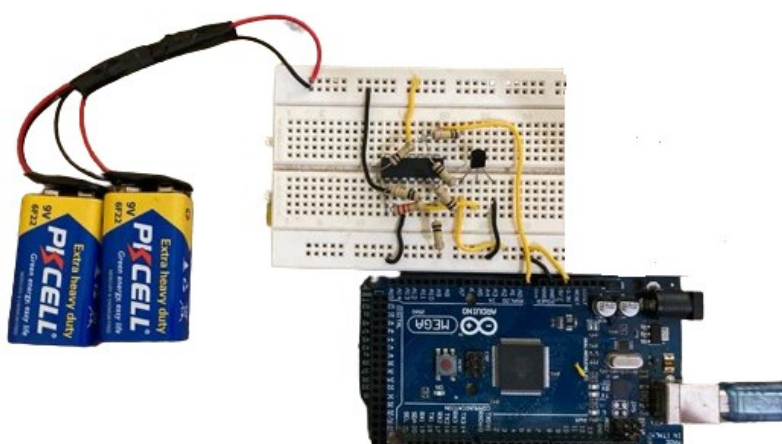
En la Figura 31, se observa el circuito simulado una vez tomado todos los cambios mencionados para que el circuito funcione de una forma correcta, esta simulación fue realizada en tinkercad, por lo que en esta herramienta no se encuentra el amplificador operacional TL084CN, pero se cuenta con un 741 y con aquel se realizaron las adaptaciones para que funcione igual que el otro amplificador, además los resultados de respiración por minuto serán visualizados a través de monitor serial.



**Figura 31.** Simulación del circuito realizado en tinkercad.

Fuente: Autoría propia.

En la Figura 32 se muestra el circuito realizado de manera física donde consta su etapa de amplificación de señal con el amplificador operacional TL084CN, el sensor utilizado que es el LM35 y finalmente el Arduino mega 2560, cabe destacar que los valores obtenidos por este circuito, serán observados a través del monitor serial de Arduino IDE, con el fin de ser usado en la etapa de pruebas a realizar.



**Figura 32.** Circuito físico de la etapa de amplificación, sensor LM35 y Arduino.

Fuente: Autoría propia.

### 3.2.2 Dispositivo electrónico aplicando el algoritmo de magnificación euleriana para medir frecuencia respiratoria no invasiva.

El algoritmo desarrollado es capaz de identificar las respiraciones de una persona, además de cuantificarlas mediante un video capturado a través de una cámara en la cual para poder contar las respiraciones amplifica el video mediante la magnificación euleriana de video.

Este algoritmo está dividido en tres etapas diferentes, las cuales se puede visualizar en la Figura 33, el fin de esta figura es detallar de una manera reducida los tres procesos que el programa realiza para lograr estimar la frecuencia respiratoria.

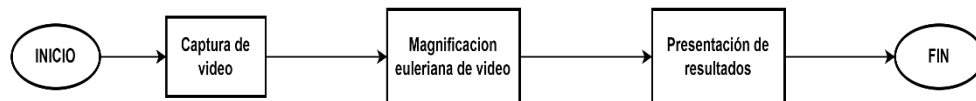


Figura 33. Procesamiento de las etapas del algoritmo.

Fuente. Autoría propia.

Mediante se va realizando cada etapa el video a realizar el procesamiento atraviesa un proceso diferente, el cual tiene el fin de preparar para que logre ingresar al siguiente bloque, esto quiere decir que cada proceso logra obtener una información específica del video el cual es analizado y da como resultado la estimación de frecuencia respiratoria.

#### 3.2.2.1. Obtención de video.

Como ya se especificó previamente para poder realizar el proceso de estimación de frecuencia respiratoria y aplicarle la magnificación euleriana de video es necesario realizar la captación de un video en la que se encuentre una persona y esto se lo realiza mediante un dispositivo que sea capaz de realizar capturas de video.

Hay que tomar en cuenta que el dispositivo seleccionado para capturar el video debe tener una resolución mínima de 1080p, debido a que varios estudios

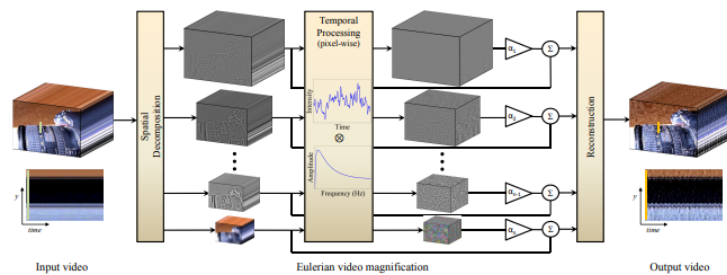
demuestran que entre más píxeles se encuentren en la imagen, la información que el logaritmo logra adquirir para trabajar es mucho mayor y por consiguiente se obtendrá una respuesta mucho más confiable y eficiente.

Para lograr capturar el video se optó por usar una cámara Noir OV5647 propia de Raspberry que cumple con todas estas características necesarias la cual se muestra en la Figura 9.

### 3.2.2.2. Aplicación de magnificación euleriana de video.

Una vez que se logra obtener el video en un formato específico (.avi) y con la resolución previamente definida lo que se procede a realizar es aplicarle el proceso de magnificación euleriana de video, cabe destacar que en esta etapa es donde se debe usar el algoritmo dado por Hao-Yu Wu en el artículo titulado “Eulerian video magnification for revealing subtle changes in the world” en el cual es propuesto el tema de magnificación y los diversos campos que puede ser aplicado [42].

Para poder llevar a cabo este procesamiento es necesario modificar y enfocar el algoritmo a la identificación de respiración de una persona la cual debe mantenerse en reposo, la magnificación euleriana aplicada en este caso se enfoca y obedece al método de aproximación lineal, este método se puede observar en el siguiente esquema de la Figura 34.



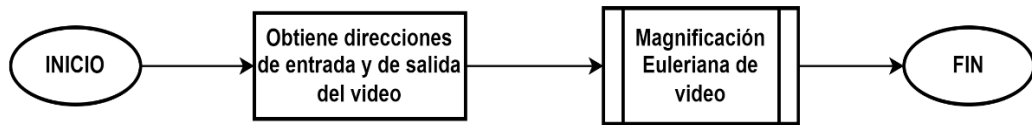
**Figura 34.** Método de aproximación lineal orientado a la magnificación euleriana de video.

Fuente: Adaptado de “Eulerian video magnification for revealing subtle changes in the world” [42].

Esta etapa está compuesta de la función principal que es la encargada de ubicar el video que es capturado y a su vez aplicarle todo el proceso, esto conlleva a la función que es la encargada de realizar todo el proceso de magnificación, lo



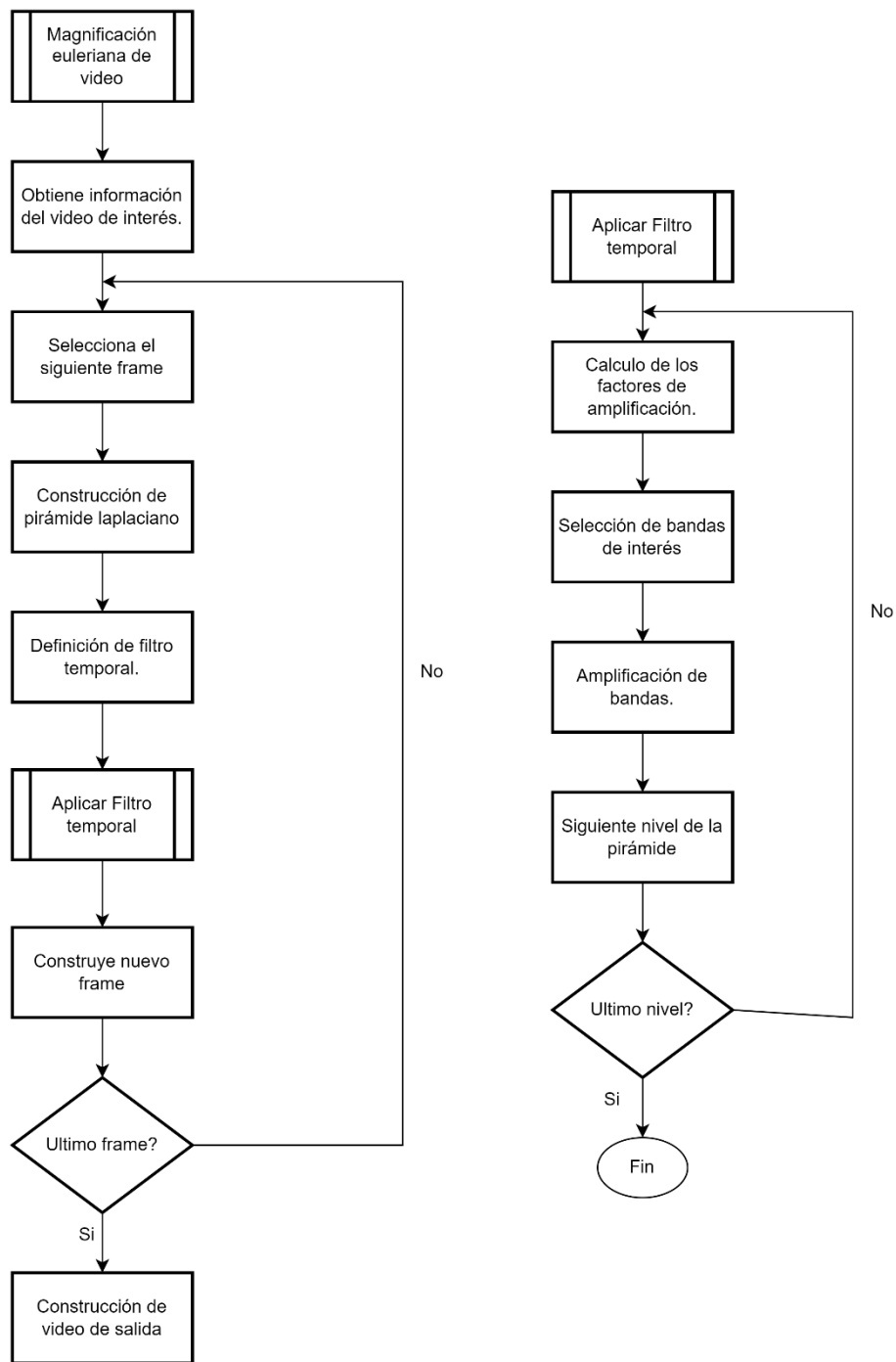
mencionado se logra observar la Figura 35, este diagrama muestra el proceso que realiza el dispositivo electrónico de monitoreo.



**Figura 35.** Diagrama de proceso de la función principal.

Fuente. Autoría propia.

La función de magnificación euleriana de video es llamada mediante el proceso de la función principal la cual debe acompañarse de parámetros que sean capaces de realizar la magnificación para la detección de movimientos, esta función puede observar en la Figura 36, donde se muestra de forma general como es aplicada la magnificación euleriana al video de interés previamente capturado por la cámara.



**Figura 36.** Diagrama de proceso de la magnificación euleriana de video.

Fuente. Autoría propia.

### 3.2.2.2.1. Obtiene información del video de interés.

Una vez capturado el video se debe ingresar a la función de magnificación euleriana de video con todos los parámetros constantes necesarios para lograr

realizar la magnificación, estos parámetros son los valores correspondientes para lograr obtener una mejor magnificación de movimientos muy parecidas a la respiración humana, estos valores se encuentran en el trabajo realizado por Wu Hao-Yu.

Además de los parámetros ya mencionados se debe tomar en cuenta que esta etapa es cuando se logra obtener la información de las características que presenta el video en las cuales se encuentran la resolución en largo y ancho, tasa de fotogramas por segundo, número de fps, etc. Cabe destacar que a partir de todos estos valores se realizaran los cálculos de magnificación e inclusive las variables delimitadas de los ciclos del proceso.

#### **3.2.2.2.2. Pirámide laplaciana.**

Esta pirámide es utilizada para lograr dividir en las diferentes frecuencias espaciales el fotograma, cabe destacar que para que se realice es usada la función *build\_gaussian\_pyramid*.

Los diferentes niveles de la pirámide para generarse deben transformar de formato de color RGB A NTSC, con este nuevo fotograma se genera la pirámide y logra definir el número de niveles. Cabe destacar que la primera pirámide se almacena como un parámetro para el filtro temporal.

#### **3.2.2.2.3. Filtro temporal.**

Una vez que se obtiene la pirámide del primer fotograma del video el algoritmo se encarga de generar la siguiente pirámide laplaciana del fotograma siguiente, luego se encarga de generar dos filtros pasa baja usando las dos pirámides y a su vez los factores de amplificación que fueron ingresados en la función mencionada.

La diferencia de los dos filtros pasa baja genera el filtro temporal, el cual es usado para cada banda de frecuencia espacial.

El filtrado ocurre en el dominio de la frecuencia y el medio para realizar es mediante una máscara la cual se construye a partir de frecuencias de corte inferior y superior, por lo tanto la convolución en el dominio temporal sucede a una multiplicación en el dominio de la frecuencia, el filtro que se aplica viene delimitado por la ecuación donde  $L$  corresponde el número de muestras tomadas,

es decir, para esta aplicación sería el número total de imágenes que se procesan y  $F_s$  corresponde a los fotogramas por cada segundo del flujo del video, cabe destacar que este filtro solo ocurre una vez por bloque de imágenes.

$$f' = \frac{2 * f * L}{F_s}$$

**Ecuación 16.** Filtro temporal.

#### **3.2.2.2.4. Aplicación de la magnificación y el filtro temporal.**

El algoritmo realiza el cálculo del factor de magnificación, cabe destacar que este factor va a depender de ciertos valores experimentales en lo que más se destaca son lo de las dimensiones del video de interés, ancho y alto.

Realiza un cálculo de un factor de cambio específico para el nivel de la pirámide en que se encuentra, dicho factor es el que ayuda y logra identificar las bandas de frecuencia espaciales a las que se requiere magnificar.

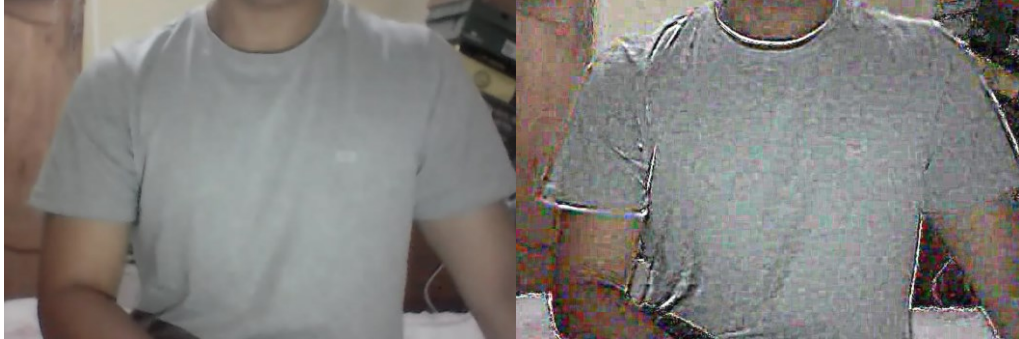
Este algoritmo es capaz de ignorar bandas de frecuencia más altas y más bajas, las cuales las deja sin realizar la magnificación, en este caso utiliza el factor que ya se calculó antes y logra filtrar las bandas de frecuencia espaciales de interés a las cuales les aplica y magnifica la banda.

Por último, divide para dos el factor de cambio de nivel y por consiguiente se mueve el siguiente nivel en la pirámide para lograr realizar el mismo procedimiento.

#### **3.2.2.2.5. Video de salida.**

Para lograr obtener el video de salida es necesario utilizar la función *reconstruct\_video* la cual se encarga de reconstruir el fotograma a partir de la pirámide laplaciana la cual es el resultante de la aplicación del filtro temporal, este dicho resultante es combinado con el fotograma original, de esta forma es generado el fotograma de salida, dicho proceso se vuelve a repetir a todos los fotogramas en el video y así finalmente se logra obtener el video de salida.

A continuación, se presenta la Figura 37 en el lado izquierdo muestra el video capturado y en el lado derecho muestra el video aplicado la magnificación euleriana de video.



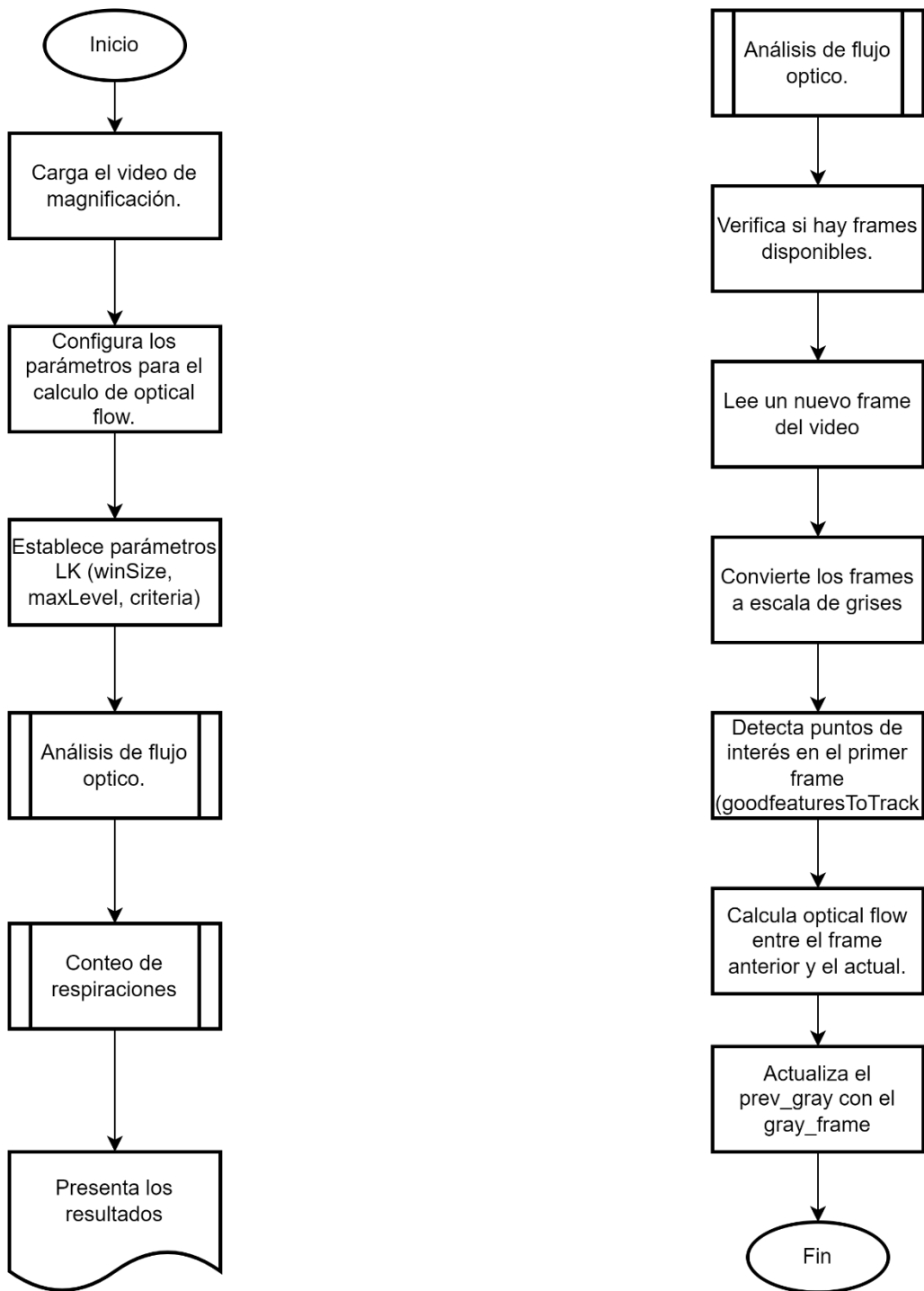
**Figura 37.** Aplicación de magnificación euleriana de video.

