

# DEVELOPMENT OF A 3D PRINTED 2 DOF ROBOTIC ARM THROUGH A PIC 18F46K22 AND MATLAB FOR IMAGES CONTOUR TRACE

Desarrollo de un brazo robótico de 2 DOF impreso en 3d a través de un PIC 18f46k22 y MATLAB para el trazo de contornos de imágenes

**Citación:** Mancipe, A. ; Barrera, N. ; Rojas, D; Barrera, F. Desarrollo de un brazo robótico de 2 DOF impreso en 3d a través de un pic 18f46k22 y MATLAB para el trazo de contornos de imágenes. I + T + C Investigación, Tecnología y Ciencia. Vol 1. Num. 17. 2023.

**Nota del editor:** Sello editorial Unicomfacauc se mantiene neutral con respecto a los reclamos derivados de los resultados de este trabajo.



**Derechos de autor:** © 2023 por los autores. Presentado para posible publicación en acceso abierto bajo los términos y condiciones de la licencia Creative Commons Attribution (CC BY NC SA) ([https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es\\_ES](https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es_ES))

Angie Paola Mancipe García<sup>1</sup> , Nicolás David Barrera Fonseca<sup>2</sup> , Diego Alfonso Rojas Sarmiento<sup>3</sup>  y Fabián Barrera Prieto<sup>4</sup> 

<sup>1</sup> Estudiante Universidad ECCI, Programa en Ingeniería Mecatrónica. [angiep.mancipeg@ecci.edu.co](mailto:angiep.mancipeg@ecci.edu.co)

<sup>2</sup> Estudiante Universidad ECCI, Programa en Ingeniería Mecatrónica. [nicolasd.barreraf@ecci.edu.co](mailto:nicolasd.barreraf@ecci.edu.co)

<sup>3</sup> MSc. Ingeniería Mecatrónica, Universidad Militar Nueva Granada, Colombia. Profesor hora cátedra Universidad ECCI, Programa en Ingeniería Mecatrónica. [drojass@ecci.edu.co](mailto:drojass@ecci.edu.co)

<sup>4</sup> Profesor tiempo completo, Universidad ECCI, Programa en Ingeniería Mecatrónica. [fbarrerap@ecci.edu.co](mailto:fbarrerap@ecci.edu.co)

\*Correspondencia: [angiep.mancipeg@ecci.edu.co](mailto:angiep.mancipeg@ecci.edu.co)

**Resumen:** Este documento describe el desarrollo de un prototipo de un brazo robótico 2R mediante servomotores, programado para diseñar los contornos de distintas imágenes a través de un microcontrolador de bajo costo (PIC 18F46K22) y MATLAB. Estos elementos fueron comunicados por medio de UART. Así mismo, la comunicación entre el PIC y los servomotores fue realizada por I2C, teniendo en cuenta el driver de servomotores (PCA9685). Además, las respectivas simulaciones de funcionamiento se llevaron a cabo utilizando el toolbox de robótica de Peter Corke en MATLAB. Los resultados obtenidos se presentan tanto en físico como en simulación, los cuales son satisfactorios, sin embargo, tienen un margen de error en los trazos. Finalmente, para su implementación, se diseñaron las piezas en SolidWorks para ser posteriormente impresas en 3D utilizando PLA, material elegido por su versatilidad y durabilidad, garantizando la precisión y resistencia requeridas para el brazo robótico 2R.

**Palabras clave:** Brazo robótico 2R 1; MATLAB 2; procesamiento de imágenes 3; PIC 18F46K22 4; Impresión 3D 5.

**Abstract:** This paper describes the development of a prototype of a 2R robotic arm, used for character writing by servo motors, programmed to design the contours of different images through a low-cost microcontroller (PIC 18F46K22) and MATLAB. These elements were communicated through UART. Likewise, the communication between the PIC and the servomotors was performed by I2C, taking into account the servomotor driver (PCA9685). In addition, the respective simulations of operation were carried out using Peter Corke's robotics toolbox in MATLAB. The results obtained are presented in both physical and simulation, which are satisfactory, however these results have a margin of error in the traces. Finally for implementation, the parts were designed in SolidWorks to be subsequently 3D printed using PLA, a material chosen for its versatility and durability, ensuring the precision and strength required for the 2R robotic arm.

**Keywords:** 2R robotic arm 1; MATLAB 2; image processing 3; PIC 18F46K22 4; 3D printing 5.

## 1. Introducción

La robótica industrial es un área que en los últimos años no solo se enfoca en el desarrollo de robots industriales, sino también en la integración de esos robots con diferentes tecnologías que ofrece la industria 4.0, las cuales facilitan y optimizan tareas de automatización en el ámbito de la producción. Hoy en día las grandes empresas que fabrican robots manipuladores los integran con otras soluciones en relación a visión computacional, detección de choques y/o presencia, comunicaciones industriales, entre otros [1].

Los brazos robóticos desempeñan un rol fundamental en la industria, contribuyendo significativamente a la mejora tanto de la producción como de la calidad en diversos procesos [2]. Un punto clave de los brazos robóticos es su extremo, denominado efector final o gripper, el cual presenta una destacada versatilidad. Este componente tiene la capacidad de llevar a cabo múltiples tareas, que van desde la recolección y soldadura hasta la pintura, el posicionamiento y la perforación, siendo aplicables en una amplia gama de contextos industriales [3].

Cabe destacar que el objetivo de este trabajo de investigación es desarrollar un prototipo de un robot de bajo costo que permita realizar el diseño de diferentes formas a través de procesamiento de imágenes, este enfoque involucra la aplicación de una serie de técnicas y algoritmos utilizados para modificar, analizar y manipular imágenes digitales con el objetivo de mejorar su calidad y extraer información relevante [4].

Para su elaboración se escogió la configuración de un robot angular con 2 grados de libertad el cual opera en un plano bidimensional y está compuesto por dos articulaciones rotatorias. Estas articulaciones le permiten moverse y posicionarse en un área específica, su funcionamiento se basa en la manipulación controlada de los ángulos de sus articulaciones para lograr distintas posiciones y orientaciones en su espacio de trabajo [5].

El control del robot 2R se ejecuta mediante el empleo de un microcontrolador PIC18F46K22 [6]. En este proceso, la programación se lleva a cabo en colaboración con dos entornos de desarrollo integrados (IDE): MPLAB y MATLAB; en el caso de MATLAB [7], su enfoque reside en la manipulación y procesamiento de datos, incluyendo cálculos complejos, simulaciones y análisis numéricos. Su capacidad para trabajar con matrices y vectores permite desarrollar algoritmos avanzados de cinemática, facilitando así la planificación y control de los movimientos del robot 2R. Por otro lado, el IDE MPLAB [8] se orienta hacia la programación específica de microcontroladores, proporcionando un ambiente dedicado a la codificación, compilación y depuración de código de bajo nivel. Esta herramienta es esencial para la gestión directa del microcontrolador PIC18F46K22, asegurando una comunicación fluida entre el controlador y los servomotores del robot 2R.

El desarrollo se enriquece con el uso del Toolbox Peter Corke de MATLAB, que habilita la visualización y simulación gráfica del robot 2R. Esta herramienta potencia la representación virtual de los movimientos del robot, brindando una mayor comprensión y análisis de su desempeño. La integración del Toolbox en MATLAB ofrece una experiencia completa y efectiva en el diseño y simulación del robot 2R [9].

Adicionalmente, se empleó la herramienta GUIDE de MATLAB para crear la Interfaz Gráfica de Usuario (GUI). Esto permite al usuario elegir el modo de operación de manera intuitiva. La herramienta GUIDE proporciona un entorno visual que simplifica la creación de la interfaz, entregando opciones interactivas al usuario para interactuar y elegir modalidades de funcionamiento. Esta sinergia entre GUIDE y MATLAB brinda una interfaz eficiente para el control y la interacción con el robot [10].

Se emplearon las técnicas de Cinemática Directa e Inversa para lograr el movimiento del robot 2R. La Cinemática Directa calculó la posición del efector final a partir de los ángulos de cada articulación, mientras que la Cinemática Inversa determinó los ángulos necesarios para posicionar el efector final. Los modelamientos matemáticos de cada articulación definieron las relaciones geométricas y trigonométricas para los

movimientos y transformaciones entre ellas y el efector final. Estas ecuaciones y algoritmos posibilitaron el control y la planificación de los movimientos. La combinación de estas técnicas brindó una comprensión detallada de la relación entre ángulos de articulación y posición del efector final, permitiendo un control preciso y programación efectiva de movimientos específicos [11].

