

# INTEGRACIÓN DE VISIÓN ARTIFICIAL Y ROBÓTICA EN LA EDUCACIÓN: UN BRAZO ROBÓTICO CONTROLADO POR RASPBERRY PI

Autor Principal: Jaime Pavesi, Universidad Mayor, jaime.pavesi@umayor.cl

Co-autor: Néstor González, Universidad Mayor, nestor.gonzalez@umayor.cl

Co-autor: Carlos Downing, Universidad Mayor, carlos.downing@mayor.cl

Co-autor: Tomás Chaigneau, Universidad Mayor, tomas.chaigneau@mayor.cl

Co-autor: Andrés Ibarra, Universidad Mayor, andres.ibarra@mayor.cl

## RESUMEN

En este trabajo se detalla la implementación de un brazo robótico de 6 grados de libertad, controlado por una Raspberry Pi 4, cuyo propósito inicial es la clasificación de objetos. El sistema permite integrar visión artificial utilizando una cámara USB y algoritmos de procesamiento de imágenes basados en OpenCV para la detección y clasificación de objetos según su forma y color. El objetivo principal de este trabajo es crear una plataforma didáctica que permita desarrollar aplicaciones educativas y de investigación en Inteligencia artificial, robótica y visión artificial, utilizando Raspberry Pi y OpenCV como herramientas principales. Con base en el brazo robótico implementado, se podrán desarrollar algoritmos eficientes de detección, clasificación y manipulación de objetos. Esta propuesta no solo demuestra la aplicación práctica de conceptos teóricos en ingeniería electrónica y computación, sino que también se presenta como una herramienta educativa innovadora, promoviendo la integración de diversas áreas de conocimiento en la formación de ingenieros. Los resultados obtenidos evidencian el potencial del sistema como recurso didáctico para mejorar las metodologías docentes y la evaluación de competencias en carreras de ingeniería.

**PALABRAS CLAVE:** Brazo robótico, Visión artificial, Raspberry Pi, OpenCV, Educación en ingeniería, Metodologías docentes.

## INTRODUCCIÓN

La carrera de Ingeniería Civil Electrónica de la Universidad Mayor incorpora en su plan de estudios materias relacionadas con Inteligencia Artificial, Visión Artificial, Robótica y Sistemas Autónomos. El propósito es entregar una formación equilibrada entre lo teórico y lo práctico, por ello, es muy necesario crear ambientes de aprendizaje adecuados. La bibliografía muestra que se pueden crear estos ambientes haciendo uso de herramientas de acceso libre y tecnologías de bajo costo. Johnston, S.J., & Cox, S.J. (2017) destaca el Raspberry Pi como una tecnología disruptiva, habilitadora de sueños, mientras que Cicolani, J. (2021) proporciona una guía detallada para comenzar con la robótica utilizando Raspberry Pi y Arduino, enfocándose en aplicaciones de visión artificial con OpenCV. En tanto que Tigani, S., Saadane, R., & Chehri, A. (2021) y Yijie Wang, et al. (2023), se refieren a la combinación de robótica con visión artificial. Estas referencias inspiran y motivan la realización del presente trabajo. Además, en GitHub Repository (s/f), se documenta el diseño y desarrollo de un robot que sigue objetos basándose en su color, utilizando Raspberry Pi y visión artificial, con aplicaciones educativas y de investigación.

Este trabajo se enmarca dentro de un proyecto cuyo objetivo es diseñar y probar algoritmos para la detección, clasificación y manipulación de objetos basado en visión

artificial, empleando la biblioteca OpenCV para el procesamiento de imágenes y la detección de figuras geométricas y colores. La incorporación de estos elementos permitirá al brazo robótico identificar y organizar objetos según su forma y color, demostrando aplicaciones prácticas en el ámbito de la automatización y robótica con proyecciones industriales.

El proyecto tiene un impacto significativo en el ámbito educativo, al mejorar de manera sustancial el aprendizaje de los estudiantes en áreas fundamentales de la ingeniería electrónica. La implementación de proyectos prácticos permite aumentar la retención de conocimientos, la motivación y la comprensión de conceptos abstractos al aplicarlos en contextos reales. Además, se potencian las habilidades prácticas en programación, electrónica y visión artificial, promoviendo al mismo tiempo el pensamiento crítico y la capacidad para resolver problemas complejos. El proyecto también fomenta el interés en la robótica y la inteligencia artificial, incentivando la colaboración y el trabajo en equipo, y preparando a los estudiantes para enfrentar los retos tecnológicos del futuro.

