



UNIÓN DE ASOCIACIONES
DE INGENIEROS TÉCNICOS
INDUSTRIALES Y GRADUADOS
EN LA INGENIERÍA DE LA
RAMA INDUSTRIAL DE ESPAÑA

UNIÓN DE ASOCIACIONES DE INGENIEROS TÉCNICOS INDUSTRIALES Y GRADUADOS EN INGENIERÍA DE LA RAMA INDUSTRIAL DE ESPAÑA (UAITIE)

“CONVOCATORIA 2026”

XI PREMIO NACIONAL DE INICIACIÓN A LA INVESTIGACIÓN TECNOLÓGICA

Título del Trabajo:

PROTOTIPO DE MANO ROBÓTICA CON VISIÓN ARTIFICIAL

AUTOR/ES:
Luis Tinas Sáenz, Gonzalo Vidal López, Guillermo Cabeza Guardia, Aiman Nya
Dkikar

BLOQUE TEMÁTICO:
Robótica

NIVEL EDUCATIVO:
Bachillerato

COORDINADOR:
M. Ángeles Nogales Delgado

MARZO 2026



Resumen

El proyecto P.M.V.A. consiste en el diseño e implementación de un sistema de teleoperación que permite controlar una mano robótica mediante el seguimiento en tiempo real de los movimientos de la mano humana, utilizando técnicas avanzadas de visión artificial y un sistema de control Arduino. Link a [Vídeo-presentación PMVA](#).

El sistema desarrollado tiene las siguientes características principales:

- **Visión artificial:** Se implementa un sistema capaz de detectar y seguir la mano humana en tiempo real mediante el procesamiento de imágenes de una cámara convencional.
- **Extracción de parámetros cinemáticos:** El sistema extrae los parámetros cinemáticos relevantes de la mano humana, como los ángulos de flexión de los dedos, y los traduce en movimientos equivalentes en la mano robótica.
- **Diseño de la mano robótica:** Se diseña con la estructura mecánica adecuada y los grados de libertad necesarios para simular los movimientos de la mano humana.
- **Control embebido:** Se integra un sistema de control basado en Arduino, que coordina la actuación de los servomotores que mueven los dedos de la mano robótica, replicando los movimientos humanos.
- **Evaluación en tiempo real:** El sistema es evaluado mediante pruebas de teleoperación en tiempo real, con la captura y procesamiento de los movimientos de los dedos humanos a través de visión artificial.
- **Interfaz web:** Se desarrolla una interfaz web accesible desde un navegador, que permite la teleoperación de la mano robótica de forma remota (a través de internet) y de forma local.

El sistema PMVA se basa en una arquitectura maestro-esclavo, en la que la mano humana actúa como dispositivo maestro y la mano robótica como sistema esclavo.



Los movimientos son capturados por una cámara conectada a un ordenador, donde se procesan mediante algoritmos de visión artificial implementados con librerías especializadas como [OpenCV](#) y [MediaPipe](#). Estas herramientas permiten detectar puntos clave (landmarks) de la mano y calcular los ángulos de flexión de los dedos en tiempo real.

Los ángulos obtenidos se transmiten al microcontrolador Arduino mediante comunicación serie, que genera las señales PWM necesarias para controlar los servomotores de la mano robótica, replicando los movimientos de la mano humana de manera sincronizada.

Palabras Clave

robótica, visión artificial, control de sistemas, teleoperación



Índice

1. Introducción.....	5
1.1 Organización y roles del equipo.....	5
1.2 Objetivos.....	7
2. Metodología.....	8
3. Descripción del proyecto.....	9
3.1 Subsistemas.....	9
3.1.1 Adquisición y procesamiento de imágenes.....	10
3.1.2 Comunicación y control embebido (Arduino).....	11
3.1.3 Sistema electromecánico de actuación.....	12
3.2 Integración de hardware y software.....	12
3.2.1 Estructura y mecánica.....	12
3.2.2 Electrónica.....	13
3.2.3 Programación.....	15
4. Resultados.....	19
5. Conclusión.....	20
6. Referencias.....	20

MANO ROBÓTICA TELEOPERADA MEDIANTE VISIÓN ARTIFICIAL

1. Introducción

PMVA es una mano robótica que se controla de forma remota mediante visión artificial, siguiendo los movimientos de la mano humana. La estructura de la mano se ha fabricado mediante impresión 3D y su funcionamiento se gestiona mediante una placa Arduino, integrando así mecánica, electrónica y programación en un mismo sistema.