Para llevar a cabo la cinemática directa en el robot 2R, se utilizó el método DH (Denavit-Hartenberg). Este método se fundamenta en un enfoque sistemático que permite establecer sistemas de coordenadas para cada uno de los eslabones del robot 2R, lo que a su vez genera los parámetros DH,  $d$ ,  $\alpha$ ,  $a$ . Al emplear el método DH en la cinemática directa, se logró establecer una relación clara y sistemática entre los ángulos de las articulaciones, la posición y orientación del efector final del robot 2R [12].

El sistema opera de la siguiente manera: a través de la interfaz gráfica desarrollada en GUIDE, se establecen los ángulos a los que el robot debe llegar. Estos ángulos se envían al microcontrolador PIC 18F46K22 utilizando el protocolo de comunicación UART [13]. Para la transmisión de los datos, se utiliza el componente TTL. Una vez que los datos llegan al microcontrolador, este los procesa y los convierte en caracteres legibles para ser mostrados en la pantalla LCD 2x16, por último, los datos son enviados a la PCA9685 [14] mediante el protocolo de comunicación I2C [15], esta se encarga de generar el movimiento en los servomotores con base en los ángulos proporcionados permitiendo que el robot se desplace según las instrucciones recibidas.

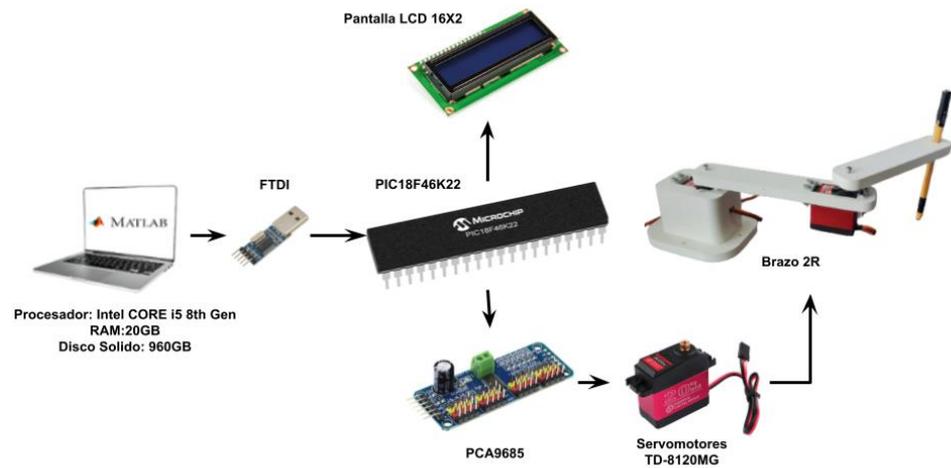
En el proceso de desarrollo del código, se consideraron tanto las dimensiones de los eslabones como los grados de libertad de cada articulación. Estos factores fueron fundamentales para la construcción física del robot. Además, se implementaron dos servomotores con un rango de movimiento de  $0^\circ$  a  $180^\circ$ .

Con base en la información recopilada, se procedió a realizar el diseño del robot 2R utilizando el software Solidworks [16]. Una vez completado el diseño, se procedió a la impresión en 3D del robot utilizando PLA (ácido poliláctico) un material ampliamente utilizado en la impresión 3D debido a sus diversas características y ventajas. Este paso marcó el inicio de la fase de pruebas y evaluación, donde se llevarán a cabo los ensayos necesarios para garantizar el funcionamiento adecuado y la precisión del robot en su ejecución.

## 2. Metodología

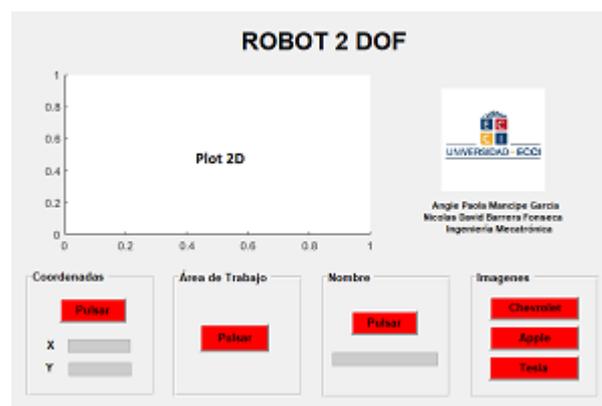
### 2.1. Integración

Este proyecto se centró en integrar la robótica de manipuladores serial y el procesamiento de imágenes con el fin de identificar los contornos de una imagen a través de Matlab y así posteriormente realizar las trayectorias de dichos contornos teniendo en cuenta la cinemática inversa del robot 2R; en la integración de los diferentes componentes que conforman el proyecto, se encuentran: (a) una computadora con Matlab en donde se realiza el procesamiento de imágenes, el cálculo de la cinemática inversa, la simulación en Peter Corke y la interfaz gráfica, (b) el PIC18F46K22 que sincroniza toda la información del proyecto, (c) una pantalla LCD 16X2 para visualizar los ángulos (grados) correspondientes a cada movimiento, (d) una PCA9685 que controla los dos servomotores del robot, (e) un FTDI que permite llevar a cabo la comunicación UART entre Matlab y el microcontrolador, así como se detalla en la Figura 1. Matlab se utilizó con cuatro propósitos, el primero fue realizar el procesamiento de imágenes a través de algunas funciones predefinidas para obtener así la caracterización de las coordenadas en X y Y de cada contorno de la imagen analizada; el segundo correspondió al cálculo de la cinemática inversa; el tercero fue aprovechar el toolbox de Peter Corke para simular el funcionamiento de los diseños de los contornos de las imágenes y el cuarto propósito fue emplear la herramienta GUIDE, para desarrollar la Interfaz Gráfica de Usuario, la cual permite al usuario interactuar con el sistema para entrada y salida de información.



**Figura 1.** Interacción entre dispositivos y periféricos

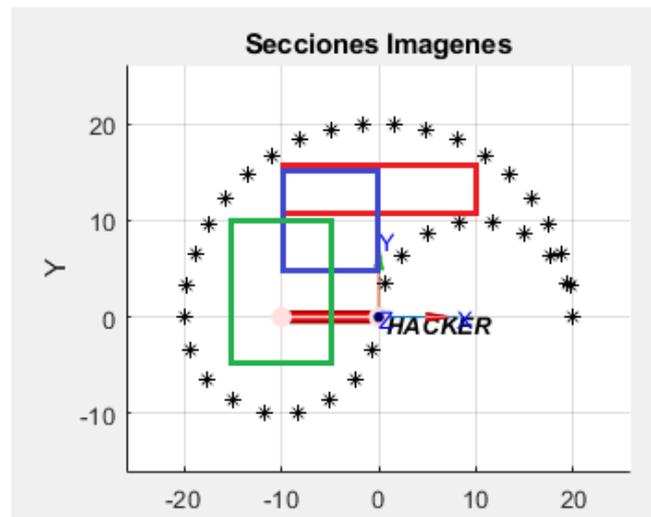
Se implementaron cuatro modos de operación para el robot 2R que son presentados en la Figura 2, donde las figuras y trazos se verán ilustrados en el *Plot 2D*. En el primer modo, el usuario proporciona las coordenadas (X y Y) dentro del espacio de trabajo del robot para que dicho robot llegue a dicha ubicación deseada, a través de la Cinemática Inversa. En el segundo modo de operación, al presionar el botón "Pulsar", se traza el contorno del robot 2R por medio de cinemática directa, el cual hace relación al límite del área de trabajo que está determinado por la longitud de los eslabones del mismo. En el tercer modo de operación, el usuario ingresa una palabra de hasta 9 caracteres en un cuadro de texto, que se visualizará tanto en la simulación como en el montaje físico; para el proceso de escritura de cada carácter se utilizan las Cinemáticas Directa e Inversa. Finalmente, en el cuarto modo de operación, el usuario elige entre tres imágenes preconfiguradas a través de botones para que el robot las dibuje en papel, mientras al mismo tiempo se va observando el comportamiento de los trazos en simulación.



