Desarrollo de un robot cilíndrico didáctico

Jose Ernesto Linares Chavez
Universidad Catolica Boliviana "San Pablo"
jose.linares@ucb.edu.bo

Mauricio Lopez Garcia
Universidad Catolica Boliviana "San Pablo"
mauricio.lopez@ucb.edu.bo

Msc. Jose Jesus Cabrera Pantoja
Universidad Catolica Boliviana "San Pablo"
Docente - Ingeniería Mecatrónica
jcabrera@ucb.edu.bo

***Resumen*–Este artículo presenta un robot cilíndrico de 4 grados de libertad, diseñado con un enfoque didáctico. Se propone utilizar un microcontrolador Arduino como procesador central del robot. La mayor parte de la estructura del robot será fabricada mediante impresión 3D. A partir de este sistema, se derivarán las transformadas homogéneas y se desarrollará la cinemática inversa, lo que permitirá el control preciso del robot y la realización de diversas pruebas experimentales.**

Palabras claves - .Arduino, Impresión 3D, Robot cilíndrico, 4 grados de libertad

I. INTRODUCCIÓN

El campo de la robótica ha avanzado significativamente en las últimas décadas, optimizando procesos y reduciendo costos en la automatización industrial. En este contexto, los robots cilíndricos destacan por su estructura sencilla y capacidad para realizar tareas repetitivas con precisión en espacios reducidos. Estas características los hacen ideales para operaciones de soldadura, ensamblaje y manipulación de materiales.

La robótica industrial se ha vuelto fundamental en sectores como la automoción, plásticos, electrónica y bienes de consumo, donde se requiere precisión y repetitividad para optimizar la producción. Los robots industriales, especialmente los de brazo articulado, pueden llevar a cabo tareas de alta precisión, como soldadura y ensamblaje, de forma eficiente, ayudando a reducir costos de operación y mejorar la calidad del producto final [1].

Según el documento desarrollado por Gautam [1] donde exploran distintos aspectos de los brazos robóticos, desde el diseño estructural hasta los sistemas de control basados en haptics o sensores de movimiento. Los estudios recientes han implementado el uso de motores paso a paso y materiales livianos como el aluminio y la fibra de carbono, buscando mejorar la durabilidad y reducir el consumo energético en brazos robóticos. Estas innovaciones muestran avances en la creación de brazos robóticos más eficientes y menos costosos.

Varios estudios han explorado algoritmos de control basados en inteligencia artificial que simplifican la planificación de trayectorias para robots industriales en espacios de búsqueda cilíndricos. Uno de los enfoques recientes propone el uso de una cuadrícula cilíndrica de cuatro dimensiones que almacena las posiciones y orientaciones del robot, facilitando el control de movimientos sin la necesidad de cálculos cinemáticos complejos. Este sistema ha demostrado ser efectivo para mejorar el tiempo de respuesta en entornos industriales, al optimizar la búsqueda de rutas libres de colisiones [2][3][4].

Los *papers* sobre el modelado y simulación de brazos robóticos de múltiples grados de libertad explora métodos como la cinemática directa e inversa para optimizar el control del movimiento. Estudios previos han desarrollado modelos matemáticos para el análisis cinemático de estos sistemas, utilizando enfoques como la matriz de transformación homogénea para calcular la posición y orientación del efector final. Estas investigaciones permiten desarrollar sistemas de control precisos y eficientes aplicables tanto en entornos educativos como industriales [5].

El uso de estos robots permiten a los estudiantes comprender los principios de la cinemática y el control de manipuladores. Estos robots, al simular movimientos similares a los de un brazo humano, representan una herramienta eficaz para la enseñanza de conceptos teóricos y prácticos en robótica, al tiempo que permiten experimentar en un entorno seguro y controlado sin riesgo para los estudiantes [5].

El objetivo de este trabajo es desarrollar un brazo robótico de cuatro grados de libertad funcional utilizando todo lo visto en la materia. Utilizando Python y Arduino, el sistema permite una retroalimentación de la posición del efector final, facilitando así la comprensión teórica y práctica de los principios de cinemática directa e inversa.

II. FUNDAMENTO TEÓRICO

A. Diagrama Cinematico

Un diagrama cinemático es una representación gráfica que ilustra la configuración de un sistema robótico, mostrando las interrelaciones entre las distintas partes del robot y sus grados de libertad. En el contexto de un robot cilíndrico de 4 grados de libertad, el diagrama incluye:

- Ejes de Rotación: Muestra los ejes alrededor de los cuales giran los motores, lo que es crucial para entender el movimiento.
- Esquema del Robot: Representa la estructura del robot, incluyendo el efector final, lo que permite visualizar cómo se conecta cada componente.

- Variables Articulares: Indica las variables que definen la posición de cada junta, como los ángulos y desplazamientos.

Este diagrama es fundamental para facilitar el análisis de la cinemática directa e inversa, ya que proporciona una comprensión visual de cómo se relacionan los movimientos de las diferentes partes del robot. Además, es útil para la programación y el control del movimiento, ya que ayuda a anticipar cómo los cambios en las variables articulares afectan la posición del efector final [6].

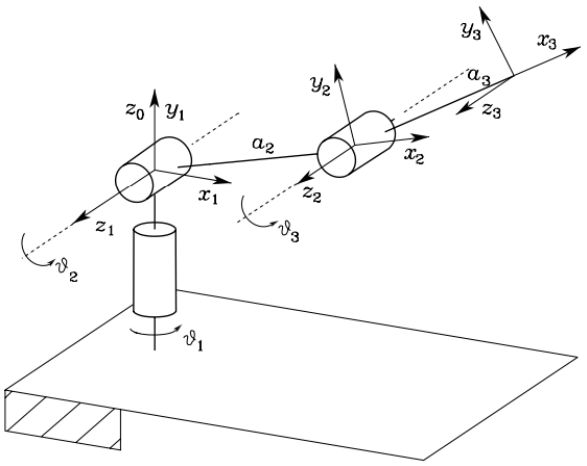


Fig 1. Diagrama cinemático de un robot antropomórfico

B. Cinemática directa

La cinemática directa se refiere al proceso de determinar la posición y orientación de un efector final de un robot, dado un conjunto específico de valores para sus variables articulares (ángulos de las juntas o desplazamientos) [7].