Fuente. Autoría propia.

### **3.2.2.3. Optical Flow.**

El método de optical Flow se usa para lograr cuantificar y analizar movimientos, donde se usan métodos de direccionamiento para lograr identificar los parámetros a utilizar, para esto se analiza el movimiento de los píxeles dentro de una escena, es decir que considera los movimientos como un flujo visible de todos los píxeles entre los fotogramas, en esta propuesta tecnología se usan técnicas de flujo óptico las cuales son usadas para lograr cuantificar el movimiento que se presenta en el pectoral al momento de que una persona respira y posteriormente analizar dicho movimiento para lograr identificar los ciclos respiratorios.

En la Figura 38 se logra observar el proceso que realiza este método de una manera simplificada donde se logra apreciar el análisis de flujo óptico para lograr realizar una estimación de frecuencia respiratoria.

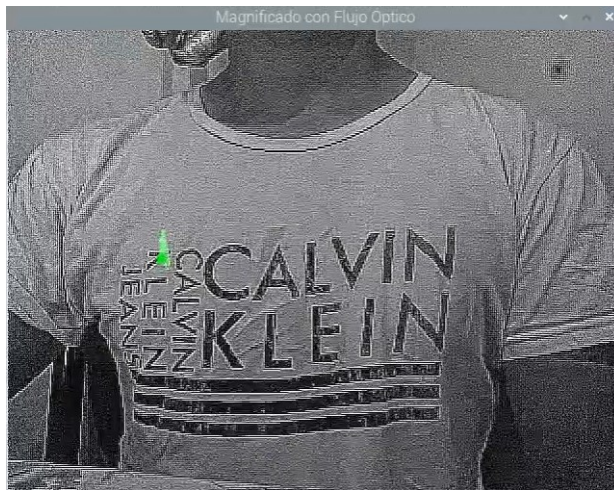


**Figura 38.** Diagrama de flujo del método optical flow.

Fuente. Autoría propia.

### 3.2.2.3.1 Evaluación y estimación del flujo óptico.

Una vez se logra amplificar el video proveniente de la etapa de magnificación euleriana de video se da por inicio la etapa de la estimación por flujo óptico, cabe destacar que en esta etapa es cuando el algoritmo utilizado debe cumplir con las características necesarias para poder realizar el proceso, en la Figura 39 se logra observar esta etapa donde se logra analizar el video magnificado en el punto más adecuado para realizar el análisis.



**Figura 39.** Evaluación de flujo óptico.

Fuente: Autoría propia.

en las cuales las características del código son las siguientes:

- El programa debe inicializarse una vez que el algoritmo detecte el video de magnificación euleriana el cual realiza la amplificación en tiempo real que es el encargado de contener la secuencia de imágenes ya procesadas de la anterior etapa.
- Se deben establecer los parámetros necesarios para poder lograr el cálculo de optical Flow, cabe destacar que estos parámetros deben incluir el tamaño de la ventana, el video máximo y los criterios de terminación.
- Se deben inicializar las variables necesarias las cuales son las encargadas de lograr almacenar los puntos de interés que se requieran.

### 3.2.2.3.2. Análisis de flujo óptico.

Una vez cumplido lo anterior mencionado entra a la función de análisis de flujo óptico que es donde se va a determinar ciertas características y ayudara a identificar los movimientos respiratorios de una persona.

Este método utilizado permite analizar los diferentes componentes de los vectores de movimiento, hay que tener en cuenta que se requiere realizar análisis de diferentes personas y en distintas posiciones por lo que no es posible realizar una parametrización de dirección en que se moverán los vectores de movimiento, es por este motivo que el parámetro a utilizar es la magnitud y los fotogramas.

En la Figura 40 se observa cómo se realiza la identificación de los vectores para verificar cuando se produce una inhalación o una exhalación en el video.

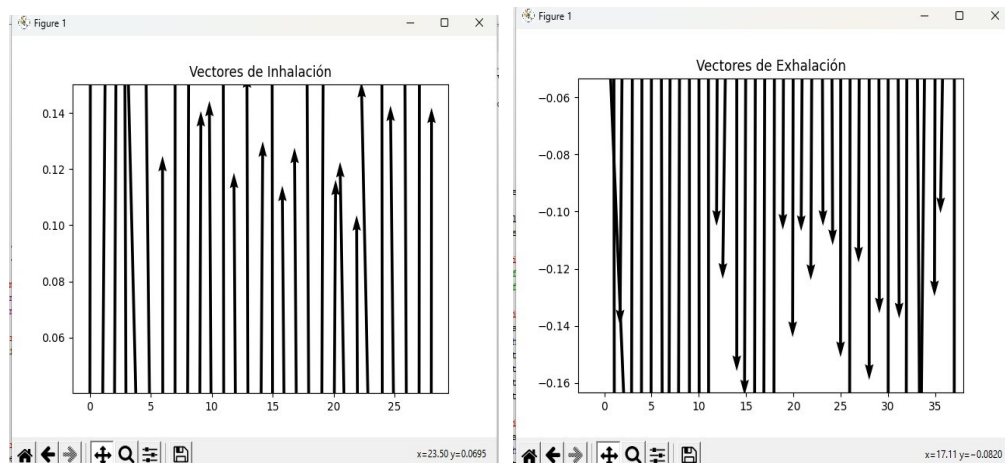
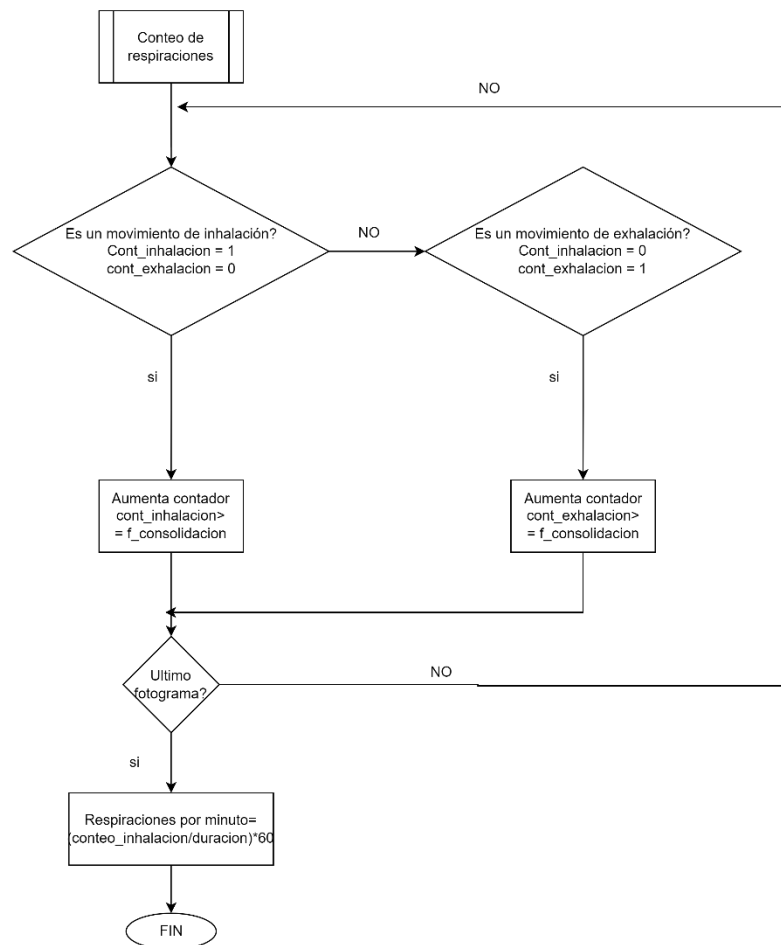


Figura 40. Identificación de dirección de vectores.

### 3.2.2.3.3 Conteo de respiraciones.

Una vez identificado tanto los movimientos respiratorios que conforman el periodo de respiración la etapa de conteo de respiraciones realizara el conteo durante la duración del video y esto es realizado con el procedimiento que se observa en la Figura 41.





**Figura 41.** Diagrama de flujo de conteo de respiraciones.

Fuente: Autoría propia.

### 3.2.2.4 Configuración de Raspberry.

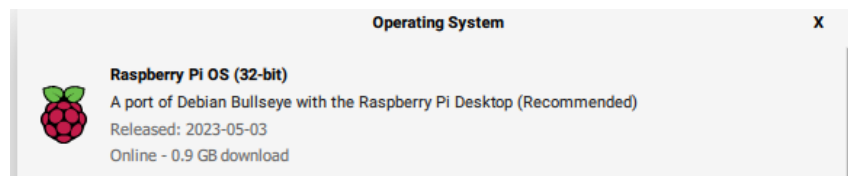
La configuración inicial de la Raspberry pi 4 se la realiza usando el programa “Raspberry Pi Imager” en el cual al descargarlo e instalarlo su interfaz es bastante interactiva, para lograr cargarle el sistema a la Raspberry pi 4 es necesario contar con una tarjeta sd la cual se la debe conectar a la computadora y el programa procede a reconocerla, la interfaz del programa se logra observar en la Figura 42.



**Figura 42.** Interfaz Raspberry Pi Imager.

Fuente: Autoría propia.

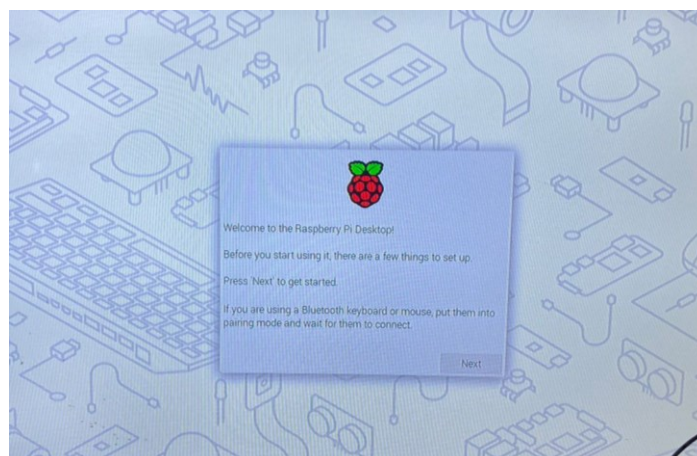
Para lograr cargar el sistema a la tarjeta SD se debe dar clic sobre “CHOOSE OS” y a continuación aparecen varias opciones, para lograr desarrollar la propuesta se seleccionó la primera opción que se logra apreciar en la Figura 43.



**Figura 43.** Sistema operativo.

Fuente: Autoría propia.

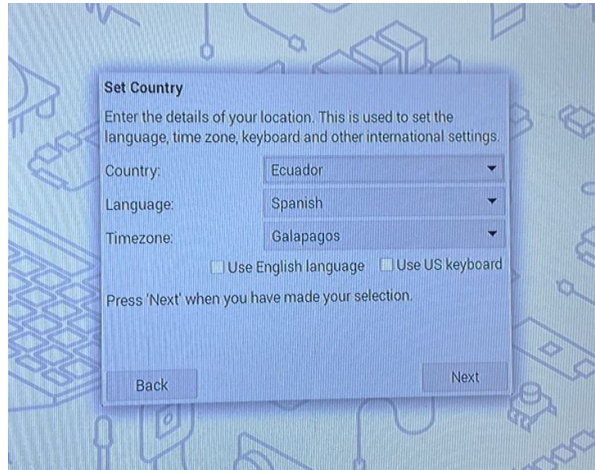
Con los pasos previamente realizados el sistema operativo quedará cargado dentro de la SD, ahora se debe realizar la configuración general de la Raspberry, en la cual al encender se mostrará un apartado que puede visualizarse en la Figura 44.



**Figura 44.** Configuración inicial de Raspberry.

Fuente: Autoría propia.

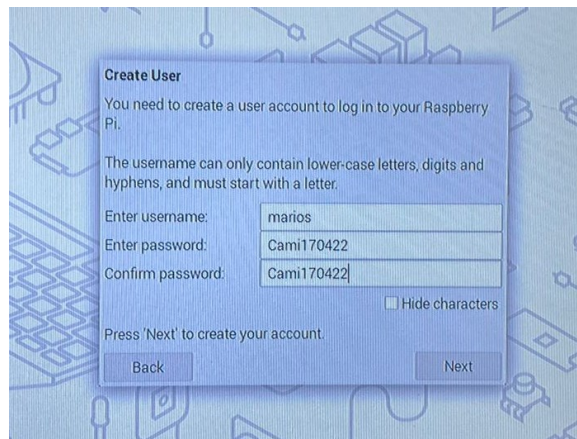
Al avanzar a la siguiente ventana, se debe seleccionar el país, idioma y la zona horaria, la cual queda configurada como se muestra en la Figura 45.



**Figura 45.** Selección de país, idioma, zona horaria en la Raspberry.

Fuente: Autoría propia.

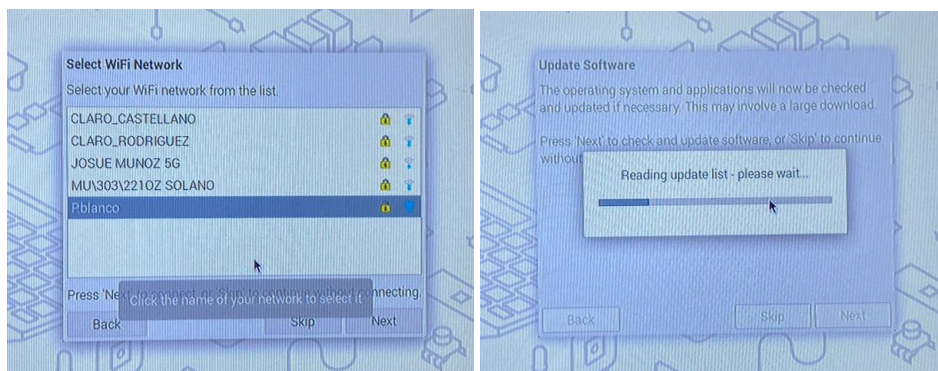
Una vez finalizado el paso anterior, la siguiente ventana se debe crear un usuario y una contraseña propios de Raspberry, esta configuración se puede observar en la Figura 46.



**Figura 46.** Usuario y contraseña de Raspberry.

Fuente: Autoría propia.

En esta propuesta es necesario que la Raspberry funcione de manera remota al computador, es por esto que se requiere conectarla a una red Wifi y esto se lo realiza como se muestra en la Figura 47.



**Figura 47.** Configuración de la red Wifi en la Raspberry.

Fuente: Autoría propia.

Una vez finalizado la configuración básica, ya podremos visualizar la pantalla de inicio tal como se muestra en la Figura 48.



**Figura 48.** Pantalla de inicio de la Raspberry.

Fuente: Autoría propia.

Para poder realizar la comunicación remota es necesario usar dos programas RealVNC Viewer y MobaXterm, estos programas van a permitir conectar la Raspberry a la computadora, para aquello hay que seguir algunas indicaciones las cuales se van a explicar a continuación:

Para lograr observar la Ip de la Raspberry es necesario ingresar al símbolo del sistema y colocar las instrucciones que se observan en la Figura 49.

```
Archivo Editar Pestañas Ayuda
marios@raspberrypi:~$ ifconfig

Archivo Editar Pestañas Ayuda
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

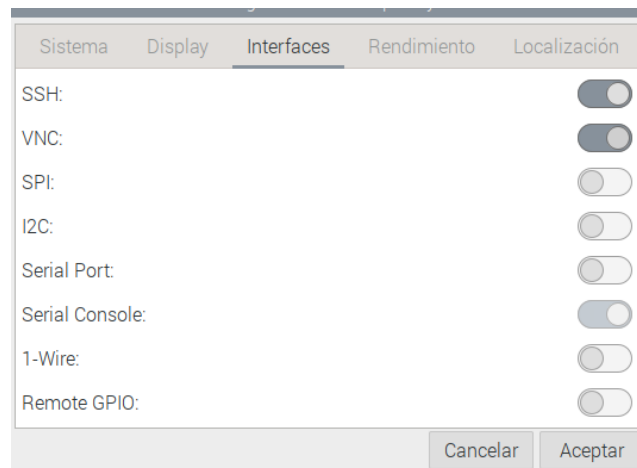
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 28 bytes 2821 (2.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 2821 (2.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.100.87 netmask 255.255.255.0 broadcast 192.168.100.255
inet6 2800:bf0:a831:e3b:5d60:13ab:c784:a26b prefixlen 64 scopeid 0x0<g
lobal>
inet6 fe80::6b0:2797:8e31:5e98 prefixlen 64 scopeid 0x20<link>
ether dc:a6:32:57:b8:47 txqueuelen 1000 (Ethernet)
RX packets 35836 bytes 43173236 (41.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 10479 bytes 1297282 (1.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Figura 49.** Ip de la Raspberry.

Fuente: Autoría propia.

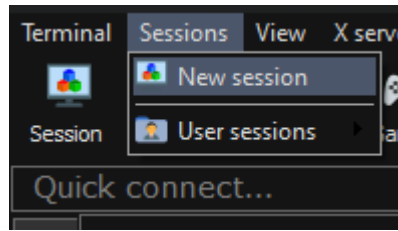
Una vez obtenida la dirección IP de la Raspberry conectada al router, es necesario activar la opción de SSH y la otra opción llamada VNC en la Raspberry, las cuales se pueden observar en la Figura 50.



**Figura 50.** Activación de opciones para conexión remota.

Fuente: Autoría propia.

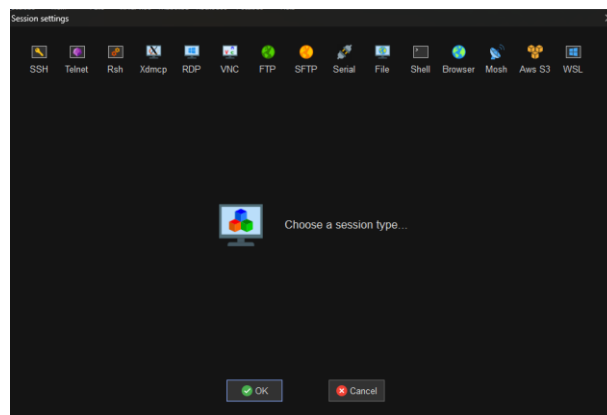
Una vez finalizado lo anterior es momento de usar MobaXterm que es el que va a realizar el enlace de conexión entre los dos dispositivos, para aquello se debe abrir el programa y en la opción de sessions seleccionar “new sesión”, tal como se observa en la Figura 51.



**Figura 51.** Nueva sesión en MobaXterm

Fuente: Autoría propia.

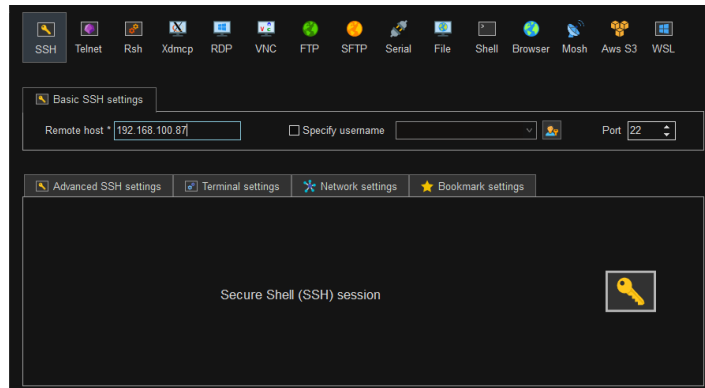
Una vez seleccionada la nueva sesión se obtendrá una nueva ventana en la cual se debe configurar con la dirección previamente obtenida de la Raspberry, la ventana donde se debe configurar se muestra en la Figura 52.



**Figura 52.** Nueva sesión ya iniciada en MobaXterm.

Fuente: Autoría propia.

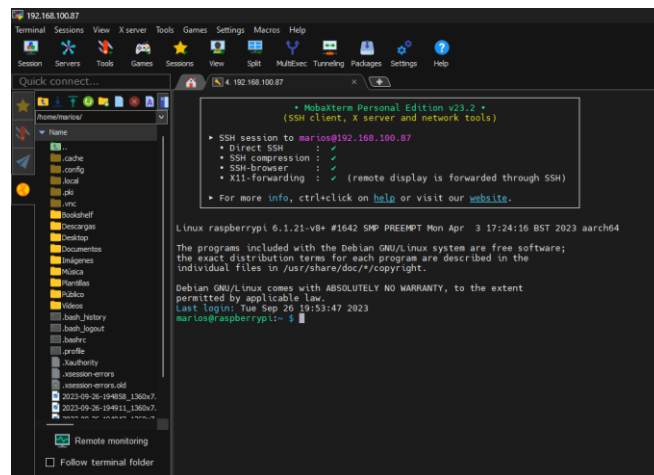
Para lograr ingresar la dirección IP es importante seleccionar la opción SSH que es la que va a permitir realizar dicha conexión remota, la manera de cómo se debe ubicar la dirección ip se muestra en la Figura 53.



**Figura 53.** Configuración de SSH con la dirección Ip.

Fuente: Autoría propia.

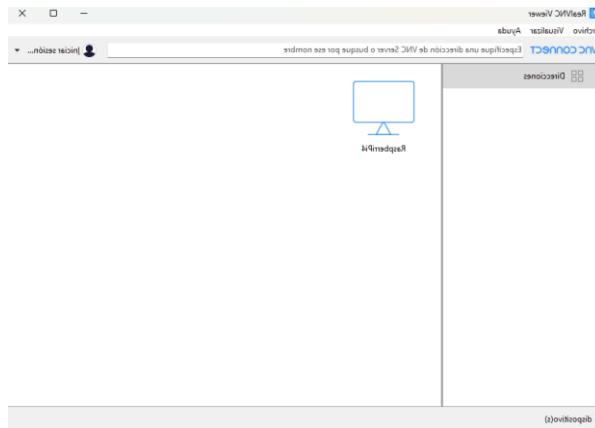
Una vez configurado lo anterior, al momento de intentar acceder a la conexión remota se debe ubicar el usuario y contraseña previamente creados en la configuración básica de la Raspberry, una vez ubicado estos datos, la conexión estará lista y se procederá a usar RealVNC Viewer, para asegurar que la conexión esta correcta debe salir un apartado igual al de la Figura 54.



**Figura 54.** Conexión correcta de MobaXterm

Fuente: Autoría propia.

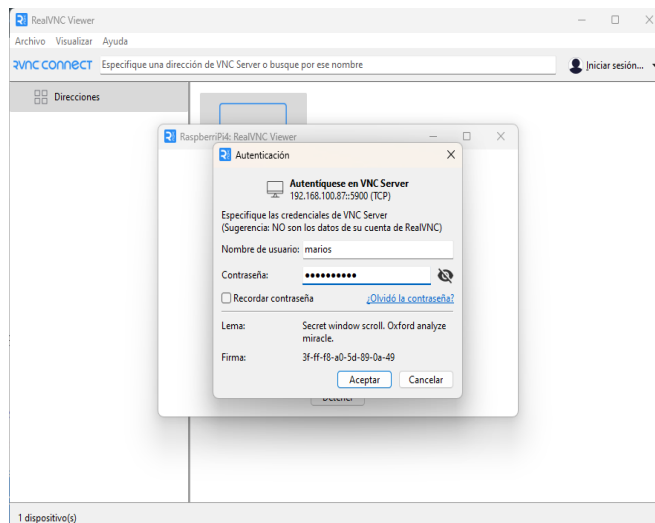
Una vez realizada la conexión se procede a usar el programa RealVnc Viewer para poder visualizar la pantalla de la Raspberry y realizar las acciones para poder funcionar el proyecto, la interfaz del programa se logra apreciar en la Figura 55.



**Figura 55.** Interfaz RealVNC viewer.

Fuente: Autoría propia.

Para poder conectarse mediante este programa, es necesario realizar una nueva conexión en la cual se debe proporcionar el nombre de usuario y contraseña registrados por la Raspberry, como se muestra en la Figura 56.



**Figura 56.** Usuario y contraseña para RealVNC.

Fuente: Autoría propia.

Finalmente se realiza la conexión, cabe destacar que al ser la primera conexión los registros del programa van a indicar que no se logra identificar la identidad, pero al saber que es el dispositivo se debe permitir la conexión, esta indicación logra observar en la Figura 57.



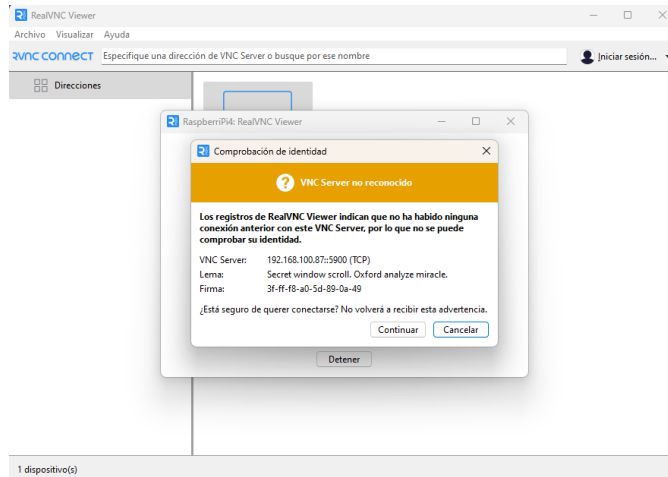


Figura 57. Permiso de conexión RealVNC.

Fuente: Autoría propia.

### 3.2.2.5 Configuraciones de librerías en Raspberry Pi.

Para poder configurar todas las librerías es necesario utilizar varios comandos los cuales son explicados a continuación.

Para poder instalar Python dentro de la Raspberry es necesario digitar el comando “sudo apt-get install Python”, tal como se muestra en la Figura 58.

```

0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
marios@raspberrypi:~$ sudo apt install python3-opencv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
 gdal-data libaec0 libarmadillo10 libarpack2 libcfitsio9 libcharls2 libdap27
 libdapclient6v5 libepsilon1 libfreexl1 libfyba0 libgdal28 libgdcm3.0
 libgeos-3.9.0 libgeos-c1v5 libgeotiff5 libhdf4-0-alt libhdf5-103-1
 libhdf5-hl-100 libheif1 libkmlbase1 libkmldom1 libkmlengine1 libleft5
 libmariadb3 libminizip1 libnetcdf18 libodbc1 libogdi4.1 libopencv-calib3d4.5
 libopencv-contrib4.5 libopencv-core4.5 libopencv-dnn4.5
 libopencv-features2d4.5 libopencv-flann4.5 libopencv-highgui4.5
 libopencv-imgcodecs4.5 libopencv-imgproc4.5 libopencv-ml4.5
 libopencv-objdetect4.5 libopencv-photo4.5 libopencv-shape4.5
 libopencv-stitching4.5 libopencv-video4.5 libopencv-videoio4.5 libpng5

```

Figura 58. Instalación de Python

Fuente: Autoría propia.

Para confirmar que si se instaló debe aparecer el apartado que se muestra en la Figura 59.

```
marios@raspberrypi:~$ sudo apt-get install python3
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.9.2-3).
fijado python3 como instalado manualmente.
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
```

**Figura 59.** Python instalado

Fuente: Autoría propia.

Una vez la Raspberry cuente con Python instalado también es necesario instalar Python IDLE, y esto se instala mediante el comando “sudo apt-get install idle”, cuando se esté instalando descargará ciertos requisitos para su funcionamiento tal como se muestra en la Figura 60.

```
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 368 kB de archivos.
Se utilizarán 1.396 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf idle-python3.9 all 3.9.2-1+rp1 [358 kB]
Des:2 http://mirror.cedia.org.ec/raspbian/raspbian bullseye/main armhf idle all 3.9.2-3 [2.020 B]
Descargados 360 kB en 2s (171 kB/s)
Seleccionando el paquete idle-python3.9 previamente no seleccionado.
(Leyendo la base de datos ... 106596 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../idle-python3.9_3.9.2-1+rp1_all.deb ...
Desempaquetando idle-python3.9 (3.9.2-1+rp1) ...
Seleccionando el paquete idle previamente no seleccionado.
Preparando para desempaquetar ../archives/idle_3.9.2-3_all.deb ...
Desempaquetando idle (3.9.2-3) ...
Configurando idle-python3.9 (3.9.2-1+rp1) ...
Configurando idle (3.9.2-3) ...
Procesando disparadores para mailcap (3.69) ...
Procesando disparadores para desktop-file-utils (0.26-1) ...
Procesando disparadores para gnome-menus (3.36.0-1) ...
Procesando disparadores para man-db (2.9.4-2) ...
```

**Figura 60.** Instalación Python IDLE.

Fuente: Autoría propia.

Para instalar todas las librerías necesarias para poder probar el algoritmo, para esto se empieza instalando la librería Opencv, la cual se debe escribir el comando “sudo apt install python3-opencv” una vez instalado ya se podrá importar dicha librería al código, para saber si se instaló debe aparecer un apartado como el que se muestra en la Figura 61.

```
marios@raspberrypi:~ $ sudo apt install python3-opencv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es nec
```

**Figura 61.** Instalación de librería Open CV.

Fuente: Autoría propia.

Para poder instalar la librería Numpy es necesario usar el comando “sudo pip3 install –upgrade numpy” con esta librería presenta varios problemas para ser encontrada es por esto que para poder instalarla se debe asegurar que versión de Python está instalado, una vez instalada al igual que la librería Opencv esta otra librería ya puede ser importada a el programa, en la Figura 62 muestra cuando ya está instalada la librería.

```
marios@raspberrypi:~ $ sudo pip3 install --upgrade numpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: numpy in /usr/lib/python3/dist-packages (1.19.5)
Collecting numpy
```

**Figura 62.** Instalación de librería Numpy

Fuente: Autoría propia.

Para que funcione el algoritmo tambien es necesaria la librería scipy es por esto que se tiene que seguir una serie de pasos para poder instalarla, ya que al igual de numpy, presenta problemas al instalar, es por esto que primero se debe hacer un llamado con el siguiente comando “sudo apt install libatlas-base-dev” una vez ejecutado el comando descargara dos archivos adicionales, como se muestra en la Figura 63.

```
marios@raspberrypi:~ $ sudo apt install libatlas-base-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libatlas3-base
Paquetes sugeridos:
  libatlas-doc liblapack-doc
Se utilizarán los siguientes paquetes NUEVOS:
  libatlas-base-dev libatlas3-base
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 5.055 kB de archivos.
Se utilizarán 26,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libatlas3-base
e armhf 3.10.3-10+rp1i [2.454 kB]
```

Figura 63. Ejecución de comando para buscar librerías automáticamente.

Fuente: Autoría propia.

Una vez realizado lo anterior se debe ejecutar el comando “pip3 install scipy” cabe destacar que este comando lo que hace es buscar la librería adecuada y la que se puede instalar en nuestro dispositivo, cabe destacar que esto se realiza gracias al comando mencionado en el apartado anterior, finalmente ya se puede importar esta librería a nuestro algoritmo, para saber que está instalada debe aparecer un apartado similar al de la Figura 64.

```
marios@raspberrypi:~ $ pip3 install scipy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting scipy
  Downloading scipy-1.11.2.tar.gz (56.0 MB)
  |████████████████████████████████████████| 56.0 MB 3.2 kB/s
Installing build dependencies ... done
Getting requirements to build wheel ... done
Installing backend dependencies ... done
Preparing wheel metadata ... error
```

Figura 64. Instalación de librería Scipy.

Fuente: Autoría propia.

Finalmente, se debe ejecutar los comandos “sudo apt update” y “sudo apt upgrade” para verificar que todo está actualizado, como se muestra en la Figura 65.

