

**Universidad Católica de Santa María**  
**Facultad de Ciencias e Ingenierías Físicas y Formales**  
**Escuela Profesional de Ingeniería Mecatrónica**



**Diseño de un Sistema de Visión Artificial para el Uso de una Mano Robótica  
en la Manipulación de Sustancias Tóxicas.**

Tesis presentada por el Bachiller:

**Mamani Zevallos, Ronal Adrian**

**ORCID: 0009-0005-8472-7905**

para optar el Título Profesional de Ingeniero Mecatrónico

Asesor:

**Mg. Collado Oporto, Christiam Guillermo**

**ORCID: 0000-0002-0529-738X**

Arequipa – Perú

2024

UCSM-ERP

**UNIVERSIDAD CATÓLICA DE SANTA MARÍA**

**INGENIERIA MECATRONICA**

**TITULACIÓN CON TESIS**

**DICTAMEN APROBACIÓN DE BORRADOR**

Arequipa, 02 de Mayo del 2024

**Dictamen: 007772-C-EPIMMEM-2024**

Visto el borrador del expediente 007772, presentado por:

**2017222981 - MAMANI ZEVALLOS RONAL ADRIAN**

Titulado:

**DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL USO DE  
UNA MANO ROBÓTICA EN LA MANIPULACIÓN DE SUSTANCIAS  
TÓXICAS**

Nuestro dictamen es:

**APROBADO**

Título Profesional/Título de Segunda Especialidad/Grado Académico a optar:

**INGENIERO MECATRÓNICO**

**29387211 - FERNANDEZ BARRIGA CAMILO GRIMALDO  
DICTAMINADOR**



**29277716 - CUADROS MACHUCA JUAN CARLOS  
DICTAMINADOR**



**46292714 - SILES NATES FERNANDO DAVID  
DICTAMINADOR**



# Diseño de un Sistema de Visión Artificial para el Uso de una Mano Robótica en la Manipulación de Sustancias Tóxicas.

## INFORME DE ORIGINALIDAD

5%

INDICE DE SIMILITUD

6%

FUENTES DE INTERNET

1%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1

[hdl.handle.net](https://hdl.handle.net)

Fuente de Internet

2%

2

[es.scribd.com](https://es.scribd.com)

Fuente de Internet

2%

3

[Submitted to University of Wollongong](#)

Trabajo del estudiante

1%

4

[repositorio.utn.edu.ec](https://repositorio.utn.edu.ec)

Fuente de Internet

1%

Excluir citas

Apagado

Excluir coincidencias < 1%

Excluir bibliografía

Apagado

## DEDICATORIA

Dedico este trabajo a mi Ma Teo,  
que aún puedo sentir su presencia  
y apoyo a través de mis familiares.



## AGRADECIMIENTO

Doy gracias a Dios por guiarme en el camino de la vida, por darme salud, brindarme protección, otorgarme vida y alegría durante mi desarrollo personal y profesional. Agradezco a mi mamá, por su arduo esfuerzo e incondicional apoyo durante mi formación profesional, a mi papá, por ayudarme en los momentos difíciles y poder obtener mis logros. A ellos dos que me dieron la vida y me guían y aconsejan antes, durante y después de esta etapa de mi vida. Agradezco a toda mi familia por sus presiones, preocupaciones y consejos, así como también por su cariño y amor hacia mi persona. También agradezco a mis amigos, compañeros y profesores, por su voto de confianza y enseñanzas brindadas.



## RESUMEN

El desarrollo de este proyecto tiene como finalidad obtener un software de visión artificial capaz de identificar una mano humana a través de una cámara, y con ella realizar un control implementado a una mano robótica con tecnología de impresión 3D. El control se basará en los movimientos independientes de cada dedo, replicando el movimiento de una mano humana que se muestre a la cámara en el prototipo impreso en 3D en tiempo real sin la necesidad de usar sensores para su funcionamiento. Con la implementación del algoritmo de control se creará además una interfaz de usuario para la manipulación y pruebas antes de iniciar con el control por visión artificial.

El trabajo realizado consta de cuatro capítulos en los cuales se describe de la siguiente manera:

El capítulo I Aspectos Generales presenta la descripción e identificación del problema, además presenta los antecedentes de la investigación, objetivos, justificación del trabajo, alcances y limitaciones.

El capítulo II Marco Teórico narra el contexto conceptual que enmarca el desarrollo del presente trabajo para entenderlo con mayor facilidad. Se describen argumentos generales y específicos desde prototipos robóticos, visión artificial, manipulación de sustancias tóxicas y los softwares utilizados para el fin de este trabajo.

En el capítulo III Diseño e Implementación se desarrolla el procedimiento desde la selección del prototipo y etapas para la creación del sistema de visión artificial, la programación y el hardware requerido.

Finalmente, en el capítulo IV Pruebas de Funcionamiento y Resultados se muestra la operatividad en cada etapa del sistema, y pruebas realizadas con el prototipo de mano robótica impreso en 3D.

**Palabras claves:** visión artificial, mano robótica, impresión 3D

## ABSTRACT

The development of this project aims to create an artificial vision software capable of identifying a human hand through a camera and using it to control a robotic hand with 3D printing technology. The control will be based on the independent movements of each finger, replicating the motion of a human hand displayed to the camera in the 3D-printed prototype in real-time without the need for sensors for its operation. With the implementation of the control algorithm, a user interface will also be created for manipulation and testing before initiating artificial vision control.

The work consists of four chapters described as follows:

Chapter I General Aspects: Presents the description and identification of the problem, provides the research background, objectives, justification of the work, scope, and limitations.

Chapter II Theoretical Framework: Describes the conceptual context framing the development of this work for better understanding. It covers general and specific aspects, including robotic prototypes, artificial vision, manipulation of toxic substances, and the software used for the purpose of this work.

Chapter III Design and Implementation: Develops the procedure from the selection of the prototype and stages for creating the artificial vision system, programming, and the required hardware.

Finally, in Chapter IV Performance Testing and Results: Demonstrates the functionality at each stage of the system and tests conducted with the 3D-printed robotic hand prototype.

**Keywords:** artificial vision, robotic hand, 3D printing.

## INTRODUCCIÓN

Desde hace algunos años, los robots se utilizan para realizar trabajos especialmente peligrosos o para moverse por zonas inhóspitas, entrar en contacto con ambientes nocivos y peligrosos para los seres humanos (Consejo de Seguridad Nuclear, 2014).

En algunos los sectores de procesos productivos de la industria, laboratorios farmacéuticos e investigaciones químicas o biológicas involucran la manipulación de sustancias tóxicas. Los operarios se encuentran expuestos a sufrir algún accidente o enfermedad a corto o largo plazo por accidentes laborales o mala capacitación en la manipulación de estas sustancias, teniendo consecuencias en su salud, afectando la economía de estas personas y sus familias e incluso estar expuestos a la muerte por la exposición a los agentes químicos.

Los productos químicos poseen un riesgo potencial para la salud, ya sea por fugas o derrame, por una situación accidental en las plantas de procesamiento industrial o por errores humanos.

Durante las últimas décadas ha ido evolucionando los métodos de control operativo y mando a distancia, desarrollándose nuevas tecnologías que exigen la adecuación de nuevos avances en aplicaciones de la vida real, como la robótica y la automatización de procesos. En los últimos años, la inteligencia artificial forma parte de esta evolución.

Dentro del presente trabajo se propone el uso de la visión por computadora, que es parte de la inteligencia artificial, para el control de una mano robótica, replicando los movimientos en tiempo real, creando así un mando a distancia. Con el desarrollo de este sistema de visión artificial, se pretende proporcionar una herramienta tecnológica para la manipulación de sustancias tóxicas evitando el contacto directo entre los operarios y dichas sustancias. Reduciendo también el porcentaje de accidentes de trabajo por agentes causantes como sustancias químicas, plaguicidas, materiales y radiaciones. Según el agente causante, estos compuestos representaron el 5.26 % de los accidentes de trabajo en el Perú durante el año 2022. (Ministerio de Trabajo y Promoción del Empleo, 2023)

## ÍNDICE GENERAL

	<u>Página</u>
DEDICATORIA .....	2
AGRADECIMIENTO .....	3
RESUMEN .....	4
ABSTRACT.....	5
INTRODUCCIÓN .....	6
ÍNDICE GENERAL .....	7
INDICE DE FIGURAS.....	11
INDICE DE TABLAS .....	14
LISTA DE ABREVIATURAS .....	15
CAPITULO I PLANTEAMIENTO TEÓRICO .....	16
1. PLANTEAMIENTO TEORICO .....	16
1.1. Descripción del problema de investigación.....	16
1.1.1. Enunciado de la investigación.....	16
1.1.2. Interrogantes de la investigación.....	16
1.1.2.1. Interrogante principal .....	16
1.1.2.2. Interrogantes específicas .....	16
1.1.3. Descripción del problema.....	17
1.2. Área de conocimiento .....	18
1.3. Antecedentes.....	19
1.3.1. Ámbito nacional .....	19
1.3.2. Ámbito internacional.....	20
1.4. Objetivos.....	21
1.4.1. Objetivo General .....	21
1.4.2. Objetivos Específicos .....	21

1.5. Justificación .....	22
1.6. Alcances.....	23
1.7. Limitaciones .....	23
CAPITULO II: MARCO TEORICO .....	25
2. MARCO TEÓRICO .....	25
2.1. Visión Artificial.....	25
2.1.1. Elementos de los Sistemas de Visión Artificial .....	26
2.1.1.1. Cámaras y Sistema de Captura.....	26
2.1.1.2. Sistema de Procesamiento Digital.....	26
2.1.1.3. Interfaz de Usuario .....	27
2.1.1.4. Sistema de Comunicación y Control.....	27
2.2. Anatomía de la mano humana .....	28
2.2.1. Estructura ósea .....	28
2.2.2. Articulaciones de la mano .....	30
2.2.3. Biomecánica de la mano .....	30
2.2.4. Movimientos de los dedos.....	30
2.2.5. Tipos de agarre .....	32
2.3. Mano Robótica.....	35
2.3.1. Prototipos de manos robóticas.....	35
2.3.2. Robot InMoov .....	36
2.3.2.1. Partes de la mano y antebrazo .....	38
2.4. Seguridad y Manipulación de Sustancias Tóxicas.....	39
2.4.1. Trasvase de líquidos .....	39
2.4.2. Mezcla o adición de un producto .....	39
2.5. Impresión 3D .....	39
2.6. Herramientas Software Para Utilizar .....	40
2.6.1. Python.....	40

2.6.2.	OpenCV.....	40
2.6.3.	MediaPipe.....	41
	CAPITULO III DISEÑO E IMPLEMENTACION.....	42
3.	DISEÑO DEL SISTEMA DE VISIÓN ARTIFICIAL .....	42
3.1.	Diagrama de bloques del sistema de visión artificial .....	42
3.2.	Selección del Prototipo .....	43
3.3.	Etapas del Sistema .....	46
3.3.1.	Diagrama de Bloques .....	46
3.3.2.	Etapa 1 – Identificación de la mano humana .....	47
3.3.3.	Etapa 2 – Programación de los Hand LandMarks.....	48
3.3.4.	Etapa 3 – Movimiento de los dedos .....	49
3.3.5.	Etapa 4 – Creación de la Interfaz de Usuario.....	49
3.4.	Implementación del Sistema.....	50
3.4.1.	Software .....	50
3.4.1.1.	Bases de programación.....	50
3.4.1.2.	Etapa 1 - Identificación de la mano humana .....	52
3.4.1.3.	Etapa 2 - Programación de los Hand LandMarks .....	60
3.4.1.4.	Etapa 3 – Movimiento de los dedos .....	62
3.4.1.5.	Etapa 4 – Creación de la interfaz de usuario .....	63
3.4.2.	Hardware .....	66
3.4.2.1.	Selección de la placa de desarrollo .....	66
3.4.2.2.	Servomotores.....	68
3.4.2.3.	Alimentación .....	74
3.4.2.4.	Impresión en 3D .....	75
3.4.2.5.	Armado de la mano .....	77
3.5.	Lista de Materiales.....	80
	CAPITULO IV PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS .....	84

4.	PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS.....	84
4.1.	Etapa 1 – Identificación de la mano humana.....	84
4.2.	Etapa 2 – Programación de los Hand LandMarks .....	86
4.3.	Etapa 3 – Movimiento de los dedos.....	90
4.4.	Etapa 4 – Creación de la interfaz de usuario .....	96
4.5.	Pruebas con recipientes de laboratorio .....	103
	CONCLUSIONES .....	105
	RECOMENDACIONES.....	106
	REFERENCIA.....	107
	ANEXOS .....	109



## INDICE DE FIGURAS

<b>Figura 1</b> <i>Mano Inteligente de robot de Festo, BionicSoftHand</i> .....	18
<b>Figura 2</b> <i>Huesos de la mano</i> .....	28
<b>Figura 3</b> <i>Vista anterior de la mano.</i> .....	29
<b>Figura 4</b> <i>Movimientos de los dedos de la mano humana</i> .....	32
<b>Figura 5</b> <i>Tipo de agarre</i> .....	32
<b>Figura 6</b> <i>Dinamómetro T.K.K. 540 Grip-D</i> .....	34
<b>Figura 7</b> <i>Robot InMoov impreso en 3D y ensamblado</i> .....	37
<b>Figura 8</b> <i>Vista 3D del conjunto mano y antebrazo</i> .....	38
<b>Figura 9</b> <i>Diagrama de bloques del sistema de visión artificial para la mano robótica</i> ..	42
<b>Figura 10</b> <i>Mano Robótica Hackberry</i> .....	43
<b>Figura 11</b> <i>Mano robótica Hero Arm</i> .....	44
<b>Figura 12</b> <i>Diagrama de bloques de las etapas del sistema de visión artificial</i> .....	47
<b>Figura 13</b> <i>Hand LandMarks</i> .....	48
<b>Figura 14</b> <i>Ventana principal para programación en Pycharm</i> .....	51
<b>Figura 15</b> <i>Ventana principal para programación en Arduino</i> .....	52
<b>Figura 16</b> <i>Instalación de Bibliotecas nuevas en Pycharm</i> .....	53
<b>Figura 17</b> <i>Detección de Hand LandMarks</i> .....	54
<b>Figura 18</b> <i>Evaluación de los Hand LandMarks en la coordenada Y</i> .....	62
<b>Figura 19</b> <i>Creación de archivo Python File</i> .....	64
<b>Figura 20</b> <i>Interfaz gráfica para el sistema de visión artificial</i> .....	65
<b>Figura 21</b> <i>Servomotor MG995</i> .....	69
<b>Figura 22</b> <i>Esquemático del Circuito</i> .....	71
<b>Figura 23</b> <i>Regulador de voltaje reductor Pololu D24V50F5</i> .....	74
<b>Figura 24</b> <i>Archivos STL en la página web de InMoov</i> .....	76
<b>Figura 25</b> <i>Piezas impresas en 3D</i> .....	77
<b>Figura 26</b> <i>Tres articulaciones en un dedo</i> .....	78
<b>Figura 27</b> <i>Unión de los dedos a la palma</i> .....	78
<b>Figura 28</b> <i>Hilo nylon o de pescar que transmitirá el movimiento de los servos</i> .....	79
<b>Figura 29</b> <i>Palma de la mano robótica armada</i> .....	79

<b>Figura 30</b> <i>Instalación de los servomotores MG995</i> .....	80
<b>Figura 31</b> <i>Prototipo de mano robótica ensamblado</i> .....	80
<b>Figura 32</b> <i>Identificación de la mano humana en la etapa 1</i> .....	85
<b>Figura 33</b> <i>Identificación de la mano humana en comparación a la mano robótica</i> .....	86
<b>Figura 34</b> <i>Dos manos identificadas</i> .....	86
<b>Figura 35</b> <i>Todos los dedos abiertos, el mensaje enviado será \$11111</i> .....	87
<b>Figura 36</b> <i>Algunos dedos abiertos, el mensaje enviado será \$01111</i> .....	87
<b>Figura 37</b> <i>Algunos dedos abiertos, el mensaje enviado será \$00111</i> .....	88
<b>Figura 38</b> <i>Algunos dedos abiertos, el mensaje enviado será \$00011</i> .....	88
<b>Figura 39</b> <i>Algunos dedos abiertos, el mensaje enviado será \$00001</i> .....	89
<b>Figura 40</b> <i>Algunos dedos abiertos, el mensaje enviado será \$01001</i> .....	89
<b>Figura 41</b> <i>Todos los dedos cerrados, el mensaje enviado será \$00000</i> .....	90
<b>Figura 42</b> <i>Todos los dedos abiertos en la interfaz y el prototipo</i> .....	91
<b>Figura 43</b> <i>Dedo pulgar cerrado en la interfaz y el prototipo</i> .....	91
<b>Figura 44</b> <i>Dedo índice cerrado en la interfaz y el prototipo</i> .....	92
<b>Figura 45</b> <i>Dedo medio cerrado en la interfaz y el prototipo</i> .....	92
<b>Figura 46</b> <i>Dedo anular cerrado en la interfaz y el prototipo</i> .....	93
<b>Figura 47</b> <i>Dedo meñique cerrado en la interfaz y el prototipo</i> .....	93
<b>Figura 48</b> <i>Todos los dedos cerrados en la interfaz y el prototipo</i> .....	94
<b>Figura 49</b> <i>Agarres tipo pinza, pulgar e índice y medio en la interfaz y el prototipo</i> .....	94
<b>Figura 50</b> <i>Gestos con la mano en la interfaz y el prototipo</i> .....	95
<b>Figura 51</b> <i>Interfaz de usuario</i> .....	96
<b>Figura 52</b> <i>Interfaz de usuario en funcionamiento iniciado el control por VA</i> .....	97
<b>Figura 53</b> <i>Todos los dedos abiertos en la interfaz</i> .....	97
<b>Figura 54</b> <i>Dedo pulgar cerrado en la interfaz</i> .....	98
<b>Figura 55</b> <i>Dedo pulgar cerrado en la interfaz</i> .....	98
<b>Figura 56</b> <i>Dedo medio cerrado en la interfaz</i> .....	99
<b>Figura 57</b> <i>Dedo anular cerrado en la interfaz</i> .....	99
<b>Figura 58</b> <i>Dedo pulgar cerrado en la interfaz</i> .....	100
<b>Figura 59</b> <i>Todos los dedos cerrados en la interfaz</i> .....	100
<b>Figura 60</b> <i>Gestos con los dedos (agarre tipo pinza) en la interfaz</i> .....	101

<b>Figura 61</b> <i>Gestos con los dedos (agarre tipo pinza) en la interfaz</i> .....	101
<b>Figura 62</b> <i>Gestos con los dedos en la interfaz</i> .....	102
<b>Figura 63</b> <i>Gestos con los dedos en la interfaz</i> .....	102
<b>Figura 64</b> <i>Mano robótica sujetando un vaso precipitado</i> .....	103
<b>Figura 65</b> <i>Mano robótica sujetando un vaso precipitado con líquido</i> .....	103
<b>Figura 66</b> <i>Mano robótica sujetando una probeta</i> .....	104
<b>Figura 67</b> <i>Mano robótica sujetando una probeta con líquido</i> .....	104



## INDICE DE TABLAS

<b>Tabla 1</b> <i>Ángulos de movimiento respecto al tipo de movimiento y grados de libertad ...</i>	31
Tabla 2 <i>Prototipos de manos robóticas para impresión en 3D .....</i>	45
Tabla 3 <i>Valores enviados a través de la comunicación serial.....</i>	60
Tabla 4 <i>Cuadro comparativo Arduino .....</i>	66
Tabla 5 <i>Selección de pines para los servomotores .....</i>	67
Tabla 6 <i>Especificaciones Arduino Uno.....</i>	68
Tabla 7 <i>Especificaciones Servomotor MG995.....</i>	71
Tabla 8 <i>Características de materiales de impresión 3D .....</i>	75
Tabla 9 <i>Componentes de la mano robótica .....</i>	76
Tabla 10 <i>Cuadro de lista de materiales .....</i>	81



## LISTA DE ABREVIATURAS

ABS:	Acrilonitrilo Butadieno Estireno
API:	Programación de Aplicaciones
ASCII:	American Standard Code for Information Interchange
BAUD:	Baudio
CCD:	Charge Coupled Device
CLI:	Interfaz de Línea de Comandos
CMOS:	Complementary Metal Oxide Semiconductor
CNC:	Control Numérico Computarizado
DIP:	Articulación Interfalángica Distal
EMG:	Electromiografía
IP:	articulación Interfalángica
KB:	Kilobyte
MCM:	Articulación Metacarpofalángicas
MHZ:	Megahercio
MP:	Articulación Metacarpiana
OCDE:	Organización para la Cooperación y el Desarrollo Económicos
OPENCV:	Open Computer Vision
PIP:	Articulación Interfalángica proximal
PLA:	Ácido Poli láctico
PWM:	Modulación de Ancho de Pulso
RAM:	Memoria de Acceso Aleatorio
RGB:	Red Green Blue
STL:	STereoLithography
UI:	Interfaz de Usuario
UV:	Ultravioleta
VA:	Visión Artificial

## CAPITULO I: PLANTEAMIENTO TEÓRICO

### 1. PLANTEAMIENTO TEORICO

#### 1.1. Descripción del problema de investigación

##### 1.1.1. Enunciado de la investigación

“Diseño de un sistema de visión artificial para el uso de una mano robótica en la manipulación de sustancias tóxicas”.

##### 1.1.2. Interrogantes de la investigación

###### 1.1.2.1. Interrogante principal

¿Cómo puede diseñarse un sistema de visión artificial para controlar una mano robótica capaz de manipular sustancias tóxicas y peligrosas, minimizando el riesgo de accidentes para los operarios en entornos industriales y de investigación?

###### 1.1.2.2. Interrogantes específicas

- ¿Cómo se puede utilizar la biblioteca de código abierto MediaPipe para desarrollar un sistema de visión artificial que permita imitar en tiempo real los movimientos de una mano humana real?
- ¿Cuáles son las características clave que debe tener una mano robótica de los diferentes prototipos disponibles en términos de funcionalidad, costos y accesibilidad para la investigación?
- ¿Cuál es la eficacia de la impresión en 3D utilizando materiales termoplásticos por filamento y resina en comparación con otros métodos de fabricación para la creación de una mano robótica?
- ¿Cuáles son las consideraciones clave en términos de precisión y seguridad al implementar el circuito electrónico para el control de movimiento de la mano robótica?

### 1.1.3. Descripción del problema

Las industrias químicas y farmacéuticas desempeñan un papel crucial en la producción de una amplia gama de productos, desde medicamentos hasta productos químicos industriales en los cuales existen procesos productivos que involucran la manipulación de sustancias tóxicas. Dentro de estos procedimientos, los operarios se encuentran expuestos a sufrir algún tipo de accidente o enfermedad laboral por la manipulación incorrecta de estas sustancias.

A pesar de los avances en tecnología y automatización en el ámbito industrial, todavía existe áreas donde la intervención humana es esencial debido a la precisión y flexibilidad requeridas en ciertos procesos. Estas tareas manuales, aunque necesarias, también presentan riesgos significativos para la salud y la seguridad de los trabajadores. Los accidentes laborales en la manipulación de sustancias tóxicas pueden tener consecuencias graves, incluyendo lesiones, quemaduras químicas y enfermedades respiratorias o dermatológicas.

Accidentes laborales en estos procedimientos, pueden evitarse con la manipulación indirecta de estos químicos. En este sentido, la robótica ha tomado mucha importancia en el área de las industrias manufactureras, automatizando la mayoría de los sistemas, permitiendo así el mejoramiento de la calidad de los productos, disminución de pasos en el proceso de producción, reducción de riesgos personales, mayor productividad e incremento de la utilización de máquinas.

Expuesto lo anterior, es importante resaltar que todavía existen trabajos industriales en donde las manos del hombre son esenciales para el acabado de la producción, debido a la precisión y flexibilidad que esta posee, pero se encuentran en peligro, ya que estas tareas pueden ocasionar lesiones, quemaduras o enfermedades a corto y largo plazo.

Para esto, se propone el desarrollo de un sistema de manipulación remota, utilizando la visión artificial en lugar de sensores, sustituyendo así, el contacto directo entre las sustancias peligrosas y el ser humano en entornos industriales, previniendo posibles lesiones que pueden ocurrir al momento de la manipulación de estas. La visión artificial permitirá la detección y seguimiento preciso de objetos y movimiento, lo que facilitará la manipulación remota de sustancias peligrosas con la precisión y la seguridad. Además, se explorará la viabilidad de integrar esta tecnología en una mano robótica, permitiendo así la manipulación remota de objetos con la destreza y flexibilidad de una mano humana.

En la figura 1 aparece la mano robótica BionicSoftHand diseñada e implementada por Festo con control remoto e imitación de los movimientos en tiempo real. El resultado de esta presente investigación deberá ser similar, sin embargo, se hará uso de la visión por computadora.

### Figura 1

*Mano Inteligente de robot de Festo, BionicSoftHand*



*Nota:* Esta figura muestra la BionicSoftHand de Festo, una mano robótica avanzada diseñada para replicar la destreza y el movimiento de una mano humana. Incorpora tecnología de punta para proporcionar una interacción intuitiva y versátil en aplicaciones de automatización y robótica. *Adaptado de Festo.*

## 1.2. Área de conocimiento

De acuerdo con la OCDE (2007), la presente investigación se clasifica en función de las áreas y subáreas del conocimiento como se presenta a continuación:

- ÁREA: Ingeniería y Tecnología
- SUB-ÁREA: Ingeniería Médica
- LÍNEA: Tecnología Médica de Laboratorio

Es importante resaltar la naturaleza interdisciplinaria del proyecto, que combina aspectos de la visión artificial (procesamiento de imágenes, inteligencia artificial), la electrónica (diseño de circuitos de control), la mecánica (diseño e impresión 3D de la mano robótica) y la programación (desarrollo de algoritmos), todas ellas áreas cubiertas por la ingeniería mecatrónica.

### 1.3. Antecedentes

En el desarrollo de esta investigación se tomarán como referencia estudios e investigaciones tanto nacionales e internacionales.

#### 1.3.1. Ámbito nacional

El primer antecedente planteado por Antony Gamonal, ingeniero mecatrónico por la Universidad tecnológica del Perú en el año 2020, titula: “DISEÑO DE UN SISTEMA DE VISION ARTIFICIAL PARA DETERMINAR LA CALIDAD DE MANDARINAS”, en este proyecto de tesis se diseña un algoritmo de clasificación, haciendo uso del software Matlab, consistiendo en la adquisición de imágenes a través de una cámara web en tiempo real, procesamiento de imágenes, segmentación, reconocimiento e interpretación. Se consiguió la correcta clasificación de las mandarinas según la forma, el color y los defectos aceptables de las frutas. Para este proyecto es primordial el empleo de una cámara que realice el captado de imágenes a procesar, además se crea una interfaz de datos y control del proceso de determinación de calidad. Al igual que en esta tesis, se realizará una interfaz de datos y control de la mano robótica al imitar en tiempo real los movimientos de la mano que captará la cámara web. Esta investigación nos permitirá identificar los elementos de un sistema de visión artificial, así como el empleo de software de programación para el proceso de identificación de imágenes y la creación de una interfaz de usuario para la interacción del sistema y el operario. El sistema de visión artificial diseñado partirá de la adquisición de imágenes, procesamiento y toma de decisiones al momento de hacer un movimiento a imitar en tiempo real. También se hace referencia a “DISEÑO Y CONTROL DE UNA MANO ROBÓTICA PARA FUNCION MOTORA DE PERSONAS DISCAPACITADAS” por tener el control de una mano robótica, que además puede usarse para uso de laboratorio pero con un control totalmente distinto al uso de la visión artificial, este proyecto de investigación consistió en diseñar y controlar un prototipo de mano robótica, usando la plataforma de Arduino a través de una comunicación bluetooth y por medio de un guante instrumentado para la imitación de movimientos de la mano, en él, usan un prototipo de robot humanoide de código abierto, el cual puede ser impreso y ensamblado por personas en sus propios hogares. Los hallazgos de esta tesis ayudarán a la actual investigación en el proceso de impresión 3D, ensamblado de la mano robótica y el control de esta misma, se hará uso del prototipo de mano robótica InMoov de código abierto, además de usar como placa de microcontrolador a Arduino para la programación y control de los

movimientos de la mano robótica, este proyecto fue presentado a la Escuela Profesional de Ingeniería Mecánica, de la Universidad Continental por Julinhio Reynoso, en el año 2021. Como tercera referencia el proyecto de investigación titulado “DISEÑO E IMPLEMENTACION DE UN SISTEMA DE CONTROL PARA UNA PROTESIS MIOELECTRICA DE MIEMBRO SUPERIOR CON CONTROL DE FUERZA Y POSICION” de la misma manera que en el anterior antecedente se resaltará el sistema de control de fuerza y posición. Aquí se diseña un algoritmo autónomo para el control de fuerza y posición de la mano y dedos y la selección de los dispositivos de control, así como el diseño de las tarjetas electrónicas. En esta resalta la implementación de sensores de posición y de fuerza, permitiendo así un control más preciso y exacto en el momento de manipular objetos, además de algunos tipos de agarre conjunto de la mano robótica. Se hacen uso de sensores mioeléctricos, que, en comparación con el uso de la visión artificial, se reducirían considerablemente los costos. El proyecto va planeado específicamente para las personas que pudieron tener una amputación en el miembro superior, sin embargo, podemos hacer una comparación de los resultados con la implementación de un sistema de visión artificial. Esta investigación fue presentada a la Escuela de Posgrado de la Universidad Católica del Perú por Giancarlo Salazar en el año 2020.

### **1.3.2. Ámbito internacional**

En el ámbito internacional se da referencia a 3 trabajos de investigación, el primero es una tesis titulada “HAND POSE ESTIMATION AND GESTURE DETECTION FROM WEBCAM IMAGES”, este trabajo de investigación consiste en estimar la postura de la mano y detectar los gestos usando el enfoque clásico de programación para visión artificial y las librerías de MediaPipe, posteriormente haciendo una comparación entre ambos enfoques, esta investigación nos permitirá justificar el uso de estas librerías pertenecientes a OpenCV las cuales también se harán uso para la detección de manos y la programación de los Hand LandMarks (puntos de referencia en la mano identificada por la cámara), para el mejor desenvolvimiento de la programación y detección de manos para la imitación de sus movimientos en tiempo real, esta tesis fue presentada a la Universidad Técnica de Múnich por Ashish Khanal en el año 2021. La segunda referencia tiene por título “SISTEMA DE CONTROL DE MANO ROBÓTICA REPLICADORA DE MOVIMIENTOS PARA MANIPULAR SUSTANCIAS QUÍMICAS”, este trabajo de investigación desarrolla el control de la mano robótica incluyendo modelos matemáticos para 4

determinados tipos de agarre, estimando las funciones de transferencia e implementando controladores proporcionales – integrales, implementándolo también en el robot de código abierto InMoov, impreso tridimensionalmente, haciendo uso de servomotores y Arduino para su programación. Se podrán realizar comparaciones con los tiempos de respuesta y la precisión entre el uso de controladores clásicos y el de visión artificial. Pertenece al artículo científico de la Universidad Católica Boliviana San Pablo, del año 2018. El tercer antecedente titula “MANEJO DE SUSTANCIAS TOXICAS POR MEDIO DE UNA MANO INALAMBRICA”, el problema detectado para la realización de esta investigación son los accidentes laborales, por ello se toma esta referencia al dar una solución similar, se construye una mano robótica y se implementa un control por sensores flex inalámbricamente por un guante. El fin de esta investigación también es imitar en tiempo real los movimientos de la mano haciendo uso de sensores que cambian su resistividad al ser flexionados. Se logra el objetivo del proyecto dando una alternativa de solución así al problema de los accidentes laborales. En del desarrollo de la presente investigación se reemplazará el uso de estos sensores por una cámara que será el inicio de la detección de imágenes para su posterior procesamiento haciendo uso de la visión artificial. Se espera poder incorporar el uso de la visión artificial en esta investigación, logrando así una innovación en este campo y reemplazo de sensores y su mantenimiento por una cámara. Este artículo de investigación fue presentado en la Escuela Politécnica Nacional Facultad de Mecánica por Jhonny Orozco, William Venegas e Ivan Zambrano en el año 2018.