1.1 Organización y roles del equipo

PMVA (Prototipo de Mano con Visión Artificial) es un proyecto desarrollado por cuatro estudiantes de segundo de bachillerato científico-tecnológico del Instituto de Enseñanza Secundaria Pablo Ruiz Picasso en Chiclana de la Frontera. Surgió cuando nuestra profesora de Tecnología nos propuso crear un proyecto de libre elección en el que pudiéramos aplicar los conocimientos adquiridos en cursos anteriores, al mismo tiempo que nos planteara un reto personal y técnico.

Así que nos pusimos manos a la obra y dejamos volar nuestra imaginación. La decisión no era fácil: queríamos desarrollar una tecnología que no sólo resultara atractiva desde el punto de vista técnico, sino que también tuviera aplicaciones en la vida real. Además, queríamos trabajar con tecnologías actuales como la robótica, la inteligencia artificial y la visión artificial, que ya forman parte de nuestro entorno y despiertan nuestra curiosidad tanto como usuarios como futuros desarrolladores.

Después de valorar varias opciones, nos decidimos por desarrollar una mano robótica, un proyecto que nos permite integrar distintas disciplinas como Tecnología e Ingeniería (mecánica, diseño 3D, electrónica y programación), matemáticas y física. De este modo satisface nuestro interés por profundizar en el ámbito de la ciencia y la tecnología.

Las manos robóticas, representan un área de especial interés debido a su capacidad para interactuar con el entorno de forma versátil y precisa, imitando en cierta medida la funcionalidad de la mano humana. Tienen aplicaciones muy diversas,

desde la robótica industrial hasta la asistencia a personas con discapacidad, la teleoperación y la investigación en ingeniería.

Como se ha indicado, elegimos desarrollar una mano robótica que se controla de forma remota mediante visión artificial, siguiendo los movimientos de la mano humana. Decidimos trabajar en este proyecto porque nos parecía muy interesante explorar cómo la tecnología puede imitar y reaccionar a nuestros propios movimientos de forma directa. La idea de controlar una mano robótica con nuestra propia mano nos resultaba intuitiva y divertida, pero al mismo tiempo un reto técnico que nos permitiría aprender cosas nuevas: desde cómo se mueven los motores, hasta cómo las cámaras y la programación pueden “entender” lo que estamos haciendo.

Además, queríamos trabajar con herramientas accesibles y de bajo coste, como Arduino, impresión 3D y cámaras normales, para demostrar que no hace falta un laboratorio profesional para iniciar un proyecto de investigación tecnológica. La idea era crear algo que funcionara de verdad, que se pueda ver, tocar y probar, y que, además, nos enseñara mucho sobre cómo investigar y resolver problemas tecnológicos reales.

Respecto a la organización del proyecto, hemos asignado las tareas del mismo en función de nuestros conocimientos previos y nuestras preferencias personales, pero sin descuidar la importancia de la coordinación y asegurándonos de que todos los miembros del equipo puedan colaborar en cualquier sistema del proyecto si fuera necesario.

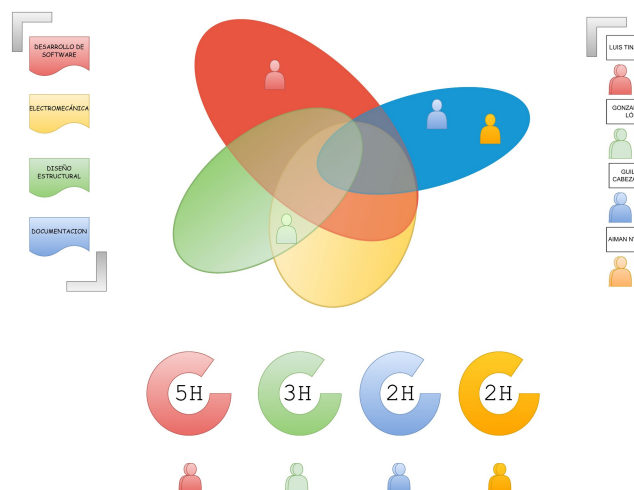


Figura 1: Diagrama de roles

1.2 Objetivos

El objetivo general del proyecto es diseñar e implementar un sistema de teleoperación que permita controlar una mano robótica mediante el seguimiento en tiempo real de la mano humana, utilizando técnicas de visión artificial y una plataforma de control basada en Arduino.

Para facilitar el desarrollo del proyecto, se han definido los siguientes objetivos específicos:

- Desarrollar un sistema de visión artificial capaz de detectar y seguir la mano humana en tiempo real.
- Extraer los parámetros cinemáticos relevantes de la mano humana y traducirlos en movimientos equivalentes de la mano robótica.
- Diseñar e implementar la mano robótica, definiendo su estructura mecánica y los grados de libertad necesarios para la manipulación.
- Integrar un sistema de control embebido basado en Arduino que permita la actuación coordinada de los motores.
- Evaluar el funcionamiento del sistema mediante pruebas de teleoperación en tiempo real, utilizando la adquisición y procesamiento del movimiento de los dedos mediante visión artificial.
- Desarrollar una interfaz web accesible mediante navegador que permita la teleoperación de la mano robótica, garantizando su funcionamiento tanto en modalidad remota (internet) como en modalidad local.

