

**Sistema de control a distancia para personas con discapacidad auditiva y del habla  
mediante el procesamiento digital de imágenes**

Sergio Andrés Gaitán Hernández

Asesor

Diego Fernando Nava Cueva

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Ingeniería electrónica

2023

## **Dedicatoria**

A mi familia, ellos hicieron posible que mi formación como ingeniero electrónico fuese posible, su apoyo y su esmero por convertirme en una persona con valores y tener un gran proyecto de vida.

### **Agradecimientos**

A mi madre que me brindo la oportunidad de estudiar esta gran profesión y apoyarme en cada oportunidad que tuvo. A mi familia por todo el apoyo que me dieron, por ayudarme a dar un paso más en mi formación personal y profesional, por último, al ingeniero Diego Fernando Nava, el cual fue el encargado de que la idea tuviera forma, por compartir sus conocimientos, por su tiempo y ayudar a que el proyecto fuese estructurado de la mejor manera.

## Resumen

Durante este proyecto se buscó la manera de mejorar la accesibilidad para personas con discapacidades en el hogar, con el enfoque en personas que presentan discapacidades auditivas y del habla, se buscó el desarrollo de un prototipo que mejore su calidad de vida donde se abordó la selección e identificación de señas, el diseño, la creación e implementación física del prototipo mediante el uso de tecnologías actuales, con ayuda del procesamiento digital de imágenes e inteligencia artificial utilizando el lenguaje de programación de Python y algunas de sus librerías como Computer Vision, TensorFlow, en la creación del sistema de procesamiento y clasificación con ayuda de Teachable Machine, así como la captura de datos para el entrenamiento del modelo. Para el entrenamiento del modelo, se usó Teachable Machine el cual se entrenó con 4 señas del lenguaje seleccionado, las señas se pasarán al programa de procesamiento y dependiendo de la seña que se reconozca en la tarjeta de desarrollo ESP32 CAM, se activara o desactivara un dispositivo eléctrico, se empleara una bombilla como ejemplo y debido a limitaciones de la tarjeta se realizara solo para 2 dispositivos, ya que, se pueden controlar hasta 10 dispositivos. El prototipo se construye con el propósito de buscar una alternativa que mejore la calidad de vida en personas con estas discapacidades mediante la aplicación de un sistema inteligente que permite el control de dispositivos a distancia en el hogar mediante señas, tal como Alexa o Google Home (con limitaciones), pero con la idea de aplicar el procesamiento digital de imágenes y no de voz. Por último, se analizan posibles mejoras del prototipo con ayuda de las tarjetas de Raspberry Pi.

**Palabras claves:** Procesamiento digital de imágenes, Discapacidad auditiva, Discapacidad del habla, Lenguaje de señas, Prototipo.

### **Abstract**

During this project, a way to improve accessibility for people with disabilities at home was sought, with the focus on people with hearing and speech disabilities, the development of a prototype was sought to improve their quality of life where it was addressed the selection and identification of signs, the design, the creation and physical implementation of the prototype through the use of current technologies, with the help of digital image processing and artificial intelligence using the Python programming language and some of its libraries such as Computer Vision, TensorFlow, in the creation of the processing and classification system with the help of the Teachable Machine, as well as the data capture for model training. For the training of the model, a Teachable Machine was used, which was trained with 4 signs of the selected language, the signs will be passed to the processing program and depending on the sign that is recognized in the ESP32 CAM development card, a signal will be activated or deactivated an electrical device, a light bulb will be used as an example and due to limitations of the card it will be carried out only for 2 devices, since up to 10 devices can be controlled. The prototype is built with the purpose of finding an alternative that improves the quality of life in people with these disabilities through the application of an intelligent system that allows remote control of devices in the home through signs, such as Alexa or Google Home (with limitations), but with the idea of applying digital image processing and not voice processing. Finally, possible improvements to the prototype are analyzed with the help of Raspberry Pi cards.

**Keywords:** Digital image processing, Hearing disability, Speech disability, Sign language, Prototype.

**Tabla de Contenido**

Introducción .....	14
Planteamiento del Problema .....	16
Árbol Causa – Efecto del Problema.....	18
Justificación .....	19
Objetivos .....	21
Objetivo General.....	21
Objetivos Específicos.....	21
Marco Referencial.....	22
Marco Conceptual:.....	22
Estado del Arte:.....	36
Metodología .....	40
Etapa de Exploración: .....	41
Etapa de Software: .....	41
Etapa de Hardware:.....	47
Protocolo de Comunicación:.....	49
Implementación Física del Prototipo .....	50
Diseño de la Solución .....	51
Selección del Lenguaje y Señas a Utilizar.....	51
Etapa de Programación .....	53
Captura de Datos y Creación de la Base de Datos.....	54
Programación del Microcontrolador .....	69

	7
Etapa de Comunicación .....	72
Etapa de Control .....	73
Planos .....	75
Implementación Física .....	83
Algoritmos .....	83
Python Código .....	83
Arduino IDE (esp32-cam): .....	83
Explicación General.....	84
Construcción del Algoritmo.....	84
Construcción del Circuito .....	86
Resultados.....	91
Errores.....	94
Circuito en Funcionamiento.....	95
Funcionamiento del Relé .....	96
Funcionamiento del TRIAC.....	101
Plan de Mejoras.....	104
Plan de Mejoras a Largo Plazo .....	104
Plan de Mantenimiento .....	105
Plan de Mantenimiento Predictivo.....	105
Plan de Mantenimiento Preventivo.....	106
Plan de Mantenimiento Correctivo .....	106
Conclusiones .....	108
Referencias.....	110

## Lista de Figuras

<b>Figura 1</b> <i>Arbol del problema</i> .....	18
<b>Figura 2</b> <i>Inteligencia artificial</i> .....	24
<b>Figura 3</b> <i>Red neuronal artificial</i> .....	25
<b>Figura 4</b> <i>ESP32-AI Thinker</i> .....	27
<b>Figura 5</b> <i>ESP32-CAM pines en uso</i> .....	28
<b>Figura 6</b> <i>ESP32-CAM tabla de pines en uso por la cámara</i> .....	28
<b>Figura 7</b> <i>ESP32-CAM pines usados para la tarjeta SD</i> .....	29
<b>Figura 8</b> <i>Diagrama y composición del tiristor</i> .....	29
<b>Figura 9</b> <i>Encapsulado del tiristor</i> .....	30
<b>Figura 10</b> <i>Raspberry pi 4</i> .....	32
<b>Figura 11</b> <i>Topología MQTT bróker</i> .....	33
<b>Figura 12</b> <i>Conexión general del sistema</i> .....	42
<b>Figura 13</b> <i>Matriz de una imagen</i> .....	43
<b>Figura 14</b> <i>Idea de seguimiento de la mano</i> .....	44
<b>Figura 15</b> <i>Seguimiento de la mano en vertical</i> .....	45
<b>Figura 16</b> <i>Seguimiento de la mano en horizontal</i> .....	45
<b>Figura 17</b> <i>Camara OV2640 de la ESP32-CAM de 66° y 120°</i> .....	48
<b>Figura 18</b> <i>Diagrama de la conexión en el sistema de control</i> .....	49
<b>Figura 19</b> <i>Esquema simple del sistema de comunicación</i> .....	50
<b>Figura 20</b> <i>Letras del lenguaje de señas colombiano</i> .....	51
<b>Figura 21</b> <i>Letra A del lenguaje de señas colombiano</i> .....	52
<b>Figura 22</b> <i>Letra B del lenguaje de señas colombiano</i> .....	52



	9
<b>Figura 23</b> <i>Letra Ab de la lengua de señas de Colombia</i> .....	53
<b>Figura 24</b> <i>Letra H del lenguaje de señas colombiano</i> .....	53
<b>Figura 25</b> <i>Idea de seguimiento de mano</i> .....	54
<b>Figura 26</b> <i>Captura de datos</i> .....	55
<b>Figura 27</b> <i>Captura de datos de la seña A</i> .....	56
<b>Figura 28</b> <i>Captura de datos de la seña Ab</i> .....	57
<b>Figura 29</b> <i>Captura de datos de la seña B</i> .....	57
<b>Figura 30</b> <i>Captura de datos de la seña H</i> .....	58
<b>Figura 31</b> <i>Maquina de aprendizaje creada por Google</i> .....	59
<b>Figura 32</b> <i>Red neuronal convolucional</i> .....	60
<b>Figura 33</b> <i>Convolución</i> .....	61
<b>Figura 34</b> <i>Filtro Lanczos</i> .....	61
<b>Figura 35</b> <i>Max Pooling y Average Pooling</i> .....	62
<b>Figura 36</b> <i>Selección del proyecto de imágenes</i> .....	63
<b>Figura 37</b> <i>Selección del modelo de imágenes estándar</i> .....	64
<b>Figura 38</b> <i>Selección de clases e inicio de entramiento del modelo</i> .....	65
<b>Figura 39</b> <i>Parámetros avanzados del entrenamiento</i> .....	66
<b>Figura 40</b> <i>Teachable Machine modelo entrenado</i> .....	67
<b>Figura 41</b> <i>Texto representante de la seña</i> .....	68
<b>Figura 42</b> <i>Conexión de cámara en Raspberry pi</i> .....	70
<b>Figura 43</b> <i>Raspberry pi 4 pines</i> .....	71
<b>Figura 44</b> <i>Sistema de comunicación</i> .....	72
<b>Figura 45</b> <i>Esquema ESP32-CAM</i> .....	73

	10
<b>Figura 46</b> <i>ESP32-CAM y programador</i> .....	74
<b>Figura 47</b> <i>Relé</i> .....	75
<b>Figura 48</b> <i>Esquema de conexión relé</i> .....	75
<b>Figura 49</b> <i>Diagrama eléctrico de relé de 1 canal</i> .....	76
<b>Figura 50</b> <i>Conexión MOC3021</i> .....	77
<b>Figura 51</b> <i>Diagrama de conexión MOC3021</i> .....	78
<b>Figura 52</b> <i>Cálculo de resistencia</i> .....	79
<b>Figura 53</b> <i>Datasheet MOC3021 valores de Led 1</i> .....	79
<b>Figura 54</b> <i>Datasheet MOC3021 valores de Led 2</i> .....	80
<b>Figura 55</b> <i>Diagrama de conexión general</i> .....	81
<b>Figura 56</b> <i>Datasheet BTA 41</i> .....	82
<b>Figura 57</b> <i>Diagrama de flujo etapa de potencia</i> .....	83
<b>Figura 58</b> <i>Circuito implementación física</i> .....	86
<b>Figura 59</b> <i>Esquema conexión de bombilla</i> .....	87
<b>Figura 60</b> <i>Módulo Relé de 1 canal</i> .....	87
<b>Figura 61</b> <i>Módulo de control mediante TRIAC</i> .....	88
<b>Figura 62</b> <i>Esquema de conexión</i> .....	89
<b>Figura 63</b> <i>Conexión de la ESP32-CAM</i> .....	90
<b>Figura 64</b> <i>Reconocimiento de la seña "A"</i> .....	91
<b>Figura 65</b> <i>Reconocimiento de la seña "Ab"</i> .....	92
<b>Figura 66</b> <i>Reconocimiento de la seña "B"</i> .....	92
<b>Figura 67</b> <i>Reconocimiento de la seña "H"</i> .....	93
<b>Figura 68</b> <i>Error en el cambio de la letra B a H</i> .....	94

	11
<b>Figura 69</b> <i>Error en paso de seña Ab a A</i> .....	95
<b>Figura 70</b> <i>Activación del relé</i> .....	96
<b>Figura 71</b> <i>Acercamiento al relé</i> .....	97
<b>Figura 72</b> <i>Activación de bombilla por relé</i> .....	98
<b>Figura 73</b> <i>Seña de desactivación del relé</i> .....	99
<b>Figura 74</b> <i>Acercamiento a la desactivación del relé</i> .....	100
<b>Figura 75</b> <i>Desactivación de la bombilla por relé</i> .....	101
<b>Figura 76</b> <i>Activación de bombilla mediante TRIAC</i> .....	102
<b>Figura 77</b> <i>Desactivación de bombilla mediante TRIAC</i> .....	103

**Lista de Tablas**

<b>Tabla 1</b> <i>Secuencia lógica de control</i> .....	47
<b>Tabla 2</b> <i>Secuencia lógica de envío de datos por MQTT</i> .....	69

**Lista de Apéndices**

<b>Apéndice A</b> <i>Algoritmo diseñado en Python</i> .....	115
<b>Apéndice B</b> <i>Algoritmo diseñado en Arduino IDE</i> .....	121

## Introducción

Los sistemas de accesibilidad inteligentes han ido evolucionando a lo largo de los años, se han centrado en diferentes ámbitos y permitido mejorar diferentes aspectos de la vida humana. Estos sistemas permiten mejorar la calidad de vida de las personas, mejoran la ergonomía y facilitan hacer cosas tan simples como encender una bombilla. Estos sistemas han evolucionado y se han aplicado a personas con dificultades, estas dificultades pueden ser dadas de manera natural o provocadas por un agente externo. Con ayuda de los sistemas inteligentes, se han logrado incluir estas mejoras en la ergonomía a este tipo de personas, sus facilidades son mayores con ayuda de tecnologías actuales, algunas de estas tecnologías se pueden observar en el artículo de Hernández, C., Pulido, J. L., y Arias, J. E. (2015), donde se dan ejemplos de diferentes usos de tecnologías tales como el ingeniero colombiano Jorge Enrique Leal Rodríguez, quien desarrollo un traductor de lenguas de señas en línea, de acceso libre. Las personas con dificultades auditivas o del habla son cada vez más frecuentes, ya sea por la propagación de ruido debido al aumento de maquinaria o de autos, así como de la urbanización, se espera que, según estudios el número de personas con estas discapacidades aumente como lo indica Muñiz, A., "...estudios realizados en Canadá, Estados Unidos y Europa, informan de que la audición se empieza a perder 20 años antes que en el pasado, por lo que cada vez es más frecuente la pérdida de audición en gente joven" (Muñiz, A., 2020). Basándose en estudios publicados en un artículo de Chávez, M. et al. (s.f). Aquí es donde se tiene la importancia de aplicar investigaciones en este ámbito, será más frecuente y, por tanto, será necesaria toda la ayuda que se pueda realizar, de esta manera por medio de aportes en cuanto a investigación se espera aportar y conseguir datos de relevancia, así como formas de abordar esta problemática mediante una solución particular aplicada a la domótica de una residencia familiar.

Este trabajo se centra en buscar una solución mediante el uso de tecnologías actuales tales como el procesamiento digital de imágenes, el uso de inteligencias artificiales y protocolos actuales de comunicación, se busca construir un prototipo que mejore la calidad de vida de las personas con dificultades auditivas y del habla de una residencia familiar. Así como impulsar futuras investigaciones en la Universidad Nacional Abierta y a Distancia.

## Planteamiento del Problema

Los problemas de accesibilidad tecnológica en el hogar producen situaciones en las que personas con algún tipo de discapacidad en las que una persona con problemas de audición no pueda escuchar un timbre cuando llaman, quieran tener control sobre su iluminación con un dispositivo como Alexa o Google Home en su hogar y les facilite sus tareas cotidianas, pero lo cierto es que en muchos casos estos dispositivos no los pueden usar pues requieren de habilidades perdidas o que nunca tuvieron, existen diferentes problemas de este tipo, pero en general son causados por los problemas en sus cuerpos, ya sean físicos o mentales. Si analizamos el problema de las discapacidades según la OMS, la Organización Mundial de la Salud en un artículo de portafolio.co:

Se calcula que 446 millones de personas, es decir más de 5% de la población mundial tienen una pérdida auditiva incapacitante. De ese total, 500.000 son colombianas, es decir, el uno por ciento de su población, que se estima en 48'000.00"0. (Portafolio, s. f.).

La accesibilidad tecnológica en el hogar para personas con estas discapacidades es reducida, los dispositivos que mejoran la ergonomía en el hogar conocidos no cuentan con el lenguaje que estas personas usan, con el lenguaje de señas por ello se busca brindar nuevos aportes que permitan que su estancia en el hogar sea mucho más cómoda. ¿Quizá mediante el uso de la tecnología? En un artículo donde se investigaba las investigaciones que se les han dado para este tipo de personas dice que:

En la actualidad, las investigaciones son centradas en el diseño, creación y validación de tecnologías de asistencia para personas con problemas auditivos, donde se destacan dispositivos de comunicación (teléfonos, pantallas, tabletas, entre otros), herramientas para el aprendizaje de lenguaje de señas, mecanismos de tutoría electrónica, útiles para el reconocimiento de voz, interpretación, traducción y gamificación; además de la generación de entornos de asistencia virtual (usando guante electrónico), esquemas de



traducción automática, avances tecnológicos basados en lengua de señas, además de la inclusión de robots y dispositivos portátiles inteligentes, apoyados de sensores, que permitan mayor accesibilidad y mejores experiencias de los usuarios. (Patiño-Toro, O. N., Patiño-Vanegas et al , 2020).

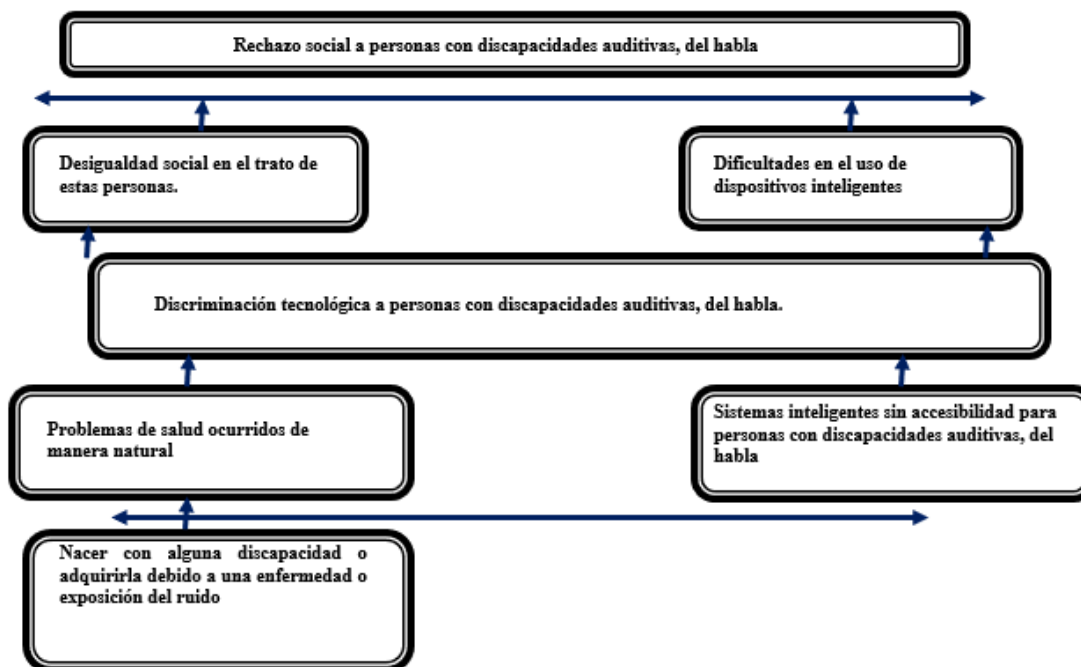
Esto permite dar una idea de cómo mediante la tecnología se puede abordar la problemática planteada.

¿Es posible mediante el diseño de un prototipo mejorar la calidad de vida en el hogar a personas con discapacidad auditiva y del habla?

## Árbol Causa – Efecto del Problema

Figura 1

Árbol del problema



Fuente. Elaboración propia.

## Justificación

Las personas con discapacidades auditivas son cada vez más frecuentes, ya sea por exposiciones constantes a ruidos o problemas de nacimiento, problemas de salud.

La Organización Mundial de la Salud (OMS) reporta que 60% de los casos de pérdida de audición infantil se debe a causas prevenibles y estima que en la actualidad 1 100 millones de jóvenes de entre 12 y 35 años de edad pueden afectar su capacidad auditiva al exponerse a ruido recreativo <sup>2</sup> el cual es considerado como la exposición a ruido excesivo en discotecas, bares, conciertos, gimnasios, eventos deportivos, salas de videojuegos, cines, así como por estudiar música, ser músico profesional y usar reproductores personales de música (RPM) con audífonos o altavoces ubicados en espacios estrechos y a volumen elevado. (Gutiérrez-Farfán et al., 2018).

Es decir, a medida que se han ido creando nuevos medios de recreación también han aumentado los riesgos a padecer una anomalía en el oído, es posible que de manera natural también se vayan perdiendo estas capacidades a medida que las personas envejecen, es importante mantener las investigaciones de estos temas de manera activa y que no se pierda el interés por buscar alternativas que mejoren la calidad de vida de estas personas, que mejoren el bienestar de la sociedad y permitan la integración de las personas con discapacidades. En este caso en particular es un enfoque dado desde la accesibilidad para mejorar la ergonomía de las personas que tienen problemas de audición, así como aquellas con problemas del habla que se tratan a la vez debido a la relación existente entre estos, como indica Rossi, M., y Labella T

Las personas aprendemos a hablar por imitación del lenguaje de nuestros semejantes. Si por lo tanto un niño tiene un déficit auditivo que le impide oír el mensaje hablado con todos sus matices, la imitación será imperfecta, es decir dislálida (Rossi, M., y Labella T., s.f).

En un artículo donde explica las diferentes formas en las que la pérdida de audición puede llevar a la pérdida parcial o completa de la habilidad de hablar por medio de diferentes

relaciones, por ello, se busca realizar mediante la electrónica, realizar mejoras en el control de sistemas inteligentes, mejoras en la accesibilidad tecnológica que englobe de manera general a este tipo de personas y permita mejoras en la estancia de su hogar, con el objetivo de integrar nuevas tecnologías para aquellas personas con estos problemas y darles una solución particular desde la tecnología.

## Objetivos

### Objetivo General

Diseñar un sistema de control a distancia para personas con discapacidad auditiva y del habla mediante el procesamiento digital de imágenes, con el fin de mejorar su accesibilidad y usabilidad en el control de dispositivos y sistemas en el hogar.

### Objetivos Específicos

Identificar el lenguaje de señas más adecuado para las personas con discapacidad auditiva y del habla, y seleccionar cuatro letras que entren dentro del alfabeto para ser utilizadas en el sistema de control a distancia.

Diseñar un algoritmo de procesamiento de imágenes que permita la captura de datos a través de una cámara web, y que realice el seguimiento de la mano de manera precisa y en tiempo real.

Definir un sistema de comunicación maquina a máquina que permita la activación y desactivación de la etapa de potencia del sistema de control a distancia en función de los gestos de la mano detectados.

Elaborar el sistema de control electrónico que permita el control de los actuadores necesarios para el control a distancia de los dispositivos y sistemas, acoplando la lógica del sistema de comunicación.

Implementar un prototipo físico del sistema de control a distancia diseñado, y realizar pruebas de su usabilidad para las personas con discapacidad auditiva y del habla.

## Marco Referencial

### Marco Conceptual

#### *Python*

“Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo.” (*Becas Santander, 2023*). Python es un lenguaje de programación, es decir, se trata de una manera en la que un usuario puede escribir para que se lleven a cabo ciertas instrucciones que se definen como algoritmo permitiendo diseñar algoritmos capaces de prácticamente todo en informática.

#### *OpenCv*

“Una biblioteca de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos. Además, está especializada en el sistema de visión artificial y machine learning.” (R. Hugo, 2021). Se trata de una biblioteca que permite comprimir muchos procesos desarrollados durante un código haciendo uso de funciones, es decir que mediante esta librería se utilizan comandos que engloban un código mucho más extenso y profundo.

#### *Procesamiento Digital de Imágenes*

La visión es sin duda el sentido más empleado por la especie humana, y por lo mismo frecuentemente se olvida su importancia. En general, el procesamiento de imágenes consiste en alterar la información visual para obtener mejores resultados o para aislar algunas características particulares de las imágenes. (P. Pérez, M. Valente, 2018).

El procesamiento digital consiste en aprovechar el uso de dispositivos inteligentes para realizar el procesamiento de imágenes y obtener rasgos característicos dentro de las imágenes de manera automática.

#### *Mapa de Bits (Bitmaps)*

“Un Bitmap es un modo elemental para representar imágenes digitales como información en el hardware, específicamente la memoria, de un computador. Consiste, básicamente, en formar arreglos de elementos (vectores, matrices, tensores) ordenados de modos específicos.” (P. Perez, M. Valente, 2018). Los bitmaps consisten entonces en una manera de representar las imágenes de manera organizada para un ordenador y que este entienda que se tiene allí.

### ***Lenguaje No Verbal***

“La comunicación no verbal, también llamada lenguaje corporal, es el proceso de enviar y recibir mensajes sin utilizar palabras, ya sean habladas o escritas” (O. Serrano, 2022). El lenguaje no verbal es aquel que permite la comunicación por medio de gestos o señas. En algunos casos permite realizar énfasis en las palabras, aunque el interés en este caso es la necesidad de este para personas con problemas auditivos o del habla.

### ***Discapacidad***

La discapacidad es un concepto que evoluciona y que resulta de la interacción entre las personas con deficiencias y las barreras debidas a la actitud y al entorno, que evitan su participación plena y efectiva en la sociedad, en igualdad de condiciones con las demás. (Convención de la ONU, 2006.)

La discapacidad es entonces un problema o una situación particular en que la personas se ve privada a realizar ciertas acciones.

### ***Accesibilidad Tecnológica***

La accesibilidad tecnológica se define como el mecanismo que permite la comprensión de los avances tecnológicos y su correcta aplicación en su vida diaria para todas las personas, sin importar sus deficiencias o discapacidades. Es decir, aquello que permite que las tecnologías actuales puedan ser usadas por todo tipo de personas, así como permitir que los servicios sean posibles para el mayor número de personas.

## *Inteligencia Artificial*

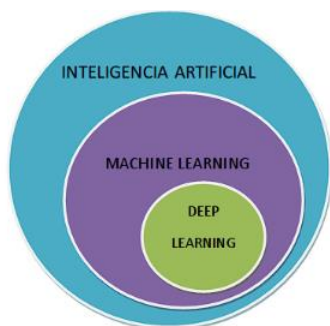
En mis seminarios, intento simplificar el tema definiendo la IA como «la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana».

Pero, para brindar una definición más detallada, podríamos decir que la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. (Rouhiainen, L., 2018).

Se puede intuir que la inteligencia es la habilidad de los ordenadores para realizar acciones o tomar decisiones como lo haría el ser humano, esta engloba términos como Machine Learning y Deep Learning como se aprecia en la *figura 2*, los cuales componen complejos procesos dentro de sí. Para este proyecto se utilizará Machine Learning con ayuda de una máquina de aprendizaje creada por Google llamada *Teachable Machine*, la cual arroja un modelo entrenado de red neuronal convolucional con una serie de imágenes capturadas previamente, este modelo se descarga para TensorFlow y se ejecuta en Python con ayuda de Keras para crear un clasificador. Se tiene la siguiente distribución dentro de la inteligencia artificial:

### **Figura 2**

*Inteligencia Artificial.*



*Nota.* Adaptado de Deep Learning [Grafico], por manurosia4, 2022, smartecorganic.