#### **1.4. Objetivos**

##### **1.4.1. Objetivo General**

Diseñar un sistema de visión artificial para controlar los movimientos de una mano robótica para la manipulación de sustancias toxicas.

##### **1.4.2. Objetivos Específicos**

1. Crear un sistema de visión artificial, haciendo uso de la biblioteca de código abierto Media pipe, desarrollada por Google, a través de la inteligencia artificial, para poder realizar una imitación de los movimientos de una mano humana real.

2. Comparar y determinar un prototipo de mano robótica entre las varias que existen actualmente, tanto las de libre uso (o código abierto) como las de pago, para tener una mejor funcionalidad con las características necesarias en el uso de esta en la investigación.
3. Imprimir en 3D la mano robótica teniendo en cuenta la rapidez de prototipado y flexibilidad en el diseño. Toda la impresión se realizará de material termoplástico por filamento (y resina que se solidifica mediante luz UV para algunos componentes) o impresión FDM (Modelado por deposición fundida), es decir capa por capa.
4. Implementar un circuito electrónico para el control de movimiento de los servomotores de la mano robótica utilizando el software PROTEUS.

### 1.5. Justificación

Para describir de mejor manera el por qué y para que de la investigación, enmarcando el impacto y efectos que la solución del problema traerá para el sector productivo, se justifica en cuatro niveles:

- Nivel académico: El desarrollo de esta investigación aportará al desarrollo e innovaciones en el ámbito de la seguridad y manipulación a distancia de sustancias tóxicas y peligrosas al contacto con el ser humano, desarrollando avances tecnológicos como la inteligencia y la visión artificiales para el uso ya mencionado.
- Nivel económico: Al desarrollar este sistema de control remoto, se podrá impulsar a la automatización de los procesos químicos y farmacéuticos, generando así, nuevos puestos de trabajos y proyectos de investigación e inversión.
- Nivel social: El interés principal de la investigación, viene ligado de la protección y bienestar de los trabajadores manipuladores y personal directamente expuesto a sustancias peligrosas, las cuales requieren un grado de protección. El impacto de esta investigación amplia la seguridad laboral y protección del medio ambiente
- Nivel ético: Con el fin de ampliar las posibilidades de procesos de automatización en la industria farmacéutica y química, bajo la percepción de la filosofía, se contribuirá al poner la tecnología al servicio del desarrollo y conocimiento humano, además de la seguridad y el bienestar de los trabajadores expuestos a sustancias peligrosas.

La manipulación de productos dañinos o peligrosos puede dar lugar a accidentes con consecuencias de distinta índole.

El proceso en el que se suele tener mayor número de accidentes es el trasvase entre recipientes, que pueden dar lugar a salpicaduras, contactos dérmicos, intoxicaciones y quemaduras. Se debe evitar pasar un líquido de un recipiente a otro de forma manual, y es fundamental que los trabajadores estén plenamente capacitados sobre los riesgos a los que están expuestos al realizar este tipo de operaciones, y realizarlas en condiciones óptimas de seguridad.

El desarrollo de este proyecto es beneficioso para la seguridad de los trabajadores que realizan acciones riesgosas con la manipulación directa de diversos tipos de sustancias que pueden traer consigo accidentes laborales, así como enfermedades a corto y largo plazo. La diferencia con otros prototipos similares es que este sistema no usa ningún tipo de sensor (fotoeléctricos o mioeléctricos), además no requerirá de realizar modelos matemáticos para el control de estas, ya que la visión artificial reemplazará este procedimiento.

### **1.6. Alcances**

El estudio de esta tesis abarcará con el tema de visión artificial para la identificación de la mano humana a través de una simple cámara, luego, el algoritmo diseñado en Python realizará las acciones con los actuadores de cada dedo de la mano robótica impresa con tecnología 3D. Todos los movimientos realizados por la mano humana serán imitados por la mano robótica en tiempo real.

### **1.7. Limitaciones**

El diseño del prototipo de mano robótica no se desarrollará desde cero; en su lugar, se empleará un modelo previamente creado para integrar el sistema de visión artificial.

En lugar de llevar a cabo un filtrado de imágenes, se utilizará la librería de detección de objetos 3D en tiempo real MediaPipe en Python.

El sistema de visión artificial se centrará en una interfaz de usuario (UI) donde se pueda mostrar los datos en video como posición, movimiento de apertura o cierre de la mano haciendo

uso de estas herramientas para lograr controlar de forma remota y en tiempo real un prototipo de la mano robótica.



## CAPITULO II: MARCO TEORICO

### 2. MARCO TEÓRICO

#### 2.1. Visión Artificial

La visión artificial es una parte de la inteligencia artificial y también una disciplina que incluye métodos de adquirir, procesar y analizar imágenes del mundo real con el fin de recopilar información que pueda ser procesada por una máquina (Borrella P. 2022), simulando así el proceso de visión biológico de los humanos. La visión artificial nos permite crear algoritmos y aplicaciones para interpretar el significado de una imagen, con esto podemos obtener información de un objeto espacial 3D a partir de su adquisición y procesamiento de una o varias imágenes digitales 2D de dicho objeto.

Su fondo va de matrices a dimensiones considerablemente grandes y su objetivo principal es que una máquina tenga la capacidad visual del ser humano de un mundo tridimensional a partir de una o varias imágenes bidimensionales, o al menos aproximarse. El uso de la visión artificial ha progresado en los últimos 20 años, con mejoras en precisión y uso del internet de las cosas para el control de procesos remotamente.

La visión artificial es una disciplina cuyo principal objetivo es procesar y analizar imágenes del mundo real, con la finalidad de que estas puedan ser entendidas y tratadas por un ordenador. Los procesos de la visión artificial pueden agruparse en 3 niveles según su complicación e implementación (Gonzales y Woods, 2002):

- Procesos de nivel bajo: se refieren a operaciones primitivas como es la captura o adquisición de imágenes y el preprocesamiento de imágenes para reducir el ruido, mejorar el contraste y nitidez de la imagen. Una característica importante de estos procesos es que sus entradas y salidas o resultados son imágenes.
- Procesos de nivel medio: se refieren a operaciones como la segmentación y la descripción de los objetos individuales presentes en una escena para reducirlos a una forma adecuada para su tratamiento informático, reconocimiento y clasificación de estos. Un rasgo importante de estos procesos es que sus entradas generalmente son imágenes, pero sus resultados son características extraídas de estas, como bordes, contornos o la identidad de los objetos por su forma en las imágenes de entrada.

- Procesos de nivel superior: se refieren a operaciones que reconocen un conjunto de objetos en la imagen y realizan funciones cognitivas que normalmente se asocian con la visión, conocido como el proceso de interpretación.

Aunque estas etapas son aparentemente secuenciales, esto no es necesario y pueden aparecer interacciones entre los diferentes niveles incluyendo la retroalimentación de los niveles altos a los inferiores (Viera, G. 2017).

### **2.1.1. Elementos de los Sistemas de Visión Artificial**

Para tener un sistema de visión artificial es necesario complementar un conjunto de subsistemas que trabajen en conjunto para dar un sistema más grande.

#### **2.1.1.1. Cámaras y Sistema de Captura**

Una cámara digital es una cámara fotográfica que captura y almacena imágenes de manera digital gracias a un dispositivo llamado sensor. Generalmente las cámaras digitales usan dos tipos de sensores, los sensores CCD y los sensores CMOS, estos sensores convierten la luz en señales eléctricas que luego se digitalizan para producir imágenes y videos que posteriormente pueden ser mostrados en la pantalla de una computadora e incluso transmitirlos a través de internet. La calidad de la resolución de una cámara digital no depende únicamente del número y distribución de píxeles, sino también de otros factores como las características del sensor y características del lente.

Para esta tesis se ha utilizado una cámara web, que es un tipo de cámara digital para el uso en red.

El sistema de captura de imágenes incluye el conjunto de dispositivos y software diseñados para adquirir, procesar y almacenar imágenes. Las principales aplicaciones que se desarrollan con IoT y Visión artificial pueden ser implementadas en educación, medicina, edificios inteligentes, sistemas de vigilancia de personas y vehículos, entre otros (Alvear, V. 2017). Algunos elementos de este sistema son: el dispositivo de captura (que incluye al sensor de imagen y los lentes de enfoque), el almacenamiento, el software de captura y visualización y las interfaces de usuario.

#### **2.1.1.2. Sistema de Procesamiento Digital**

El procesamiento de imágenes desempeña un papel fundamental en los sistemas de visión artificial. Este proceso implica una serie de técnicas y algoritmos que se aplican a las imágenes al extraer información significativa como detectar objetos, realizar mediciones, mejorar la calidad de imágenes y tomar decisiones basadas en la información visual. Estas técnicas se utilizan en una amplia gama de aplicaciones, algunas de estas técnicas de procesamiento de imágenes se dividen en dos grupos de métodos:

**Métodos de dominio espacial:** Hace referencia a las modificaciones que se le hacen a la imagen con manipulación directa de los píxeles.

**Métodos de dominio de frecuencia:** Hace referencia a las modificaciones hechas mediante la transformada de Fourier de una imagen (Gomez, J. 2007).

Para esta tesis el sistema de procesamiento digital viene con el desarrollo del algoritmo en Python haciendo uso de las herramientas de OpenCV y MediaPipe para la identificación de la mano con sus puntos estratégicos, y para la programación y accionamiento de los actuadores que harán posible el movimiento de los dedos de la mano robótica.

### **2.1.1.3. Interfaz de Usuario**

La finalidad de una aplicación de software es permitir a un usuario realizar una tarea, que puede ir desde la recuperación y despliegue de información, la captura y el procesamiento de datos, entre otras (Berzunza, G. 2008). La interfaz de usuario (UI) es una parte del software en donde el usuario interactúa directamente con el sistema, permitiendo el intercambio de información a través de elementos visuales como botones, menús, ventanas y gráficos, es importante para facilitar la interacción del usuario con el sistema o aplicación. Existen además otros tipos de interfaces como la programación de aplicaciones (API) o interfaz de línea de comandos (CLI) que se comunican de diferente manera a la UI.

En el sistema de visión artificial a desarrollar se incluirá una interfaz de usuario que permita a los operadores humanos interactuar con el sistema, configurar parámetros y revisar resultados. Una interfaz de usuario facilita la utilización efectiva del sistema y la toma de decisiones.

### **2.1.1.4. Sistema de Comunicación y Control**

La comunicación y el control son aspectos esenciales que permiten que el sistema funcione de manera efectiva y se integre en aplicaciones más amplias. Estos sistemas a menudo se utilizan en entornos en tiempo real y requieren una comunicación fluida entre los componentes del sistema y la capacidad de tomar decisiones basadas en la información visual. En el desarrollo de este proyecto la comunicación será en tiempo real utilizando una conexión cableada mediante USB con la tarjeta integrada Arduino que usa un microcontrolador desarrollado por Microchip Technology, que será el intermediario entre la computadora y los actuadores de la mano robótica (servomotores).

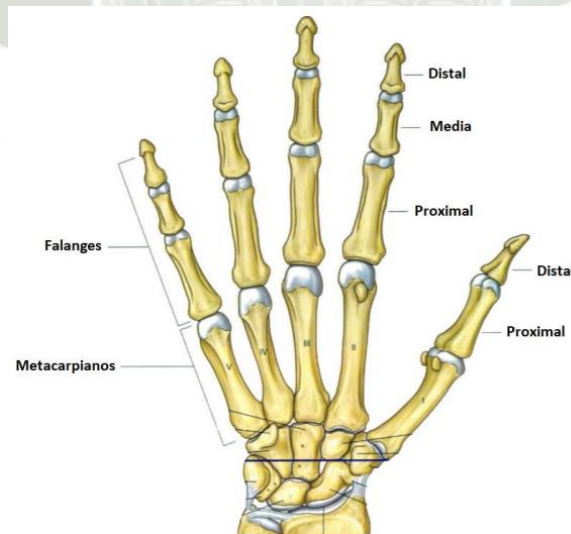
## 2.2. Anatomía de la mano humana

Se describe el comportamiento y los tipos de movimiento realizados por la mano humana.

### 2.2.1. Estructura ósea

Los metacarpianos de la palma de la mano se articulan con una falange proximal cada una, y estos a la vez con las siguientes falanges.

**Figura 2**  
*Huesos de la mano*



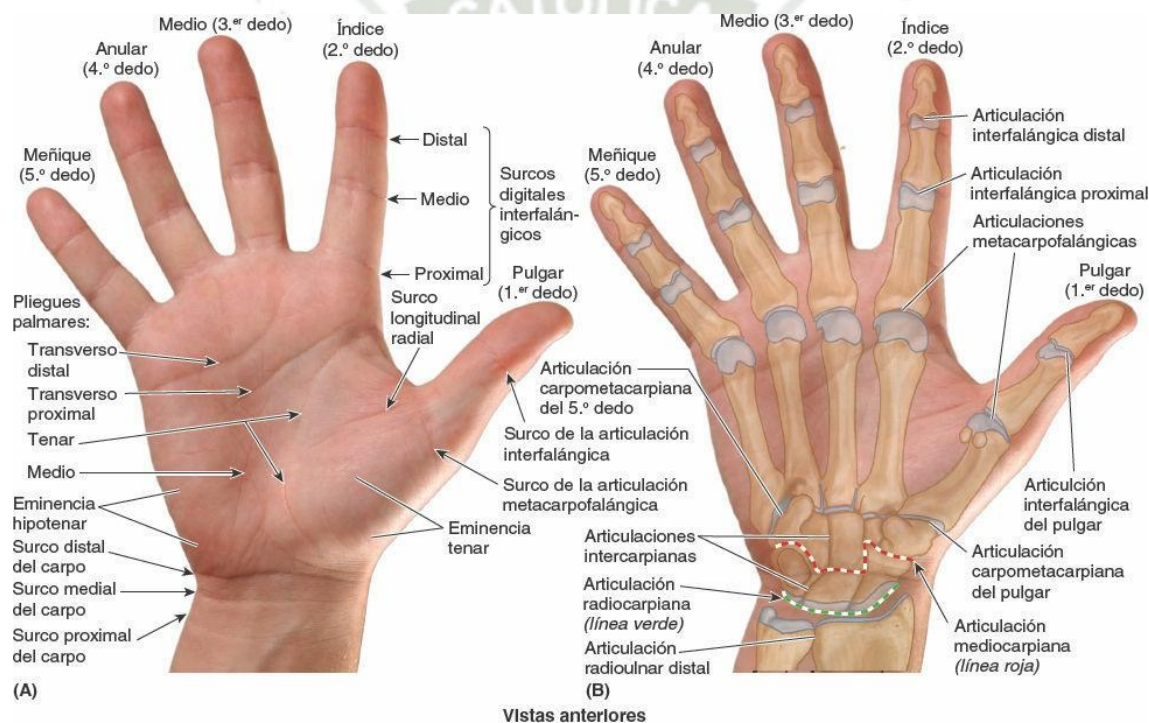
*Nota:* Esta figura muestra la estructura ósea de la mano humana, con anotaciones para las falanges (huesos de los dedos), metacarpianos (huesos de la palma) y las regiones distal, media y proximal de los dedos. La representación detallada ayuda en la comprensión de la compleja anatomía de la mano, fundamental tanto en la práctica clínica como en la educación médica. *Adaptado de Gray, H. (2020)*

Los dedos: meñique, anular, medio e índice cuentan con tres falanges, es decir, con tres articulaciones móviles: la articulación metacarpofalángica (MP), que une a los metacarpianos con la falange proximal; la articulación interfalángica proximal (PIP, por sus siglas en inglés), que une la falange proximal con la media; y la articulación interfalángica distal (DIP), que une la falange media con la distal tal como se muestra en la figura 2.

En el caso del pulgar, este solo cuenta con dos falanges (proximal y distal), articulados en MP y una articulación interfalángica (IP).

### Figura 3

Vista anterior de la mano. (A) Secciones y dígitos de la mano. (B) Huesos y articulaciones.



*Nota:* Esta imagen proporciona una visión detallada de la anatomía de la mano humana, mostrando tanto una vista superficial de los dígitos y pliegues palmares (A) como una vista esquemática de los huesos y articulaciones subyacentes (B). Es un recurso educativo útil para estudiar la complejidad de la estructura manual y sus funciones. Adaptado de "Anatomía de la mano", por H. Gray, (2020)

La muñeca está constituida por los huesos del carpo, la palma por 5 huesos del metacarpo y las falanges constituyen los dedos. Las articulaciones igualmente se dividen en: articulaciones del carpo, carpometacarpianas, metacarpofalángicas e interfalángicas. Los movimientos de los dedos se dan gracias a los tendones flexores y extensores que contiene el capuchón extensor.

### 2.2.2. Articulaciones de la mano

Las articulaciones de la mano incluyen un grupo de estas, corresponden a:

- Articulación de la muñeca: la articulación de conecta el antebrazo con la mano y permite la flexión y extensión de la mano.
- Articulación del carpo: Conecta los huesos del carpo con los huesos del antebrazo y permite la flexión, extensión, abducción y aducción de la mano.
- Articulaciones metacarpofalángicas (MCM): Conectan los huesos metacarpianos con las falanges proximales de los dedos y permiten la flexión, abducción y aducción de los dedos.
- Articulaciones interfalángicas (IP): Conectan las falanges proximales con las falanges distales de los dedos y permiten la flexión y extensión de los dedos.

### 2.2.3. Biomecánica de la mano

La mano es una estructura compleja que consta de huesos, músculos, tendones, ligamentos y nervios. Tiene una gran capacidad de movimiento y precisión, lo que la hace esencial para muchas actividades para el ser humano. La biomecánica de la mano también se utiliza en la rehabilitación de lesiones de la mano y la muñeca.

Cinemáticamente, la mano es una cadena de eslabones cinemáticos cuyo origen está en la muñeca y final en los dedos. Tiene un total de 21 grados de libertad, por lo que su construcción es una de las más complejas para la biomecánica.

### 2.2.4. Movimientos de los dedos

La mano presenta 4 movimientos principales: extensión, flexión, abducción y aducción (Gray, H. 2020). Los movimientos de flexión y extensión cumplen la función de agarre y manipulación de objetos. El pulgar, puede hacer realizar un movimiento de oposición aparte de los otros 4 movimientos. Los ángulos de amplitud para flexión son iguales en los dedos índice, medio, anular y meñique. Para una mejor visualización se puede observar la figura 4.

**Tabla 2**

*Ángulos de movimiento respecto al tipo de movimiento y grados de libertad*

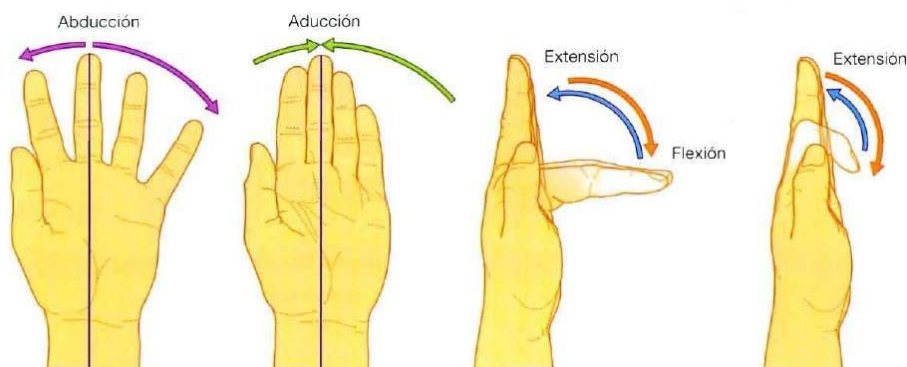
<b>Dedo</b>	<b>Tipo de movimiento</b>		<b>Rango angular</b>	<b>DOF</b>
<b>Índice</b>	Aducción – Abducción		15°	1
	Desde falange distal		15°-90°	1
	Flexión – extensión	Desde falange media	25°-90°	1
		Desde falange proximal	60°-90°	1
<b>Medio</b>	Aducción – Abducción		10°	1
	Desde falange distal		15°-90°	1
	Flexión – extensión	Desde falange media	25°-90°	1
		Desde falange proximal	60°-90°	1
<b>Anular</b>	Aducción – Abducción		26°	1
	Desde falange distal		15°-90°	1
	Flexión – extensión	Desde falange media	25°-90°	1
		Desde falange proximal	60°-90°	1
<b>Meñique</b>	Aducción – Abducción		35°	1
	Desde falange distal		15°-90°	1
	Flexión – extensión	Desde falange media	25°-90°	1
		Desde falange proximal	60°-90°	1
<b>Pulgar</b>	Flexión – extensión		80°	2
	Aducción – Abducción		35°-40°	1
	Oposición		45°-60°	2

*Nota:* Esta tabla resume los ángulos de movimiento y grados de libertad (DOF) para cada dedo de la mano humana. Cada dedo, excepto el pulgar, tiene tres tipos de movimiento en la aducción-abducción y flexión-extensión, con rangos angulares específicos. El pulgar presenta movimientos adicionales de oposición y tiene un total de dos DOF en flexión-extensión y uno en aducción-abducción. *Adaptado de "Ángulos de movimiento de la mano", por C. Fuentes, 2020.*

Los cuatro dedos: índice, medio, anular y meñique, presentan similares grados de flexión cuando se consideran individualmente. Estos grados están detallados en la Tabla 2 mostrando ejemplos según el tipo de acción y el punto de inicio. En contraposición, el pulgar posee una variedad más amplia de movimientos comparado con los demás dedos (Fuentes, C. 2020).

**Figura 4**

*Movimientos de los dedos de la mano humana: Abducción, Aducción, Extensión y Flexión*



*Nota:* La figura ilustra los movimientos fundamentales de los dedos de la mano humana, proporcionando una representación visual de la abducción, aducción, extensión y flexión. Estos movimientos son esenciales para la función manual y la destreza, y son importantes en el estudio de la anatomía, la rehabilitación y el diseño robótico. Las flechas indican la dirección del movimiento de cada dedo, facilitando la comprensión de los diferentes rangos de movimiento posibles en las articulaciones de la mano. *Adaptado de Gray, H. (2020).*

**2.2.5. Tipos de agarre**

Los objetos que tomamos tienen diferentes formas y tamaños, pero los estudios demuestran las tareas diarias de recoger y colocar objetos se resumen en 6 tipos de agarre.

**Figura 5**

*Tipo de agarre*



*Nota:* Esta figura muestra los diferentes tipos de agarre que puede realizar la mano humana, los cuales son fundamentales para el diseño de prótesis y herramientas ergonómicas. *Adaptado de Kapandji, A. I. (2012). The Physiology of the Joints*

Como se puede apreciar en la Figura 5 existen 6 tipos de agarre y cada uno aporta en diferentes puntos la fuerza necesaria para sostener un objeto en específico con la geometría diferente para cada tipo. Estos movimientos, en la tarea rutinaria normal de recoger y colocar objetos, toman aproximadamente de 172 a 200 grados por segundo y la velocidad de flexión de los dedos es aproximadamente 20 a 775 grados por segundo (Fuentes, C. 2020). A continuación, se detallará los tipos de agarre mostrados en la figura 8.

**Agarre palmar:**

El dedo índice, medio y pulgar se retraen juntos hacia la palma envolviendo al objeto, se desarrolla una fuerza en toda el área de contacto, no únicamente en la punta de los dedos.

**Agarre cilíndrico:**

Similar al agarre palmar solo que en este caso se desarrolla para objetos de mayor tamaño donde se requiere mantener la palma más abierta, mayormente los dedos son los que aplican fuerza.

**Agarre lateral:**

El pulgar se encuentra abducido haciendo presión sobre la falange media del dedo índice. Para realizar esta maniobra se requiere un punto estable, es por ello que en la mayoría de los casos se realiza contra el dedo índice.

**Agarre esférico:**

Cuando las articulaciones interfalángicas distal y proximal se flexionan aproximadamente  $30^\circ$  y el pulgar se encuentra en posición de abducción palmar de tal manera que las fuerzas son producidas entre los dedos y el pulgar.

**Agarre de punta:**

Este agarre depende de la flexión de la articulación interfalángica proximal del pulgar y de la articulación distal del dedo índice. La punta de los dedos son las encargadas de aplicar la fuerza en el objeto a agarrar.

**Agarre de gancho:**

Depende de la flexión de la articulación interfalángica proximal y la extensión de la articulación metacarpofalángica, no requiere del uso del pulgar, dado que para este tipo de agarre la fuerza es producida por la presión de los dedos contra la palma.

**2.2.6. Presión de la Mano****Fuerza de la Mano para una Población Adulta**

La medida de la fuerza de mano con un dinamómetro de mano evalúa la fuerza isométrica de los dedos de la mano y del antebrazo. Además, guarda una relación con el estado de nutrición de la persona.

### **Efecto de la tarea en el agarre en la manipulación de botellas**

El problema de la redundancia motriz en la producción de fuerza por la combinación de varios dedos ha sido remarcado por estudios previos, de manera que la misma fuerza de agarre total se puede producir con diferentes contribuciones de los dedos individuales, lo que se conoce como la contribución de fuerzas.

La manipulación de botellas u objetos similares es una actividad de la vida cotidiana típica que es incluida en los protocolos de evaluación de la capacidad de agarre (Cepria, J. 2016). La fuerza de agarre o de presión de la mano puede medirse con un dinamómetro de Jamar (Figura 7), que es el método por el cual se mide la función y fuerza muscular de las extremidades superiores, así como dispositivos que cumplen la misma función como Martin Vigorimeter y Jamar Hidraulic Pinc Gauge.

**Figura 6**  
*Dinamómetro T.K.K. 540 Grip-D*



*Nota:* Esta figura muestra un dinamómetro T.K.K. 540 Grip-D, un instrumento de medición de la fuerza de agarre que es esencial en la evaluación de la aptitud física y la recuperación de la función de la mano en terapias de rehabilitación. Este modelo particular es conocido por su precisión y facilidad de uso, con una pantalla digital que muestra la fuerza de agarre en kilogramos o libras. La herramienta es ampliamente utilizada en pruebas deportivas, evaluaciones clínicas y estudios de ergonomía. *Adaptado de J.A. Oteo, P. Benavente, M. Garzón (2015).*

## **Caracterización del agarre en la realización de tareas cotidianas**

Existen trabajos previos que, en lugar de utilizar los objetos cotidianos reales para estudiar el agarre, han utilizado objetos de laboratorio o instrumentados con el fin de extrapolar los resultados obtenidos a los objetos reales (Cepria, J. 2016).

El interés del estudio de las fuerzas de agarre sobre los objetos reales consiste en que permite obtener resultados que estén influenciados en la menor medida posible por el parecido al objeto instrumentado o del equipo de laboratorio con el objeto real. Además, la variedad de tareas realizadas con la mano humana viene condicionada sobre todo por el objeto que se manipula y la tarea que se realiza con el mismo, durante las actividades de la vida diaria.

### **2.3. Mano Robótica**

Es un dispositivo mecánico diseñado para imitar la funcionalidad de una mano humana. Está diseñada para agarrar, sostener, manipular objetos y/o realizar diversas tareas que normalmente serían realizadas por una mano humana. Estas manos robóticas son utilizadas en una variedad de aplicaciones en campos como la industria, la medicina, la investigación y la robótica.

Las manos robóticas suelen estar compuestas por dedos articulados y una palma que imita la anatomía de una mano humana. Los dedos pueden tener articulaciones similares a las articulaciones de los dedos humanos y están conectados a actuadores, ya sean motores eléctricos o neumáticos, que les permiten moverse y agarrar objetos. Algunas manos robóticas están equipadas con sensores táctiles en las puntas de los dedos para detectar la presión, la fuerza y la textura de los objetos que manipulan. Esto les permite realizar tareas más precisas y seguras.

Las manos robóticas tienen diversas aplicaciones médicas, de asistencia a personas con discapacidades e investigación y desarrollo de tecnologías avanzadas como la inteligencia artificial y la visión por computadora. Estas manos pueden ser utilizadas para entrenar algoritmos de aprendizaje automático en tareas de percepción y manipulación.

#### **2.3.1. Prototipos de manos robóticas**

Existen diversos tipos de prototipos diseñados para diferentes aplicaciones y con distintas características, cada uno de ellos satisface una necesidad específica en una amplia variedad de campos. Una de las más comunes son las manos robóticas de rehabilitación, estas manos se utilizan en personas con discapacidades o lesiones en las manos, ya sea congénito o accidentes

comúnmente laborales. Ayudan en la recuperación de la función de agarre y movimiento de la mano, requiriendo además de un equipo multidisciplinario para lograr la incorporación y adaptación de la mano robótica de rehabilitación a la persona con discapacidad. Existen además las manos robóticas humanoides, que imitan la apariencia y los movimientos de una mano humana.

Los materiales de construcción de una determinada mano robótica dependen mucho del uso que se le vaya a dar como aleaciones de aluminio, acero inoxidable, titanio que es biocompatible, plásticos de ingeniería y materiales compuestos como la fibra de carbono y fibra de vidrio en resina epoxi. Existen múltiples prototipos de manos robóticas construidas con diferentes materiales, como:

I Limb Ultra, una mano desarrollada para su uso en prótesis para personas con amputaciones, desarrollada por Touch Bionics.

Dexterous Hand, una mano robótica humanoide para uso de investigación y desarrollo en universidades, desarrollada por Shadow Robot Company.

Bionic Soft Hand, mano robótica neumática con inteligencia artificial para robots en espacios colaborativos con humanos, desarrollada por Festo.

Inmoov es un robot humanoide de código abierto impreso en 3D para uso de investigación y desarrollo tecnológico, desarrollado por el francés Gael Langevin.

Una alternativa de bajo costo para construir estas piezas robóticas, son las impresiones en 3D, que además permite una fabricación rápida y personalizada de piezas complejas.