**Figura 2.** Interfaz Gráfica del brazo robótico de 2 DOF

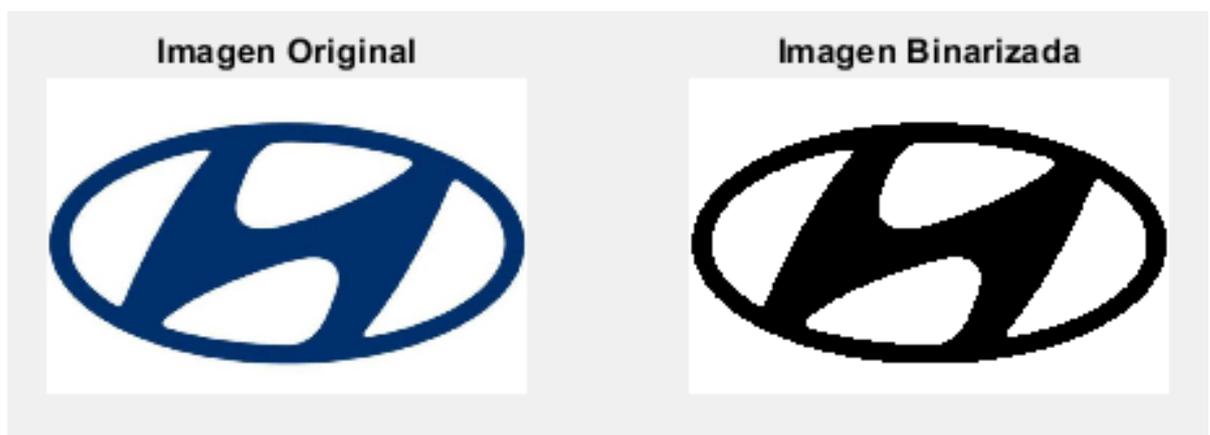
## 2.2. Procesamiento de imágenes

La fase inicial del procesamiento de imágenes, consistió en dividir el espacio de trabajo del robot en tres secciones distintas, cada una destinada a trazar imágenes específicas de diferente tamaño y forma, así como se presenta en la Figura 3, con el objetivo de hacer uso eficiente del área de trabajo del robot 2R. La sección de color rojo se designó para imágenes de formato horizontal, la sección azul para aquellas de formato cuadrado, y finalmente la sección verde se reservó para imágenes verticales.



**Figura 3.** Secciones para las Imágenes

Posteriormente a que se ha establecido el espacio de trabajo para una imagen específica, el siguiente paso consiste en binarizar la imagen, tal como se muestra en la Figura 4. Este proceso de conversión resulta fundamental para simplificar la representación visual, facilitando así el avance hacia la siguiente etapa del proceso.

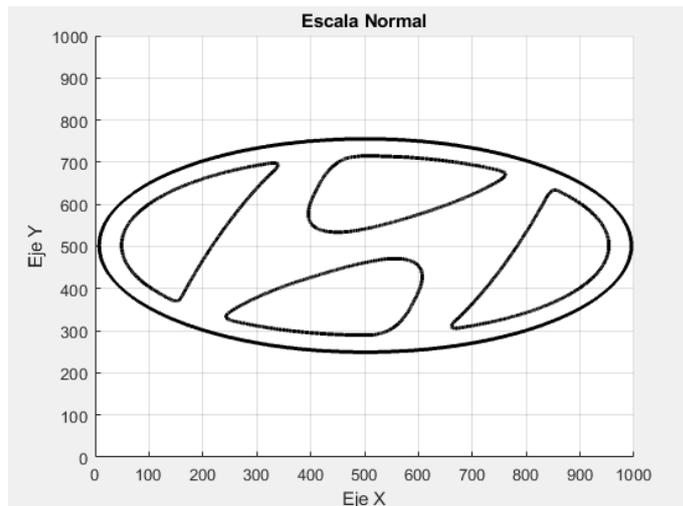


(a)

(b)

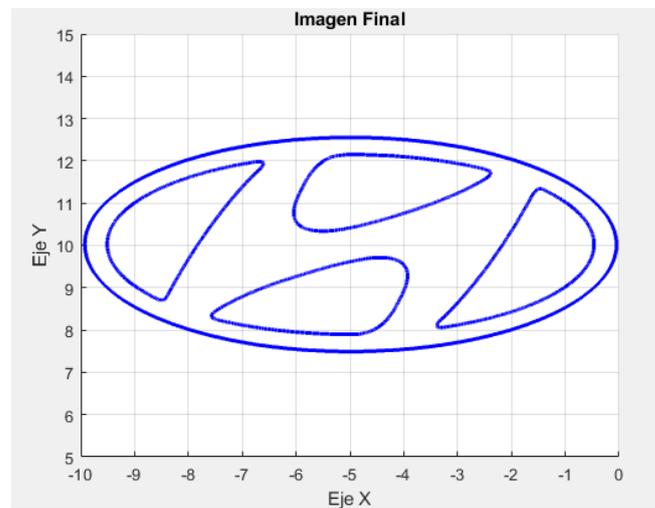
**Figura 4.** (a) Imagen en formato Original; (b) Imagen Binarizada.

Posteriormente, utilizando la función integrada de Matlab `bwboundaries`, se extraen las coordenadas (X y Y) de los contornos tanto externos como internos, y los resultados de este proceso se ilustran en la Figura 5. Cada contorno identificado en la imagen se almacena en un vector; en este caso, se detectaron cinco contornos.



**Figura 5. (a)** Imagen a escala original.

Después, se modificó la escala de la Figura 5, a través de una división de cada coordenada (X y Y) entre el valor de 100 con el propósito de reducir la imagen. Finalmente, en la Figura 6 se presenta la imagen desplazada a las coordenadas predefinidas en la Figura 3, es decir, se trazó en la sección cuadrada que corresponde a la sección cuadrada. De igual manera, es relevante enfatizar que este proceso es compatible con una amplia variedad de imágenes y/o logotipos.



**Figura 6.** Imagen Final

### 2.3. Cinemática Directa 2R

El objetivo es determinar la posición (traslación) y orientación (rotación) del efector final, es decir, el TCP, con respecto al sistema de referencia. En el IDE de MATLAB, esta cinemática se utiliza para graficar y adquirir las coordenadas X y Y y graficarlas en el robot 2R de Peter Corke. Para lograr esto, se requiere conocer los ángulos de las articulaciones y la longitud de sus eslabones previamente. En este caso específico, se empleó el método sistemático Denavit-Hartenberg (DH), que recibe su nombre de sus creadores, Denavit y Hartenberg, quienes lo desarrollaron en 1955; dicho método es un enfoque matricial que implica establecer de manera sistemática un sistema de coordenadas final con respecto al sistema de coordenadas de referencia teniendo en cuenta tres pasos: 1. Asignación de sistemas coordenados, 2. Parámetros DH y 3. Matriz de Transformación Homogénea.

Es importante destacar que este robot Planar 2R se desplaza únicamente en un plano (X y Y), pues no posee una articulación vertical, lo que significa que sus movimientos están limitados a los ejes X e Y. En la Figura 7 se presentan los sistemas coordenados del robot 2R desarrollado en este proyecto, teniendo en cuenta una longitud de eslabones  $L_1 = 13.5$  cm,  $L_2 = 11$  cm.

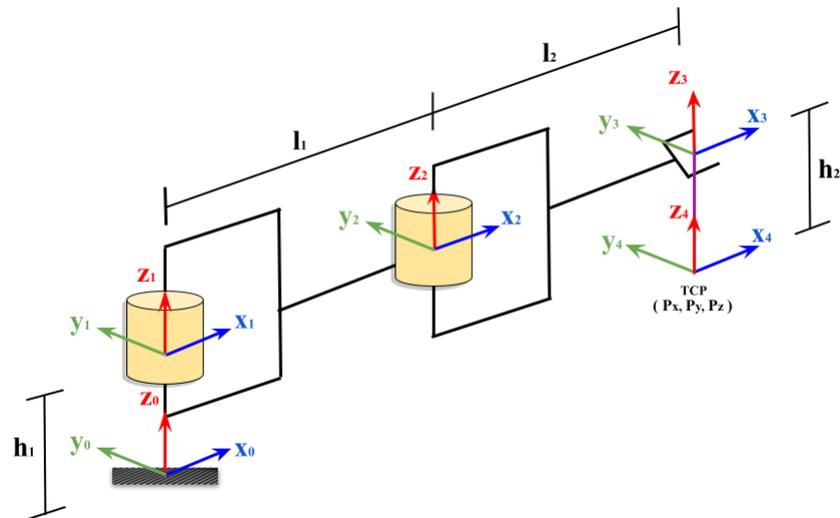


Figura 7. Sistemas Coordinados (SC) robot 2R

El cálculo de la cinemática directa del robot 2R, como se ilustra en la Figura 7, conduce a una Matriz de Transformación Homogénea (MTH). Esta MTH refleja la posición y orientación finales del TCP en relación con el sistema de coordenadas de referencia. Estos resultados se formalizan mediante la Ecuación 1.

$$T_3^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\text{sen}(\theta_1 + \theta_2) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \text{sen}(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1 \text{sen}(\theta_1) + l_2 \text{sen}(\theta_1 + \theta_2) \\ 0 & 0 & 1 & h_1 - h_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 1. MTH del robot 2R

#### 2.4. Cinemática Inversa 2R

El concepto opuesto a la Cinemática Directa es la Cinemática Inversa. En la Cinemática Inversa, se busca determinar los movimientos angulares y/o lineales de las articulaciones, en este caso, dos articulaciones rotacionales a partir de las coordenadas a las que debe llegar el robot en los ejes X e Y; es decir, se calculan los ángulos correspondientes ( $\theta_1$  y  $\theta_2$ ) que permiten posicionar el robot en una ubicación específica en el plano bidimensional. Esto es posible solucionarlo mediante cálculos inversos basados en la geometría y estructura del robot, siendo especialmente útil cuando se desea programar el robot para que alcance una posición y orientación determinada en el espacio de trabajo. En la Figura 8 se presenta el comportamiento de las dos articulaciones, las cuales se relacionan directamente a los ángulos  $\theta_1$  y  $\theta_2$ .