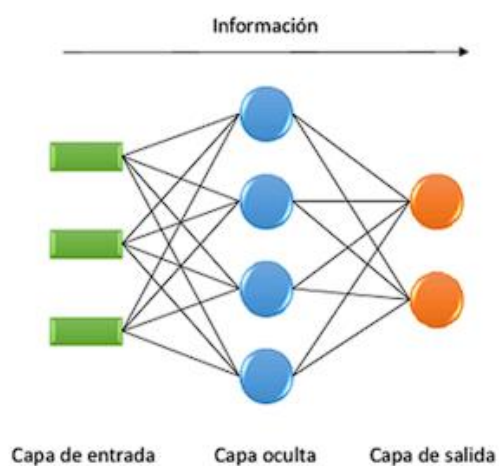
## ***Redes Neuronales Artificiales***

Una Red Neuronal Artificial (RNA) es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro, y que es utilizada para resolver una amplia variedad de problemas. Debido a su flexibilidad, una misma red neuronal es capaz de realizar diversas tareas convirtiéndose pilar del Deep Learning. (manusoria40, 2022).

Es decir que una red neuronal es una interpretación digital del funcionamiento estructuras del cerebro humano, en la siguiente imagen se puede apreciar su estructura:

### **Figura 3**

*Red neuronal artificial.*



*Nota.* Tomado de Redes Neuronales Artificiales - ANN [Grafico], por manurosia4, 2022, smartecorganic.

Se puede observar que se tiene una entrada de datos, un procesamiento de estos en la capa oculta y una capa de salida que sería las acciones o los eventos a realizar según las entradas del sistema, para este proyecto se utilizan redes neuronales convolucionales con ayuda de Teachable Machine, el cual arroja un modelo de este tipo ante muchas entradas.

### ***Redes Neuronales Convolucionales***

El aprendizaje profundo (Deep Learning) basado en redes neuronales convolucionales es un algoritmo de clasificación, los algoritmos de aprendizaje profundo solamente necesitan imágenes de entrada para clasificar, mientras que otros algoritmos necesitan que se describa la información de entrada al inicio. Los algoritmos infieren las características más relevantes y así pueden implementarlos como filtros de textura, color o forma los cuales son los más utilizados. Gracias a lo mencionado anteriormente, estos algoritmos de aprendizaje profundo se utilizan con antelación para reconocimiento de imágenes. (Suaste S y Peña E, 2020).

Se trata de una red neuronal que simplifica en gran medida el proceso, ya que no es necesaria la descripción que otros algoritmos necesitan, con las imágenes de entrada se tiene un clasificador que permite ser usado en el procesamiento de imágenes.

### ***Tarjeta De Desarrollo***

Las placas de desarrollo incluyen cosas como una fuente de alimentación, soporte para conectar sensores y, a veces, incluso sensores y actuadores a bordo. También permiten a los usuarios conectar rápidamente sensores y actuadores (si no están ya incluidos en la placa) y el software que los acompaña facilita la creación y el despliegue del código. (Lemus I, 2019).

Las tarjetas de desarrollo permiten tener un todo en uno a la hora de diseñar prototipos de manera sencilla. Cuenta con todo tipo de herramientas dentro de un circuito electrónico programable que dependiendo del tipo cuenta con un puerto para programarse o se es necesario un circuito externo.

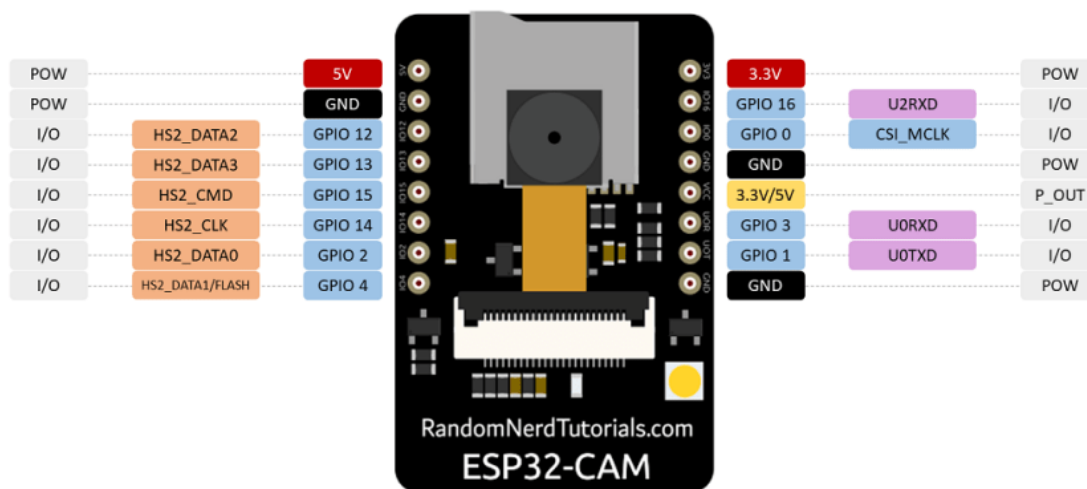
### ***ESP32-CAM***

ESP32-CAM, es un dispositivo que puede llamarse un todo en uno. Aparte de la conectividad Wifi y Bluetooth que viene de fábrica, pines GPIO, se le han añadido dos opciones más. Lleva integrado una pequeña cámara de video y una conexión para una tarjeta MicroSD, donde podremos almacenar fotos o videos. (Pascual C, 2022).

La ESP32-CAM es una tarjeta de desarrollo basada en un chip integrado que cuenta con diversas características entre las cuales a destacar se encuentra la cámara y conexiones por medio de wifi, bluetooth en la *figura 4* se puede observar su esquemático:

#### Figura 4

*ESP32-CAM AI THINKER.*

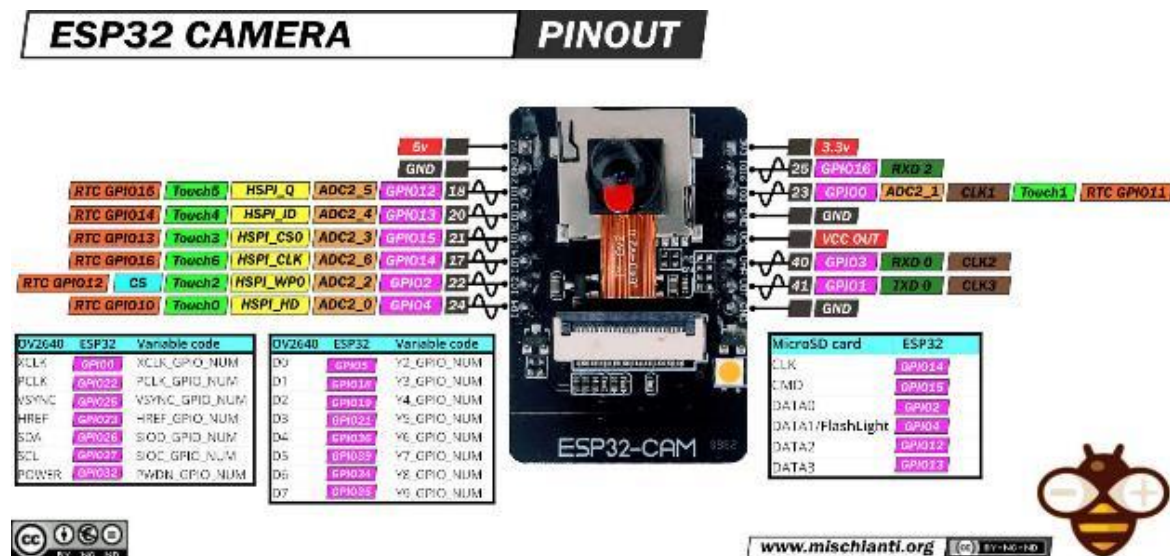


*Nota.* Tomado de ESP32-CAM AI Thinker Pinout Guide [Imagen], 2022, randomnerdtutorials.

Como la ESP-32 CAM es una variación de la tarjeta ESP32 con el mismo chip, utiliza ciertos pines para el uso de la cámara y una ranura de tarjeta SD. A continuación, en la *figura 5* se pueden observar estas y algunas otras características relacionadas con cada puerto de conexión de la tarjeta.

Figura 5

ESP32-CAM pines en uso.



*Nota.* Tomado de ESP32-CAM: pinout, specs and Arduino IDE configuration [Imagen], por Renzo Mischianti ,2021, mischianti.

En la anterior imagen se pueden observar los pines con los que cuenta la ESP32-Cam. La tabla de OV2640 (referencia de la cámara) relaciona la variable con su código y el pin que utiliza se puede observar en la figura 6:

Figura 6

ESP32-CAM Tabla de pines en uso por la cámara.

OV2640	ESP32	Variable code
XCLK	GPIO0	XCLK_GPIO_NUM
PCLK	GPIO22	PCLK_GPIO_NUM
VSYNC	GPIO25	VSYNC_GPIO_NUM
HREF	GPIO23	HREF_GPIO_NUM
SDA	GPIO26	SIOD_GPIO_NUM
SCL	GPIO27	SIOC_GPIO_NUM
POWER	GPIO32	PWDN_GPIO_NUM

OV2640	ESP32	Variable code
D0	GPIO5	Y2_GPIO_NUM
D1	GPIO18	Y3_GPIO_NUM
D2	GPIO19	Y4_GPIO_NUM
D3	GPIO21	Y5_GPIO_NUM
D4	GPIO36	Y6_GPIO_NUM
D5	GPIO39	Y7_GPIO_NUM
D6	GPIO34	Y8_GPIO_NUM
D7	GPIO35	Y9_GPIO_NUM

*Nota.* Tomado de ESP32-CAM: pinout, specs and Arduino IDE configuration [Imagen], por Renzo Mischianti ,2021, mischianti.

Por otra parte, los pines que son utilizados para la ranura de la tarjeta SD son los siguientes:

### Figura 7

*ESP32-CAM pines usados para la tarjeta SD.*

MicroSD card	ESP32
CLK	GPIO14
CMD	GPIO15
DATA0	GPIO2
DATA1/FlashLight	GPIO4
DATA2	GPIO12
DATA3	GPIO13

*Nota.* Tomado de ESP32-CAM: pinout, specs and Arduino IDE configuration [Imagen], por Renzo Mischianti ,2021, mischianti.

Lo que deja a la tarjeta de desarrollo con 10 pines GPIO que se pueden configurar como salidas o entradas con ciertas características cada uno.

### Tiristor

El tiristor es un dispositivo similar al transistor, su base funciona con el mismo método. Es un semiconductor diseñado con la mezcla de diferentes materiales, pero con una estructura interna diferente, como se aprecia a continuación en la *figura 8*:

### Figura 8

*Diagrama y composición del tiristor.*

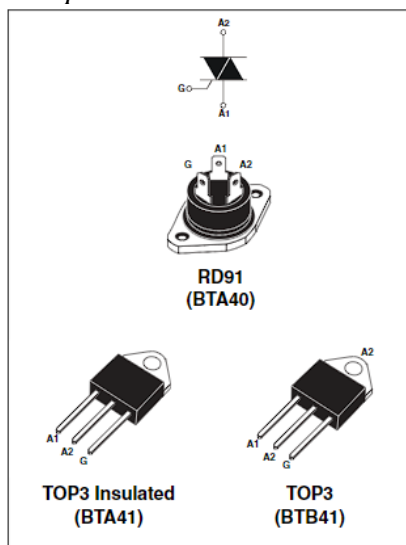


*Nota:* Tomado de Que es un triac y cuáles son sus aplicaciones [Imagen], por Ingeniería Mecafenix ,2018, ingmecafenix.

Como se puede observar se cuenta con una estructura similar a la de un transistor, pero en práctica se tienen usos diferentes ya que este permite la conducción de corriente alterna mediante su excitación en el gate. El tiristor que se utilizará durante el desarrollo del proyecto será el BTA41, el cual se encargará de la activación de una de las etapas de potencia dejando circular la corriente mediante el uso de la señal correcta. Estos se pueden encontrar de la siguiente manera como se muestra en la *figura 9*:

### Figura 9

*Encapsulados del tiristor.*



*Nota.* Tomado de hoja de datos de BTA40, BTA41 and BTB41 Series [Imagen], por datasheet ,s.f, ALLDATASHEET.

### *Ley de Ohm*

“Esta ley de la electricidad fue postulada por Georg Simon Ohm, y establece que el voltaje (V) que existe a través de una resistencia (R) es directamente proporcional a la corriente que circula por la misma (I).” (Macías García et al., 2018). La ley de Ohm es una ley base en electricidad, permite comprender el funcionamiento mediante el cual se rige la electricidad en un

circuito y calcular los valores que hacen al circuito funcionar. Valores como el voltaje, resistencia e intensidad.

### ***Leyes de Kirchhoff***

Las dos leyes de Kirchhoff permiten comprender el funcionamiento de circuitos eléctricos complejos mediante dos bases que siempre se cumplirán a la hora de analizar un circuito.

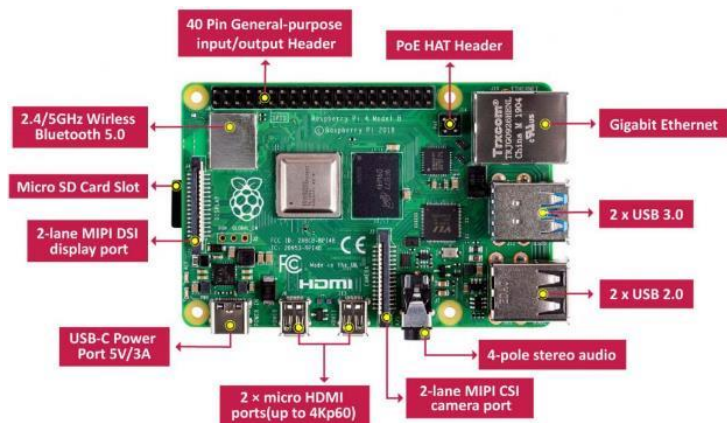
La primera ley establece que: “La suma de las corrientes que entran a un nodo (corriente total de entrada) es igual a la suma de las corrientes que salen de dicho nodo (corriente total de salida).”

La segunda ley establece que: “En un circuito, la suma de todas las caídas de voltaje localizadas en una trayectoria cerrada única es igual al voltaje de fuente total encontrado en dicha espira”

### ***Raspberry Pi***

La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que corre un sistema operativo Linux capaz de permitirle a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python. (Raspberry Pi, 2019).

Se trata de una tarjeta de desarrollo potente y se define como un “miniordenador” debido a la integración de un propio sistema operativo y la capacidad de procesamiento que tiene. Es una alternativa con más potencia mediante la cual se puede realizar el proyecto cumpliendo con todas las necesidades de este, pero no se utilizará durante el desarrollo del proyecto.

**Figura 10***Raspberry pi 4.*

*Nota.* Tomado de Raspberry Pi 4 GPIO Pinout [Imagen], por AliOs ,2019, *keytosmart*.

En la *figura 10* se puede observar la Raspberry pi 4 y sus características, como es de esperarse, es una tarjeta más potente debido a su capacidad de procesamiento entre otros. Se trata de un miniordenador que cuenta con todo lo necesario para ejecutar programas dentro además de tener 40 pines para uso general que pueden ser programados, cuenta con puertos USB, ranura para tarjeta SC, módulo de bluetooth, conexión para Display Port, 2 micro HDMI, conexión para audio estéreo y una ranura para el módulo de cámara.

### ***Protocolo MQTT***

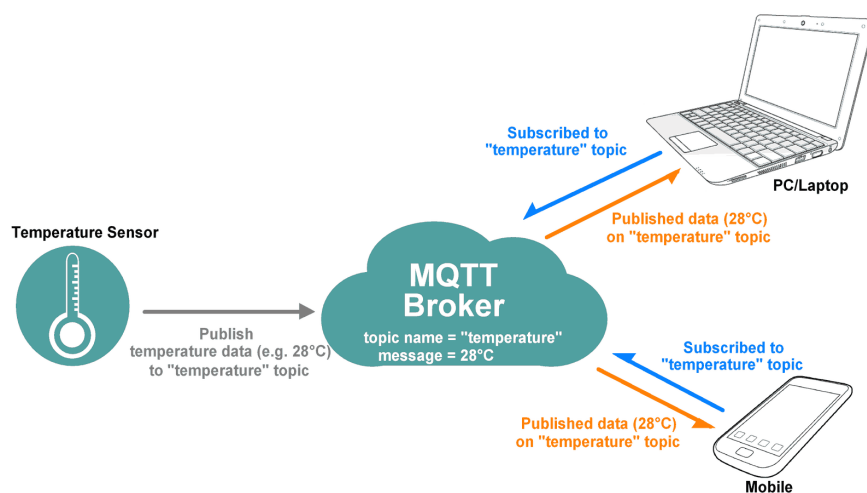
El protocolo MQTT funciona sobre TCP/IP o sobre otros protocolos de red con soporte bidireccional y sin pérdidas de datos, teniendo como principal característica el uso de mensajes “broadcast” para suscripción y publicación de datos con independencia de la aplicación. En cuanto a su arquitectura, sigue una topología en estrella, donde existe un nodo central o bróker con capacidad para trabajar con un gran número de clientes y también encargado de gestionar la red y transmitir los mensajes. (Aguirre et al., 2019).



Este protocolo permite la comunicación entre dispositivos, es decir es un tipo de comunicación maquina a máquina, es decir una comunicación entre dispositivos electrónicos sin intervención humana siempre y cuando se tenga este soporte.

## Figura 11

*Topología MQTT bróker.*



*Nota.* Tomado de Que es MQTT. [Imagen], por Firtec ,s.f, Firtec.

En la anterior imagen, la figura 11, se puede observar un ejemplo de su funcionamiento, un sensor realiza la captura de una variable, esta es enviada al bróker y otro dispositivo, celular o portátil puede acceder o recibir la información que es capturada en el sensor, permitiendo pues acceder a esta información siempre y cuando el protocolo se encuentre en funcionamiento, así como que el dispositivo esté conectado a la red.

### ***PubSubClient.h***

Se trata de una librería de Arduino que permite al cliente tener una mensajería mediante el cual un dispositivo recibe y el otro envía. Esta mensajería se realiza mediante un servidor que soporte MQTT.

### ***Paho.mqtt***

Es una librería para Python que permite la comunicación entre dos dispositivos mediante el envío de mensajes cortos, un dispositivo debe ser el suscriptor y otro debe ser el que envía el mensaje. Sus mensajes deben enviarse mediante un servidor que permita una conexión MQTT.

### ***Web Server.h***

Consiste en una librería que permite la creación de un servidor en la tarjeta de desarrollo, permitiendo la creación de una cámara web.

### ***Esp32cam.h***

Es una librería que añade una amplia variedad de librerías dentro de sí, todas diseñadas para trabajar con la tarjeta de desarrollo ESP32-CAM. Dentro de estas librerías se encuentran aquellas que permiten simplificar la programación para la captura de datos mediante la cámara de la tarjeta de desarrollo.

### ***Tensorflow***

Es una librería de código abierto que permite hacer uso del aprendizaje automático con facilidad. Tiene recursos diseñados por Google que permiten la implementación de Machine Learning mediante código abierto. Estas librerías se pueden usar en diferentes entornos de programación y con diferentes compiladores, como los son Python, C++ entre otros lenguajes de programación reconocidos.

### ***Teachable Machine***

Diseñado por Google, se define a sí mismo como "una herramienta basada en la web que hace que la creación de modelos de aprendizaje automático sea rápida, fácil y accesible para todos." (*Teachable Machine*, s. f.). Es una herramienta que facilita la creación de modelos de aprendizaje mediante el uso de procesamiento de imágenes o de voz.

### ***Keras***

Keras es una aplicación diseñada para crear, probar y realizar diferentes modelos mediante redes neuronales. Se encuentra diseñada en el lenguaje de Python y se inicia mediante Frameworks como TensorFlow, Microsoft Cognitive Toolkit o Theano. Para este proyecto se utilizará mediante Python con TensorFlow en el entorno de desarrollo de PyCharm.

### ***PyCharm***

Es un entorno de desarrollo que permite la programación de Python, permite la compilación de código, aplicación de librerías y se encuentra programado en Python junto con Java.

### ***Numpy***

Consiste en una librería desarrollada para Python, esta librería permite el manejo de grandes cantidades de datos en forma de vectores y matrices. Se utiliza en el manejo de la matriz creada en la captura de datos, debido a las imágenes capturadas se manejan en forma de matrices,

### **Estado del Arte**

En el artículo de Luo, Congbo, Hao, Yunhui y Tong, Zihe. (2018) se encuentra un análisis del estado del procesamiento digital de imágenes realizado en el 2018 se encuentra compuesto por diferentes partes de interés, en particular su funcionamiento y aplicaciones que se les puede dar de manera teórica tiene mucha importancia pues sirve como base de explicación a lo que se quiere lograr mediante el uso de esta tecnología, sus aplicaciones y uso ayuda a comprender el alcance que tienen y permitiendo determinar algunas herramientas claves para su uso y aplicación. Existen muchos artículos con diferentes aplicaciones que se le han dado a el procesamiento digital de señales, pero para el proceso con el que se está trabajando es importante conocer el estado en que se encuentra el lenguaje de señas, para ello R. Avila (2022) ha realizado un estado del arte donde expresa la importancia a la hora de hablar de inclusión, accesibilidad y calidad de vida donde permite comprender el estado actual de las tecnologías para personas con discapacidades auditivas y del habla, se trata de un estado del arte, que permite profundizar en el tema que se está tratando, así como también permite conocer las aplicaciones que se han hecho, permite conocer posibles mejoras que se puedan realizar en diferentes entornos, por lo que es determinante para abordar la problemática de una manera más precisa y dar a conocer la importancia que tiene la creación de nuevos sistemas de accesibilidad.

Para entrar en profundidad del procesamiento de imágenes, se tiene el artículo de Tyagi, Vipin. (2018). Este artículo permite profundizar en el procesamiento digital de imágenes, sus conceptos, su funcionamiento y la metodología de cada sistema, aspectos como la representación de escalas de colores, los mapas de bits son fundamentales comprenderlos a la hora de manejar con sistemas de procesamiento de imágenes, en este artículo se trabaja con Matlab, el cual es una de las herramientas comúnmente usadas para este fin, esto permite un acercamiento de manera

teórica a una profundización a los sistemas digitales de procesamiento de imágenes, tema con el que se espera dar solución a la problemática. En el procesamiento digital de imágenes se suele usar con frecuencia el lenguaje de Python y este es el caso de Shreya, Debbata. (2021), el cual consiste en un artículo que es un ejemplo de un sistema de reconocimiento digital de imágenes aplicado mediante el uso de Python, el cálculo de escala de grises, la creación de una red profunda para el reconocimiento de imágenes son aspectos importantes para tener en cuenta, el artículo resume la aplicación de un sistema para el reconocimiento facial, lo cual se acerca en mayor medida a la solución deseada permitiendo tomar este artículo como ejemplo permite comprender el funcionamiento de un sistema de este tipo creado en Python y analizar sus resultados para determinar su alcance y posibles aplicaciones mediante este lenguaje, otro ejemplo de aplicación para el lenguaje de signos, en este caso usando Deep learning con la herramienta de TensorFlow creado por D. Someshwar, D. Bhanushali, V. Chaudhari y S. Nadkarni (2020), en el cual se utiliza el procesamiento digital de imágenes para construir un sistema de asistencia virtual mediante gestos, se resumen aspectos importantes como el uso de TensorFlow para la creación de la red neuronal profunda, entrenando el sistema para su reconocimiento. Por otra parte, se explica los diferentes softwares que se usaron en la metodología, los conceptos como convolución que son claves en el procesamiento digital de imágenes, se trata proyecto aplicado y permite conocer el paso a paso del planteamiento, por lo que es de utilidad sobre todo para conocer una posible solución usando este como una base del sistema que se busca diseñar.

Existen muchos usos dados a el procesamiento digital de imágenes de manera global, muchos de ellos sobre aplicaciones como las presentadas anteriormente y en Colombia, han utilizado esta tecnología en diferentes espacios, en el artículo de Zuleta, A. M. P., y Sanjuán, E.

D. J. Z. (2018) se realiza un estudio de los diversos métodos de teledetección de alta resolución mediante UAVs (Vehículos Aéreos No Tripulados) o satélites, el artículo es una recopilación de los trabajos realizados con el procesamiento digital de imágenes para el desarrollo y mantenimiento de diferentes cultivos donde se definen diferentes usos que se le da a la información capturada mediante imágenes, algunas de estas capturas ayudan a prevenir enfermedades, revisar ciertos factores importantes en los cultivos, por ejemplo, un estudio realizado fue el siguiente:

El Grupo de Investigación Macrypt de la Universidad de los Llanos, y el Grupo de Investigación en Procesamiento de Señales DSP- UPTC de la Universidad Pedagógica y Tecnológica de Colombia, desarrollaron un software para determinar el comportamiento del índice de vegetación NVDI y los agentes que afectan el desarrollo de las plantas. Para realizar esto, primero se capturaron imágenes por medio de un UAV de un cultivo de prueba (Palma de Aceite, posteriormente se procesaron las imágenes en la aplicación desarrollada en PYTHON, esto con el fin de extraer los Componentes de la información espectral de la cobertura y determinar el índice de vegetación NDVI (Jiménez et al., s.f.).

El procesamiento digital de imágenes permite determinar factores importantes en la agricultura y permitir una agricultura de precisión mediante el análisis de los datos que se obtienen, también existen usos más amplios que se le puede dar al procesamiento digital de imágenes, algunas de ellas son en el análisis de tierras, un análisis multitemporal de las coberturas y usos del suelo del municipio de San Francisco de Sales, ubicado al noroccidente del departamento de Cundinamarca en Colombia, en el trabajo de grado de Niño Martínez, Y. M. (2020), se puede observar algunos aspectos claves en el manejo de procesamiento digital de imágenes, en este caso no son imágenes en tiempo real y muchas de ellas no son recientes, se manejan imágenes de diferentes años, diferentes momentos tomados por satélites (Landsat 4 TM y Sentinel-2 MSI) y tratadas para realizar un análisis en los cambios en las coberturas terrestres y

para su procesamiento digital utiliza programas especializados como QGIS, ARGIS y ENVI, los cuales son programas para el estudio geográfico de terrenos, el tipo de procesamiento permite comprender aspectos como valores de reflectancia, aspectos que permiten obtener una mejor imagen para el tipo de estudio que se desee. Prosiguiendo, se tiene el artículo de Escobar. A. et al. (2023) en el cual se realiza la aplicación de un sistema de procesamiento de imágenes para analizar las irregularidades en el pavimento de Colombia, en el artículo se aplican diferentes métodos a la hora de realizar el procesamiento digital de imágenes los cuales dan a conocer una idea de lo que es aplicarlos a un entorno más amplio como el estudio de carreteras, muchas de las técnicas usadas son técnicas básicas a la hora de aplicar el procesamiento de imágenes, el escalado de grises, el tratamiento de las imágenes capturadas mediante filtros, todo con el objetivo de encontrar cualquier grieta o desgaste en algunas de las carreteras que ellos analizaron, se trata de un artículo interesante que profundiza sobre todo en el preprocesamiento de las imágenes, pues se le aplican diversas técnicas para obtener buenos resultados, técnicas como la binarización de por triangulo y transformación de perspectiva inversa son dos aspectos de importar, pues se buscara realizar una captura de datos con ayuda de estas y otras técnicas a la hora de capturar imágenes en la base de datos, un ejemplo cercano desarrollado con respecto al lenguaje de señas es el caso de Pinzón-Bayona, G., y Sanabria-Orjuela, Y. G. (2022), quienes desarrollaron una aplicación que permite traducir el lenguaje de señas mediante servicios web, en el trabajo realizado se ven diferentes aspectos como lo es el procesamiento de imágenes y el funcionamiento de diferentes etapas a la hora de realizar el seguimiento de las manos y si bien profundiza en aspectos de diseño de aplicaciones y demás aspectos que no son relevantes en este documento, permite comprender el funcionamiento base de sistema web que utilizo para realizar la traducción.