### **2.3.2. Robot InMoov**

InMoov es el proyecto personal del escultor y diseñador francés Gael Langevin, que inició en enero del 2012 como la primera prótesis de mano de código abierto y ha dado lugar a proyectos como biónicos, robóticos, E-Nable y muchos otros.

InMoov es el primer robot de tamaño natural impreso en 3D de código abierto. Replicable en cualquier impresora 3D domestica con un área de 12x12x12 cm, está concebida como una plataforma de desarrollo para universidades, laboratorios y centros de investigación.

**Figura 7**

*Robot InMoov impreso en 3D y ensamblado*



*Nota:* La imagen captura el robot InMoov, un diseño de código abierto que ha sido impreso en 3D y ensamblado. Este modelo es notable por ser el primer robot de tamaño humano impreso en 3D que puede ser replicado y personalizado por entusiastas de la robótica en todo el mundo. Sus aplicaciones varían desde la educación y la investigación hasta la robótica asistencial. La figura muestra la estructura compleja del robot, que incluye articulaciones motorizadas y componentes electrónicos que permiten la simulación de movimientos humanos. *Adaptado de InMoov.*

El concepto de InMoov se basa en compartir y en la comunidad, lo que ha permitido ser reproducido para innumerables proyectos en todo el mundo con diferentes propósitos.

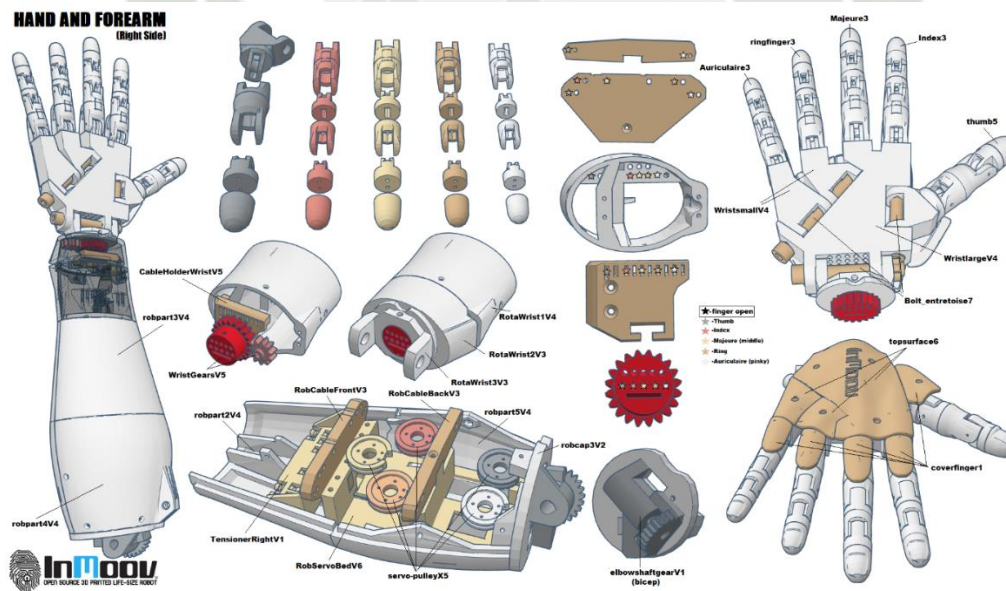
El robot InMoov (Figura 7) se compone de varias partes que se pueden imprimir en 3D y ensamblar para construir el robot completo. A continuación, se presentan algunas de las partes del robot InMoov.

### 2.3.2.1. Partes de la mano y antebrazo

Las partes de la mano y el antebrazo son parte esencial del robot, pues le permiten agarrar objetos y realizar tareas de manipulación, imitando la estructura y funcionalidad de una mano y brazo humanos. Las manos tienen dedos móviles independientes y articulaciones. La muñeca del robot permite el giro de la mano, permitiéndole realizar variedad de movimientos. Todos los dedos son articulados, los dedos: índice, medio, anular y meñique tienen 3 articulaciones o falanges, lo que permite a la mano del robot los movimientos de flexión y extensión. El pulgar en cambio tiene 2 articulaciones, imitando el pulgar humano, lo que le permite oponerse a los otros dedos y agarrar objetos de manera similar a la mano humana.

Cada dedo es independiente de los demás, se usa un servomotor para cada dedo, además se necesitan cables y piezas pequeñas impresas en 3D para conectar los servomotores, y otros componentes al hardware de control.

**Figura 8**  
Vista 3D del conjunto mano y antebrazo



*Nota:* Esta figura presenta una vista desglosada en 3D del ensamblaje de la mano y el antebrazo del robot InMoov, ilustrando la complejidad del diseño y la precisión requerida para su montaje. Cada componente está meticulosamente diseñado para encajar dentro del sistema robótico, ofreciendo una simulación realista de la movilidad humana. Esta imagen es especialmente útil para educadores, aficionados a la robótica y fabricantes que buscan entender o participar en el montaje de sistemas robóticos. *Adaptado de InMoov.*

## 2.4. Seguridad y Manipulación de Sustancias Tóxicas

La manipulación de sustancias tóxicas en laboratorios farmacéuticos o de investigación química o biológica requiere medidas de seguridad adicionales para proteger a los trabajadores y operarios. El riesgo a la salud es la probabilidad de que una sustancia química peligrosa pueda causar directa o indirectamente lesión temporal, permanente o la muerte del trabajador por ingestión, inhalación o contacto (Gavilán, I. 2014). Las operaciones más habituales llevadas a cabo con los productos químicos son: el trasvase de líquidos y la mezcla o adición de un producto.

### 2.4.1. Traspase de líquidos

La manipulación y el trasvase de agentes químicos son procesos en los que es necesario prestar la mayor atención posible. El trasvase suele estar relacionado con un elevado número de accidentes como: proyecciones, salpicaduras, intoxicaciones, incendios entre otros (Dirección General de Función Pública, 2008).

### 2.4.2. Mezcla o adición de un producto

Durante este tipo de operaciones puede darse una reacción química lo que lleva asociado una serie de peligros como explosión, proyección, incendios y desprendimiento de vapores o gases a una temperatura, presión y velocidad tales que causen daños en los alrededores.

Según el anuario estadístico sectorial 2022, publicado por el Ministerio de Trabajo y Promoción del Empleo, las notificaciones de accidentes de trabajo según el agente causante dieron lugar a 1598 casos reportados por Materiales, sustancias y radiaciones. y 96 casos por Sustancias químicas y plaguicidas; lo que conjuntamente representa el 5.26 % del total de accidentes laborales reportados durante el 2022.

## 2.5. Impresión 3D

La impresión 3D es un proceso de fabricación que produce objetos tridimensionales mediante la adición de material capa por capa. Este proceso también se conoce como manufactura

por adición. La impresión 3D es una tecnología de manufactura digital que permite la creación de geometrías complejas que antes no eran posibles mediante otros procesos de fabricación.

La impresión 3D se utiliza en una variedad de industrias, incluyendo la aeroespacial, la médica y la automotriz para la creación de prototipos y la producción personalizada en masa. Una de las ventajas significativas para la impresión 3D es la velocidad de fabricación, ya que la impresión 3D de un objeto puede llevar horas o incluso días enteros, sigue siendo mucho más rápida que los métodos de producción habituales como tornos y fresadoras CNC.

Antes de realizar cualquier impresión, los fabricantes deben preparar modelos 3D. Estos modelos pueden ser diseñados por el usuario o descargados de una biblioteca en línea. La impresión 3D ha recorrido un largo camino desde sus inicios con la manufactura de figuras decorativas, y su futuro es más brillante que nunca a medida que la tecnología va evolucionando.

## **2.6. Herramientas Software Para Utilizar**

### **2.6.1. Python**

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado en 1991 por Guido van Rossum y se ha convertido en uno de los lenguajes más populares en la actualidad debido a su facilidad de uso, legibilidad y versatilidad.

Python es utilizado en una amplia variedad de aplicaciones, desde el desarrollo web y la ciencia de datos hasta la inteligencia artificial y el aprendizaje automático. Una de las ventajas de Python es su sintaxis clara y concisa, lo que lo hace fácil de aprender para los principiantes de programación.

Python es un lenguaje de programación de código abierto, lo que significa que su código fuente está disponible para que cualquiera lo use y lo modifique. Además, cuenta con una gran comunidad de desarrolladores que contribuyen a su desarrollo y mejora continua.

### **2.6.2. OpenCV**

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto. Fue creada para proporcionar una infraestructura común para aplicaciones de visión por computadora y acelerar el uso de la

percepción de maquina en productos comerciales. OpenCV cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de visión por computadora y aprendizaje automático clásicos y de última generación.

OpenCV tiene interfaces en varios lenguajes de programación, incluyendo C++, Python, Java y Matlab, y es compatible con Windows, Linux, Android y Mac OS. Se enfoca principalmente en aplicaciones de visión en tiempo real.

### 2.6.3. MediaPipe

MediaPipe es una biblioteca de software de aprendizaje automático perteneciente a Google Open Source, quienes han desarrollado más de 2 mil proyectos que podemos explorar desde su página oficial. Esta biblioteca es un framework multimodal y multiplataforma que aplica machine learning, que permite a los desarrolladores crear soluciones de aprendizaje automático en dispositivos móviles, de escritorio o en la web. MediaPipe está optimizado para el desarrollo de extremo a extremo de soluciones de aprendizaje automático y es compatible con una amplia variedad de dispositivos y sistemas operativos como Android, iOS, C++, Python, JS.

MediaPipe contiene todo lo que se necesita para personalizar y desplegar soluciones de aprendizaje automático. La biblioteca también contiene modelos para la detección de rostros, tracking de manos, segmentación de cabello, detección y tracking de objetos, y otra gama amplia de modelos de uso y desarrollo libre.

## CAPITULO III: DISEÑO E IMPLEMENTACION

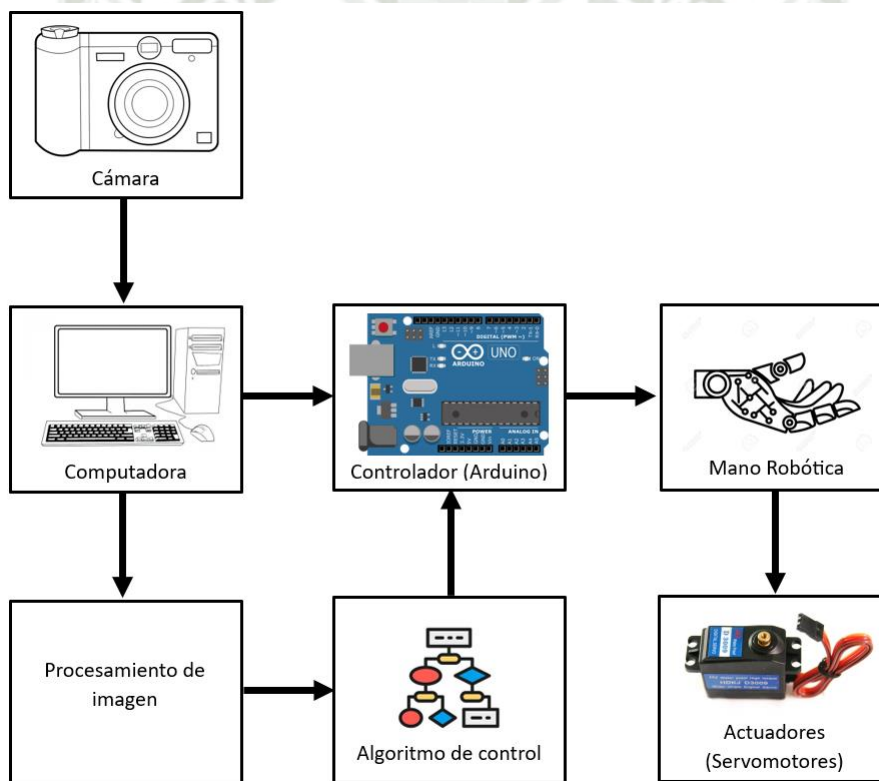
### 3. DISEÑO DEL SISTEMA DE VISIÓN ARTIFICIAL

#### 3.1. Diagrama de bloques del sistema de visión artificial

En la Figura 9 se muestra el esquema de bloques del sistema de conjunto de visión artificial para el control de la mano robótica, la cámara se conectará directamente con la computadora para realizar el procesamiento de imagen, realizar el algoritmo de control y mandar las señales al controlador, para hacer efecto en los actuadores de la mano robótica en tiempo real a como la pacta la cámara.

**Figura 9**

*Diagrama de bloques del sistema de visión artificial para la mano robótica*



*Nota:* La figura ilustra el flujo de procesamiento de información en un sistema de visión artificial diseñado para el control de una mano robótica. El proceso comienza con la captura de imagen por la cámara, seguido por el procesamiento de dicha imagen en una computadora. La información procesada es entonces enviada al controlador Arduino, que ejecuta un algoritmo de control. Este algoritmo ajusta los actuadores, que son servomotores, para manipular la mano robótica de acuerdo con los comandos derivados del análisis visual. Este diagrama es esencial para entender la interacción entre los componentes del sistema y el flujo de datos necesario para el funcionamiento autónomo de la mano robótica.

### 3.2. Selección del Prototipo

Existen ya diversos prototipos de manos robóticas diseñados para diferentes proyectos de investigación en robótica. Muchos de estos tienen mejoras continuas y funcionalidades más completas que crear un diseño nuevo, además, el uso de estas herramientas permite dar lugar a investigaciones nuevas en desarrollo e innovación. Sin embargo, son pocos los prototipos que son de libre uso para investigaciones, desarrollo de prótesis, innovación, y otros campos en los que se puedan aplicar.

El robot InMoov viene desarrollándose desde el año 2012, y se pueden obtener sus archivos stl en su página web, al igual que los archivos del modelo Hackberry (Figura 10) se encuentran a disponibilidad en la web, el primer lanzamiento de este prototipo fue en 2014. El modelo Hero Arm (Figura 11) de la empresa británica Open Bionics es un modelo más sofisticado, sin embargo, con una solicitud pueden obtenerse algunas piezas para su impresión en 3D. Esta empresa se fundó en el año 2014 con el fin y especialidad de diseño y desarrollo en prótesis biónicas de extremidades para mejorar la calidad de vida de las personas amputadas.

**Figura 10**  
*Mano Robótica Hackberry*



*Nota:* Esta figura muestra la mano robótica Hackberry, un ejemplo destacado de innovación en la comunidad de código abierto. Hackberry es diseñada para ser una prótesis de mano de bajo costo y de fácil acceso, utilizando impresión en 3D y componentes electrónicos simples para su construcción. Su diseño modular y adaptable permite a los usuarios personalizar la prótesis según sus necesidades específicas, fomentando una atmósfera de colaboración y mejora continua dentro de la comunidad de código abierto. *Adaptado de Open Source Community.*

Se tienen en cuenta otros prototipos de manos robóticas, a continuación se mostrará un cuadro comparativo (Tabla 3) de los modelos del Robot InMoov, Hackberry y Hero Arm,

describiendo algunas características principales como tamaño, grados de libertad, disponibilidad, número de piezas y otros para su comparación, herramientas que pueden obtenerse sus archivos STL a través de internet.

### Figura 11

*Mano robótica Hero Arm*



*Nota:* Esta figura presenta la Hero Arm, una prótesis de mano robótica avanzada creada por Open Bionics. Diseñada para ser ligera y versátil, la Hero Arm ofrece nuevas posibilidades para la mejora de la calidad de vida de las personas con amputaciones de miembro superior. La prótesis es notable por su diseño modular que puede ser adaptado a las necesidades individuales del usuario y por el uso de la impresión en 3D para su fabricación, lo que la hace accesible y personalizable. *Adaptado de Open Bionics.*

**Tabla 1**  
*Prototipos de manos robóticas para impresión en 3D*

<b>Mano Robótica</b>			
	<b>InMoov</b>	<b>Hackberry</b>	<b>Hero Arm</b>
Origen	Proyecto de código abierto desarrollado por Gael Langevin	Producto desarrollado por la empresa japonesa Exii	Producto destacado de la empresa británica Open Bionics
Impresión 3D	Gran parte de su estructura se puede imprimir en 3D, lo que permite su personalización	Algunas versiones tienen componentes fabricados con otros métodos	Se emplea tecnología de impresión 3D para una producción eficiente y capacidad de personalizar la prótesis.
Dimensiones	200 x 120 x 80 mm	225 x 150 x 60 mm	178 x 80 x 55
Número de Piezas	54	64	No se tienen datos exactos
Cantidad de actuadores	5 servomotores, uno por cada dedo	3 servomotores, un pulgar, un índice y uno para el medio, anular y el meñique	3 servomotores, uno para el pulgar, otro para el índice y medio, y otro para el anular y meñique
Grados de Libertad	16	No se tienen datos exactos	5
Control	Software de código abierto	Utiliza su propio software de control desarrollado por Exii	Al ser desarrollada como prótesis, se controla con EMG
Aplicaciones Típicas	Proyectos de robótica y experimentación	Principalmente desarrollada para prótesis	Prótesis biónicas de mano
Disponibilidad	Se puede construir a partir de planos y componentes disponibles públicamente	Se puede comprar como producto comercial	Solicitud a través de la página web de la empresa
Facilidad de integración	Debido a que todo el proyecto es un robot humanoide, la mano fácilmente se puede integrar al brazo y torso	Su diseño solo se basa en prótesis de miembro superior	Su diseño solo se basa en prótesis de miembro superior

*Nota:* Esta tabla compara diferentes prototipos de manos robóticas diseñadas para la impresión en 3D. Incluye información sobre el origen, las dimensiones, el número de piezas, los actuadores, los grados de libertad, el control, las aplicaciones típicas y la disponibilidad de cada modelo. Se destacan las particularidades de los modelos InMoov, Hackberry y Hero Arm, proporcionando una visión general de sus capacidades y aplicaciones.

Se realizó un cuadro comparativo entre diferentes tipos de prototipos de manos robóticas que se pueden mostrar en la Tabla 3 con sus diferentes características desde el rigén hasta la facilidad de integración de estos prototipos. La mano del robot InMoov tiene independencia en cada dedo por tener un servomotor en cada uno, en comparación a la mano Hackberry que solo tiene 3. El modelo de Hackberry está diseñado principalmente para implementar prótesis de miembro superior al igual que el modelo Hero Arm, y el modelo de InMoov puede integrarse al robot humanoide completo.

La implementación del sistema de visión artificial se realiza en el prototipo de la mano del robot InMoov por su facilidad de integración, la cantidad de actuadores y la disponibilidad.

### 3.3. Etapas del Sistema

El control mediante visión por computadora esta ordenado en etapas, que contiene 3 principales y 1 de monitoreo:

- Etapa 1: Identificación de la mano Humana
- Etapa 2: Programación de los Hand LandMarks
- Etapa 3: Movimiento de los dedos
- Etapa 4: Interfaz de Usuario

Estas etapas van desde la identificación de la mano humana a través de una cámara web, la programación de los puntos de interés, la construcción la mano y movimiento de los dedos, y la interacción con el usuario a través de una interfaz visual.

Donde el usuario en la etapa uno mostrará su mano a través de la cámara web para ser identificada y a los puntos de referencia de la mano (Hand LandMarks) de MediaPipe. En la etapa dos se programará los Hand LandMarks para arrojar un vector de 5 elementos el cual será el valor detectado de cada dedo si está abierto o cerrado. Luego, en la etapa 3, la implementación del software al hardware construido. Finalmente, la visualización de lectura de la entrada y salidas del sistema e interacción con el usuario.

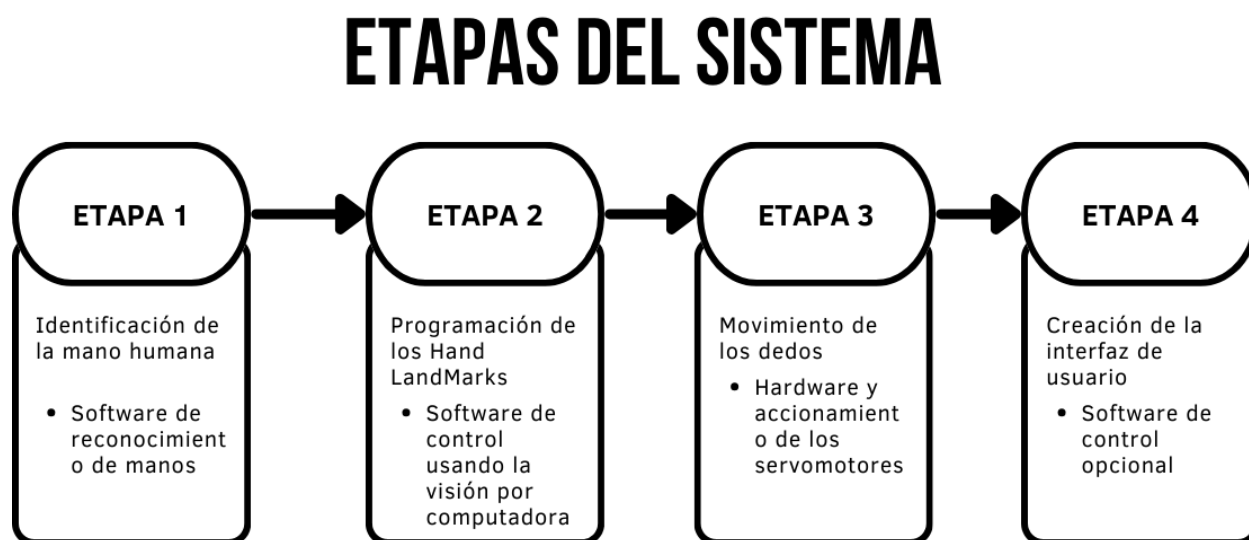
#### 3.3.1. Diagrama de Bloques

Este sistema se basa en el uso de una computadora para el procesamiento de las imágenes adquiridas por la cámara web, aplicando la visión artificial para la identificación de una mano

humana y reconocimiento de los puntos de referencia o Hand LandMarks de MediaPipe, estos puntos nos servirán para crear el código de control y saber en qué momento se tendrán que mover los dedos en la mano robótica. El movimiento de los dedos tendrá un código de accionamiento para los motores a través de la plataforma virtual del microcontrolador integrado a una tarjeta de desarrollo. Finalmente se desarrollará una interfaz capaz de interactuar con el usuario de manera intuitiva y flexible. Gráficamente podemos observar las etapas del sistema en la figura 12.

**Figura 12**

*Diagrama de bloques de las etapas del sistema de visión artificial*



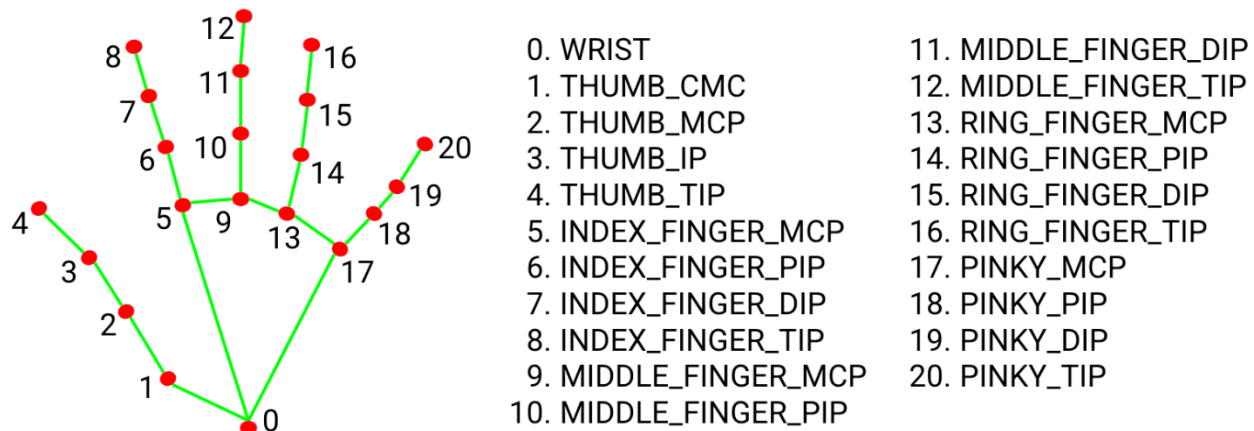
*Nota:* Este diagrama de bloques detalla las fases clave en el desarrollo del sistema de visión artificial para la identificación y control de movimientos de la mano. Desde el reconocimiento inicial hasta la interacción del usuario final, cada etapa es crítica para el funcionamiento integral del sistema. El diagrama proporciona una representación esquemática que puede servir como una guía para los desarrolladores y técnicos en la implementación y la mejora de sistemas de control basados en visión por computadora.

### 3.3.2. Etapa 1 – Identificación de la mano humana

En el trabajo de esta implementación no se hará uso de sensores para imitar los movimientos de la mano en tiempo real, lo cual hace del proyecto presente una reducción de costos, pudiendo hacer del sistema incluso más rentable que la instalación y el mantenimiento de sensores físicos para su uso, además de la facilidad de adaptación en diversos entornos variables con diferentes condiciones de iluminación. El uso de las bibliotecas OpenCV y MediaPipe serán los puntos de partida para crear el código que permita reconocer la mano humana a través de una cámara. Con la combinación de estas 2 bibliotecas podremos obtener los puntos clave o

ubicaciones específicas en la mano que representan características anatómicas de esta misma (Hand LandMarks).

**Figura 13**  
*Hand LandMarks*



*Nota:* Esta figura presenta un esquema de los puntos de referencia de la mano, comúnmente usados en la visión por computadora para detectar y seguir la posición y el movimiento de la mano en tiempo real. La imagen ilustra 21 puntos distintos que corresponden a la muñeca, la base de cada dedo, las articulaciones de los dedos y las puntas de los dedos. Este esquema es crucial para desarrollar interfaces de usuario que funcionen con gestos de la mano y para aplicaciones de realidad aumentada que requieren interacciones precisas. *Adaptado de MediaPipe Hand Landmarks, Google Developers.*

Estos puntos de la Figura 13 son el objetivo final de la primera etapa, los cuales tienen que detectarse en cualquier mano mostrada por el usuario para la correcta identificación.

La visión artificial en esta aplicación presenta una alternativa no invasiva al uso común de electrodos EMG de agujas, del tipo no invasivo usando bandas en los brazos para leer las señales eléctricas del cuerpo humano que mandan un movimiento o sensores de aceleración y fuerza que miden la actividad muscular.

### 3.3.3. Etapa 2 – Programación de los Hand LandMarks

Obtenidos los Hand LandMarks tienen que sujetarse en un código en el cual se trabajen como puntos principales a [4, 8, 12, 16, 20] que son las puntas de los dedos para que trabajen según sea el caso:

En el dedo índice: si el punto [8] (que es la punta del dedo índice) está por debajo del punto [5] (que se encuentra al borde de la palma), este deberá estar cerrado en el prototipo de mano robótica.

En el dedo medio: si el punto [12] (que es la punta del dedo medio) está por debajo del punto [9] (que se encuentra al borde de la palma), este deberá estar cerrado en el prototipo.

En el dedo anular: si el punto [16] (la punta del dedo anular) se encuentra por debajo del punto [13] (en el borde de la palma), este deberá estar cerrado en el prototipo.

En el dedo meñique: si el punto [20] (la punta del dedo meñique) se encuentra por debajo del punto [17] (en el borde de la palma), este deberá estar cerrado en el prototipo.

Finalmente, en el dedo pulgar será visiblemente de diferente manera, puesto que, al cerrar este dedo no se encuentra por debajo de los puntos [3], [2] o [1], sino más bien se acerca más a los dedos índice y medio, entonces: si el punto [4] (punta del dedo pulgar) se encuentra más a la derecha del punto [5], este se encontrará cerrado en el prototipo.

El objetivo final de esta etapa es crear un código que arroje un vector de 5 elementos (un elemento por dedo) los cuales tendrán dos posibles valores, 0 y 1, lo definirá si los dedos estarán cerrados o abiertos.

#### **3.3.4. Etapa 3 – Movimiento de los dedos**

En esta etapa se seguirá una secuencia de pasos para aplicar una lógica de control para enviar señales de salida a los servomotores para ejecutar las acciones e imitar los movimientos de los dedos en la cámara web que detecta la mano humana. Para concretar esta etapa se necesita identificar cuantos actuadores se van a utilizar en el sistema.

En la mano robótica InMoov, el sistema mecánico requiere 5 actuadores, uno por cada dedo. El microcontrolador por el método de control de PWM, enviara las señales de pulsos a los servomotores según los valores del vector obtenido de la programación en Python.

#### **3.3.5. Etapa 4 – Creación de la Interfaz de Usuario**

En esta etapa se creará una interfaz con el objetivo de tener la visibilidad de las cámaras que actúen en el sistema completo. El protocolo de comunicación es serial, donde se envía y reciben los datos entre el software de Arduino y el de Python. En el funcionamiento de la interfaz se verán en vivo 2 videos, uno correspondiente a la cámara web de identificación y proceso de la visión artificial en la mano del usuario, y la otra cámara (auxiliar) de visualización de la mano robótica en tiempo real imitando los movimientos de la mano del usuario. Además de la

implementación de botones de encendido de la cámara auxiliar y una serie de pruebas antes de iniciar el proceso de control por visión artificial para verificar el correcto funcionamiento de los servomotores en cada dedo.

### **3.4. Implementación del Sistema**

Para la implementación del sistema es necesario el uso de software y hardware

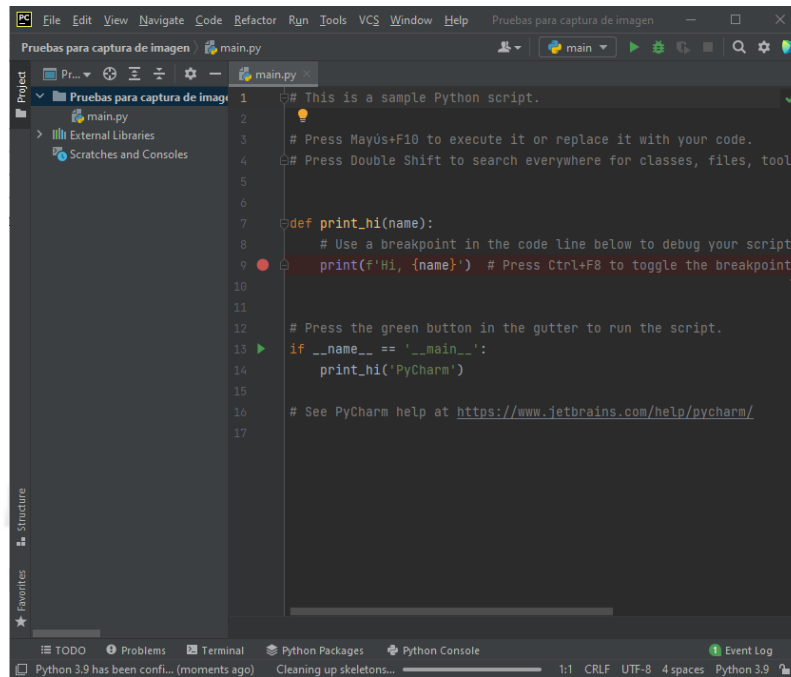
#### **3.4.1. Software**

##### **3.4.1.1. Bases de programación**

Programación en Pycharm

El editor de Python, Pycharm, o entorno de desarrollo integrado, nos permitirá integrar las librerías de OpenCV y MediaPipe con el fin de la detección de la mano a través de la cámara web. Además de programar los puntos de detección o Hand LandMarks a través de la cámara web. Este entorno de desarrollo también nos permite comunicarnos con Arduino IDE y con su dispositivo o placa de desarrollo a través del puerto serial. Finalmente se usará Pycharm para crear y diseñar la interfaz de usuario.