Este cálculo se realiza mediante la multiplicación de matrices de transformación homogénea, que combinan tanto las rotaciones como los desplazamientos de las articulaciones del robot. La transformación homogénea general para una articulación q en un robot es representada por la matriz "T", que incluye una rotación "C" y un desplazamiento "r".

$$T = \begin{bmatrix} Ci & r \\ 0 & 1 \end{bmatrix} \tag{1}$$

donde:

T Matriz de tranformada homogenea
C_i Matriz de rotación
r Vector de desplazamiento

Las rotaciones en el espacio tridimensional son fundamentales para la cinemática directa, especialmente cuando el robot realiza movimientos en un entorno tridimensional. La matriz de rotación C describe cómo cambia la orientación del efector final respecto al sistema de coordenadas base. Las rotaciones en torno a los ejes "x", "y", y "z" se describen por las siguientes matrices:

$$Cx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2}$$

$$Cy = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \sin(\theta) \end{bmatrix} \tag{3}$$

$$Cz = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

donde:

C_i Matriz de rotación
Θ Ángulo del motor (rad)

El desplazamiento de coordenadas se refiere al cambio de origen entre dos sistemas de referencia, comúnmente entre la base del robot y el efector final. Cuando se considera un sistema de coordenadas (x,y,z) en la base del robot, y un sistema de coordenadas (x′,y′,z′) en el efector final, la relación entre ambos se puede modelar mediante una matriz de transformación homogénea que incluye tanto la rotación como el desplazamiento.

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

donde:

r Vector de traslación

Para calcular la posición y orientación del efector final (E) del robot en relación con un sistema de referencia inicial (I). Cada matriz Ti-1, i define la transformación desde el sistema de coordenadas de la articulación "i-1" hasta el sistema de coordenadas de la articulación "i", incorporando tanto rotaciones como desplazamientos entre ellas.

$$T_{IE} = T_{IO} \cdot \Pi_{i=1}^{n} T_{i-1,i} \cdot T_{nE} \tag{5}$$

donde:

T_{IE} Matriz de transformación de I a E
T_{I0} Matriz de transformación inicial
$T_{i-1,i}$ Matriz de transformación homogénea entre la articulación i−1 y la articulación i.
T_{nE} Matriz de transformación homogénea de la articulación final hasta el efector final.

La Ecuación (6) describe la transformación completa desde el sistema de referencia base “I” hasta el efector final “E” de un robot con n articulaciones. En esta formulación, cada matriz Ti−1, i representa la transformación homogénea entre dos articulaciones consecutivas, incluyendo tanto rotaciones como desplazamientos. El producto de todas estas transformaciones, comenzando desde la base I hasta el efector final E, permite determinar la posición y orientación del efector final en función de la configuración de cada articulación.

C. Cinemática inversa

La cinemática inversa es el proceso que busca determinar los valores de las variables articulares necesarios para alcanzar una posición y orientación deseadas del efector final. Este proceso puede ser más complejo que la cinemática directa y no siempre tiene una solución única. Dependiendo de la configuración del robot y las restricciones del entorno, pueden ser necesarias técnicas numéricas o heurísticas para encontrar las soluciones adecuadas.[8]

Hay dos formas de resolver esto mediante método cerrado los cuales son:

El método algebraico, también conocido como método analítico, utiliza herramientas algebraicas y trigonométricas para encontrar una solución directa y exacta a las ecuaciones que describen la cinemática inversa. Este método se basa en el análisis matemático de las transformaciones homogéneas que representan el movimiento del robot.

El método geométrico se basa en analizar las relaciones geométricas entre los eslabones y las articulaciones del manipulador, usando herramientas como triángulos, ángulos y longitudes. Este método se centra en el entendimiento gráfico y trigonométrico del problema.

D. Grados de libertad

Los grados de libertad (DoF) de un robot se refieren a la cantidad de movimientos independientes que puede realizar. En un robot cilíndrico con 4 grados de libertad, esto implica que puede moverse en varias direcciones, lo que le permite posicionar su efector final de manera flexible en el espacio tridimensional. Los grados de libertad son esenciales para evaluar la capacidad del robot para realizar tareas específicas y para determinar su versatilidad en diversas aplicaciones.[9]

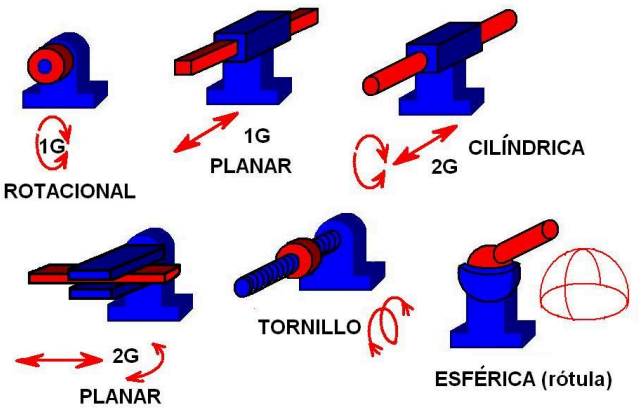


Fig 2. Grados de libertad para un robot

E. Motor paso a paso

Los motores paso a paso son dispositivos electromecánicos que convierten impulsos eléctricos en movimientos mecánicos precisos. Cada impulso provoca un giro de un número fijo de grados, lo que permite un control exacto de la posición y la velocidad. Estos motores son ampliamente utilizados en aplicaciones robóticas debido a su capacidad para realizar movimientos precisos y repetibles, lo que los hace ideales para tareas que requieren alta precisión.[10]

