

## EXPERIMENTACIÓN MODULAR DE UN ROBOT DE MANIPULACIÓN CON VA

### EXPERIMENTATION OF A MODULAR MANIPULATION ROBOT WITH A VA

Franco Gabriel Caballero Julián<sup>1</sup>, Eric Mario Silva Cruz<sup>2</sup>, César Hernández Sánchez<sup>3</sup>,  
Anayansi Cristina Hernández Abrego<sup>4</sup>, Martín Vidal Reyes<sup>5</sup>

<https://doi.org/10.61117/ipsumtec.v7i2.317>

<sup>1</sup>Maestro en Ciencias. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Departamento de Ingeniería Electrónica. Correo electrónico: [franco.caballero@itoaxaca.edu.mx](mailto:franco.caballero@itoaxaca.edu.mx), teléfono: 951 1346062. Dirección: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, <https://orcid.org/0000-0002-5924-7759>

<sup>2</sup>Doctor en Ciencias. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Departamento de Ingeniería Electrónica. Correo electrónico: [eric.cruz@itoaxaca.edu.mx](mailto:eric.cruz@itoaxaca.edu.mx), teléfono: 9511065627. Dirección: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, <https://orcid.org/0000-0002-0496-9682>

<sup>3</sup>Maestro en Ciencias. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Departamento de Ingeniería Electrónica. Correo electrónico: [cesar.hernandez@itoaxaca.edu.mx](mailto:cesar.hernandez@itoaxaca.edu.mx), teléfono: 9511959974. Dirección: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, <https://orcid.org/0009-0002-2890-8099>

<sup>4</sup>Maestra en Ciencias. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Departamento de Sistemas y Computación. Correo electrónico: [anayansi.hernandez@itoaxaca.edu.mx](mailto:anayansi.hernandez@itoaxaca.edu.mx), teléfono: 9515690697. Dirección: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, <https://orcid.org/0000-0002-8882-4842>

<sup>5</sup>Maestro en Educación. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Departamento de Ingeniería Electrónica. Correo electrónico: [martin.vidal@itoaxaca.tecnm.mx](mailto:martin.vidal@itoaxaca.tecnm.mx), teléfono: 951 2291428. Dirección: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, <https://orcid.org/0009-0009-1361-0645>

**Resumen** - En este trabajo se hace una segunda entrega en la revista *IPSUMTEC* del *TecNM/Instituto Tecnológico de Milpa Alta*, de los resultados parciales alcanzados en el proyecto de un *Robot de Manipulación* [1] con *Visión Artificial (VA)* para aplicaciones con frutos y semillas en la industria agrícola y de los alimentos. Aquí se presenta el diagrama esquemático y de cableado entre la tarjeta *Arduino Mega* y la estructura del robot, sus servomotores y los potenciómetros que por medio de estructuras de engranajes, leen la posición de cada uno de los servomotores que emulan los eslabones del tronco o cintura, del hombro y del codo, en un comportamiento como el robot *Cincinnati Milacron T<sup>3</sup>* y el robot *PUMA* de *Unimation*[2].

En la operación, partimos de una imagen en formato jpg de la que se extrae una figura circular de color rojo, verde o azul, porque el fondo se colorea de blanco; de esa figura se obtiene el color y su posición en un plano dividido en tres renglones por tres columnas o 9 sectores. En la operación maestro-esclavo, la computadora le pasa a la tarjeta *Arduino-Mega* el color y el sector en el que se identifica la figura.

Con los datos de color y posición recibidos, el sistema digital esclavo ejecuta las rutinas de posicionamiento de un señalador luminoso que apunta en un plano hacia el sector en el que se encontró la figura.

Lo que sigue de este proyecto, el trabajo futuro; es el desarrollo de aplicaciones simples por medio de movimientos con *cinemática inversa* [3,4].

**Palabras clave:** Arduino Mega, control, Python, robot, Visión Artificial.

**Abstract** - In this issue, a second installment is made in the *IPSUMTEC* magazine of the *TecNM/Technological Institute of Milpa Alta*, of the partial results achieved in the project of a *Manipulator Robot with Artificial Vision (VA)* for applications with fruits and seeds in the agricultural industry and of food. Here is the schematic and wiring diagram between the *Arduino Mega* card and the structure of the robot, its servomotors and the potentiometers that, through gear structures, read the position of each of the servomotors that emulate the links of the trunk or waist, of the shoulder and elbow, in a behavior like the *Cincinnati Milacron T<sup>3</sup>* robot and the *PUMA* robot from *Unimation*.

In the operation, we start from an image in jpg format from which a circular figure of red, green or blue is extracted, because the background is colored white; from that figure we obtain the color and its position in a plane divided into three lines by three columns or 9 sectors. In master-slave operation, the computer passes to the *Arduino-Mega* card the color and sector in which the figure is identified.

With the color and position data received, the slave digital system executes the positioning routines of a light marker that points in a plane towards the sector in which the figure was found.

What follows from this project, future work; is the development of simple applications through movements with inverse kinematics.

**Key words:** Arduino Mega, control, Python, robot, Artificial vision.

## INTRODUCCIÓN

Este artículo es una segunda entrega del avance de resultados del trabajo de investigación y experimental en el desarrollo de un *Robot de Manipulación con Visión Artificial* en el área de Ingeniería Electrónica del TecNM/Instituto Tecnológico de Oaxaca, cuyo antecedente está en la publicación número 6, volumen 6, edición especial de la revista *IPSUMTEC* año 2023.