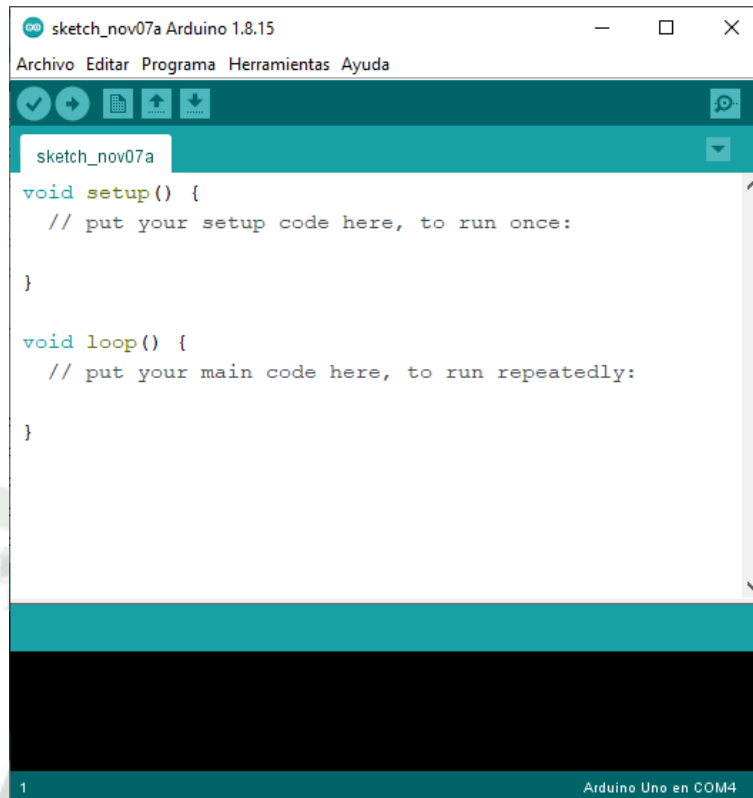
**Figura 14**  
*Ventana principal para programación en Pycharm*



## Programación en Arduino IDE

También es un entorno de desarrollo integrado de Arduino, se utilizará para escribir y compilar los códigos desarrollados para el movimiento de los servomotores de cada dedo, se usarán bibliotecas propias de Arduino para el control preciso del ángulo de rotación del actuador, y la entrada serial a través del puerto USB.

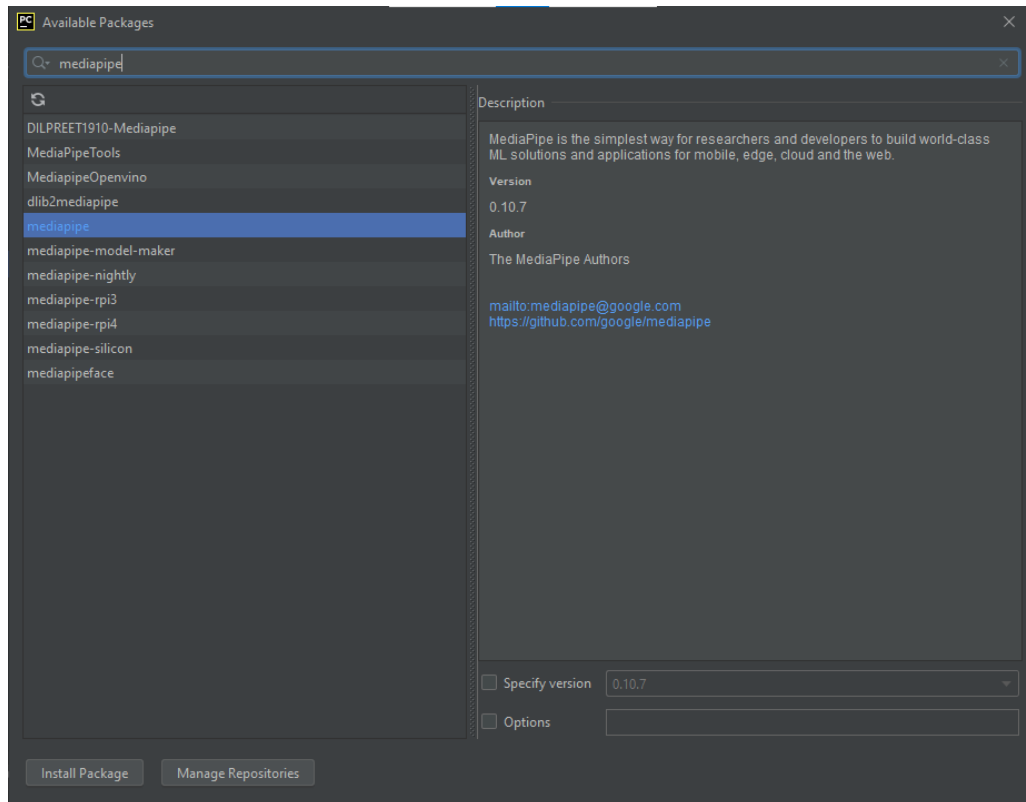
**Figura 15**  
*Ventana principal para programación en Arduino*



#### 3.4.1.2. Etapa 1 - Identificación de la mano humana

Se instalaron las bibliotecas necesarias de OpenCV (cv2) y MediaPipe (mediapipe) a través del editor de códigos Pycharm en la ventana de “Available Packages” como se muestra en la Figura 16, con la finalidad de poder capturar y mostrar el video en tiempo real y la funcionalidad de detectar y seguir las manos. En las primeras líneas de código se importarán las bibliotecas “cv2” y “mediapipe as mp”.

**Figura 16**  
*Instalación de Bibliotecas nuevas en Pycharm*

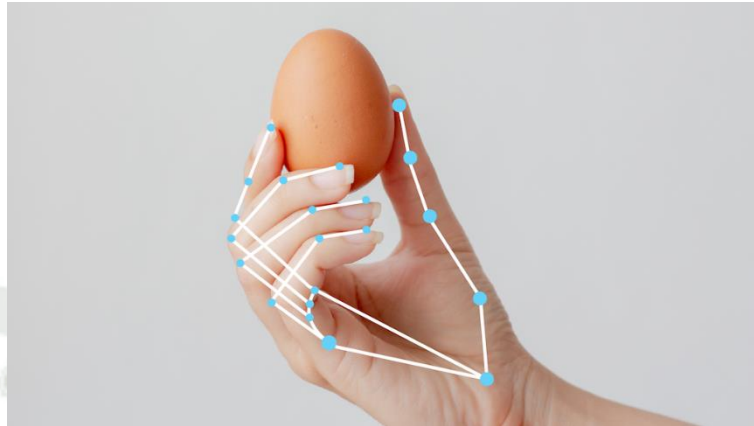


A continuación, se configuró MediaPipe para detectar las manos importando submódulos específicos de MediaPipe: “mp\_drawing” y “mp\_hands”, utilizados para detectar y seguir las manos, además de dibujar los Hand LandMarks y las conexiones entre estos puntos en el video.

Ahora se iniciará la cámara web para proceder con el bucle principal, este procedimiento captura el video en tiempo real y procesa cada fotograma de la cámara, se definen el número máximo de manos a detectar y se establece el umbral de confianza mínimo para la detección de manos. El procesamiento de cada fotograma se encuentra dentro del bucle, y se realizan operaciones en el fotograma para prepararlo para la detección de manos.

Se utiliza el objeto “hands” para procesar el fotograma en formato RGB y detectar las manos, los resultados de la detección se almacenan en la variable “results”, y si se detectan manos, se recorre los Hand LandMarks y se dibujan en el fotograma. El fotograma con los 21 puntos de referencia dibujados se muestra en una ventana de video en vivo mediante “cv2.imshow”.

**Figura 17**  
*Detección de Hand LandMarks*



*Nota:* La figura ilustra la detección de puntos de referencia en la mano humana utilizando tecnología de visión por computadora. Cada punto azul representa un Hand LandMark específico en la mano, y las líneas blancas indican la estructura esquelética inferida entre ellos. Esta tecnología es fundamental para el desarrollo de interfaces de usuario basadas en gestos, juegos interactivos, y aplicaciones de realidad aumentada, proporcionando una forma intuitiva de interacción entre el usuario y la máquina. *Adaptado de MediaPipe Hand Landmarks, Google Developers.*

El bucle terminará cuando se presione la tecla de escape o al detener directamente el código en Pycharm, una vez se termine el bucle, se liberará la cámara web y se cerrarán todas las ventanas abiertas.

### **Descripción general del sistema de visión artificial basado en MediaPipe.**

En el desarrollo de esta tesis, se ha implementado un sistema de visión artificial para controlar los movimientos de una mano robótica, con el objetivo de permitir la manipulación remota de sustancias tóxicas, minimizando así los riesgos para los operarios. El enfoque utilizado se basa en la biblioteca de código abierto MediaPipe, desarrollada por Google.

MediaPipe es un marco de trabajo de visión artificial multiplataforma que ofrece soluciones listas para usar en tareas de detección y seguimiento de objetos, reconocimiento de gestos, estimación de poses y más. En particular, esta investigación se ha centrado en el uso del módulo de detección y seguimiento de manos (Hand Tracking) de MediaPipe, el cual aprovecha modelos de aprendizaje profundo para identificar y localizar manos en imágenes o secuencias de video en tiempo real.

Una vez detectada la mano, este módulo proporciona las coordenadas de puntos de referencia (Hand LandMarks) que representan la ubicación de características clave de la mano,

como las articulaciones de los dedos, la muñeca y la palma. Estos puntos de referencia se actualizan en cada fotograma, permitiendo realizar un seguimiento preciso de los movimientos y gestos de la mano.

### **Explicación del concepto de detección de manos y seguimiento de puntos de referencia (Hand LandMarks).**

La detección de manos y el seguimiento de puntos de referencia (Hand LandMarks) son componentes clave del sistema de visión artificial implementado en esta tesis. A continuación, se explican brevemente estos conceptos:

- **Detección de manos:**

El proceso de detección de manos implica identificar y localizar las manos presentes en una imagen o secuencia de video. Esto se logra mediante el uso de modelos de aprendizaje profundo entrenados con grandes conjuntos de datos de imágenes de manos. El modelo de detección analiza cada fotograma y produce una predicción sobre la presencia y ubicación de las manos dentro de la imagen.

- **Seguimiento de puntos de referencia Hand LandMarks:**

Una vez que se ha detectado una mano, el siguiente paso es estimar la ubicación precisa de puntos de referencia clave en la mano. Estos puntos de referencia, también conocidos como Hand LandMarks, representan características anatómicas específicas, como las articulaciones de los dedos, la muñeca, la palma y los nudillos. El módulo de MediaPipe utilizado en esta tesis es capaz de detectar y rastrear hasta 21 puntos de referencia en cada mano. Estos puntos se representan como coordenadas 2D o 3D en el espacio de la imagen o el espacio real, dependiendo de la configuración utilizada. El seguimiento preciso de estos puntos de referencia es fundamental para interpretar los movimientos y gestos de la mano. A medida que la mano se mueve o cambia de posición, las coordenadas de los puntos de referencia se actualizan en consecuencia, permitiendo capturar los cambios en la postura y la forma de la mano. Esta información detallada sobre la ubicación de los puntos de referencia se utiliza posteriormente para controlar los movimientos de la mano robótica, mapeando las posiciones y ángulos de los dedos de la mano real a los actuadores correspondientes en la mano robótica.

### **Detalle del flujo de procesamiento de fotogramas y detección de manos.**

El proceso de detección de manos en el sistema de visión artificial implementado en esta tesis sigue un flujo específico que involucra varias etapas. A continuación, se detallan estas etapas:

- **Captura de video:**

El primer paso consiste en adquirir una secuencia de video en tiempo real desde una cámara web o dispositivo de captura de video. En el código proporcionado, se utiliza la biblioteca OpenCV para inicializar y acceder a la cámara mediante la función `cv2.VideoCapture()`.

- **Preprocesamiento de imágenes:**

Antes de enviar los fotogramas de video al modelo de detección de manos, es necesario realizar un preprocesamiento. En este caso, se realiza un volteo horizontal de la imagen (`cv2.flip()`) para corregir la orientación de la cámara, y se convierte el espacio de color de BGR a RGB (`cv2.cvtColor()`), ya que el modelo de MediaPipe espera imágenes en el espacio de color RGB.

- **Detección de manos con MediaPipe:**

Una vez que los fotogramas están preparados, se envían al modelo de detección de manos de MediaPipe. En el código, se inicializa una instancia de la clase `Hands` de MediaPipe, configurando parámetros como el modo de imagen estática o dinámica, el número máximo de manos a detectar y el umbral de confianza para la detección.

Luego, en un bucle que se ejecuta continuamente, se procesa cada fotograma llamando al método `hands.process(frame_rgb)`. Este método devuelve los resultados de la detección, que incluyen información sobre las manos detectadas y la ubicación de los puntos de referencia en cada mano.

- **Renderizado de puntos de referencia:**

Si se detectan manos en el fotograma, el código utiliza la función `mp_drawing.draw_landmarks()` de MediaPipe para renderizar los puntos de referencia sobre la imagen original. Esta función dibuja círculos o líneas que representan las articulaciones y la estructura de la mano, lo que permite visualizar fácilmente la detección y el seguimiento de manos.

Finalmente, el fotograma procesado se muestra en una ventana utilizando `cv2.imshow()`.

### **Estructura lógica del código de detección de manos**

Este flujo se repite para cada fotograma de video, lo que permite una detección y seguimiento de manos en tiempo real. La información de los puntos de referencia obtenida en cada iteración se puede utilizar posteriormente para controlar los movimientos de la mano robótica,

mapeando las posiciones de los puntos de referencia a los ángulos y posiciones de los actuadores de la mano robótica.

- **Importación de bibliotecas y configuración inicial:**

En esta sección, se importan las bibliotecas necesarias para el proyecto y se configuran los módulos de MediaPipe que se utilizarán.

```
import cv2 # Biblioteca OpenCV para procesamiento de imágenes y video
import mediapipe as mp # Biblioteca MediaPipe de Google para visión
artificial

# Inicialización de Los módulos de MediaPipe
mp_drawing = mp.solutions.drawing_utils # Módulo para renderizar puntos de
referencia
mp_hands = mp.solutions.hands # Módulo para detección y seguimiento de manos
```

- **Inicialización de la cámara y el detector de manos:**

Aquí, se inicializa la cámara web utilizando OpenCV y se configura el detector de manos de MediaPipe con los parámetros deseados.

```
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW) # Inicialización de La cámara web

# Configuración del detector de manos de MediaPipe
with mp_hands.Hands(
    static_image_mode=False, # Modo de imagen dinámica (video)
    max_num_hands=2, # Número máximo de manos a detectar
    min_detection_confidence=0.5 # Umbral de confianza para La detección
) as hands:
    # Código para procesar cada fotograma de video
```

- **Bucle principal de procesamiento de fotogramas:**

En esta sección, se procesa cada fotograma de video capturado por la cámara web. Se realiza el preprocesamiento de la imagen y se envía al detector de manos de MediaPipe.

```
while True:
    ret, frame = cap.read() # Captura del fotograma de video
    if ret == False: # Si no se pudo capturar el fotograma, salir del bucle
        break

    height, width, _ = frame.shape # Obtener Las dimensiones del fotograma
    frame = cv2.flip(frame, 1) # Voltear La imagen horizontalmente
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convertir a espacio
de color RGB

    # Procesar el fotograma con el detector de manos de MediaPipe
    results = hands.process(frame_rgb)
```

- **Detección de manos y renderizado de puntos de referencia:**

Finalmente, se accede a los resultados devueltos por el detector de manos de MediaPipe y se renderiza la información de los puntos de referencia sobre la imagen original. Luego, se muestra el fotograma procesado en una ventana.

```
if results.multi_hand_landmarks is not None: # Si se detectaron manos
    for hand_landmarks in results.multi_hand_landmarks: # Para cada mano
        detectada
            # Renderizar Los puntos de referencia de La mano
            mp_drawing.draw_landmarks(
                frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)

cv2.imshow('Camara 1', frame) # Mostrar el fotograma procesado

if cv2.waitKey(1) & 0xFF == 27: # Salir del bucle si se presiona la tecla ESC
    break

cap.release() # Liberar la cámara
cv2.destroyAllWindows() # Cerrar todas las ventanas
```

### **Integración del sistema de detección de manos con el control de la mano robótica.**

El sistema de detección de manos basado en MediaPipe proporciona la información clave para controlar los movimientos de la mano robótica de manera precisa y en tiempo real. Específicamente, las coordenadas 2D o 3D de los 21 puntos de referencia detectados en cada fotograma se utilizan para determinar los ángulos y posiciones de los actuadores (servomotores o motores paso a paso) de la mano robótica.

La clave de esta integración radica en la utilización de la información de los Hand LandMarks detectados por MediaPipe en cada fotograma de video para controlar los actuadores de la mano robótica. A continuación, se detalla este proceso:

- **Mapeo de puntos de referencia:**

Las coordenadas de los puntos de referencia se mapean a los ángulos y posiciones específicas de los actuadores de la mano robótica. Por ejemplo, las coordenadas de los puntos de referencia en las puntas de los dedos se pueden mapear a los ángulos de los servomotores que controlan la flexión de cada dedo.

- **Cálculo de posiciones y distancias entre los Hand LandMarks:**

Una vez mapeados los puntos de referencia, se utilizan técnicas geométricas, se calculan los ángulos y posiciones necesarios para que la mano robótica imite la postura de la mano real

detectada. Estos cálculos pueden involucrar aspectos como la distancia entre puntos de referencia, los ángulos entre segmentos de dedos, la orientación de la palma, entre otros factores. Además, se deben considerar las restricciones físicas y limitaciones de movimiento de la mano robótica para asegurar que los ángulos y posiciones calculados sean factibles y seguros.

- **Envío de comandos a los actuadores:**

Una vez calculados los ángulos y posiciones deseados para cada actuador, se envían las instrucciones correspondientes a la mano robótica. Esto se puede realizar a través de una interfaz de programación de aplicaciones (API) o mediante comunicación serial con un microcontrolador o placa de control que controle los actuadores directamente.

En el caso de utilizar servomotores, las instrucciones pueden ser valores de ángulo o posición específicos. Si se utilizan motores paso a paso, las instrucciones pueden ser el número de pasos a avanzar o retroceder.

- **Frecuencia de actualización:**

Para garantizar una imitación fluida y en tiempo real de los movimientos de la mano real, los comandos a los actuadores de la mano robótica se actualizan en cada fotograma procesado por el sistema de detección de manos de MediaPipe. Esto significa que el ciclo de detección de manos, mapeo de puntos de referencia, cálculo de ángulos y posiciones, y envío de comandos se repite a la misma velocidad de fotogramas por segundo (FPS) que la cámara utilizada.

Idealmente, esta actualización se debe realizar con un retraso mínimo, de manera que la mano robótica responda de manera casi instantánea a los movimientos de la mano real.

- **Consideraciones y limitaciones:**

Es importante tener en cuenta que la precisión de los movimientos de la mano robótica dependerá de varios factores. En primer lugar, la precisión del sistema de detección de manos de MediaPipe y la calidad de la cámara utilizada tendrán un impacto directo en la exactitud de las coordenadas de los puntos de referencia.

Además, la resolución y capacidad de los actuadores de la mano robótica también influirán en la precisión de los movimientos. Actuadores con una resolución más alta y una mayor capacidad de carga permitirán una imitación más precisa de los movimientos de la mano real.

Otra consideración importante es la velocidad de respuesta del sistema. Factores como el rendimiento de la computadora, la velocidad de comunicación con la mano robótica y la latencia

en el procesamiento de imágenes pueden introducir retrasos que afecten la fluidez de los movimientos imitados.

Finalmente, es necesario considerar las limitaciones físicas de la mano robótica, como su capacidad de carga, rango de movimiento y resistencia a fuerzas externas. Estas limitaciones pueden restringir la manipulación de objetos pesados o voluminosos, así como ciertos movimientos complejos o de alta velocidad.

Para abordar estas consideraciones y limitaciones, es importante realizar pruebas exhaustivas y ajustes en el sistema, como la optimización del código, la selección adecuada de componentes y el diseño cuidadoso de la mano robótica.

En resumen, la integración del sistema de detección de manos basado en MediaPipe con el control de la mano robótica implica un mapeo preciso de los puntos de referencia, cálculos geométricos, envío de comandos a los actuadores y una actualización constante en tiempo real. Al abordar las consideraciones y limitaciones adecuadamente, se puede lograr una imitación precisa y fluida de los movimientos de la mano real, cumpliendo así con el objetivo principal de la tesis.

### 3.4.1.3. Etapa 2 - Programación de los Hand LandMarks

En el desarrollo de esta etapa se buscó obtener un vector de 5 elementos, con valores de 1 y 0 para realizar el movimiento de abrir o cerrar los dedos, cada elemento representa un dedo de la mano.

**Tabla 3**

*Valores enviados a través de la comunicación serial*

\$	1	1	1	0	1
	Pulgar	Índice	Medio	Anular	Meñique

*Nota:* Esta tabla presenta los valores enviados a través de la comunicación serial para el control de los dedos de una mano robótica. Los valores están organizados por dedo y su correspondiente señal digital enviada.

Se agregarán las bibliotecas HandTrackingModule, time y serial, para el seguimiento de las manos, un tiempo corto antes de iniciar la cámara y la comunicación serial con el dispositivo de Arduino respectivamente. Para la comunicación serial se tendrá en cuenta el puerto ('COM4') y la velocidad de datos en bits por segundo (BAUD) para la transmisión de datos en serie. Se creará un objeto "handDetector" utilizando el módulo personalizado de "HandTrackingModule", y una variable "detectionCon" para el umbral de confianza para la detección de manos.

En la Tabla 3 se muestra un ejemplo de los valores que pueden tomar cada uno de los dedos, en este caso todos los dedos se encuentran cerrados, excepto el dedo anular. Cada dedo que tome valor 0 estará cerrado y cada dedo que tome el valor de 1 estará abierto.

A continuación, se escogen los puntos de referencia de las puntas de los dedos identificados con MediaPipe, cada dedo tiene un último punto enumerado, y se almacenan en un vector de 5 elementos “tipIds”, que representa las posiciones de los puntos clave de la mano detectada. La lista “tipIds” contiene los siguientes valores:

“tipIds[0]”: Punto ultimo del dedo pulgar.

“tipIds[1]”: Punto ultimo del dedo índice.

“tipIds[2]”: Punto ultimo del dedo medio.

“tipIds[3]”: Punto ultimo del dedo anular.

“tipIds[4]”: Punto ultimo del dedo meñique.

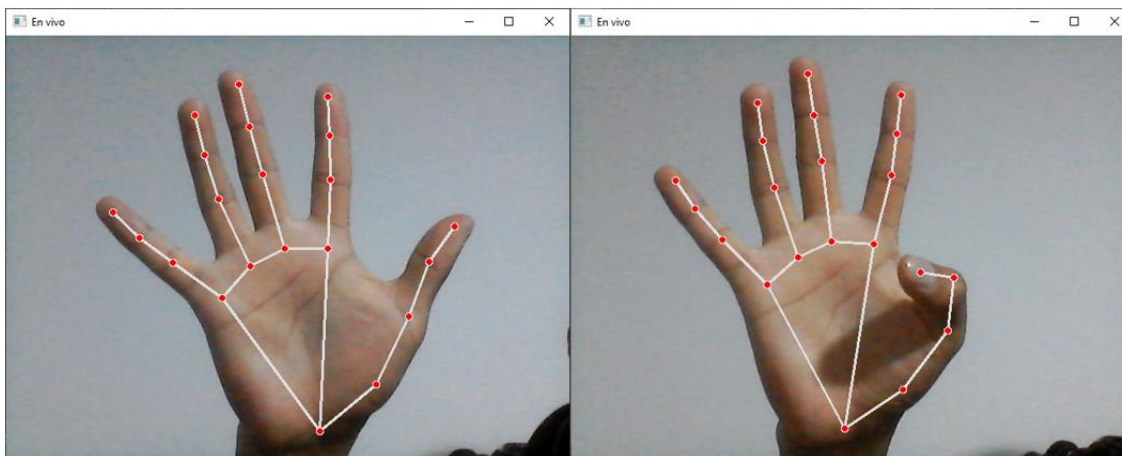
Los puntos enumerados respectivamente son  $tipIds = [4, 8, 12, 16, 20]$ . Estos valores se utilizan posteriormente para acceder a las coordenadas de las puntas de los dedos en “lmList”.

Después, se evaluará si el pulgar y los otros cuatro dedos están abiertos o cerrados. Los valores del vector “tipIds” serán comparados con otro vector que almacene los valores de los puntos inferiores a los de “tipIds”.

El valor [4] del dedo pulgar se evaluará en la coordenada Y, y comparada con el punto [2] del mismo dedo. Mientras el valor [4] esté a la derecha del valor [2], es decir, mientras que el dedo pulgar este abierto para la cámara, también estará abierto en el prototipo. Y cuando este esté a la izquierda (cuando el dedo pulgar esté cerrado para la cámara), también estará cerrado en el prototipo, tal como se muestra en la Figura 18.

**Figura 18**

*Evaluación de los Hand LandMarks en la coordenada Y con los puntos '4' y '2'*



*Nota:* Esta figura muestra una captura de pantalla del proceso de evaluación de Hand Landmarks en la coordenada Y. Se destacan los puntos '4' y '2', que son críticos para analizar el movimiento vertical de la mano. Este tipo de análisis es fundamental en la visión por computadora para aplicaciones que requieren seguimiento preciso de la mano, como la interpretación de lenguaje de señas, control de interfaces gestuales, o para la creación de sistemas de realidad aumentada que interactúan con los usuarios de manera natural y dinámica.

Para los dedos restantes se crea otro bucle para verificar y evaluar la profundidad, es decir en la coordenada Z, ya que los valores [8], [12], [16] y [20] serán evaluados, si alguno de ellos se encuentra por debajo de los puntos [6], [10], [14] y [18] respectivamente, querrá decir que los dedos están cerrados y se verá reflejado en el prototipo. En cambio, si se encuentran encima, los dedos de la mano y el prototipo estarán abiertos.

Con la programación descrita, a continuación, se inicializa una lista vacía “fingers” que se utilizará para almacenar el estado de cada dedo. Cuando el programa determine que el dedo esté cerrado almacenará un 0 en el vector “fingers” y cuando determine que el dedo esté abierto almacenará un 1 en el mismo vector, teniendo así independencia en cada dedo del prototipo.

Para enviar estos datos al dispositivo de Arduino, primero se convertirán en una cadena de caracteres que representa el estado de los dedos, el envío de datos será a través de la comunicación serial. Se usará la función “write” del objeto “ser” con una codificación en formato ASCII, que convertirá la cadena de caracteres en una secuencia de bytes para asegurar que los datos se transmitan y se interpreten correctamente entre los dispositivos.

#### **3.4.1.4. Etapa 3 – Movimiento de los dedos**

Para la programación de esta etapa se utilizará el entorno de desarrollo de Arduino, haciendo uso de la librería “<Servo.h>” que proporciona funciones y métodos que facilitan el control de servomotores en Arduino, proporcionando una interfaz para su uso y configuración de la posición angular de los servomotores.

Esta etapa servirá para poder realizar el desplazamiento de los dedos a través de los servomotores, la placa de desarrollo y el microcontrolador que procesarán la información descrita en la etapa 2 integrando además esta misma etapa.

El desarrollo de este código consiste en controlar los 5 servomotores mediante la comunicación serie que obtendremos en el vector arrojado por Python desde la computadora. Incluida ya la librería “<Servo.h>”, se definirán dos constantes para determinar el número de valores recibidos (5) y la cantidad de dígitos por valor (1). Se crearán después objetos Servo para controlar los cinco servomotores y se declararán variables para almacenar la información relacionada con la recepción de datos.

La función “receieveData()” se encargará de recibir y procesar los datos serie, cuando se detecta el inicio de un nuevo conjunto de datos marcado por el símbolo “\$”, se activa un contador y comienza a leer los caracteres siguientes. La cadena completa se almacena en “receivedString” y luego se divide para obtener valores individuales que se almacenan en el array “valsRec”.

En el bucle principal o “void loop” se llama a la función “receieveData” para procesar continuamente los datos serie. Luego, se ajustan los servomotores de acuerdo con los valores recibidos. Si un valor es 1, el servo se mueve a la posición de 180 grados; de lo contrario, si un valor es 0, se moverá a la posición de 0 grados.

Este código permite controlar cinco servomotores mediante la recepción de datos serie. Los datos se reciben de Python en el formato “\$XXXXX”, donde cada “X” es un dígito que determina el estado de un servo específico (1 para 180 grados, y 0 para 0 grados).

#### **3.4.1.5. Etapa 4 – Creación de la interfaz de usuario**

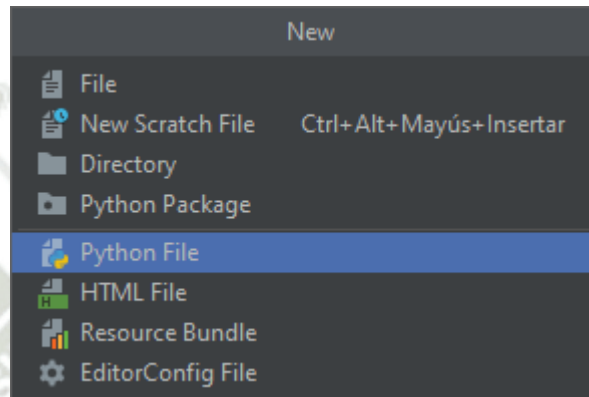
En esta etapa se realiza la interacción entre el sistema creado y el usuario a través de un programa en Python llevando el control con botones de encendido y pruebas antes de realizar el control de movimiento.

Para esto se desarrolló un programa capaz de recibir los datos de la cámara siguiendo los pasos de la etapa dos y tres.

Se hizo uso de la biblioteca estándar de Python “Tkinter” para la creación de la interfaz gráfica de usuario, esta biblioteca permite crear ventanas, botones, cuadros de texto, etiquetas y muchos otros elementos de interfaz gráfica.