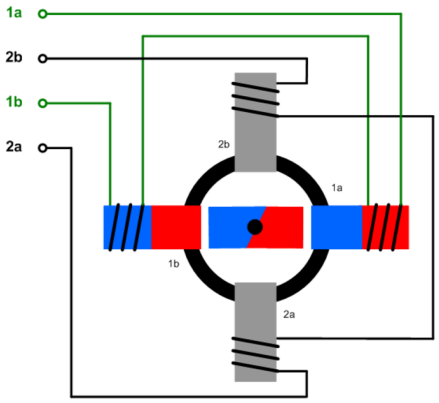


Fig 3. Motor bipolar

Otra división de los motores paso a paso resulta del método de hacer el devanado en motores de 2 fases. Dependiendo de él, los motores se dividen en unipolares y bipolares. La principal diferencia es que el motor unipolar funciona con una polaridad de corriente (voltaje), mientras que el motor bipolar funciona con dos polaridades, como se ve en la Fig 2, lo que significa que la dirección del flujo de corriente en la bobina es variable.

F. Microsteps

El control de *microsteps* es una técnica que mejora la resolución angular de los motores paso a paso al dividir cada paso completo del motor en fracciones más pequeñas. Esto se logra controlando las corrientes que pasan por las bobinas del motor para generar posiciones intermedias entre los pasos discretos definidos por el fabricante. Por ejemplo, un motor con 200 pasos por revolución (1.8° por paso) puede alcanzar una resolución de 3200 pasos por revolución al

configurarse con microsteps de 1/16. Este aumento en la resolución permite:

- Movimientos más suaves y precisos.
- Reducción de vibraciones y ruido.
- Mejora en la calidad de posicionamiento, especialmente en aplicaciones que requieren alta precisión.

El control de *microsteps* se implementa mediante el uso de drivers específicos que gestionan las corrientes en las bobinas del motor para lograr las posiciones intermedias. Sin embargo, es importante mencionar que aunque los microsteps incrementan la resolución teórica, la precisión real también depende de factores como la carga mecánica y la calidad del driver. [11]

G. Driver A4988

El *driver* A4988 es un controlador popular para motores paso a paso que soporta el control de *microsteps*. Este driver permite configurar resoluciones de 1, 1/2, 1/4, 1/8 y 1/16 de paso mediante la combinación de tres pines de configuración (MS1, MS2 y MS3). Además, el A4988 incluye funcionalidades como:

Regulación de corriente: El A4988 permite limitar la corriente que pasa por las bobinas del motor, protegiéndolo de sobrecalentamientos y optimizando su eficiencia.

La regulación de corriente se da mediante la medición de un potenciómetro integrado dada por la siguiente fórmula:

$$V_r = \frac{Lc}{0.7} 8Rc \tag{6}$$

donde:

V_r Voltaje de referencia

L_c Límite de corriente

R_c Resistencia constante interna del driver

Protecciones integradas: El *driver* incluye mecanismos de protección contra sobrecorriente, sobrecalentamiento y apagado por bajo voltaje, lo que garantiza un funcionamiento seguro y confiable.

Configuración de *microsteps*: Para configurar el nivel de microsteps, se establecen los pines MS1, MS2 y MS3 en combinaciones específicas.

Tabla 1 Tabla de resolución A4988

MS1	MS2	MS3	Resolución de *microsteps*
0	0	0	Paso completo (1)
1	0	0	1/2
1	1	0	1/4
1	1	1	1/16

El *driver* A4988 ofrece una solución eficiente y versátil para controlar motores paso a paso, permitiendo implementar *microsteps* y regular la corriente de manera sencilla. [12]

III. ESPECIFICACIONES DEL PROYECTO

El robot cilíndrico cuenta con especificaciones definidas para su desempeño. En la Tabla 1 se presentan los parámetros clave, incluyendo velocidad máxima, capacidad de carga y dimensiones principales.

Tabla 2 Parámetros del robot

Parámetro	Valor	Unidad
Velocidad máxima (Vmax)	0.01	m/s
Velocidad angular máxima (Wmax)	π/6	rad/s
Capacidad de carga (M)	0.5	kg
Longitud del brazo 1 (l1)	0.12	m
Longitud del brazo 2 (l2)	0.12	m

IV. MARCO PRÁCTICO

A. Diagrama cinemático

En el diagrama cinemático de la Figura 3, se identifican cuatro motores que permiten el movimiento del sistema. El motor q1 realiza un movimiento prismático en el eje z, mientras que los motores q2 y q3 tienen un movimiento de rotación alrededor del eje z. Finalmente, el motor Q4 ejecuta una rotación en el eje x.

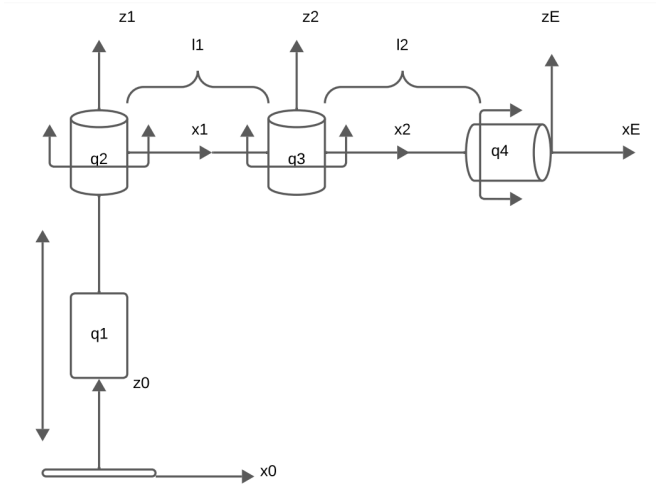


Fig 3. Diagrama cinemático del robot cilíndrico

Cuando se hace referencia a q1, q2, q3 o q4, se está hablando de los respectivos motores que generan estos movimientos en el sistema.