```
marios@raspberrypi:~ $ sudo apt update
Obj:1 http://archive.raspberrypi.org/debian bullseye InRelease
Obj:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
marios@raspberrypi:~ $ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
```

**Figura 65.** Actualización de todas las librerías.

Fuente: Autoría propia.

### 3.2.2.6 Diseño 3D del dispositivo electrónico.

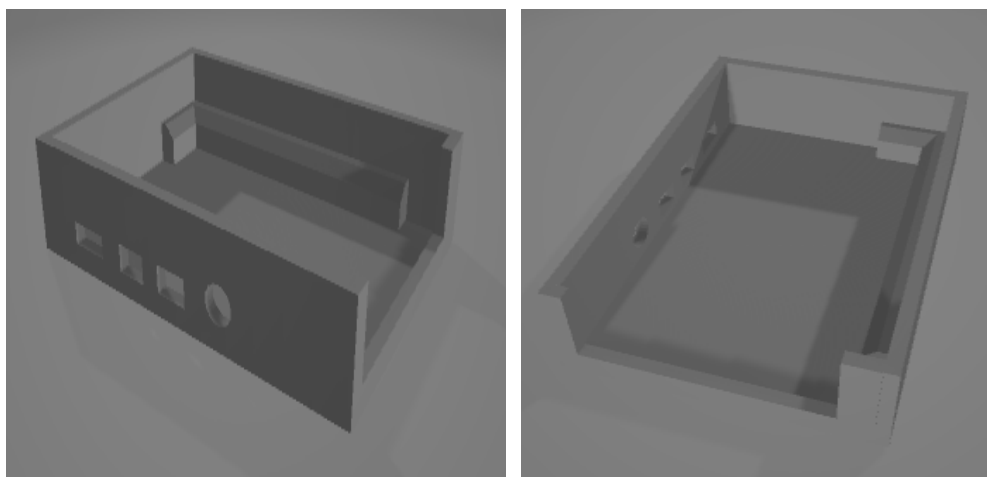
Para poder realizar el diseño del dispositivo fue necesario realizarlo en el software de Solidworks, el diseño se logra observar en la Figura 66.



**Figura 66.** Diseño 3D del dispositivo electrónico.

Fuente: Autoría propia.

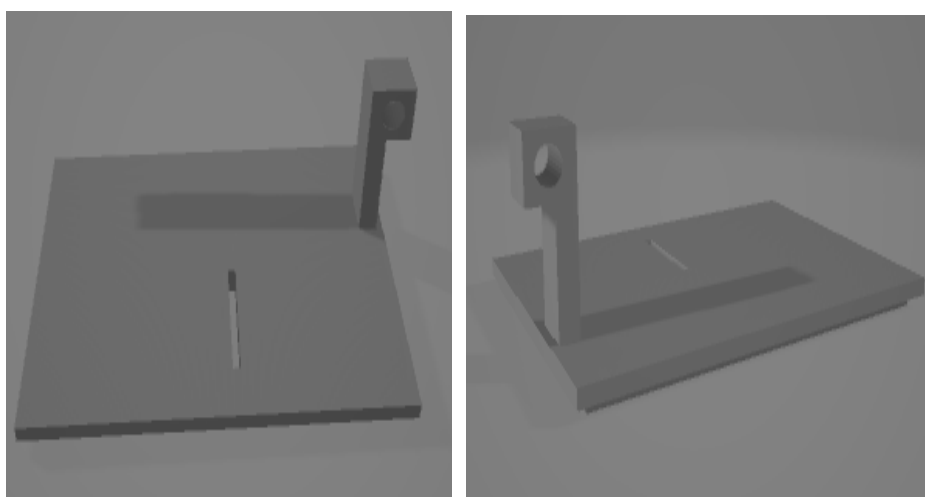
La estructura consta de la base que es la encargada de alojar la placa Raspberry pi 4, tal como se puede observar en la Figura 67.



**Figura 67.** Base del diseño.

Fuente: Autoría propia.

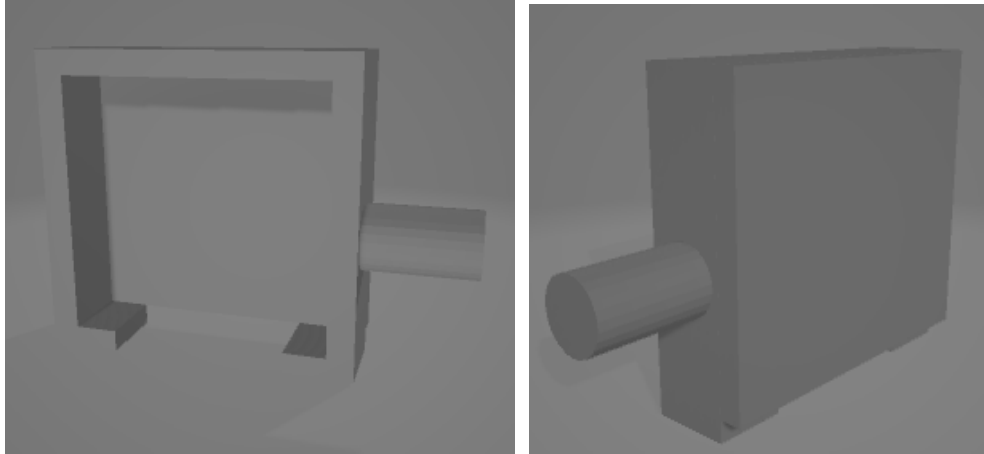
La tapa que tiene una abertura que es por donde pasara el flex de la cámara Raspberry OV5647, como se puede observar en la Figura 68.



**Figura 68.** Tapa de la base.

Fuente: Autoría propia.

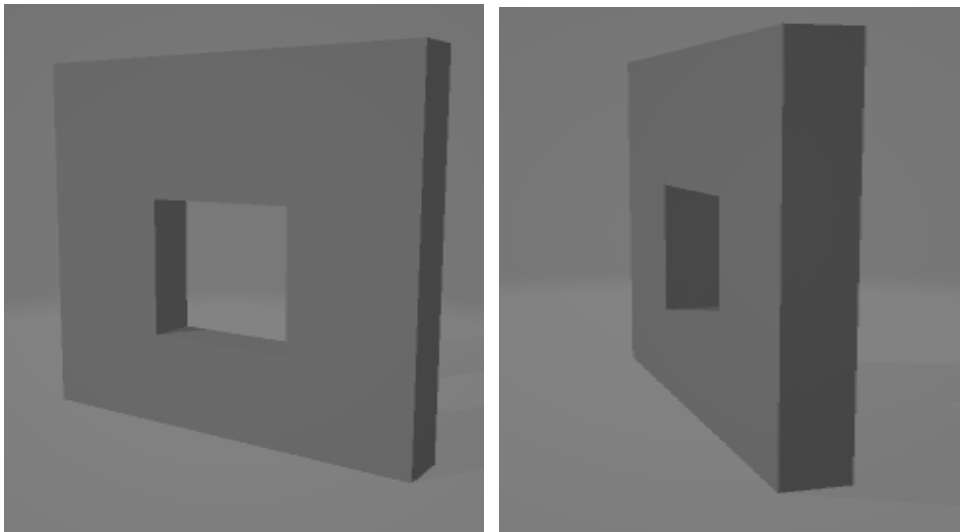
La base donde se alojará la cámara que permitirá adaptar los movimientos de abajo hacia arriba, se puede observar en la Figura 69.



**Figura 69.** Base para la cámara.

Fuente: Autoría propia.

Para la cámara también es necesario colocarle un recubrimiento para que se logre quedar fija, se puede observar en la Figura 70.



**Figura 70.** Tapa de la cámara.

Fuente: Autoría propia.

La impresión que se realizó cuenta con todas las características previamente mencionadas, puede observarse en la Figura 71.

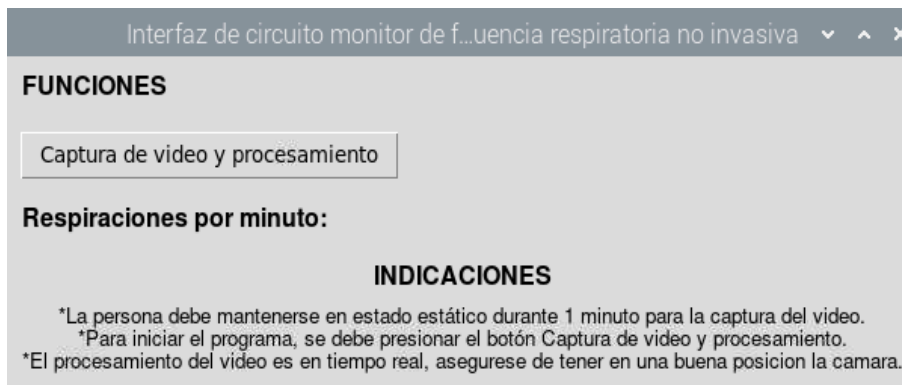


**Figura 71.** Impresión 3D de la estructura.

Fuente: Autoría propia.

### 3.2.3 Interfaz gráfica.

Para poder crear la interfaz gráfica se debe usar la librería tkinter, que permite realizar interfaces en Python, para esto es necesario agregar unas instrucciones antes de inicializar con los monitoreos., para esto es necesario agregar unas instrucciones antes de inicializar con los monitoreos, es decir que para poder inicializar el programa se debe presionar un botón, tal como se muestra en la Figura 72.

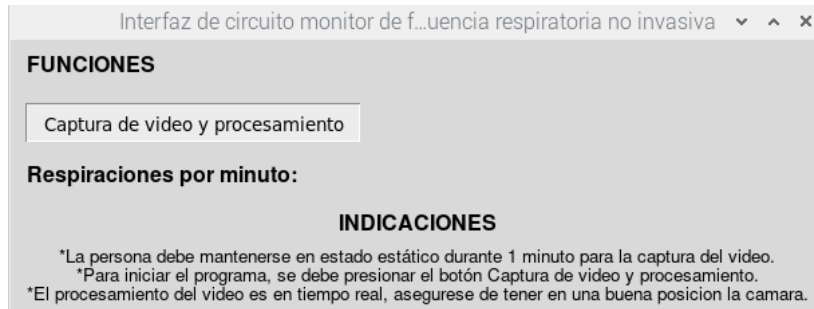


**Figura 72.** Interfaz gráfica.

Fuente: Autoría Propia.

La interfaz está constituida de un botón el cual permite inicializar todo el programa, el botón es llamado “captura de video y procesamiento”, una vez inicializado el programa el botón permanece bloqueado hasta que termine el monitoreo, el bloqueo del botón se puede observar en la Figura 73.

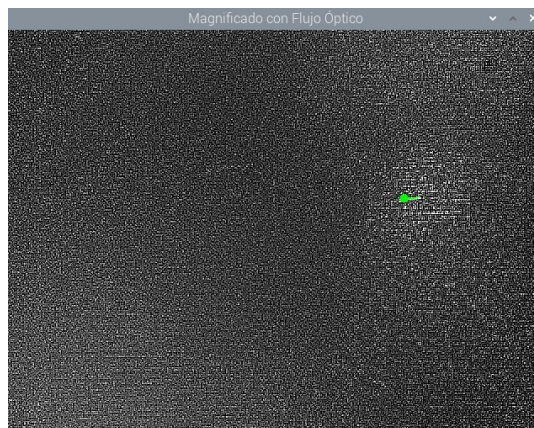




**Figura 73.** Bloqueo de botón durante procesamiento.

Fuente: Autoría propia.

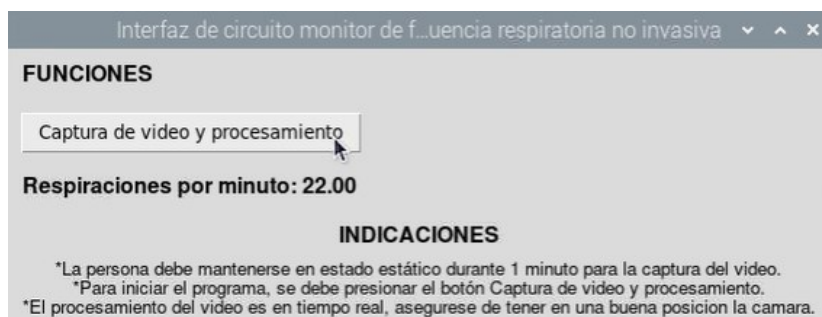
Una vez inicializado el programa, obtendremos una ventana llamada “Magnificado con flujo óptico” donde se muestra como está analizando el video magnificado y como está identificando los movimientos, en esta ventana se logra observar la posición de la persona y como su movimiento se vuelve permisible, en la Figura 74 se logra apreciar la ventana que nos arroja el dispositivo electrónico.



**Figura 74.** Ventana para análisis en tiempo real.

Fuente: Autoría propia.

Tambien cuenta con un apartado donde se imprimirán las respiraciones por minuto donde al iniciar el programa se encontrará vacía hasta que se realice un monitoreo, cabe destacar que las respiraciones son contabilizadas por el número de inhalaciones que realiza la persona durante un minuto, cabe destacar que este valor se actualiza con respecto a cada monitoreo, en la Figura 75 se logra observar cuando se imprime el valor luego del monitoreo.



**Figura 75.** Valor de respiración impreso en la interfaz gráfica.

Fuente: Autoría propia.

### 3.3. Estudio de Factibilidad

Para determinar el costo de la propuesta de la implementación de un dispositivo electrónico de monitoreo a través de visión artificial que permita realizar mediciones de frecuencia respiratoria no invasiva, se detallan los costos en la Tabla 8 [82].

**Tabla 8.** Valores para la implementación de la propuesta.

Cantidad	Descripción	Costo
1	Raspberry pi 4.	\$200,00
1	Arduino Mega 2560.	\$25,00
1	Sensor LM35	\$3,80
7	Resistencias	\$2,25
1	Amplificador TL084CN	\$3,00
1	Cámara Raspberry pi OV5647	\$45,00
1	Impresión de la carcasa 3D.	\$10.00
<b>TOTAL</b>		<b>\$289,05</b>

### **3.4. Resultados**

Una vez se obtienen los funcionamientos tanto del dispositivo electrónico de monitoreo de frecuencia respiratoria no invasiva y el Circuito monitor de frecuencia respiratoria con Arduino mega 2560 y termistor LM35 se procede a ejecutar las pruebas respectivas, donde ambos sistemas funcionaran de forma simultánea.

#### **3.4.1 Pruebas empleadas.**

En las pruebas empleadas se detalla todos los diferentes pasos que se utilizaron para cuantificar y determinar el comportamiento y desempeño del circuito desarrollado.

Cada prueba consiste en grabar un video con duración de 1 minuto (60 segundos) a una persona tomando varias consideraciones para lograr garantizar el funcionamiento correcto del algoritmo, cabe destacar que durante cada grabación la persona que se encuentra realizando la prueba debe usar el sensor de temperatura LM35, es decir que ambos circuitos funcionan en conjunto, esto se realiza con el fin de comprar ambas medidas de frecuencia respiratoria en la cual una no usa ningún tipo de sensor y el otro si, además la persona a la que se está realizando el análisis cuenta las respiraciones que realiza y a su vez una persona externa tambien cuenta las respiraciones.

Las diferentes pruebas realizadas deben contar con un tipo de iluminación específico para que el circuito funcione correctamente en este caso se usó la iluminación direccional, cabe destacar que la luz impacta directamente a la persona, se puede observar en la Figura 76.

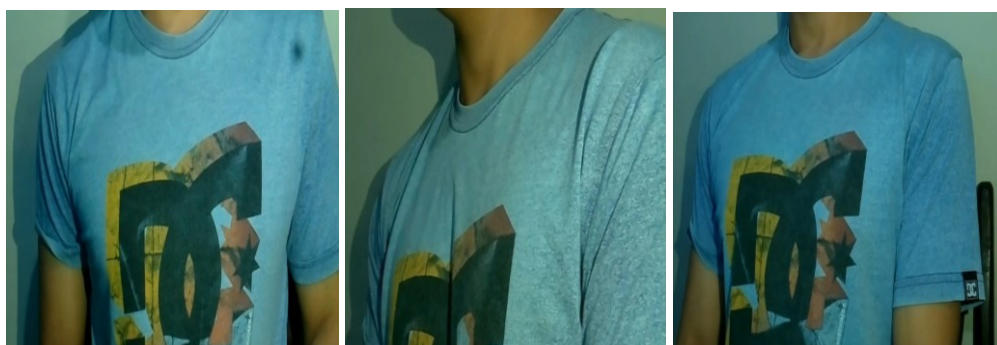


**ILUMINACION  
DIRECCIONAL**

**Figura 76.** Iluminaciones usadas en las pruebas respectivas.

Fuente: Autoría propia.

Adicionalmente, la persona que se encuentra realizando la prueba sentada frente a la cámara opta por tomar tres posiciones diferentes, posición frontal, lateral y la última con un Angulo de 45 con respecto a la cámara, como se muestra en la Figura 77.



**Figura 77.** Posición frontal, lateral y a 45 grados.

Fuente: Autoría propia.

Para las pruebas el procedimiento se realizó a 6 personas, en las cuales tomaron las posiciones mencionadas.

El video capturado debe tener una duración de un minuto, a la cual se le aplica la magnificación euleriana de video en tiempo real, a su vez también se aplica el método de optical Flow y el conteo respectivo de las respiraciones.

Para poder realizar el análisis es necesario tomar en cuenta los movimientos que se van analizar, para esto en el código es necesario ubicar dichos parámetros de movimiento para que sean considerados como respiraciones, y esto se logra con la

siguiente línea de código para inhalación  $umbral\_inhalacion = 2.75$  la cual puede ser ajustada según los requerimientos que se necesite, en cuanto a la exhalación es una línea bastante parecida que se constituye de la siguiente forma  $umbral\_exhalacion = -2.75$  a su vez puede ser parametrizada según los requerimientos que se desee analizar, estos cambios influyen ya que a mayor valor el movimiento debe verse más reflejado.

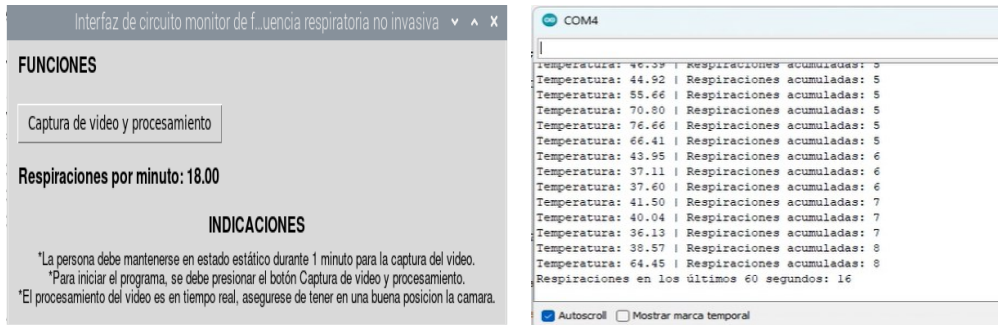
Además, como se mencionó que el monitoreo es en tiempo real, se ubicó una fórmula para que cumpla la función de conteo de respiraciones en base a la dirección de respiración que realiza la persona, donde la formula a utilizar se muestra en la siguiente línea de código  $Respiraciones\ por\ minuto = (conteo\_inhalacion / duracion) * 60$  como se observa contabiliza las inhalaciones realizadas sobre el tiempo establecido (1 min), es contabilizada de esta forma, ya que varios artículos científicos demuestran que la mejor manera de contar las respiraciones es mediante las inhalaciones que realiza una persona por el transcurso de un minuto.

### Valores obtenidos de los resultados.

El procesamiento consto de la obtención de 40 videos de las diferentes personas de prueba, tomando en cuenta las diferentes posiciones, cabe destacar que los videos fueron analizados con diferentes cambios en el algoritmo en cuanto al análisis de flujo óptico.

Los datos obtenidos en posición lateral se observan en la Figura 78, en donde se encuentra una imagen a la persona que se le está realizando el monitoreo, obteniendo los valores mediante la interfaz gráfica y el monitor serial del Arduino.





**Figura 78.** Resultados obtenidos de la primera persona en posición lateral.

Fuente: Autoría propia.

En la Figura 79, se logra apreciar el análisis a la misma persona, pero esta vez en posición frontal, los resultados obtenidos también se muestran mediante la interfaz gráfica y el monitor serial.



**Figura 79.** Resultados obtenidos de la primera persona en posición frontal.

Fuente: Autoría propia.

En la Figura 80 se observa el monitoreo a la primera persona, pero en este caso en una posición de 45 grados con respecto a la cámara, de la misma manera su resultado del monitoreo puede apreciarse en la interfaz gráfica y el monitor serial.



**Figura 80.** Resultados obtenidos de la primera persona en posición 45 grados.

Fuente: Autoría propia.

Con el fin de conservar todos los datos obtenidos de la primera prueba, se procede a detallar en la Tabla 9 los conteos respectivos de cada monitoreo.

**Tabla 9.** Datos obtenidos de la primera persona.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
1	Lateral	17	17	16	18
	Frontal	19	19	18	20
	45	18	18	17	19

También como protocolo de pruebas para su respectivo análisis a la primera persona se pidió que realice un poco de ejercicios y posteriormente realizar la prueba nuevamente.

En posición lateral la persona obtuvo los resultados mostrados en la Figura 81.



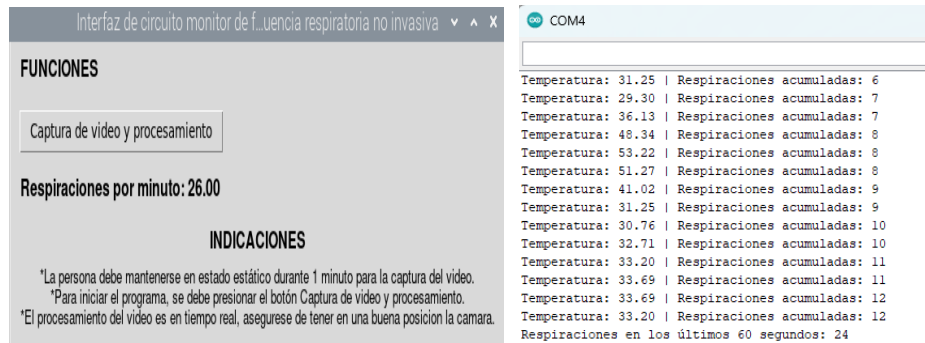
**Figura 81.** Resultados obtenidos de la primera persona agitada en posición lateral.

Fuente: Autoría propia.

En posición frontal la persona obtuvo tanto en el circuito no invasivo y el circuito con el sensor LM35 los resultados mostrados en la Figura 82



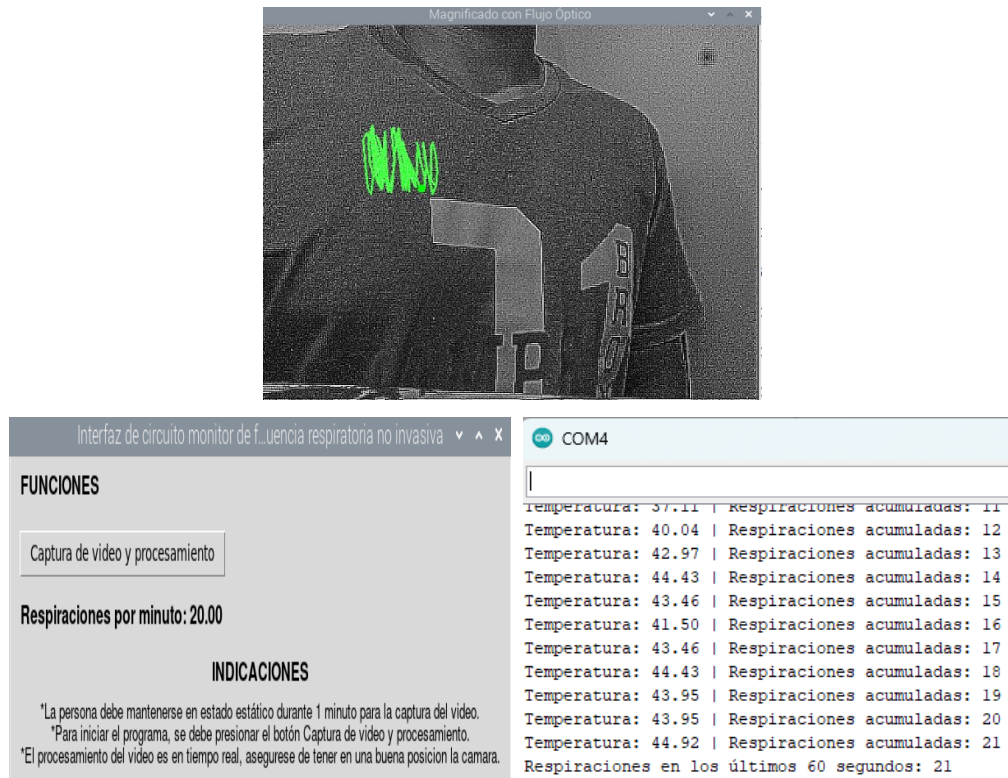




**Figura 82.** Resultados obtenidos de la primera persona agitada en posición frontal.

Fuente: Autoría propia.

En la posición con ángulo de 45 grados respecto a la cámara, la persona obtuvo los resultados que se observan en la Figura 83.



**Figura 83.** Resultados obtenidos de la primera persona agitada en posición 45 grados.

Fuente: Autoría propia.

Al igual que en la primera prueba también es importante guardar los datos de este segundo monitoreo, los datos obtenidos se pueden observar en la Tabla 10.

**Tabla 10.** Datos obtenidos de la primera persona con agitación.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
1	Lateral	23	23	22	23
	Frontal	25	24	24	26
	45	22	22	21	20

En cuanto a la siguiente prueba se destaca que se usó una prenda con un logotipo, para comprobar cómo se comportaba el dispositivo electrónico en esta situación.

En posición lateral los resultados que se obtuvieron tanto en la interfaz gráfica y en el monitor serie se pueden observar en la Figura 84.



**Figura 84.** Resultados obtenidos de la segunda persona en posición lateral.

Fuente: Autoría propia.

El monitoreo realiza en posición frontal se obtuvieron los resultados que se muestran en la Figura 85.