## Metodología

La investigación para realizar es del tipo cuantitativa no experimental, se busca construir un prototipo con la idea de mejorar la calidad de vida de las personas con discapacidades auditivas y del habla, así como una investigación exploratoria en busca del alcance que tiene el sistema con el objetivo de dar solución al problema, el prototipo englobará a las personas con discapacidades auditivas y del habla, pero tendrá un enfoque en la domótica, es decir se realizará en una residencia de la ciudad de Pitalito-Huila con dos personas con dificultades del habla, auditivas siendo estos los beneficiarios del proyecto, con la idea de ayudar a futuras investigaciones sobre la temática se realiza la prueba con una de estas dos personas.

Para ello se utilizarán métodos de observación del sistema construido mediante un sujeto de prueba (humano) el cual realizará la comprobación y verificación del funcionamiento del sistema, así como también se apoyará en diferentes casos de estudios en cuanto al uso del lenguaje de señas para complementar y contrarrestar los datos obtenidos.

Como solución se considera la implementación de un sistema de control mediante un lenguaje de señas, este sistema será desarrollado con una división importante en software y hardware por lo que es necesario un medio de comunicación para la unión de estas dos etapas, el software es procesado en un ordenador remoto y la parte del hardware realizara la captura de datos y activación desactivación de las etapas de potencia.

El desarrollo de la solución requiere construir un prototipo que permita mejorar la calidad de vida para personas con estas dificultades para ello, primero se busca tener un lenguaje de señas del que partir.



## **Etapas de Exploración**

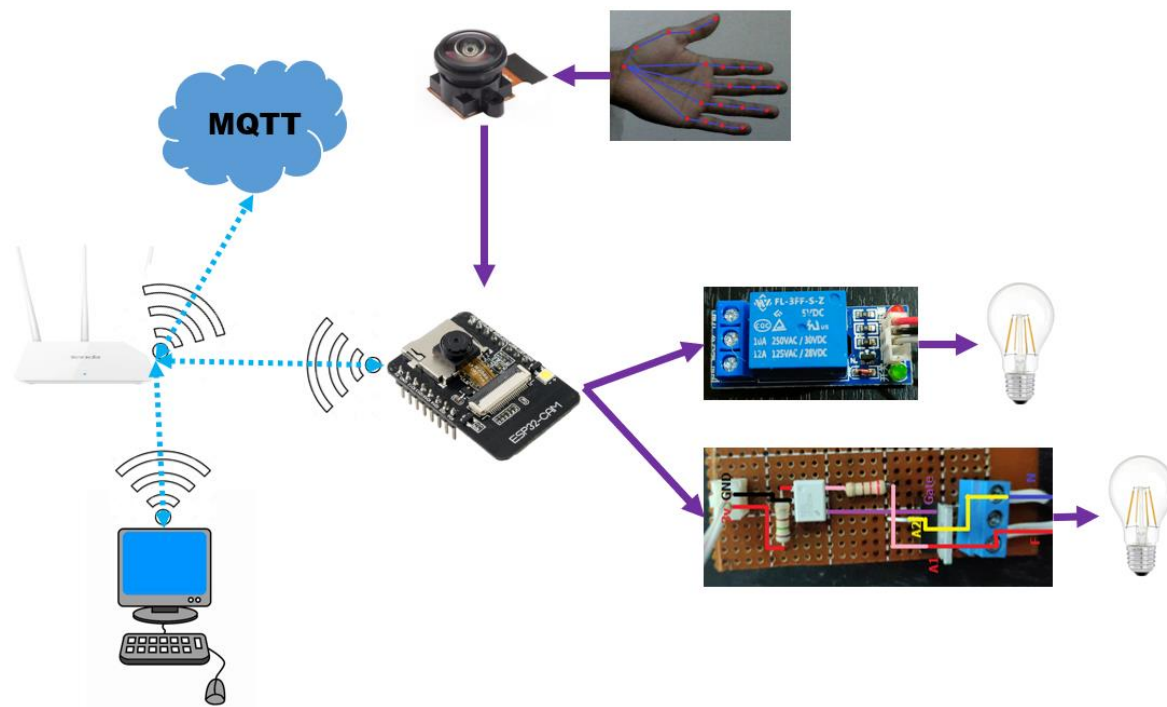
En esta etapa es realizada la búsqueda de información, y se seleccionan tanto el lenguaje de señas a utilizar como las letras a usar, existen muchos tipos de lenguajes de señas, algunos pueden ser incluso propios de comunidades y se busca un estándar, algo que englobe una comunidad y este dentro un mismo idioma pues muchas veces difieren ciertos términos o letras por ello se selecciona el lenguaje de señas de Colombia, que engloba algo nacional y permite tener un lenguaje del mismo entorno sociocultural, de este lenguaje de señas se seleccionan cuatro letras correspondientes al alfabeto utilizado en el idioma español, estas letras permitirán simplificar el gesto para que el prototipo pueda actuar con más rapidez, dos serán de encendido y dos de apagado.

## **Etapas de Software**

Esta etapa será la que une todo, será la encargada de hacer que todo funcione, empieza con los datos capturados por la cámara, en la *figura 12* se puede observar el esquema general, los datos se enviarán a el servidor creado mediante una conexión con el Router en la cual se van enviando datos en tiempo real y para esto se apoyará en el uso de diversas librerías, estas librerías simplificarán enormemente el proceso y la cantidad de líneas de código a escribir, con el servidor formado, un software que se construirá en Python permitirá la captura de la cámara web y se podrá visualizar de esta manera por medio de la IP lo que se está capturando en tiempo real, con esto ya se tendrían los datos en el ordenador para ser procesados, .

**Figura 12**

*Conexión general del sistema.*



*Fuente.* Elaboración propia

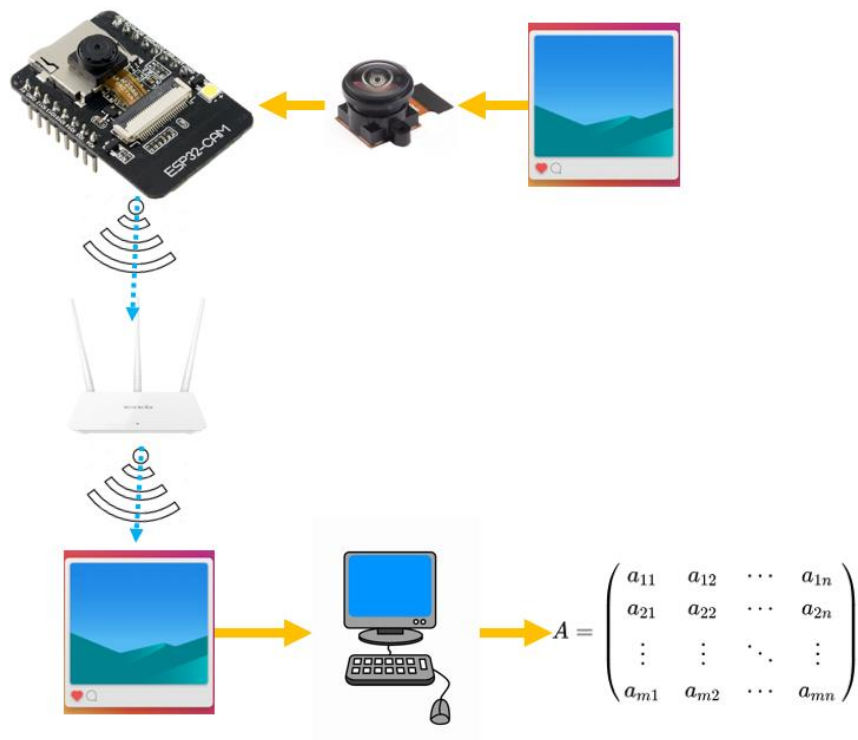
La captura de los datos se realiza mediante el comando `cap = cv2.VideoCapture(1)`, mediante el cual se realiza la captura de la cámara web con ayuda de las librerías de Computer Vision.

El paso siguiente es el procesador, es necesario transformar los datos obtenidos, ya que estos llegan en forma de imágenes, imágenes tomadas muy rápido y que es necesario convertir en matrices para poder manipularse de manera informática y para este proceso se utilizarán librerías (Numpy), pues es un proceso complejo y el uso de librerías permitirán que sea más rápido, se transforman entonces imágenes a las cuales se les puede modificar el color, escalas de grises, entre otros parámetros que pueden comprimir y facilitar el tratamiento de las matrices, con el procesamiento del mapa de bits se tendrá entonces matrices de información que se

utilizaran para detectar patrones, en la *figura 13* se puede observar una ilustración que explica el proceso que se lleva a cabo:

**Figura 13**

*Matriz de una imagen*



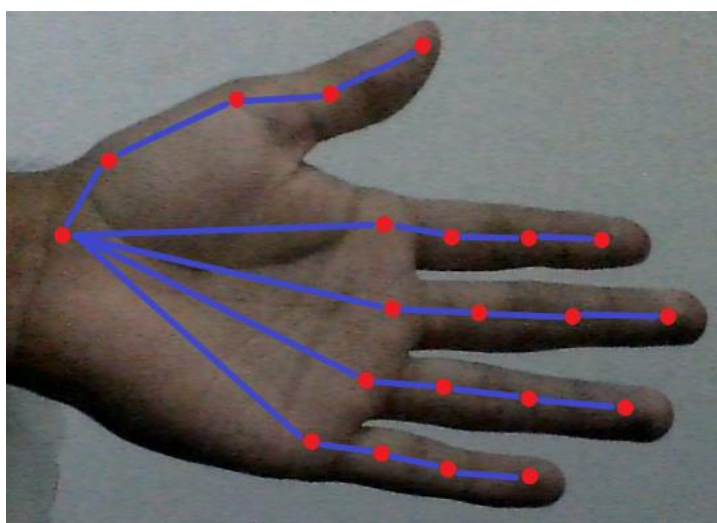
*Fuente.* Elaboración propia

El uso de las imágenes capturadas como matriz de datos se realizará mediante líneas de código, en estas líneas se define el tipo de variable, la url de la cámara web mediante `urllib.request.urlopen(url)` permitirá ir abriendo las imágenes que se capturan y se utilizara `np.array` para manejar las imágenes en forma de matrices, con ayuda del código donde estará la captura de las manos llamado `HandTrackingModule` como `HandDetector` permitirá ajustar los parámetros de los patrones de las manos, en estos patrones de las manos estarán en cada articulación donde se buscará dibujar un círculo y trazar una línea entre estos para construir una especie de “esqueleto” de la mano consiguiendo con esto una visualización del movimiento de

las manos, pero ahora de manera informática, es decir que se sabrá por medio de matrices con ayuda de `imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)`, donde se transforman imágenes en matrices, esto dirá como se están moviendo las manos y que están haciendo y con esto se tendrá el sistema de seguimiento de manos como se observa en la *figura 14*:

### Figura 14

*Idea de seguimiento de la mano.*

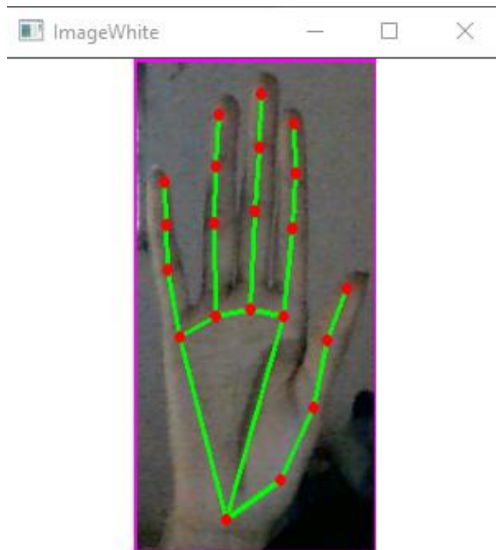


*Fuente.* Elaboración propia

Luego, se busca realizar un mapeo de las señas, es decir enseñarle a un programa informático las matrices que quieren decir una letra en el lenguaje de señas, pero para ello se deben tomar muchas imágenes de la mano haciendo esa seña, así, aunque este de lado o un poco movida, pueda identificar correctamente que la seña es la letra en cuestión, para esto es necesaria una base de datos, en esta base de datos se tendrá un estándar, el estándar será que cada imagen tendrá el mismo tamaño (300x300 pixeles) y estará dentro de un rectángulo automáticamente creado con base en lo mucho que se extienda la mano, a continuación se muestran dos ejemplos, una mano extendida en vertical y otra en horizontal como se muestra en las *figuras 15 y 16* respectivamente.

**Figura 15**

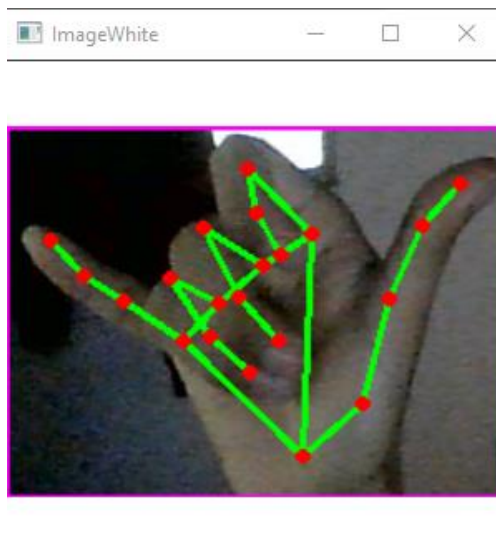
*Seguimiento de la mano en vertical.*



*Fuente.* Elaboración propia

**Figura 16**

*Seguimiento de la mano en horizontal.*



*Fuente.* Elaboración propia

Con el estándar que tendrá cada imagen, se realiza la captura de alrededor de 300 imágenes de una mano haciendo el gesto deseado para cada uno, para esto se realizará un programa que permita la captura rápida de datos, mientras este presionada una tecla se guardaran imágenes, alrededor de 1200 imágenes entre los 4 gestos, se espera adaptar el siguiente comando para la realización de la captura de datos, mientras se esté presionada, ira realizando el guardado de imágenes en una carpeta deseada.

Con estas imágenes se realiza el entrenamiento de la inteligencia artificial la cual creara un modelo, este modelo será creado mediante Tensorflow con ayuda de Teachable Machine y se manipulara mediante el lenguaje de programación de Python, como se busca que una active la etapa de potencia y otra lo apague, con ayuda del protocolo MQTT se unen entonces todo el cálculo realizado por el ordenador y la tarjeta de desarrollo la cual activa/desactiva los actuadores, así como de realizar la función de cámara web.

Por último, se tiene la lógica del sistema en la tarjeta de desarrollo, aquello que hará cuando reciba datos del ordenador por medio del Bróker y esta será una parte simple pues simplemente tendrá que activar, desactivar según la letra que coincida, el ordenador enviara una “a” o una “b” y la tarjeta determinara cual letra es y la acción a llevar a cabo. En esta parte se debe tener en cuenta las limitaciones de la ESP32-CAM que es la tarjeta con la que se está manejando ya que esta cuenta con 10 pines libres para programarse, es decir que se podría tener un dispositivo que permita conectar hasta 10 dispositivos electrónicos a la tarjeta y controlarlos mediante gestos, a su vez que sería necesaria la ampliación del modelo de aprendizaje pues por cada gesto de activación se requiere uno que lo desactive, esto representa un esfuerzo mayor pues sería necesario en caso de usar al máximo las capacidades de la tarjeta, tener 20 gestos mapeados para ser reconocidos, por lo que a manera de prototipo se manejaran dos dispositivos

diferentes, es decir existirán 4 gestos, dos activaran dos dispositivos eléctricos diferentes y otros dos gestos se encargaran de realizar el apagado.

**Tabla 1**

*Secuencia lógica de control*

Seña	Relé	Triac
A	1	X
Ab	X	1
B	0	X
H	X	0

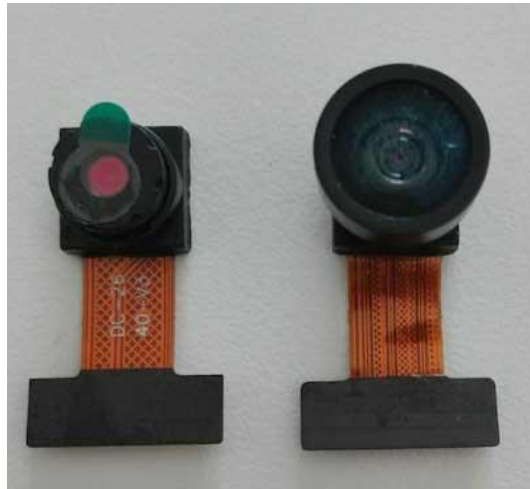
*Fuente.* Elaboración propia

### **Etapas de Hardware**

En la etapa de hardware se tiene la captura de datos, esta captura esta realizada por una cámara, esta cámara es una tarjeta de desarrollo que cuenta con una cámara integrada, la ESP32-CAM, mediante esta tarjeta de desarrollo se realizará la captura y el procesamiento de los datos de esta captura para enviarlos a un servidor local y funcionar como una cámara web, la tarjeta capturara entonces el video y se podrá acceder a este en tiempo real mediante una IP, siempre que se encuentre dentro de la misma red, es decir que el Router funcionara como intermediario para la comunicación con el dispositivo.

**Figura 17**

*Cámara OV2640 de la ESP32-CAM normal de 66° y otra con amplitud de 120°.*



*Nota.* Tomado de ESP32 CAM introducción y primeros pasos [Imagen], por Charly Pascual, 2022, *programarfacil*.

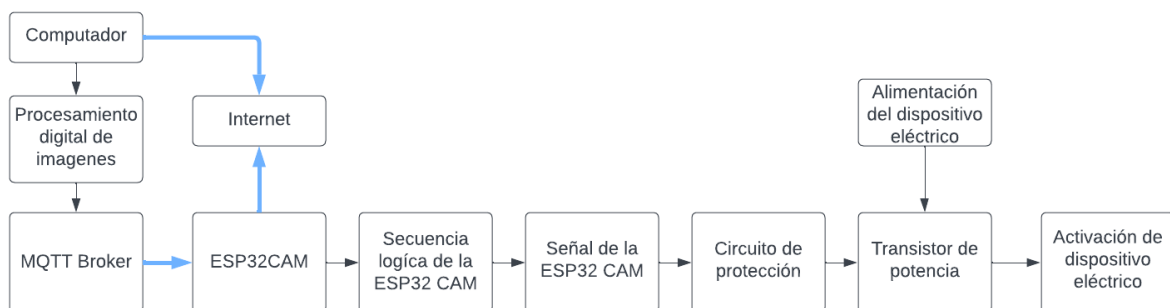
Esta misma tarjeta de desarrollo (ESP32-CAM) realiza el procesamiento del protocolo de comunicación, para ello se apoyará en el uso de librerías, tanto de la cámara web como del protocolo de comunicación, el medio de comunicación adicional a la red creada con el Router, se realiza con el único objetivo de enviar la señal que activará y desactivará las etapas de potencia, la activación y desactivación de las etapas se realiza por medio de la tarjeta de desarrollo la cual activará un controlador de potencia (relé) para dejar que el flujo de corriente llegue al actuador que será una bombilla, de esta manera, la parte de hardware consistirá en una tarjeta de desarrollo, un relé, un fajón, una bombilla y algunos cables para su respectiva conexión, debido a que es un sistema abierto, es decir se plantea conectar diferentes sistemas eléctricos, esta etapa es susceptible a cambios, un relé común puede manejar una potencia cercana de 10 amperios (cantidad dependiente del relé) y genera problemas con algunos dispositivos (chispas y sobrecalentamiento en dispositivos inductores, capacitivos o de gran carga), esto lleva a realizar modificaciones ante el dispositivo a conectar, por lo que se realizara un conexión y



configuración con un TRIAC, permitiendo manipular otro dispositivo de potencia, se puede observar en la *figura 18* como se espera que el sistema esté conectado, donde las líneas azules indican una conexión inalámbrica.

### Figura 18

*Diagrama de la conexión en el sistema de control.*



*Fuente.* Elaboración propia

Por último, se hablará de las diferencias y las mejoras de realizar el proyecto con una Raspberry Pi, las posibilidades de mejora que se pueden obtener en un futuro.

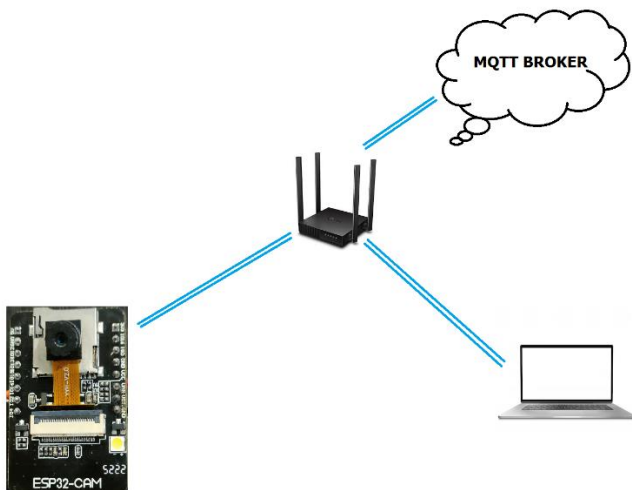
### Protocolo de Comunicación

El protocolo de comunicación será aquel que realice el envío de datos, aquellos datos que la inteligencia artificial comparará, es decir si detecta la señal correcta para activar enviará una señal de activación mientras que para una señal incorrecta será una señal de desactivación, el protocolo está constituido por un MQTT Bróker el cual permite el envío de pequeños paquetes de datos entre máquinas y funciona mediante el protocolo TCP/IP lo cual simplifica y otorga seguridad en el manejo de datos, el sistema permitirá mediante una cámara realizar la captura de imágenes las cuales serán procesadas y mediante comparación permitirá realizar la activación de un dispositivo electrónico (una bombilla, un cierre electrónico, etc.) esto permitirá implementar

el lenguaje no verbal a la domótica, esta comunicación inalámbrica se puede observar en la *figura 19*:

### **Figura 19**

*Esquema simple del sistema de comunicación.*



*Fuente.* Elaboración propia

### **Implementación Física del Prototipo**

La implementación consistirá en la creación de un programa de procesamiento digital de imágenes accionado en un ordenador, posterior a esto, se creará un sistema electrónico que funcionará con base en el programa de procesamiento digital de imágenes, para esto se realizará con ayuda de una tarjeta de desarrollo con cámara, la tarjeta enviará el impulso que permitirá hacer funcionar el circuito de control y con las partes que se encuentran en funcionamiento ya, queda realizar la construcción del prototipo y se irán uniendo las etapas, verificando su funcionamiento.

## Diseño de la Solución

### Selección del Lenguaje y Señas por Utilizar

El lenguaje seleccionado es el lenguaje nacional de señas colombiano, este lenguaje consta de un diccionario con gestos para diferentes letras y términos. Para la selección de señas se enfocó en las configuraciones manuales, aquellas que se refieren a letras específicas, a continuación, se puede observar una imagen con algunas de ellas.

**Figura 20**

*Letras del lenguaje de señas colombiano.*



*Nota.* Tomado de Diccionario Básico De la Lengua De Señas Colombiana [Imagen], por Cabarcas H. et al., s.f, *insor*.

Para ello se selecciona la mano en B que es familiar con el gesto de “parar” que consiste en tener ambas manos abiertas moviéndolas hacia delante y abajo simultáneamente. La letra B es entonces la letra que detendrá el sistema. La mano en A será la correspondiente a la activación del sistema. Las letras seleccionadas son entonces:

### **Figura 21**

*Letra A del lenguaje de señas colombiano.*

Mano en A



*Nota.* Tomado de Diccionario Básico De la Lengua De Señas Colombiana [Imagen], por Cabarcas H. et al., s.f, *insor*.

### **Figura 22**

*Letra B del lenguaje de señas colombiano.*

Mano en B



*Nota.* Tomado de Diccionario Básico De la Lengua De Señas Colombiana [Imagen], por Cabarcas H. et al., s.f, *insor*.

Para un segundo dispositivo se utilizarán otras dos letras, es decir otras dos señas que encenderán y apagarán el dispositivo correspondiente:

**Figura 23**

*Letra Ab de la lengua de señas de Colombia.*

Mano en Ab



*Nota.* Tomado de Diccionario Básico De la Lengua De Señas Colombiana [Imagen], por Cabarcas H. et al., s.f, *insor*.

**Figura 24**

*Letra H del lenguaje de señas colombiano.*

Mano en H



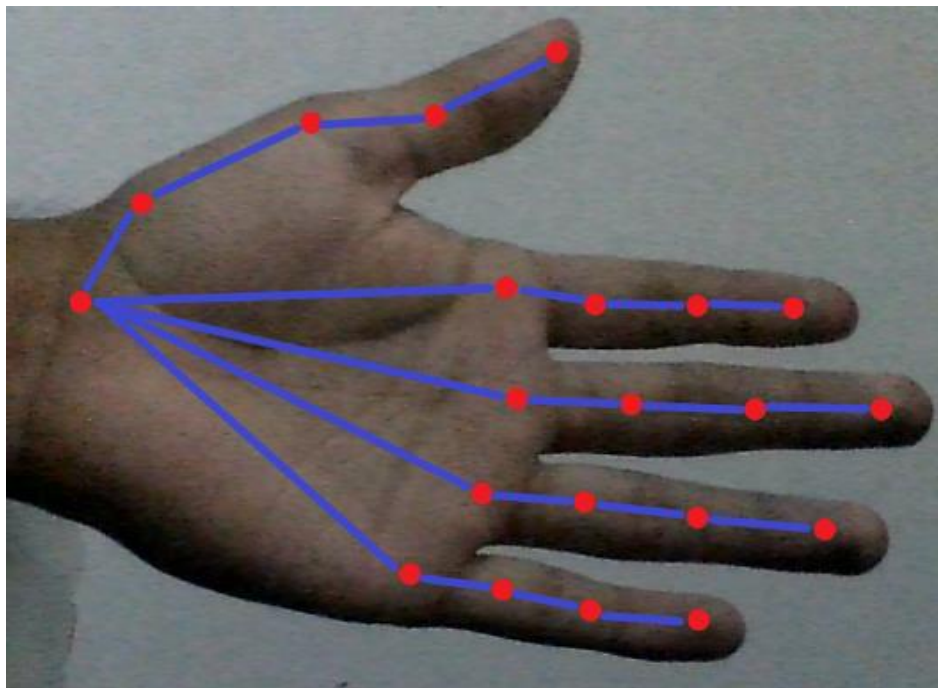
*Nota.* Tomado de Diccionario Básico De la Lengua De Señas Colombiana [Imagen], por Cabarcas H. et al., s.f, *insor*.

**Etapas de Programación**

En la etapa de programación, se diseñará en primera instancia el módulo que realiza la lectura del movimiento de las manos. En este módulo se programa el módulo de seguimiento que se encarga de “leer” la mano mediante diferentes puntos. Con ayuda de las librerías de Computer vision en Python se realiza el módulo de seguimiento que se guía de las uniones en las manos.