B. Cálculos de cinemática directa

Inicialmente, se realizan los cálculos de cinemática directa utilizando la técnica de la transformación homogénea. La ecuación que describe el proceso es la siguiente:

$$TIE = TI0 * T01 * T12 * T2E \quad (7)$$

Los valores de cada matriz están dadas por la matriz de giro y de traslación, teniendo en cuenta que TI0 es una matriz identidad de 4x4 y el valor de las otras son:

$$T01 = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 1 & q1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$T12 = \begin{bmatrix} C3 & -S3 & 0 & l1 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$T2E = \begin{bmatrix} 0 & 0 & 0 & l2 \\ 0 & C3 & -S3 & 0 \\ 0 & S3 & C3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

A través de esta fórmula, se obtiene la transformación completa mediante la concatenación de las transformaciones individuales de cada eslabón, lo que permite calcular la transformación global que describe el movimiento del sistema desde el punto de origen hasta el efector final del robot. Este enfoque asegura un modelado preciso y continuo del desplazamiento del robot en su espacio de trabajo.

$$TIE = \begin{bmatrix} C23 & -S23C4 & S4S23 & L1C2 + L2C23 \\ S23 & C4C23 & -S4C23 & L1S2 + L2S23 \\ 0 & S4 & C4 & q1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Con la misma transformada homogénea se usa para sacar la jacobiana, agarrando los primeros tres valores de la última columna.

Con estos valores se hacen derivadas parciales, donde la primera columna se divide por el primer motor y así sucesivamente.

$$Ja = \begin{bmatrix} 0 & -L1S2 - L2S23 & -L2S23 & 0 \\ 0 & L1C2 + L2C23 & L2C23 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (12)$$

Debido a que se buscan las singularidades y se tiene un matriz 3x4 la cual no es cuadrada para hallar la determinante, se opta por sacar la jacobiana sin tomar en cuenta el último motor.

Teniendo la transformada homogénea se hace lo mismo para hallar la jacobiana analítica, dando el siguiente resultado.

$$Ja = \begin{bmatrix} 0 & -L1S2 - L2S23 & -L2S23 \\ 0 & L1C2 + L2C23 & L2C23 \\ 1 & 0 & 0 \end{bmatrix} \quad (13)$$

Con los valores obtenidos de la jacobiana y siendo que ya es una matriz 3x3, o sea cuadrada, se calculo su determinante y así sacando las singularidades.

$$l_1 l_2 \, sin(q_3) = 0 \quad (14)$$

Para evitar las singularidades se tiene que evitar que la ecuación quede en 0, teniendo en cuenta que los valores de los eslabones son mayores a cero el único que coasionaria eso sería q3, entonces.

$$sin(q_3) = 0 \quad (15)$$

Los únicos valores para q3 que llegan a hacer que el valor de la ecuación 13 sean igual a 0, provocando una singularidad, son los siguientes:

$$q_3 = 0,\ \pi,\ 2\pi$$

C. Cálculos de cinemática inversa

Con la finalidad de encontrar los valores de q1, q2, q3 y q4 en base a una posición final deseada, se realiza la cinemática inversa utilizando métodos cerrados.

Partiendo del vector de traslación r_{IE} se tienen las siguientes 3 ecuaciones.

$$l_1C_2 + l_2C_{23} = x \quad (16)$$

$$l_1S_2 + l_2S_{23} = y \quad (17)$$

$$q_1 = z \quad (18)$$

De estas 3 ecuaciones la Ecuación (19) representa la relación directa que tiene q1 con el valor final en z, por lo que q1 es directamente la posición en z del efector final.

Para hallar el valor de q3, se elevan al cuadrado la Ecuación (17) y (18) para luego sumarlas. El valor resultante luego de simplificarlo se muestra en la Ecuación (20).

$$x^2 + y^2 = l_1^2 + l_1l_2C_3 + l_2^2 \quad (19)$$

Despejando la variable C_3 se obtiene la Ecuación (21).

$$C_3 = \frac{x^2+y^2-l1^2-l2^2}{2l1l2} \quad (20)$$

Como la solución de q3 impone que los resultados sean -1 < C_3 <1, por lo que es necesario hallar la solución relacionada con atan2.

Teniendo esto en cuenta se halla el valor de S3 de la siguiente manera.

$$S_3 = \pm\sqrt{1 - C_3^2} \quad (21)$$

por lo que la solución de q3 se representa con la Ecuación (23).

$$q_3 = atan2(S_3, C_3) \quad (22)$$

Con la finalidad de hallar la solución para q2, se analiza el robot desde la vista superior, como se ilustra en la figura 4.

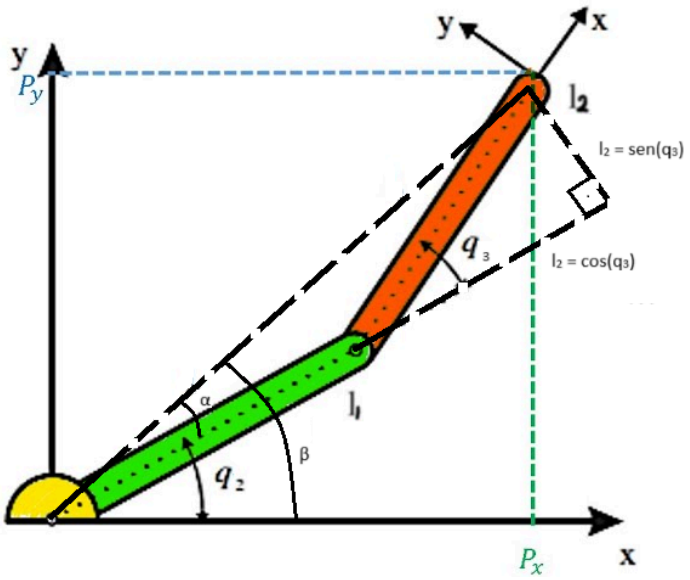


Fig 4. Vista del robot desde arriba

Tomando de referencia la Figura 4 se toma en cuenta que el q2 está dado por la resta entre el ángulo mayor o sea β menos el ángulo menor α, como se muestra en la ecuación 24.