Asimismo, la adopción de una metodología estructurada asegura un aprendizaje progresivo y la adquisición de competencias técnicas y transversales esenciales para la formación integral del ingeniero. El proceso se desarrolla en cinco fases: i) Diseño, donde los estudiantes conceptualizan el sistema y sus componentes; ii) Implementación del Hardware, que involucra el ensamblaje físico del brazo robótico y la configuración de la Raspberry Pi; iii) Desarrollo del Software, que incluye la programación de algoritmos de control y visión artificial; iv) Pruebas y Optimización, etapa en la que se evalúa el rendimiento del sistema y se aplican ajustes iterativos; y v) Documentación y Presentación, donde los estudiantes elaboran informes técnicos y realizan demostraciones prácticas ante sus compañeros y en ferias tecnológicas.

## **CONFIGURACIÓN DE HARDWARE**

El brazo robótico implementado, está compuesto por servomotores digitales controlados por una placa PCA9685, conectada a la Raspberry Pi 4. Una cámara USB captura imágenes del entorno, las cuales son procesadas en tiempo real para identificar y clasificar los objetos. La precisión en el control de los servomotores y la eficacia de los algoritmos de visión artificial permitirán que el sistema realice tareas de manipulación de objetos con alta exactitud.

La configuración del hardware es una parte esencial para la implementación del sistema de clasificación de objetos mediante el brazo robótico. A continuación, se detallan las especificaciones técnicas de los componentes principales utilizados en este proyecto, así como el montaje físico del sistema.

### ***Especificaciones técnicas de los componentes***

#### **1. Raspberry Pi 4 Model B:**

- **Sistema en un chip:** Broadcom BCM2711.
- **CPU:** Procesador de cuatro núcleos a 1,5 GHz con arquitectura ARM Cortex-A72.
- **GPU:** VideoCore VI.
- **Memoria:** Disponible en versiones de 1GB, 2GB o 4GB LPDDR4 RAM.
- **Conectividad:** Wi-Fi 802.11ac, Bluetooth 5.0, Gigabit Ethernet.

- **Vídeo y sonido:** Dos puertos micro-HDMI soportan pantallas de hasta 4K a 60Hz, puerto de pantalla MIPI DSI, puerto de cámara MIPI CSI, salida estéreo de 4 polos y puerto de vídeo compuesto.
  - **Puertos:** Dos puertos USB 3.0 y dos puertos USB 2.0.
  - **Alimentación:** 5V/3A vía USB-C, 5V vía cabezal GPIO.
  - **Expansión:** Cabezal GPIO de 40 pines.
2. **Servomotor Digital de Alto Torque DS3218:**
- **Torque:**
    - A 5V: 19 kg/cm.
    - A 6.8V: 21.5 kg/cm.
  - **Velocidad:**
    - 0.16 sec/60° a 5V.
    - 0.14 sec/60° a 6.8V.
  - **Voltaje de funcionamiento:** 4.8 ~ 6.8 Voltios DC.
  - **Ángulo de rotación:** 270°.
  - **Material de piñones:** Cobre y aluminio.
3. **Servomotores MG996R:**
- **Rotación:** 180°.
  - **Engranajes metálicos.**
  - **Voltaje de operación:** 4.8 a 7.2V.
  - **Torque:** 11 kg/cm.
  - **Velocidad de operación:** 0.15 seg/60 grados.
  - **Temperatura de operación:** -30°C a 60°C.
4. **Cámara USB IMILAB modelo CMVJ22A:**
- **Resolución:** 1920 x 1080 (Full HD).
  - **Chipset:** Sonix W88s.
  - **Distancia focal:** 8mm a infinito.
  - **Definición:** 2MP.
  - **Velocidad máxima de fotogramas:** 30 FPS.
  - **Interfaz:** USB 2.0.
5. **Disipador de calor GeekPi ICE:**
- **Método de enfriamiento:** Aire.
  - **Material:** Cobre.
  - **Velocidad máxima de rotación:** 2200 RPM.