El proyecto consiste en un prototipo de brazo robótico con la capacidad de realizar tareas de agarre, manipulación, y de posicionamiento [5]. En el prototipo tomamos de referencia el comportamiento y funcionalidad del brazo humano, para la ejecución de diversas tareas de rutina.

En este proyecto de un *Robot Manipulador* de tres grados de libertad y pinza con *Visión Artificial* [6] dividido en dos secciones se pretende:

- 1) Diseñar y construir el robot por medio de una selección apropiada de materiales y componentes mecánicos, eléctricos, Sistemas embebidos, computadora, *encoders* y software de programación.
- 2) Programación de su entorno y diseño de aplicaciones.

En el artículo del año 2023, entregamos los resultados de la sección 1. Para resolver la sección 2 en este trabajo presentamos la experimentación modular dividida en dos partes: de Reconocimiento de imágenes y traza de movimiento respectivamente por medio de rutinas básicas preestablecidas; y se deja todavía para trabajo futuro, la experimentación y el diseño de aplicaciones en la solución de necesidades en el sector agrícola y en la preparación de alimentos.

En este artículo se exponen tres aspectos:

- a) En la sección de reconocimiento de imágenes se describe el trabajo para la identificación un objeto [7] por su color dominante, y su posición en el plano de visión.
- b) La sección Mecánica expone la ejecución de movimientos armónicos en los tres motores por medio de rutinas programadas en *Python-Arduino*.
- c) La interfase de reconocimiento-ejecución en la que en función del resultado del procesamiento de la imagen se ejecuta un accionamiento en el robot.

## Entorno de Programación

El sistema digital es por medio de la comunicación en un esquema Maestro-Eslavo, en el que el maestro es una PC con Sistema operativo *Windows 10* y el esclavo es un módulo *Arduino Mega*. En la PC instalamos *Python 3.11* para 64 bits y se habilita desde *Visual Studio Code* para hacer más amigable el entorno de desarrollo.

La interfaz entre *Python* y *Arduino* permite establecer una comunicación bidireccional entre ambos, lo que facilita el intercambio de información y comandos entre el software de control de *Python* y el hardware del brazo robótico controlado por *Arduino*. La instalación de *Arduino* es más del dominio público entre los especialistas en tecnología, y es de interés la elección del puerto serial de comunicación, la misma tarjeta de desarrollo y la velocidad de transmisión de datos a 9600 *baudios*.

*Python* tiene una amplia variedad de librerías y módulos que pueden ser importados en los programas para realizar diferentes tareas. Estas librerías y módulos incluyen herramientas para el análisis de datos, la creación de gráficos, la manipulación de imágenes y la construcción de interfaces gráficas de usuario, entre otros.

*Numpy* es una biblioteca de *Python* que nos crea un objeto con *array multidimensional*, objetos derivados y una serie de opciones para *arrays*. Las matrices, en un sentido estricto, son un tipo de *array* de dos dimensiones, ya que poseen un largo y un ancho preciso.

*Numpy* se basa en una estructura de datos llamada "arreglo" que es similar a las matrices y vectores matemáticos. Los arreglos de *Numpy* permiten realizar operaciones numéricas de manera eficiente en grandes conjuntos de datos, lo que a menudo es más rápido que usar bucles *for* en *Python*.

Se puede instalar desde CMD por medio del siguiente comando.

```
pip install numpy
```

La librería serial proporciona una interfaz para trabajar con dispositivos serie, como sensores, actuadores, *Arduino*, microcontroladores y otros dispositivos que se conectan a través de un puerto serie. Permite a los desarrolladores enviar y recibir datos a través del puerto serie utilizando el protocolo de comunicación serie.

La librería serial también proporciona funciones para configurar y controlar la velocidad de transmisión de datos, el número de bits de datos, la paridad y el número de bits de parada. Estos parámetros se establecen para que el dispositivo serie y el software se comuniquen correctamente.

Se puede instalar desde CMD por medio del siguiente comando:

```
pip install serial
```

La librería *Open CV2*

Su utilización principal abarca la detección de objetos y rostros, especialmente en aspectos como la seguridad, el marketing o incluso la fotografía.

Se puede instalar desde CMD por medio del siguiente comando:

```
pip install opencv-python
```

### Reconocimiento de imágenes

Por medio de *Open CV*, una librería de *Intel*, y de una cámara de vídeo externa dotamos a *Python* de herramientas que hacen más fácil la interacción, se captura una imagen, se pone en formato jpg y a partir de ahí se hace el procesamiento. En esta entrega, el plano se divide en 9 cuadrantes y las imágenes de prueba son círculos rellenos solamente para los tres colores primarios: rojo, azul y verde; cualquier otro color lo va a asociar a estos. En la identificación se devuelve una variable que combina color y posición.

### Sección Mecánica

En esta sección se organizan los movimientos de las tres articulaciones, para tres colores y 9 sectores.

Los grados de libertad en este modelo de brazo robótico se refieren a las diferentes formas en las que puede moverse o posicionarse. El número de grados de libertad [8] de este brazo robótico puede variar dependiendo de su diseño y configuración, pero los grados de libertad comunes incluyen:

**Grado de libertad en la base:** Este grado de libertad permite que todo el brazo robótico gire alrededor de su base. Es el movimiento de rotación en el eje vertical que permite al brazo cambiar su orientación general.

**Grado de libertad en el hombro:** Este grado de libertad permite que el brazo pueda desplazarse hacia arriba y hacia abajo en un plano vertical.

**Grado de libertad en el codo:** Este grado de libertad permite que el brazo se flexione o se extienda en un plano vertical. Es el movimiento de flexión o extensión del brazo y se realiza en el eje horizontal.