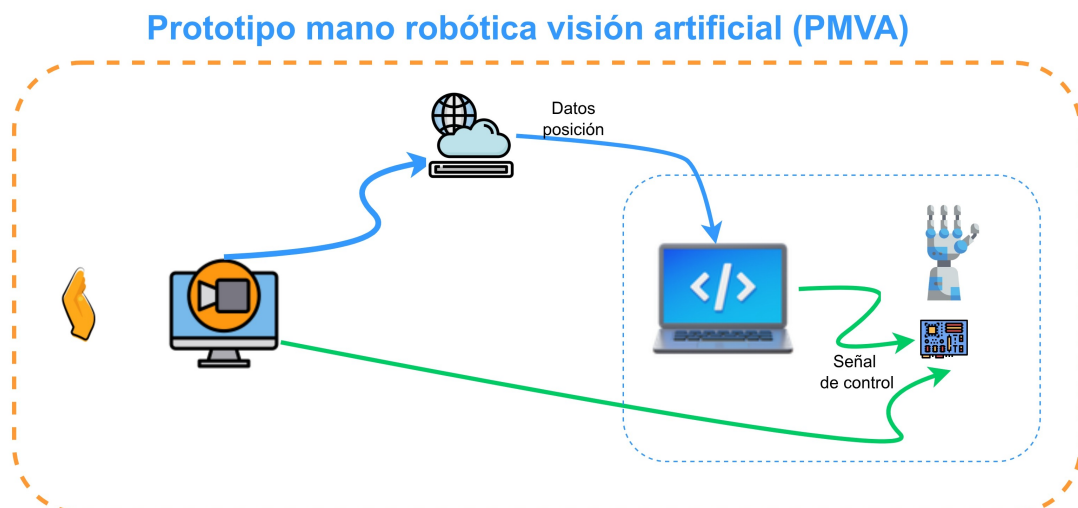


Figura 2: Descripción general

2. Metodología

El desarrollo del proyecto se ha llevado a cabo siguiendo una metodología en cascada, caracterizada por ser un proceso lineal en el que las distintas fases se ejecutan de forma secuencial y progresiva. Dichas fases se representan en el diagrama siguiente.



Figura 3: Metodología. Fases.

3. Descripción del proyecto

Este proyecto consiste en el diseño e implementación de un sistema de teleoperación que permita controlar una mano robótica mediante el seguimiento en tiempo real.

El sistema PMVA se fundamenta en una arquitectura maestro–esclavo, donde la mano humana actúa como dispositivo maestro y la mano robótica como sistema esclavo. La captura de los movimientos se realiza mediante una cámara convencional conectada a un ordenador, donde se procesan las imágenes utilizando algoritmos de visión artificial implementados con librerías especializadas como OpenCV y MediaPipe. Estas herramientas permiten detectar y rastrear puntos clave (landmarks) de la mano en tiempo real, a partir de los cuales se calculan los ángulos de flexión de cada dedo.

Los ángulos obtenidos son transmitidos mediante comunicación serie al microcontrolador Arduino, encargado de generar las señales PWM necesarias para el accionamiento de los servomotores que conforman la mano robótica. De este modo, los movimientos de la mano humana son replicados de manera sincronizada por el sistema robótico.

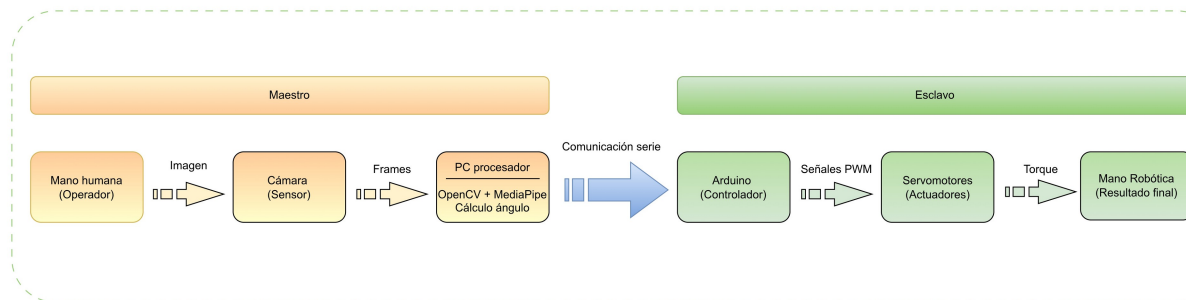


Figura 4: Arquitectura del sistema

3.1 Subsistemas

Para facilitar la comprensión del proyecto distinguiremos tres subsistemas principales:

- Adquisición y procesamiento de imagen
- Comunicación y control embebido
- Sistema electromecánico de actuación.