Interfaz de circuito monitor de frecuencia respiratoria no invasiva

**FUNCIONES**

Captura de video y procesamiento

**Respiraciones por minuto: 21.00**

**INDICACIONES**

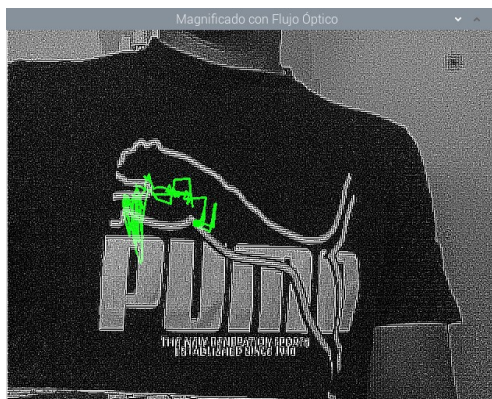
\*La persona debe mantenerse en estado estático durante 1 minuto para la captura del video.  
 \*Para iniciar el programa, se debe presionar el botón Captura de video y procesamiento.  
 \*El procesamiento del video es en tiempo real, asegúrese de tener en una buena posición la cámara.

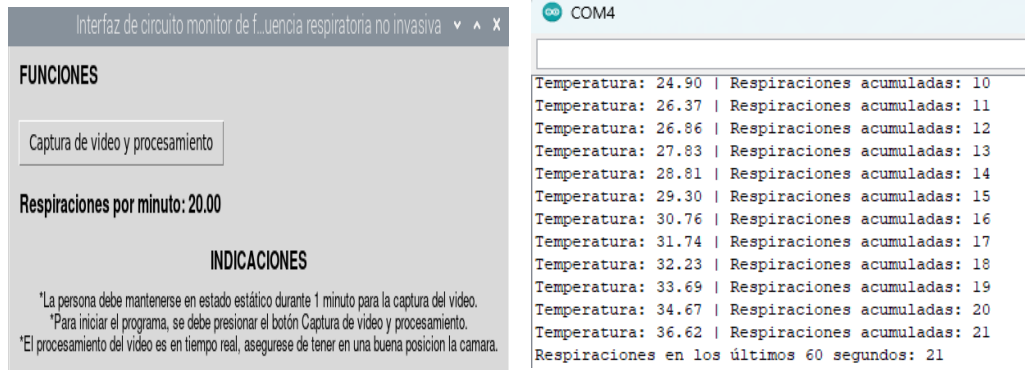
COM4

Temperatura: 61.52	Respiraciones acumuladas: 8
Temperatura: 54.20	Respiraciones acumuladas: 8
Temperatura: 44.92	Respiraciones acumuladas: 9
Temperatura: 36.13	Respiraciones acumuladas: 10
Temperatura: 35.64	Respiraciones acumuladas: 11
Temperatura: 35.64	Respiraciones acumuladas: 12
Temperatura: 35.64	Respiraciones acumuladas: 13
Temperatura: 35.16	Respiraciones acumuladas: 14
Temperatura: 36.62	Respiraciones acumuladas: 15
Temperatura: 36.13	Respiraciones acumuladas: 16
Temperatura: 34.67	Respiraciones acumuladas: 17
Temperatura: 33.69	Respiraciones acumuladas: 18
Respiraciones en los últimos 60 segundos: 18	

**Figura 85.** Resultados obtenidos de la segunda persona en posición frontal.

Los resultados que se obtuvieron en posición de 45 grados con respecto a la cámara, son los que se puede observar en la Figura 86.





**Figura 86.** Resultados obtenidos de la segunda persona en posición frontal.

Fuente: Autoría propia.

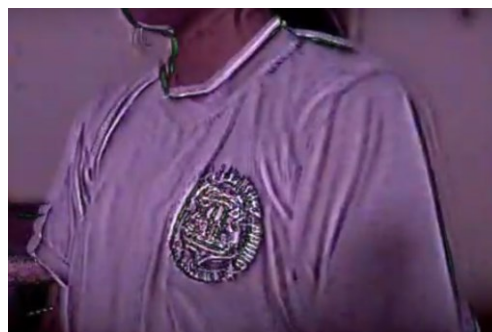
Al igual que en las dos pruebas anteriores, los datos que se obtuvo de esa prueba se muestran en la Tabla 11.

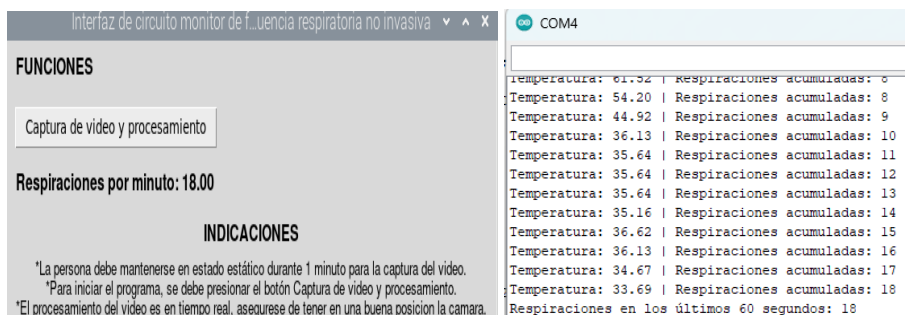
**Tabla 11.** Datos obtenidos de la segunda persona.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Circuito no invasivo
2	Lateral	19	19	20	19
	Frontal	20	20	18	21
	45	18	18	21	20

Para la prueba de la tercera persona que es de sexo femenino con edad de 12 años se obtuvieron los resultados variando el valor de amplificación, esto se debe a que la respiración de la persona era muy baja, pero el algoritmo respondió de forma excelente a estos cambios producidos.

En la posición lateral los resultados se pueden observar en la Figura 87

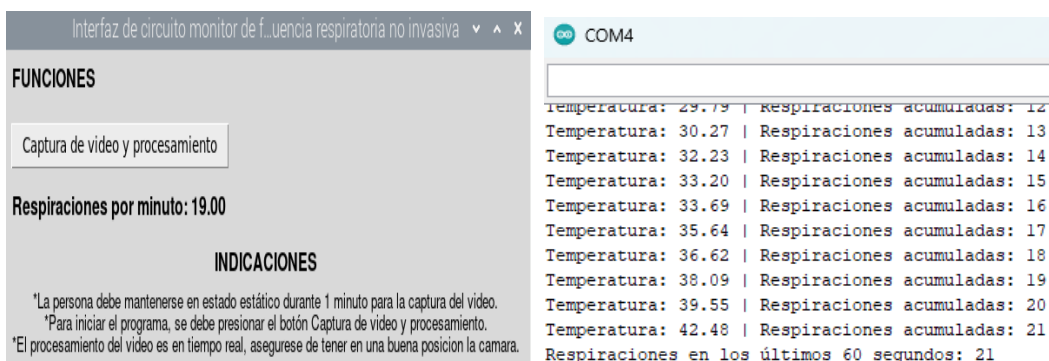
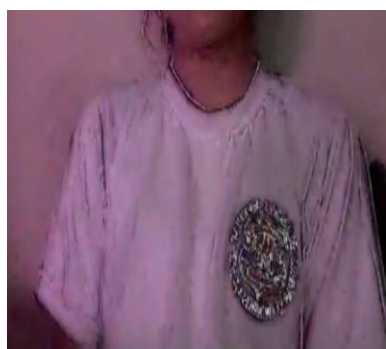




**Figura 87.** Resultados obtenidos de la tercera persona en posición lateral.

Fuente: Autoría propia.

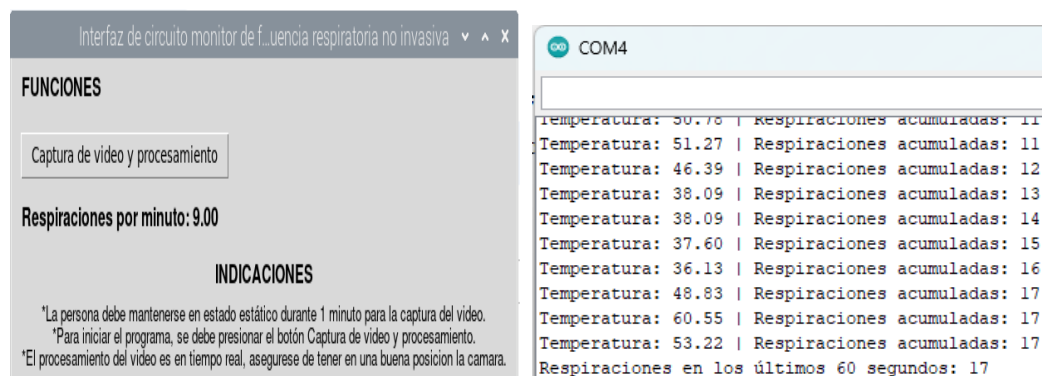
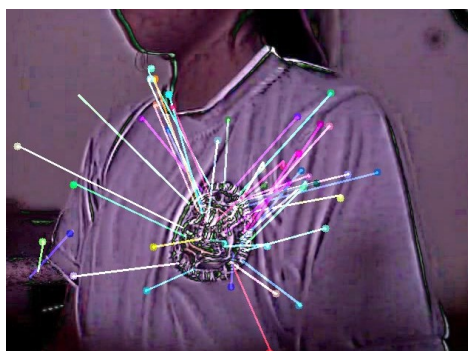
Los resultados del monitoreo en posición frontal se pueden observar en la Figura 88.



**Figura 88.** Resultados obtenidos de la tercera persona en posición frontal.

Fuente: Autoría propia.

En la prueba de monitoreo a 45 grados el sistema presento problemas de conteo y los valores obtenidos se pueden observar en la Figura 89.



**Figura 89.** Resultados obtenidos de la tercera persona en posición 45 grados.

Fuente: Autoría propia.

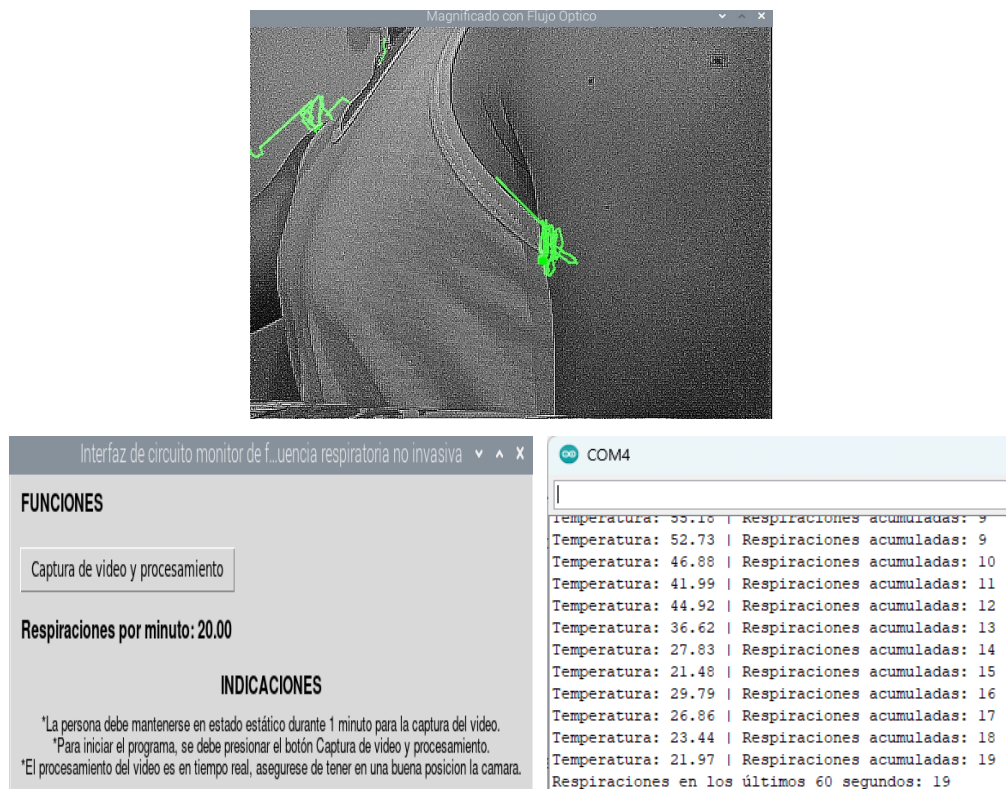
En la Tabla 12 se presentan los valores que se obtuvieron de esa persona en distintas posiciones, cabe destacar que su camiseta tenía un logo, por ende, la filtración para el conteo con el optical Flow vario de tal punto que en la prueba de 45 grados presento problemas.

**Tabla 12.** Datos obtenidos de la tercera persona.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
3	Lateral	21	20	18	18
	Frontal	20	20	21	19
	45	18	18	17	9

Para el caso de la cuarta persona se le realizo la prueba esta vez con una indumentaria básica, es decir, sin ningún logotipo, esto se realiza con el fin de observar que cambios pueden verse reflejados en prendas.

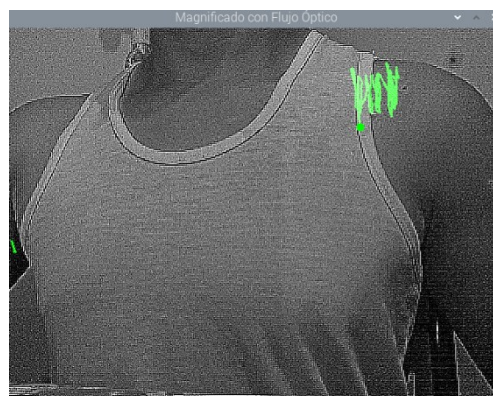
En cuanto a los resultados obtenidos tanto de la interfaz gráfica y el monitor serial del Arduino en posición lateral, se pueden observar en la Figura 90.



**Figura 90.** Resultados obtenidos de la cuarta persona en posición lateral.

Fuente: Autoría propia.

Para el caso de la persona en posición frontal, se obtuvieron los resultados mostrados en la Figura 91.



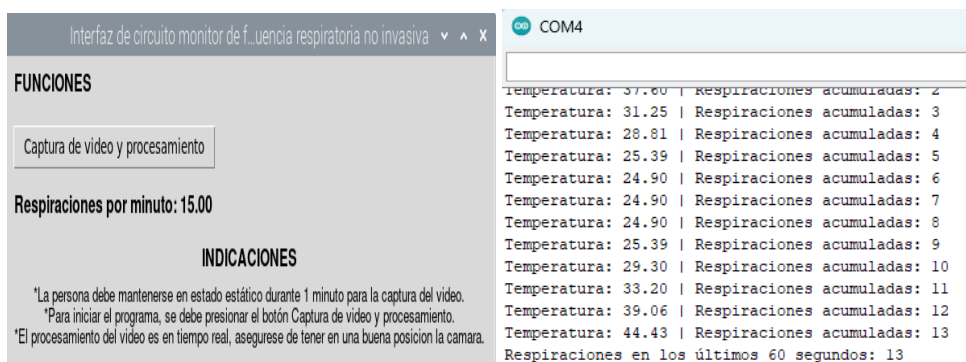


Figura 91. Resultados obtenidos de la cuarta persona en posición frontal.

Fuente: Autoría propia.

Los resultados obtenidos para la posición de 45 grados con respecto a la cámara, se pueden observar en la Figura 92.



Figura 92. Resultados obtenidos de la cuarta persona en posición 45 grados.

Fuente: Autoría propia.

De la misma forma que en las demás pruebas se necesita tener todos los datos agrupados y estos se pueden observar en la Tabla 13



**Tabla 13.** Datos obtenidos de la cuarta persona.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
4	Lateral	20	20	19	20
	Frontal	15	14	13	15
	45	17	17	17	18

En cuanto a la siguiente prueba en este caso la persona a realizar el monitoreo contaba con una indumentaria con varios colores, pero el algoritmo detecto más el contorno del pecho por lo que pudo realizar la prueba sin ningún inconveniente.

Los resultados del monitoreo en posición lateral se pueden observar en la Figura 93.



**Figura 93.** Resultados obtenidos de la quinta persona en posición lateral.

Fuente: Autoría propia.

Los valores obtenidos durante el monitoreo de la persona en posición frontal se muestran en la Figura 94.



**Figura 94.** Resultados obtenidos de la quinta persona en posición frontal.

Fuente: Autoría propia.

En el caso del posicionamiento de la persona a un ángulo de 45 grados con respecto a la cámara, los resultados obtenidos se muestran en la Figura 95.



**Figura 95.** Resultados obtenidos de la quinta persona en posición 45 grados.

Fuente: Autoría propia.

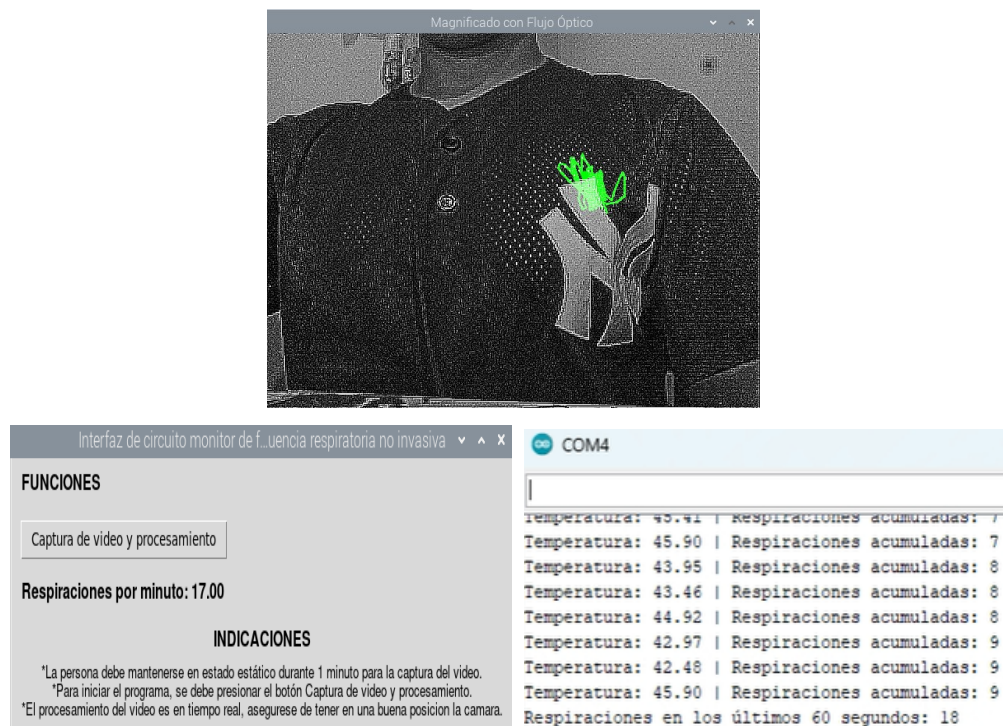
Los datos que se obtienen en este monitoreo se pueden observar en la Tabla 14.

**Tabla 14.** Datos obtenidos de la quinta persona.

Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
5	Lateral	19	19	18	20
	Frontal	17	17	17	18
	45	20	20	18	20

Para el monitoreo de la última persona se optó por cambiar un poco la iluminación con el fin de que no se observen tantos detalles como en las otras pruebas, este cambio de cierto modo mejoro el análisis del algoritmo en cuanto a la posición frontal y la de 45 grados.

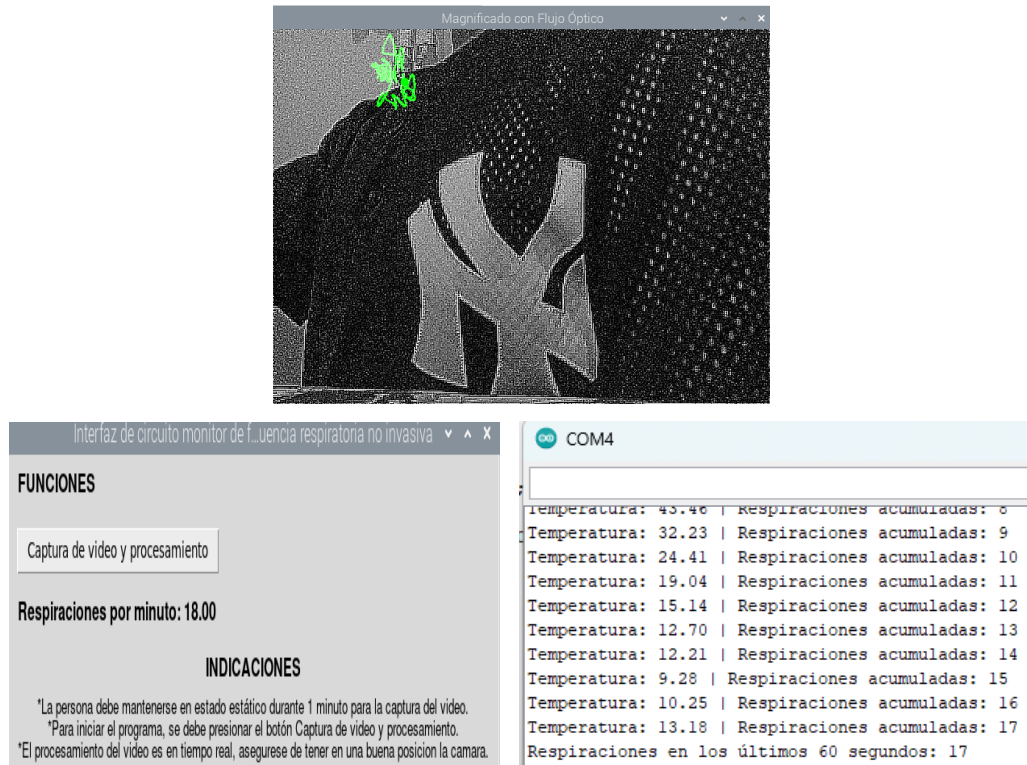
Los datos obtenidos en esta prueba en posición frontal, son los que pueden apreciarse en la Figura 96.



**Figura 96.** Resultados obtenidos de la sexta persona en posición lateral.

Fuente: Autoría propia.

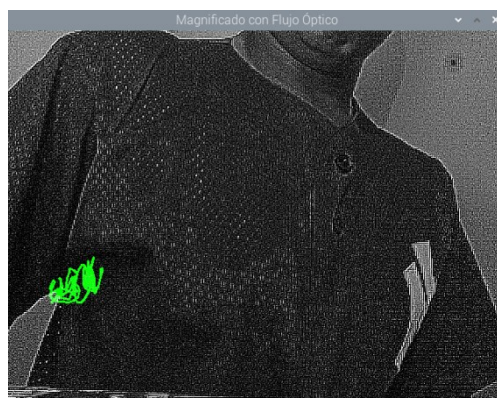
En cuanto a la posición frontal de la persona los resultados obtenidos en este monitoreo se pueden observar en la Figura 97.

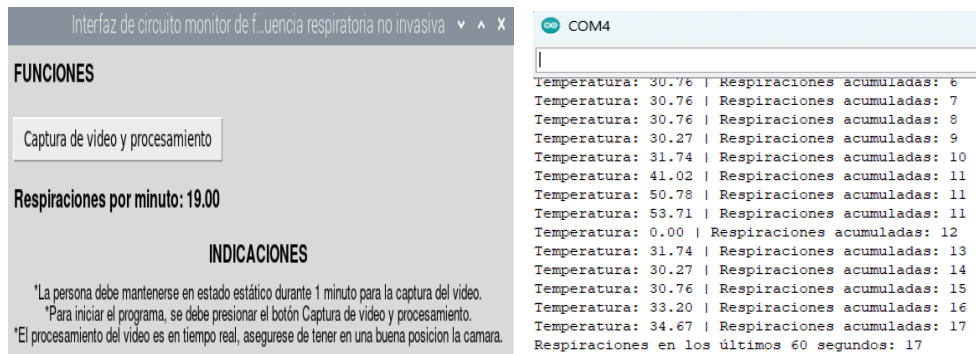


**Figura 97.** Resultados obtenidos de la sexta persona en posición frontal.

Fuente: Autoría propia.

Por ultimo, para la posición de la persona en un ángulo de 45 grados con respecto a la cámara se obtuvieron los resultados que se muestran en la Figura 98.





**Figura 98.** Resultados obtenidos de la sexta persona en posición 45 grados.

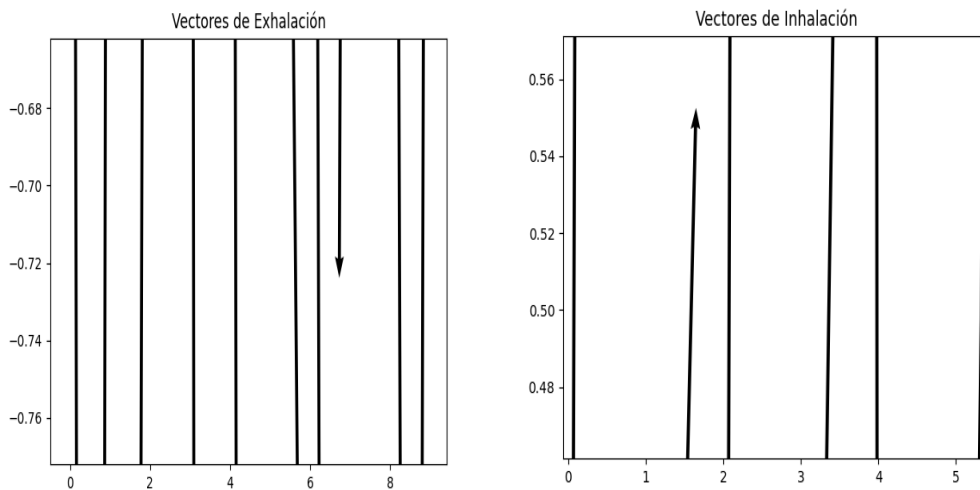
Fuente: Autoría propia.

Los resultados agrupados de esta prueba se logran apreciar en la Tabla 15.

**Tabla 15.** Datos obtenidos de la sexta persona.

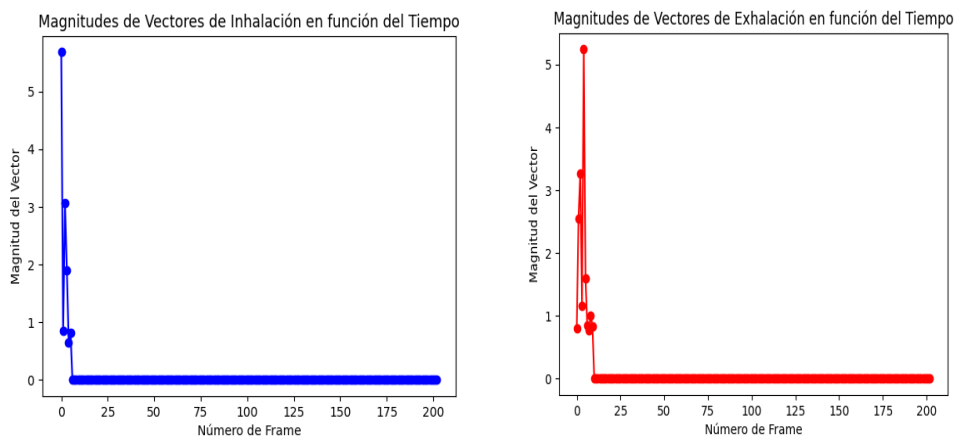
Persona	Vista	Real r/m	Observador	Circuito Arduino	Dispositivo electrónico no invasivo
6	Lateral	19	19	18	17
	Frontal	18	18	17	18
	45	18	18	17	19

Para lograr corroborar el funcionamiento del algoritmo se realizó de manera grafica los resultados que se presentaron, esto con el fin de mostrar cuando el algoritmo detectaba una inhalación o una exhalación, además verificar la magnitud de los vectores de cada inhalación, los gráficos de los vectores pueden observarse en la Figura 99, además destacando que estos pueden variar con respecto a cada análisis, por ende, pueden aumentar o disminuir.



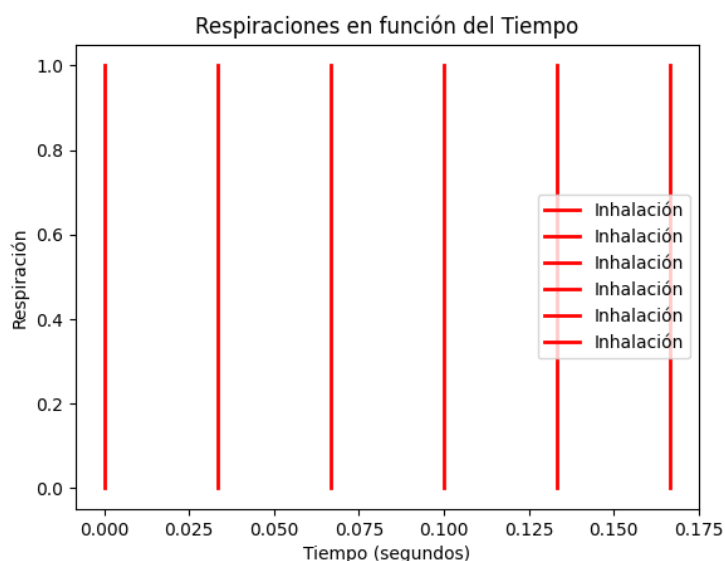
**Figura 99.** Vectores de movimiento en pruebas.

En la Figura 100 se muestran las magnitudes de los vectores a través del tiempo que se realizó dicha prueba, esto se utiliza para poder verificar la funcionalidad del algoritmo.



**Figura 100.** Magnitud de vectores en pruebas.

También es importante destacar que el algoritmo debe entregar las respiraciones por minuto de cada prueba, pero como se está trabajando con videos de 20 segundos de duración, se debe multiplicar por una constante aun así se muestra de manera gráfica las respiraciones reales que se está realizando en el transcurso del video, esto puede observarse en la Figura 101 que es donde muestra dicho gráfico de Respiraciones por minuto.



**Figura 101.** Respiraciones por minuto de pruebas.

Para determinar la precisión del algoritmo se tomaron varios parámetros analizar, como ya se sabe las pruebas fueron realizadas en diferentes direcciones por lo tanto se debe determinar el porcentaje de error de cada uno de las pruebas. Para realizar esta actividad se debe utilizar la Ecuación 17.

$$\%error = \frac{\text{valor del algoritmo} - \text{valor real}}{\text{valor real}} * 100\%$$

**Ecuación 17.** Fórmula para determinar el porcentaje de error.

Para determinar el porcentaje de error que existe, se empezó a determinar la precisión de cada persona en la posición frontal, destacando que solo se está tomando en cuenta el valor real y el valor otorgado por el circuito electrónico de monitoreo no invasivo, los valores de precisión se pueden observar en la Tabla 16.

**Tabla 16.** Precisión del algoritmo en posición frontal.

Posición: Frontal	Precisión del Dispositivo electrónico no invasivo
Persona 1	95,35%
Persona 2	95,24%
Persona 3	95%

Persona 4	100%
Persona 5	94,12%
Persona 6	100%

Ahora la posición que se procede analizar es la lateral, tomando en cuenta que el algoritmo puede variar las mediciones en cuanto a valores que pueden presentar errores por la misma iluminación, estos valores pueden observarse en la Tabla 17.

**Tabla 17.** Precisión del algoritmo en una posición lateral.

Posición: Lateral	Precisión del dispositivo electrónico no invasivo
Persona 1	97,06%
Persona 2	100%
Persona 3	85,74%
Persona 4	100%
Persona 5	94,74%
Persona 6	89,47%

Posteriormente se procede a determinar la precisión del circuito electrónico de monitoreo no invasivo pero esta vez para una persona con una posición de 45 grados con respecto a la cámara, estos valores se observan en la Tabla 18.

**Tabla 18.** Precisión del algoritmo en una posición de 45 grados.

Posición: 45 grados	Precisión del Dispositivo electrónico no invasivo
Persona 1	92,69%
Persona 2	88,89%
Persona 3	50%
Persona 4	94,12%



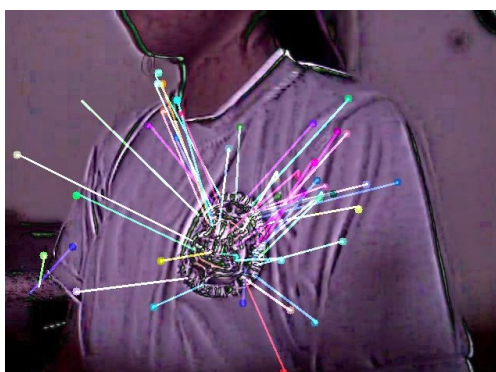
Persona 5	100%
Persona 6	94,45%

Con los datos obtenidos se puede hacer un porcentaje general de la precisión con respecto a la posición que se encontraban todas las personas y el circuito electrónico de monitoreo no invasivo, esta precisión total del algoritmo con respecto a la posición puede observarse en la Tabla 19.

**Tabla 19.** Porcentaje total de precisión con respecto a la posición.

Posición	Precisión del dispositivo electrónico no invasivo
Frontal.	96,61%
Lateral.	94,50%
45 grados.	86,69%

En los datos obtenidos de la tercera persona se mostró un grave error en cuanto a los valores obtenidos para el análisis de flujo óptico, al analizar los vectores de movimiento en ese video, se logró observar la razón por la que ocurría el error, era debido al logotipo que se encuentra en su indumentaria, la razón de que el algoritmo no detecta de forma correcta ese objeto, se debe a que las regiones del logotipo están bien definidas, por lo tanto al magnificar el video el algoritmo toma como referencia a todo ese contorno y por ende es lo que cuenta, en la Figura 102 se muestra como está obteniendo los vectores en ese monitoreo.