**Grado de libertad en la pinza:** Este grado de libertad permite abrir o cerrar la pinza o herramienta de manipulación del brazo robótico (genera el movimiento de agarre o soltar objetos).

### Estructura Mecánica

**Acoplamiento mecánico:** Para conocer las trayectorias se mide el posicionamiento de un servomotor utilizando potenciómetros de 100K. debido a que estos pueden proporcionar información más precisa y en el tiempo real sobre la posición actual del servomotor [9].

**El control de posición:** Conociendo la posición actual del servomotor a través de los potenciómetros, es posible implementar algoritmos de control de posición. Esto permite establecer una posición deseada y utilizar la información de retroalimentación para ajustar la señal de control y lograr que el servomotor se posicione en el destino deseado.

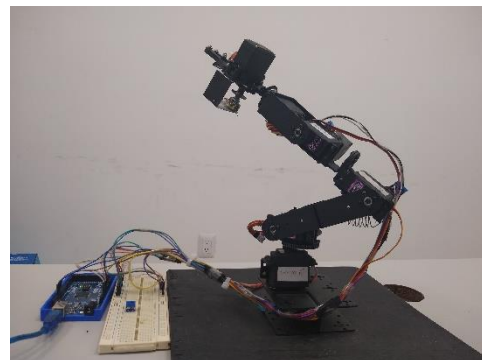
El objetivo de esta prueba es colocar los potenciómetros 100K en la estructura del robot, empleando mecánica, transmisión de energía, fuerza de tensión, acoplamientos extra, etc. Para lograr estos acoplamientos se hicieron pruebas básicas, moviendo servomotor por servomotor para identificar el punto de posicionamiento 0. Lograda la identificación se optó por las siguientes mejoras al modelo:

**Acoplamiento de una base más resistente:** El acoplamiento de la base o plataforma, posee el fin de mantener el robot siempre de pie.

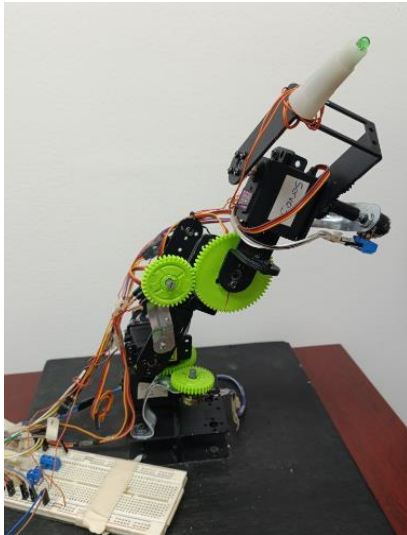
**Acoplamiento de engranajes:** son utilizados con la finalidad de transmitir energía de un servomotor a un potenciómetro, gracias los principios fundamentales de la mecánica.

**Acoplamiento de soportes:** Son materiales esenciales para poder mantener fijo el potenciómetro en una articulación del robot [10], y quede fijo al momento de la transmisión de energía mecánica.

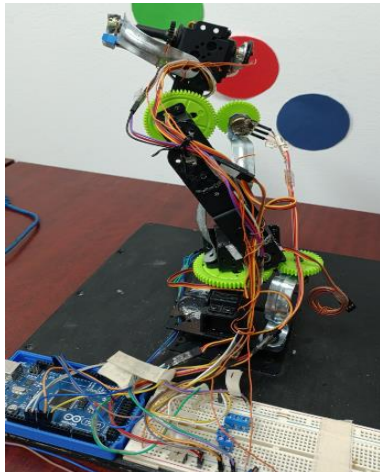
**Acoplamiento de borneras:** Las borneras de 3 espacios fueron utilizadas con la finalidad mantener fijos los cables de alimentación del potenciómetro, puesto que estos se zafaban mucho de los potenciómetros y podía causar retraso en la elaboración del prototipo.



**Figura 1.** Ensamble de la estructura original del prototipo.



**Figura 2.** Nueva presentación de la estructura mecánica del prototipo para aligerar la carga, mejorar el acoplamiento y la interacción.



**Figura 3.** Vista completa de la estructura mecánica del prototipo con sus conexiones hacia la tarjeta Arduino Mega.

## DESARROLLO

En la entrega que antecede a esta publicación, se presentó el diagrama a bloques de la estructura general del *Robot de Manipulación con Visión Artificial* y la forma general del diagrama de flujo de la proyección de su funcionamiento. En esta sección se presentan el diagrama eléctrico del prototipo, el procedimiento para la identificación de figuras a partir de una captura de imagen y las tres rutinas de ejecución preestablecidas.

### Diagrama eléctrico

El diagrama eléctrico se presenta por medio de dos productos en versión académica, el de simulación en *TinkerCad* en el que se utiliza una tarjeta de *Arduino UNO* y el de simulación en *Proteus* es en el que se utiliza una tarjeta *Arduino Mega*, cuya implementación se lleva al prototipo físico.

En el diseño se hacen las siguientes asignaciones:

Un par de cables, la alimentación de 5V y su tierra para cada uno de los tres servos denominados *servo1*, *servo2* y *servo3*.

*servo1* conectado para su control a D9 en la tarjeta *Arduino*.

*servo2* conectado para su control a D10 en la tarjeta *Arduino*.