La correcta interacción entre estos subsistemas permite lograr un control en tiempo real con latencias reducidas y un comportamiento estable.

3.1.1 Adquisición y procesamiento de imágenes

Esta fase se enfoca en la creación del sistema de visión artificial para el seguimiento de la mano humana, que es la base para el control de la mano robótica. Incluye:

- Captura de imagen:

Se emplea una cámara digital convencional para la adquisición de video en tiempo real (cámara integrada del portátil) con una tasa estándar de 30 FPS. Suficientes para garantizar la fluidez en el seguimiento.

- Detección y seguimiento de la mano:

El procesamiento de imágenes se implementa en un ordenador utilizando Python como lenguaje de programación, apoyado en las librerías OpenCV y MediaPipe. Este sistema identifica la mano humana, segmenta la imagen y extrae características como la posición y orientación de los dedos.

Para realizarlo, se utiliza un modelo de detección de puntos clave (landmarks) proporcionado por MediaPipe, el cual permite identificar 21 puntos anatómicos de la mano. A partir de las coordenadas espaciales obtenidas, se procede al cálculo de los ángulos articulares de cada dedo mediante relaciones geométricas entre vectores definidos por los puntos detectados.

En cuanto a la validación del seguimiento, a medida que se desarrollan los algoritmos, se realizan pruebas continuas para garantizar la precisión y fiabilidad del sistema de seguimiento. Se validan diferentes condiciones de iluminación, ángulos y velocidades de movimiento para asegurar que el sistema funcione en un entorno realista.

Finalmente, con el fin de reducir el ruido y las fluctuaciones propias del procesamiento visual, se implementan técnicas de suavizado. Esto permite mejorar la estabilidad del sistema y evitar movimientos abruptos en la mano robótica.

3.1.2 Comunicación y control embebido (Arduino)

En esta fase se desarrolla la comunicación y el control de la mano robótica mediante Arduino. Incluye:

- **Diseño del Protocolo de Comunicación**

Para la transmisión de datos entre el ordenador y el sistema embebido se emplea comunicación serie a través de un puerto USB, configurado a una velocidad de 115200 bps.

Se define un protocolo de transmisión basado en tramas estructuradas que contienen la identificación de los dedos y los valores angulares correspondientes a cada dedo. La estructura general del mensaje es $\langle \text{cos}_i, id_i \rangle$ donde id representa la identificación del dedo y cos representa el ángulo de flexión del dedo en grados. Se envía cada vez que se detecta un cambio de posición en los dedos.

No se establecen delimitadores de inicio y fin de trama porque, comprobado su funcionamiento, no se detectan errores de interpretación en el microcontrolador.

- **Recepción y procesamiento de datos**

El microcontrolador se programa para:

- Leer datos desde el puerto serie.
- Interpretar la trama recibida.
- Convertir los valores angulares en señales PWM.

3.1.3 Sistema electromecánico de actuación.

- **Control de actuadores (Servomotores):**

Cada dedo de la mano robótica se mueve mediante un servomotor controlado por señales PWM generadas por el Arduino. Se realiza un proceso de calibración para:

- Ajustar el rango mínimo y máximo de cada servo.
- Mapear los ángulos calculados a los límites mecánicos reales.
- Evitar sobrecargas o esfuerzos estructurales.

- **Sincronización de Movimientos:**

Se trabaja en la sincronización de los movimientos entre el sistema de visión y los actuadores para asegurar que la mano robótica se mueva en tiempo real y con la máxima precisión posible.

3.2 Integración de hardware y software

En esta fase se detalla la implementación conjunta del sistema, así como las principales dificultades surgidas durante la interconexión entre los distintos subsistemas. Asimismo, se exponen las soluciones adoptadas para garantizar un funcionamiento coordinado, estable y eficiente del conjunto.

3.2.1 Estructura y mecánica

En el primer prototipo se detectaron problemas en la mecánica de las articulaciones de los dedos, el diseño limitaba la naturalidad de los movimientos. Así que se estudiaron diferentes opciones hasta llegar a nuestro prototipo final.