Figura 8. Vistas del robot Planar 2R

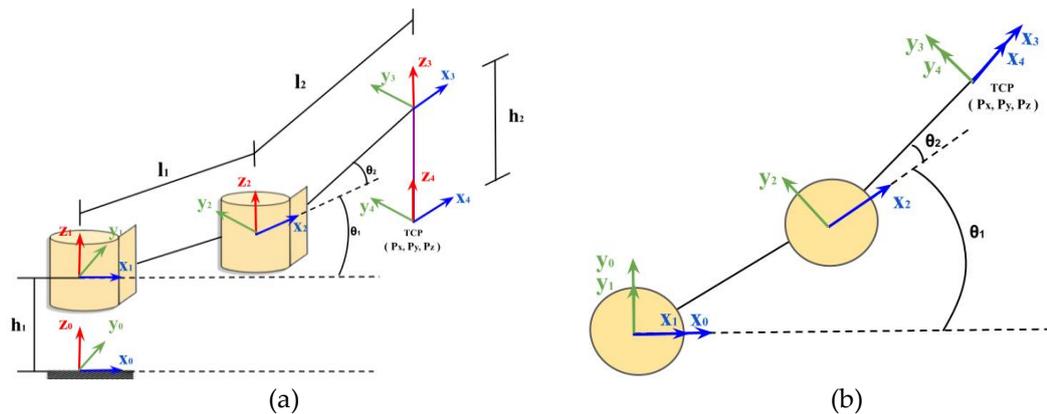


Figura 8. (a) Modelo Cinemático 3D; (b) Modelo Cinemático 2D (Vista Superior).

A través de un análisis geométrico del robot de la Figura 8 se presentan las ecuaciones finales que representan la cinemática inversa del robot 2R en términos de  $\theta_1$  y  $\theta_2$  [17].

$$\theta_1 = \tan^{-1} \left( \frac{Py}{Px} \right) - \tan^{-1} \left( \frac{l_2 \cdot \text{sen } \theta_2}{l_1 + l_2 \cdot \text{cos } \theta_2} \right) \quad (1)$$

$$\theta_2 = \tan^{-1} \left( \frac{\text{sen } \theta_2}{\text{cos } \theta_2} \right) \quad (2)$$

### 2.5. Comunicación entre Matlab y Pic

Luego de obtener los ángulos por medio de la cinemática inversa, se inicia la comunicación serial entre Matlab y el Pic, donde cada ángulo se convierte en una cadena de caracteres antes de ser enviado individualmente a través del puerto serial. Una vez completada la transmisión de datos, se cierra el puerto serial. Este proceso se repetirá siempre que sea necesario para mover las articulaciones del robot.

La secuencia de recepción de datos en el Pic para controlar los servos se realiza mediante el protocolo de comunicación UART. La trama de datos comienza con la letra 'A' indicando la transmisión de datos próximos del servo 1. Posteriormente se envía el ángulo al cual se moverá el servomotor, seguido por la letra 'O' como primera confirmación, indicando que se ha recibido el primer ángulo y poder transformarlo a un formato decimal.

Este procedimiento se repite una segunda vez, pero esta vez se usa la letra 'B' para obtener el ángulo del servo 2. Finalmente, se emite una confirmación total con la letra 'K', indicando que ambos ángulos están codificados y listos para mover los servos de manera casi simultánea. Véase la Figura 9 para mayor comprensión.

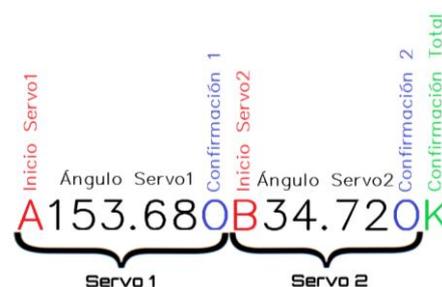


Figura 9. Recepción trama de datos

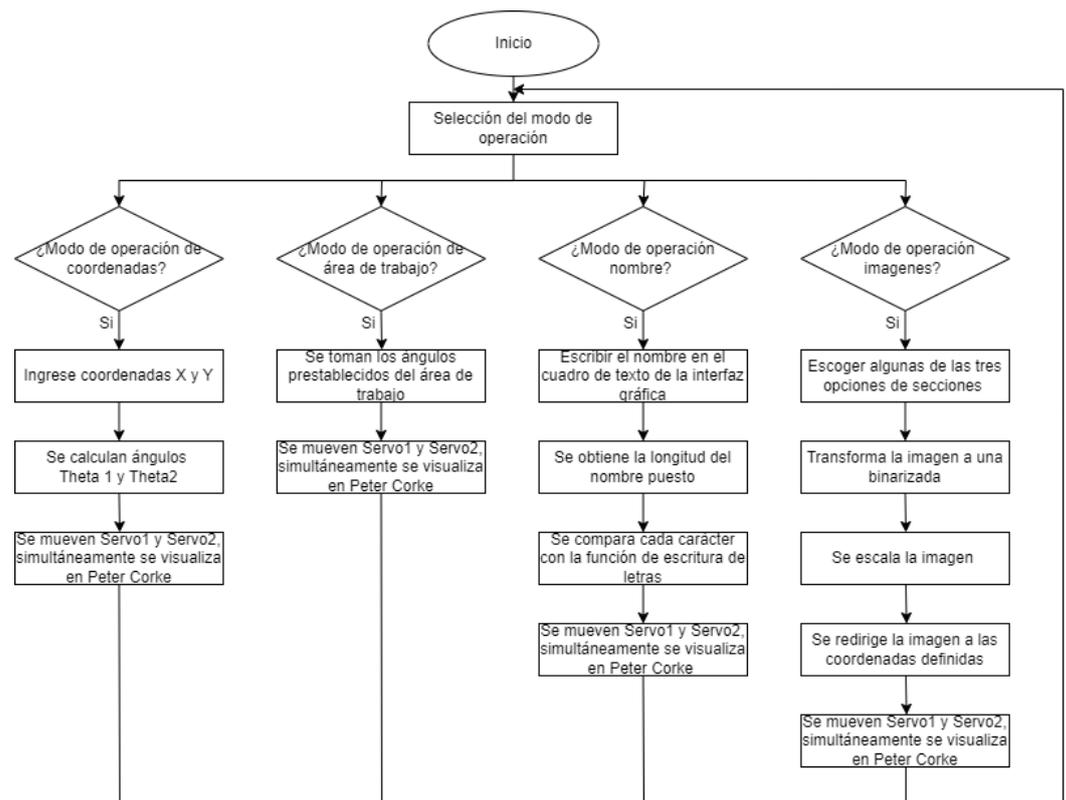
Para confirmar y visualizar con precisión los ángulos que se envían y reciben por comunicación serial, se utiliza una pantalla LCD 16x2 como se muestra en la Figura 10. La cual permite verificar los ángulos a los que se moverán las articulaciones del robot de manera efectiva.



**Figura 10.** Visualización de trama de datos

### 2.6. Diagramas de Flujo

En la Figura 11, se presenta el diagrama de flujo generalizado de la programación en Matlab, que logra resumir de manera concisa cada uno de sus cuatro modos de operación.



**Figura 11.** Diagrama de flujo modos de operación del Robot

Se muestra la codificación de recepción serial entre el Pic y Matlab, véase Figura 12.