### Figura 19

*Creación de archivo Python File*



Primero crearemos un nuevo proyecto de integración de interfaz y el código de las etapas dos y tres. Este proyecto tiene tres elementos principales:

1. Ventana principal: La ventana principal de la interfaz gráfica se crea utilizando ‘tk.Tk’ con la biblioteca instalada y se almacena en la variable ‘root’, esta ventana contiene todos los elementos de la interfaz.
2. Canvas: Este elemento de Tkinter sirve para mostrar el video capturado en tiempo real. Este elemento tiene una dimensión de 1080 x 720 pixeles.
3. Etiquetas y botones: Para crear y mostrar texto en la interfaz se utilizan etiquetas (‘label’); y los botones se usarán para realizar acciones específicas como encender la cámara, iniciar y detener la detección de manos, realizar pruebas antes de iniciar el control por visión artificial y controlar la apertura y cierre de la mano y cada uno de los dedos. Estos elementos se colocan en posiciones específicas utilizando coordenadas relativas ‘relx’ y ‘rely’, esto permite organizar y diseñar la disposición de los elementos de la ventana principal.

La interfaz gráfica contará con 9 botones de control de la mano y dedos para las pruebas y funcionalidades asociadas a cada uno de ellos.

El botón ‘Encender cámara principal’ (‘botonEncenderCamPrin’) inicia la captura de video sin realizar la detección de manos y muestra el feed de la cámara en el ‘canvas’, pero no envía datos seriales, tan solo para visualizar el video.

El botón ‘Iniciar control por Visión Artificial’ (‘botonIniciarControlVA’) inicia la detección de manos y envía los datos seriales, los datos de la posición de los dedos se procesan, y se envían los comandos a través de la comunicación serial con Arduino.

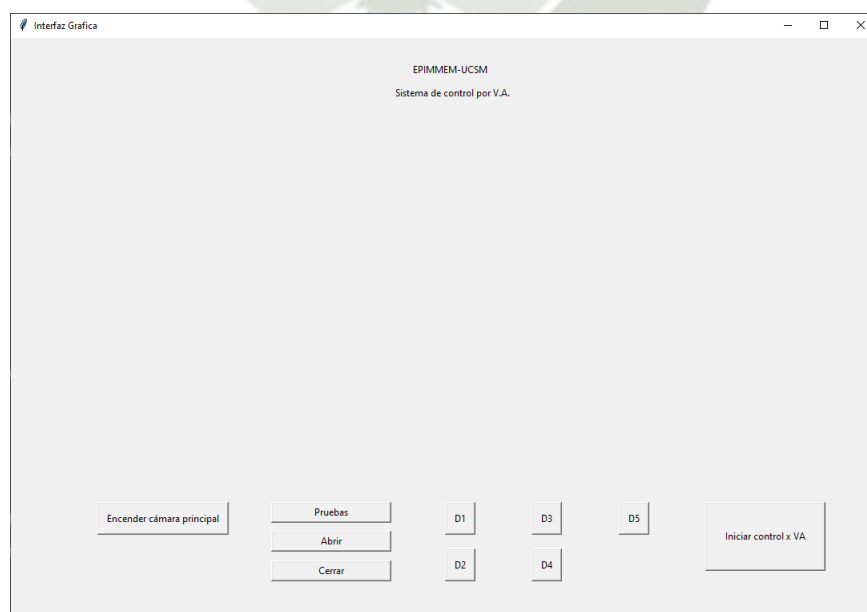
Los botones de pruebas son 7 independientes en funcionalidad, el botón ‘Abrir’ envía un comando para abrir completamente el prototipo de mano. El botón ‘cerrar’ envía un comando para cerrar completamente la mano. Con la creación de la interfaz también es posible hacer las pruebas en cada uno de los dedos individualmente, para ello son los botones ‘D1, D2, D3, D4 y D5’ que hacen referencia a los dedos pulgar, índice, medio, anular y meñique respectivamente (Ver Figura 20).

Estos botones permiten controlar la posición de los dedos de manera individual. Cada botón envía un comando a través de la conexión serial para manipular el estado del dedo correspondiente.

Estos controles proporcionan facilidad de interacción a la interfaz con el sistema de detección de manos y controlar la manipulación de la mano y los dedos. Al presionar estos botones se ejecutan funciones específicas que interactúan con el prototipo de la mano robótica y envían comandos a través de la conexión serial para controlar el hardware asociado.

### **Figura 20**

*Interfaz gráfica para el sistema de visión artificial*



### 3.4.2. Hardware

El hardware constituye los componentes físicos y electrónicos del proyecto, la infraestructura física sobre la cual se ejecutará la programación, y tiene la responsabilidad de llevar a cabo la ejecución y verificación del correcto funcionamiento del sistema a medida que interactúa con el software. Los componentes electrónicos están constituidos con la placa de desarrollo, los servomotores y la fuente de alimentación. Los componentes físicos constituyen principalmente la impresión 3D de la mano robótica y los materiales para su ensamblaje.

#### 3.4.2.1. Selección de la placa de desarrollo

Teniendo en cuenta las características necesarias del sistema a implementar, se considera el uso de 5 servomotores para cada dedo, por lo que es necesario una placa con al menos 5 pines de PWM para hacer el control de movimiento.

**Tabla 4**  
*Cuadro comparativo Arduino*

Características	Arduino			
	Uno	Nano	Micro	Mini
Microcontrolador	ATmega328P	ATmega328P	ATmega32U4	ATmega328P
Voltaje de funcionamiento	5v	5v	5v	5v/3.3v
Tamaño	68.6x53.3mm	18x45mm	18x44mm	30x18mm
E/S Digital/PWM	14/6	14/6	20/7	14/6
Frecuencia	16 MHz	16 MHz	16 MHz	16 MHz
Memoria flash	32KB	32KB	32KB	32KB
Memoria RAM	2KB	2KB	2.5KB	2KB

*Nota:* Esta tabla muestra un cuadro comparativo de diversas versiones de Arduino, destacando sus características principales como el microcontrolador utilizado, voltaje de funcionamiento, tamaño, entradas/salidas digitales y de modulación de ancho de pulso (PWM), frecuencia de operación, memoria flash y RAM.

La comunicación de la placa Arduino con el software de Python será Serial USB para la transferencia de datos con la computadora que tendrá las conexiones con las cámaras y el programa

en Python. Teniendo en cuenta los pines PWM de las placas Arduino en comparativa, por la relación de características bastante altas y suficientes, se selecciona la placa de desarrollo Arduino Uno. La Tabla 6 muestra un resumen comparativo de diferentes modelos de Arduino lo que permite la selección adecuada para las características requeridas.

Se selecciona los pines de los puertos que manejarán cada señal PWM del microcontrolador en la placa como se muestra en la Tabla 5.

**Tabla 5**

*Selección de pines para los servomotores*

Componente	Pin	Etiqueta
Servomotor Pulgar	PD9	Servo1
Servomotor Índice	PD10	Servo2
Servomotor Medio	PD11	Servo3
Servomotor Anular	PD5	Servo4
Servomotor Meñique	PD6	Servo5

*Nota:* Esta tabla detalla la asignación de los pines del microcontrolador para la conexión de los servomotores correspondientes a cada dedo de una mano robótica. Los pines están identificados tanto por su designación en el microcontrolador como por una etiqueta específica para facilitar la programación y el montaje.

### Controlador

Se hará uso de la tarjeta de adquisición de datos y plataforma de desarrollo Arduino, y el dispositivo Arduino Uno, en su placa de circuito impreso cuenta con un microcontrolador ATmega328P fabricado por Microchip Technology de 8 bits que funciona a 16 MHz, tiene las características detalladas a continuación:

**Tabla 6**  
*Especificaciones Arduino Uno*

<b>Especificaciones</b>	
Microcontrolador	ATMega328P
Velocidad de Reloj	de 16 MHz
Voltaje de Trabajo	de 5 v
Voltaje de Entrada	de 7,5 a 12 v
Pines de Salida	14 pines digitales (6 PWM) y 6 pines analógicos
Memoria	32KB Flash, 2KB RAM y 1KB Eeprom

*Nota:* Esta tabla resume las especificaciones técnicas del microcontrolador Arduino Uno, que es una plataforma de hardware libre basada en una placa con un microprocesador y pines de entrada/salida. Las características incluyen detalles sobre la velocidad de reloj, voltaje de operación, y capacidades de memoria, entre otros. Este tipo de información es esencial para diseñadores y desarrolladores que planean utilizar el Arduino Uno en proyectos de electrónica y robótica. *Adaptado de Arduino Data Sheet.*

Teniendo en cuenta ficha de datos técnicos del Arduino Uno, se realizó la Tabla 2 donde se resume sus principales características. El Arduino Uno es una de las placas más populares en la familia de Arduino debido a su versatilidad y facilidad de uso. Los proyectos creados con este dispositivo pueden variar desde simples ejemplos educativos hasta sistemas más complejos que involucran sensores, actuadores y comunicación con otros dispositivos.

#### **3.4.2.2. Servomotores**

El sistema de control tiene 5 actuadores, el ciclo de trabajo de la señal PWM se ajusta para controlar la potencia entregada a los componentes actuadores. Se usarán los servomotores del tipo MG955 (Figura 21). Estas se seleccionan por recomendación del espacio establecido en la mano robótica de InMoov, ya que no todos los servomotores tienen el mismo tamaño y es un tamaño estándar el cual encaja con las dimensiones del prototipo impreso en 3D y con la magnitud de fuerza requerida para poder sostener objetos. A continuación, se hará el cálculo de la cantidad de corriente que necesitan los servomotores MG995 seleccionados con los datos brindados por la marca como voltaje, velocidad y torque.

**Figura 21**  
*Servomotor MG995*



*Nota:* La imagen presenta un servomotor MG995, ampliamente utilizado en robótica por su durabilidad y capacidad para manejar cargas pesadas. Este servomotor es capaz de rotar hasta 180 grados y es preferido en aplicaciones que requieren movimientos controlados y precisos. Su diseño estándar y conectividad facilitan la integración con diversos controladores y sistemas electrónicos. *Adaptado de Servo Motor TowerPro MG995 180 Degrees.*

Datos:

Voltaje: 4.8v.

Velocidad: 0.2 sec/60°.

Torque: 8.5 Kgf.cm a 4.8v

Realizamos la conversión de unidades al torque de Kgf.cm a Nm:

$$T = 8.5 \text{ Kgf.cm} * \frac{9.81\text{N}}{1\text{Kgf}} * \frac{1\text{m}}{100\text{cm}}$$

$$T = 0.83385\text{N.m}$$

Convertimos las unidades de la velocidad a velocidad angular

$$w = \frac{\text{rad}}{\text{seg}}$$

$$w = \frac{60 * \frac{\pi}{180}}{0.2}$$

$$w = 5.24 \frac{\text{rad}}{\text{seg}}$$

Ahora calculamos la potencia demandada por el servomotor:

$$P = \tau * \omega$$

Donde:

*P: Potencia*

*τ: Torque*

*ω: Velocidad angular*

$$P = 0.83385 \text{ N.m} * 5.24 \frac{\text{rad}}{\text{seg}}$$

$$P = 4.37 \text{ W}$$

Corriente en cada servomotor

$$P = V * I$$

Donde:

*P: Potencia*

*V: Voltaje*

*I: Corriente*

Despejando la corriente y reemplazando los valores ya conocidos, tenemos:

$$I = \frac{P}{V}$$

$$I = \frac{4.37 \text{ W}}{4.8 \text{ V}}$$

$$I = 910.42 \text{ mA}$$

Entonces la corriente total consumida por los 5 servomotores será:

$$I_{Total} = I * N$$

Donde:

*N: Número de servomotores usados*

$$I_{Total} = 910.42 * 5$$

$$I_{Total} = 4552.1 \text{ mA}$$

### Actuador

Se hará uso de servomotores MG995 para el movimiento independiente de cada uno de los dedos de la mano robótica.

El MG995 es un tipo de servomotor que se usa comúnmente en robótica y proyectos que requieren un control preciso del movimiento. Es un servomotor digital de alto par con engranaje metálico que viene en un tamaño estándar y es capaz de entregar hasta 10 kgf.cm de torque a 6 v. El MG995 es utilizado por su resistencia y durabilidad, gracias a sus engranajes metálicos y rodamientos dobles. Este dispositivo funciona con pulsos eléctricos, o PWM desde la plataforma y dispositivo Arduino, proporcionando un ángulo de funcionamiento de aproximadamente 180°. El MG995 viene con 3 brazos, tornillos y los cables de alimentación, tierra y de control.

**Tabla 7**  
*Especificaciones Servomotor MG995*

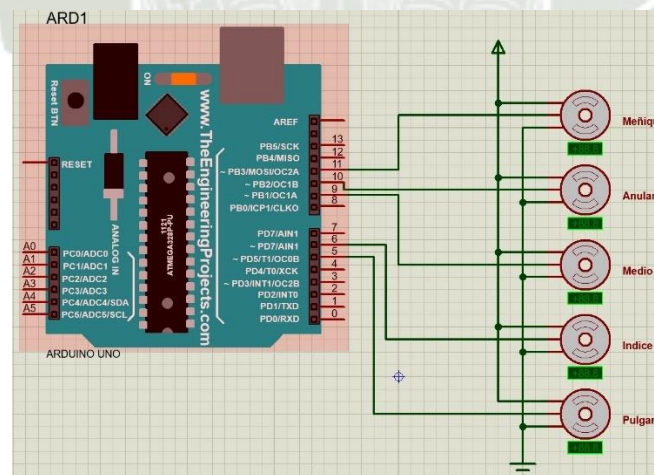
Especificaciones	
Peso	55g
Dimensiones	40,7 x 19,7 x 42,9 mm
Par o Torque	8,5 kgf.cm (4,8 v); 10 kgf.cm (6v)
Velocidad de Funcionamiento	0,2/60° (4,8 v); 0,16s/60° (6 v)
Voltaje de Funcionamiento	4,8 a 7,2 v
Tipo de engranaje	engranajes metálicos
Rango de Temperatura	0 °C a 55 °C

*Nota:* La tabla presenta un resumen de las características técnicas del servomotor MG995, destacando su peso, dimensiones, par o torque, velocidad de operación, voltaje de funcionamiento, tipo de engranajes y rango de temperatura operativa. Estos datos son críticos para diseñadores e ingenieros al seleccionar servomotores para proyectos de robótica y automatización. *Adaptado de Data Sheet MG995.*

Teniendo en cuenta la ficha de datos técnicos de los servomotores MG995, se realizó la Tabla 7 donde se resume sus principales características y especificaciones.

### Esquemático del Circuito

**Figura 22**  
*Esquemático del Circuito*



*Nota:* El esquemático representa la interconexión de los servomotores controlados por la placa Arduino Uno, utilizando la modulación de ancho de pulso (PWM) para la operación de un prototipo de mano robótica. Cada servomotor está asignado a un dedo específico, lo que permite un control detallado y una manipulación precisa.

- **Placa Arduino Uno:**

La placa Arduino Uno es un microcontrolador basado en el ATmega328P. Es ampliamente utilizado en robótica y prototipos electrónicos debido a su facilidad de uso y versatilidad. En tu

esquemático, se muestra como el cerebro del circuito, encargado de enviar señales de control a los servomotores.

- **Conexiones PWM:**

Los servomotores se controlan mediante señales de modulación de ancho de pulso (PWM). En la placa Arduino Uno, los pines digitales 3, 5, 6, 9, y 10 son capaces de producir tales señales. En el esquemático, cada uno de estos pines está conectado a un servomotor, lo que permite controlar individualmente el ángulo de cada dedo del prototipo de mano robótica.

- **Servomotores:**

Los círculos etiquetados con los nombres de los dedos representan los servomotores, cada uno correspondiente a un dedo en la mano robótica. Los servomotores son actuadores rotativos que permiten un control preciso del movimiento y la posición. En este caso, el modelo MG995 se ha elegido por su torque adecuado y dimensiones compatibles con el diseño del prototipo.

- **Líneas de Conexión:**

Las líneas verdes representan las conexiones eléctricas entre los pines de salida PWM del Arduino y los servomotores. Es fundamental que las conexiones sean robustas y seguras para garantizar una transmisión de señal fiable.

- **Fuente de Alimentación:**

Aunque no se muestra explícitamente en el esquemático, cada servomotor requiere una fuente de alimentación adecuada, que generalmente es de 4.8V a 6V para el modelo MG995. Dada la corriente potencialmente alta que los servomotores pueden requerir cuando operan bajo carga, es recomendable utilizar una fuente de alimentación externa en lugar de alimentarlos directamente desde la placa Arduino para evitar dañar el microcontrolador.

- **Tierra (GND):**

La línea negra en el esquemático probablemente representa la conexión a tierra (GND). Es crucial conectar todos los GND de los servomotores y la placa Arduino a un punto común para evitar bucles de tierra y asegurar un funcionamiento estable del sistema.

### **Justificación del Diseño:**

- **Eficiencia en la selección de pines:**

La elección de pines PWM en el Arduino Uno permite un control fino del ángulo de rotación de los servomotores, lo cual es esencial para simular con precisión el movimiento de los dedos humanos.

- **Compatibilidad y estandarización:**

Utilizar una placa Arduino Uno y servomotores MG995 garantiza que el proyecto sea replicable y accesible, ya que ambos componentes son estándares en la industria y fáciles de adquirir.

- **Adaptabilidad:**

El uso de un microcontrolador programable como el Arduino Uno permite la flexibilidad para cambiar o ajustar el software y el control del hardware sin necesidad de alterar el circuito físico, facilitando la iteración y mejora del prototipo a lo largo del tiempo.

- **Seguridad del Hardware:**

Al alimentar los servomotores de forma externa y no directamente desde la placa Arduino, se protege al microcontrolador de posibles sobrecargas de corriente que podrían dañarlo. Además, se garantiza que los servomotores dispongan de la corriente necesaria para su óptimo funcionamiento.

- **Control Preciso:**

La elección de pines PWM específicos que tienen una resolución de señal más alta en el Arduino Uno permite un control más preciso de los servomotores, lo cual es crucial para movimientos delicados y precisos requeridos en las manos robóticas.

- **Diseño Modular:**

El esquemático muestra un diseño que facilita la modularidad. Cada servomotor puede ser controlado de manera independiente, lo que significa que cada dedo de la mano robótica puede moverse por separado, permitiendo una gama más amplia de movimientos y gestos.

- **Capacidad de Diagnóstico:**

Al utilizar un microcontrolador con capacidades de comunicación digital, es posible agregar diagnósticos y retroalimentación en el software para monitorear el estado y el rendimiento de cada servomotor, lo cual es esencial para la detección de errores y el mantenimiento preventivo.

- **Escalabilidad:**

El diseño es escalable, permitiendo la adición de más servomotores o la integración de otros sensores y actuadores. Esto es particularmente útil para futuras expansiones del proyecto, como la adición de un sentido del tacto o la integración de más grados de libertad.

- **Documentación y Comunidad de Soporte:**

El uso de componentes populares como Arduino y servomotores MG995 asegura el acceso a una amplia documentación y una comunidad activa, facilitando la solución de problemas y el intercambio de ideas.

### 3.4.2.3. Alimentación

Los servomotores podrían ir conectados a la placa de Arduino, sin embargo, por la corriente consumida por el total de servomotores, la placa podría dañarse o presentar fallas, ya que ésta, al estar conectada a la computadora, no podrá entregar la suficiente cantidad de corriente eléctrica para todos los servomotores actuando en simultáneo.

Siendo esta la situación se requiere de una fuente de alimentación externa al Arduino, debiéndose usar 2 convertidores de nivel de tensión. Primero un convertidor AC-DC que realizará el cambio de 220 VAC a 19 VDC y el segundo convertidor DC-DC que reducirá el voltaje a 5VDC alimentando a los 5 servomotores.

Para seleccionar esta fuente de alimentación nos centraremos en el consumo de energía que consumen los 5 servomotores, entonces la corriente máxima consumida es:

$$I_{Total} = 4552.1 \text{ mA}$$

A partir de este dato podemos determinar el conversor de voltaje POLOLU D24V50F5 (Figura 23) que tiene una salida de 5VDC con una corriente continua típica de hasta 5000 mA.

#### Figura 23

*Regulador de voltaje reductor Pololu D24V50F5*



*Nota:* Esta figura ilustra un regulador de voltaje reductor Pololu D24V50F5, utilizado para proporcionar una salida de 5V desde una fuente de voltaje superior. Diseñado para ser eficiente y compacto, es una solución comúnmente implementada en proyectos de robótica y electrónica donde se necesitan voltajes estabilizados para componentes sensibles. *Adaptado de Pololu - 5V, 5A Step-Down Voltage Regulator D24V50F5.*

La placa de desarrollo seleccionada se alimentará por el puerto USB ya que requiere de la comunicación serial con la interfaz gráfica desarrollada y esta solo enviará las señales de control a través de los pines PWM.

#### 3.4.2.4. Impresión en 3D

El estudio de los compuestos en materiales para la impresión en 3D permite dar solución y seleccionar el material correspondiente para la mano robótica, considerando que los más utilizados en las impresiones 3D son el ABS y el PLA.

**Tabla 8**  
*Características de materiales de impresión 3D*

Material	Densidad (g/cm <sup>3</sup> )	Resistencia a la tracción (MPa)	Temperatura de impresión (°C)	Biodegradable
ABS	1.04	40 - 50	240 - 260	No
PLA	1.24	37 - 54	190 - 230	Si

*Nota:* Esta tabla detalla las propiedades de dos materiales comúnmente usados en impresión 3D, el ABS y el PLA. Se comparan factores clave como la densidad, la resistencia a la tracción, la temperatura de impresión y si son biodegradables o no. Estos datos son fundamentales para decidir qué material utilizar en función de las necesidades específicas de durabilidad, resistencia y sostenibilidad ambiental de un proyecto.

En la Tabla 8 se describen algunas características relevantes de los materiales ABS y PLA para impresión en 3D. Considerando estas características, el material más adecuado para la fabricación de la mano robótica es el ABS que es un polímero compuesto por tres bloques, acrilonitrilo, butadieno y estireno. Cada uno de los tres bloques aporta características distintas. El acrilonitrilo aporta rigidez, resistencia a ataques químicos, dureza y estabilidad a las altas temperaturas. El butadieno, tenacidad a la temperatura cuando esta es especialmente baja y resistencia al impacto; y el estireno, resistencia mecánica, rigidez y dureza.

La impresión se realizó en materiales de filamento ABS y algunos engranajes en resina para la mayor durabilidad de las partes móviles (Figura 27).

Todos los archivos 3D diseñados o elaborados en los softwares de diseño asistido por computadora, necesitan transformarse en formato STL para poder ser materializados por una impresora 3D. En la Tabla 9 se describe los componentes por secciones de la mano robótica.

El hardware mecánico de la mano robótica InMoov de código abierto, se encuentra accesible a través de su página web, proporcionando a los usuarios la flexibilidad de replicar el

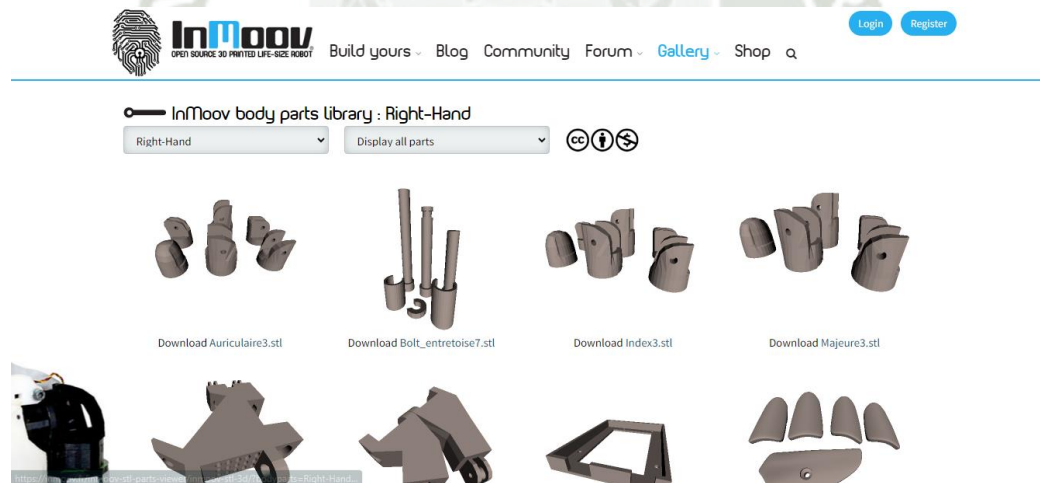
proyecto o realizar modificaciones según sus necesidades. Además, estos ya se encuentran en formato STL (Figura 24).

**Tabla 9**  
*Componentes de la mano robótica*

Componente	Cantidad
Dedo pulgar	5
Dedo índice	6
Dedo medio	6
Dedo anular	6
Dedo meñique	6
Palma	4
Sujeción en la palma	3
Muñeca	6
Antebrazo y Servos	15

*Nota:* La tabla resume la cantidad de componentes individuales que se requieren para ensamblar cada dedo de la mano robótica, incluyendo el pulgar, índice, medio, anular y meñique, así como la palma, sujeción de la palma, muñeca y el antebrazo junto con los servomotores necesarios. Este desglose es esencial para la gestión de inventario y planificación de la producción de manos robóticas.

**Figura 24**  
*Archivos STL en la página web de InMoov*



*Nota:* La figura muestra una selección de archivos STL disponibles en la página web de InMoov, que proporciona recursos abiertos para la impresión 3D de partes del robot InMoov, un proyecto de hardware libre y de código abierto. Esta comunidad en línea ofrece una plataforma para compartir diseños, mejoras y experiencias en la creación de robots humanoides. *Adaptado de InMoov - Open Source 3D Printable Life-Size Robot.*

**Figura 25**  
*Piezas impresas en 3D*



*Nota:* Esta figura muestra las piezas impresas en 3D que se utilizan para el ensamblaje de un dispositivo mecánico. Cada componente se ha fabricado individualmente para encajar dentro de una estructura más grande, demostrando el proceso detallado y las etapas de producción en un entorno doméstico.

#### **3.4.2.5. Armado de la mano**

Para armar y encajar todas las piezas fueron necesarias otras herramientas como destornilladores, taladro para circuitos impresos, herramienta giratoria, hilo de pescar, pinzas, alicates, adhesivos y selladores.

Teniendo todas las piezas impresas, se procedió al armado de la mano robótica empezando de los dedos articulados, la palma de la mano, muñeca y brazo.

**Figura 26**

*Tres articulaciones en un dedo*



*Nota:* Esta imagen muestra tres articulaciones de un dedo impresas en 3D, evidenciando la posibilidad de emular la mecánica del movimiento humano en aplicaciones de robótica y prótesis

Las piezas de los dedos se unen con pegamento acrílico y silicona líquida, para las uniones de los dedos se usaron clavos pasantes para dejar un libre movimiento giratorio entre las articulaciones.

**Figura 27**

*Unión de los dedos a la palma*

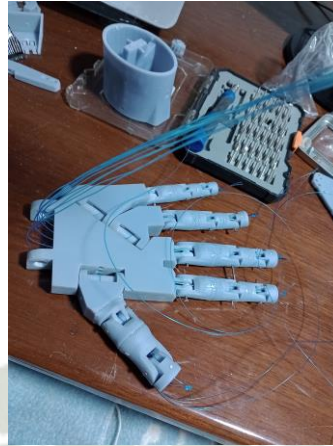


*Nota:* Esta imagen ilustra la fase de ensamblaje de la mano robótica, donde los dedos impresos en 3D están siendo unidos a la palma. Este proceso es crucial para la funcionalidad y movilidad de la mano en aplicaciones de robótica y prótesis avanzadas.

El hilo nylon o de pescar servirá para transmitir el movimiento angular de los servomotores a los dedos de la mano robótica y convertirse en movimiento lineal. Después de unir los eslabones de cada dedo y unirlos a la palma, para dar movimiento se atan los hilos nylon en cada dedo.

**Figura 28**

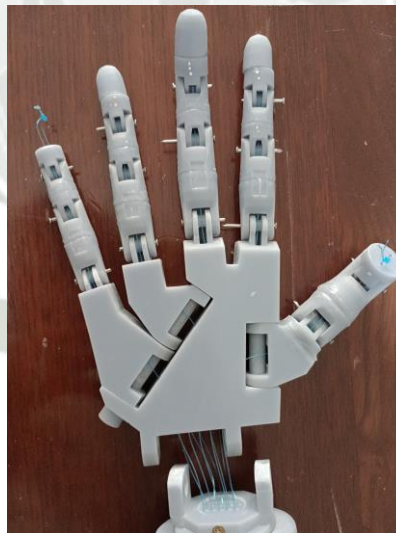
*Hilo nylon o de pescar que transmitirá el movimiento de los servos*



*Nota:* La imagen muestra el proceso de instalación del hilo de nylon, también conocido como hilo de pescar, que es fundamental para transmitir el movimiento generado por los servomotores a las articulaciones de los dedos de una mano robótica. Este método es una parte esencial del ensamblaje que asegura la funcionalidad mecánica del dispositivo.

**Figura 29**

*Palma de la mano robótica armada*



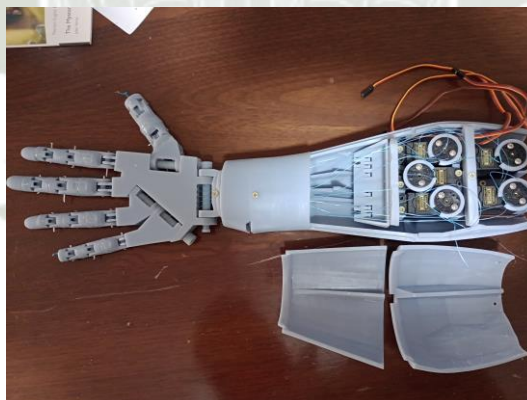
*Nota:* Esta imagen muestra la palma de mano robótica completamente ensamblada. La estructura muestra la disposición de los dedos y la palma, destacando la articulación y la conexión mecánica necesarias para la movilidad robótica. Los elementos visibles son clave para comprender el montaje y el funcionamiento de las prótesis robóticas y otros dispositivos de automatización que imitan la complejidad de la mano humana.

Para unir las piezas del brazo también se usa pegamento y silicona, además de los tornillos. El modelo InMoov impreso tiene un espacio para encajar los 5 servomotores los cuales se unen a los dedos del prototipo con los hilos de nylon tensados.

**Figura 30***Instalación de los servomotores MG995*

*Nota:* Esta figura muestra la instalación de los servomotores modelo MG995 en la estructura de la mano robótica. Los servomotores son esenciales para proporcionar el movimiento preciso de cada dedo en la prótesis robótica, actuando como músculos artificiales que simulan la función y la destreza de una mano humana.

Se imprimieron 5 poleas para cada servomotor y ser acoplados por uno de los componentes que viene incluido en este, por ellas pasarán los hilos y transmitirán el movimiento giratorio de los servomotores.

**Figura 31***Prototipo de mano robótica ensamblado*

*Nota:* La imagen muestra el ensamblaje completo del prototipo de mano robótica, incluyendo los dedos impresos en 3D y los servomotores MG995 instalados.