**Figura 25**

*Idea de seguimiento de mano.*



*Fuente.* Elaboración propia

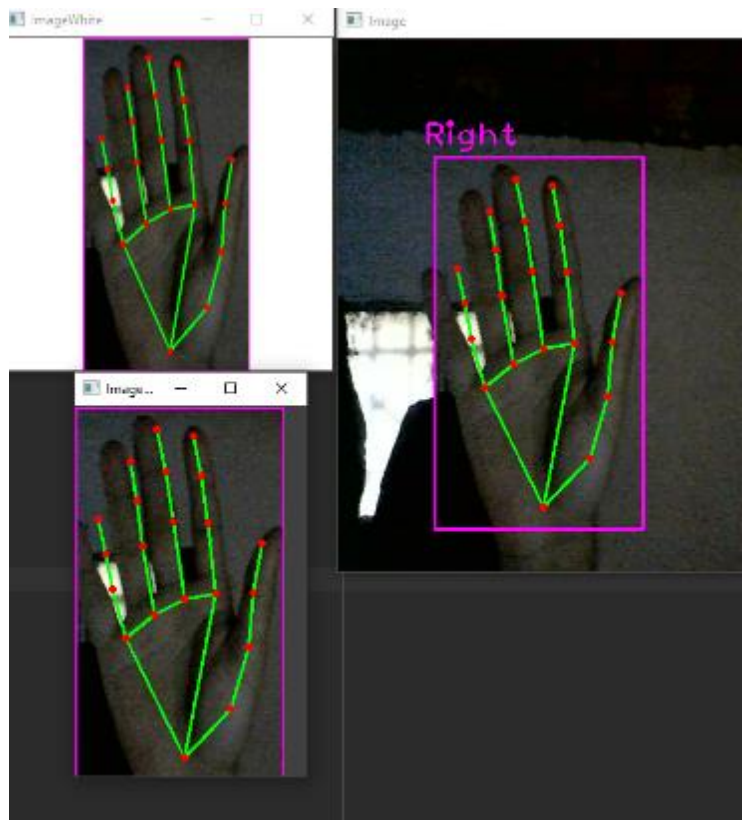
Con esta etapa realizada se busca la manera de realizar una lectura básica de señales no verbales, como existen muchas señas y para poder “leerlas” es necesario entrenar un algoritmo que permita determinarlas, se realiza algo más sencillo que es un contador de dedos, el contador de dedos permite que con un número determinado se realice la comparación y activación de un dispositivo.

### **Captura de Datos y Creación de la Base de Datos**

Para poder crear el modelo de aprendizaje es necesaria una base de datos, esta base de datos se encuentra conformada por una gran cantidad de imágenes, para realizar estas capturas se realiza un pequeño script el cual mientras se presione una tecla ira realizando captura de las imágenes.

**Figura 26**

*Captura de datos.*



*Fuente.* Elaboración propia

El programa funciona de tal manera que mientras se tenga presionada la letra “s” en el teclado, se realizara captura de imágenes en la ventana de ImageWhite.

Esta ventana está programada de tal manera que las imágenes allí tengan una resolución de 300 por 300 pixeles, permitiendo tener un estándar a la hora de capturar la información.

Dentro de la carpeta del proyecto se crean 4 carpetas correspondientes a cada letra en el lenguaje de señas que fue escogido, en estas carpetas se guardan las imágenes tomadas en .jpg,

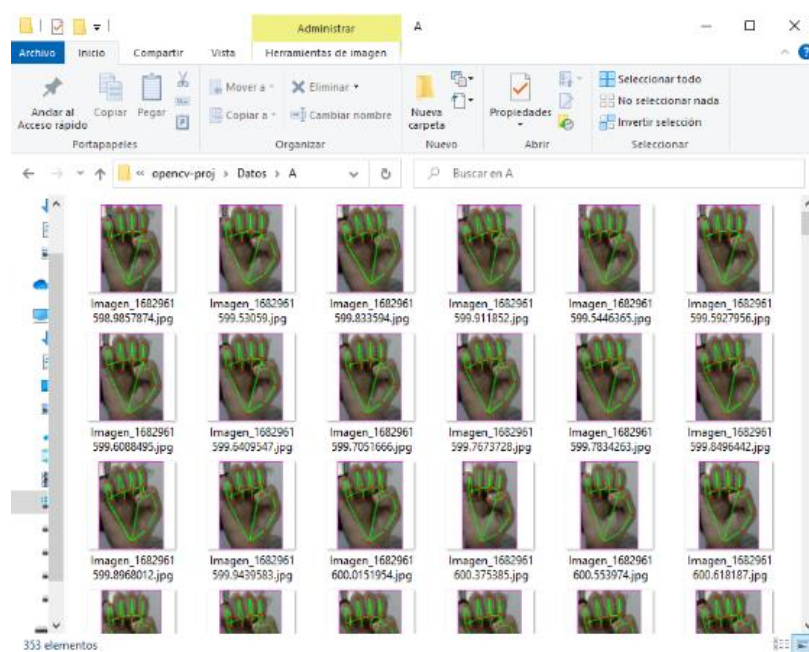
A medida que se presione la letra “s” el contador ira contando y con vc2.imwrite se ira guardando en la carpeta folder y se le asignara la etiqueta de “Imagen\_” seguida de un número

que indica la fecha en la cual fue capturada la imagen, este número es un numero flotante que expresa la fecha y la hora en segundos, todo en UTC.

La carpeta folder a su vez se cambiará en cada captura, es decir Datos/A; Datos/Ab; Datos/B; Datos/H respectivamente.

## Figura 27

*Captura de datos de la seña A.*



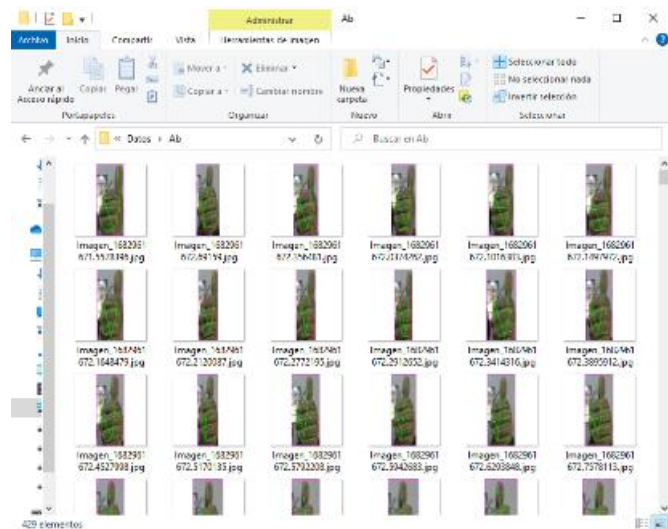
*Fuente.* Elaboración propia

Para la letra A se realiza la captura de 353 imágenes que corresponden a la seña desde diferentes posturas.



**Figura 28**

*Captura de datos de la seña Ab.*

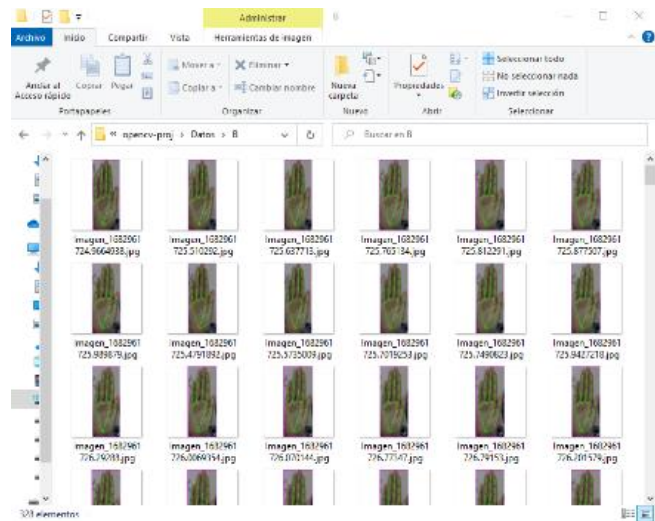


*Fuente. Elaboración propia*

Para la seña Ab se realiza la captura de 429 imágenes que corresponden a la seña desde diferentes posturas.

**Figura 29**

*Captura de datos de la seña B.*

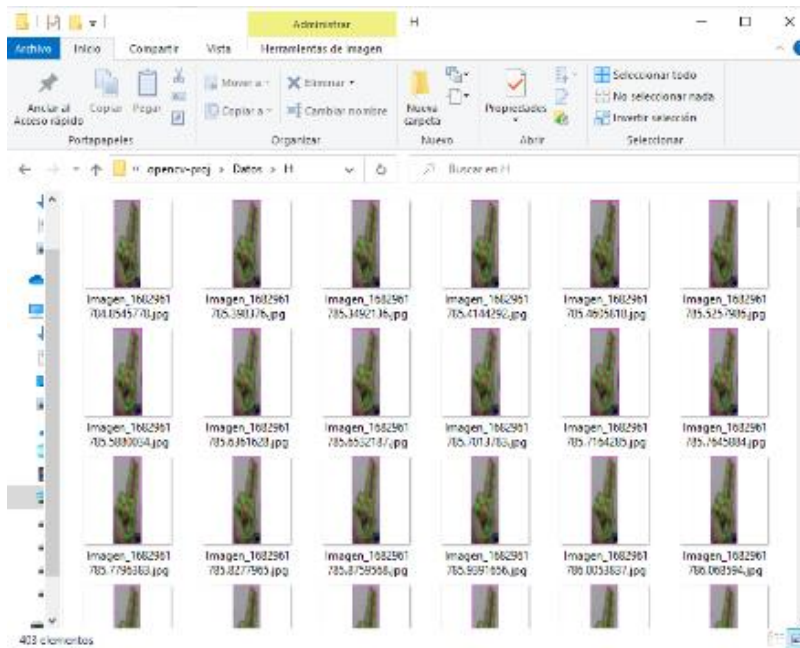


*Fuente. Elaboración propia*

Para la seña B se realiza la captura de 328 imágenes que corresponden a la seña desde diferentes posturas.

### Figura 30

*Captura de datos de la seña H.*



*Fuente.* Elaboración propia.

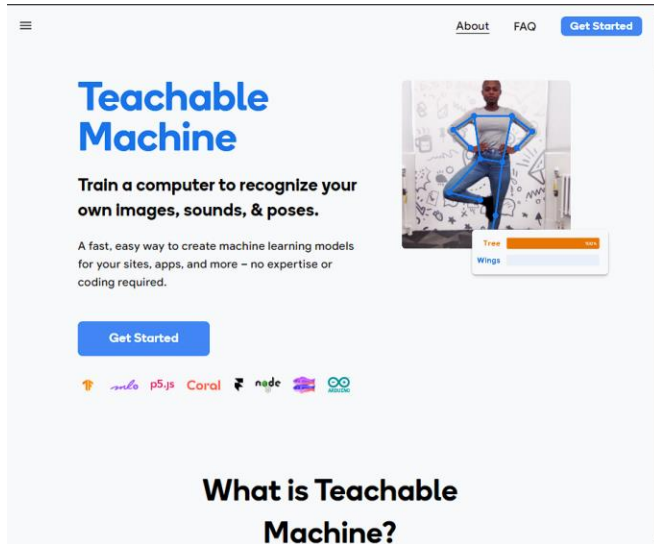
Para la seña H se realiza la captura de 403 imágenes que corresponden a la seña desde diferentes posturas.

Con la fuente de datos realizada, se tienen 1513 imágenes. Se realiza el paso al modelo de aprendizaje con ayuda de Teachable Machine, durante este proceso se tienen que subir las imágenes de la base de datos a la página web, en la página web

<https://teachablemachine.withgoogle.com/>.

## Figura 31

*Máquina de aprendizaje creada por Google.*

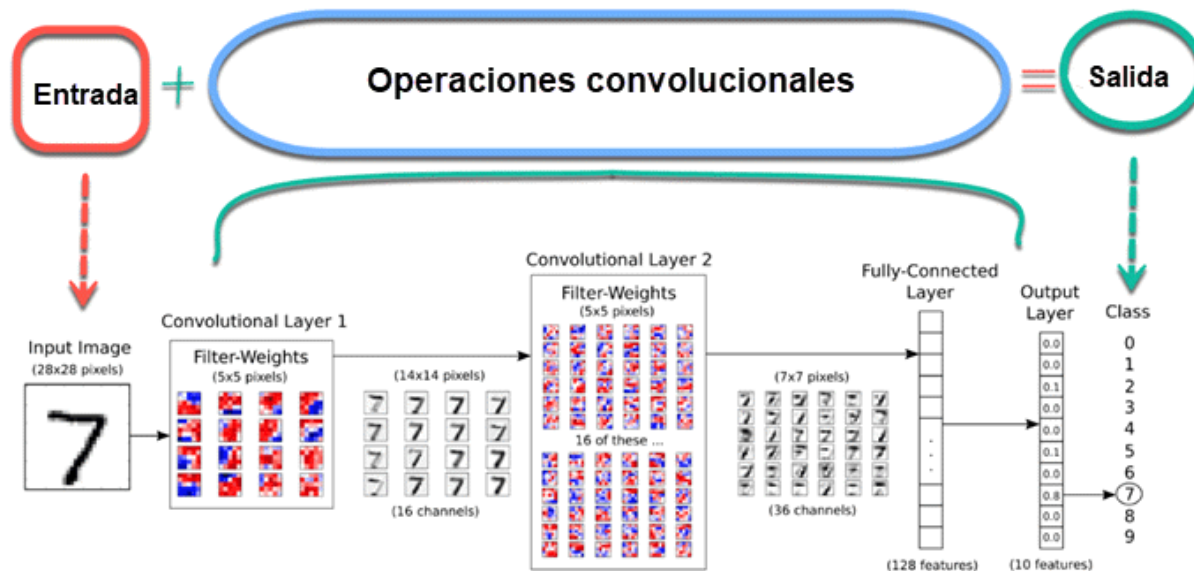


*Nota.* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

Teachable Machine, es un programa construido por Google con base en TensorFlow, el cual es una biblioteca de Deep learning creada por Google, tiene varios usos en el aprendizaje profundo, pero en este caso es de interés su uso en las CNN o Redes Neuronales Convolucionales que es el modelo de aprendizaje usado por Teachable Machine en el procesamiento digital de imágenes, la estructura de una red neuronal convolucional se puede observar en la *figura 32*:

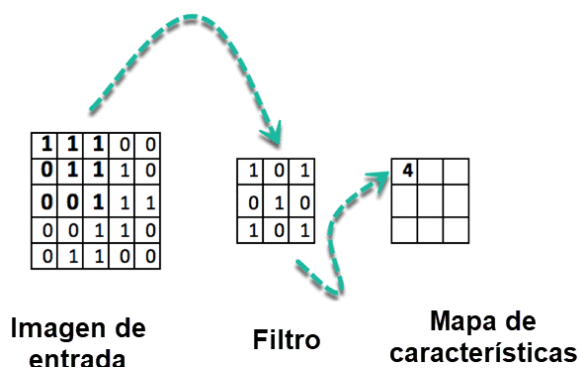
Figura 32

## Red Neuronal Convolutiva



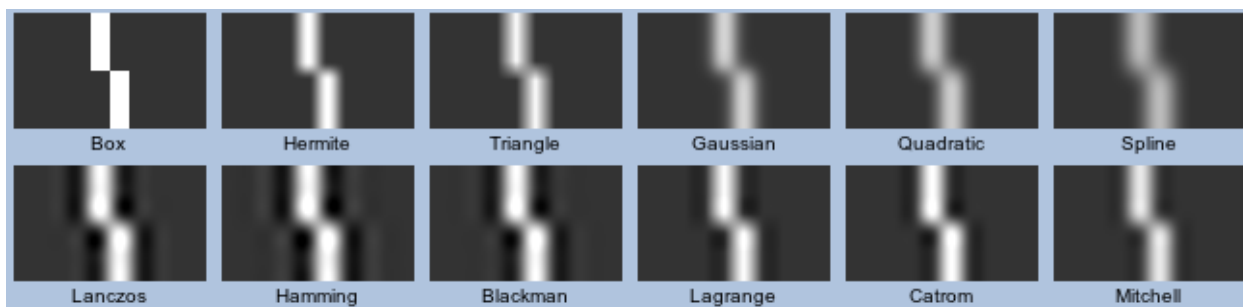
*Nota.* Tomado de Guru99es [Imagen], por Guru99es, 2020, Clasificación de imágenes de TensorFlow.

En una red neuronal convolutiva se tiene entonces la entrada de la imagen la cual es tratada con un preprocesamiento digital obteniendo imágenes de 300x300 píxeles, pasa por una serie de operaciones convolucionales y se obtiene una salida. Las convoluciones son multiplicaciones, se toma una parte de la imagen usualmente 3x3 y se multiplica por un filtro, a continuación, la *figura 33* explica su acción:

**Figura 33***Convolución.*

*Nota.* Tomado de Guru99es [Imagen], por Guru99es, 2020, Clasificación de imágenes de TensorFlow.

Los filtros son usados para conseguir características activadas a la hora de pasar los filtros por zonas de relevancia de una imagen, se buscan sombras, líneas, contornos, aspectos que permitan tener una manera de identificación dentro de la imagen, teachable machine realiza un filtro Lanczos, el cual le da un leve suavizado a la imagen interpolándola suavemente entre sus muestras tal y como se puede observar en la *figura 34*:

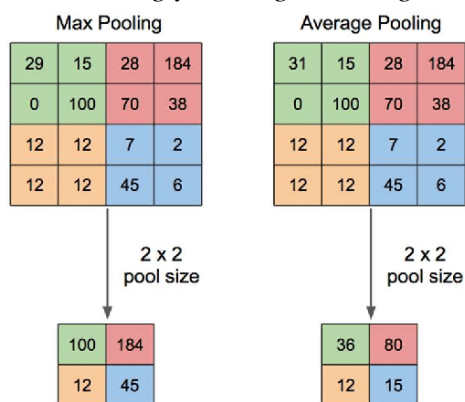
**Figura 34***Filtro Lanczos.*

*Nota.* Tomado de Filter Comparison. [Imagen], por Thyssen A, s.f, Imagemagick.

Posterior a la aplicación del filtro, el sistema creado realiza una normalización de las imágenes con el objetivo de que estas cuenten con la misma escala. Se tienen 3 procesos de importancia dentro de la red neuronal, estos son: el *Pooling*, el cual es un resumen de los datos tomados, es decir se tiene una salida con un pequeño resumen de los valores tomados, en el pooling existen diferentes funciones tales como el *Max Pooling* que da como salida el valor máximo de la región dada, así como otras que permiten el cálculo del promedio o la media, en la *figura 35* se puede observar un ejemplo de Max Pooling y Average Pooling que es el máximo y el promedio respectivamente.

**Figura 35**

*Max Pooling y Average Pooling*



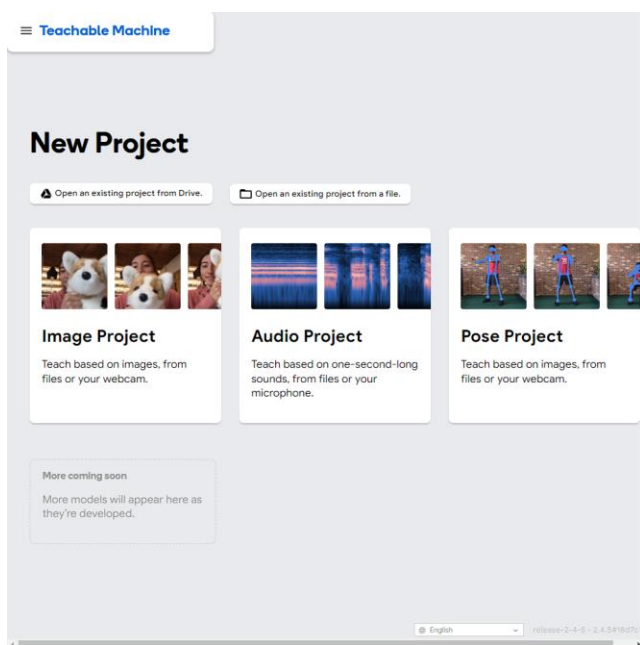
*Nota.* Tomado de Journal of Physics: Conference Series [Imagen], por Yani, et al, 2019, Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail

La siguiente capa es la *capa completamente conectada*, aquí se añaden una o dos capas completamente conectadas, se realiza la multiplicación lineal de un filtro y la salida del Max Pooling, se realiza la suma ponderada de los parámetros por el valor de la entrada y el que tenga mayor puntuación será el de mayor probabilidad, obteniendo como resultado una clasificación que indica la probabilidad de cada clase.

Por último, está la *capa de clasificación*, en esta se genera el vector resultado que da información de la probabilidad de que la imagen de la entrada pertenezca a la clase específica. Todos estos procesos se encuentran realizados de manera automática dentro del sistema que es diseñado por Teachable Machine, lo que permite manejar menos código y facilitar enormemente el proceso de aprendizaje. Prosiguiendo con la creación del sistema en la página de Teachable Machine, no se es necesario iniciar sesión, simplemente se dirige hacia “Get Started” para comenzar con el diseño del modelo de procesamiento de imágenes.:

### Figura 36

*Selección del proyecto de imágenes.*



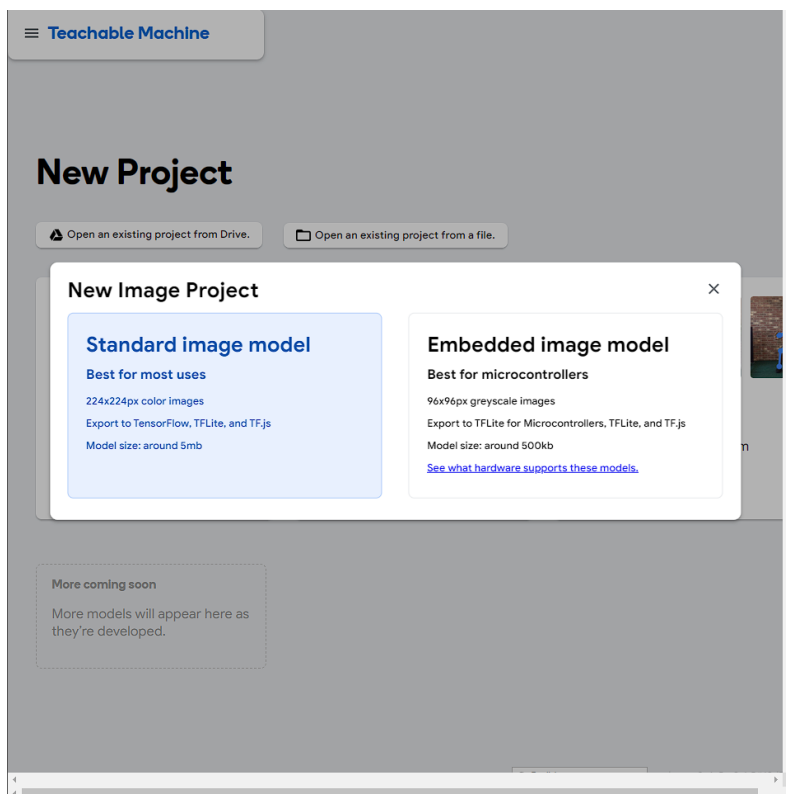
*Nota.* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

Se selecciona el proyecto de imágenes y se elige un modelo de imágenes estándar, pues será ejecutado en el ordenador, el propio software indica las diferencias entre “Standard” y “Embedded” que es básicamente su tamaño, resolución, al trabajar en un ordenador, se permiten

imágenes de mayor tamaño, es decir en “Estándar”, mientras que para trabajar con un microcontrolador recomienda el uso de imágenes “Embedded”

### Figura 37

*Selección del modelo de imágenes estándar.*



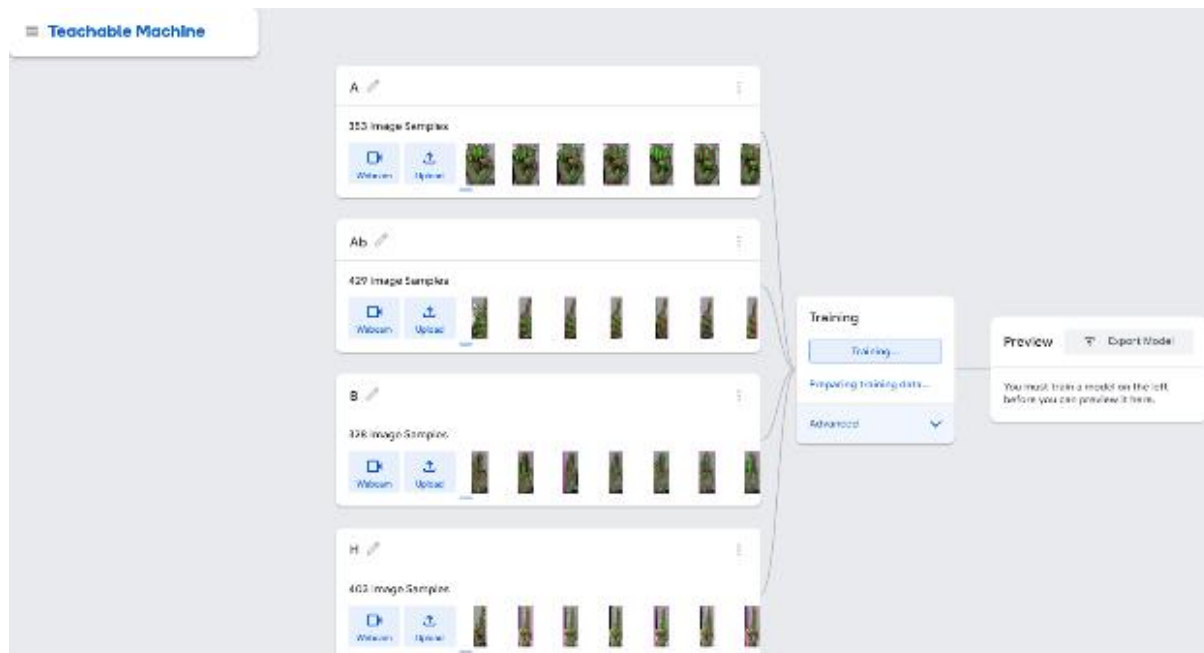
*Nota.* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

Posteriormente se realizará la selección de las 4 clases, cada clase corresponde a una señal y se subirán las imágenes correspondientes a cada una, para luego empezar a entrenar el modelo dando click en “Train Model”, el modelo estándar utiliza imágenes a color, por lo que la profundidad de los filtros será de 3 debido a los tres colores “RGB” que corresponden a Red, Green y Blue respectivamente, a su vez este realiza un preprocesamiento al ingresar las imágenes asegurándose de las imágenes tengan por lo menos una resolución de 224x224.



**Figura 38**

*Selección de clases e inicio de entrenamiento del modelo.*



*Nota:* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

Luego de que el modelo se encuentre entrenado, se realiza la exportación a Tensorflow para manipularlo mediante Keras. Teachable Machine permite manipular tres variables que son importantes conocer:

Épocas, se trata de un número que indica la cantidad de veces que se va a repetir la alimentación del entrenamiento, teniendo en cuenta que las imágenes ya fueron pasadas, un número mayor de épocas permite obtener mejores resultados a la hora de utilizar las predicciones.