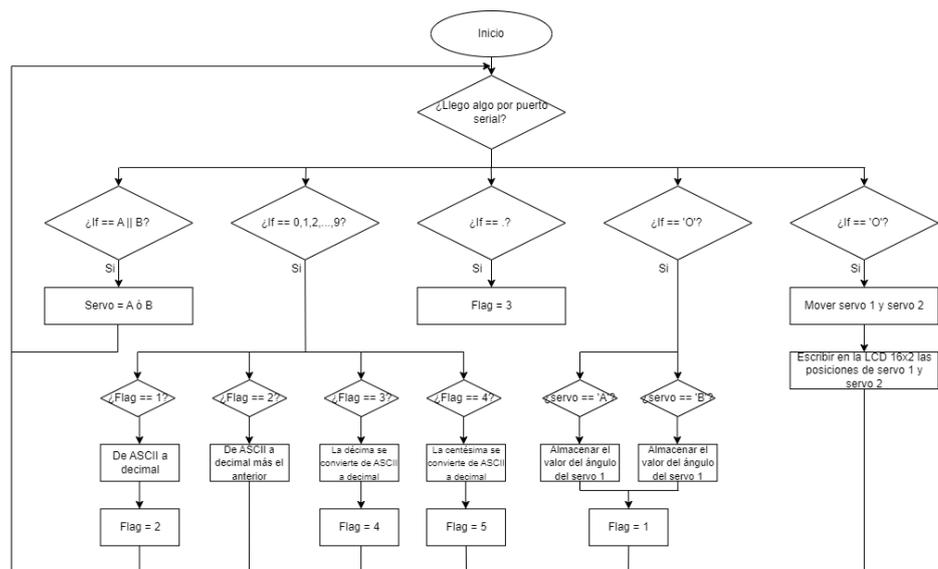


Figura 12. Recepción de datos serial Pic - Matlab

### 3. Resultados

La operación del robot 2R se desglosa en cuatro modos distintos: (1) Modo Manual de Coordenadas, (2) Modo de Área de Trabajo, (3) Modo Nombre y (4) Modo Selección de imagen previamente establecida, estas opciones son seleccionadas por el usuario a través de la interfaz gráfica proporcionada por la herramienta GUIDE de MATLAB. Además de la elección del modo, la interfaz permite simular la posición del robot 2R en su área de trabajo.

En la Figura 13 se presenta el resultado en el montaje físico de la integración de los componentes detallados en la Figura 1, en la que se ilustra el trazo del contorno de un logo previamente establecido en la interfaz gráfica.

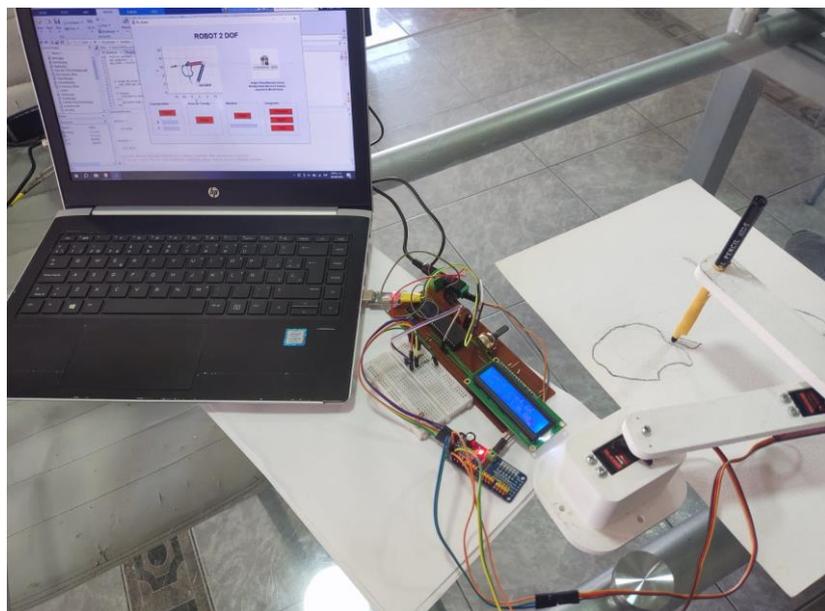


Figura 13. Logo Apple con el PIC 18F46K22 y la LCD 16X2

En el primer modo de funcionamiento, se ejecuta la trayectoria del robot 2R hasta alcanzar las coordenadas X y Y designadas por el usuario, un ejemplo es presentado en la Figura 14, donde se exhibe la trayectoria de tipo MoveJ (interpolación de ángulos) realizada por el robot 2R tanto en su simulación como en su configuración física.

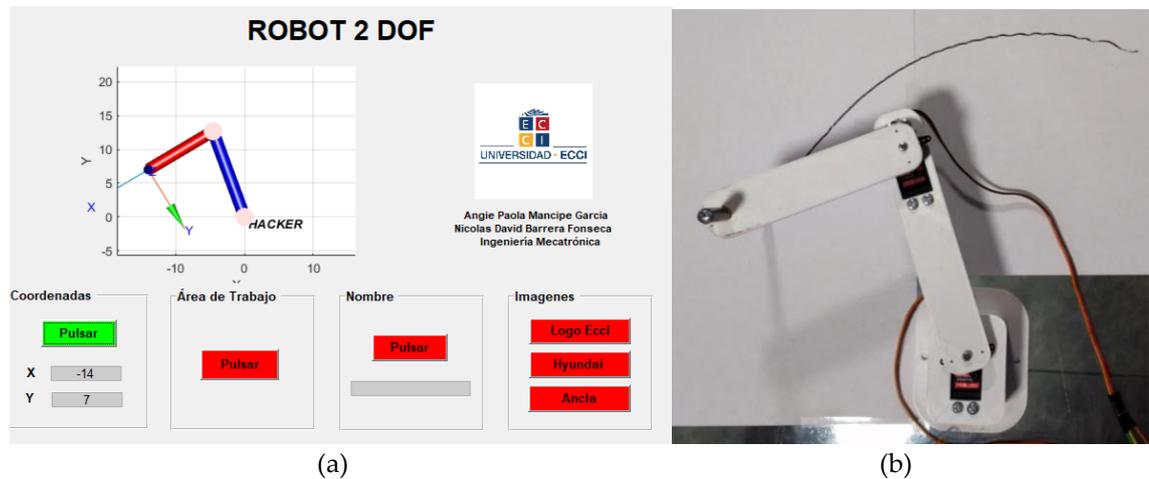


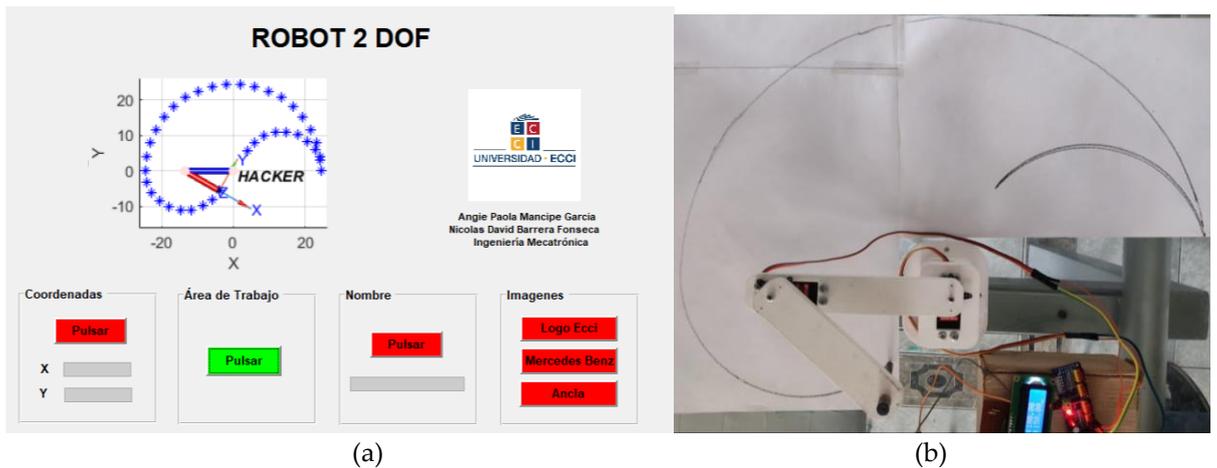
Figura 14. Trayectoria en modo de operación coordenadas (a) Simulación; (b) Montaje

Cuando el usuario ingresa unas coordenadas X y Y que no se encuentran dentro del área de trabajo del robot, la interfaz muestra un mensaje de error (ver Figura 15) donde le explica al usuario que las coordenadas especificadas están fuera de los límites del espacio de trabajo del robot 2R y que deben ser modificadas para garantizar una operación correcta.