## **MONTAJE FÍSICO DEL SISTEMA**

El montaje físico del sistema comenzó con la instalación de la Raspberry Pi 4 en una carcasa impresa en 3D, diseñada para proporcionar un acceso conveniente a los puertos y proteger la placa. Se utilizó el disipador de calor GeekPi ICE para evitar el sobrecalentamiento durante la operación continua.

Los servomotores fueron instalados en el brazo robótico, siguiendo las especificaciones de diseño para asegurar los 6 grados de libertad como se ilustra en la Fig.1. El servomotor DS3218 fue utilizado para proporcionar un torque alto en las articulaciones más exigentes, mientras que los servomotores MG996R se encargaron de las demás articulaciones.

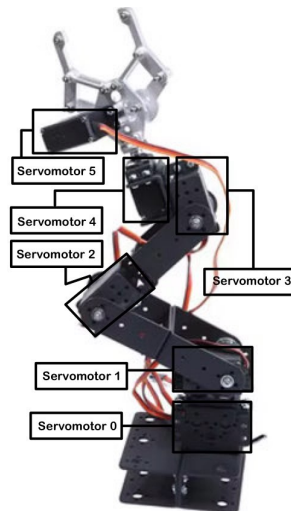


Fig. 1. Esquema del brazo robótico con 6 grados de libertad.

Una cámara USB IMILAB fue montada en una posición fija que permite una vista clara del área de trabajo del brazo robótico. Esta cámara captura imágenes en tiempo real que son procesadas en la Raspberry Pi para la detección y clasificación de objetos.

Para la comunicación y control, se conectó el módulo controlador de servos PCA9685 a la Raspberry Pi a través del bus I2C. Este módulo permite el control preciso de los servomotores mediante señales PWM generadas por la Raspberry Pi.

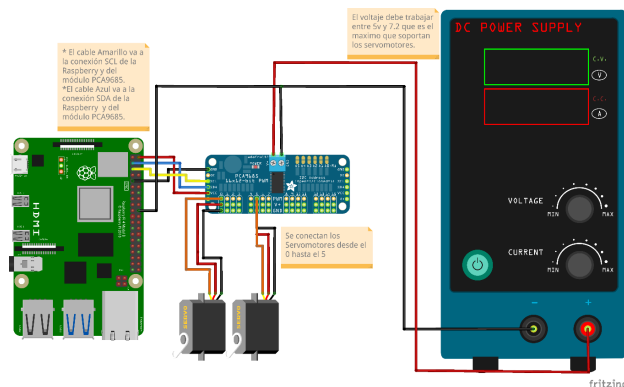
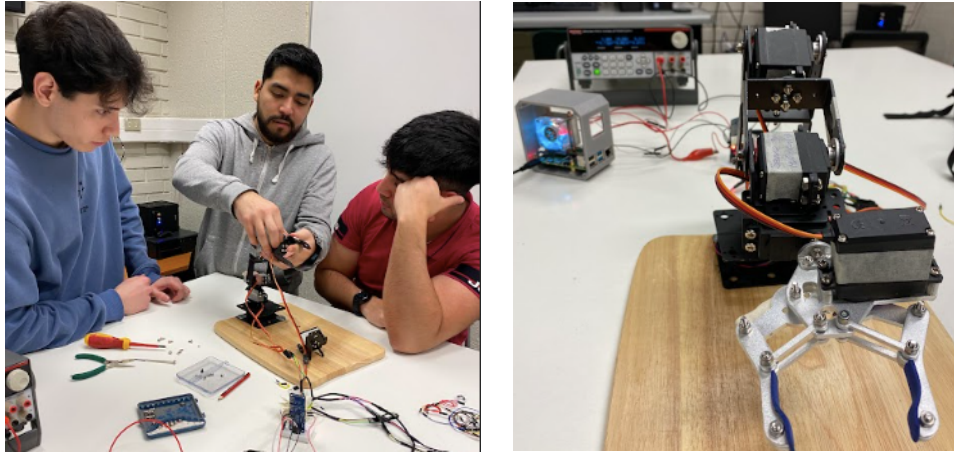


Fig. 2. Diagrama de componentes del sistema.