$$q_2 = \beta - \alpha \quad (23)$$

Ya que se saben lo puntos del efector final y dibujando una recta desde el origen se puede crear un triángulo rectángulo que nos da el valor de β, matemáticamente mostrado en la ecuación 25.

$$\beta = atan2(x, y) \quad (24)$$

De un modo parecido se puede sacar el angulo α, ya que se tiene el valor de C_3 y S_3 podemos hacer un triángulo rectángulo con l2, así para tomar el valor de α simplemente se saca la tangente, teniendo en cuenta que el cateto adyacente es la suma de l1 y l2, como se ve en la ecuación 26.

$$\alpha = atan2[l_2sen(q_3),\ l_1 + l_2cos(q_3)] \quad (25)$$

De esta forma se reemplazan los valores de β y α en la ecuación 24 para poder hallar el valor de q2.

D. Torques y transmisión

Para realizar los cálculos de torque se toma en cuenta los motores que vayan a tener mayor esfuerzo, en el caso del robot cilíndrico son los primeros dos motores.

El motor q1 encargado de levantar toda la estructura y el motor q2 encargado de mover los dos eslabones del brazo.

En primer caso para el motor q1 primero se calcula el torque del tornillo sin fin con la ecuación 25.

$$t_t = (\frac{Fd}{2})(\frac{L+\pi\mu d}{\pi d-\mu L}) \quad (26)$$

Donde:

- F: Fuerza aplicada sobre la tuerca. (N)
- d: Diámetro de paso de las cuerdas. (m)
- L: Avance de la cuerda. (m)
- μ: Coeficiente de fricción entre la tuerca y las cuerdas.

Usando la anterior ecuación con los datos preestablecidos del robot nos da como el valor del torque del tornillo de 0.0102Nm.

$$t_1 = (\frac{Fd}{2})(\frac{L+\pi\mu d}{\pi d-\mu L}) \quad (27)$$

Para el motor q2 se tiene que calcular la inercia de la estructura, en este caso se toma el peor caso que es el brazo lo más estirados posible.

En la ecuación 26 se calcula la inercia de los brazos tanto con las masas de cada brazo que son de 25g y su longitud, pero además de eso se le añade una masa la cual es la que carga el brazo del robot.

$$I = m_1 l2^2 + m_2 l2^2 + m_l \quad (28)$$

la ecuación anterior si bien no da la inercia necesaria para el brazo del robot pero a este se le tiene que añadir un porcentaje de seguridad, en este caso es del 1.5%, que nos da el valor de 0.51 kg/m^2.

Teniendo la inercia obtenida anteriormente se calcula el torque necesario del motor q_2 según la ecuación 27.

$$t_2 = I\, m_{max} \quad (29)$$

El torque necesario es de 0.27kg/m^2 que llevados a kg/cm nos da un valor de 2.72kg/cm.

A través de pruebas, se determinó que por cada vuelta completa que realiza el motor Q1, la base móvil del robot se desplaza 2 mm.

Para el análisis de la primera parte del brazo, o sea de la l1, se emplea una correa dentada de paso 20mm, conectado a un eje de entrada con un diámetro de 16mm a un eje de salida con diámetro de 37mm. La relacion de transmision R1 se calcula como el cociente entre los diámetros de los ejes:

$$R1 = D_S / D_e \quad (30)$$

Donde:

- D_s es diámetro de salida
- D_e es diámetro de entrada

Esto indica que el eje de salida gira aproximadamente 2.31 veces más lento que el eje de entrada, considerando una transmisión sin deslizamiento.

En el segundo eslabón del sistema, la transmisión se realiza mediante un eje de entrada con un diámetro de 16mm y dos ejes de salida, ambos con un diámetro de 37mm. Debido a que los dos ejes de salida están interconectados y comparten el mismo diámetro, su relación de transmisión relativa es igual a 1. Esto implica que ambos ejes de salida giran a la misma velocidad angular.

Por lo tanto, el eje de entrada transmite una velocidad angular reducida en un factor de aproximadamente 2.31 a ambos ejes de salida, manteniendo la consistencia en el diseño del sistema de transmisión.

E. Diseño 3D

El diseño replicado de la página Thingiverse corresponde a un robot cilíndrico con 3 grados de libertad. Este modelo permite realizar 2 movimientos rotacionales y 1 movimiento prismático.

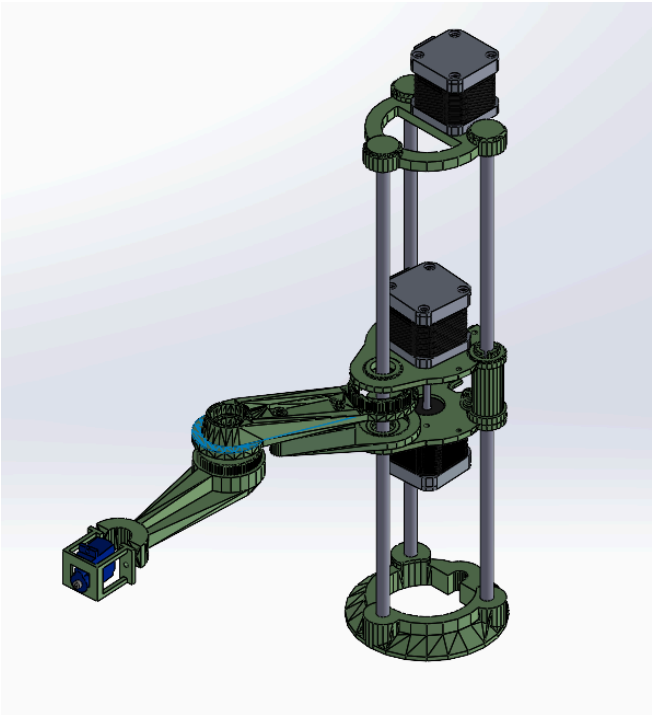


Fig 5. Robot cilíndrico conseguido de Thingiverse

Dado que el modelo original con tres grados de libertad no cumplía con los requisitos del proyecto, se modificó la parte final para añadir un cuarto grado de libertad mediante un

motor extra en la muñeca. Con esta adaptación, el diseño ahora cumple con las especificaciones establecidas.