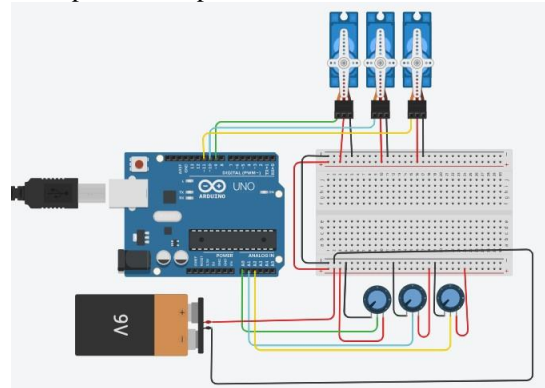
*servo3* conectado para su control a D11 en la tarjeta *Arduino*.

Un par de cables, la alimentación de 5V y su tierra para cada uno de los tres potenciómetros de posición denominados *Pot1*, *Pot2* y *Pot3* asociados a los servomotores *servo1*, *servo2* y *servo3*.

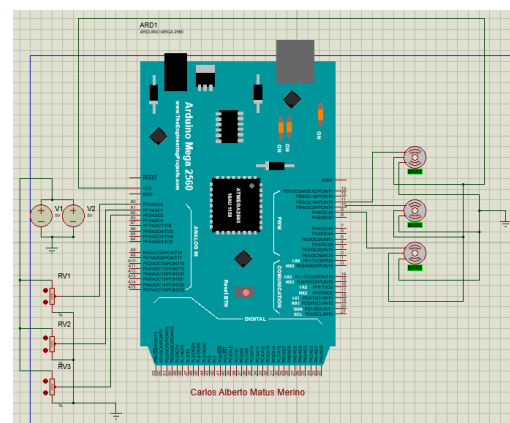
*Pot1* conectado a la entrada analógica A0 en la tarjeta *Arduino* para leer la posición de *servo1*.

*Pot2* conectado a la entrada analógica A1 en la tarjeta *Arduino* para leer la posición de *servo2*.

*Pot3* conectado a la entrada analógica A2 en la tarjeta *Arduino* para leer la posición de *servo3*.



**Figura 4.** Representación del diagrama eléctrico para simulación en *Arduino UNO*.



**Figura 5.** Representación del diagrama eléctrico para simulación en *Arduino MEGA*.

### Relación de los mecanismos

En este prototipo se dispone de las articulaciones de las operaciones de giro (*roll*), elevación (*pitch*) y desviación, que representan el tronco, el hombro y el codo, en referencia a los robots *Cincinnati Milacron T<sup>3</sup>* y el robot *PUMA de Unimation*.

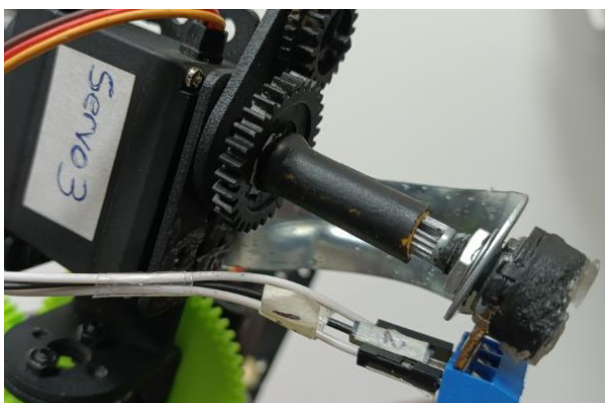
Los mecanismos *servo1* y *servo2*, que corresponden a tronco y hombro tienen un sistema de transmisión del movimiento por medio de engranes con una relación de dientes 60:36; tal que en el recorrido de solamente 180 grados, transmiten casi un giro completo en el potenciómetro.



**Figura 6.** Transmisión de la posición del tronco y del hombro por medio de engranes.

Con una resolución de seis grados o 166.6 mV en el avance o retroceso en el tronco o en el hombro por cada diente.

Para el mecanismo del codo, denominado *servo3* tiene un sistema de transmisión del movimiento directo al potenciómetro 3, sin un engrane de por medio.

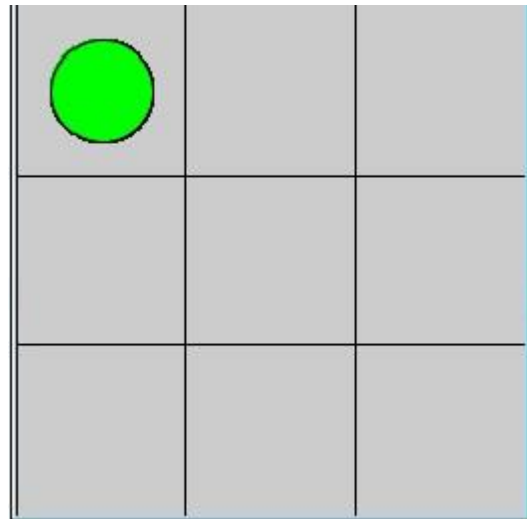


**Figura 7.** Transmisión de la posición del codo.

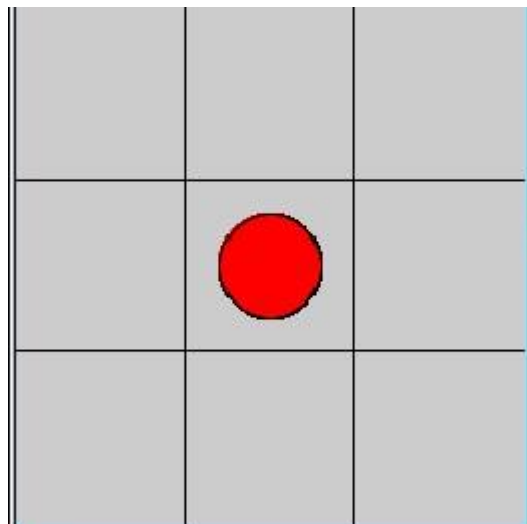
### Extracción de imágenes

El trabajo experimental[11] es con plantillas en formato jpg creadas en *processing*, de 250 x 250 pixeles de tamaño cercano a 10 KB en el entendido que de la cámara se extraen capturas de vídeo (*frames*), en cada plantilla está una figura redonda que puede ser del color de cualquiera de los colores primarios: rojo, verde o azul.