**Figura 102.** Error en la sección analizar.

Fuente: Autoría propia.

Para de cierto modo evitar esta confusión del conteo, se debe reducir de cierta forma la magnificación, con el único fin de que al realizar los movimientos de inhalación y exhalación se observen movimientos más reducidos y el análisis de flujo óptico pueda enfocarse en el contorno realmente que se requiere analizar, en este caso en todo el pecho.

También es importante comparar los resultados obtenidos entre ambos circuitos usados, es decir, que ahora para calcular el error se deberá seguir usando la Ecuación 1 para determinar la precisión del circuito con el termistor LM35. Es por esto que solo se toma en cuenta los valores obtenidos en la prueba de posición frontal, que fue donde se desarrolló mejor el circuito no invasivo, los valores obtenidos de precisión para el circuito con el Arduino y el termistor LM35 se logran observar en la Tabla 20.

**Tabla 20.** Precisión del circuito con Arduino y LM35.

Personas	Precisión del circuito con Arduino y LM35.
1	94.74%
2	90%
3	95%
4	86,67%
5	100%
6	95,45%

Ahora se realiza una tabla comparativa entre estos dos circuitos como ya se mencionó anteriormente, los resultados se observan en la Tabla 21.

**Tabla 21.** Comparativa de precisión entre ambos circuitos.

Circuito	Porcentaje de precisión.
Dispositivo electrónico de monitoreo no invasivo.	92,6%

Circuito con Arduino y termistor LM35.	93,64%
--	--------

Para corroborar que tanto el dispositivo electrónico de monitoreo de frecuencia respiratoria no invasiva y el Circuito monitor de frecuencia respiratoria con Arduino mega 2560 y termistor LM35 se tomó como referencia el conteo que realizaba la persona que se estaba realizando la prueba

El resultado de todas estas pruebas realizadas se observa que el circuito con LM35 tiende a ser un poco más preciso que el de visión artificial por un porcentaje de diferencia de 1,04%, esto se debe a que las pruebas que se realizaron las sombras afectaron de cierto modo al análisis del conteo de respiraciones.

El circuito monitor con el LM35 funciono como referencia para comprobar la precisión del dispositivo electrónico de monitoreo a través de visión artificial con lo que se demostró el buen funcionamiento de ambos dispositivos.

## CONCLUSIONES

El algoritmo de magnificación euleriana de video fue adaptado de manera exitosa, debido a que al realizar pruebas se observaba que el video entregado cumplía con todas las características planteadas en la cual para obtenerlo se aplicó pirámide gaussiana, pirámide laplaciana, filtro ideal temporal de banda, filtro de banda butterworth , permitiendo el paso a la etapa de análisis de flujo óptico donde se aplicaron técnicas de reconocimiento mediante vectores donde se logró diferenciar las inhalaciones de las exhalaciones para el respectivo conteo de respiraciones.

Al realizar el análisis de movimiento en el transcurso del video, se pudo determinar que el dispositivo que se optó por usar fue la Raspberry pi 4 con la cámara Raspberry Pi OV5647 con el fin de ser transportada a diferentes sitios, respondía en cualquier posición que se encontraba la persona e inclusive podía variar su ángulo. Cabe destacar, que en la posición frontal el algoritmo se desempeñaba de mejor forma.

Al utilizar un computador monoplaca en este caso una Raspberry pi 4, tiene ciertas limitaciones en cuanto al procesar video, es por esto que se optó por realizarlo en tiempo real y se disminuyó el procesamiento de amplificación, esto conlleva a que el sistema sea más eficiente y responda de mejor forma a los monitoreos, aunque es importante destacar que en algunos casos el análisis de flujo óptico llega a tener ciertos errores en cuanto a identificación de contornos.

El dispositivo electrónico de monitoreo de frecuencia respiratoria no invasiva fue capaz de estimar la respiración de una persona sin la necesidad del contacto físico o de contar con una personal capacitado, debido a que cuenta con la interfaz gráfica intuitiva, donde permitió que el algoritmo fuera ejecutado de una manera sencilla, cabe destacar que la interfaz se desarrolló para ser visualizado mediante un computador.

Los circuitos usados tanto el del termistor nasal y el monitoreo no invasivo trabajan con variables físicas distintas, ya que el circuito con el termistor usando el LM35 con el Arduino mega 2560 funciona a base de la lectura de temperatura y el no invasivo funciona a base de la Raspberry pi 4 mediante el análisis de los

movimientos captados por la cámara Raspberry pi 0V5647, la precisión que se obtuvo de ambos sistemas es muy similar, por lo que también ayudó a realizar una comparación entre ambos sistemas y la verificación de la funcionalidad del algoritmo, con lo cual se obtuvo por parte del circuito con el LM35 un porcentaje de 93,64%, mientras que por parte del dispositivo electrónico de monitoreo mediante visión artificial un porcentaje de 92,6%, por lo que se determina que ambos valores cumplen con la función que se requiere.

## RECOMENDACIONES

Al realizar el proceso de medición es importante seguir las indicaciones debido a que el circuito fue desarrollado en un ambiente controlado, destacando que durante la captura del video se debe enfocar a la parte de interés (torax), cabe destacar que al aplicar el algoritmo de magnificación euleriana y el de análisis de movimiento en ciertos casos puntuales no logra obtener una diferenciación entre la respiración de la persona y el ruido que existe en un video.

No se debe presionar varias veces el botón de inicio del proceso, ya que al ser un circuito secuencial debe realizar todo el procesamiento antes de empezar uno nuevo, destacando que al hacer esta acción se estaría retrasando el procesamiento y no puede funcionar correctamente.

No es recomendado realizar todo el análisis de magnificación euleriana de video en una Raspberry pi 4 debido a que el procesamiento de todo el algoritmo es bastante robusto, por lo tanto, es recomendable adaptarlo a una amplificación mas baja y sin tanto procesamiento para que funcione en tiempo real.

Al querer comparar el circuito no invasivo con uno físico, no es recomendable hacerlo con un sensor que se encuentre cerca de la cara, sino más bien con un sensor que pueda detectar ciertos movimientos en el pecho y así proceder a realizar mediciones de frecuencia respiratoria.

## BIBLIOGRAFÍA

- [1] Garcia, I. S y V. Caranqui, «La visión artificial y sus campos de aplicación,» 2015.
- [2] XLSTAT, «XLSTAT, Your data analysis solution,» [En línea]. Available: [https://www.xlstat.com/es/soluciones/funciones/analisis-de-componentes-principales-acp#:~:text=El%20an%C3%A1lisis%20de%20componentes%20principales%20\(ACP\)%20consiste%20en%20expresar%20un,la%20variabilidad%20de%20los%20datos..](https://www.xlstat.com/es/soluciones/funciones/analisis-de-componentes-principales-acp#:~:text=El%20an%C3%A1lisis%20de%20componentes%20principales%20(ACP)%20consiste%20en%20expresar%20un,la%20variabilidad%20de%20los%20datos..) [Último acceso: 2022 06 06].
- [3] E. A. Jimenez Makin, «Identificación de estructuras mediante registros de video usando magnificación de movimiento basado en fase,» Universidad de Chile, 2023.
- [4] S. IBM, «IBM,» [En línea]. Available: <https://www.ibm.com/mx-es/topics/computer-vision>. [Último acceso: 17 09 2023].
- [5] L. Briega, Visión por computadora - Libro online de IAAR.
- [6] Anonimo, «Electronics,» [En línea]. Available: [http://www.logicelectronics.net/INGENIERIA\\_TECNOLOGIA\\_iluminacion.html](http://www.logicelectronics.net/INGENIERIA_TECNOLOGIA_iluminacion.html). [Último acceso: 12 06 2022].
- [7] Anonimo, «INGENET,» [En línea]. Available: <http://bitacora.ingenet.com.mx/2013/03/ctin-algoritmo-revela-movimientos-que-no-podemos-captar-a-simple-vista/>. [Último acceso: 24 05 2022].
- [8] N. Wadhwa, M. Rubinstein, F. Durand y W. T. Freeman, «Phase-Based Video Motion Processing,» 2013.
- [9] G. A. Parra, «Centro de investigaciones en optica,» 2012. [En línea]. Available: <https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/316/1/15585.pdf>. [Último acceso: 23 05 2022].
- [10] O. C. J. H. D. & S. D. Gómez, «Monitoreo de señales a/d con transmisión de video (estándar IEEE 802.11),» Machala, 2020.
- [11] O. W. & S. L. J. I. Gómez Morales, « Artículo Científico-Diseño e implementación de un polisomnógrafo con transmisión de datos inalámbricos.,» 2014.
- [12] J. Lillo, «Flujo óptico y conducción: características,» *revistas.ub.edu*, 1995.
- [13] L. Briega, Visión por computadora - Libro online de IAAR.

- [14] S. Alam, S. P. Singh y U. Abeyratne, «Consideraciones sobre la estimación de la frecuencia respiratoria portátil mediante un enfoque de ampliación de vídeo estabilizado,» pp. 4293-4296, 2017.
- [15] D. Alinovi, L. Cattani, L. Ferrari, G. Pisani y F. Raheli, «Spatio-temporal video processing for respiratory rate estimation,» 05 2015.
- [16] N. Wadhwa, M. Rubinstein, . F. Durand y W. T. Freeman , «Phase-Based Video Motion Processing,» 2013.
- [17] «IBM in Deutschland, Österreich und der Schweiz | IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/computer-vision>. [Último acceso: 23 09 10].
- [18] G. Gomez, *Vision computacional*, Puebla, Mexico.
- [19] «MedlinePlus,» [En línea]. Available: <https://medlineplus.gov/spanish/ency/article/002341.htm>. [Último acceso: 08 10 2023].
- [20] Y. Picon Jaimes, J. Orozco Chinome, J. Molina Franky y M. Franky Rojas, «Control central de la temperatura corporal y sus alteraciones: fiebre, hipertermia e hipotermia,» 31 03 2020. [En línea]. Available: <https://revistas.unab.edu.co/index.php/medunab/article/view/3714/3219#toc>. [Último acceso: 05 10 2023].
- [21] «Nemours KidsHealth - the Web's most visited site about children's health,» 09 2019. [En línea]. Available: [https://kidshealth.org/es/teens/lungs.html#:~:text=Los%20pulmones%20y%20el%20aparato%20respiratorio%20nos%20permiten%20respirar.,el%20nombre%20de%20\"respiraci3n\"..](https://kidshealth.org/es/teens/lungs.html#:~:text=Los%20pulmones%20y%20el%20aparato%20respiratorio%20nos%20permiten%20respirar.,el%20nombre%20de%20\) [Último acceso: 13 09 2023].
- [22] «Ferrovial,» [En línea]. Available: <https://www.ferrovial.com/es/stem/algoritmos/#:~:text=¿Qu3%20son%20los%20algoritmos%3F,seguir%20para%20alcanzar%20un%20objetivo..> [Último acceso: 15 09 2023].
- [23] «Inicio - Biblioguías y tutoriales - Guías BibUpo at Universidad Pablo de Olavide,» [En línea]. Available: [https://guiasbib.upo.es/edicion\\_multimedia\\_video/frame](https://guiasbib.upo.es/edicion_multimedia_video/frame). [Último acceso: 11 09 2023].
- [24] «Significados,» [En línea]. Available: <https://www.significados.com/microprocesador/>. [Último acceso: 18 09 2023].
- [25] «VOLTA,» [En línea]. Available: <https://volta.pe/blog/qu-es-un-ordenador-de-placa-nica-y-cu-l-me-conviene>. [Último acceso: 12 08 2023].



- [26] «MedlinePlus - Health Information from the National Library of Medicine,» [En línea]. Available: <https://medlineplus.gov/spanish/ency/article/002341.htm>. [Último acceso: 20 09 2023].
- [27] S. Penagos, L. Salazar y F. Vera, «Control de signos vitales. Guías para manejo de Urgencias.,» *Con-Ciencia Boletín Científico de la Escuela Preparatoria No. 3*, vol. 9, nº 17, pp. 1465-1473, 2005.
- [28] C. Almeida, A. Almeida y E. Forti, «EFEITOS DO MÉTODO MÃE CANGURU NOS SINAIS VITAIS DE,» *Revista brasileira de fisioterapia*, vol. 11, nº 1, pp. 1-5, 2007.