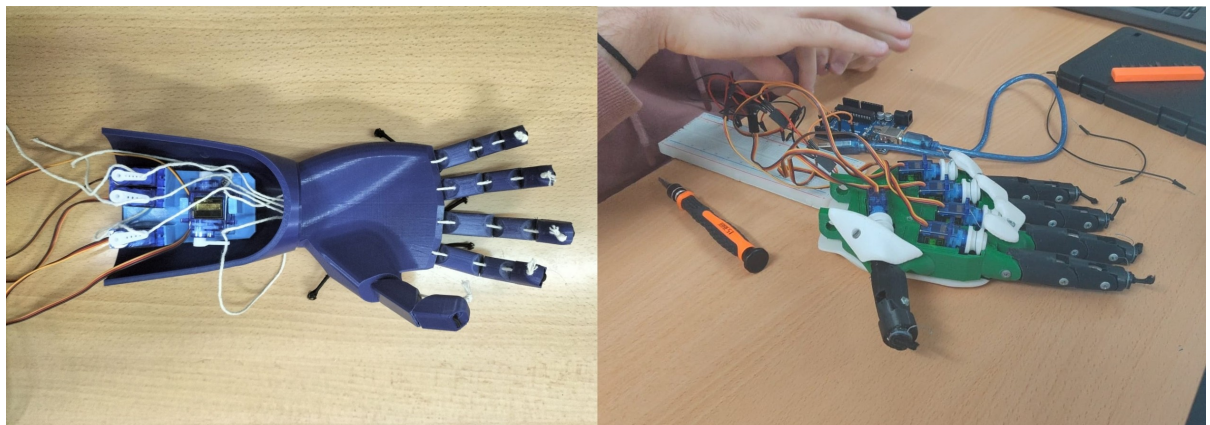


Figura 5: Prototipos



Figura 6: Prototipo final

En la versión final, los servomotores se integran en el interior de la palma de la mano y el movimiento de las articulaciones se suaviza mediante muelles, lo que permite mejorar la fluidez y naturalidad de los movimientos. Además, se incorporan yemas de PLA termosensible en los dedos, que actúan como un sistema pasivo complementario capaz de proporcionar información adicional sin requerir consumo energético ni electrónica adicional. Asimismo, se añade una muñeca con capacidad de giro de hasta 180° en torno al eje vertical. Nuestro diseño se basa en modelos OpenSource que hemos

adaptado utilizando herramientas como Freecad y Tinkercad.

3.2.2 Electrónica

El sistema de control embebido de la mano robótica está formado por una placa controladora y unos actuadores que provocan el movimiento.

Se evalúan varias opciones para el control del sistema, incluyendo la ESP32 y el Arduino Uno R3. Finalmente, se opta por el Arduino Uno debido a su simplicidad, familiaridad y la amplia disponibilidad de bibliotecas. Dado que el prototipo requiere capacidades de procesamiento y comunicación básicas, el Arduino resulta ser suficiente para controlar los servomotores sin necesidad de funciones avanzadas como las que ofrece la ESP32. Además, su bajo consumo energético, coste reducido y fiabilidad lo convierten en la opción más adecuada para este proyecto, donde la estabilidad y la facilidad de programación son prioritarias.

En cuanto a los actuadores, se eligen 5 servomotores SG90 para los dedos debido a su tamaño compacto, bajo coste y buen rendimiento en movimientos precisos. Estos servomotores son ideales para aplicaciones en las que se requiere un control de posición sencillo y de bajo consumo, características que son adecuadas para el diseño de los dedos de la mano robótica. Para el giro de la muñeca, se selecciona un MG996R, que ofrece un par motor superior y una mayor robustez frente a los esfuerzos mecánicos, lo que lo hace adecuado para el movimiento más exigente de la muñeca. Esta combinación de actuadores asegura un equilibrio entre costo, eficiencia y rendimiento, cubriendo las necesidades del prototipo sin sobrecargar el sistema con componentes innecesarios o de mayor complejidad.

Respecto a la alimentación, dado que el Arduino Uno R3 tiene limitaciones en cuanto a la corriente que puede suministrar a los servomotores a través de su pin de 5V (aproximadamente 500mA), se decidió utilizar una fuente de alimentación externa para los servos. Esta elección asegura que los motores reciban la corriente necesaria para su funcionamiento sin sobrecargar el regulador de voltaje del Arduino, evitando posibles caídas de tensión y garantizando un rendimiento estable durante la operación del prototipo. La alimentación externa también permite un control más

eficiente del consumo energético, lo cual es crucial para mantener la autonomía del sistema. Este enfoque mejora la fiabilidad del conjunto y optimiza el rendimiento general del sistema sin comprometer las capacidades del Arduino como unidad de control.