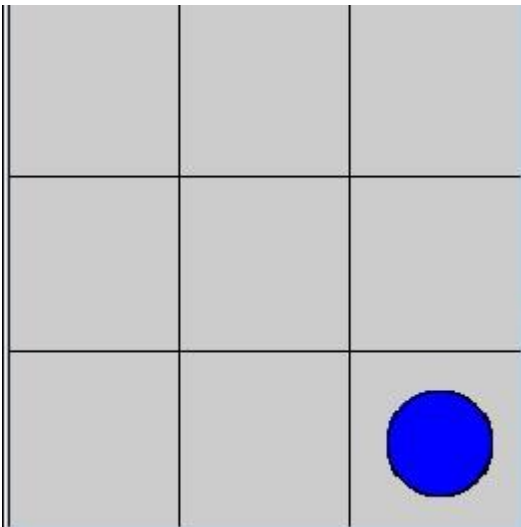
El fondo de cada plantilla es gris, como un tablero y se divide en tres renglones por tres columnas; en nueve sectores que en lo sucesivo se denominarán *sector n* para  $n=1, \dots, 9$ . Todas las posibles combinaciones son en total 3 colores x 9 sectores, en total 27 posibles combinaciones, sólo un color a la vez.



**Figura 8.** Una figura de color verde con posición en el renglón1, columna 1, nombrada sector 1; extraída de la cámara de vídeo para su procesamiento.



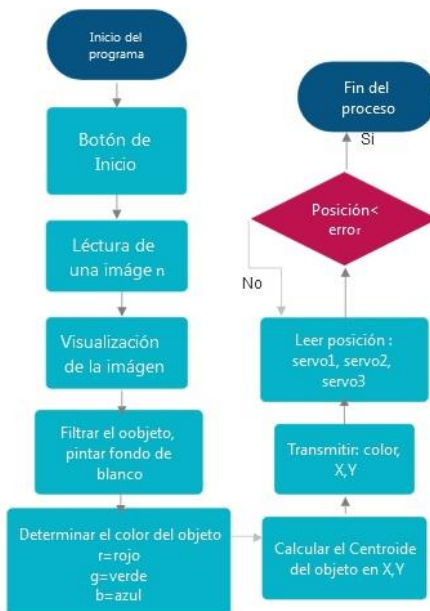
**Figura 9.** Una figura de color rojo con posición en el renglón2, columna 2, nombrada sector 5; extraída de la cámara de vídeo para su procesamiento.



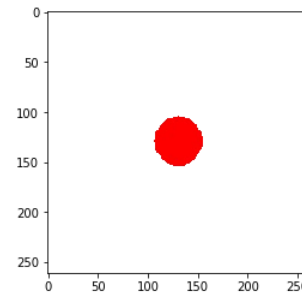
**Figura 10.** Una figura de color azul con posición en el renglón3, columna 3, nombrada sector 9; extraída de la cámara de vídeo para su procesamiento.

### Reconocimiento

La etapa de reconocimiento [12] consiste en obtener una imagen en formato jpg, esa imagen de origen está en tonos BGR, se pasa a RGB, se extrae la figura que en este caso es un círculo concéntrico y se aplica un filtro para dejar la figura en un lienzo con un fondo blanco para eliminar cualquier confusión; se identifica el color y de la media entre el máximo y mínimo de un arreglo que colecciona el número de renglones y columnas en las que está la figura se obtiene su posición X,Y. La computadora transmite a la tarjeta *Arduino*, el color y la posición X,Y.



**Figura 11.** Diagrama de flujo del proceso de identificación de una imagen, filtrado, colocación de fondo blanco, identificación del color, posición X,Y; sector y transmisión de variables.



**Figura 12.** Filtrado de la imagen con fondo blanco, se puede ver su centroide X,Y; y se le asigna un sector.

```
In [43]: arr2
Out[43]: array([130., 131., 125., ..., 134., 135., 136.])

In [44]: np.amín(arr2)
Out[44]: 106.0

In [45]: np.amax(arr2)
Out[45]: 155.0
```

**Figura 13.** De un arreglo se extraen las columnas en las que está pintada la figura, del máximo y mínimo se obtiene el punto medio, se asigna a Y. Se hace lo mismo para obtener X a partir de los renglones. Se le asigna un sector.

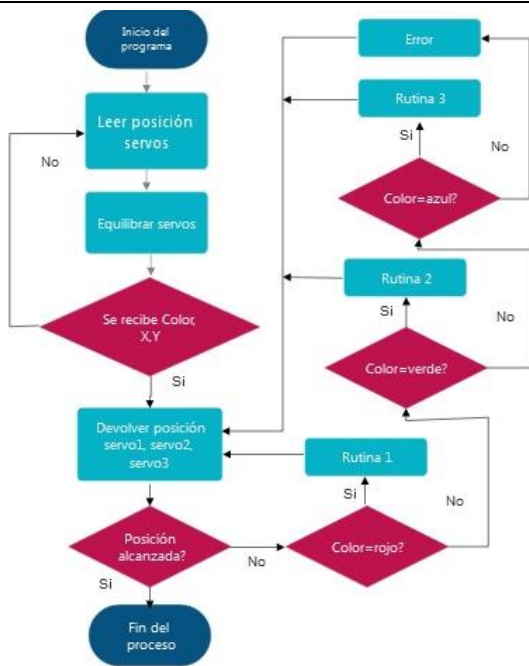
Para el alcance de esta publicación, solamente se dispone de tres rutinas simples[13], que consisten en el movimiento regulado de los tres motores en las secuencias que se indican.