**3.5. Lista de Materiales**

La siguiente tabla muestra la lista de materiales usados para la fabricación, desarrollo e implementación del prototipo de mano robótica InMoov, así tenemos:

**Tabla 10**  
*Cuadro de lista de materiales*

Nombre de Materiales	Descripción	Cantidad	Costo (PEN)	Total (PEN)
Arduino	Arduino Uno	1	65	65
Protoboard	-	1	30	30
Servomotor	MG995	5	35	175
Regulador de voltaje	Pololu 5V, 5A	1	190	190
Adaptador Regulable AC-DC	220 VAC a 12 VDC	1	65	65
Impresión 3D	Varios elementos	1	650	650
Cables	Macho/Hembra	1	10	10
Hilo nylon de pescar	1 y 0.5 mm	1	20	20
Pegamento	Acrílico	1	24	24
Tornillos	Varios	1	10	10
Silicona	Liquida	1	10	10
Componentes electrónicos	Resistencias, diodos	1	20	20
<b>Total</b>				<b>1269</b>

*Nota:* Esta tabla proporciona un listado detallado de los materiales y componentes necesarios para el montaje de una mano robótica, junto con su descripción, cantidad requerida, costo individual y costo total en nuevos soles peruanos (PEN). Incluye elementos esenciales como el microcontrolador Arduino Uno, servomotores MG995, materiales para la impresión 3D y componentes electrónicos adicionales. Este cuadro es fundamental para la planificación del presupuesto y la adquisición de componentes para el proyecto.

### **Sustento Técnico de los Elementos Empleados en la Construcción del Sistema de Mano Robótica**

El diseño e implementación del sistema de mano robótica están fundamentados en decisiones técnicas meticulosas y consideraciones pragmáticas para lograr un prototipo funcional y eficiente. Cada elemento ha sido seleccionado y validado a través de un enfoque sistemático que involucra análisis comparativos, experimentación y optimización continua.

#### **1. Selección de la Placa de Desarrollo:**

La placa Arduino Uno, que incorpora el microcontrolador ATmega328P, fue seleccionada por su robustez, capacidad de procesamiento y amplia comunidad de soporte. Su entorno de desarrollo establecido y la disponibilidad de bibliotecas para control de actuadores facilitan la programación y el diseño de circuitos. La plataforma Arduino Uno ofrece 14 pines digitales de entrada/salida, de los cuales 6 proporcionan salida PWM (Pulse Width Modulation), esenciales

para el control de servomotores. Además, la frecuencia de operación de 16 MHz y la memoria flash de 32KB son suficientes para manejar la lógica de control compleja requerida por el sistema de visión artificial sin incurrir en retardos perceptibles.

## **2. Elección de Actuadores:**

Los servomotores MG995 se han seleccionado como actuadores para cada dedo de la mano robótica. Estos servomotores destacan por su relación torque-dimensiones, brindando un torque de hasta 10 kgf.cm a 6V, lo que es significativamente adecuado para el accionamiento de dedos robóticos que pueden requerir manipulación de objetos de variadas dimensiones y pesos. Su velocidad de operación de 0.2 segundos por 60 grados a 4.8V garantiza una respuesta ágil, lo que es crucial en aplicaciones de robótica donde la latencia puede resultar en una disminución del control y la precisión.

## **3. Configuración de Alimentación:**

La corriente total requerida por los servomotores MG995 supera la capacidad de la placa Arduino Uno cuando se alimenta a través de una conexión USB estándar. Por ello, se ha implementado una fuente de alimentación externa dedicada para los servomotores utilizando el regulador de voltaje reductor Pololu D24V50F5, que proporciona una salida estable de 5VDC con capacidad de corriente continua de hasta 5000 mA. Este regulador no solo previene los riesgos asociados a una sobrecarga de la placa Arduino, sino que también asegura una entrega de potencia constante y sin fluctuaciones, que podría afectar el rendimiento de los servomotores.

## **4. Materiales de Impresión 3D:**

Para la fabricación de los componentes físicos de la mano robótica, se ha optado por el material ABS sobre el PLA debido a su mayor resistencia a la tracción y su durabilidad superior. El ABS puede soportar temperaturas de impresión más altas (entre 240 y 260°C) y no es biodegradable, lo que resulta en piezas más resistentes y adecuadas para aplicaciones robóticas que requieren durabilidad. Aunque el PLA es conocido por ser más fácil de imprimir y biodegradable, no proporciona la misma resistencia mecánica ni la durabilidad a largo plazo que el ABS, lo cual es crítico en aplicaciones de robótica donde las piezas están sujetas a fuerzas repetitivas y estrés mecánico.

## **5. Diseño y Ensamblaje Mecánico:**

El diseño mecánico de la mano robótica ha sido optimizado para la impresión en 3D, permitiendo la creación de componentes con geometrías complejas que serían difíciles o costosas de fabricar utilizando métodos tradicionales. El ensamblaje de la mano utiliza clavos pasantes para las articulaciones, asegurando un movimiento suave y confiable. Además, el uso de hilo de nylon para conectar los dedos a los servomotores proporciona una solución ligera y resistente para la transmisión de fuerza, evitando la rigidez que limitaría la gama de movimiento.

#### **6. Integración y Pruebas del Sistema:**

La integración de estos componentes ha sido verificada mediante pruebas exhaustivas, incluyendo la validación de la comunicación serial entre la placa Arduino y la interfaz de usuario, así como la calibración de la respuesta de los servomotores a los comandos del software de visión artificial. Las pruebas han demostrado que la configuración actual satisface los requisitos de precisión, velocidad de respuesta y confiabilidad.

En conclusión, la elección de los elementos para construir el sistema de mano robótica no se ha tomado a la ligera, sino que se ha basado en consideraciones técnicas sólidas que justifican su adecuación para el proyecto. La placa Arduino Uno, los servomotores MG995, el material ABS para impresión 3D, y la configuración de la alimentación se han seleccionado para proporcionar un equilibrio entre rendimiento, costo, disponibilidad y capacidad de expansión para futuras mejoras y aplicaciones.

## CAPITULO IV: PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS

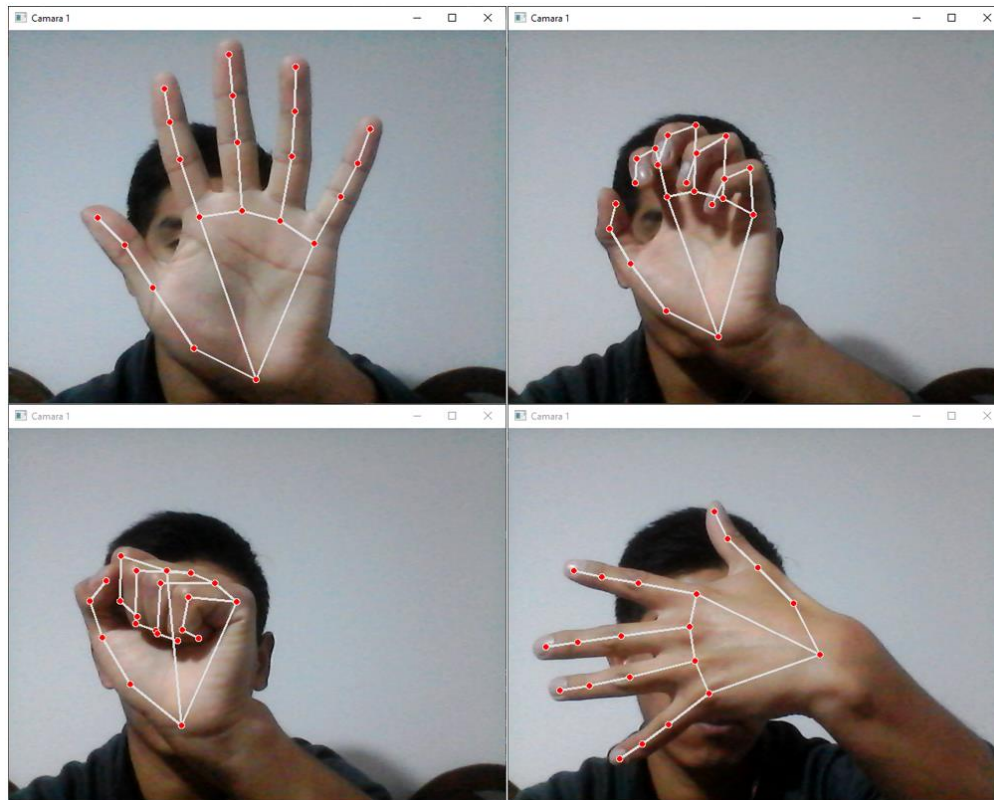
### 4. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS

Se realizan las pruebas de funcionamiento de cada etapa en secuencia que servirán para corroborar el correcto funcionamiento de la programación hecha en Python y Arduino en el prototipo y circuito ensamblado para el desarrollo de este capítulo. En cada etapa de este capítulo se describirá gráficamente los resultados y funcionamiento.

#### 4.1. Etapa 1 – Identificación de la mano humana

En esta etapa verificamos que el programa desarrollado en Python identifica las manos humanas, y brinda los 21 puntos o Hand LandMarks que nos proporciona MediaPipe. La cámara web, con ayuda de la visión artificial, es capaz de reconocer la mano humana en cualquier posición, cierre o apertura de los dedos

**Figura 32**  
*Identificación de la mano humana en la etapa 1*

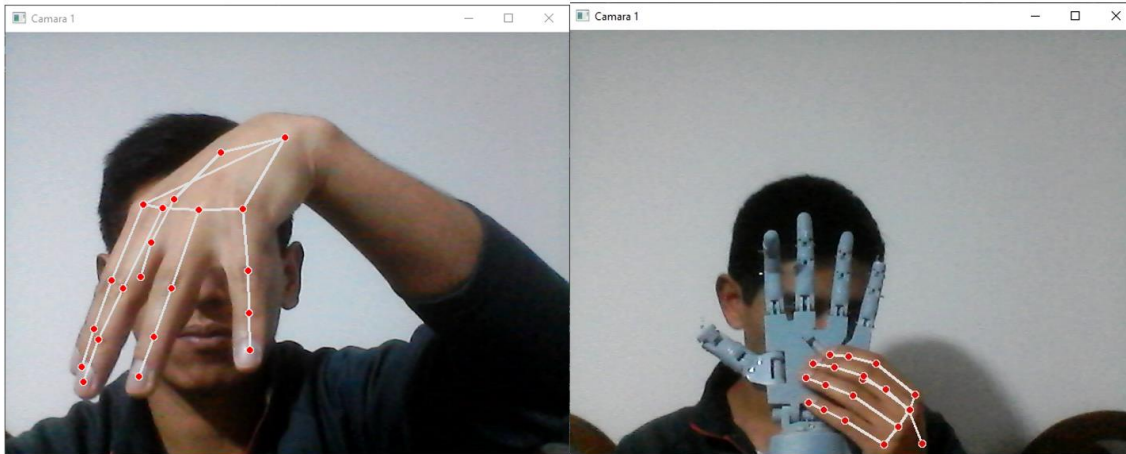


*Nota:* La imagen muestra la identificación de puntos clave en una mano humana mediante software de visión por computadora, lo que representa la primera etapa de un sistema para el control de una mano robótica.

El programa además identifica correctamente una mano humana y la diferencia del prototipo armado, la orientación de la mano puede estar hacia abajo y aun así la reconoce.

**Figura 33**

*Identificación de la mano humana en comparación a la mano robótica*

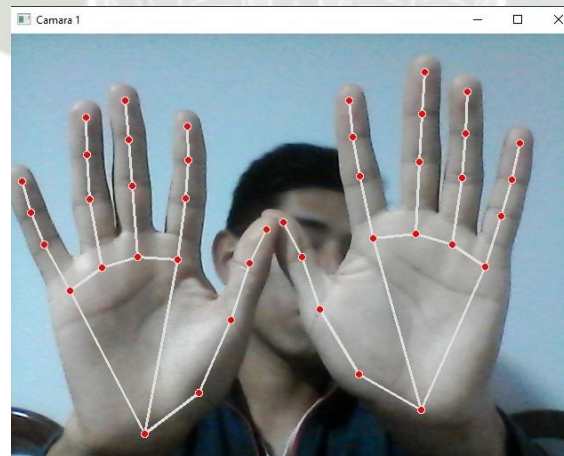


*Nota:* Esta imagen compara la identificación de puntos clave en una mano humana con su correspondencia en una mano robótica, mostrando la aplicación de software de visión por computadora para el mapeo y control de prótesis robóticas.

Como podemos apreciar el programa también es capaz de reconocer ambas manos del usuario, tanto la derecha como la izquierda, en ambas mostrando los Hand LandMarks.

**Figura 34**

*Dos manos identificadas*



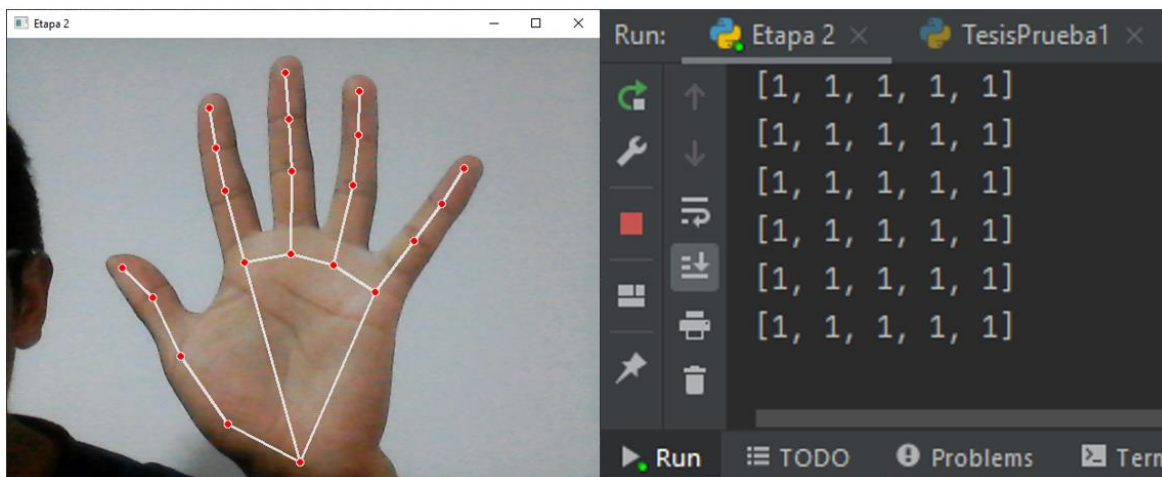
*Nota.* Esta imagen muestra la identificación de las marcas de referencia en dos manos humanas utilizando software de visión por computadora. La detección precisa de puntos clave en las manos es esencial para aplicaciones que van desde la interpretación del lenguaje de señas hasta el control de prótesis robóticas avanzadas.

#### 4.2. Etapa 2 – Programación de los Hand LandMarks

Primero iniciamos la cámara web y en la ventana inferior el programa arrojará números según si la mano está cerrada o abierta.

**Figura 35**

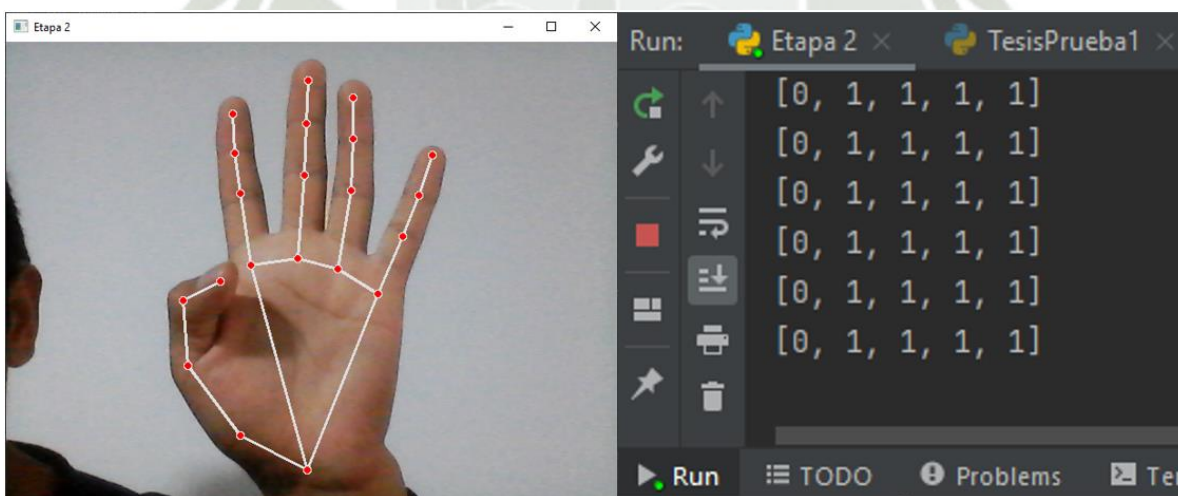
*Todos los dedos abiertos, el mensaje enviado será \$11111*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando el software de visión artificial, que detecta y resalta los puntos clave en la mano. El mensaje "\$11111" indica que todos los dedos están abiertos, lo cual es parte de la comunicación serial para el control de una mano robótica.

**Figura 36**

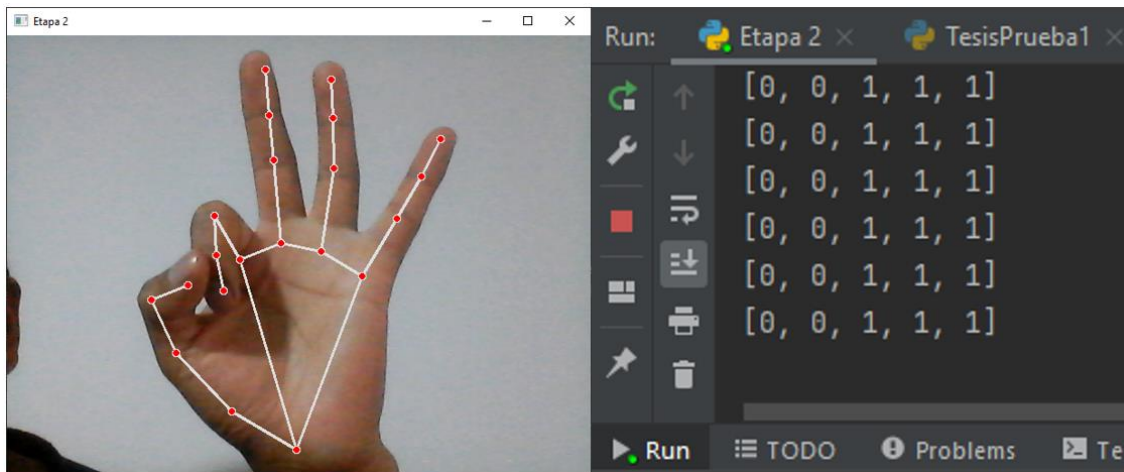
*Algunos dedos abiertos, el mensaje enviado será \$01111*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$01111" indica que el dedo pulgar está cerrado (0) y los demás dedos están abiertos (1). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

**Figura 37**

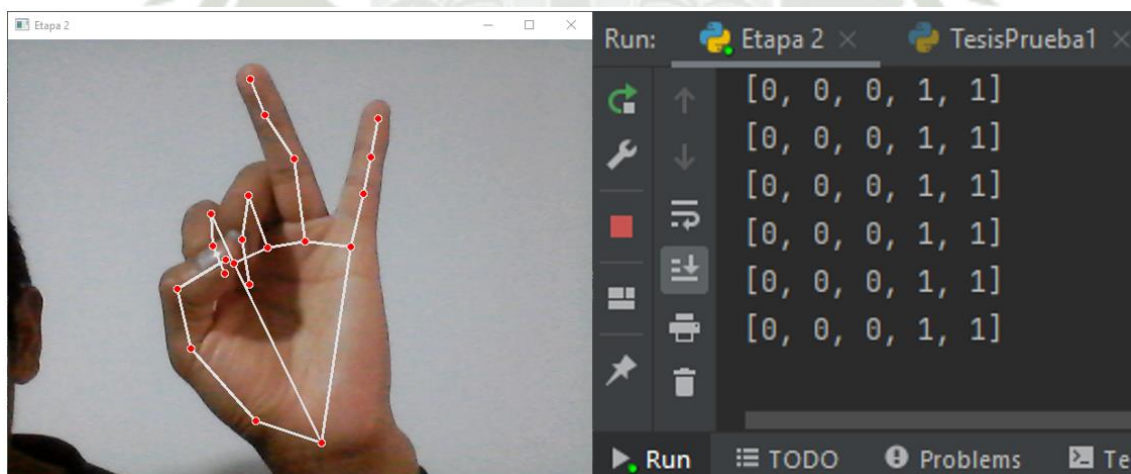
*Algunos dedos abiertos, el mensaje enviado será \$00111*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$00111" indica que el dedo pulgar y el dedo medio están cerrados (0), mientras que el índice, anular y meñique están abiertos (1). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

**Figura 38**

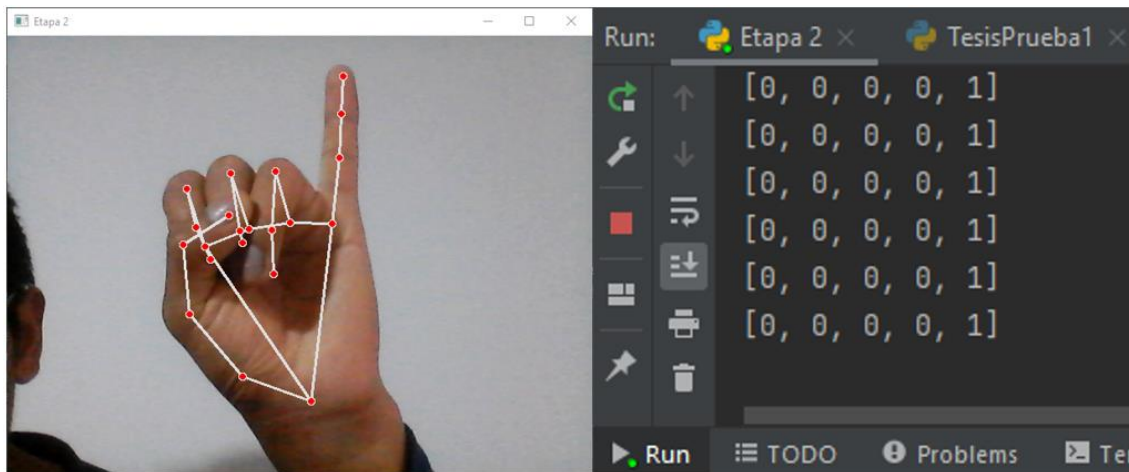
*Algunos dedos abiertos, el mensaje enviado será \$00011*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$00011" indica que solo el dedo meñique está abierto (1) mientras que los demás dedos están cerrados (0). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

**Figura 39**

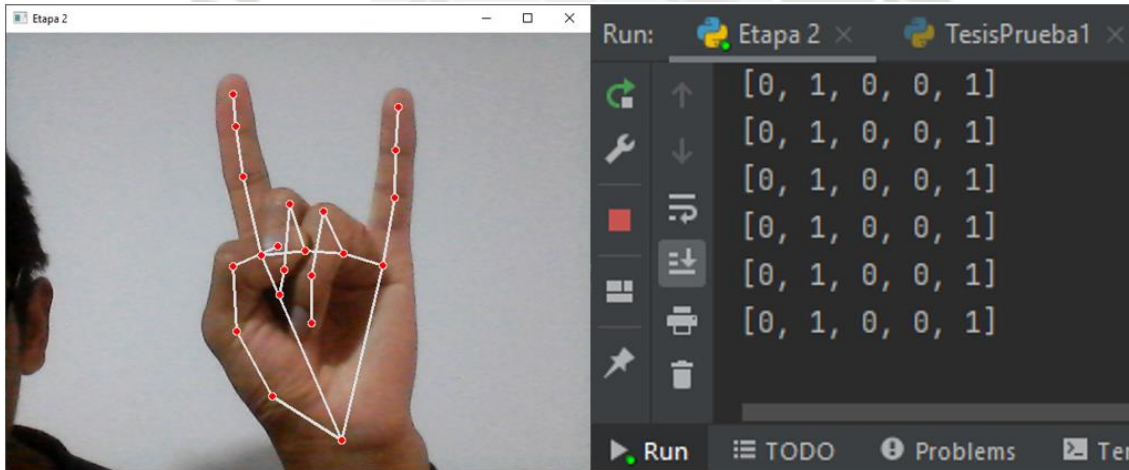
*Algunos dedos abiertos, el mensaje enviado será \$00001*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$00001" indica que solo el dedo índice está abierto (1) mientras que los demás dedos están cerrados (0). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

**Figura 40**

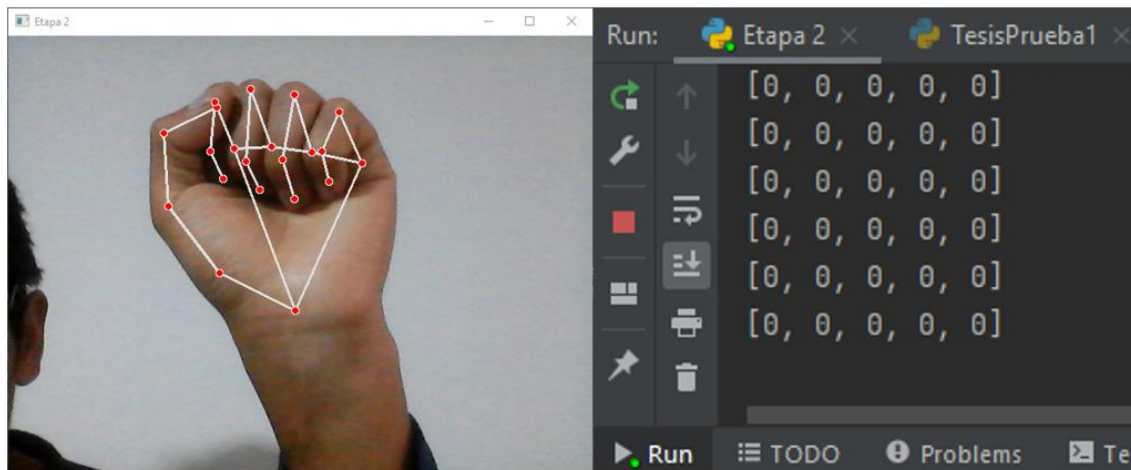
*Algunos dedos abiertos, el mensaje enviado será \$01001*



*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$01001" indica que el dedo pulgar (0) y el dedo índice (1) están abiertos, mientras que los demás dedos están cerrados (0). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

**Figura 41**

*Todos los dedos cerrados, el mensaje enviado será \$00000*



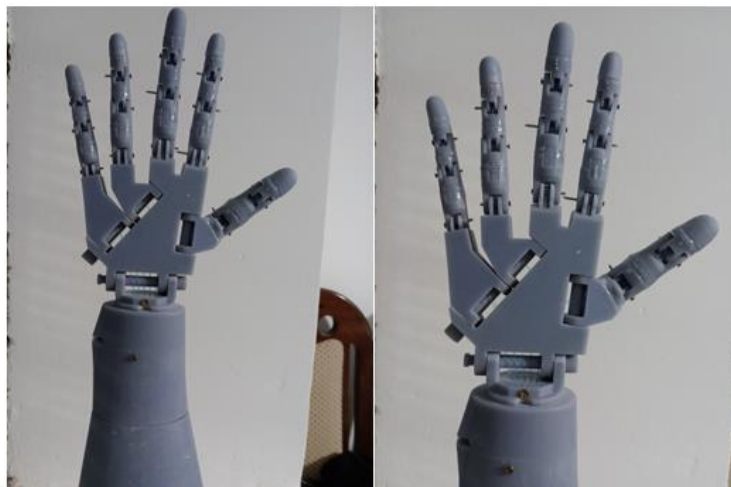
*Nota:* Esta imagen muestra el resultado del sistema de reconocimiento de gestos de la mano humana utilizando software de visión artificial. El software detecta y resalta los puntos clave en la mano para determinar la posición de los dedos. El mensaje "\$00000" indica que todos los dedos están cerrados (0). Este gesto puede ser parte de la comunicación serial para el control de una mano robótica.

### 4.3. Etapa 3 – Movimiento de los dedos

Los resultados de esta etapa van de la mano con la interfaz gráfica ya creada, aplicándose el control por visión artificial se imitan los movimientos de la mano humana captadas por la cámara web.

**Figura 42**

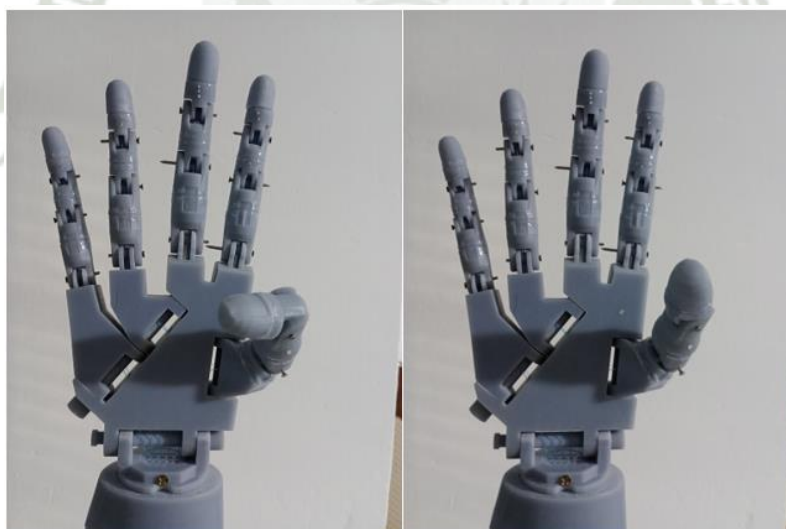
*Todos los dedos abiertos en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con todos los dedos abiertos. El prototipo de la mano robótica también refleja esta posición, con todos sus dedos extendidos.

**Figura 43**

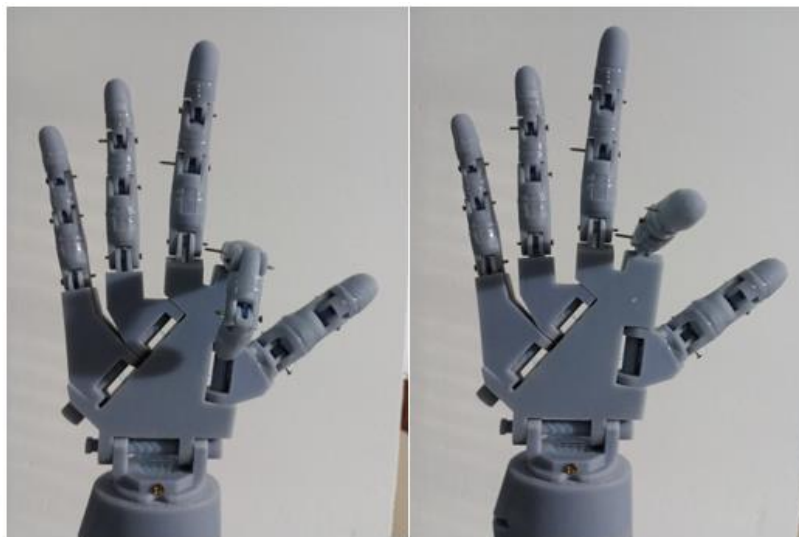
*Dedo pulgar cerrado en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con el dedo pulgar cerrado. El prototipo de la mano robótica también refleja esta posición, con el dedo pulgar flexionado.