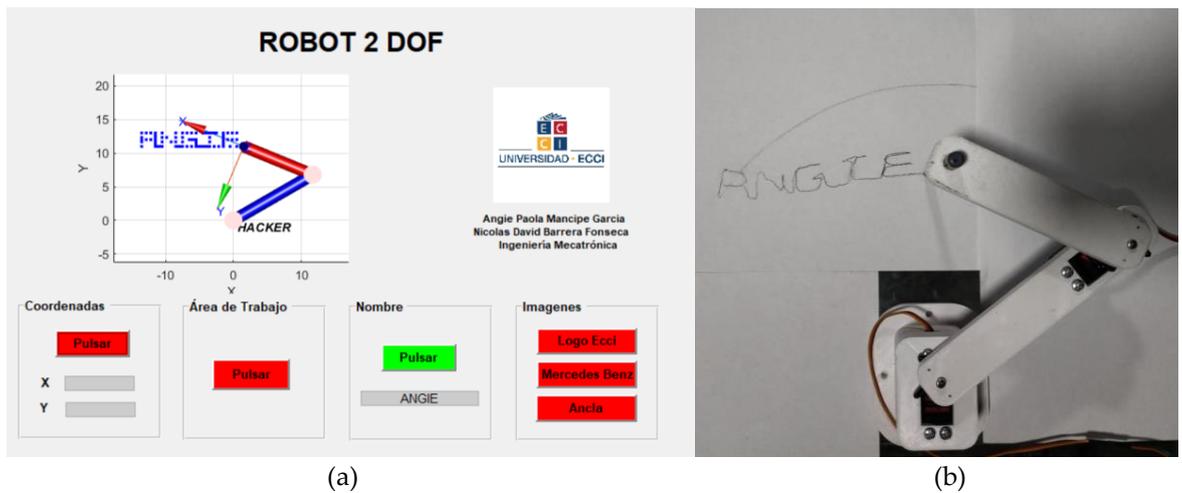
Figura 15. Mensaje de Error

En el segundo modo de operación, se lleva a cabo la definición del área de trabajo del robot tanto en la simulación como en el ensamblaje físico, lo cual es presentado en la Figura 16. En este proceso, se consideran los rangos de movimiento rotacional de  $0^\circ$  a  $180^\circ$  de cada uno de los servomotores.



**Figura 16.** Trayectoria en modo de operación área de trabajo (a) Simulación; (b) Montaje.

En el tercer modo de operación se efectúa la generación de una trayectoria correspondiente a una cadena de caracteres (Ej. nombre) ingresada por el usuario a través de la interfaz gráfica, lo cual es presentado en la Figura 16 en la que se detalla con claridad cómo el robot 2R es capaz de escribir el string, tanto en la simulación como en el montaje físico. Estas imágenes resaltan la capacidad del robot para realizar esta tarea específica, demostrando así su funcionalidad en la escritura de caracteres alfanuméricos.

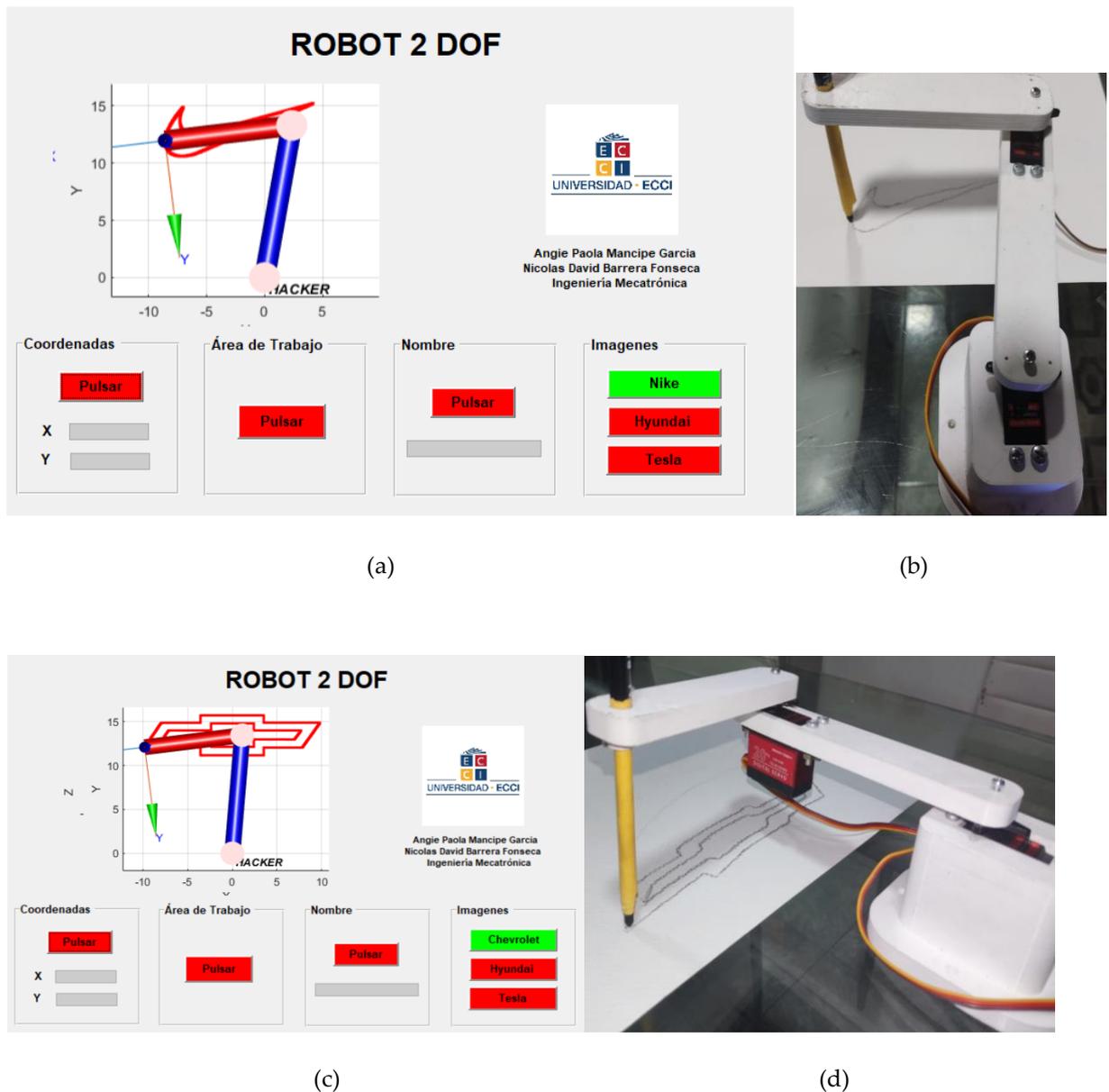


**Figura 16.** Trayectoria en modo de operación nombre (a) Simulación; (b) Montaje.

Es importante resaltar que en la Figura 16, sección b, se pueden observar ciertas imperfecciones en la escritura del nombre "Angie". Estas posibles deficiencias podrían ser atribuidas a la falta de puntos de caracterización, lo cual afecta la precisión en la ejecución de los trazos. Además, el ajuste mecánico del lápiz también podría estar contribuyendo a estas inconsistencias en el trazado.

En el cuarto y último modo de operación se genera una representación gráfica de una imagen previamente elegida, como logotipos o ilustraciones. A estas imágenes se les realiza el procesamiento descrito en la sección 2.2, mediante la programación en MATLAB; este procesamiento es necesario para que los contornos puedan ser caracterizados en sus coordenadas X y Y, y posteriormente dibujados en alguno de los tres segmentos del área de trabajo (horizontal, cuadrado, vertical) del robot 2R presentados en la Figura 3.

La Figura 17 muestra una representación gráfica trazada en el segmento horizontal. Se presentan dos ejemplos en la figura 14: a) y b) corresponden al logo de Nike en simulación y montaje físico respectivamente, y c) y d) muestran el logo de Chevrolet en simulación y montaje físico respectivamente.



**Figura 17.** Trayectoria en modo de operación imagen horizontal (a) Simulación Nike; (b) Montaje Nike; (c) Simulación Chevrolet; (d) Montaje Chevrolet.

La Figura 18 muestra ejemplos de representaciones gráficas trazadas en el segmento cuadrado. En la sección a) y b) se presenta el logo de KIA en simulación y montaje físico, respectivamente. La sección c) y d) muestra el logo de Apple de manera similar, primero en simulación y luego en el montaje físico del robot.