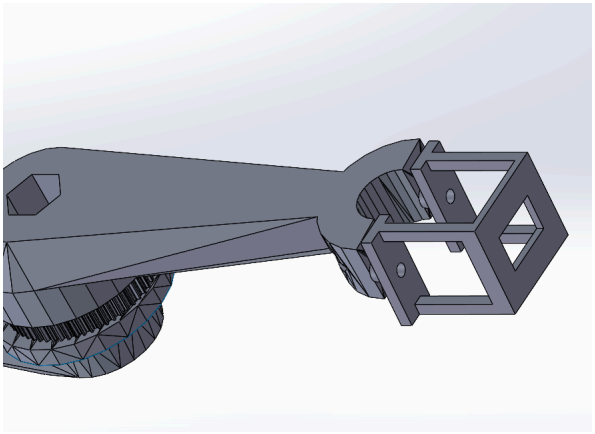


Fig 6. Modificación para el cuarto grado de libertad

F. Espacio de trabajo

El espacio de trabajo de un robot se refiere al volumen tridimensional en el que puede operar y realizar tareas. Este concepto es fundamental para entender las limitaciones y capacidades del robot en su entorno de trabajo.

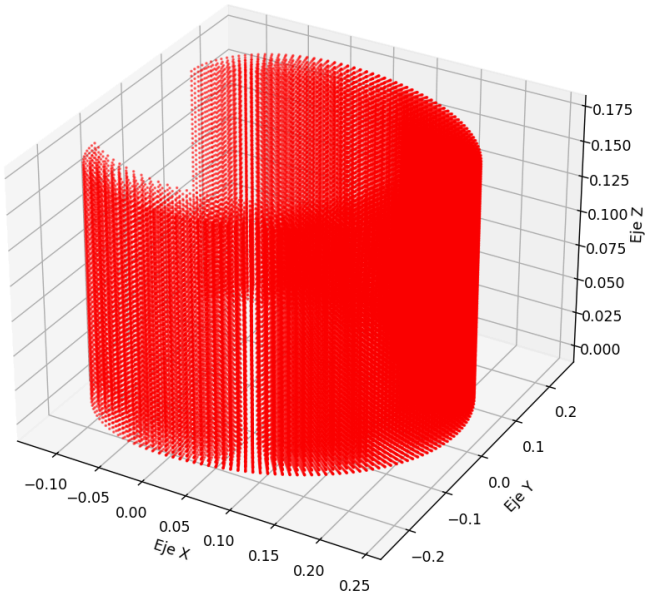


Fig 7. Espacio de trabajo en 3D

En el caso del robot, este se mueve de forma prismática a lo largo del eje Z, como se ve en la Fig 7, lo que le permite realizar desplazamientos verticales hasta 17 cm.

Además, cuenta con dos articulaciones rotativas en el mismo eje, lo que amplía su rango de movimiento y le permite alcanzar diferentes posiciones y orientaciones.

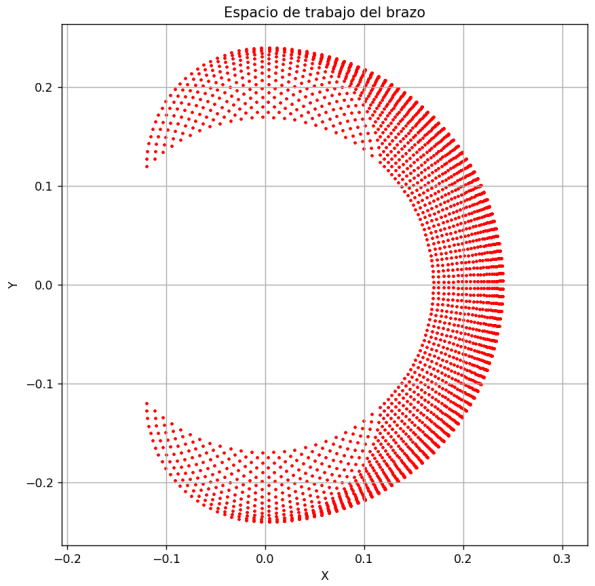


Fig 8. Espacio de trabajo visto de arriba

En la Fig 8, se ve el desplazamiento de los brazos tomando en cuenta que el desplazamiento de ambos solo es 180 grados por limitaciones del robot.

G. Selección de componentes

Para la construcción del robot se usaron los siguientes componentes mostrados en la tabla 2.

Tabla 2 Selección de componentes

Componente	*Cantidad*
PLA 1kg	*1*
LM8UU rodamiento	*4*
608-ZZ rodamiento (8x22x7)	*2*
6002-ZZ rodamiento (15x32x9)	*3*
Linear bearing rod	*3*
M3 bolt 8mm	*10*
M3 bolt 30mm	*2*
M8 bolt 40mm	*1*
GT2 pulley 20 teeth	*2*
GT2 belt 320mm 160 teeth	*1*
GT2 belt 160mm 80 teeth	*2*
Threaded rod M8	*1*
NEMA17 Stepper Motor 0.7A 14N.cm	*3*

Micro Servo MG90S 2.5Kg	*1*
Driver A4988	*3*
CNC Shield V3	*1*

En la tabla se menciona desde rodamientos, tornillos, correas y los actuadores del robot, todos estos igual de necesarios para realizar el robot cilíndrico.