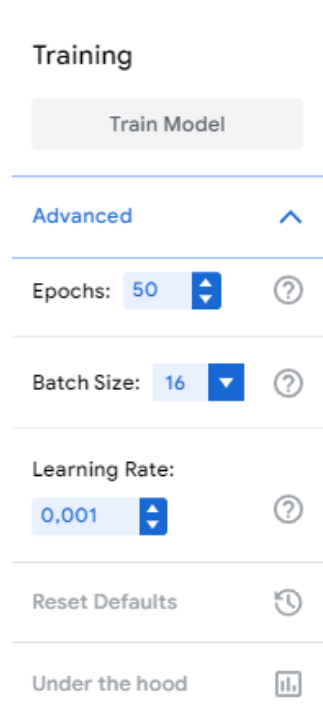
Tamaño de lote, este parámetro indica el conjunto de muestras que se utilizan en la iteración, es decir permite ordenar las imágenes en una cantidad que es especificada, por ejemplo, si se tuvieran 20 imágenes y se le indicara un valor 4, se tendrían 5 lotes y cada vez que se pasen los 5 lotes se cumple una época.

Tasa de aprendizaje, es un valor que da a conocer el cambio del modelo con base al error que se estime cada vez que se actualicen los pesos del modelo, mientras más pequeño, más retardado es el aprendizaje y mientras más grande, más rápido será el aprendizaje.

Los parámetros utilizados en el desarrollo de la red convolucional en Teachable machine se pueden apreciar en la *figura 39*:

### Figura 39

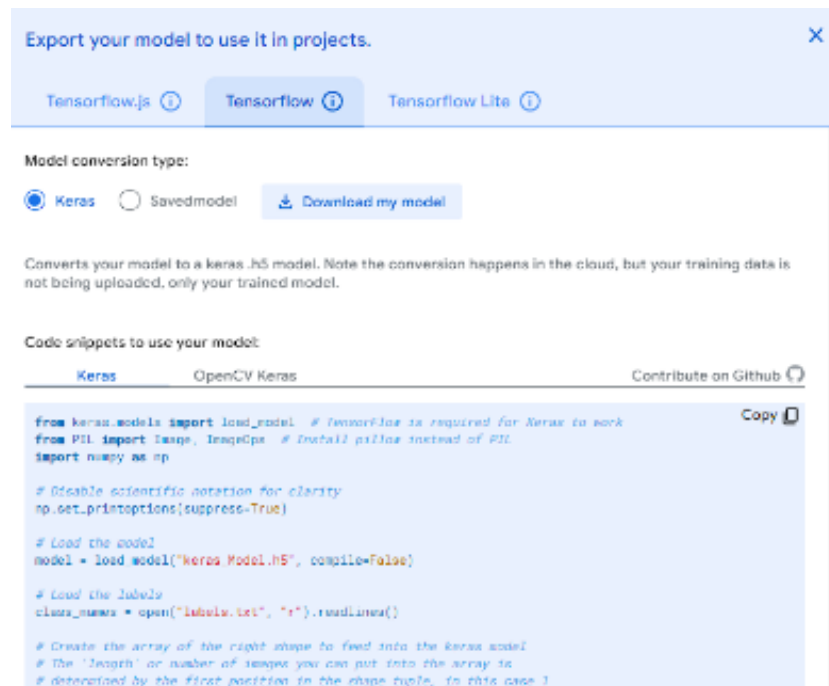
*Parámetros avanzados del entrenamiento.*



*Nota.* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

**Figura 40**

*Teachable Machine modelo entrenado.*



*Nota.* Tomado de Teachable Machine [Imagen], por Google, s.f, *Teachable Machine*.

Con el modelo ya en el formato de Keras, se descarga y se añade a una carpeta llamada “Model” en la misma carpeta que creada por el compilador y se trabaja con él desde el entorno de desarrollo y ayuda de la librería de Tensorflow en PyCharm, en PyCharm se añadirá el módulo de clasificación.

La importación del módulo de clasificación se realizó mediante: (*from cvzone.ClassificationModule import Classifier*) y se seleccionó el modelo descargado como clasificador, adicional es necesaria la selección de los “labels”, los cuales son la referencia que el clasificador tendrá en cuenta, luego mediante un comando If para cuando se detecta la mano y se realiza el seguimiento de la misma (una mano), se realizara el mismo proceso de captura de

imagen que en la captura realizada en la base de datos, para luego con ayuda del clasificador se realice la predicción del index al que hace referencia la imagen capturada en el momento.

De esta manera se tendrá el sistema que predecirá que label es a la que hace referencia la imagen capturada en tiempo real y con ayuda del comando print, se mostrara en la consola del 0 al 3 cual seña se está realizando, numero correspondiente a la ubicación de la letra/seña en el vector de labels = ["A", "Ab", "B", "H"] donde 0 será "A" y "H" será 3 respectivamente, para que el proceso sea más intuitivo y no sea necesaria la visualización de la consola, se creara un pequeño texto en el cual se representara la letra a la seña que se está realizando en pantalla Como se aprecia en la *figura 41* a continuación:

#### **Figura 41**

*Texto representante de la seña.*



*Fuente.* Elaboración propia

De esta manera, el comando `cv2.putText(ImgOutpup,labels[index]...` indicara en pantalla la seña que se está realizando con base en la predicción del index que el software haga en el momento en color fucsia como se puede apreciar en la *figura 41*, con el sistema de predicción funcionando se realiza el envío de datos por medio de bróker con la secuencia lógica siguiente:

**Tabla 2**

*Secuencia lógica de envío de datos por MQTT*

Índex	Numero enviado	Letra
0	3	A
1	2	Ab
2	1	B
3	4	H

*Fuente.* Elaboración propia

Si el índice = ["A", "Ab", "B", "H"] es 0, es decir corresponde a la seña "A" se envía mediante el cliente "sagaitanh" el número 3, si es 1, seña "Ab" se envía un 2 y así respectivamente hasta el índice 3, seña "H" donde se envía un 4.

### **Programación del Microcontrolador**

Se eligió una tarjeta de desarrollo a programar. La ESP32-Cam permite abastecer todo lo necesario para este proyecto, cuenta con Wifi, bluetooth y algunas salidas/entradas que si bien son pocas, cumple con lo que se requiere hacer que es activar dos salidas, para su programación se utilizara el software de Arduino IDE el cual es gratuito y tiene miles de librerías para tarjetas ESP32, su programación consiste en crear una cámara web que permita acceder mediante una IP y poder usarla en Python, este proceso es llevado a cabo con ayuda de las librerías de WebServer.h, Wifi.h y esp32cam.h para lo cual es necesario wifi para conectarse a un punto de

acceso, un Router y tener acceso a internet, esto se realiza con ayuda de la librería Wifi.h, luego de realizar esto, se realizara la conexión a el servidor MQTT para lo cual se utiliza la librería PubSubClient.h y se realiza la secuencia lógica que realiza la activación o desactivación de las dos salidas, el código implementado en el microcontrolador se encuentra en el anexo 2.

La Raspberry pi representa muchas mejoras en comparación a la ESP32-CAM, esto debido a su gran capacidad de almacenamiento, su cantidad de puertos de entrada y salida, así como su potencial a la hora de tener una cámara integrada de una buena definición su principal diferencia a la hora de manejar la Raspberry pi, es que es un miniordenador capaz de reconocer el lenguaje de Python por si sola, esto representa un cambio radical en el proyecto debido a que etapas como la de comunicación no sería necesaria para una instancia local pues esta misma tarjeta realizaría el procesamiento digital de imágenes y su clasificación, así como la activación de las etapas de potencia que se desee o se logren acoplar en el espacio de la tarjeta, por último, una de las diferencias es la conexión de la cámara, esto debido a que no se encuentra por si sola como en la ESP32 CAM, a continuación se observa la cámara y su conexión a la Raspberry Pi:

#### **Figura 42**

*Conexión de cámara en Raspberry pi.*



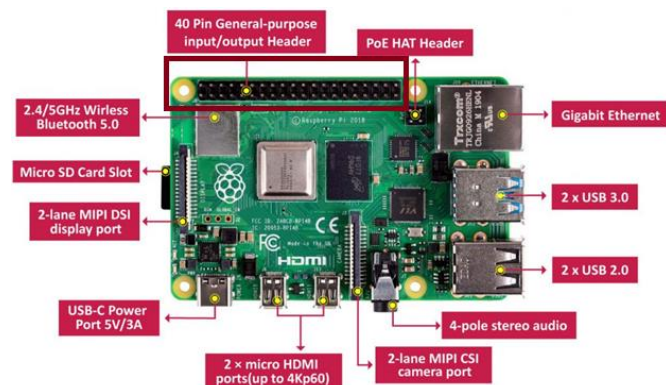
*Nota.* Tomado de Raspberry Pi Camera Module - Operating System notice [Imagen], por Raspberry Pi Foundation, s.f, *projects.raspberrypi*.

La cámara representa una mejora con respecto a la ESP32 CAM debido a que no es necesaria la aplicación de una cámara web y por lo tanto el Ping, el retraso debido a la conexión vía Wifi no estará presente, la cámara de la ESP32 es de tan solo 2 megapíxeles, mientras que la Raspberry cuenta con una cámara 1080p de 5 megapíxeles lo cual permite una ampliación del rango de reconocimiento que en la ESP32 CAM se encuentra cerca de un metro.

Por otra parte, la Raspberry en su última versión, las Raspberry pi 4 cuenta con 40 pines para uso general, como se puede apreciar a continuación:

### Figura 43

*Raspberry pi 4 Pines.*



*Nota.* Tomado de Raspberry Pi 4 GPIO Pinout [Imagen], por AliOs ,2019, *keytosmart*.

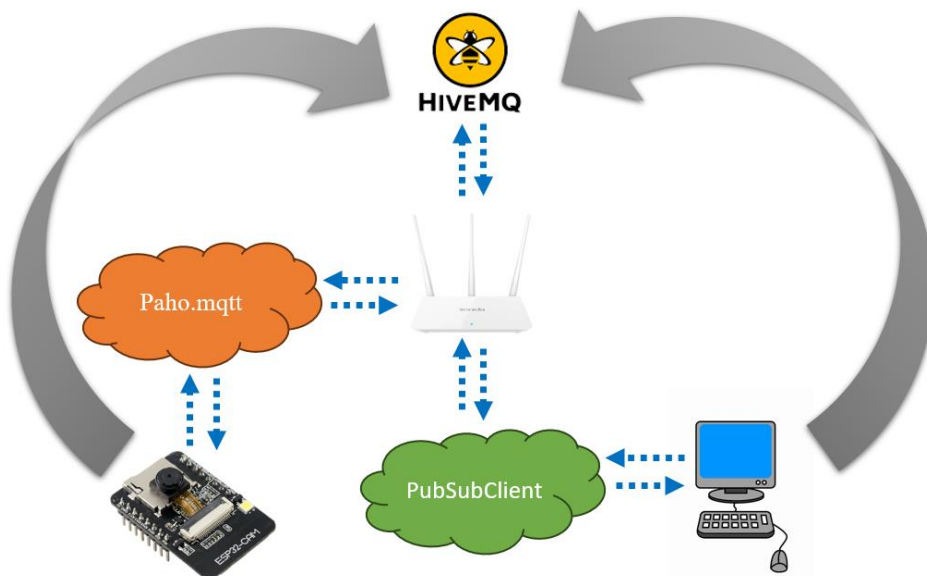
Esto representa una mejora a la hora de manejar diversos dispositivos, ya que se puede realizar mediante estos propios pines o bien, ya que cuenta con un chip Wireless integrado, se puede realizar el control mediante Bluetooth o Wifi con otro modulo, de esta manera configurado como punto de acceso se puede realizar la conexión machine to machine para tener un hud central mediante el cual se puedan controlar dispositivos a grandes distancias sin necesidad de cables extensos.

## Etapa de Comunicación

Como se trabaja con una parte de software que es realizada por un ordenador y otra realizada por un circuito, es necesaria una comunicación entre estos y para ello se realiza uso del protocolo MQTT (MQ Telemetry Transport) anteriormente conocido como Message Queuing Telemetry Transport. Este protocolo se encuentra basado en la pila TCP/IP por lo cual es sencillo de manejar desde un ordenador, para lo cual se utilizó el bróker de HiveMQ el cual provee un servicio sencillo para el envío de mensajes cortos, se utilizan librerías en Arduino IDE de PubSubClient, y para Python, se utiliza paho.mqtt, este se puede observar en la *figura 44*:

### Figura 44

*Sistema de comunicación.*



*Fuente.* Elaboración propia

En el primer parte, se tiene la importación de la librería y la conexión al bróker, así como su selección, mientras que en la segunda parte se tiene el envío de datos por medio de este teniendo la secuencia lógica mencionada anteriormente en la etapa de software, en la *figura 44* se



observa como el intermediario es el router que se encarga de funcionar como mediador para las conexiones entre el ordenador, la tarjeta de desarrollo y el bróker MQTT, por otra parte, el Código implementado en Arduino para el sistema de comunicación se encuentra en el [anexo 2](#).

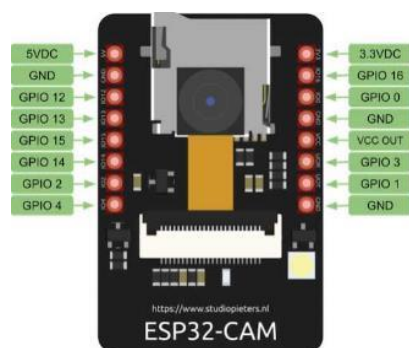
En la programación del sistema de comunicación se tiene, la inclusión de la librería y su respectiva conexión, así como el indicativo de que el dispositivo se encuentra conectando por medio del puerto serial de Arduino, luego se tiene la función “mqttReceive” la cual se encarga de acoplar los datos recibidos en una variable “string” y por último la lectura de esa variable así como, la secuencia lógica que se encarga de activar o desactivar los dos dispositivos, el relé y el TRIAC mediante la activación/desactivación de sus salidas.

### Etapa de Control

La etapa de control, como se decía anteriormente está dada por la tarjeta de desarrollo ESP32-cam la cual cuenta con una cámara de 2 megapíxeles, modulo wifi 802.11b/g/n/e/i, bluetooth 4.2 con BLE, su código de programación se encuentra en [el anexo 2, código 1](#).

### Figura 45

*Esquema ESP32-CAM.*



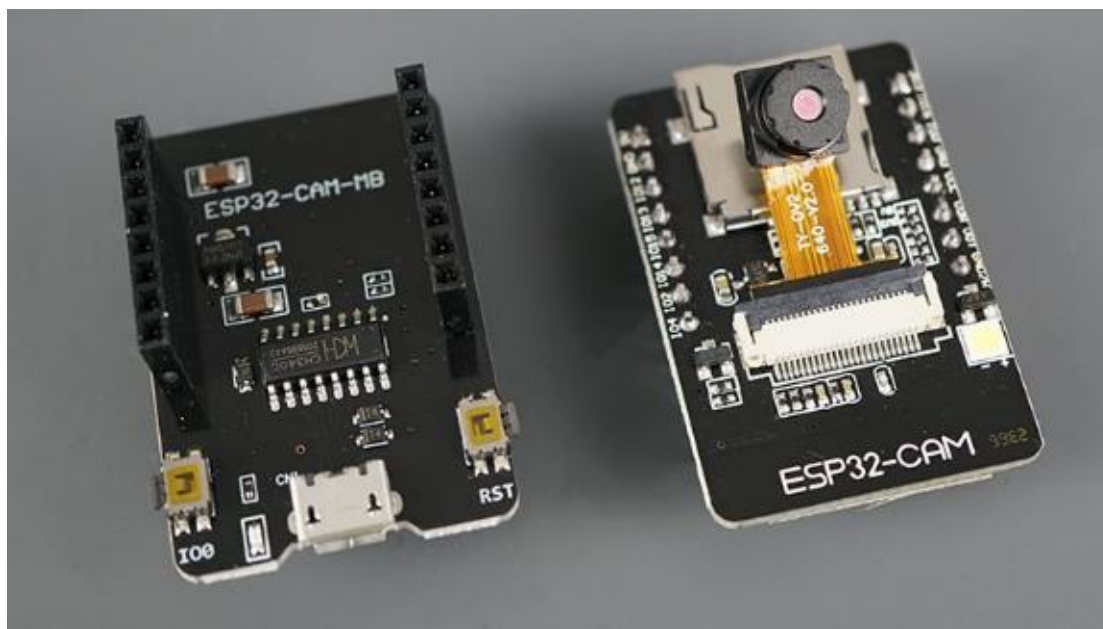
*Nota.* Tomado de ESP32 CAM introducción y primeros pasos [Imagen], por Charly Pascual, 2022, *programarfacil*.

Como se observa en la anterior figura cuenta con 10 pines GPIO (General Purpose Input Output). Para su programación se utiliza la placa ESP32-CAM-MB, la cual permite usar un cable micro USB para su conexión. Esta controlara un relé de 3v que viene en una tarjeta de 1 canal, por lo cual no es necesaria la protección para el voltaje de desconexión, pues esta trae un circuito interno que ya la realiza.

Para su programación se utilizará el programador ESP32-CAM-MB el cual cuenta con un chip integrado, el CH340C que facilita su manejo ya que funciona como una Shield en la cual se monta la tarjeta de desarrollo. Este programador se puede apreciar a continuación:

#### **Figura 46**

*ESP32-CAM y programador.*



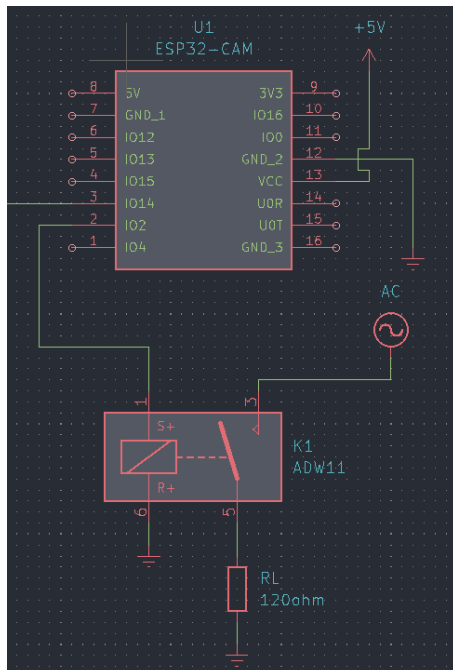
*Nota.* Tomado de *Upload Code to ESP32-CAM Using ESP32-CAM-MB USB* [Imagen], 2021, randomnerdtutorials.

Como se observa en la anterior imagen se tiene una conexión por medio de un cable micro USB.

**Figura 47***Relé.**Fuente.* Elaboración propia

El relé realizará la activación de una etapa de potencia, que será una bombilla. Siendo controlada por la cámara de la ESP32.

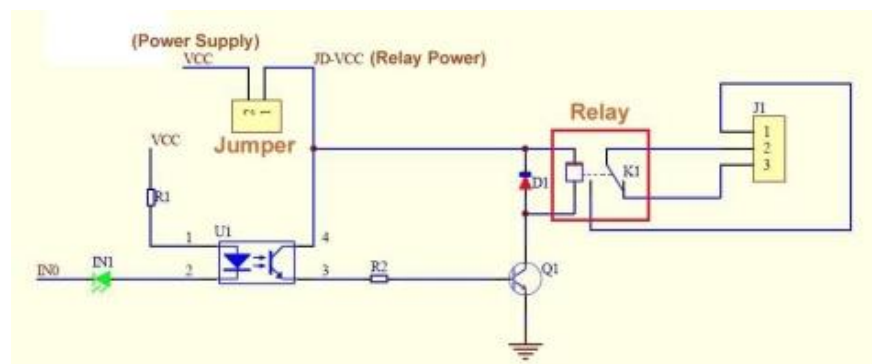
## Planos

**Figura 48***Esquema de conexión relé.**Fuente.* Elaboración propia

El esquema corresponde a una conexión sencilla en la cual la ESP32-CAM realiza la activación de un relé de 3v. En la etapa de potencia se encuentra conectada una resistencia de 120 Ohm que bien puede ser variable ya que es un ejemplo de una bombilla incandescente.

### Figura 49

*Diagrama eléctrico de relé de 1 canal.*



*Nota.* Tomado de *MÓDULO DE RELÉS* [Imagen], 2012, por REGATA.

El funcionamiento del módulo de relé está diseñado para funcionar con un pulso de señal que generalmente suele ser a partir de una amplitud de 3 voltios. Esta amplitud se encuentra dada por el U1 el cual es un opto transistor que permite aislar zonas, una zona fría en la cual se tiene poco voltaje y una zona caliente donde se tiene la corriente alterna, alrededor de 120 voltios RMS. El opto transistor permite dejar pasar la corriente y activar el transistor Q1 el cual otorga la energía necesaria para la activación del electroimán del relé. Debido a que el relé produce una chispa debido a la fuerza contra electromotriz de miles de voltios cuando se desconecta, se coloca un diodo D1 para que consuma esta energía y proteja junto al transistor el circuito de control.

Otro esquema de conexión se trata del uso de un tiristor para realizar la activación de la etapa de potencia mediante un impulso eléctrico. Los componentes que usar son: un Opto-

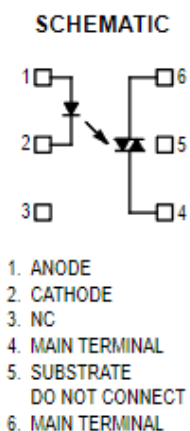


controlarse, pues no son filtros obligatorios y en cargas pequeñas como las de una bombilla led común, cuya potencia es de alrededor de 10 y 25w dependiendo de la propia bombilla.

El circuito del MOC3021 interno es el siguiente:

### Figura 51

*Diagrama de conexión MOC3021.*

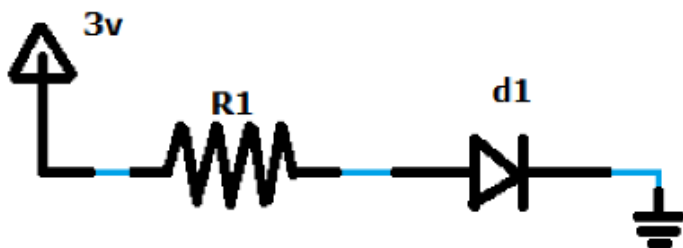


*Nota.* Tomado de hoja de datos de MOC 3021, 3022, 3023 Series [Imagen], por datasheet, s.f, *ALLDATASHEET*.

Se puede apreciar que las terminales 1 y 2 se encuentran aisladas de la 6 y 4, mientras que las 3 y 4 no se usan. El circuito se encuentra unido por el diodo que envía la señal en forma de luz y un fotorreceptor es el encargado de activar el TRIAC uniendo las terminales 6 y 4. Como se está manejando un diodo, este tiene un consumo en potencia. Para determinar la resistencia a colocar se analiza el siguiente circuito:

**Figura 52**

*Cálculo de resistencia.*



*Fuente.* Elaboración propia

$$V(\text{fuente}) = V(\text{led}) + V(\text{Resistencia}) \quad (1)$$

Se debe tener en cuenta los valores dados por el comerciante para que se pueda superar la barrera de potencial del diodo. De acuerdo con el Datasheet del anunciante:

**Figura 53**

*Datasheet MOC3021 valores de Led 1.*

**INPUT LED**

Reverse Leakage Current ( $V_R = 3\text{ V}$ )	$I_R$	—	0.05	100	$\mu\text{A}$
Forward Voltage ( $I_F = 10\text{ mA}$ )	$V_F$	—	1.15	1.5	Volts

*Nota.* Tomado de hoja de datos de MOC 3021, 3022, 3023 Series [Imagen], por datasheet, s.f,

*ALLDATASHEET.*

El voltaje mínimo es entonces a partir de los 1.15 voltios para que la corriente pueda circular a través del diodo. Por otra parte, es necesaria la corriente a trabajar para poder calcular la resistencia:

**Figura 54**

*Datasheet MOC3021 valores de Led 2.*

LED Trigger Current, Current Required to Latch Output (Main Terminal Voltage = 3 V(3))	<u>MOC3021</u> MOC3022 MOC3023	$I_{FT}$	—	<u>8</u> — —	<u>15</u> 10 5	mA
Holding Current, Either Direction		$I_H$	—	100	—	$\infty$ A

*Nota.* Tomado de hoja de datos de MOC 3021, 3022, 3023 Series [Imagen], por datasheet, s.f, *ALLDATASHEET*.

El valor necesario para la activación del led oscila entre los 8 y 15 mA. Por lo que si se supone una corriente de 10mA es suficiente.

$$3v = 1,5v + V(\text{Resistencia})$$

Mediante la aplicación de la ley de ohm se tiene la siguiente resistencia (El valor de la resistencia, así como el valor de corriente no tienen que ser exactos pero un cálculo cercano permite tener una mejor calidad de vida en el circuito).

$$V = I * R ; I = \frac{V}{R} ; R = \frac{V}{I} \quad (2)$$

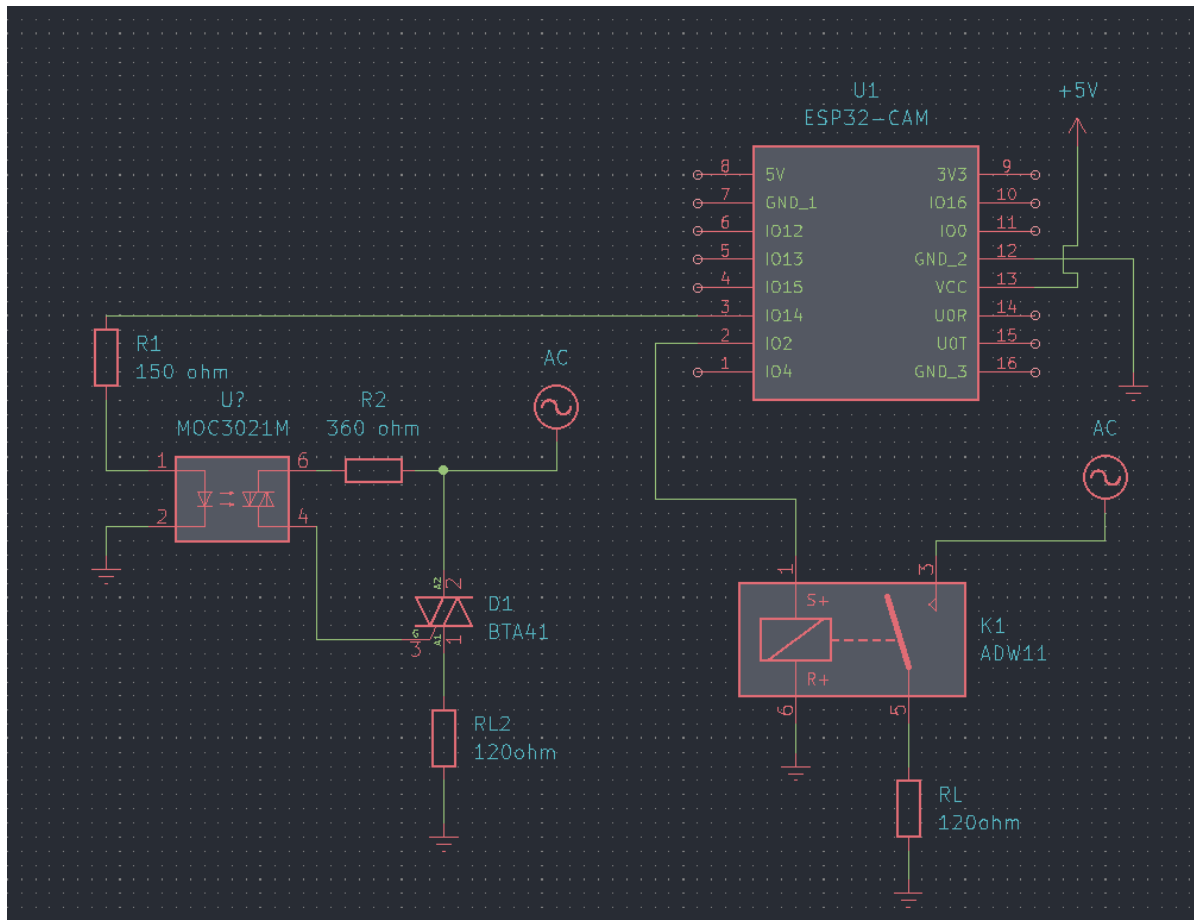
$$R = \frac{3 - 1,5}{0,010} = 150 \text{ Ohm} \quad (3)$$

Se tiene entonces el siguiente circuito resultante teniendo en cuenta los cálculos realizados:



**Figura 55**

*Diagrama de conexión general.*



*Fuente.* Elaboración propia.