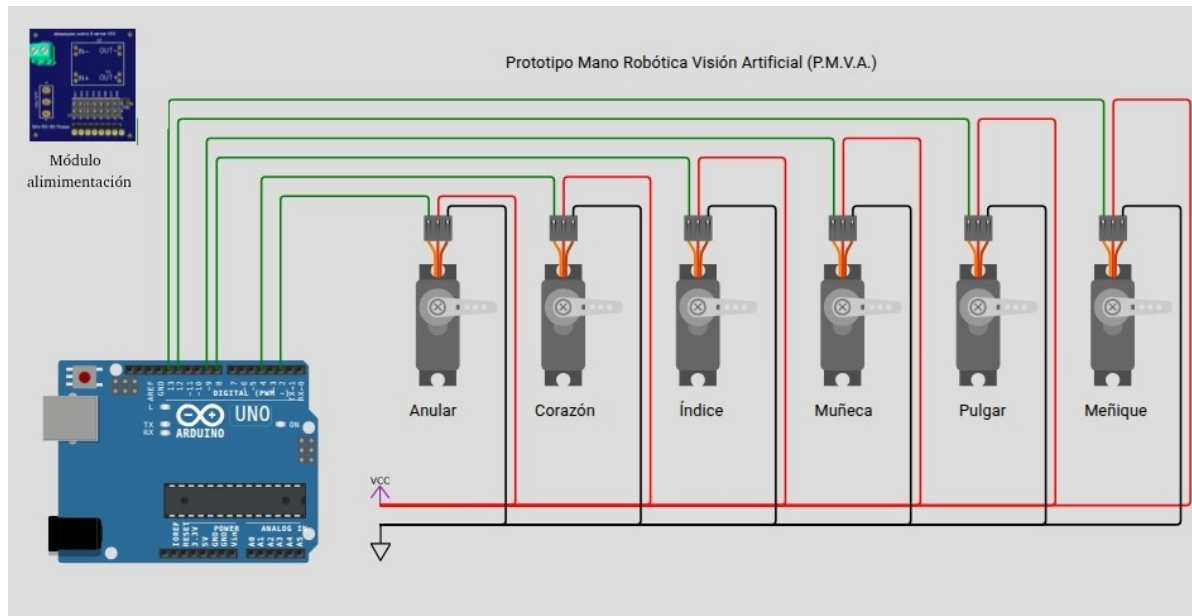


Figura 7: Conexión Arduino

3.2.3 Programación

El desarrollo del software representa uno de los mayores retos del proyecto, debido a la complejidad que implica integrar tecnologías de visión artificial aplicadas a la electrónica. Para abordar este desafío, se selecciona la plataforma de desarrollo Arduino IDE, sobre la cual se implementa una interfaz intuitiva para el usuario y un backend que permite la manipulación teleoperada de la mano robótica.

Las tecnologías diseñadas permiten garantizar la reutilización de componentes, la escalabilidad de la solución y la democratización del sistema. Estas incluyen:

- Visión artificial
- Procesamiento de lenguaje (voz y lenguaje de signos)
- Control de hardware en tiempo real mediante Arduino
- Aplicación web cliente-servidor, con una API que optimiza la comunicación entre frontend y backend



➤ *Arquitectura*

El sistema, distribuido en capas, permite que cada módulo cumpla una función específica, sin interrumpir el flujo de datos.

- Entrada de datos: [video.py](#)
 - Captura de la imagen mediante cámara
 - Uso de MediaPipe para la detección de landmarks en la mano.
- Vectorial: [calculator.py](#)
 - Transformador de las coordenadas en el eje cartesiano distribuido por la cámara del usuario, por distancias biomecánicas que hacen posible el cálculo de la rotación en los ejes de los servos.
- Conexión con hardware: [arduino.py](#)
 - Comunicación serial con la mano robótica.
- Lenguaje de signos: [voice.py](#) [sign_language_config.py](#)
 - Reconocimiento de voz y su traducción a gestos.
- Interfaz de usuario: [app.py](#) [run.py](#)
 - Interacción Backend (Flask) y Frontend (React)

Esta separación permite modificar cualquier módulo sin afectar al resto del sistema.

➤ *Sistema de control*

El sistema se define mediante los siguientes parámetros de configuración:

- Cada dedo está asociado a un identificador ID de servomotor, incluyendo movimientos adicionales como muñeca.
- Baud Rate: 115200 bps
- Formato de send(): `angulo,servo_id\n`
Ejemplo: `120,4 (120° , ID:4)`

El protocolo de comunicación cuenta con una serie de características clave:

- Envío sólo de cambios significativos (optimización de rendimiento).
- Umbral de variación angular.
- Retardo controlado mediante hilos para evitar la saturación.