**Figura 44**

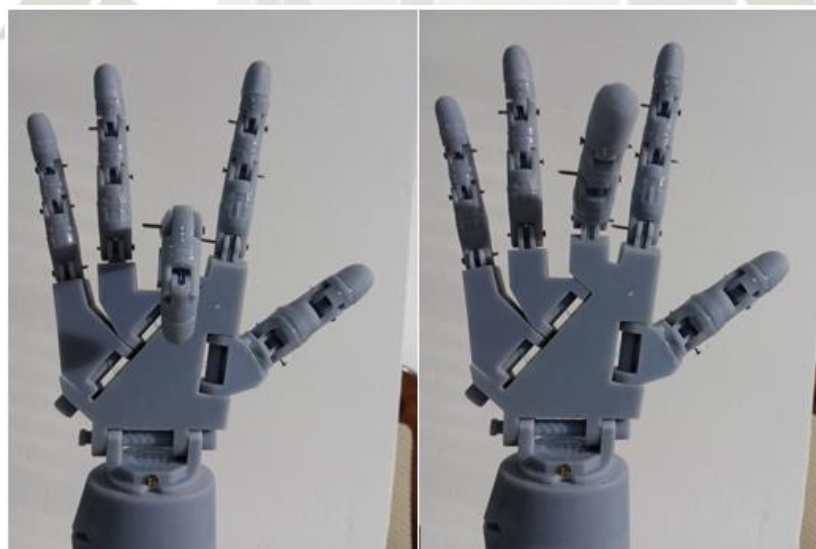
*Dedo índice cerrado en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con el dedo índice cerrado. El prototipo de la mano robótica también refleja esta posición, con el dedo índice flexionado.

**Figura 45**

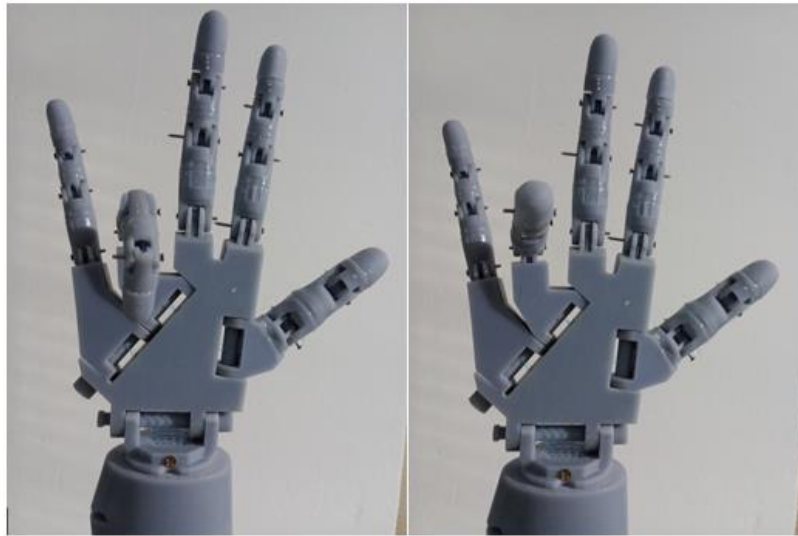
*Dedo medio cerrado en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con el dedo medio cerrado. El prototipo de la mano robótica también refleja esta posición, con el dedo medio flexionado.

### Figura 46

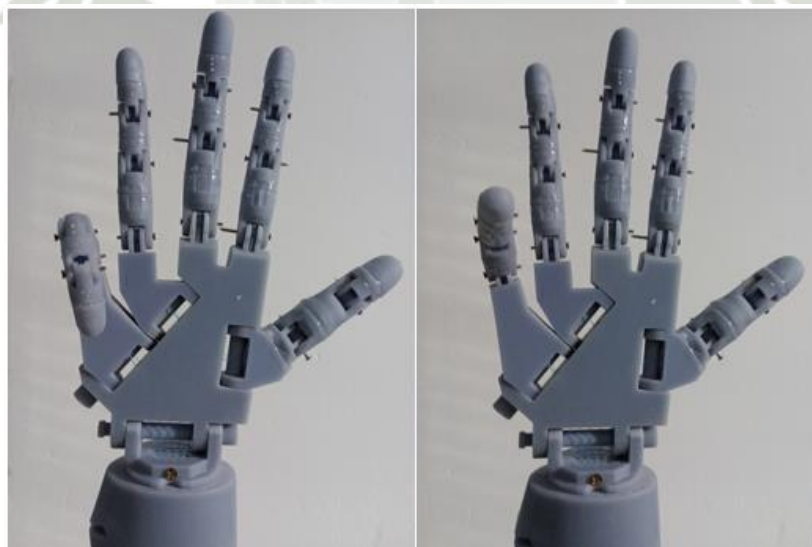
*Dedo anular cerrado en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con el dedo anular cerrado. El prototipo de la mano robótica también refleja esta posición, con el dedo anular flexionado.

### Figura 47

*Dedo meñique cerrado en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con el dedo meñique cerrado. El prototipo de la mano robótica también refleja esta posición, con el dedo meñique flexionado.

**Figura 48**

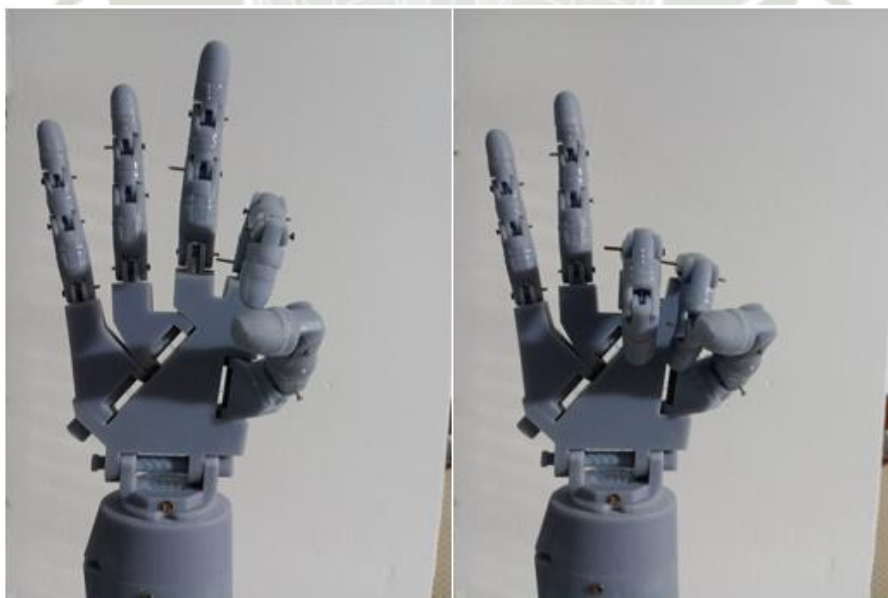
*Todos los dedos cerrados en la interfaz y el prototipo*



*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observa una imagen de una mano con todos los dedos cerrados. El prototipo de la mano robótica también refleja esta posición, con todos sus dedos flexionados.

**Figura 49**

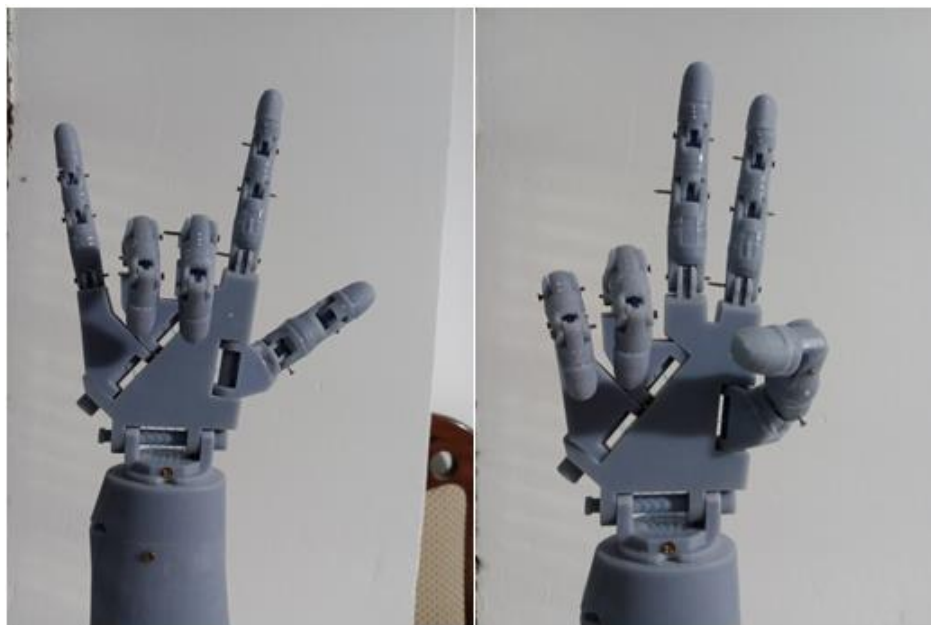
*Agarres tipo pinza, pulgar e índice y medio en la interfaz y el prototipo*



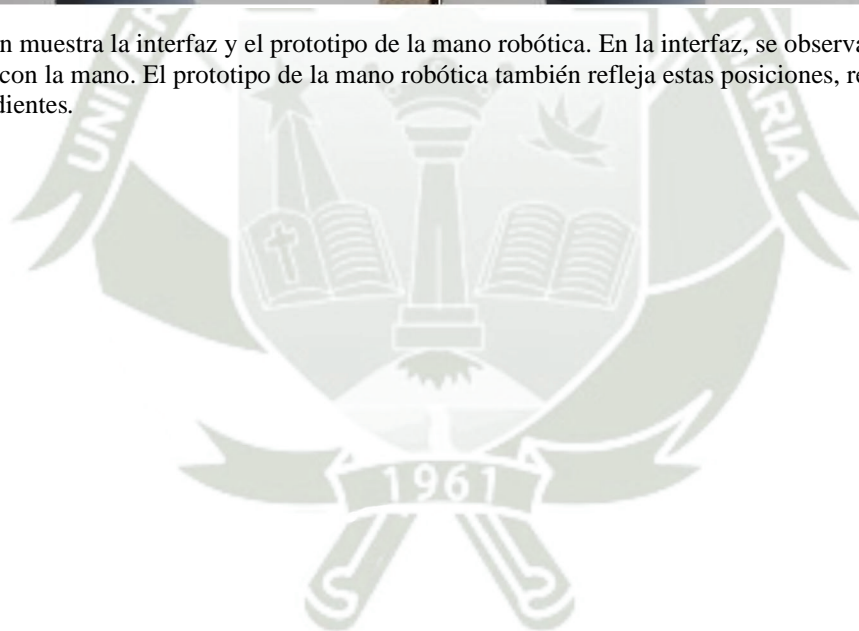
*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observan imágenes de dos tipos de agarres: pinza con pulgar e índice y pinza con pulgar y dedo medio. El prototipo de la mano robótica también refleja estas posiciones, realizando los agarres correspondientes.

**Figura 50**

*Gestos con la mano en la interfaz y el prototipo*



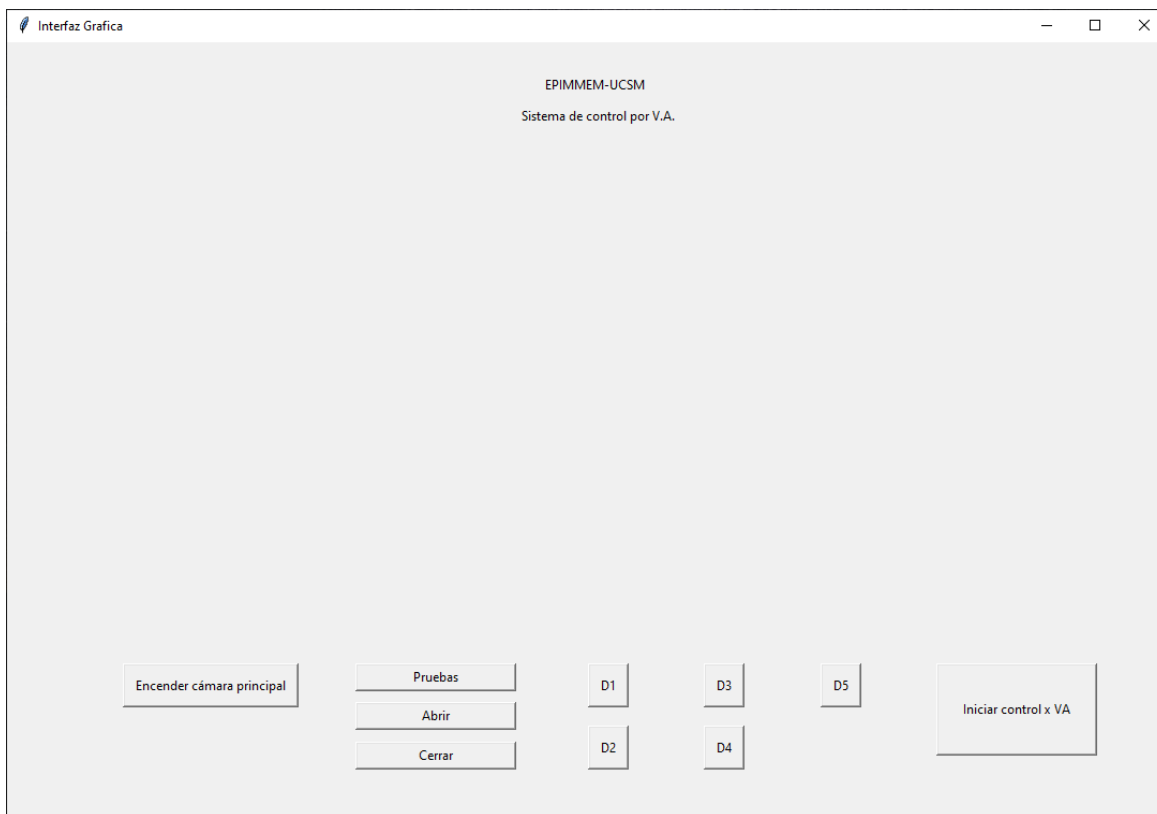
*Nota:* Esta imagen muestra la interfaz y el prototipo de la mano robótica. En la interfaz, se observan imágenes de diferentes gestos con la mano. El prototipo de la mano robótica también refleja estas posiciones, realizando los gestos correspondientes.



#### 4.4. Etapa 4 – Creación de la interfaz de usuario

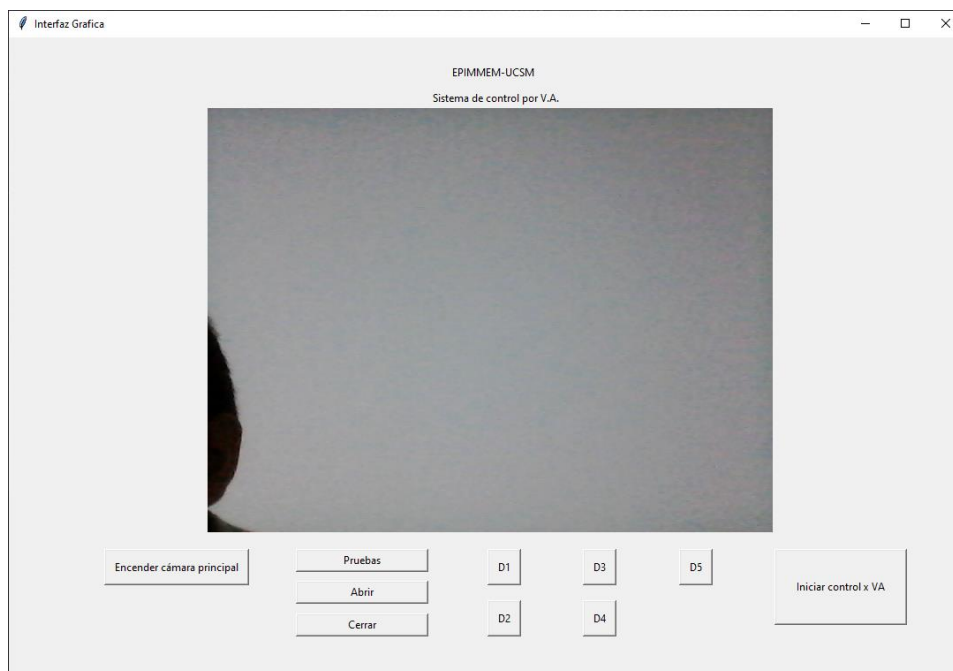
A continuación, se mostrará gráficamente como se ve en la interfaz los diferentes movimientos de los dedos en la mano humana que fueron replicadas y mostradas en la etapa 3.

**Figura 51**  
*Interfaz de usuario*



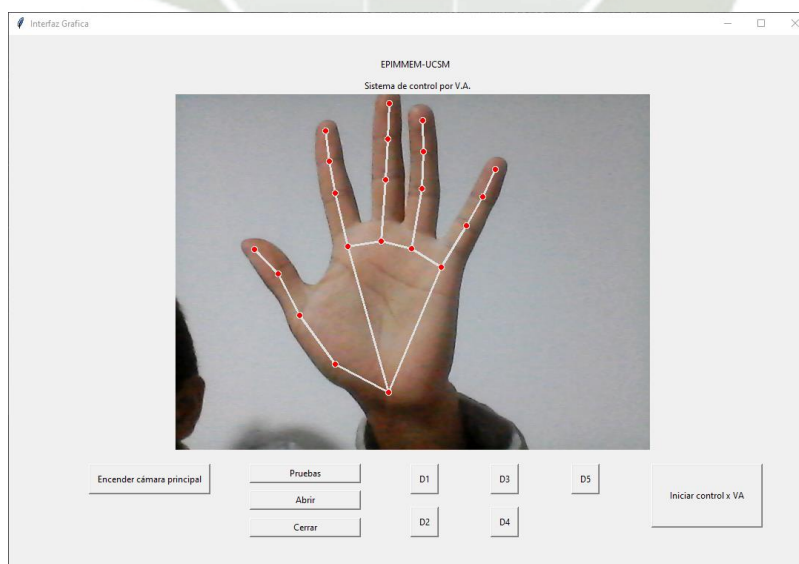
*Nota:* Esta imagen muestra la interfaz de usuario desarrollada para el control de la mano robótica. La interfaz presenta una representación gráfica de la mano humana, con cada dedo individualizado. A medida que el usuario realiza movimientos con sus dedos, la interfaz refleja estos movimientos en la mano virtual. Esto permite al usuario visualizar de forma intuitiva cómo se están traduciendo sus gestos en el control de la mano robótica.

**Figura 52**  
*Interfaz de usuario en funcionamiento iniciado el control por VA*



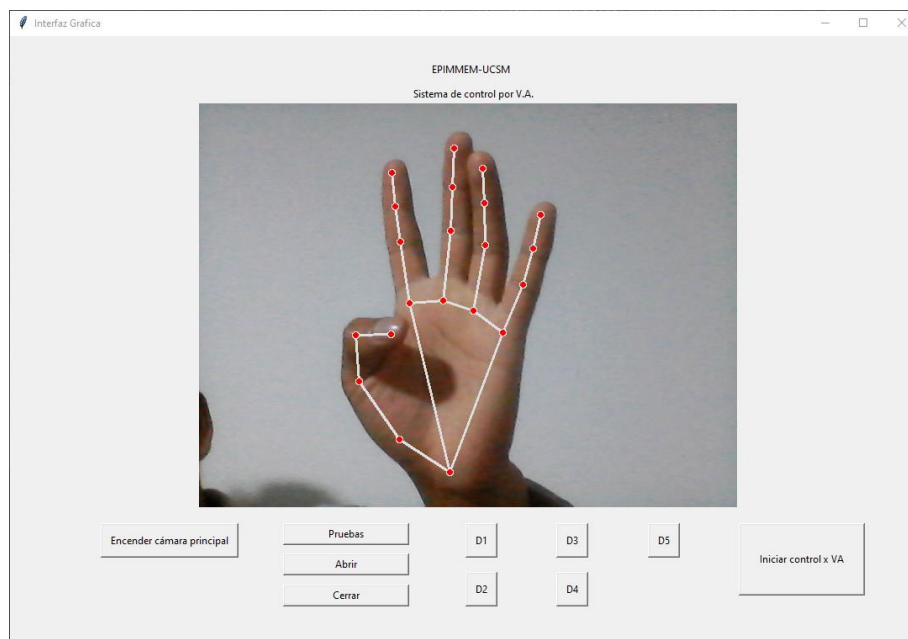
*Nota:* Esta imagen muestra la interfaz de usuario en funcionamiento con el control por voz activado. La interfaz presenta una serie de comandos de voz que se pueden utilizar para controlar la mano robótica. El usuario puede pronunciar estos comandos para realizar diferentes acciones, como abrir y cerrar la mano, mover los dedos o realizar gestos específicos.

**Figura 53**  
*Todos los dedos abiertos en la interfaz*



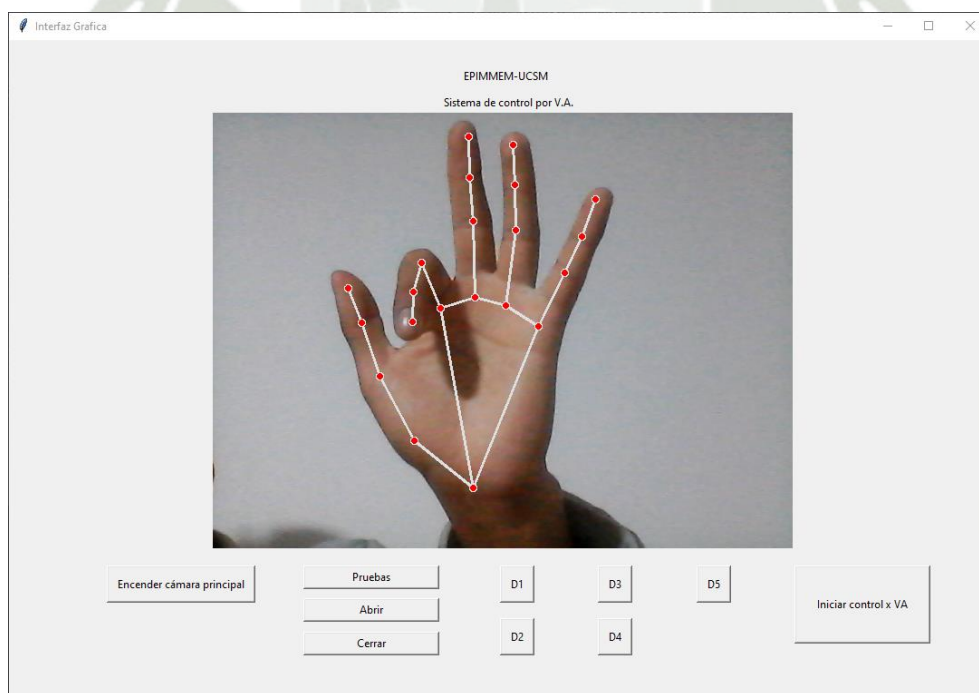
*Nota:* Esta imagen muestra la interfaz de usuario del sistema de control de la mano robótica. En la interfaz se observa una imagen de una mano con todos los dedos abiertos

**Figura 54**  
*Dedo pulgar cerrado en la interfaz*



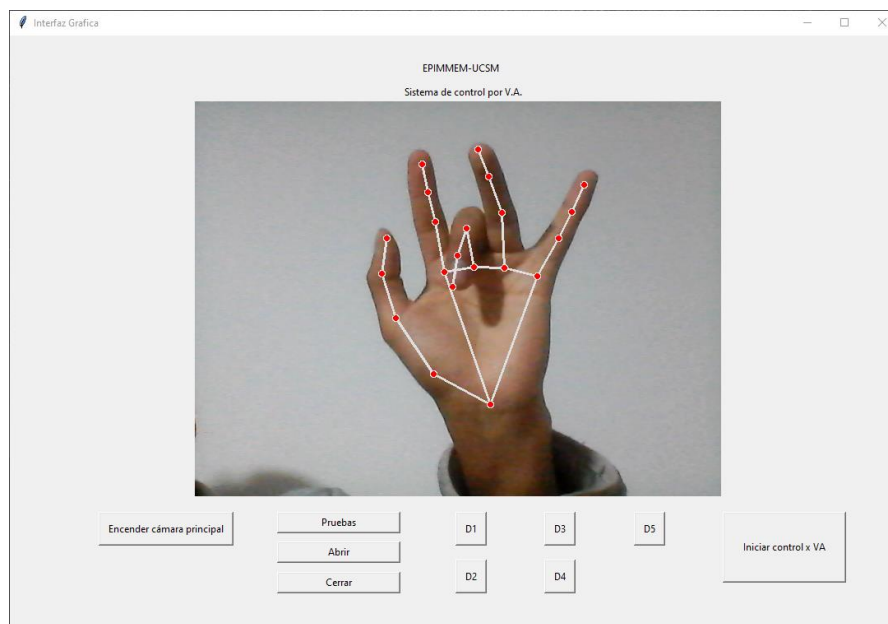
*Nota:* Esta imagen muestra la interfaz de usuario del sistema de control de la mano robótica. En la interfaz se observa una imagen de una mano con el dedo pulgar cerrado

**Figura 55**  
*Dedo pulgar cerrado en la interfaz*



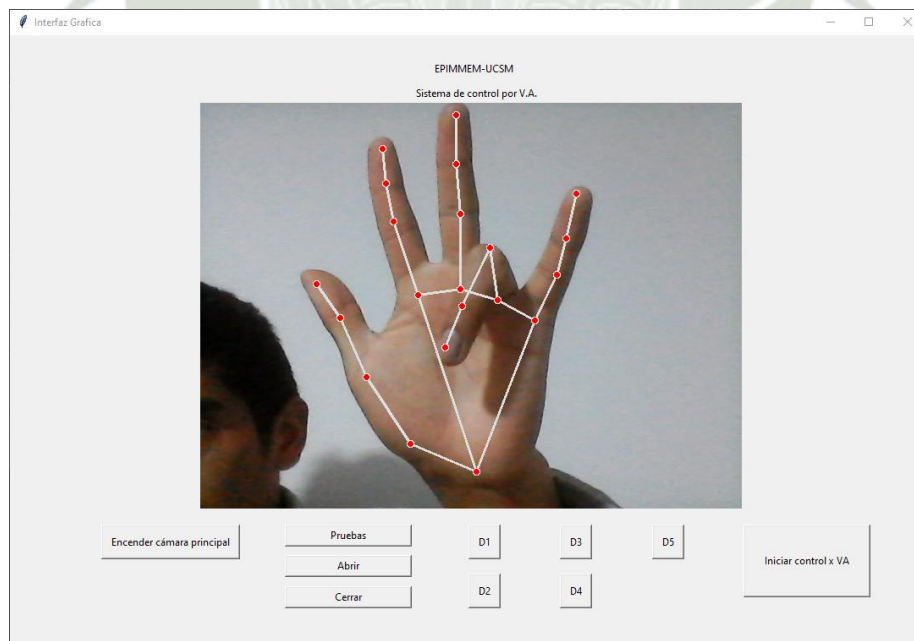
*Notan:* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual con el dedo pulgar cerrado y los demás dedos extendidos.

**Figura 56**  
*Dedo medio cerrado en la interfaz*



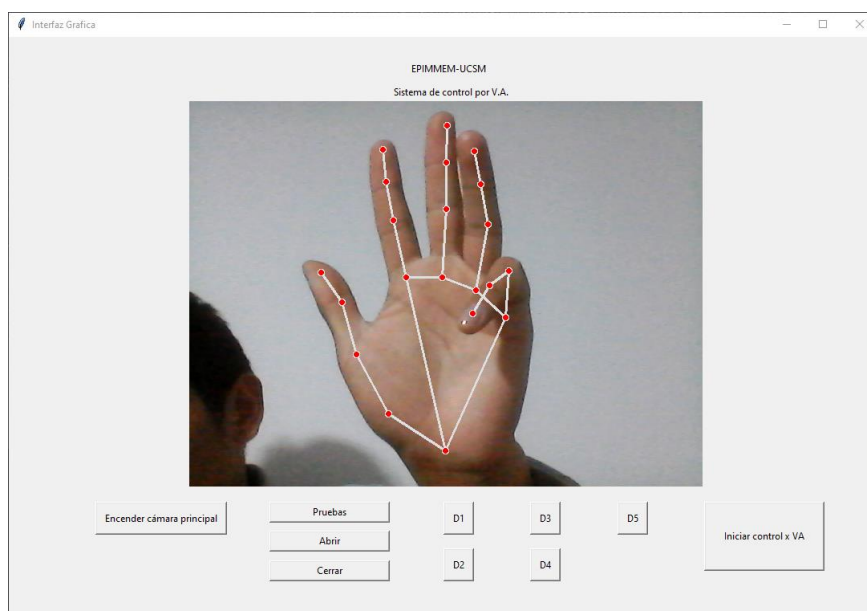
*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual con el dedo medio cerrado y los demás dedos extendidos.

**Figura 57**  
*Dedo anular cerrado en la interfaz*



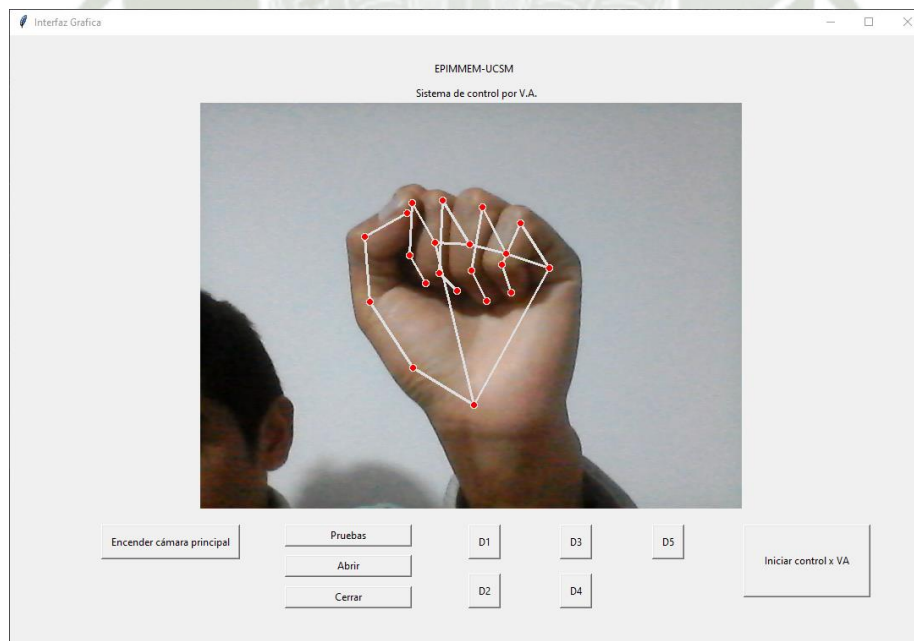
*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual con el dedo anular cerrado y los demás dedos extendidos.