Se puede observar que el circuito se encuentra diseñado con un TRIAC de BTA41 el cual es un tiristor de potencia, según su hoja de datos:

**Figura 56**

*Datasheet BTA 41.*

**Table 1: Main Features**

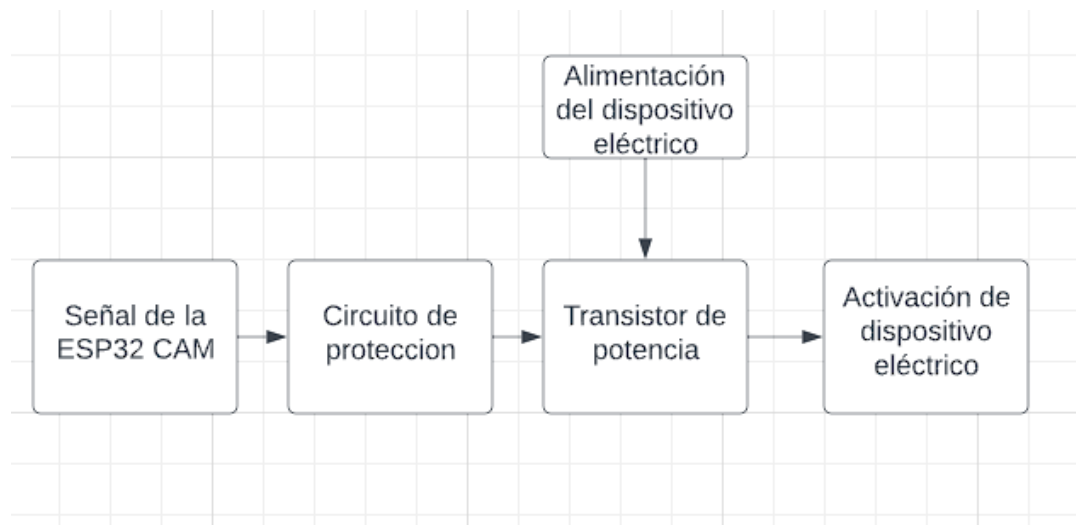
Symbol	Value	Unit
$I_{T(RMS)}$	40	A
$V_{DRM}/V_{RRM}$	600 and 800	V
$I_{GT}(Q_1)$	50	mA

*Nota.* Tomado de hoja de datos de BTA40, BTA41 and BTB41 Series [Imagen], por datasheet, s.f, *ALLDATASHEET*.

Sus valores aproximados son hasta 40 amperios y de 600 Voltios en picos que se suelen repetir a 800 voltios en picos de voltaje reverso. Es un dispositivo para una cantidad considerable de potencia y su potencia no es aprovechada en el ejemplo anterior. Esta etapa se encuentra diseñada de tal manera que se puede modificar según las cargas para ahorrar en componentes y tener un circuito fiable. Pero en práctica, siempre existirá un componente de potencia utilizado para la activación de cargas en corriente alterna, las diferencias serán más notorias cuando se utilicen cargas más pesadas, pues se requieren circuitos de protección como el ya mencionado filtro snubber. Entonces para el diseño de una etapa de potencia bien se puede utilizar el circuito anterior con este filtro en muchos casos, estos limitados por la potencia que es capaz de manejar el BTA41 que en su máximo es de 40 amperios en RMS. El diseño de esta etapa siempre seguirá el siguiente diagrama de flujo:

**Figura 57**

*Diagrama de flujo etapa de potencia.*



*Fuente.* Elaboración propia

## **Implementación Física**

### *Algoritmos*

### *Código en Python*

El primer algoritmo corresponde a el desarrollo del seguimiento de la mano, este se encarga de (como se mencionó anteriormente) realizar un seguimiento a la mano en diferentes articulaciones realizando un muestreo de cada una de ellas, así como la clasificación de las diferentes señas y configuración del cliente MQTT. El código se encuentra en el [anexo 1, código 1](#). El código dos de Python corresponde al módulo de seguimiento [anexo 1, código 2](#) y por último el código tres corresponde al algoritmo que se realizó para la captura de datos [anexo 1, código 3](#).

***Arduino IDE (esp32-cam):***

El código desarrollado para la placa corresponde a el encargado de realizar uso del sensor de la cámara para recibirlos en una ventana de Python, para ello se crea una cámara web en un host local. En este código se realiza la activación de la salida para el control de una etapa de potencia mediante un relé. Este código se puede ubicar en el anexo 2, código 1.

### **Explicación General**

La solución se divide en dos partes importantes, la primera es la construcción de software y posteriormente la segunda es la elaboración del hardware.

#### ***Construcción del Algoritmo***

El algoritmo a su vez se divide en dos etapas, la primera constituye la programación de la ESP32-CAM. En la tarjeta de desarrollo se realiza un código en el cual se crea una cámara web. Para ello se realiza uso de las librerías de WebServer las cuales permiten crear un servidor web. Con el servidor web, se utiliza las librerías de la ESP32-CAM en Arduino para simplificar el proceso de captura de imagen y envío, pues este consta de alrededor de 4 subcódigos que permiten el funcionamiento y activación de la cámara de la tarjeta de desarrollo para lo cual también hace falta usar las librerías de Wifi para realizar la conexión al Router, el cual hará de hud para permitir su acceso a cualquier dispositivo conectado allí.

Con la cámara en funcionamiento se inicia el monitor serial para 115000 baudios en el cual se mostrará la dirección IP que ha obtenido la cámara, así como su correcta conexión al Router.

Con esto queda la cámara web. Posteriormente se realiza la programación en donde se conecta el MQTT Bróker, el cual utiliza librerías para simplificar el proceso, en el código se indica el nombre del bróker, el servidor y posterior a esto se realiza la configuración del envío de datos (cuando este conecta) se modifica el tipo de variable y con esto se realiza la etapa final de

la ESP32-CAM la cual consiste en un simple condicional que realiza la activación o desactivación del relé que se controlara en el pin 2 el cual permite la escritura de 3.3v en alto.

Con el algoritmo de la tarjeta de desarrollo se realiza entonces el módulo de seguimiento de las manos, el cual como se mostró anteriormente (en la figura 10) realizará una captura de las articulaciones y las unirá con líneas. Este código realiza uso de las librerías de Computer Vision lo que permite utilizar funciones que simplifican parte del proceso de reconocimiento. Computer vision requiere de indicar la captura con la cual se estará trabajando, para lo cual se realiza uso de la dirección IP dada por la ESP32-CAM. A su vez, es necesario el uso de las librerías de Numpy para utilizar la captura en forma de una matriz y trabajar con esta a la hora de realizar el reconocimiento de los dedos.

Con el módulo de seguimiento en funcionamiento, se utiliza en el código principal para realizar el clasificador. Con este clasificador se realiza la comparación de las imágenes que se procesan mediante la cámara web y mediante un indicador se establece si pertenece a la letra 0 correspondiente a la “A”, la letra 1 correspondiente a “Ab”, la letra 2 correspondiente a “B” y la letra 3 que corresponde a “H”, con estos números, se envía el dato (el numero) mediante el MQTT Troker, para esto se realiza uso de las librerías de Pubsub, las cuales permiten configurar el bróker de manera más compacta. Con esto mediante una función Case en Arduino Idle se realiza la comparación y se crea la secuencia lógica de activación y desactivación de las salidas, en caso de la seña “A” se activa el relé, en caso de la seña “Ab” se activa el TRIAC, en caso de la seña “B” se desactiva el relé y por último en caso de la seña “H” se desactiva la salida que lleva a la activación del TRIAC.

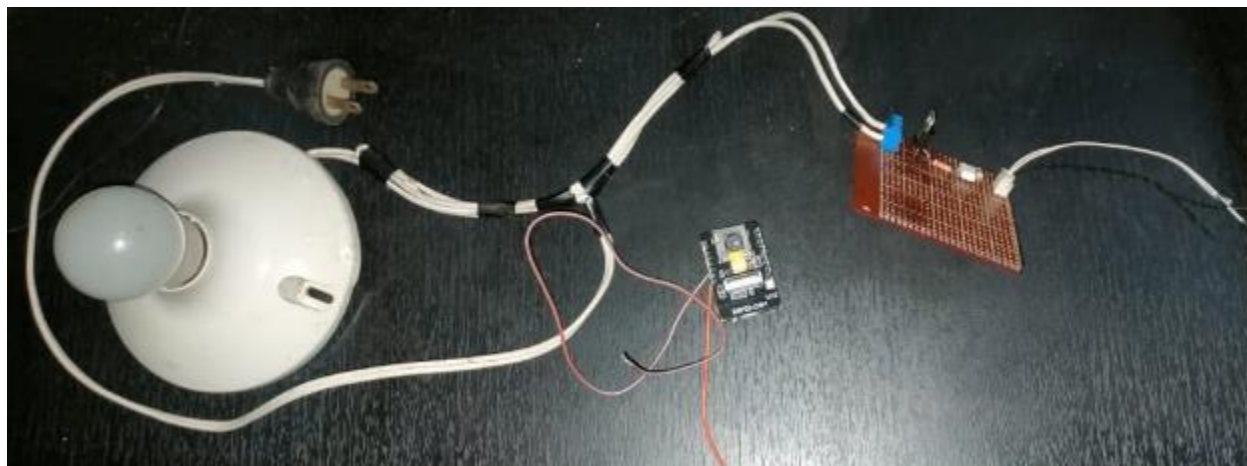
Con esto se desarrolla entonces el primer parte, la cual une los dos códigos y permite el funcionamiento del primer componente de la parte de hardware, que es la ESP32-CAM.

## Construcción del Circuito

En la parte de Hardware se tiene una construcción sencilla, la ESP32-CAM va conectada a alimentación. Esta a su vez, en el pin (2) realiza la activación de una salida digital de 3.3v en estado alto y cerca de 0 en bajo. Este pin fue programado en la primera etapa, por lo que aquí simplemente se conecta al relé de 1 canal. El relé ira conectado a su vez en una parte al neutro de la línea del hogar y en la otra se tendrá la fase con la carga (una bombilla). Como se tiene otro dispositivo, esta ira conectada directamente a la tarjeta de desarrollo y a la entrada del controlador del TRIAC (el MOC3021).

### Figura 58

*Circuito implementación física.*



*Fuente.* Elaboración propia

Para la etapa de potencia, se tiene un fajón conectado en serie junto con un conector para un enchufe. El circuito termina con dos cables, uno con la fase de corriente alterna y otro con la carga junto con el neutro. A continuación, se tiene una figura que lo simplifica:

**Figura 59**

*Esquema conexión de bombilla.*



*Fuente. Elaboración propia*

De esta manera simplemente se requiere cerrar el circuito para que la bombilla comience a funcionar, simplificando las conexiones en la electrónica.

Posterior a esto se tienen los circuitos de control, aquellos que permiten el control de componentes eléctricos en alterna.

**Figura 60**

*Modulo Relé de 1 canal.*

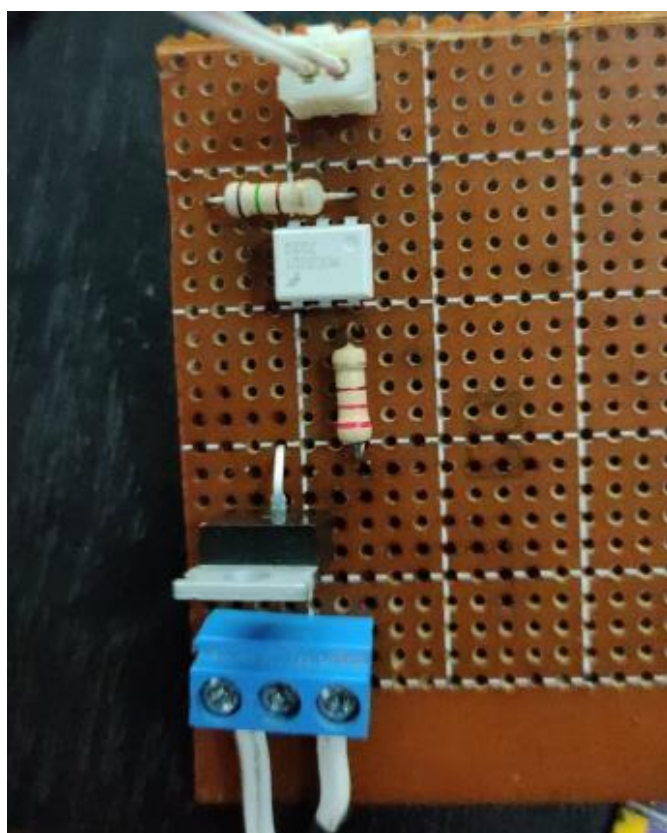


*Fuente. Elaboración propia*

El módulo de relé de un canal corresponde a un relé que tiene una configuración electrónica que permite la activación de un interruptor que une dos puntos, estos dos puntos corresponden a un cable con la fase de la corriente alterna y otro con la carga y el neutro como se observa en la *figura 61*.

### **Figura 61**

*Módulo de control mediante TRIAC.*



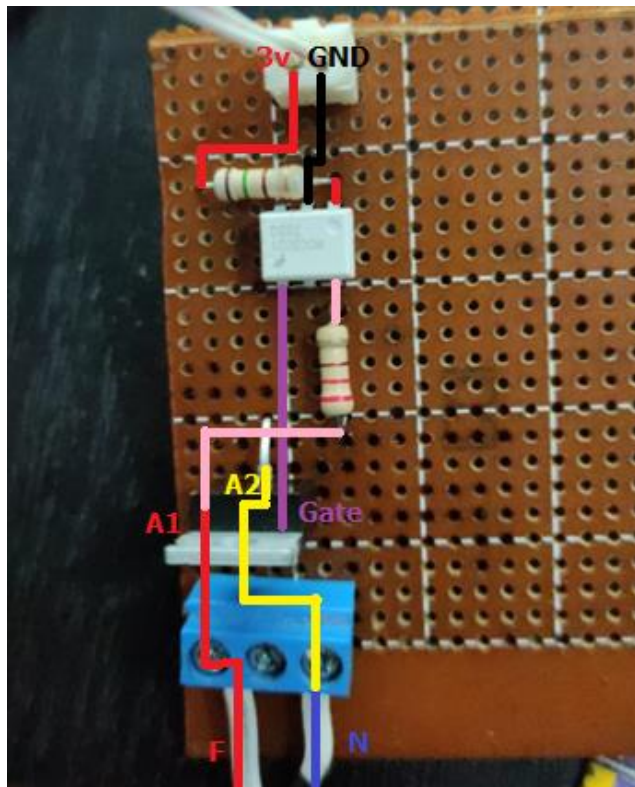
*Fuente.* Elaboración propia

El control mediante TRIAC se encuentra soldado en una baquelita pre perforada, la cual permite facilitar el proceso de construcción del circuito simplemente uniendo las terminales (Como se observa en la *Figura 62*) de cada componente electrónico a la baquelita, soldarlo y unir los cables que se comunican entre estos.



**Figura 62**

*Esquema de conexión.*



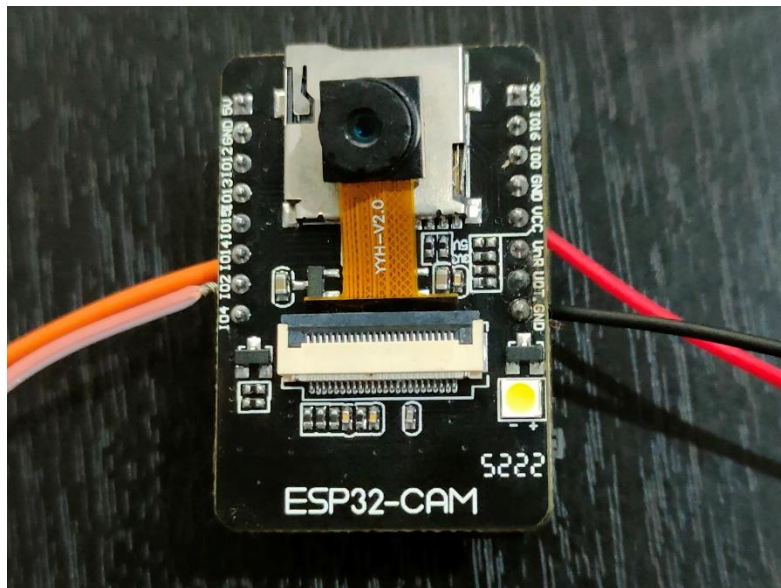
*Fuente.* Elaboración propia.

Los 3 voltios en la parte superior de la imagen son los provenientes de la tarjeta de desarrollo, con el optoacoplador (El componente blanco) se realiza la activación del TRIAC el cual a su vez cierra el circuito y permite la activación de la bombilla.

Por último, se tiene la unión de estos dos módulos de control con el microcontrolador, la ESP32 CAM. Estos se encuentran unidos mediante cables, ya que el microcontrolador también hace de cámara web por lo que su ubicación debe ser estática y para que exista la activación del componente eléctrico, es necesaria la colocación del módulo de control junto con el cable que comunica la tarjeta de desarrollo con la etapa de control.

**Figura 63**

*Conexión de la ESP32 CAM.*



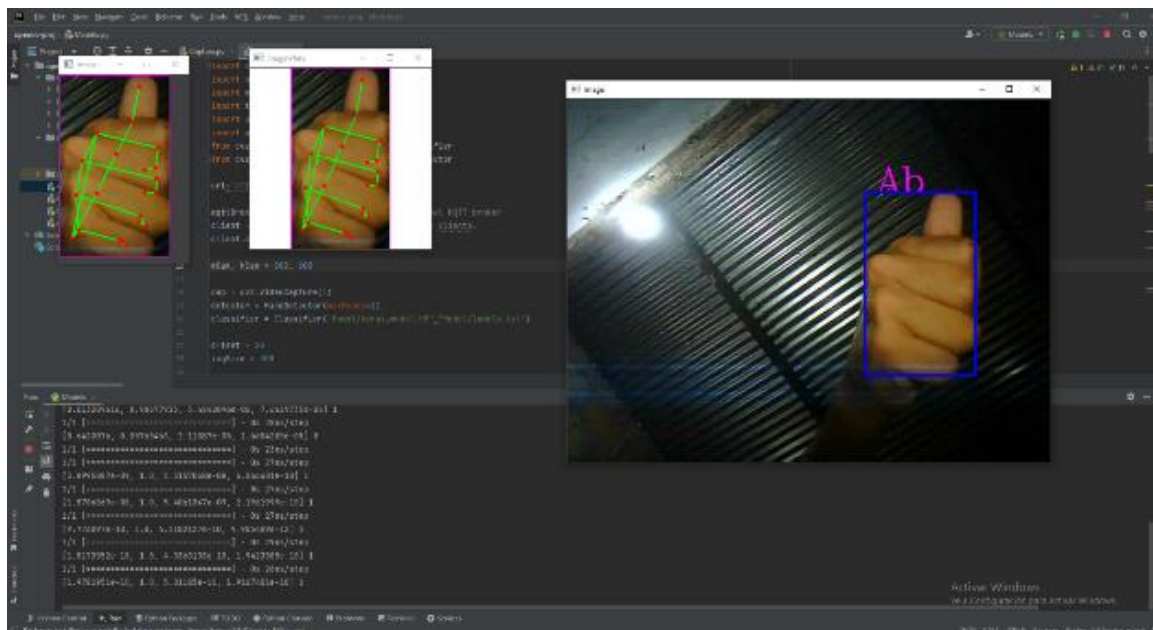
*Fuente.* Elaboración propia

En la ESP32 CAM se tienen los dos cables que activan las dos etapas de potencia, cada activación ligada a la clasificación que haga el software en el procesamiento de imágenes.



**Figura 65**

*Reconocimiento de la seña "Ab".*



*Fuente. Elaboración propia*

**Figura 66**

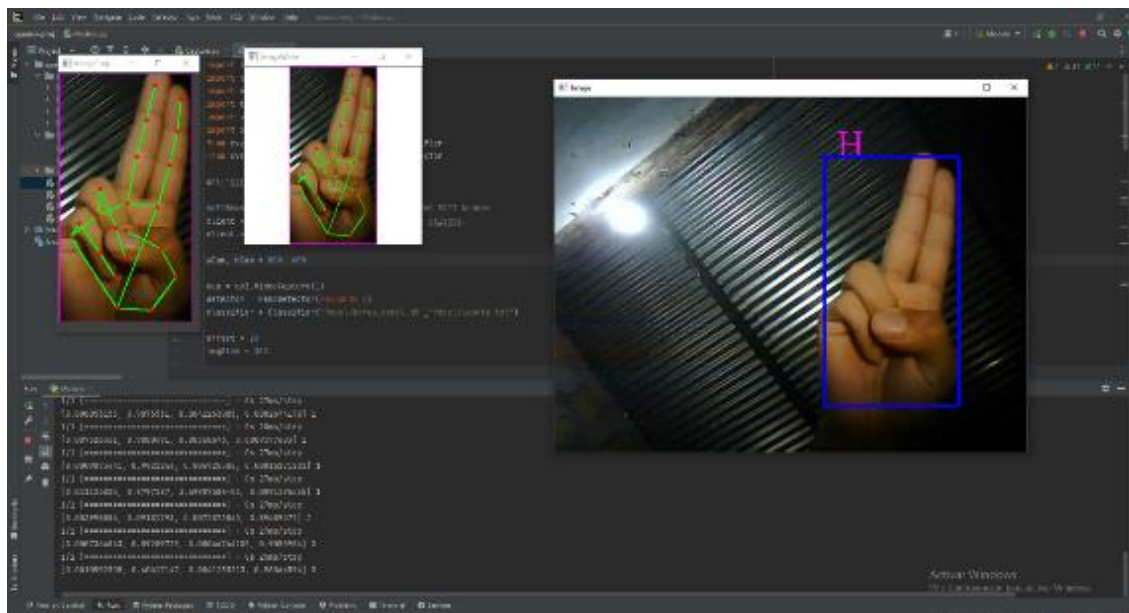
*Reconocimiento de la seña "B".*



*Fuente. Elaboración propia*

**Figura 67**

*Reconocimiento de la seña “H”.*



*Fuente.* Elaboración propia

Las letras Ab, H y A presentan algunas complicaciones a la hora de realizar el reconocimiento debido a la calidad de la cámara, es necesaria una buena iluminación y la capacidad de procesamiento de la ESP32 CAM limita el potencial debido a la cantidad de fotogramas que envía, se alcanza a percibir algunos retrasos debido a que se trata de una cámara web.

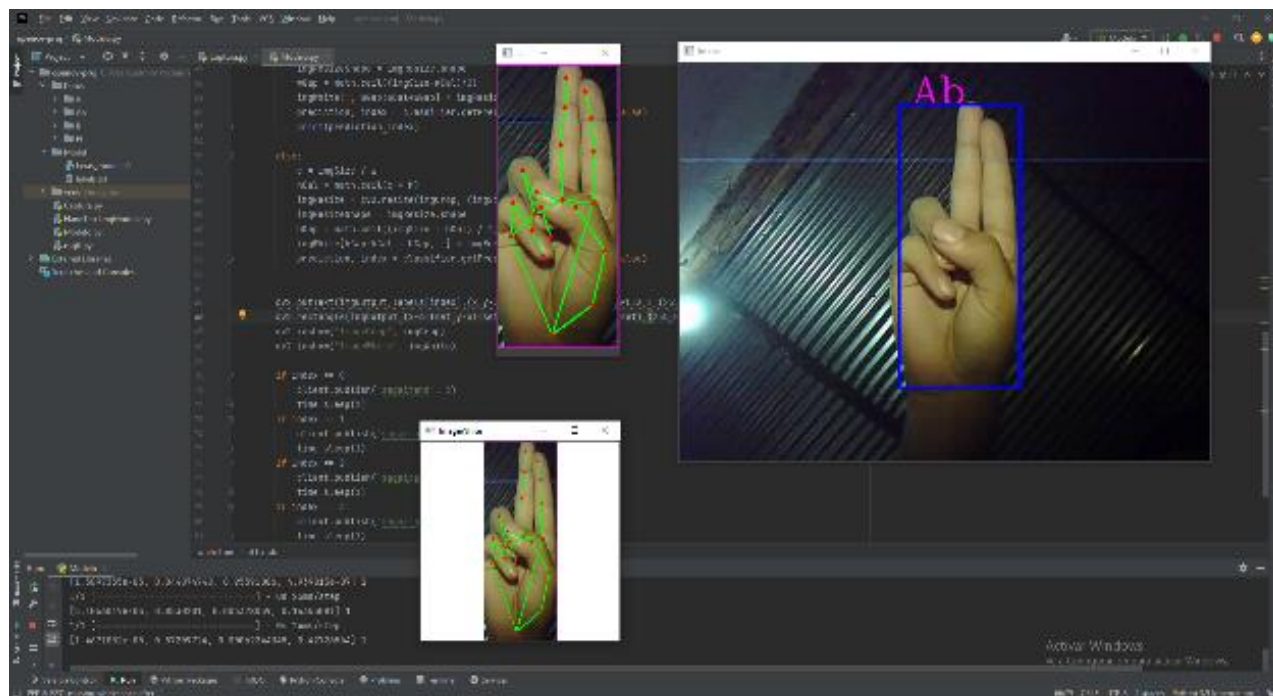
Existen algunos errores ocasionados por mala iluminación y debido a la cantidad de fotogramas por segundo que captura la cámara, por un instante en el momento en que se cambia de seña delante de la Cámara, esta muestra la clasificación de otra seña. A continuación, se puede observar algunos de estos momentos.