- [29] «CENIEH,» *Scientific Reports*, 01 07 2020.
- [30] «National Heart, Lung, and blood institute,» 24 03 2022. [En línea]. Available: <https://www.nhlbi.nih.gov/es/salud/corazon/actividad-fisica/beneficios#:~:text=Cuando%20se%20hace%20peri%C3%B3dicamente%2C%20la,ox%C3%ADgeno%20en%20la%20sangre%20aumentan..> [Último acceso: 18 08 2023].
- [31] M. Cretikos, R. Bellomo , K. Hillman, J. Chen , S. Finfer y A. Flabouris, «Respiratory rate: the neglected vital sign,» *PubMed*, 2008.
- [32] M. Marjaana, «Valoración de la función respiratoria,» *ELSEVIER*, vol. 8, nº 21, pp. 24-26, 2003.
- [33] J. Villegas Gonzalez, O. A. Villegas Arenas y V. Villegas Gonzalez, «Semiología de los signos vitales: Una mirada novedosa a un problema vigente,» *Redalyc.org*, vol. 12, nº 2, pp. 221-240, 2012.
- [34] W. Aguilar, G. Rodriguez, L. Alvarez y S. Sandoval, «Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing,» de *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 2017.
- [35] I. Garcia S y V. Caranqui, «La visión artificial y los campos de aplicación,» *Tierra Infinita*, vol. 1, nº 1, 2015.
- [36] B. D. Tenazaca, C. Canchignia, W. Aguilar y D. Mendoza, «Implementation of Dubin Curves-Based RRT\* Using an Aerial Image for the Determination of Obstacles and Path Planning to Avoid Them During Displacement of the Mobile Robot,» de *Developments and Advances in Defense and Security*, vol. 152, Springer, 2019, pp. 205-215.
- [37] W. G. Aguilar, V. S. Salcedo, S. D. Sandoval y B. Cobeña, «Computational Neuroscience: First Latin American Workshop, LAWCN 2017, Porto Alegre, Brazil, November 22–24, 2017, Proceedings,» de *Computational Neuroscience*, Porto Alegre, Springer International Publishing, 2017, pp. 94-105.

- [38] W. G. Aguilar, M. A. Luna, J. F. Moya y V. Abad, «Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps,» de *International Work-Conference on Artificial Neural Networks*, 2017.
- [39] W. G. Aguilar, F. J. Quisaguano, L. G. Alvarez, J. A. Pardo y Z. Proaño, «Monocular Depth Perception On a Micro-UAV Using Convolutional Neuronal Networks,» *Lecture Notes in Computer Science*, pp. 392-397, 2018.
- [40] A. de la Escalera Hueso, «Visión por computador: fundamentos y métodos.,» de *Journal of Advanced Transportation 2018*, 2018.
- [41] K. Kamble, N. Jagtap, R. Patil y A. Bhurane, «Eulerian Video Motion Magnification,» de *International Journal of Innovative Research in Computer and Communication Engineering*, 2015.
- [42] H. Yu Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand y W. T. Freeman, «Eulerian Video Magnification for Revealing Subtle Changes in the World,» *MIT*, 2012.
- [43] J. A. Chaves Barrantes, «IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA MEDICION DEL PULSO CARDIACO MEDIANTE PROCESAMIENTO DEL VIDEO EN LA TARJETA DE DESARROLLO JETSON TX1,» Tecnológico de Costa Rica, Costa Rica, 2019.
- [44] W. G. Aguilar y C. Angulo, «Real-time video stabilization without phantom movements for micro aerial vehicles,» 2014.
- [45] J. L. Jover, «Flujo óptico y conducción: características generales y velocidad percibida,» *Anuario de psicología*, nº 65, pp. 101-114, 1995.
- [46] B. KP Cuerno y B. Schunck, «Determinación del flujo óptico,» *ELSEVIER*, vol. 17, nº 1 a 3, pp. 185-203, 1981.
- [47] S. Beauchemin y J. Barrón, «El cálculo del flujo óptico.,» *Revistas ACM.*, vol. 27, nº 3, p. 433-466, 1995.
- [48] J. Sánchez, E. Meinhardt-Llopis y G. Facciolo, «TV-L1 optical flow estimation,» pp. 1-13, 2012.
- [49] J. S. Riascos Segura y J. A. Cardona Gallego, «determinación de la cinemática de objetos móviles bajo condiciones controladas mediante imágenes empleando técnicas de flujo óptico,» Universidad Tecnológica de Pereira, Pereira, 2015.
- [50] L. Calvo, «GoDaddy,» 10 03 22. [En línea]. Available: <https://es.godaddy.com/blog/que-es-raspberry-pi/>. [Último acceso: 17 08 23].
- [51] «Xataka - Tecnología y gadgets, móviles, informática, electrónica,» 24 06 2019. [En línea]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>. [Último acceso: 20 09 2023].

- [52] «Novatronic – Ingeniería, Automatización y robótica,» [En línea]. Available: <https://novatronic.com/index.php/product/raspberry-pi-4-4g-ram/>. [Último acceso: 20 09 2023].
- [53] «AV Electronics,» [En línea]. Available: <https://avelectronics.cc/producto/raspberry-pi-camera-module-vision-nocturna/>. [Último acceso: 06 10 2023].
- [54] «tiendavirtual,» [En línea]. Available: <https://tiendavirtual.com/tienda/camara-raspberry-pi-noir-ov5647-vision-noctuna-1080p-5mpx>. [Último acceso: 06 10 2023].
- [55] «ARDUINO.cl,» [En línea]. Available: <https://arduino.cl/producto/arduino-mega-2560/#:~:text=Arduino%20Mega%20es%20una%20tarjeta,implementa%20el%20lenguaje%20Processing%2FWiring..> [Último acceso: 19 09 2023].
- [56] «Proyecto Arduino,» [En línea]. Available: <https://proyectoarduino.com/arduino-mega-2560/>. [Último acceso: 20 09 2023].
- [57] «Arduino en español,» [En línea]. Available: <http://manueldelgadocrespo.blogspot.com/p/arduino-mega-2560.html>. [Último acceso: 19 09 2023].
- [58] «AV Electronics,» [En línea]. Available: <https://avelectronics.cc/producto/sensor-temperatura-lm35dz/>. [Último acceso: 04 10 2023].
- [59] J. Figueroaa, «scribd,» [En línea]. Available: <https://es.scribd.com/document/262369253/Conversor-AD-DA>. [Último acceso: 06 10 2023].
- [60] «instrumentacion,» [En línea]. Available: <https://reneom9313.wixsite.com/instrumentacion/sensor-lm35>. [Último acceso: 05 10 2023].
- [61] «Texas Instruments,» [En línea]. Available: <https://www.ti.com/product/es-mx/LM35>. [Último acceso: 13 12 2023].
- [62] «UNIT ELECTRONICS,» [En línea]. Available: <https://uelectronics.com/producto/tl084cn-amplificador-operacional-cuadruple-dip-14/>. [Último acceso: 05 10 2023].
- [63] [En línea]. Available: <https://www.alldatasheet.es/datasheet-pdf/pdf/842775/T11/TLC084CN.html>. [Último acceso: 13 12 2023].
- [64] «Electronica Plug and play,» [En línea]. Available: <https://www.electronicaplugandplay.com/circuitos-integrados/product/908-amp-op-tl084-dip14>. [Último acceso: 05 10 2023].

- [65] «Amazon Web Services, Inc.,» [En línea]. Available: <https://aws.amazon.com/es/what-is/python/>. [Último acceso: 17 08 2023].
- [66] «Logos-world - the most famous brands and company logos in the world,» [En línea]. Available: <https://logos-world.net/python-logo/>. [Último acceso: 17 08 2023].
- [67] «Programando a medianoche,» [En línea]. Available: <https://www.programandoamedianoche.com/2023/02/comparar-imagenes-con-opencv-y-net/>. [Último acceso: 19 09 2023].
- [68] DataScientest, «Formation Data Science | DataScientest.com,» [En línea]. Available: <https://datascientest.com/es/numpy-la-biblioteca-python#:~:text=%C2%BFQu%C3%A9%20es%20NumPy%3F,las%20matem%C3%A1ticas%20o%20la%20ciencia..> [Último acceso: 16 09 2023].
- [69] «Aprende con Alf,» [En línea]. Available: <https://aprendeconalf.es/docencia/python/manual/numpy/>. [Último acceso: 19 09 2023].
- [70] «Icy Science,» [En línea]. Available: <https://es.theastrologypage.com/scipy>. [Último acceso: 19 09 2023].
- [71] «Home page | Interactive Chaos,» [En línea]. Available: <https://interactivechaos.com/es/manual/tutorial-de-scipy/tutorial-de-scipy>. [Último acceso: 19 09 2023].
- [72] «DataScientest,» 25 04 2023. [En línea]. Available: <https://datascientest.com/es/todo-sobre-matplotlib>. [Último acceso: 16 08 2023].
- [73] «jvatpoint,» [En línea]. Available: <https://www.jvatpoint.com/matplotlib>. [Último acceso: 16 08 2023].
- [74] «KeepCoding Bootcamps,» [En línea]. Available: <https://keepcoding.io/blog/ques-tkinter/#:~:text=implementar%20lo%20aprendido%3F-,Qu%C3%A9%20es%20Tkinter%2C%20la%20librer%C3%ADa%20de%20Python,gr%C3%A1fica%20de%20escritorio%20con%20Python..> [Último acceso: 18 08 2023].
- [75] «Stack Overflow,» [En línea]. Available: <https://stackoverflow.com/tags/tkinter/info>. [Último acceso: 18 08 2023].
- [76] «Raspberry Pi Espana,» [En línea]. Available: <https://raspberrypi-espana.es/salida-raspberry-pi-imager/>. [Último acceso: 20 09 2023].

- [77] «Businesswire,» 10 06 2022. [En línea]. Available: <https://www.businesswire.com/news/home/20220609005214/es/>. [Último acceso: 08 10 2023].
- [78] «GetApp,» [En línea]. Available: <https://www.getapp.es/software/2035051/mobaxterm#:~:text=MobaXterm%20es%20un%20software%20de,o%20aplicaciones%20desde%20computadoras%20remotas..> [Último acceso: 11 10 2023].
- [79] «RedesZone,» [En línea]. Available: <https://www.redeszone.net/analisis/software/mobaxterm-terminal-windows/>. [Último acceso: 11 10 2023].
- [80] «Homepage - NI,» [En línea]. Available: <https://www.ni.com/es-cr/shop/product/multisim.html#:~:text=Multisim%20es%20un%20software%20est%C3%A1ndar,la%20educaci%C3%B3n%20y%20la%20investigaci%C3%B3n..> [Último acceso: 10 10 2023].
- [81] «Inicio | Edutools,» [En línea]. Available: <https://edutools.tec.mx/es/colecciones/tecnologias/multisim>. [Último acceso: 08 10 2023].
- [82] «Idealo,» [En línea]. Available: <https://www.idealo.es/precios/6628198/raspberry-pi-4-model-b.html#:~:text=59%2C90%20%E2%82%AC%20env%C3%ADo%20incl..> [Último acceso: 14 12 2023].
- [83] Anonimo, «Electronics,» [En línea]. Available: [http://www.logicelectronics.net/INGENIERIA\\_TECNOLOGIA\\_iluminacion.html](http://www.logicelectronics.net/INGENIERIA_TECNOLOGIA_iluminacion.html).
- [84] Anonimo, «INGENET,» [En línea]. Available: <http://bitacora.ingenet.com.mx/2013/03/ctin-algoritmo-revela-movimientos-que-no-podemos-captar-a-simple-vista/>.
- [85] G. A. P. Escamilla, «Centro de investigaciones en optica,» 2012. [En línea]. Available: <https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/316/1/15585.pdf>.
- [86] c. d. Wikipedia, «Wikipedia, la enciclopedia libre,» 15 10 2021. [En línea]. Available: [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_de\\_componentes\\_principales](https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales).
- [87] E. Jiménez Makin, 2020. [En línea]. Available: <https://repositorio.uchile.cl/handle/2250/175720>.
- [88] anonimo, «RECURSOS HIDRÁULICOS,» s.f. [En línea]. Available: <http://julianrojo.weebly.com/eulerian-motion-magnification.html>.

- [89] c. d. Wikipedia, «Wikipedia, la enciclopedia libre,» 19 09 2019. [En línea]. Available: [https://es.wikipedia.org/wiki/Flujo\\_%C3%B3ptico](https://es.wikipedia.org/wiki/Flujo_%C3%B3ptico).
- [90] c., «canalSALUD,» 05 05 2022. [En línea].
- [91] W. Freman, 06 2015. [En línea]. Available: <http://people.csail.mit.edu/mrub/vidmag/>.
- [92] Greneker, 1997. [En línea].
- [93] «Team Equipalia,» [En línea]. Available: <https://www.teamequipalia.es/webcam-fujikam-812h.html>. [Último acceso: 20 09 2023].
- [94] «Naylamp Mechatronics - Perú,» [En línea]. Available: <https://naylampmechatronics.com/raspberry-pi/277-camara-v2-raspberry-pi-sony-8mpx.html>. [Último acceso: 30 09 2023].
- [95] «Electrocomponentes,» [En línea]. Available: <https://www.electrocomponentes.com/tienda/raspberry/hats/raspberry-pi-3-camara-v21-1080p-8-mpix-original-element14#:~:text=La%20Cámara%20para%20Raspberry%20Pi,de%201080p30%2C%20720p60%20y%20640x480p90..> [Último acceso: 30 09 2023].
- [96] «Clearview Imaging: Machine Vision Technology Experts,» [En línea]. Available: <https://www.clearview-imaging.com/es/blog/la-importancia-de-la-iluminación-para-visión-artificial>. [Último acceso: 12 09 2023].

## ANEXOS

**Anexo 1.** Código para el dispositivo electrónico de monitoreo a través de visión artificial para medir frecuencia respiratoria no invasiva.