Como se ilustra en la Fig. 2, todos los componentes se conectan a una fuente de alimentación adecuada para garantizar un suministro estable de energía.

El montaje físico y la configuración de hardware completados proporcionan una base sólida para la implementación de los algoritmos de visión artificial y control del brazo robótico, permitiendo un funcionamiento eficiente y preciso del sistema desarrollado. En la Fig. 3 se muestra evidencia del trabajo de los estudiantes en el armado del brazo robótico.



*Fig. 3. Estudiantes trabajando y brazo robótico armado.*

## **ESPECIFICACIONES DEL SOFTWARE**

La implementación del sistema de clasificación de objetos mediante el brazo robótico con visión artificial requiere la configuración y utilización de varios software esenciales. A continuación, se detallan las especificaciones y configuraciones del software utilizado en este proyecto.

### ***Sistema Operativo y Configuración Inicial***

#### **1. Sistema Operativo Raspberry Pi OS:**

- La Raspberry Pi 4 utilizó el sistema operativo Raspberry Pi OS (anteriormente conocido como Raspbian), el cual es una distribución basada en Debian específicamente optimizada para el hardware de Raspberry Pi.
- La imagen del sistema operativo fue descargada desde la página oficial de Raspberry Pi y cargada en una tarjeta microSD utilizando la herramienta Raspberry Pi Imager, también disponible en el sitio web oficial.

### ***Programas Externos para Conexión y Control Remoto***

#### **2. Advanced IP Scanner:**

- Este software fue utilizado para detectar la dirección IP de la Raspberry Pi cuando estaba conectada a la misma red local. Esto facilitó la conexión remota sin necesidad de un monitor o teclado físico conectado a la Raspberry Pi.
- La herramienta puede ser descargada desde su página web oficial: [Advanced IP Scanner](#).

#### **3. MobaXterm:**

- MobaXterm es una herramienta de computación remota que permite acceder a la consola de comandos de la Raspberry Pi a través de SSH (Secure Shell).
- Una vez conocida la IP de la Raspberry Pi, se utilizó MobaXterm para acceder remotamente y ejecutar comandos en la terminal.
- Disponible para descarga en [MobaXterm](#).

#### 4. RealVNC:

- RealVNC se utilizó para acceder de manera remota al escritorio de la Raspberry Pi mediante el protocolo VNC (Virtual Network Computing), permitiendo un control completo de la interfaz gráfica.
- Después de habilitar la opción VNC en la configuración de la Raspberry Pi (sudo raspi-config), se utilizó RealVNC para conectar y controlar la Raspberry Pi desde otro dispositivo.
- Disponible en [RealVNC](#).

### **Entorno de Desarrollo y Librerías**

#### 5. Thonny IDE:

- Thonny es un entorno de desarrollo integrado (IDE) preinstalado en Raspberry Pi OS, utilizado para escribir y ejecutar scripts en Python.
- Este IDE proporciona una interfaz amigable y es ideal para el desarrollo de aplicaciones que requieren control de hardware y procesamiento de datos en la Raspberry Pi.

#### 6. Librerías de Python:

- **NumPy**: Librería fundamental para operaciones matemáticas complejas necesarias para el control del brazo robótico y cálculos de cinemática inversa. Instalación mediante pip3 install numpy.
- **os**: Librería estándar de Python para interactuar con el sistema operativo.
- **time**: Utilizada para medir el tiempo y gestionar temporizaciones en el control del brazo.
- **adafruit-pca9685**: Librería específica para controlar el módulo controlador de servos PCA9685. Instalación mediante pip3 install adafruit-pca9685.
- **adafruit-motor**: Librería para el control de los servomotores. Instalación mediante pip3 install adafruit-motor.
- **adafruit-circuitpython**: Incluye board y busio para la configuración de pines de la Raspberry Pi y la comunicación I2C. Instalación mediante pip3 install adafruit-circuitpython.
- **ikpy**: Librería para la implementación de cinemática inversa, que incluye ikpy.chain y ikpy.link. Instalación mediante pip3 install ikpy.
- **OpenCV**: Librería principal para el procesamiento de imágenes y la implementación de visión artificial. Instalación mediante pip3 install opencv-python.