## Errores

**Figura 68**

*Error en el cambio de la letra B a H.*

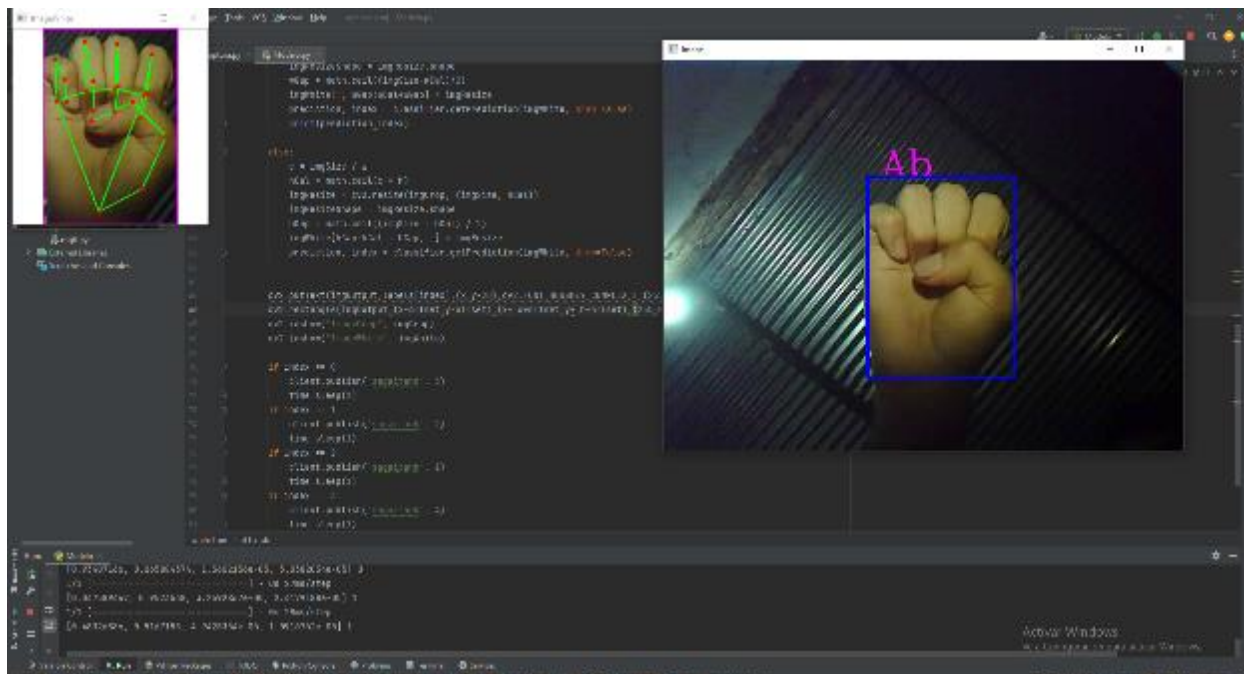


*Fuente. Elaboración propia*

En el cambio de seña "B" que es la palma de la mano abierta a la seña "H" se genera un error debido a la cantidad de fotogramas de captura la Cámara web, ya que estos son limitados. Por esto en los instantes en que la mano cambia de seña, se capturan rastros de ese movimiento que son procesados y llevan al error.

**Figura 69**

*Error en paso de seña Ab a A.*



*Fuente.* Elaboración propia

Al igual que en la figura anterior, este error está relacionado con la cantidad de fotogramas que se capturan y el paso de imágenes en el software de clasificación. Existe un rastro en el cual el software interpreta una “Ab “en lugar de una “A” durante el movimiento de la mano en el transcurso de una seña a la otra.

### **Circuito en Funcionamiento**

Debido a la complejidad del sistema y la manera de verificar que este se encuentre en funcionamiento, se es necesaria una cámara que permita la captura de imagen del circuito para ver que este se esté activando mientras se realiza el reconocimiento en tiempo real. Para esto se utilizará el software de DroidCam, el cual permite la captura de imagen en el ordenador de una

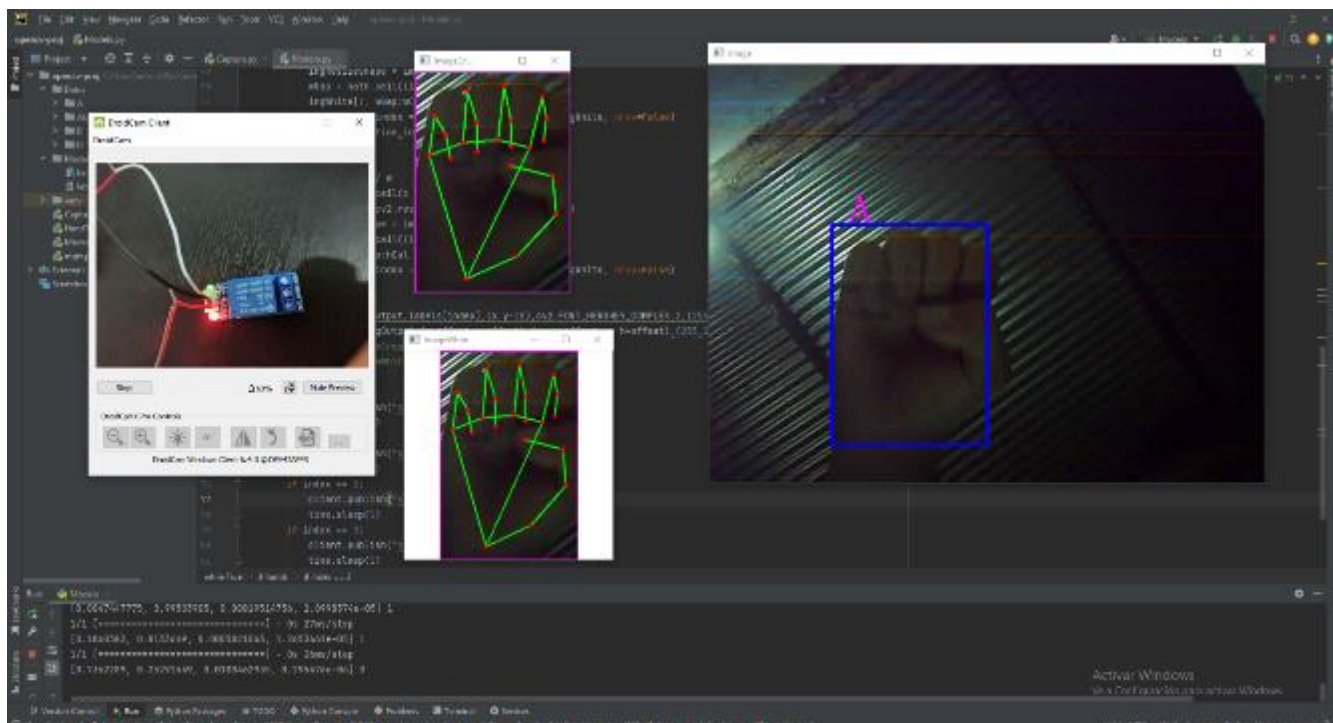
cámara de un SmartPhone, se mostrará entonces una ventana con la cámara en tiempo real cuando se realiza la clasificación de señas.

### ***Funcionamiento del Relé***

Para la activación del relé, este tiene un led el cual permite indicar cuando este se encuentra cerrado, es decir activando la etapa de potencia. En la siguiente imagen se puede observar como el relé tiene el bombillo verde encendido cuando se realiza la clasificación de la letra “A”

### **Figura 70**

*Activación del relé.*



*Fuente.* Elaboración propia

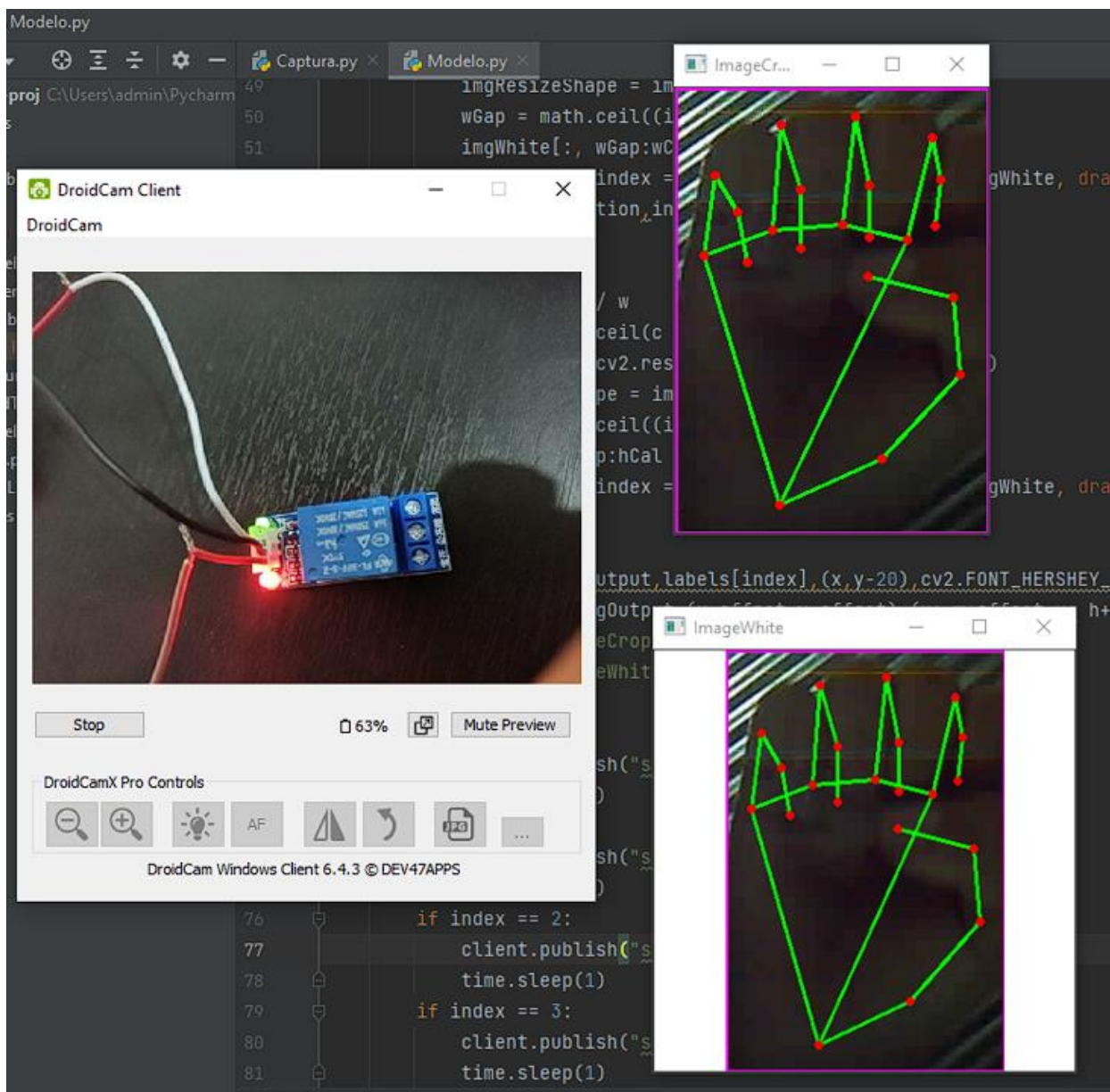
Durante el seguimiento de la mano no se tienen errores. Los errores debido a la diferencia de fotogramas se solucionan debido a que existe un pequeño retraso en la activación de la salida de la ESP32 CAM. Este retraso de 1 segundo permite que el sistema de comparación funcione de



manera correcta. A continuación, se realiza un acercamiento para ver de manera correcta su activación.

### Figura 71

*Acercamiento al relé.*

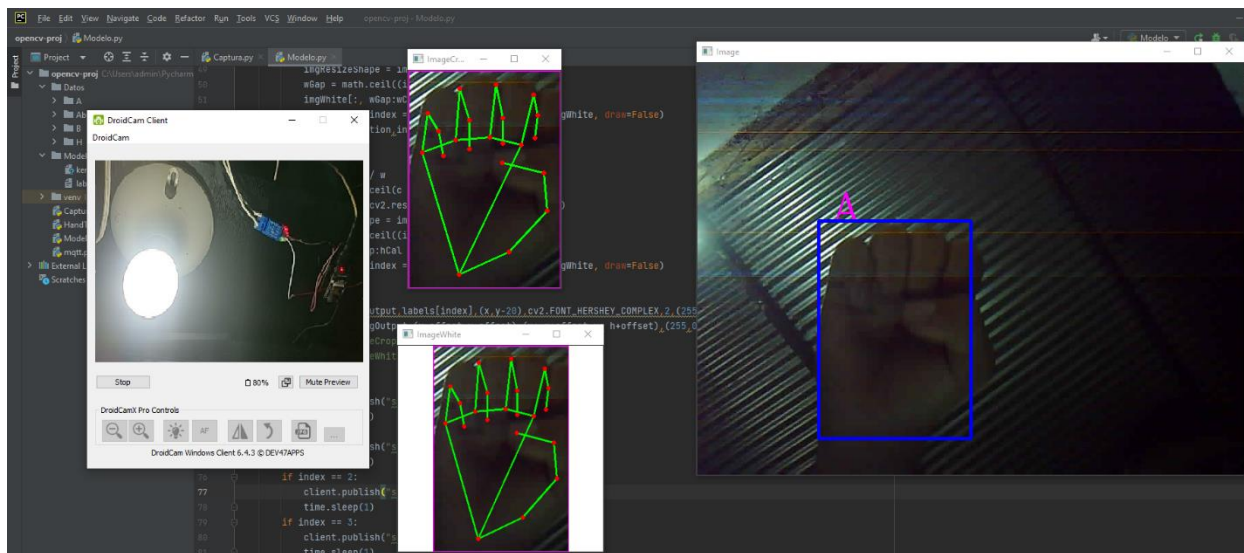


*Fuente. Elaboración propia*

A continuación, se observa la activación de la bombilla, la cual demuestra que el flujo de corriente está siendo activado.

**Figura 72**

*Activación de Bombilla por relé.*

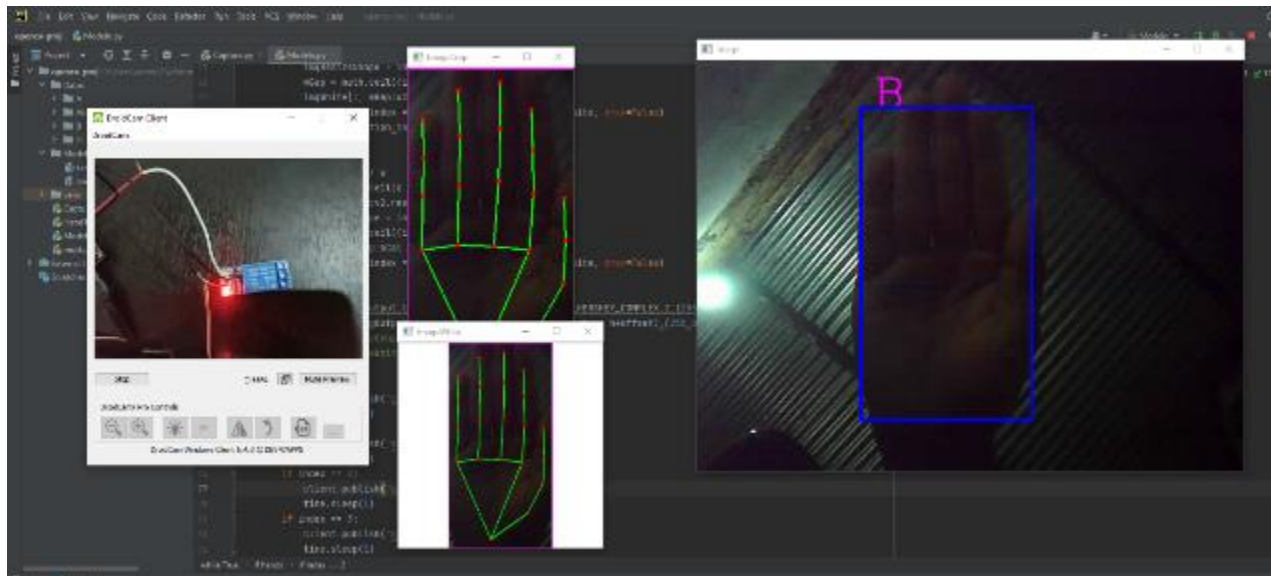


*Fuente.* Elaboración propia

Para la señal de desactivación, la cual es la señal “B” Se tienen buenos resultados, el relé se desactiva y se tiene una correcta clasificación.

**Figura 73**

*Seña de desactivación del relé.*

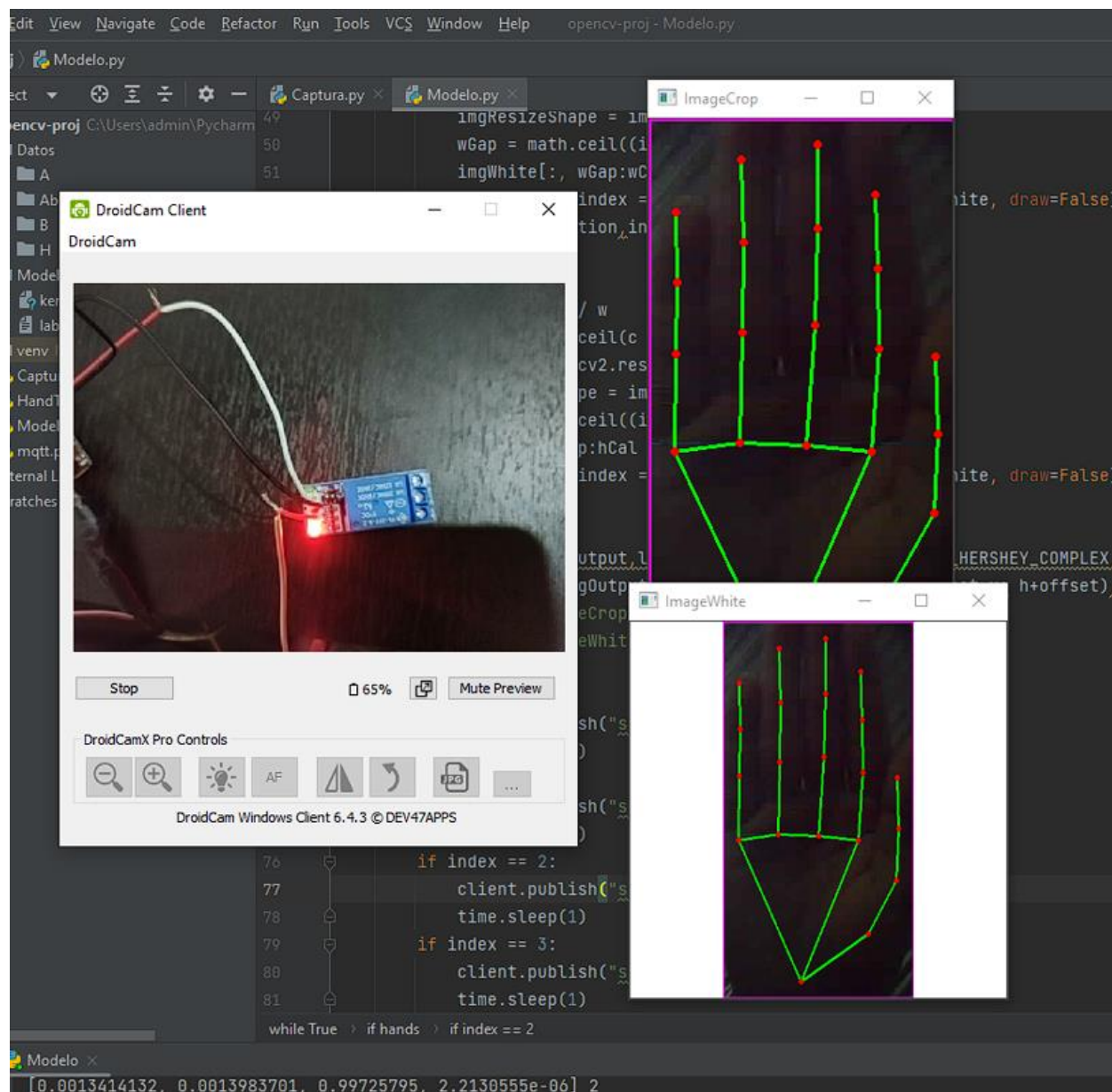


*Fuente.* Elaboración propia

A continuación, se realiza un acercamiento para observar el led indicador de la activación del relé y se puede comprobar que este se encuentra en estado apagado.

Figura 74

*Acercamiento a la desactivación del relé.*

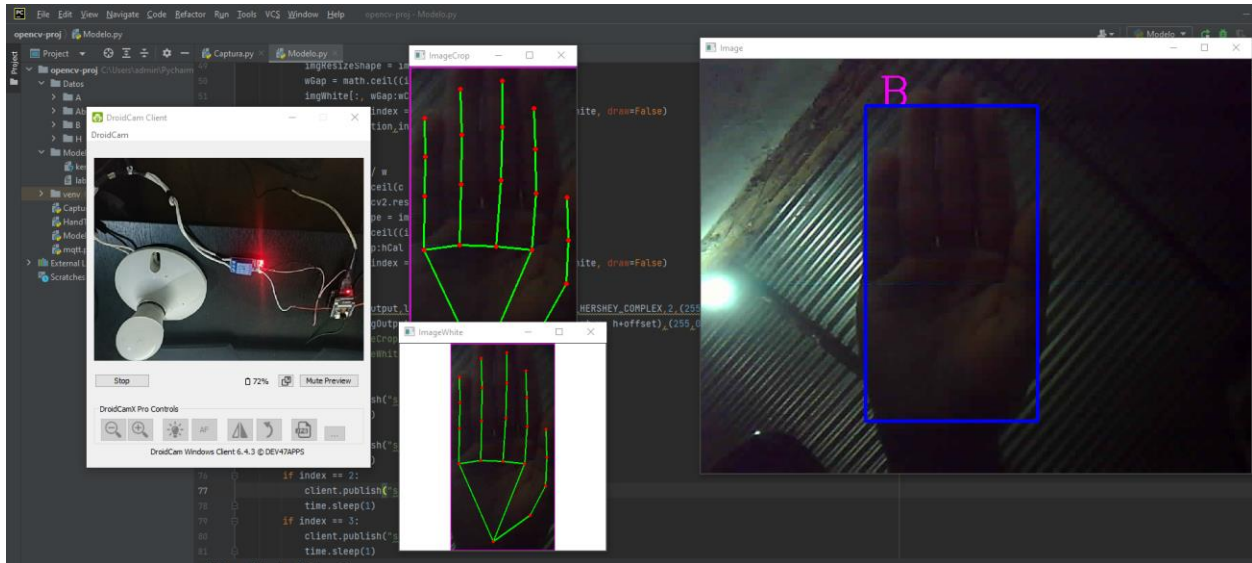


*Fuente.* Elaboración propia

Desactivación de la etapa de potencia, se observa la bombilla ahora desactivada, por lo que el flujo de corriente se corta de manera correcta.

**Figura 75**

*Desactivación de la bombilla por relé.*



*Fuente.* Elaboración propia

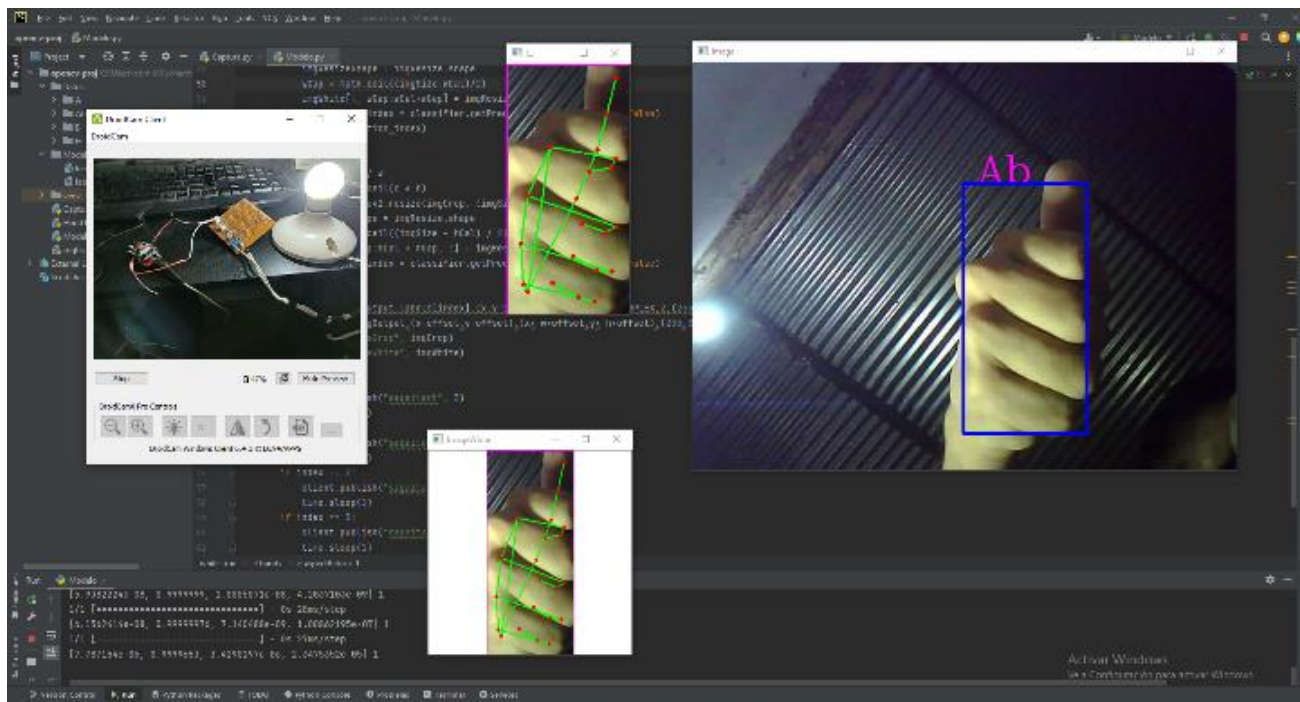
### ***Funcionamiento Del TRIAC***

El circuito de control del TRIAC funciona de manera correcta, y el software clasifica sin errores las dos señas correspondientes, activación (seña Ab) y desactivación (seña H). A continuación, se tiene el funcionamiento con la bombilla, ya que este no tiene indicador como el relé.



**Figura 76**

*Activación de bombilla mediante TRIAC.*



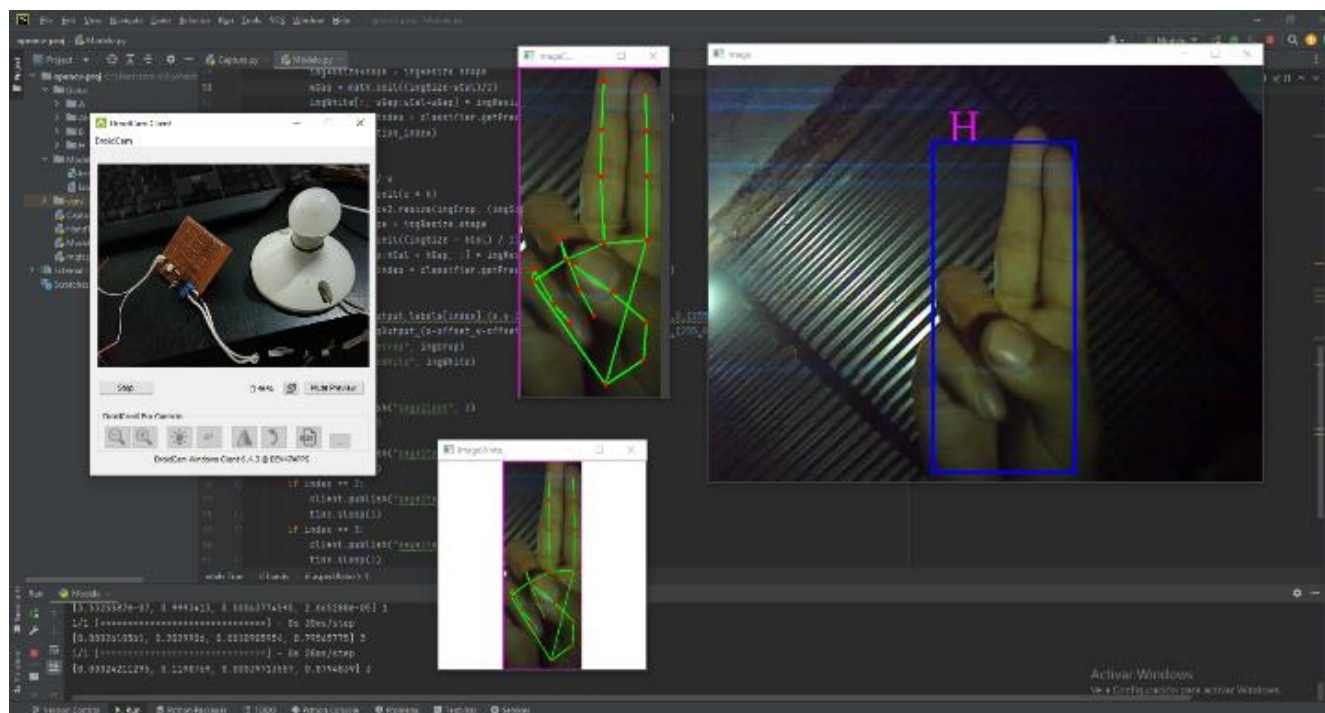
*Fuente.* Elaboración propia

Se observa el reconocimiento y clasificación de la seña Ab la cual corresponde a la segunda etapa de potencia. En esta se puede observar la bombilla en el momento encendida mediante el uso de la seña correspondiente.