```
import cv2

import numpy as np

import scipy.signal as signal

import scipy.fftpack as fftpack

import time

import tkinter as tk

from picamera import PiCamera

from picamera.array import PiRGBArray

import picamera

import picamera.array

import os

# Convertir RBG a YIQ

def rgb2ntsc(src):

    [rows, cols] = src.shape[:2]

    dst = np.zeros((rows, cols, 3), dtype=np.float64)

    T = np.array([[0.114, 0.587, 0.298], [-0.321, -0.275, 0.596], [0.311, -0.528, 0.212]])

    for i in range(rows):

        for j in range(cols):

            dst[i, j] = np.dot(T, src[i, j])

    return dst

# Convertir YIQ a RBG

def ntsc2rbg(src):

    [rows, cols] = src.shape[:2]

    dst = np.zeros((rows, cols, 3), dtype=np.float64)

    T = np.array([[1, -1.108, 1.705], [1, -0.272, -0.647], [1, 0.956, 0.620]])
```

```

for i in range(rows):
    for j in range(cols):
        dst[i, j] = np.dot(T, src[i, j])
return dst

# Construir la pirámide gaussiana
def build_gaussian_pyramid(src, level=3):
    s = src.copy()
    pyramid = [s]
    for i in range(level):
        s = cv2.pyrDown(s)
        pyramid.append(s)
    return pyramid

# Construir la pirámide laplaciana
def build_laplacian_pyramid(src, levels=3):
    gaussianPyramid = build_gaussian_pyramid(src, levels)
    pyramid = []
    for i in range(levels, 0, -1):
        GE = cv2.pyrUp(gaussianPyramid[i])
        L = cv2.subtract(gaussianPyramid[i - 1], GE)
        pyramid.append(L)
    return pyramid

def load_video(video_filename, max_frames=None, target_resolution=None):
    cap = cv2.VideoCapture(video_filename)
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    width, height = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = int(cap.get(cv2.CAP_PROP_FPS))

```



```

video_tensor = np.zeros((frame_count, height, width, 3), dtype='uint8')
x = 0
while cap.isOpened():
    ret, frame = cap.read()
    if ret is True:
        video_tensor[x] = frame
        x += 1
    else:
        break
return video_tensor, fps

```

# Aplicar filtro ideal temporal de banda a video gaussiano

```

def temporal_ideal_filter(tensor, low, high, fps, axis=0):
    fft = fftpack.fft(tensor, axis=axis)
    frequencies = fftpack.fftfreq(tensor.shape[0], d=1.0 / fps)
    bound_low = (np.abs(frequencies - low)).argmin()
    bound_high = (np.abs(frequencies - high)).argmin()
    fft[:bound_low] = 0
    fft[bound_high:-bound_high] = 0
    fft[-bound_low:] = 0
    iff = fftpack.ifft(fft, axis=axis)
    return np.abs(iff)

```

# Construir la pirámide gaussiana para video

```

def gaussian_video(video_tensor, levels=3):
    for i in range(0, video_tensor.shape[0]):
        frame = video_tensor[i]

```

```

pyr = build_gaussian_pyramid(frame, level=levels)
gaussian_frame = pyr[-1]
if i == 0:
    vid_data = np.zeros((video_tensor.shape[0], gaussian_frame.shape[0],
gaussian_frame.shape[1], 3))
    vid_data[i] = gaussian_frame
return vid_data

```

# Amplificar el video

```

def amplify_video(gaussian_vid, amplification=50):
    return gaussian_vid * amplification

```

# Reconstruir video a partir del video original y el video gaussiano

```

def reconstruct_video(amp_video, origin_video, levels=3):
    final_video = np.zeros(origin_video.shape)
    for i in range(0, amp_video.shape[0]):
        img = amp_video[i]
        for x in range(levels):
            img = cv2.pyrUp(img)
        img = img + origin_video[i]
        final_video[i] = img
    return final_video

```

# Guardar video en archivos

```

def save_video(video_tensor):
    fourcc = cv2.VideoWriter_fourcc('M', 'J', 'P', 'G')
    [height, width] = video_tensor[0].shape[0:2]
    writer = cv2.VideoWriter("out.avi", fourcc, 30, (width, height), 1)
    for i in range(0, video_tensor.shape[0]):
        writer.write(cv2.convertScaleAbs(video_tensor[i]))

```

```

writer.release()

# Construir la pirámide laplaciana para video
def laplacian_video(video_tensor, levels=3):
    tensor_list = []
    for i in range(0, video_tensor.shape[0]):
        frame = video_tensor[i]
        pyr = build_laplacian_pyramid(frame, levels=levels)
        if i == 0:
            for k in range(levels):
                tensor_list.append(np.zeros((video_tensor.shape[0], pyr[k].shape[0],
pyr[k].shape[1], 3)))
            for n in range(levels):
                tensor_list[n][i] = pyr[n]
    return tensor_list

# Filtro de banda Butterworth
def butter_bandpass_filter(data, lowcut, highcut, fs, order=5):
    omega = 0.5 * fs
    low = lowcut / omega
    high = highcut / omega
    b, a = signal.butter(order, [low, high], btype='band')
    y = signal.lfilter(b, a, data, axis=0)
    return y

# Reconstruir video a partir de la pirámide de tensores
def reconstruct_from_tensorlist(filter_tensor_list, levels=3):
    final = np.zeros(filter_tensor_list[-1].shape)
    for i in range(filter_tensor_list[0].shape[0]):
        up = filter_tensor_list[0][i]

```

```

    for n in range(levels - 1):
        up = cv2.pyrUp(up) + filter_tensor_list[n + 1][i]
    final[i] = up
return final

# Magnificar movimiento
def magnify_motion(video_name,low,high,levels=3,amplification=15):
    t,f=load_video(video_name)
    lap_video_list=laplacian_video(t,levels=levels)
    filter_tensor_list=[]
    for i in range(levels):
        filter_tensor=butter_bandpass_filter(lap_video_list[i],low,high,f)
        filter_tensor*=amplification
        filter_tensor_list.append(filter_tensor)
    recon=reconstruct_from_tensorlist(filter_tensor_list)
    final=t+recon
    save_video(final)

def capture_and_process_video(duracion_segundos=4, resolucion=(1280, 720),
framerate=60):
    camera = PiCamera()
    camera.resolution = resolucion
    camera.framerate = framerate
    raw_capture = PiRGBArray(camera, size=resolucion)

    real_time_video_filename = "real_time_video.h264"

    video_writer = cv2.VideoWriter(real_time_video_filename,
cv2.VideoWriter_fourcc(*'H264'), framerate, resolucion)

```

```
start_time = time.time()
```

```
for frame in camera.capture_continuous(raw_capture, format="bgr",  
use_video_port=True):
```

```
    image = frame.array
```

```
    cv2.imshow("Real-Time Video", image)
```

```
    # Guardar cada frame en el archivo de video
```

```
    video_writer.write(image)
```

```
raw_capture.truncate(0)
```

```
if time.time() - start_time >= duracion_segundos:
```

```
    break
```

```
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
    break
```

```
cv2.destroyAllWindows()
```

```
# Liberar el objeto cv2.VideoWriter al finalizar la captura
```

```
video_writer.release()
```

```
# Convertir el video a formato AVI usando OpenCV
```

```
avi_filename = "real_time_video.avi"
```

```
cap_h264 = cv2.VideoCapture(real_time_video_filename)
```

```
fourcc = cv2.VideoWriter_fourcc(*'XVID')
```

```
out_avi = cv2.VideoWriter(avi_filename, fourcc, framerate, resolucion)
```

```

while True:
    ret, frame = cap_h264.read()
    if not ret:
        break
    out_avi.write(frame)

cap_h264.release()
out_avi.release()

# Luego de la conversión, realizar el procesamiento
# Agrega aquí tu código original para cargar y procesar el video, por ejemplo:
load_video(avi_filename)
magnify_motion(avi_filename,0.4, 3)

# Crear la ventana de la interfaz gráfica
root = tk.Tk()
root.title("Interfaz de circuito monitor de frecuencia respiratoria no invasiva")
# Etiqueta para las funciones
funciones_label = tk.Label(root, text="FUNCIONES", font=("Helvetica", 12,
"bold"))
funciones_label.pack(side="top", anchor="w", padx=10, pady=(10, 5))

# Botón para capturar el video en tiempo real y procesarlo
capture_button = tk.Button(root, text="Captura de video y procesamiento",
command=capture_and_process_video)
capture_button.pack(side="top", anchor="w", padx=10, pady=10)

# Contenedor para organizar los elementos
container = tk.Frame(root)
container.pack(side="left", anchor="sw", padx=10, pady=10)

```

```

# Etiqueta para las indicaciones
indicaciones_label = tk.Label(container, text="INDICACIONES",
font=("Helvetica", 12, "bold"))
indicaciones_label.pack(side="top", pady=(0, 5))

# Etiqueta para las instrucciones
instructions_label = tk.Label(container, text="*La persona debe mantenerse en
estado estatico durante 1 minuto para la captura del video.\n"
"*Para iniciar el programa, se debe presionar el botón
Captura de video y procesamiento.\n"
"*El procesamiento del video puede tardar de 6 a 7
minutos.", font=("Helvetica", 10))
instructions_label.pack(side="top")
#)

# Etiqueta para las instrucciones
#instructions_label = tk.Label(root, text="Para iniciar el programa, presiona el
botón 'Captura de video y procesa'.")
#instructions_label.pack()

# Botón para capturar el video en tiempo real y procesarlo
#capture_button = tk.Button(root, text="Captura de video y procesamiento",
command=capture_and_process_video)
#capture_button.pack()
root.mainloop()

```

**Para el flujo óptico, se tiene el siguiente código:**

```

import numpy as np
import cv2

```

```

cap = cv2.VideoCapture('out.avi')

feature_params = dict(maxCorners=100,
                      qualityLevel=0.3,
                      minDistance=7,
                      blockSize=7)

lk_params = dict(winSize=(15, 15),
                 maxLevel=2,
                 criteria=(cv2.TERM_CRITERIA_EPS
cv2.TERM_CRITERIA_COUNT, 10, 0.03))

color = np.random.randint(0, 255, (100, 3))

ret, old_frame = cap.read()
old_gray = cv2.cvtColor(old_frame, cv2.COLOR_BGR2GRAY)
frame_gray = cv2.cvtColor(old_frame, cv2.COLOR_BGR2GRAY)

highpass_kernel = np.array([[ -1, -1, -1],
                             [-1,  8, -1],
                             [-1, -1, -1]])

frame_filtered = cv2.filter2D(frame_gray, -1, highpass_kernel)

contornos, _ = cv2.findContours(frame_filtered, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

if contornos:
    mayor_contorno = max(contornos, key=cv2.contourArea)
    x, y, w, h = cv2.boundingRect(mayor_contorno)
    roi = old_gray[y:y + h, x:x + w]
    p0 = cv2.goodFeaturesToTrack(roi, mask=None, **feature_params)
    p0 = p0.reshape(-1, 1, 2) + np.array([x, y])

```



```

else:
    x, y, w, h = cv2.selectROI(old_frame)
    roi = old_gray[y:y + h, x:x + w]
    p0 = cv2.goodFeaturesToTrack(roi, mask=None, **feature_params)
    p0 = p0.reshape(-1, 1, 2) + np.array([x, y])

p0 = p0.astype(np.float32).reshape(-1, 1, 2)
mask = np.zeros_like(old_frame)

inhalaciones = 0
exhalaciones = 0

historial_cambios = []

umbral_movimiento_pequeno = 0.25

esperando_nuevo_cambio = False

cont_inhalacion = 0
cont_exhalacion = 0

frames_consolidacion = 20

total_movimientos_inhalacion = 0

while True:
    ret, frame = cap.read()
    if not ret:
        break

    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    frame_filtered = cv2.filter2D(frame_gray, -1, highpass_kernel)

```

```

p1, st, err = cv2.calcOpticalFlowPyrLK(old_gray, frame_filtered, p0, None,
**lk_params)
good_new = p1[st == 1]
good_old = p0[st == 1]

for i, (new, old) in enumerate(zip(good_new, good_old)):
    a, b = new.ravel()
    c, d = old.ravel()
    mask = cv2.line(mask, (int(a), int(b)), (int(c), int(d)), (0, 255, 0), 2)
    frame = cv2.circle(frame, (int(a), int(b)), 5, (0, 255, 0), -1)

img = cv2.add(frame, mask)

average_magnitude = np.mean(np.linalg.norm(good_new - good_old, axis=1))

umbral_inhalacion = 2.75
umbral_exhalacion = -2.75

cambios_inhalacion = 0
cambios_exhalacion = 0
for i in range(len(good_new)):
    delta_y = good_new[i, 1] - good_old[i, 1]

    if np.abs(delta_y) < umbral_movimiento_pequeno:
        continue

    if delta_y > umbral_inhalacion:
        cambios_inhalacion += 1
    elif delta_y < umbral_exhalacion:
        cambios_exhalacion += 1

```

```

if esperando_nuevo_cambio:
    if cambios_inhalacion > 0 or cambios_exhalacion > 0:
        esperando_nuevo_cambio = False
    else:
        historial_cambios.append((cambios_inhalacion, cambios_exhalacion))
        promedio_inhalacion = np.mean([c[0] for c in historial_cambios])
        promedio_exhalacion = np.mean([c[1] for c in historial_cambios])

        inhalaciones += int(promedio_inhalacion)
        exhalaciones += int(promedio_exhalacion)

        esperando_nuevo_cambio = True

if promedio_inhalacion > promedio_exhalacion:
    cont_inhalacion += 1
    cont_exhalacion = 0
else:
    cont_exhalacion += 1
    cont_inhalacion = 0

if cont_inhalacion >= frames_consolidacion:
    inhalaciones_detectadas = cont_inhalacion // frames_consolidacion
    total_movimientos_inhalacion += inhalaciones_detectadas
    print(f"¡Movimiento de inhalación detectado! Número de inhalaciones:
{inhalaciones_detectadas * 20}")
    cont_inhalacion = 0
elif cont_exhalacion >= frames_consolidacion:
    print("Movimiento de exhalación detectado")
    cont_exhalacion = 0

cv2.imshow('frame', img)
k = cv2.waitKey(30) & 0xff

```

```

if k == 27:
    break

old_gray = frame_filtered.copy()
p0 = good_new.reshape(-1, 1, 2)

print(f"Total de movimientos de inhalación detectados:
{total_movimientos_inhalacion}")

cv2.destroyAllWindows()
cap.release()

```

**Anexo 2.** Código para el circuito monitor de frecuencia respiratorio con Arduino 2560.

```

float tempC;
int pinLM35 = A0;
int respiraciones = 0;
unsigned long tiempoInicio;
unsigned long tiempoUltimaInhalacion;
const int intervaloMedicion = 30000;
const int tiempoEsperaInhalacion = 3000;

void setup() {
    Serial.begin(9600);
    tiempoInicio = millis();
    tiempoUltimaInhalacion = millis();
}

void loop() {
    tempC = analogRead(pinLM35);
    tempC = (5.0 * tempC * 100.0) / 1024.0;

```

```
if (tempC < 50.0) {  
  if (millis() - tiempoUltimaInhalacion >= tiempoEsperaInhalacion) {  
    respiraciones++;  
    tiempoUltimaInhalacion = millis();  
  }  
}
```

```
Serial.print("Temperatura: ");  
Serial.print(tempC);  
Serial.print(" | Respiraciones acumuladas: ");  
Serial.println(respiraciones);
```

```
if (millis() - tiempoInicio >= intervaloMedicion) {  
  Serial.print("Respiraciones en los últimos 60 segundos: ");  
  Serial.println(respiraciones*2);  
  respiraciones = 0;  
  tiempoInicio = millis();  
}
```

```
delay(1000);  
}
```