G. Diseño de software

En el desarrollo del software para el movimiento experimental del robot cilíndrico, se utilizó Arduino junto con la librería AccelStepper, que permitió configurar y controlar los motores de manera eficiente. Dado que se empleó la CNC Shield v3.

Como se puede ver en la Figura 20 del anexo, se configuraron los pines de dirección y paso del motor que ya están predeterminados en la placa: 2 y 5 para el eje q1, 3 y 6 para el eje q2, y 4 y 7 para el eje q3. Esta configuración simplifica la asignación de puertos en Arduino.

En el apartado void setup, se establece la configuración inicial de los motores, definiendo los valores máximos y mínimos para sus desplazamientos. Esto asegura que cada motor opere dentro de los límites físicos y funcionales del sistema.

Finalmente, en el apartado void loop, se ejecuta la función motor.run() para cada uno de los motores. Esto permite que los motores estén en un estado constante de espera para recibir comandos de movimiento, asegurando una respuesta inmediata y fluida a las instrucciones enviadas durante la operación del robot cilíndrico.

V. RESULTADOS

Para validar los resultados obtenidos mediante la cinemática inversa, se realizó una simulación utilizando MATLAB y la biblioteca Robotics Toolbox. Esta herramienta permite modelar, simular y analizar robots de manera eficiente, proporcionando un entorno versátil para la creación y simulación de sistemas robóticos personalizados.

En primer lugar, se modeló el robot utilizando las librerías disponibles en Robotics Toolbox. Como se muestra en la Figura 23, se definieron los parámetros principales del robot, incluyendo la longitud de los eslabones y la altura máxima. Además, se añadieron 5 cm adicionales a la altura para facilitar una visualización clara y adecuada del modelo.

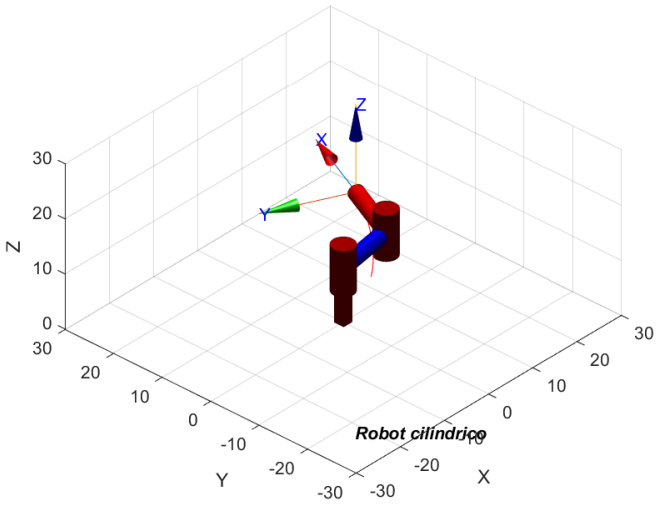


Fig 9. Robot cilíndrico visto en matlab

Con el robot cilíndrico modelado en MATLAB, como se muestra en la Figura. 9, se implementan las ecuaciones obtenidas mediante la cinemática inversa para calcular los valores de q. Para ello, se desarrolló la función correspondiente que incorpora dichas ecuaciones.

La función mostrada en la Figura 24, del anexo, se utiliza posteriormente para simular el movimiento del robot, especificando unas coordenadas iniciales y las coordenadas de destino.

En la Figura 25 se presentan los valores iniciales del robot, establecidos en (5, 5, 0), sin incluir la base añadida de 5 cm. El punto final deseado se define como (12, 12, 5), mientras que el valor 50 representa el número de pasos o puntos a simular en el trayecto para cada eje. Este parámetro permite ajustar la fluidez o velocidad de la simulación.

Este bloque de código mostrado en la Figura 26 simula el movimiento del robot y traza la trayectoria de su efector final. Dentro de un bucle for, se recorren todos los puntos generados en los vectores x_vals, y_vals y z_vals, que representan las posiciones deseadas del efector final en cada iteración. Para cada punto, se asignan las coordenadas actuales x, y y z, y luego se calcula la configuración de las articulaciones (q1, q2 y q3) utilizando la función de cinemática inversa. Este cálculo permite determinar cómo deben posicionarse las articulaciones del robot para alcanzar el punto objetivo.

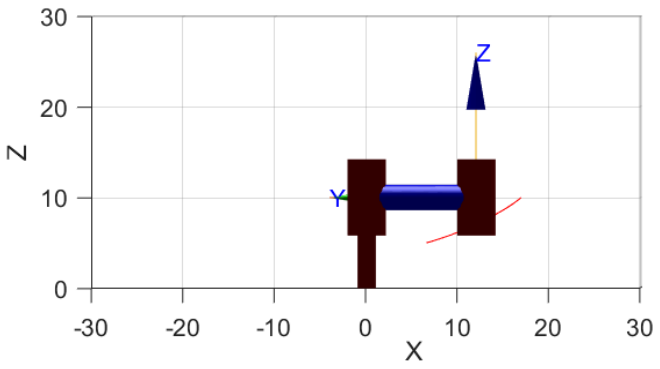


Fig 10. Altura del robot final cinemática inversa

Las coordenadas dadas fueron (12, 12, 5), y en la Fig. 16 se observa que el valor de z es 10. Esto se debe a los 5 cm añadidos de base, lo que indica que el robot alcanzó la altura deseada.