En la desactivación. El sistema lo hace inmediatamente se reconoce la seña funcionando de manera correcta, a continuación, su funcionamiento:

Figura 77

*Desactivación de bombilla mediante TRIAC.*



*Fuente. Elaboración propia*

## **Plan de Mejoras**

### **Plan de Mejoras a Largo Plazo**

La placa de desarrollo es de las más sencillas, su velocidad de procesamiento y características técnicas son básicas. Para poder que el sistema sea más estable y la cámara de mejor calidad, es necesaria la renovación del hardware, para ello se puede utilizar una tarjeta de desarrollo más potente o en su defecto utilizar una Raspberry pi mediante la cual no sería necesario el ordenador para que realice todo el procesamiento digital con sus recursos, si no que la tarjeta pueda realizar todo el procesamiento de imágenes por si sola y a su vez la activación/desactivación de ciertos puertos.



## Plan de Mantenimiento

### Plan de Mantenimiento Predictivo

La temperatura del sistema construido debe ser revisada debido a que existe una etapa de potencia y se pueden ocurrir fallos en la captura de imágenes (en el procesamiento digital) debido a climas extremos, es posible que, en algunos casos este se vea afectado por lo que es necesario entonces, un análisis termográfico que permita la regulación de temperatura en el sistema construido, en caso de temperaturas extremas, se es necesaria la aplicación de un sistema de refrigeración, razón por la cual es necesaria también la aplicación de aislantes, se está manejando un dispositivo con corriente alterna que puede llegar a ser peligroso para las personas, por lo cual es necesario que el sistema de control se encuentre debidamente aislado y en una zona donde no esté al alcance común para evitar posibles lesiones. En los dispositivos de control de potencia como se menciona anteriormente, están sujetos a condiciones que pueden alterar su comportamiento, en lo que está el consumo energético, por lo que es necesaria la revisión del consumo cada cierto tiempo y sobre todo si se observa un incremento notable en el consumo energético.

Por último, se tiene la verificación de la cámara, para ello se tiene en caso de la cámara web, una dirección IP a la cual puede acceder un dispositivo por medio de un navegador, esta dirección es arrojada en el monitor serie de Arduino IDE, mediante esta IP se debe ir revisando cada cierto tiempo, puede ser cada semana, sobre todo en caso de que el dispositivo presente alguna falla, para realizar esta revisión, se conecta la tarjeta ESP32-CAM por medio de micro usb y una vez se encuentre conectada, se dirigirá al software de Arduino IDE (se debe descargar en caso de no tenerlo instalado en: <https://www.arduino.cc/en/software/> ) y con el software iniciado, se inicia el monitor serial para 115000 baudios en el cual se mostrará la dirección IP

que ha obtenido la cámara, así como su correcta conexión al Router a continuación, su estructura será (“http://Direccion Ip/cam-lo”), en el apartado de /cam seguirá la resolución de la misma en tres dimensiones (low= baja; mid=media; hi=alta), de esta manera se puede conectar un único dispositivo y verificar que la cámara web se encuentre en su correcto funcionamiento.

### **Plan de Mantenimiento Preventivo**

Es probable que se deba realizar la limpieza del lente de la cámara con frecuencia y esta se ve directamente afectada por el entorno en que se encuentre, en entornos de humedad/suciedad, es necesaria la limpieza con más frecuencia, por otra parte, el sistema de potencia cuenta con un relé y un tiristor, este último tiende a calentarse con su uso y en caso de conectar cargas elevadas es necesario el uso de un sistema de disipación, aquí es donde es importante tener en cuenta el dispositivo a conectar, pues su etapa de diseño de potencia puede variar, es necesaria la vigilancia pues de la electrónica encargada de potencia pues estos son los que más sufren sobre todo al encenderse y apagarse muy seguido. Por otra parte, es necesaria una revisión de que el sistema de clasificación y predicción esté funcionando de manera correcta, en caso de que un dispositivo no esté funcionando como se debe a la hora de usar las señas, comprobar que no sea el dispositivo quien presente los fallos y de ser así, realizar una revisión general al sistema, revisar que los cables estén bien y que las recomendaciones anteriores sean aplicadas.

### **Plan de Mantenimiento Correctivo**

Se debe realizar el cambio del dispositivo de captura, para ello se deberá programar el nuevo hardware con las mejoras correspondientes, es decir que la persona encargada realizara la programación del dispositivo con antelación y con este cambio se realizará la respectiva conexión e instalación del dispositivo con las debidas instrucciones para que el dispositivo

funcione de manera autónoma, dependiendo de la carga a conectar, son necesarios cambios en el diseño de control de la etapa de potencia. Por otra parte, es necesarios algunos retoques en el sistema de predicción pues presenta algunos errores, debido al dispositivo de captura y por la latencia en el protocolo de comunicación, se requiere que el sistema cuente con un mecanismo de transferencia de datos más sólido, así como eliminar por completo la cámara web para que no exista latencia entre la captura de datos y el procesamiento de estos.

Por último, se tiene un sistema de control de potencia que tiene que estar activo para su funcionamiento, la electrónica en el sistema de control puede ser mejorada y disminuir aún más el consumo que pueda presentar, esto con ayuda de un sistema de lógica con mejor diseño, como de un sistema electrónico más inteligente, así como un sistema de refrigeración para el mismo.

## Conclusiones

Se consigue diseñar e implementar el prototipo de control accionado por un lenguaje de señas aplicado a un establecimiento familiar, las 4 señas seleccionadas cumplen con el objetivo de apagar y encender un dispositivo eléctrico, logrando mediante un lenguaje de señas tener un sistema de control centralizado que permite la activación de una bombilla o del dispositivo eléctrico que desee, teniendo en cuenta su respectivo diseño de su circuito en grandes potencias, durante el desarrollo del prototipo se requirieron diferentes conocimientos en el ámbito de ingeniería, se hizo necesaria la búsqueda de información para la selección de señas y su significado en el diccionario, así como también se hizo necesaria la aplicación de protocolos de comunicación y control mediante este por una secuencia lógica diseñada mediante programación, por ultimo fueron necesarios conocimientos de ingeniería electrónica tanto para el diseño de la lógica en programación, como el diseño de control.

Por otra parte, su diseño esta realizado en la tarjeta de desarrollo ESP32-CAM lo cual constituye una tarjeta de pocos recursos y de bajo presupuesto que es libre, para su programación se utilizó el IDE de Arduino el cual también es un software libre que con ayuda de librerías de la ESP32-CAM se consigue programar mediante este, el prototipo final, es un prototipo diseñado para bajo presupuesto, las limitaciones son notorias en la calidad de captura de datos, lo que perjudica las otras etapas, pero aun así se consigue un buen resultado, en estos se puede observar que las pruebas realizadas para activación y desactivación funciona correctamente, con la excepción debida a las diferencias de fotogramas por segundo capturados y procesados, esto debido a que se está utilizando una cámara web con la ESP32-CAM, al usar la cámara web se limita la cantidad de datos que se puede procesar a causa de que se realiza su envío mediante un

enlace a internet mediante el Router (Tenda F3) dejando a la tarjeta con limitaciones la hora de enviar datos por medio de wifi, quedando con pequeños cortes durante su captura de datos.

El prototipo construido se trata de una solución de bajos recursos que permite abordar un tema como la discapacidad auditiva y del habla, es un prototipo funcional que cumple con lo necesario, pero durante el documento se habla de la aplicación de una tarjeta más potente que podría hacer el proyecto más conciso y permitir dar una solución de mayor calidad, esta sería una solución que requeriría de mayores recursos y sus mejoras serian claramente notables.

Finalmente, este prototipo puede dar pie a futuras investigaciones del tema, incluso aplicaciones con una tarjeta más potente y obtener mejores resultados, parte del desarrollo de este proyecto, es impulsar a las investigaciones tecnológicas para personas con discapacidades que permitan una mejora en su estilo y calidad de vida.

### Referencias Bibliográficas

- Aguirre, N., Aranda, N., y Balich, N. (2019). Seguridad en el envío de mensajes mediante protocolo MQTT en IoT. *INNOVA UNTREF. Revista Argentina de Ciencia y Tecnología*.  
<https://www.revistas.untref.edu.ar/index.php/innova/article/view/1000>
- AliOs. (2019, noviembre 6). Raspberry Pi 4 GPIO Pinout. *Key to Smart*.  
<https://keytosmart.com/single-board-computers/raspberry-pi-4-gpio-pinout/>
- BTA41 pdf, BTA41 Description, BTA41 Datasheet, BTA41 view: ALLDATASHEET :::* (s. f.).  
<https://pdf1.alldatasheet.com/datasheet-pdf/view/22033/STMICROELECTRONICS/BTA41.html>
- Chávez, M. Alvarez, Y. De la Rosa, A. Virgen, M. Castro, S. Déficit Auditivo en Pacientes Atendidos en Otorrinolaringología del IMSS en Guadalajara. *Rev. Med. Inst. Mex. Seguro Soc*; 46 (3): 315-322
- Chiquito, V. N. M., Vera, E. P. V., Rodríguez, A. G. P., Alay, L. P. M., y Merino, M. J. M. (2022). Transistores: componente innovador en la electrónica. *UNESUM-Ciencias. Revista Científica Multidisciplinaria*. ISSN 2602-8166, 6(3), 13-19.
- D. Someshwar, D. Bhanushali, V. Chaudhari and S. Nadkarni, "Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 595-600, doi: 10.1109/ICIRCA48905.2020.9183179.
- Escobar, A. J. P., Flórez, P. L. D., y Fernández-M, D. S. (2023). Estimación de irregularidades en pavimentos mediante técnicas de procesamiento digital de imágenes. *Revista Politécnica*, 19(37), 20-28.

*ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained* | Random Nerd Tutorials.

(2020, marzo 10). <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>

Firtec. (s. f.). *Que es MQTT*. <https://www.firtec.com.ar/cms/53-que-es-mqtt>

*Getting started with the Camera Module | Python | Coding projects for kids and teens*. (s. f.).

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>

Google. (s. f.) *Teachable Machine*. <https://teachablemachine.withgoogle.com/>

Gutiérrez-Farfán, I., Arch-Tirado, E., Lino-González, A. L., y Jardines-Serralde, L. J. (2018).

Daño auditivo inducido por ruido recreativo. *Salud Pública de México*, 60, 126-126.

<https://doi.org/10.21149/9042>

Guru99es. (2020). Clasificación de imágenes de TensorFlow: CNN (Red Neural Convolutacional).

Guru99. <https://guru99.es/convnet-tensorflow-image-classification/>

Hernández, C., Pulido, J. L., y Arias, J. E. (2015). Las tecnologías de la información en el aprendizaje de la lengua de señas. *Revista de Salud Pública*, 17, 61-73.

Jiménez, A., Jiménez, F., y García, D. (n.d.). PIND – Precision Index 1. 0. Remote Sensing Software for Precision Agriculture Studies, 181–187.

Keras | TensorFlow Core. (s. f.). TensorFlow. <https://www.tensorflow.org/guide/keras?hl=es-419>

Lemus, I. (2019, agosto 21). ¿Qué es una tarjeta de desarrollo microcontrolador para IOT?

*Conocimiento Libre*. <https://conocimientolibre.mx/microcontrolador/>

Luo, Congbo y Hao, Yunhui y Tong, Zihe. (2018). Research on Digital Image Processing Technology and Its Application. 10.2991/meici-18.2018.116.

Macías García, M. E., Contreras Hinojosa, C., Anaya Zamora, R., García, M. E. M., Hinojosa, C.

C., y Zamora, R. A. (2018). *Ley de Ohm*. <https://repositorio.tec.mx/handle/11285/631150>

Manusoria40. (2022, junio 30). Deep Learning Colombia. *Inteligencia Artificial Aplicada*

*Colombia*. <https://ia.smartecorganic.com.co/deep-learning/>

Mecafenix, I. (2018, octubre 17). ¿Qué es un triac y sus aplicaciones? *Ingeniería Mecafenix*.

<https://www.ingmecafenix.com/electronica/componentes/triac/>

Ministerio De Salud. (s. f.). *Discapacidad*.

<https://www.minsalud.gov.co/proteccionsocial/promocion-social/Discapacidad/Paginas/discapacidad.aspx>

Ministerio Nacional Para Sordos (s.f). Diccionario básico de la lengua de señas colombiana.

[http://www.insor.gov.co/descargar/diccionario\\_basico\\_completo.pdf](http://www.insor.gov.co/descargar/diccionario_basico_completo.pdf)

Mischianti, R. (2021, agosto 30). *ESP32-CAM: Pinout, specs and Arduino IDE configuration -*

*I. Renzo Mischianti*. <https://www.mischianti.org/2021/08/30/esp32-cam-pinout-specs-and-arduino-ide-configuration-1/>

*MOC3021 pdf, MOC3021 Description, MOC3021 Datasheet, MOC3021 view:*

*ALLDATASHEET* ::: (s. f.). <https://pdf1.alldatasheet.com/datasheet-pdf/view/5039/MOTOROLA/MOC3021.html>

Muñiz, A. (2020). *Conducción con discapacidad auditiva*. Fundación MAPFRE.

Niño Martínez, Y. M. (2020). *Análisis multitemporal mediante imágenes de sensores remotos para la determinación de los cambios de uso de suelo en el municipio de San Francisco de Sales, Cundinamarca (Colombia) entre los años 90'sy 2018* (Doctoral dissertation, Universidad Nacional de La Plata).

Pascual C. (2022). *Programar fácil. ESP32 CAM introducción y primeros pasos*.

<https://programarfacil.com/esp32/esp32-cam/>



- Patiño-Toro, O. N., Patiño-Vanegas, J. C., Fernández Toro, A., y Jiménez Guzmán, A. (septiembre-diciembre, 2020). Tendencias investigativas en el estudio de tecnologías inclusivas para población sorda. *Revista Virtual Universidad Católica del Norte*, (61), 283-303. <https://www.doi.org/10.35575/rvucn.n61a17>
- Pinzón-Bayona, G., y Sanabria-Orjuela, Y. G. (2022). Desarrollo de una aplicación móvil para traductor de lenguaje de señas mediante el uso de servicios web.
- Portafolio*. (s. f.). *500.000 personas en Colombia con discapacidad auditiva*. <https://www.portafolio.co/mas-contenido/500-000-personas-en-colombia-con-discapacidad-auditiva-545068>
- P.Perez, M. Valente (2018) *Fundamentos básicos del procesamiento de imágenes— Documentación de Curso de imágenes médicas—1.0*. <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>
- Raspberry Pi (2019). ¿Qué es Raspberry Pi?. *Raspberry Pi*. <https://raspberrypi.cl/que-es-raspberry/>
- R. Avila (2022) Sign language, accessibility, technology, and translation: State of the art, reflections, and proposals. Tesis. URI: <http://hdl.handle.net/10396/22707>
- R. Hugo (2021). ¿Qué es OpenCV y para qué sirve?. *Transformación digital*. [https://www.crehana.com. https://www.crehana.com/blog/transformacion-digital/que-es-opencv/](https://www.crehana.com.https://www.crehana.com/blog/transformacion-digital/que-es-opencv/)
- REGATA. (2012, octubre 8). *Módulo de Relés*. <https://tallerarduino.com/2012/10/08/modulo-de-reles/>
- Rossi, M., y Labella T. (s.f). Audición Y Lenguaje. El Niño Sordo. Hospital Clínico Universitario de Santiago de Compostela. <https://seorl.net/PDF/Otologia/031%20->

[%20AUDICI%C3%93N%20Y%20LENGUAJE.%20EL%20NI%C3%91O%20SORDO.](#)

pdf

Rouhiainen, L. (2018). Inteligencia artificial. Madrid: Alienta Editorial.

Serrano, O. (2022, junio 22). Comunicación no verbal: Elementos y características. *Agencia comma*. <https://agenciacomma.com/formacion-en-comunicacion/comunicacion-no-verbal/>

Shreya, Debbata. (2021). DIGITAL IMAGE PROCESSING AND RECOGNITION USING PYTHON. *International Journal of Engineering Applied Sciences and Technology*. 5. 10.33564/IJEAST. 2021.v05i10.046.

Suaste, S. y Peña, E. (2020). Aplicación de redes neuronales convolucionales en el análisis de estructuras. <https://www.boletin.upiita.ipn.mx/index.php/ciencia/899-cyt-numero-81/1852-aplicacion-de-redes-neuronales-convolucionales-en-el-analisis-de-estructuras>

Thyssen A. (s. f.). Imagemagick. Filter Comparison. <https://imagemagick.org/Usage/filter/>

Tyagi, Vipin. (2018). Understanding Digital Image Processing. 10.1201/9781315123905.

*Upload Code to ESP32-CAM using ESP32-CAM-MB USB | Random Nerd Tutorials*. (2021, enero 28). <https://randomnerdtutorials.com/upload-code-esp32-cam-mb-usb/>

Yani, Muhamad & Irawan, S, & Setianingsih, Casi. (2019). Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail. *Journal of Physics: Conference Series*. 1201. 012052. 10.1088/1742-6596/1201/1/012052.

Zuleta, A. M. P., y Sanjuán, E. D. J. Z. (2018). Agricultura de precisión en Colombia utilizando teledetección de alta resolución. *Suelos Ecuatoriales*, 48(1), 41-49.

## Apéndices

### Apéndice A

*Algoritmo diseñado en python*

#### ***Código 1 Algoritmo Principal Python***

El siguiente código es el principal del prototipo, es el encargado de realizar la captura de la cámara web y realizar un tratamiento de las imágenes capturadas, con el tratamiento de la captura de imágenes realizada en tiempo real, realiza la aplicación del modelo entrenado en Teachable Machine y con ayuda de Computer Vision se realiza un clasificador de las señas A, Ab, B y H, con este clasificador se realiza la predicción del tipo de seña que reciba por medio de la cámara web, con esta predicción se realiza un pequeño texto que muestre la letra en pantalla para comprobar su funcionamiento, ya que la predicción se muestra en la terminal mediante el predictor, este se encarga de definir la letra por medio de un número del 0 al 3 correspondiente a un vector llamado "labels" donde se encuentran las letras a las cuales hace referencia cada seña (A, Ab, B y H. Por último, se tiene el sistema de lógica y envió de datos por medio del servidor mqtt, el cual consiste en una comparación, ya que el sistema indica el valor de la posición del vector labels para indicar la letra a la que hace referencia.

```
import cv2
import numpy as np
import math
import time
import urllib.request
import paho.mqtt.client as mqtt
from cvzone.ClassificationModule import Classifier
from cvzone.HandTrackingModule import HandDetector

url='http://192.168.0.108/cam-hi.jpg'

mqttBroker = "broker.hivemq.com" # Elección del MQTT broker
client = mqtt.Client("sagaitanh") # Nombre de cliente.
client.connect(mqttBroker)
```

```

wCam, hCam = 800, 600

cap = cv2.VideoCapture(1)
detector = HandDetector(maxHands=1)
classifier = Classifier("Model/keras_model.h5", "Model/labels.txt")

offset = 20
imgSize = 300

folder = "Datos/H"
contador = 0

labels = ["A", "Ab", "B", "H"]

while True:
    success, img = cap.read()
    img_resp = urllib.request.urlopen(url)
    imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
    img = cv2.imdecode(imgnp, -1)
    imgOutput = img.copy()
    hands, img = detector.findHands(img)
    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']
        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y-offset:y + h+offset, x-offset:x + w+offset]
        imgCropShape = imgCrop.shape
        aspectRatio = h/w

        if aspectRatio > 1:
            c = imgSize/h
            wCal = math.ceil(c*w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize-wCal)/2)
            imgWhite[:, wGap:wCal+wGap] = imgResize
            prediction, index = classifier.getPrediction(imgWhite,
draw=False)
            print(prediction, index)

        else:
            c = imgSize / w
            hCal = math.ceil(c * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize
            prediction, index = classifier.getPrediction(imgWhite,
draw=False)

        cv2.putText(imgOutput, labels[index], (x, y-
20), cv2.FONT_HERSHEY_COMPLEX, 2, (255, 0, 255), 2)
        cv2.rectangle(imgOutput, (x-offset, y-offset), (x+ w+offset, y+
h+offset), (255, 0, 0), 4)

```



```

        return img

    def findPosition(self, img, handNo=0, draw=True):

        lmList = []
        if self.results.multi_hand_landmarks:
            myHand = self.results.multi_hand_landmarks[handNo]
            for id, lm in enumerate(myHand.landmark):
                h, w, c = img.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                lmList.append([id, cx, cy])
                if draw:
                    cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

        return lmList

def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(1)
    detector = handDetector()
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])

        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)

        cv2.imshow("Image", img)
        cv2.waitKey(1)

if __name__ == "__main__":
    main()

```

### ***Código 3 Algoritmo de Captura de Imágenes para la Base de Datos Python***

Este algoritmo es el encargado de la captura de datos, funciona de tal manera que mientras se presione la letra “s”, se realizara el guardado de una imagen con el nombre de “Imagen\_” seguido de un número que indica la fecha y la hora del momento en UTC, en este código se tiene un tratamiento de las imágenes capturadas, se recortan y se pasan por diferentes

capas para poder obtener una imagen de 300x300 pixeles y se acomode automáticamente de manera vertical como horizontal a la hora de realizar un seguimiento de la mano, realizando un recorte que tiene como limite el esqueleto dibujado por el sistema de seguimiento más el “offset”, permitiendo así, tener enfocada la mano en el momento que se realiza el reconocimiento de la misma. Este mismo código es utilizado como base para el principal.

```
import cv2
import numpy as np
import math
import time

from cvzone.HandTrackingModule import HandDetector

cap = cv2.VideoCapture(1)
detector = HandDetector(maxHands=1)

offset = 20
imgSize = 300

folder = "Datos/H"
contador = 0

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)
    if hands:
        hand = hands[0]
        x,y,w,h = hand['bbox']
        imgWhite = np.ones((imgSize,imgSize,3),np.uint8)*255
        imgCrop = img[y-offset:y+h+offset,x-offset:x+w+offset]
        imgCropShape = imgCrop.shape
        aspectRatio = h/w

        if aspectRatio > 1:
            c = imgSize/h
            wCal = math.ceil(c*w)
            imgResize = cv2.resize(imgCrop,(wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize-wCal)/2)
            imgWhite[:, wGap:wCal+wGap] = imgResize
        else:
            c = imgSize / w
            hCal = math.ceil(c * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize
```

```
cv2.imshow("ImageCrop", imgCrop)
cv2.imshow("ImageWhite", imgWhite)

cv2.imshow("Image",img)
key = cv2.waitKey(1)
if key == ord("s"):
    contador += 1
    cv2.imwrite(f'{folder}/Imagen_{time.time()}.jpg',imgWhite)
    print(contador)
```



## Apéndice B

*Algoritmo diseñado en Arduino IDE*

### ***Código 1 Algoritmo de Programación para la ESP32-CAM Arduino IDE***

Este código corresponde al algoritmo desarrollado para la tarjeta ESP32-CAM AI THINKER, es el código compilado en la tarjeta de desarrollo, contiene la configuración del mqtt bróker, la definición de los puertos de salida, conexión a wifi, conexión al troker, creación de la cámara web, lectura de los datos del mqtt bróker y por último de la secuencia lógica de la activación de las dos salidas.

```
#include <Arduino.h>
#include <PubSubClient.h>
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"

#define RELAY 2
#define TRIAC 14

const char* WIFI_SSID = "-----";
const char* WIFI_PASS = "-----";
const char* mqttBroker = "broker.hivemq.com";
WiFiClient client;
PubSubClient mqtt(client);

WiFiClient espClient;

void connectWifi();
void connect_mqtt();
void mqttReceive(char *topic, byte*msg, unsigned int msgLength);

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
```

```
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("Captura de camara fallida");
        server.send(503, "", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
        static_cast<int>(frame->size()));

    server.setContentLength(frame->size());
    server.send(200, "image/jpeg");
    WiFiClient client = server.client();
    frame->writeTo(client);
}

void handleJpgLo()
{
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg();
}

void handleJpgHi()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
    serveJpg();
}

void handleJpgMid()
{
    if (!esp32cam::Camera.changeResolution(midRes)) {
        Serial.println("SET-MID-RES FAIL");
    }
    serveJpg();
}

void setup(){
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
```

```

pinMode(RELAY, OUTPUT);
Serial.begin(115200);
Serial.println();
mqtt.setServer(mqttBroker, 1883);
mqtt.setCallback(mqttReceive);
{
    using namespace esp32cam;
    Config cfg;
    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "Camara conectada" : "Fallo en la camara");
}
WiFi.persistent(false);
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASS);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
}
Serial.print("http://");
Serial.println(WiFi.localIP());
Serial.println(" /cam-lo.jpg");
Serial.println(" /cam-hi.jpg");
Serial.println(" /cam-mid.jpg");

server.on("/cam-lo.jpg", handleJpgLo);
server.on("/cam-hi.jpg", handleJpgHi);
server.on("/cam-mid.jpg", handleJpgMid);

server.begin();
}

void loop()
{
    server.handleClient();
    if(!mqtt.connected()){
        connect_mqtt();
        Serial.println("MQTT troker conectado");
    }
    mqtt.loop();
    delay(100);
}

```

```
}  
  
void connect_mqtt(){  
    while (!mqtt.connected())  
    {  
        Serial.println("Conectando al troker MQTT...");  
        if(mqtt.connect("mqtt_test")){  
            mqtt.subscribe("sagaitanh");  
        }  
    }  
}  
  
}int value;  
  
void mqttReceive(char *topic, byte*msg, unsigned int msgLength){  
  
    String data;  
    for(int i = 0; i < msgLength; i++){  
        Serial.print(char(msg[i]));  
        data += (char)msg[i];  
    }  
    Serial.println("");  
  
    int value = data.toInt();  
    digitalWrite(RELAY, 0);  
  
    switch (value)  
    {  
    case 3:  
        digitalWrite(RELAY, 0);  
        break;  
  
    case 1:  
        digitalWrite(RELAY, 1);  
        break;  
  
    case 2:  
        digitalWrite(TRIAC, 1);  
        break;  
  
    case 4:  
        digitalWrite(TRIAC, 0);  
        break;  
    }  
}
```