### **Desarrollo del Software**

El desarrollo del software involucra la programación de scripts en Python para controlar los servomotores, procesar imágenes capturadas por la cámara y realizar cálculos de cinemática inversa para posicionar el brazo robótico. Para realizar pruebas se utilizó el entorno Thonny IDE para escribir, depurar y ejecutar los scripts. La Fig. 4 muestra el diagrama de flujo del trabajo en desarrollo del software.

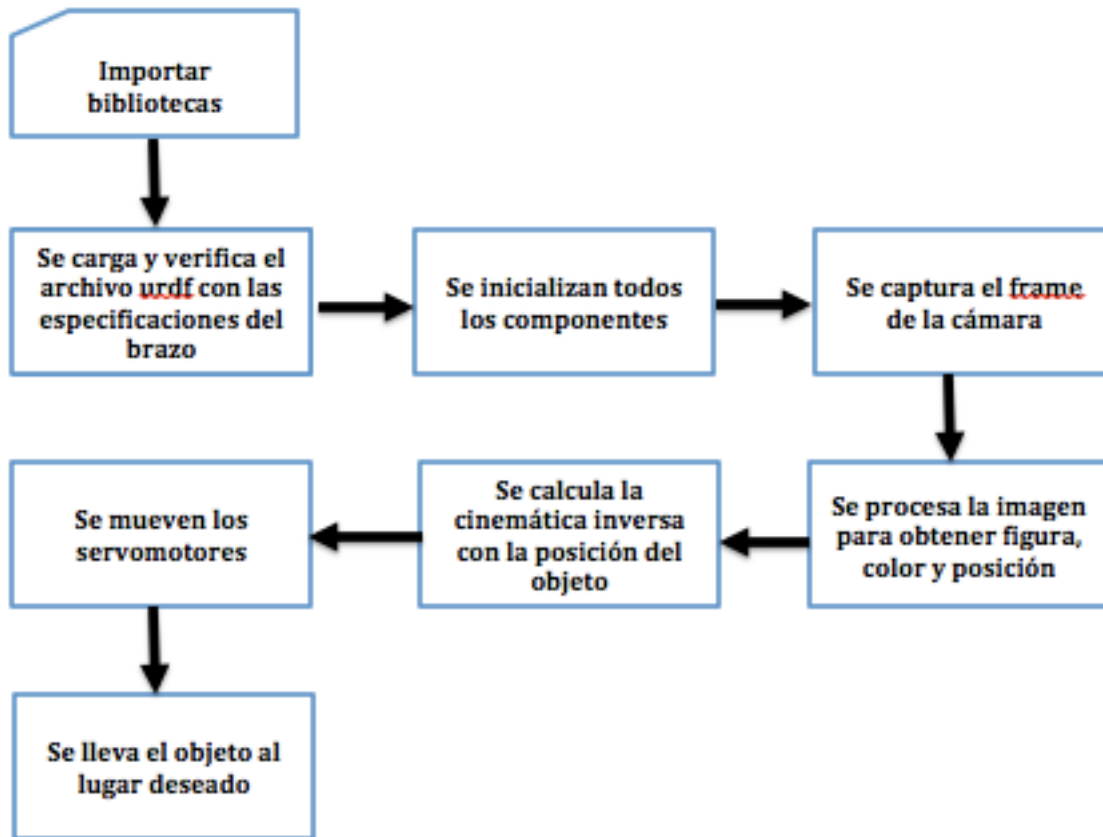


Fig. 4. Diagrama del flujo de trabajo del software.

### **Configuración de la Conexión Remota**

La configuración de la conexión remota incluyó los siguientes pasos:

1. **Detección de la IP de la Raspberry Pi:** Utilizando Advanced IP Scanner para identificar la dirección IP.
2. **Acceso a la Terminal:** Conexión a través de MobaXterm mediante SSH, permitiendo ejecutar comandos directamente en la Raspberry Pi.
3. **Acceso al Escritorio:** Configuración del servidor VNC y conexión con RealVNC para el control completo del entorno gráfico de la Raspberry Pi.