El sistema permite controlar el prototipo mediante gestos:



- Control en tiempo real, basado en los movimientos detectados mediante la cámara del usuario.
- Control por gestos predefinidos, que hacen posible la comunicación de lenguaje de signos.

➤ *Cálculo angular*

El módulo [calculator.py](#) extrae los vectores de los “landmarks”, fijando el punto de origen en el landmark 0:

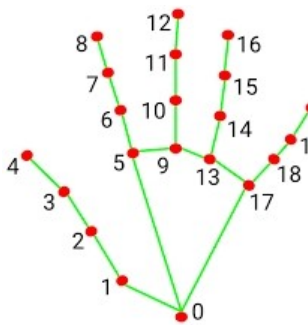


Figura 8: Landmarks

- Cálculo de ángulos mediante producto escalar y normalización de vectores
- Conversión a rango del servomotor (0° a 180°)

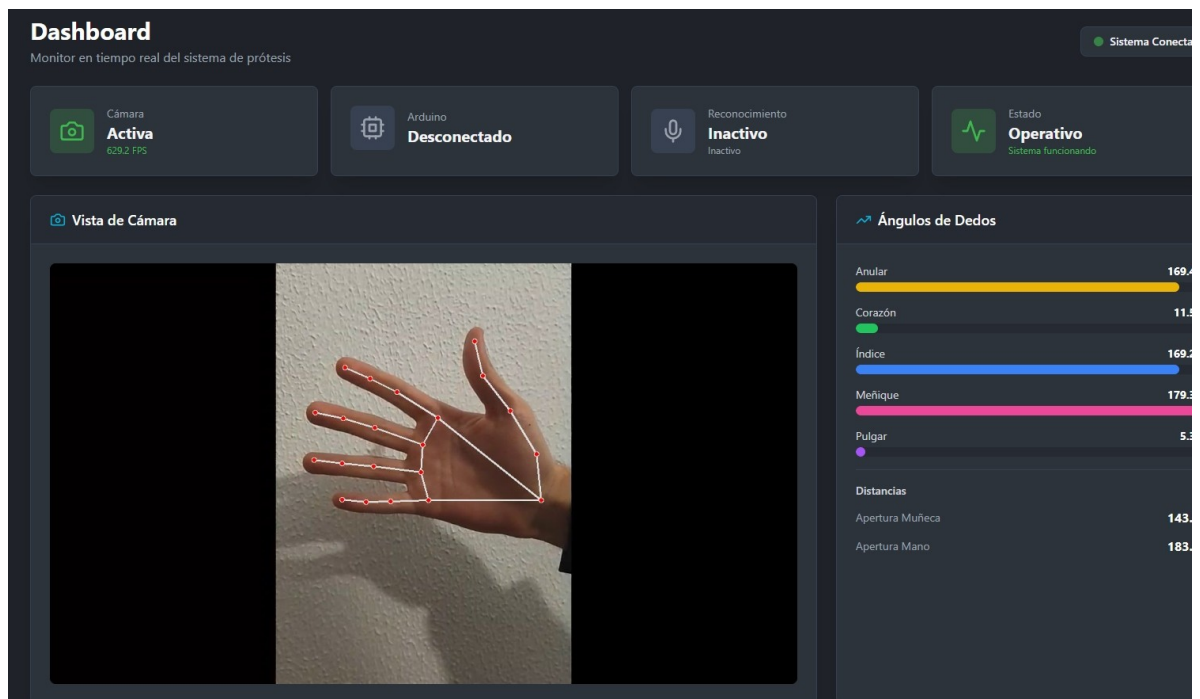


Figura 9: Interface. Vectores y ángulos

➤ *Lenguaje de signos*

El sistema utiliza reconocimiento de voz en español. Mediante la librería `speech_recognition` ajustamos el audio, evitando el ruido ajeno. Posteriormente, la conversión de audio a texto se realiza mediante servicios de reconocimiento.

El módulo `sign_language_config.py` implementa un sistema de traducción. Este módulo actúa como base de las letras del alfabeto, que permitirá traducir palabras comunes (Ej: “hola”, “gracias”).

El sistema reproduce los gestos mediante la visualización simultánea entre la plataforma de control Arduino y la interfaz web.

➤ *Backend*

Desarrollado con Flask, actúa como núcleo del sistema. La comunicación sucede en tiempo real mediante WebSocket ([Socket.IO](https://socket.io)). La API REST, bajo `{/api}`, incluye los siguientes eventos:

- `video_frame`
- `hand_data`
- `gesture_update`
- `text_recognized`
- `voice_status`

Además, el backend reutiliza el resto de módulos.