Se define *varTransm*, un entero que se obtiene a partir del reconocimiento de una figura de color y su posición.

- VarTransm=0* al inicio del programa y sí no se reconoce una figura.
- VarTransm=1* sí se reconoce una figura de color verde en el renglón1, columna 1 del tablero.
- VarTransm=2* sí se reconoce una figura de color rojo en el renglón 2, columna 2 del tablero.
- VarTransm=3* sí se reconoce una figura de color azul en el renglón3, columna 3 del tablero.

### Movimientos del Robot

El accionamiento del robot consiste en señalar con un apuntador de luz, en función del color y la posición de la imagen en el plano; así, para un color verde en el sector 1 de la imagen apunta hacía el sector 1 del plano de visión. Para un color rojo en el sector 5 de la imagen apunta hacía el sector 5 de su plano de visión. Para un color azul en el sector 9 de la imagen apunta hacía el sector 9 de su plano de visión. Tener en cuenta que el plano de visión del robot no está bien definido porque no es un robot cartesiano, sino que su desplazamiento es en un espacio de volumen.



**Figura 14.** Accionamiento del Robot. Por medio de tres diferentes rutinas se posiciona el apuntador en el sector 1, sector 5 o en el sector 9.

Al momento, en función del valor de la variable entera *VarTransm*, se ejecuta alguna de las tres rutinas programadas en la secuencia que se indica.

Rutina 1.- Se ejecuta cuando se identifica la figura de color verde en cualquiera de las tres posiciones del primer renglón.

Rutina 1: Motor 1, motor 2, motor 3

Rutina 2.- Se ejecuta cuando se identifica la figura de color rojo en cualquiera de las tres posiciones del segundo renglón.

Rutina 2 : Motor 2, motor 3, motor 1

Rutina 3.- Se ejecuta cuando se identifica la figura de color azul en cualquiera de las tres posiciones del tercer renglón.

Rutina 3: Motor 3, motor 1, motor 2

Los servomotores en este prototipo [14] giran en el sentido contrario a las manecillas del reloj a partir de su referencia en cero para valores del ángulo deseado dados en valores positivos, así el Servomotor 1, montado en el eje Y, gira en el plano X; representa el tronco, tiene un barrido que va desde 30° para posicionarse en la columna 3, manda a 60° para posicionarse en la columna 2 y manda a 90° para posicionarse en la columna 1.

El Servomotor 2, montado en el eje Z, gira en el plano Y; representa el hombro, tiene un barrido que va desde 30° para posicionarse en el renglón 3, manda a 60° para posicionarse en el renglón 2 y manda a 90° para posicionarse en el renglón 1.

El Servomotor 3, montado en el eje X, gira en el plano Z; representa el codo, tiene una función de ajuste espacial de compensación de barrido que va desde 90° como extensión en el renglón 3, manda a 60° para compensación en el renglón 2 y manda a 30° para compensación en el renglón 1.

```
void moveRutina1() {
servo1.write(90);
Serial.println("Movimiento 1 del Tronco: ");
Serial.println("Voltaje potenciómetro a 90 grados: ");
Serial.println(Vout0);
delay(2000);
servo2.write(90);
Serial.println("Movimiento 2 del hombro: ");
Serial.println("Voltaje potenciómetro a 90 grados: ");
Serial.println(Vout1);
delay(2000);
servo3.write(90);
Serial.println("Movimiento 3 del codo: ");
Serial.println("Voltaje potenciómetro a 90 grados: ");
Serial.println(Vout2);
delay(2000);
}

void moveRutina2() {
servo2.write(60);
Serial.println("Movimiento 2 del hombro: ");
Serial.println("Voltaje potenciómetro a 60 grados: ");
Serial.println(Vout1);
delay(2000);
servo3.write(60);
Serial.println("Movimiento3 del codo: ");
Serial.println("Voltaje potenciómetro a 60 grados: ");
Serial.println(Vout2);
delay(2000);
servo1.write(60);
Serial.println("Movimiento1 del tronco: ");
Serial.println("Voltaje potenciómetro a 60 grados: ");
Serial.println(Vout0);
delay(2000);
}

void moveRutina3() {
servo3.write(90);
Serial.println("Movimiento3 del codo: ");
Serial.println("Voltaje potenciómetro a 30 grados: ");
Serial.println(Vout2);
delay(2000);
servo1.write(30);
Serial.println("Movimiento1 del tronco: ");
Serial.println("Voltaje potenciómetro a 30 grados: ");
Serial.println(Vout0);
delay(2000);
servo2.write(30);
Serial.println("Movimiento2 del brazo: ");
Serial.println("Voltaje potenciómetro a 60 grados: ");
Serial.println(Vout1);
delay(2000);
}
```