**Figura 58**  
*Dedo pulgar cerrado en la interfaz*



*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual con el dedo pulgar cerrado y los demás dedos extendidos.

**Figura 59**  
*Todos los dedos cerrados en la interfaz*



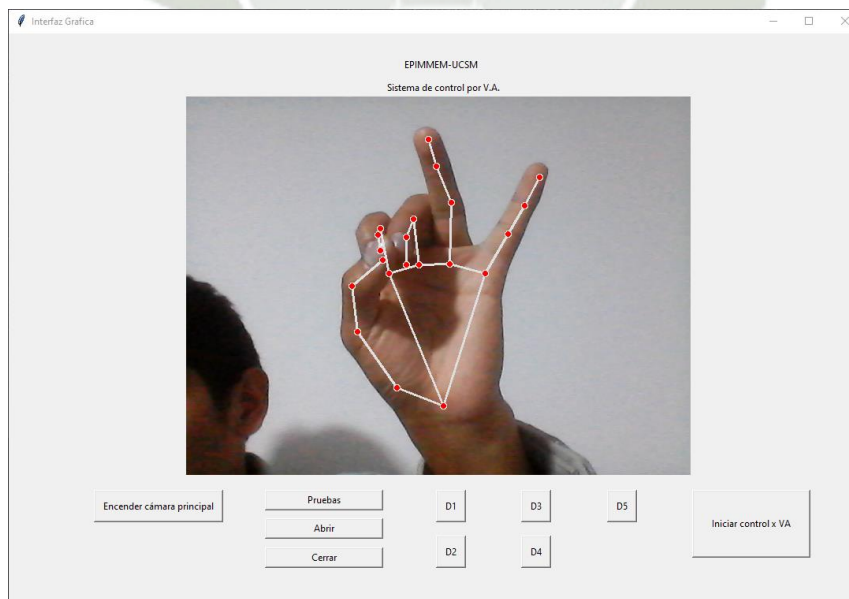
*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual con todos los dedos cerrados en forma de puño

**Figura 60**  
*Gestos con los dedos (agarre tipo pinza) en la interfaz*



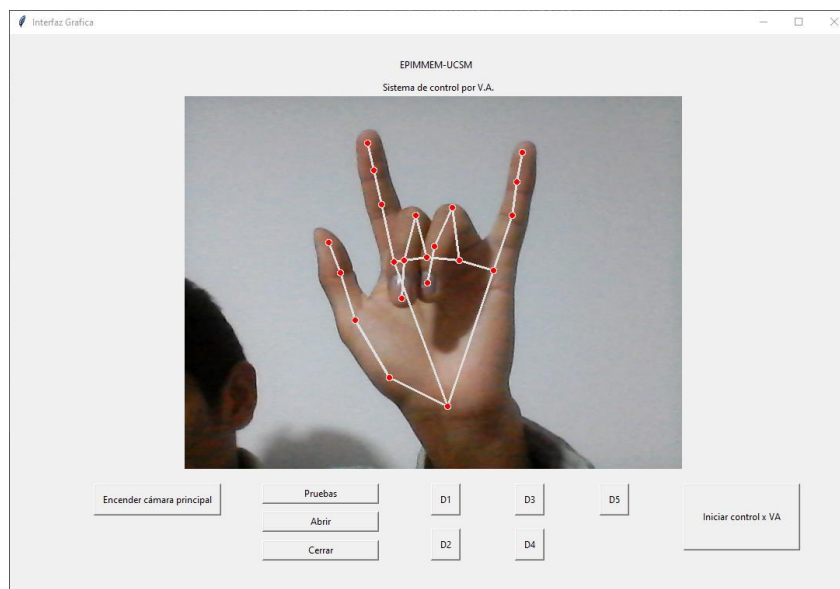
*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual realizando un gesto con los dedos pulgar e índice en forma de pinza.

**Figura 61**  
*Gestos con los dedos (agarre tipo pinza) en la interfaz*



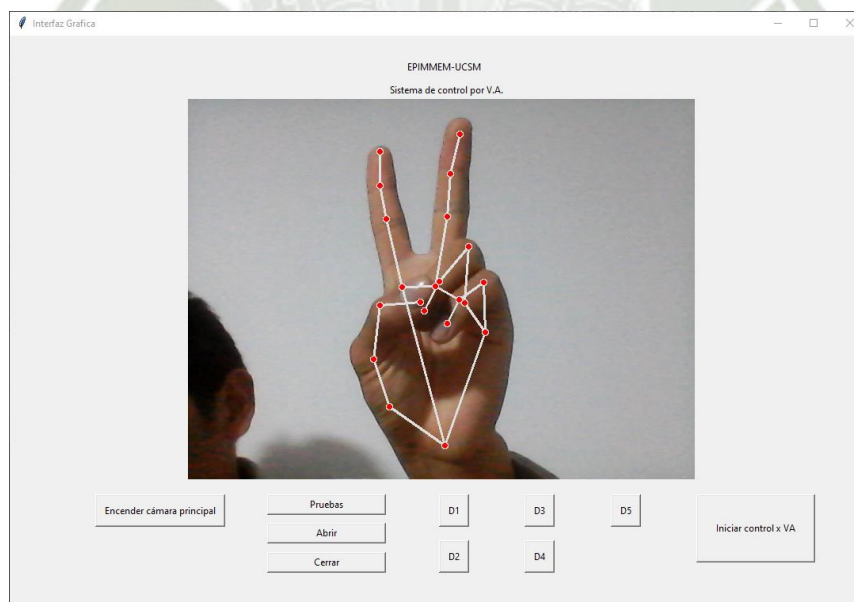
*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual realizando un gesto con los dedos pulgar e índice en forma de pinza.

**Figura 62**  
*Gestos con los dedos en la interfaz*



*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual realizando un gesto con los dedos índice y medio extendidos, mientras que el dedo pulgar está flexionado y tocando la punta del dedo índice.

**Figura 63**  
*Gestos con los dedos en la interfaz*

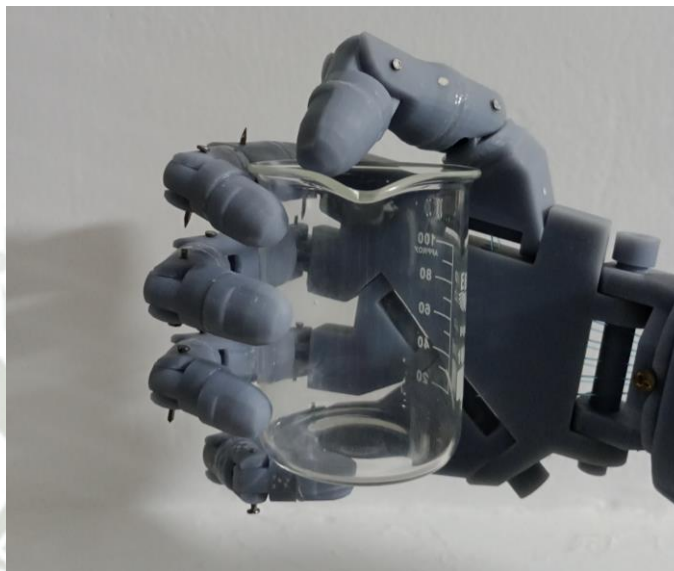


*Nota.* La imagen muestra la interfaz de usuario del sistema de control de la mano robótica. Se observa una mano virtual realizando un gesto con los dedos índice y medio extendidos en forma de V, mientras que el dedo pulgar está flexionado y tocando la punta del dedo índice

#### 4.5. Pruebas con recipientes de laboratorio

##### **Figura 64**

*Mano robótica sujetando un vaso precipitado*



*Nota:* La figura muestra la capacidad de agarre de la mano robótica, demostrando su funcionalidad al sostener un vaso de precipitado de vidrio comúnmente utilizado en laboratorios.

##### **Figura 65**

*Mano robótica sujetando un vaso precipitado con líquido*



*Nota:* Esta imagen ilustra la destreza y la precisión de la mano robótica, capaz de sostener un vaso de precipitados con líquido sin derramarlo, evidenciando la fina coordinación entre los componentes mecánicos y el sistema de control.

**Figura 66**

*Mano robótica sujetando una probeta*



*Nota:* La figura muestra la capacidad de agarre de la mano robótica sosteniendo una probeta comúnmente usada en laboratorios.

**Figura 67**

*Mano robótica sujetando una probeta con líquido*



*Nota:* Esta imagen ilustra el agarre de la mano robótica de la probeta con líquido dentro de ella.

## CONCLUSIONES

- **PRIMERA:**  
Se logró desarrollar un sistema que puede controlar los movimientos de una mano robótica. Este sistema representa un avance significativo en la interacción humano – robot.
- **SEGUNDA:**  
Se creó exitosamente un sistema de visión artificial capaz de realizar una imitación de los movimientos de una mano humana real mediante el uso de la inteligencia artificial de la biblioteca de código abierto MediaPipe de Google. El sistema fue capaz de identificar y seguir con precisión los 21 puntos clave o Hand LandMarks.
- **TERCERA:**  
Se realizó una comparación de prototipos que permitió realizar una selección informada y garantizar la elección del prototipo óptimo de la mano robótica InMoov. Durante la evaluación de los diferentes 3 prototipos se tomaron en consideración diversos criterios, como la movilidad de los dedos, la facilidad de ensamblaje, la compatibilidad con componentes electrónicos estándar y la disponibilidad de documentación y soporte.
- **CUARTA:**  
Se logró construir la mano robótica con tecnología de impresión en 3D con material termoplástico por filamento y resina, lográndose además una reducción de tiempo de impresión del 50% al realizarlo de 2 materiales. El ensamblaje y la puesta en marcha de la mano robótica seleccionada, permitió realizar pruebas de apertura y cierre independientes de cada dedo.
- **QUINTA:**  
El uso del software PROTEUS permitió diseñar y simular el circuito electrónico y por lo tanto contribuyó a reducir los tiempos y costos asociados en aproximadamente un 30% al evitar hacer con prueba-error el armado del circuito, al tiempo que garantizo la calidad y fiabilidad de este.

## RECOMENDACIONES

- PRIMERA:

En futuras investigaciones ampliar la funcionalidad de la mano robótica para que pueda realizar una mayor variedad de movimientos y tareas. Esto implicaría la incorporación de sensores de presión, proximidad y/o inercia, o también el desarrollo de algoritmos más complejos.

- SEGUNDA:

Realizar pruebas y validaciones adicionales en diferentes escenarios y entornos para evaluar la robustez y versatilidad de la mano robótica en condiciones diversas.

- TERCERA:

Investigar y evaluar para futuras investigaciones diferentes materiales para la construcción de la mano robótica con el objetivo de mejorar su resistencia, durabilidad y peso.

- CUARTA:

Mantener la documentación detallada del desarrollo y pruebas de la mano robótica para facilitar el mantenimiento futuro y la replicación del proyecto por parte de otros investigadores.

## REFERENCIA

- Ana, P. (s.f.). Sistema de movimientos de imagen para manipulacion de alto riesgo en almacen industrial usando vision artificial.
- Beatriz, B. P. (2022). Introduccion a la vision artificial: Porcesos y Aplicaciones.
- Cruz, C. (5 de julio de 2018). El reconocimiento facial llega al aeropuerto internacional de Sidney, Australia. *C/NET en español*. Obtenido de <https://www.cnet.com/es/noticias/qantas-reconocimiento-facial-sidney-privacidad/>
- Eduardo, Z. C. (2019). CONTROL POR OPTOMIOGRAFIA (OMG) Y CONSTRUCCION DE UNA MANO ROBOTICA CON TECNOLOGIA DE IMPRESION 3D.
- Freepik. (2020). Obtenido de [https://www.freepik.es/fotos-premium/hombre-oficina-tarjeta-identificacion-escanear-maquina-sistema-control-acceso\\_9449162.htm](https://www.freepik.es/fotos-premium/hombre-oficina-tarjeta-identificacion-escanear-maquina-sistema-control-acceso_9449162.htm)
- Fuentes Collazos, A. (2020). DISEÑO Y PROTOTIPADO DE UNA PRÓTESIS DE ACCIONAMIENTO MECÁNICO PARA AMPUTACIONES TRANSFALÁNGICAS DE LA MANO. Lima, Perú.
- Gabriela, V. M. (2017). Procesamiento de imagenes usando OpenCV aplicado en raspberry pi para la clasificacion del cacao.
- Giancarlo, S. Y. (2020). DISEÑO E IMPLEMENTACION DE UN SISTEMA DE CONTROL PARA UNA PROTESIS MIOELECTRICA DE MIEMBRO SUPERIOR CON CONTROL DE FUERZA Y POSICION. PUCP.
- Gonzales, R. C., & Woods, R. E. (2002). Digital Image Processing.
- Gray, H. (2020). *Gray's Anatomy for Students*.
- Gustavo Alexander, B. R. (2008). Adaptacion de infomracion en sistemas conscientes del contexto.
- Irma, G. G., Santos Santos, E., Cano Diaz, S., & Crespo y Mena, J. (2014). Guia de clasificacion de riesgo, peligrosidad y primeros auxilios de sustancias quimicas de uso en los laboratoria de nivel medio superior.

- Ivan, P. (s.f.). Desarrollo de una mano robotica con capacidad de manipulacion.
- J.A. Oteo, P. B. (2015). Valores normativos de la fuerza de puño en la población española en edad laboral. *ScienceDirect*.
- Jacobo, G. A. (2007). Filtrado adaptativo local en el dominio de las transformadas deslizantes.
- JAVier, C. B. (2016). Analisis de la distribucion de presiones en la mano humana durante el agarre.
- Jhonny, O. (s.f.). Manejo de sustancias toxicas por medio de una mano inalambrica.
- Jorge, D. A. (2021). VISION POR COMPUTADOR PARA EL USO DE REALIDAD AUMENTADA EN UNITY3D.
- Julinhio, R. V. (2021). DISEÑO Y CONTROL DE UNA MANO ROBOTICA PARA FUNCION MOTORA DE PERSONAS DISCAPACITADAS.
- Kapandji, A. I. (2012). *Fisiología Articular*.
- Khanal, A. (2021). HAND POSE ESTIMATION AND GESTURE DETECTION FROM WEBCAM IMAGES.
- Natalia, Q. P. (s.f.). Sistema de control de mano robotica replicadora de moviminetos para manipular sustancias quimicas. *Acta Nova*.
- Nuclear, C. d. (2014). Revista de seguridad nuclear y proteccion radiologica. 76.
- Vanessa, A. P. (2017). Internet de las cosas y vision artificial, funcionamiento y aplicaciones.
- Victor, V. B. (2010). Sistema de vision artificial para el control de calidad en piezas cromadas.

## ANEXOS

### ANEXO 1 Especificaciones Técnicas Servomotor MG995

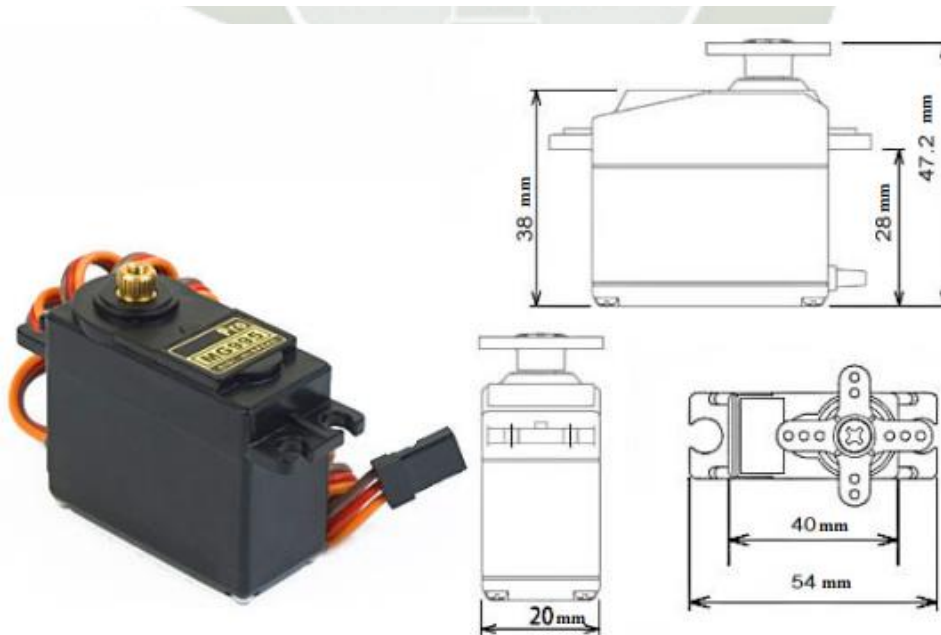
## 31150-MP MG995 High Speed Servo Actuator

The unit comes complete with color coded 30cm wire leads with a 3 X 1 pin 0.1" Pitch type female header connector that matches most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed servo actuator is not code dependant; You can use any servo code, hardware or library to control them. The MG995 Actuator includes arms and hardware to get started.

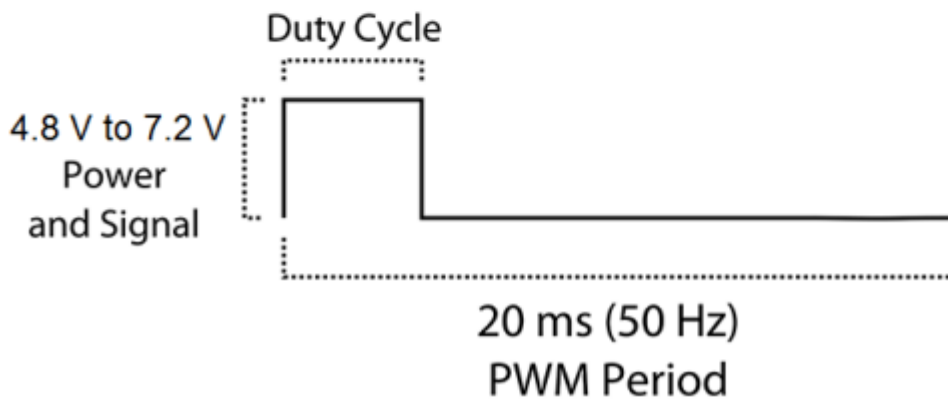
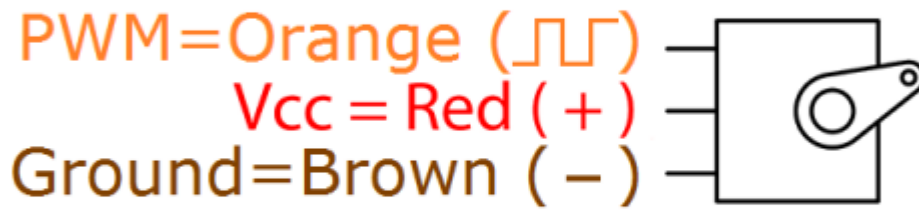
### Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V ), 10 kgf·cm (6 V)
- Rotation Angle: 120deg. (+- 60 from center)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Metal Gears for longer life
- Temperature range: 0 °C – 55 °C



# 31150-MP

## MG995 High Speed Servo Actuator



Information obtained from or supplied by mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.



**MARLIN P. JONES & ASSOC., INC.**  
P.O. Box 530400 Lake Park, FL 33403  
800-652-6733 FAX 561-844-8764  
WWW.MPJA.COM

## Pololu 5V, 5A Step-Down Voltage Regulator D24V50F5



Pololu item #: 2851 **201** in stock  
Brand: **Pololu** [supply outlook](#)  
Status: Rationed (Active)   
**✓ RoHS3**

Price break	Unit price (US\$)
1	32.95
5	30.31

Quantity:

[Add to cart](#)

[backorders](#) allowed

[Add to wish list](#)



This small synchronous switching step-down (or buck) regulator takes an input voltage of up to 38 V and efficiently reduces it to **5 V**. The board measures only 0.7" × 0.8", but it allows a typical continuous output current of up to **5 A**. Typical efficiencies of 85% to 95% make this regulator well suited for high-power applications like powering motors or servos. High efficiencies are maintained at light loads by dynamically changing the switching frequency, and an optional shutdown pin enables a low-power state with a current draw of a few hundred microamps.

[Description](#) [Specs \(10\)](#) [Pictures \(10\)](#) [Resources \(3\)](#) [FAQs \(0\)](#) [On the blog \(3\)](#)

### Dimensions

Size: 0.7" × 0.8" × 0.35"<sup>1</sup>

Weight: 3.0 g<sup>1</sup>

### General specifications

Minimum operating voltage:	6 V <sup>2</sup>
Maximum operating voltage:	38 V
Continuous output current:	5 A <sup>3</sup>
Output voltage:	5 V
Reverse voltage protection?:	Y
Maximum quiescent current:	0.8 mA <sup>4</sup>

## Identifying markings

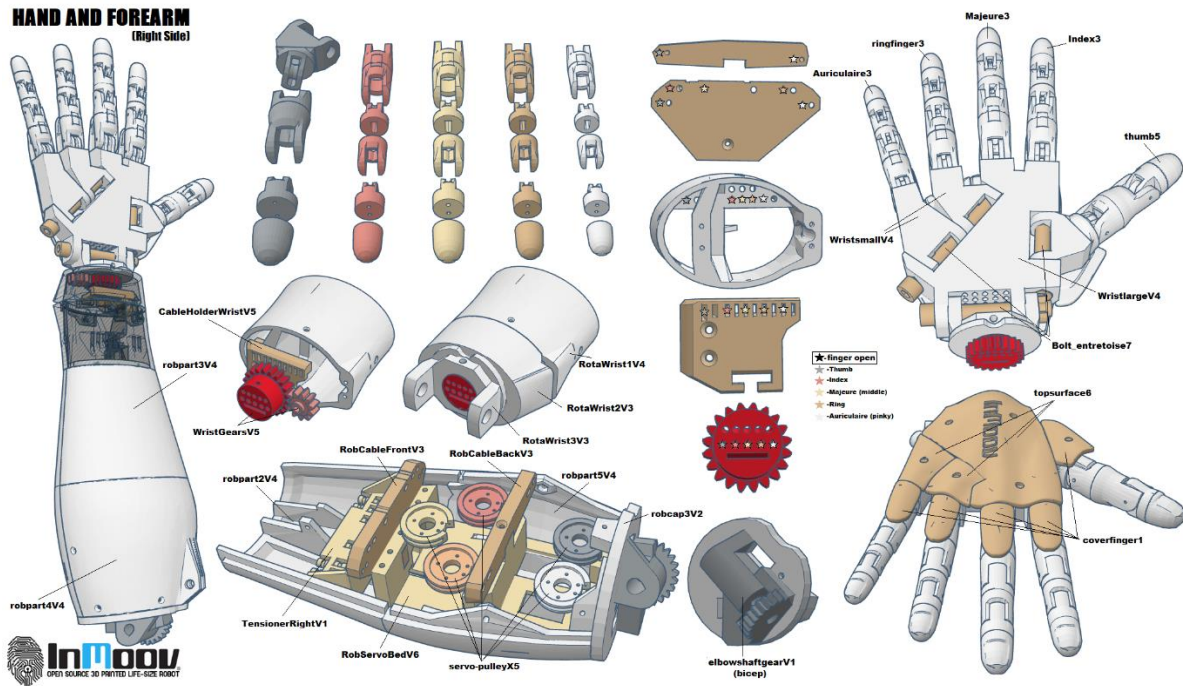
<b>PCB dev codes:</b>	reg15d
<b>Other PCB markings:</b>	0J8511

## Notes:

- 1 Without included optional headers.
- 2 For small loads; this voltage rises approximately linearly up to around 6.2 V at 5 A output.
- 3 Typical. Actual continuous output current limited by thermal dissipation.
- 4 Typical worst case (i.e. with  $V_{IN}$  close to 6V). The ENABLE pin can be used to reduce the quiescent current to a few hundred microamps.



# ANEXO 3 Plano de Explosión de mano y brazo InMoov



## ANEXO 4 Código de programación en Python

```

import tkinter as tk
from PIL import Image, ImageTk
import cv2
import time
import os
import HandTrackingModule as htm
import serial

class HandDetectionApp:
    def __init__(self, root, video_source=0):
        self.root = root
        self.root.title("Interfaz Grafica")

        self.cap = cv2.VideoCapture(video_source)
        self.canvas = tk.Canvas(root, width=1080, height=720)
        self.canvas.pack()

        self.is_detecting = False
        self.detector = htm.handDetector(detectionCon=0.75)
        self.tip_ids = [4, 8, 12, 16, 20]

        # -----
        self.COM = 'COM4'
        self.BAUD = 9600
        self.ser = serial.Serial(self.COM, self.BAUD)
        time.sleep(2)
        # -----

    def start_detection(self, send_serial_data=True):
        self.is_detecting = True
        self.run_detection(send_serial_data)

    def stop_detection(self):
        self.is_detecting = False

    def run_detection(self, send_serial_data=True):
        while self.is_detecting:
            success, img = self.cap.read()
            img = cv2.flip(img, 1)
            lm_list = self.detector.findHands(img)
            if len(lm_list) != 0:
                fingers = []

                # Pulgar
                if lm_list[self.tip_ids[0]][1] < lm_list[self.tip_ids[0] -
1][1]:
                    fingers.append(1)
                else:
                    fingers.append(0)

                # 4 dedos
                for id in range(1, 5):
                    if lm_list[self.tip_ids[id]][2] < lm_list[self.tip_ids[id]

```

```

-                                                                 2][2]:
        fingers.append(1)
    else:
        fingers.append(0)

    if
        pulgar = send_serial_data:
        indice = str(fingers[0])
        medio = str(fingers[1])
        anular = str(fingers[2])
        menique = str(fingers[3])
        menique = str(fingers[-1])
        cadena = "$" + pulgar + indice + medio + anular + menique
        self.ser.write(cadena.encode('ascii'))

    self.show_frame(img)

    if cv2.waitKey(1) & 0xFF == 27:
        break

    self.cap.release()
    cv2.destroyAllWindows()

    def show_frame(self, frame):
        photo = ImageTk.PhotoImage(image=Image.fromarray(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB)))
        self.canvas.create_image(225, 80, image=photo, anchor=tk.NW)
        self.root.update_idletasks()
        self.root.update()

    def abrir_mano(self):
        self.ser.write(b"$11111")

    def cerrar_mano(self):
        self.ser.write(b"$00000")

    def abrir_D1(self):
        self.ser.write(b"$10000")

    def abrir_D2(self):
        self.ser.write(b"$01000")

    def abrir_D3(self):
        self.ser.write(b"$00100")

    def abrir_D4(self):
        self.ser.write(b"$00010")

    def abrir_D5(self):
        self.ser.write(b"$00001")

if __name__ == "__main__":
    root = tk.Tk()
    app = HandDetectionApp(root)

    etiqueta = tk.Label(root, text="EPIMMEM-UCSM")
    etiquetaTitulo = tk.Label(root, text="Sistema de control por V.A.")
    botonEncenderCamPrin = tk.Button(root, text="Encender cámara principal",

```

```

width=22,                                                                 height=2,
                                command=lambda:
app.start_detection(send_serial_data=False))
    botonIniciarControlVA = tk.Button(root, text="Iniciar control x VA",
width=20,                        height=5,                        command=app.start_detection)
    botonPruebas = tk.Button(root, text="Pruebas", width=20, height=1,
command=app.stop_detection)
    botonAbrir = tk.Button(root, text="Abrir", width=20, height=1,
command=app.abrir_mano)
    botonCerrar = tk.Button(root, text="Cerrar", width=20, height=1,
command=app.cerrar_mano)
    botonDedo1 = tk.Button(root, text="D1", width=4, height=2,
command=app.abrir_D1)
    botonDedo3 = tk.Button(root, text="D3", width=4, height=2,
command=app.abrir_D3)
    botonDedo2 = tk.Button(root, text="D2", width=4, height=2,
command=app.abrir_D2)
    botonDedo4 = tk.Button(root, text="D4", width=4, height=2,
command=app.abrir_D4)
    botonDedo5 = tk.Button(root, text="D5", width=4, height=2,
command=app.abrir_D5)

    etiqueta.place(relx=0.46, relx=0.46, rely=0.04)
    etiquetaTitulo.place(relx=0.44, relx=0.44, rely=0.08)
    botonEncenderCamPrin.place(relx=0.1, relx=0.1, rely=0.8)
    botonIniciarControlVA.place(relx=0.8, relx=0.8, rely=0.8)
    botonPruebas.place(relx=0.3, relx=0.3, rely=0.8)
    botonAbrir.place(relx=0.3, relx=0.3, rely=0.85)
    botonCerrar.place(relx=0.3, relx=0.3, rely=0.9)
    botonDedo1.place(relx=0.5, relx=0.5, rely=0.8)
    botonDedo3.place(relx=0.6, relx=0.6, rely=0.8)
    botonDedo2.place(relx=0.5, relx=0.5, rely=0.88)
    botonDedo4.place(relx=0.6, relx=0.6, rely=0.88)
    botonDedo5.place(relx=0.7, relx=0.7, rely=0.8)

root.mainloop()

```



## ANEXO 5 Código de programación en Arduino

```
#include <Servo.h>

#define numOfValsRec 5
#define digitsPerValRec 1

Servo servoThumb;
Servo servoIndex;
Servo servoMiddle;
Servo servoRing;
Servo servoPinky;

int valsRec[numOfValsRec];
//$00000
int stringLength=numOfValsRec*digitsPerValRec+1; //$00000
int counter=0;
bool counterStart=false;
String receivedString;

void setup() {
  Serial.begin(9600);
  servoThumb.attach(5);
  servoIndex.attach(6);
  servoMiddle.attach(9);
  servoRing.attach(10);
  servoPinky.attach(11);
}
void receieveData() {
  while(Serial.available())
  {
    char c=Serial.read();
```

```

Serial.println(c);

if (c=='$') {
    counterStart=true;
}
if (counterStart) {
    if (counter<stringLength){
        receivedString=String(receivedString+c);
        counter++;
    }
    if (counter>=stringLength){
        //$00000
        for (int i=0; i<numOfValsRec; i++)
        {
            int num=(i*digitsPerValRec)+1;
            valsRec[i]=receivedString.substring(num,num+digitsPerValRec).toInt();
        }
        receivedString="";
        counter=0;
        counterStart=false;
    }
}
}
}

void loop() {
    receiveData();
    if (valsRec[0] == 1){servoThumb.write(0);}
    else {servoThumb.write(180);}
    if (valsRec[1] == 1){servoIndex.write(180);}
    else {servoIndex.write(0);}
}

```

```
if (valsRec[2] == 1){servoMiddle.write(180);}
  else {servoMiddle.write(0);}
if (valsRec[3] == 1){servoRing.write(180);}
  else {servoRing.write(0);}
if (valsRec[4] == 1){servoPinky.write(180);}
  else {servoPinky.write(0);}
}
```