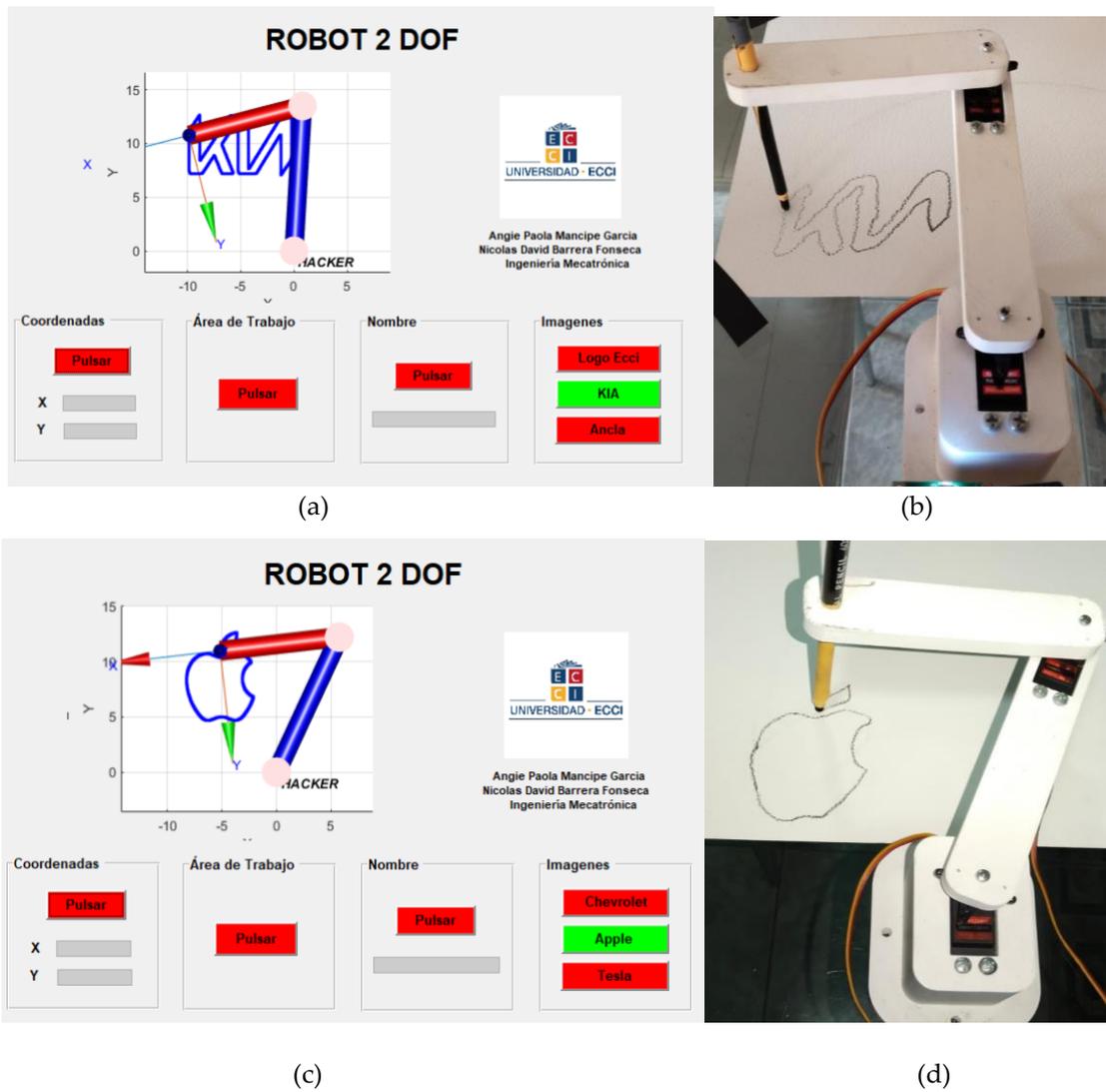
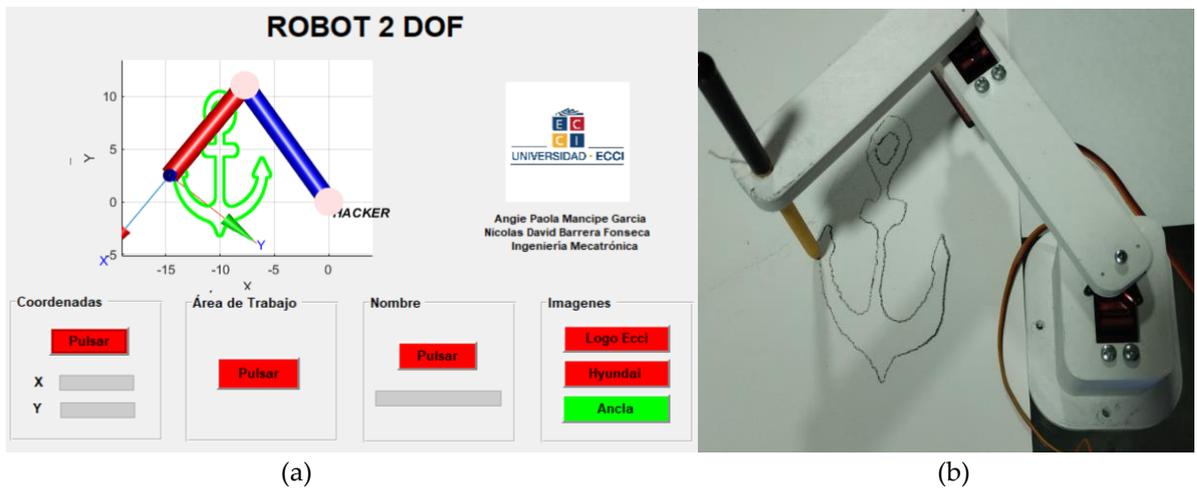
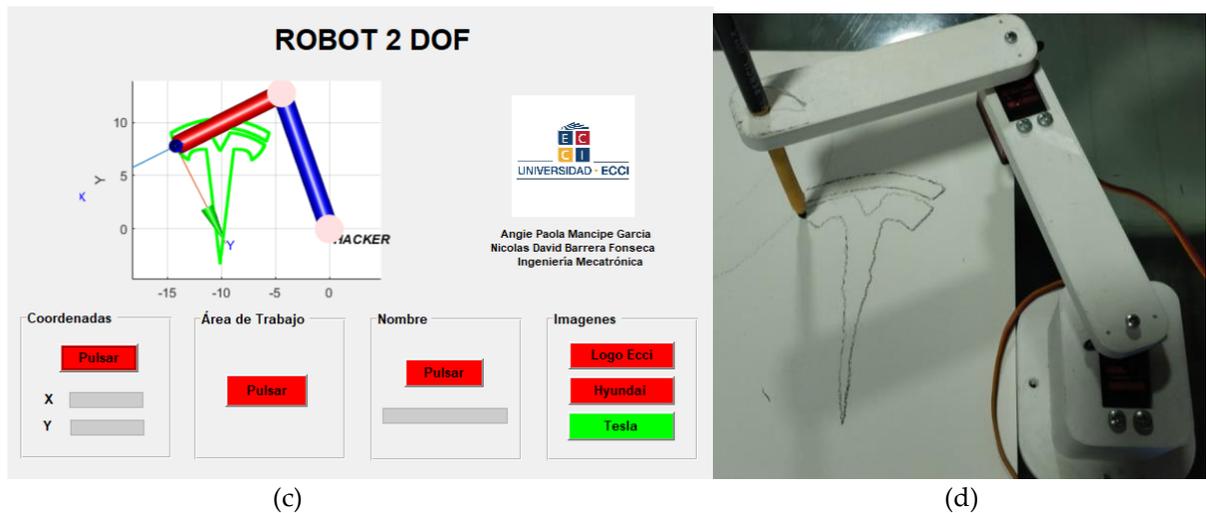


Figura 18. Trayectoria en modo de operación imagen Apple (a) Simulación Kia; (b) Montaje Kia; (c) Simulación Apple; (d) Montaje Apple.

La Figura 19 ilustra ejemplos de representaciones gráficas trazadas en el segmento vertical. En la sección a) y b) se muestra la imagen de un ancla en simulación y montaje físico, respectivamente. La sección c) y d) presenta el logo de Tesla en el mismo formato, primero en simulación y luego en el montaje físico del robot.





**Figura 19.** Trayectoria en modo de operación imagen (a) Simulación Ancla; (b) Montaje Ancla (c) Simulación Tesla; (d) Montaje Tesla.

En las distintas figuras previamente presentadas se evidencian diversas limitaciones que podrían estar relacionadas con la configuración del Efecto Final (en este caso, el lápiz). Estas limitaciones se refieren principalmente a la precisión del ajuste mecánico del lápiz. Además, es importante considerar que la calidad del trazado puede estar sujeta a la exactitud del control de los servomotores mediante la técnica de modulación por ancho de pulso (PWM). Adicionalmente, la transmisión secuencial de los ángulos desde el microcontrolador PIC hasta la PCA9685 la cual maneja los dos servomotores.