## DISCUSIÓN Y ANÁLISIS DE RESULTADOS.

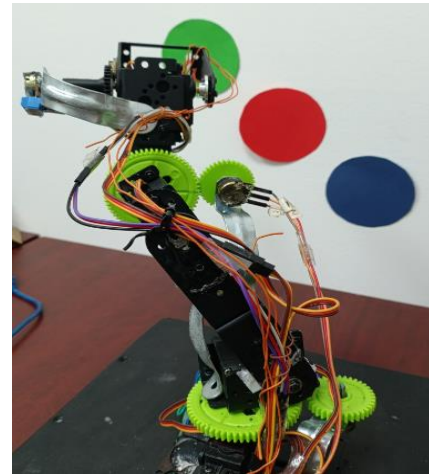
La estructura en uso, a pesar de las diferentes configuraciones; resulta robusta, duradera y consistente. Los servomotores responden fino a las instrucciones, hemos hecho algunos ajustes para liberar del peso demás a la estructura, los sistemas de engranaje transmiten en forma correcta el movimiento y los potenciómetros devuelven el valor de voltaje en un rango de 0-5 Volts. Se están presentando problemas por desajustes imprevistos tales como el ligero desplazamiento en el engranaje o por inesperados falsos contactos en los diferentes puntos de contacto del cableado. En el proceso de pruebas, tanto en *Python* como por medio del puerto de monitoreo serial de *Arduino*, se leen esos datos de posición; que como se indica en los diagramas de flujo, permiten asegurar el posicionamiento de los servomotores ante la posibilidad de desplazamientos inesperados sobre todo en los ejes X y Z.

En la identificación y reconocimiento, elegimos un ejemplo muy fácil; por su color se pueden identificar los frutos y no sería suficiente hallar su centroide, sin duda también va a ser un dato importante su contorno para mejorar su caracterización. La comunicación de los parámetros de identificación hacía el sistema esclavo es muy fácil en *Python*.

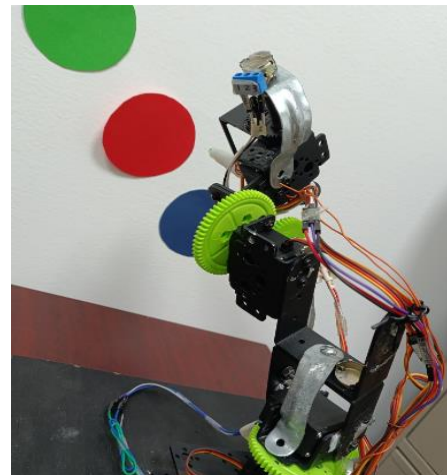
Del lado de los mecanismos del robot, todavía el funcionamiento es totalmente lineal porque no pueden funcionar ni dos servomotores al mismo tiempo; al momento solamente se reflexiona sobre el hecho que el módulo esclavo se puede resolver con un arreglo de multiprocesamiento para mejorar el rendimiento.

Para el ejemplo que está probado, tenemos buena precisión, pero no está considerado un alto grado de exactitud; de hecho, el señalador apunta en un rango muy amplio de los nueve sectores de un plano muy reducido porque al no ser un modelo cartesiano, su desplazamiento es espacial, lo que se tiene que considerar exponencialmente en la subdivisión diferencial en su volumen de desplazamiento.

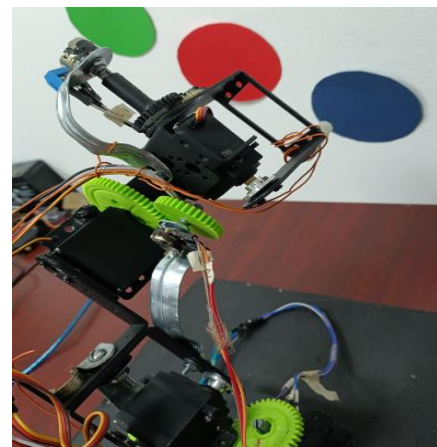
Lo que implica elevar el grado de complejidad [15] del prototipo, con un análisis más sofisticado en la identificación y reconocimiento de las imágenes y en la ejecución de las aplicaciones de interés.



**Figura 15.** Ilustración del accionamiento del manipulador para posicionar el apuntador en la región en la que se identifica un objeto de color verde, sector 1.



**Figura 16.** Ilustración del accionamiento del manipulador para posicionar el apuntador en la región en la que se identifica un objeto de color rojo, sector 5.



**Figura 17.** Ilustración del accionamiento del manipulador para posicionar el apuntador en la región en la que se identifica un objeto de color azul, sector 9.

En este proyecto se tiene como referencia el modelo cinemático de un brazo robótico redundante de 7 DOF de

un humanoide llamado *Mex-One*, en el que se aplica encontrar la solución de la cinemática inversa, con la finalidad de posicionar el brazo robótico en un punto específico en el espacio. Ese trabajo de 7 *DOF* es del equipo de Investigación desde hace 9 años en el *CINVESTAV* unidad Guadalajara.

También son referentes en este estudio, fuera del contexto solamente del trabajo teórico, los resultados experimentales de prototipos que se hacen en Instituciones de educación públicas en México como es el caso de la tesis de Maestría mencionada en [16] en el *Centro de Innovación y Desarrollo Tecnológico en Cómputo del IPN* que desarrolla un *Robot Móvil con Inteligencia Artificial* y también por la elaboración de su estructura mecánica tenemos de referencia el de la *ESIME, Sección de Estudios de Posgrado e Investigación del IPN* en el que se refiere un *Manipulador robótico* para el entrenamiento de personal para cirugía incisiva [17].

### CONCLUSIONES

El trabajo representa un avance significativo en un proyecto de gran alcance. Los resultados parciales son alentadores, se requieren muchas horas de planeación, ejecución y de imaginación; horas y días de pruebas para alcanzar los resultados.