### **Implementación del Software**

El código fuente para pruebas incluyó la inicialización de los componentes, la captura y procesamiento de imágenes, y la ejecución de algoritmos de control del brazo robótico. Los scripts desarrollados permitieron integrar todos los componentes del sistema, asegurando una operación coordinada y eficiente.

En resumen, la configuración y utilización de este software permite la implementación exitosa del sistema de clasificación de objetos, demostrando la viabilidad de utilizar herramientas de código abierto y hardware accesible en proyectos educativos y de automatización.

## CONCLUSIÓN

El desarrollo del sistema de clasificación de objetos mediante un brazo robótico con visión artificial, controlado por una Raspberry Pi 4, ha demostrado ser una herramienta eficaz tanto para aplicaciones prácticas en la automatización como para la enseñanza en el ámbito de la ingeniería. A través de este proyecto, se ha logrado integrar diversos componentes de hardware y software para crear un sistema capaz de detectar, clasificar y manipular objetos en función de su forma y color. La utilización de tecnologías accesibles y de código abierto ha permitido construir una solución viable y educativa que fomenta el aprendizaje interdisciplinario y práctico.

No obstante, el sistema aún se encuentra en una fase de desarrollo, y se han identificado varias áreas de mejora y expansión. Por ejemplo, la precisión y rapidez de los algoritmos de visión artificial pueden ser optimizadas mediante el uso de técnicas avanzadas de procesamiento de imágenes y aprendizaje automático. Además, la robustez del hardware puede ser incrementada para soportar aplicaciones más exigentes y variadas. También se podrían explorar métodos para integrar sensores adicionales que proporcionen información complementaria, como sensores de proximidad o de fuerza, para mejorar la capacidad de interacción del brazo robótico con su entorno.

A futuro, se planea continuar refinando el sistema y expandir sus capacidades para cubrir un espectro más amplio de aplicaciones. Esto incluye la mejora del algoritmo de cinemática inversa para manejar escenarios más complejos y la incorporación de interfaces de usuario más intuitivas para facilitar su uso en entornos educativos. Asimismo, se evaluará la posibilidad de implementar un sistema de aprendizaje adaptativo que permita al brazo robótico mejorar su desempeño a través de la experiencia y el entrenamiento continuo.

Este proyecto ha proporcionado una base sólida para el desarrollo de sistemas de clasificación automatizados y ha demostrado su potencial como recurso educativo innovador. La integración de robótica y visión artificial en la formación de ingenieros no solo enriquece el proceso de aprendizaje, sino que también prepara a los estudiantes para enfrentar los desafíos tecnológicos del futuro con habilidades prácticas y teóricas sólidas. La continuación de este trabajo, que involucra a docentes y estudiantes de varias asignaturas de la carrera Ingeniería Civil Electrónica permitirá aportar mejoras significativas y desarrollar aplicaciones en el campo de la robótica educativa y de la automatización.

En conclusión, el proyecto de desarrollo del brazo robótico con visión artificial ofrece un enfoque integrador que combina diversas disciplinas de la ingeniería. Este trabajo no solo destaca la importancia de la robótica y la visión artificial en la ingeniería moderna, sino que también subraya su papel crucial en la formación de futuros ingenieros, proporcionando un puente entre la teoría y la práctica y su potencial para ser utilizado como la base de una herramienta didáctica innovadora.

## REFERENCIAS

Johnston, S.J., & Cox, S.J. (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. *Electronics* 2017, 6(3), 51; <https://doi.org/10.3390/electronics6030051>

Cicolani, J. (2021). *Beginning Robotics with Raspberry Pi and Arduino: Using Python and OpenCV*. Apress Berkeley, CA. <https://doi.org/10.1007/978-1-4842-6891-9>

Tigani, S., Saadane, R., & Chehri, A. (2021). Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping. *Appl. Sci.* 2021, 11(17), 7917; <https://doi.org/10.3390/app11177917>

Yijie Wang, et al. (2023). Design of a Four-Axis Robot Arm System Based on Machine Vision. *Appl. Sci.* 2023, 13(15), 8836; <https://doi.org/10.3390/app13158836>

GitHub Repository (s/f). FRIDAY-The-Object-Following-Robot: A Project for Color-Based Object Tracking Using Raspberry Pi.  
<https://github.com/farooquesamiya/FRIDAY-The-Object-Following-Robot>