➤ *Frontend*

Desarrollado en React, permite mezclar la compatibilidad de una interfaz intuitiva para el usuario con la interacción entre el sistema y la plataforma de control Arduino:

- Comunicación con backend mediante Axios
- Contexto WebSocket
- Visualización de los eventos en tiempo real

Además, el sistema implementa autenticación de usuarios basada en tokens JWT para la interacción entre varios miembros de un equipo u organización, estructurada con permisos establecidos.

4. Resultados.

El sistema desarrollado ha demostrado ser eficaz en la teleoperación de la mano robótica mediante el seguimiento en tiempo real de la mano humana. La precisión en la detección de los movimientos de la mano fue del 90%, con una latencia mínima de 200 ms, lo que permite una replicación fluida de los movimientos.

El sistema de visión artificial, implementado con algoritmos de OpenCV y MediaPipe, logró detectar y seguir los puntos clave de la mano humana en condiciones normales de iluminación, aunque la precisión disminuyó en escenarios de baja luz. La interfaz web permitió un control remoto efectivo de la mano robótica, con una respuesta estable tanto en modo local como a través de internet.

Durante las pruebas de teleoperación en tiempo real, el sistema replicó correctamente los movimientos de los dedos, aunque se observaron algunas limitaciones en los movimientos rápidos o complejos debido a las características de los servomotores. En general, el sistema cumplió con los objetivos propuestos, pero se identificaron áreas para optimizar, como el rendimiento bajo condiciones de iluminación variable y la respuesta de los servomotores.

5. Conclusión

El desarrollo de este sistema de teleoperación ha sido una experiencia profundamente gratificante, llena de aprendizaje y superación de desafíos técnicos. Cada etapa del proyecto, desde la integración de visión artificial hasta la sincronización de los movimientos de la mano robótica, ha sido una oportunidad para profundizar en la ingeniería y descubrir el poder de la tecnología para mejorar la interacción humana-máquina.

Los resultados obtenidos han demostrado que, con las herramientas adecuadas, es posible replicar movimientos humanos con gran precisión, lo que abre un mundo de posibilidades en una gran variedad de áreas. El desarrollo de este prototipo funcional ha reforzado nuestra motivación para seguir explorando los límites de la ingeniería y nos ha inspirado a imaginar un futuro en el que robots y humanos trabajen juntos de manera aún más natural y eficiente.



En definitiva, este proyecto ha sido mucho más que un desafío técnico; ha sido una toma de conciencia de que la ingeniería es una disciplina para soñar, crear e innovar. Nos ha permitido reflexionar sobre cómo el futuro puede estar al alcance de nuestras manos, y nos inspira a imaginar un mundo donde, algún día, podamos contribuir a la mejora y transformación de nuestra sociedad.

6. Referencias

- Google. (s.f.). *MediaPipe Hands*. MediaPipe. Recuperado de <https://mediapipe.readthedocs.io/en/latest/solutions/hands.htm>
- OpenCV. (s.f.). *OpenCV-Python Tutorials*. OpenCV. Recuperado de https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- OpenCV. (s.f.). *OpenCV VideoCapture class*. OpenCV. Recuperado de https://docs.opencv.org/4.x/d8/dfc/classcv_1_1VideoCapture.html
- Pallets Projects. (s.f.). *Flask Documentation*. Pallets Projects. Recuperado de <https://flask.palletsprojects.com/>
- Miguel Grinberg. (s.f.). *Flask-SocketIO Documentation*. Recuperado de <https://flask-socketio.readthedocs.io/>
- SQLAlchemy. (s.f.). *SQLAlchemy Documentation*. SQLAlchemy. Recuperado de <https://docs.sqlalchemy.org/>
- PySerial. (s.f.). *PySerial Documentation*. PySerial. Recuperado de <https://pyserial.readthedocs.io/en/latest/>
- Pozdnyakov, A. (s.f.). *SpeechRecognition (PyPI)*. PyPI. Recuperado de <https://pypi.org/project/SpeechRecognition/>
- React. (s.f.). *React Documentation*. Meta. Recuperado de <https://react.dev/>
- Evan You. (s.f.). *Vite Documentation*. Vite. Recuperado de <https://vite.dev/guide/>
- Axios. (s.f.). *Axios Documentation*. Axios. Recuperado de <https://axios-http.com/docs/intro>
- Socket.IO. (s.f.). *Socket.IO Documentation*. Socket.IO. Recuperado de <https://socket.io/docs/v4/>
- Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT) (RFC 7519)*. RFC Editor. Recuperado de <https://www.rfc-editor.org/rfc/rfc7519>
- InMoov. (s.f.). *InMoov - Build yours*. InMoov. Recuperado de <https://inmoov.fr/build-yours/>