Se tiene que trabajar en tres aspectos; primero en un análisis y procesamiento de imágenes más sofisticado, el segundo aspecto es la dotación del prototipo para uso en aplicaciones de utilidad y finalmente tenemos que considerar dos líneas de investigación para trabajo futuro: técnicas de cinemática inversa que caracteriza los robots comerciales para alcanzar el posicionamiento por medio de operaciones numéricas matriciales; y Multiprocesamiento por medio de varios procesadores en ambiente colaborativo para la distribución de las tareas con la finalidad de mejorar el tiempo de respuesta.

### AGRADECIMIENTOS

Al *TecNM/Instituto Tecnológico de Oaxaca* por brindarnos la oportunidad de llevar a efecto nuestras ideas que inician en la mente y parten del conocimiento que se comunica y se investiga en este espacio educativo, también al *Programa de Desarrollo Profesional (PRODEP)* que tiene los mecanismos para motivar el trabajo creativo de los profesores y los estudiantes.

### BIBLIOGRAFÍA

[1] Martínez, C. (2017). Desarrollo de un robot manipulador blando e hiper-redundante. Trabajo de fin de grado para la obtención del título de graduado en ingeniería de tecnologías industriales. Escuela técnica superior de Ingenieros industriales. Universidad Politécnica de Madrid.  
 [2] Fu, K.S., González, R.C. & Lee, C.S. (1988). *Robótica: Control, detección, visión e inteligencia*. McGrawHill/ Interamericana de España S. A.

ISBN: 84-7615-214-0

[3] Reyes, F. (2011). *Robótica. Control de Robots Manipuladores*. Alfaomega Grupo Editor, S.A. de C.V.  
 [4] Alvarez, J.G. (1999). *Técnicas de control para robots manipuladores con flexibilidad en la articulación*. Tesis de Doctorado. Facultad de Ingeniería Mecánica y Eléctrica. Universidad Autónoma de Nuevo León. México.  
 [5] Carlson, W. B. (2015). *TESLA Inventor de la era eléctrica*. Ediciones culturales Paidós, S.A. de C.V. Obra editada en colaboración con Editorial Planeta-España.  
 [6] Gevarter, W. (1982). *An overview of artificial intelligence and robotics Volume II- Robotics*. National Aeronautics and space Administration. Headquarters. Washington D.C. 20546. U.S. Department of Commerce.  
 [7] International federation of Robotics (2022). *Artificial Intelligence in Robotics*. Position paper. Frankfurt, Germany.  
 [8] Smith, J, et al (2018). *Advances in Manipulator Arms: Industrial Applications and Challenges*. Robotics and Automation, 35(2), 45-58.  
 [9] Jones, A., et al. (2019). *Flexible Manipulator Arms: Applications and Challenges*. Journal of Robotics Engineering, 42(4), 123-136.  
 [10] Liu, S., et al. (2020). *Impact of Manipulator Arms on Industrial Efficiency: A Case Study*. International Journal of Automation and Robotics, 25(3), 87 – 102.  
 [11] Parzibyte. (2021). *Grabar video de cámara con Python, Flask y OpenCV – Cámara de vigilancia*. Parzibyte's blog. <https://parzibyte.me/blog/2021/02/13/grabar-video-camara-python-flask-opencv-camara-vigilancia/>  
 [12] Rodríguez, C., Botero, J., Quintero, H.(2007). *Aplicación de la Teoría de Robots Manipuladores a la biomecánica del brazo humano*. Revista Iberoamericana de Automática e Informática Industrial. ISSN:1697-7912 Vol. 4 pp51-57 <http://riai.isa.upv.es>  
 [13] Reyes, C., et al. (2017). *Estrategias de Control para Robot Manipulador Industrial de 2GDL para generación de trayectoria*. Academia Journals.com. ISSN 2153-3318. Vol. 6 No. 2 pp29.  
 [14] Greyrat, R. (2022b, julio 5). *Detección de múltiples colores en tiempo real usando Python-Open CV- Barcelona Geeks*. <https://barcelonageeks.com/deteccion-de-multiples-colores-en-tiempo-real-usando-python-opencv/>  
 [15] Torres, I.,(2017). *Desarrollo de un algoritmo de visión artificial para el guiado de un robot móvil*. Trabajo de fin de grado. Universidad Politécnica de Valencia.  
 [16] Orozco, R.(2014) *Robot Móvil con Visión Artificial*. Tesis de Maestría en Tecnología de Cómputo. Centro de Innovación y Desarrollo Tecnológico en Cómputo. IPN.  
 [17] Chavez, A., et al. (2009). *Diseño y desarrollo de un manipulador robótico para el entrenamiento de personal en cirugía incisiva*. Científica Vol.13 No. pp 145-155 ESIME, IPN. ISSN 1665-0654.

### ROLES DE CONTRIBUCIÓN

Rol	Autor(es)
-----	-----------

Conceptualización	Franco Gabriel Caballero Julián
Curación de datos	Anayansi Cristina Hernández Abrego
Metodología	César Hernández Sánchez
Administración del Proyecto	Martín Vidal Reyes
Recursos	Martín Vidal Reyes
Software	Anayansi Cristina Hernández Abrego
Supervisión	Eric Mario Silva Cruz
Validación	Franco Gabriel Caballero Julián
Visualización	Franco Gabriel Caballero Julián
Redacción borrador original	César Hernández Sánchez (Grado de contribución: igual) Anayansi Cristina Hernández Abrego (Grado de contribución: igual) Martín Vidal Reyes (Grado de contribución: igual)
Redacción, Revisión y edición	Franco Gabriel Caballero Julián (Grado de contribución: principal) Eric Mario Silva Cruz (Grado de contribución: que apoya)



Esta obra está bajo  
 una licencia internacional  
 Creative Commons Atribución 4.0.