#### 4. Discusión

Los resultados obtenidos en este estudio representan un avance significativo en la integración de la robótica y el procesamiento de imágenes, reafirmando las hipótesis planteadas al inicio de la investigación. La creación de un robot 2R controlado por el microcontrolador PIC18F46K22 y programado con MATLAB, combinado con la capacidad de procesar imágenes, ha demostrado ser una combinación interesante con implicaciones valiosas.

Desde la perspectiva de estudios previos, la interacción entre la robótica y el procesamiento de imágenes ha sido explorada en diferentes contextos. Sin embargo, este estudio aporta un enfoque específico en la escritura de caracteres alfabéticos y la representación gráfica a través de un robot 2R. Los hallazgos confirman que el procesamiento de imágenes expande las capacidades del robot, permitiéndole interactuar con su entorno visual de manera efectiva y ejecutar tareas precisas y complejas, como la escritura y el trazado de imágenes.

Esta convergencia tecnológica abre prometedoras oportunidades en áreas como la automatización industrial y la educación. En el ámbito industrial, la combinación de robótica y procesamiento de imágenes ofrece la posibilidad de lograr una mayor eficiencia en las tareas de manufactura y control de calidad. En la educación, brinda una plataforma interactiva para enseñar conceptos complejos y promover el aprendizaje práctico en campos relacionados con la robótica y la programación.

A medida que continuamos explorando las diferentes posibilidades, se vuelve evidente que ofrece un camino hacia la creación de soluciones más adaptables e inteligentes en diversas áreas de aplicación. El enfoque en la combinación de componentes mecánicos y procesamiento de imágenes resalta la importancia de considerar ambos aspectos en el diseño y desarrollo de soluciones robóticas avanzadas.

En cuanto a direcciones futuras de investigación, se pretende optimizar los algoritmos de procesamiento de imágenes para lograr trazados y escritura más precisos. Además, se explorarán enfoques para superar las limitaciones que se han observado en la ejecución secuencial de los movimientos del robot y en el ajuste mecánico del efector final. Se considera la posibilidad de emplear microcontroladores más avanzados, como Raspberry Pi, con el fin de mejorar el control y la sincronización del robot. Estas acciones buscan potenciar la eficacia y las capacidades de la tecnología en el futuro.

## 5. Conclusión

El robot 2R controlado a través de un microcontrolador de bajo costo (PIC 18F46K22) y MATLAB para el desarrollo de la interfaz gráfica, de las cinemáticas directa e inversa y del procesamiento de imágenes, demostró ser una solución eficaz y versátil para aplicaciones de expresión gráfica y diseño, sin embargo la calidad de los trazos de los contornos de las imágenes presentan un error, el cual se debe principalmente a tres razones: 1. el ajuste mecánico del lápiz al segundo eslabón, 2. la precisión del control de los servomotores por PWM, para lo cual se debería desarrollar un control digital en el PIC con motores DC para controlar la posición y 3. el envío de los ángulos al primer y segundo servomotor desde el PIC a la PCA9685 ya que la ejecución del algoritmo es secuencial, para lo cual se debería implementar una frecuencia mayor de oscilación o implementar una arquitectura de ejecución en paralelo (FPGAs o GPUs). Así mismo, es importante mencionar que el procesamiento computacional realizado por MATLAB es considerable, lo que da lugar a tiempos de espera significativos en la comunicación serial UART, lo que representa demoras considerables en el trazo de un logo, así como el número significativo de coordenadas X y Y.

**Fondos:** Agregue: Esta investigación no recibió financiamiento externo.

**Conflictos de interés:** Los autores declaran no tener ningún conflicto de intereses.

## Referencias

1. Aspectos Básicos de la industria 4.0 - MINTIC. Disponible en: [https://colombiatic.mintic.gov.co/679/articles-124767\\_recurso\\_1.pdf](https://colombiatic.mintic.gov.co/679/articles-124767_recurso_1.pdf) (consultado 17-07-2023)
2. A. D. Shakibjoo and M. D. Shakibjoo, "2-DOF PID with reset controller for 4-DOF robot arm manipulator," 2015 International Conference on Advanced Robotics and Intelligent Systems (ARIS), 2015, pp. 1-6, doi: 10.1109/ARIS.2015.7158355.
3. R. K. Megalingam, S. Boddupalli and K. G. S. Apuroop, "Robotic arm control through mimicking of miniature robotic arm," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-7, doi: 10.1109/ICACCS.2017.8014622.
4. Procesamiento de Imágenes Con Matlab - US. Disponible en: [https://personal.us.es/majiro/2014\\_10\\_10\\_clase2.pdf](https://personal.us.es/majiro/2014_10_10_clase2.pdf) (consultado 17-07-2023).
5. Cordoba, M.L. Control de Movimiento y Planificación de UN Robot Esférico. Disponible en: <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/2032/1/LunaCM.pdf> (consultado 24-07-2023).
6. (PIC18F46K22) Microchip technology. Disponible en: <https://ww1.microchip.com/downloads/en/DeviceDoc/40001412G.pdf> (consultado 24-07-2023)
7. Pérez López, C. (2014). Introducing MATLAB and the MATLAB Working Environment. MATLAB Differential Equations, 1–31. doi:10.1007/978-1-4842-0310-1\_1 (consultado 24-07-2023) .
8. GUÍA RÁPIDA DE MPLAB RIUBU Principal Disponible en: [https://riubu.ubu.es/bitstream/handle/10259/3560/Crear\\_un\\_proyecto\\_en\\_MPLAB\\_X%20V2.01.pdf;jsessionid=307303AE6A57F8DEB87EEA16FA869809?sequence=1](https://riubu.ubu.es/bitstream/handle/10259/3560/Crear_un_proyecto_en_MPLAB_X%20V2.01.pdf;jsessionid=307303AE6A57F8DEB87EEA16FA869809?sequence=1) (consultado 25-07-2023)
9. Corke, P. I. (1996). A robotics toolbox for MATLAB. IEEE Robotics and Automatío Magazine, 3(1), 24–32. <https://doi.org/10.1109/100.486658>
10. B. R. Hunt, R. L. Lipsman y J. M. Rosenberg. "A Guide to MATLAB for Beginners and Experienced Users". Cambridge. Disponible en: <http://www.uop.edu.pk/ocontents/A%20Guide%20to%20MATALB.pdf> (consultado 25-07-2023)

11. F. I. Tiberiu Petrescu y R. V. Petrescu. "Direct and inverse kinematics to the anthropomorphic robots". Engevista. Disponible en: [https://www.researchgate.net/publication/305701095\\_Direct\\_and\\_inverse\\_kinematics\\_to\\_the\\_anthropomorphic\\_robots](https://www.researchgate.net/publication/305701095_Direct_and_inverse_kinematics_to_the_anthropomorphic_robots) (consultado 25-07-2023)
12. A. D. Joya Barón, J. A. Martínez Guevara y D. A. Pardo Murcia. "Metodo DH en MATLAB Robot Fanuc". Stuvia. Disponible en: <https://www.stuvia.com/doc/570269/metodo-dh-en-matlab-robot-fanuc> (consultado 27-07-2023).
13. Díaz Mulas, B. (2015) UART: Universal asynchronous receiver-transmitter, e. Disponible en: <https://e-archivo.uc3m.es/handle/10016/23730> (consultado 27-07-2023)
14. Ripoll Ramzi, C. (1970). Disponible en: <https://repositorio.comillas.edu/xmlui/handle/11531/22910> (consultado 28-07-2023)
15. Comunicación - Bus I2C - Bolanosdj.com.ar. Disponible en: <https://bolanosdj.com.ar/MOVIL/ARDUINO2/ComunicacionBusI2C.pdf> (consultado 28-07-2023)
16. Introducción a SOLIDWORKS - official Solidworks Community. Disponible en: [https://my.solidworks.com/solidworks/guide/SOLIDWORKS\\_Introduction\\_ES.pdf](https://my.solidworks.com/solidworks/guide/SOLIDWORKS_Introduction_ES.pdf) (consultado 28-07-2023).
17. *Robot de 2 DOF programado con MatLab® (guide y Peter Corke) y Arduino uno para escritura de caracteres alfabéticos* XVII CONGRESO INTERNACIONAL DE ELECTRÓNICA CONTROL Y TELECOMUNICACIONES. pp. 793-814 Disponible en: [https://www.dropbox.com/s/mazilw4a6f5vo68/MEMORIAS\\_CIECT\\_XVII\\_FINAL.pdf?dl=0](https://www.dropbox.com/s/mazilw4a6f5vo68/MEMORIAS_CIECT_XVII_FINAL.pdf?dl=0) (consultado 28-07-2023)