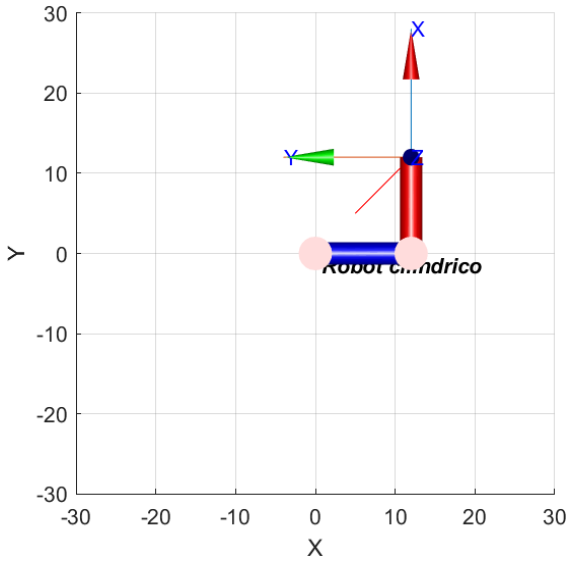


Fig 11. Coordenadas eje x y cinemática inversa

Por otro lado los valores de tanto como x y y han logrado llegar al punto final deseado, como se ve en la Figura 11.

Teniendo el robot en la posición final deseada los valores de cada una de las variables son representadas en la tabla 3:

Tabla 3 Resultados de inversa

q1	*10*
q2	*0°*
q3	*90°*

Considerando que es posible alcanzar un mismo punto en el espacio de trabajo mediante diferentes trayectorias, se llevó a cabo una simulación para explorar estas diversas formas de llegar al mismo destino.

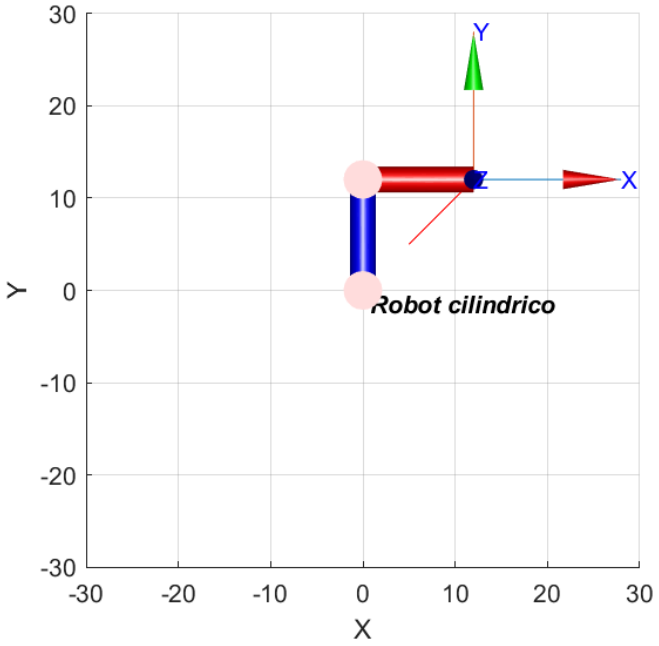


Fig 12. Coordenadas eje x y, de otra forma

Al mantener constantes las coordenadas de altura, se observa que esta no presenta variaciones significativas. La única diferencia perceptible radica en las coordenadas x e y, las cuales se ilustran en la Figura 12. Este análisis resalta cómo las variaciones en la posición horizontal impactan el alcance del robot sin alterar su elevación.

Estas mismas coordenadas fueron dadas para el prototipo en físico, resultando en una posición similar a la simulación. Este resultado se puede ver en la Figura 13 y Figura 14 .

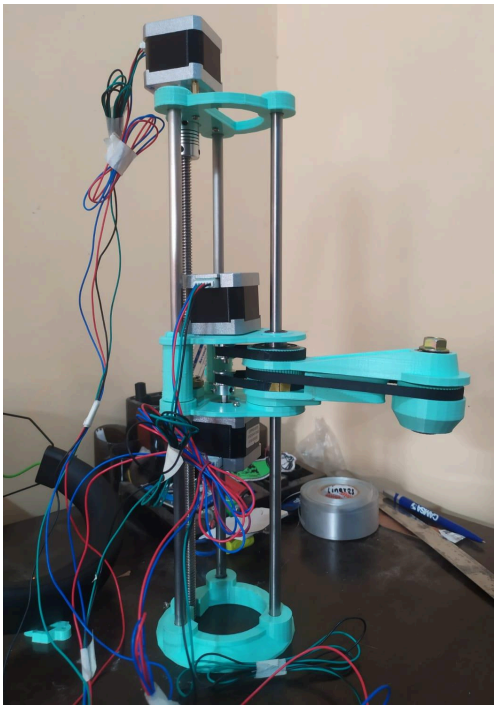


Fig 13. coordenadas alcanzadas en prueba experimental

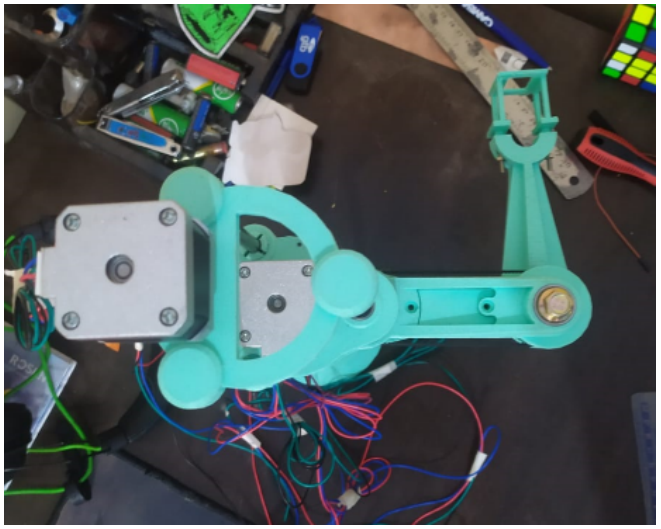


Fig 14. coordenadas alcanzadas en prueba experimental desde arriba

En este caso lo valores de cada variable son representadas en la tabla 4:

Tabla 4 Resultados de inversa de diferentes forma

q1	*10*
q2	*90°*
q3	*-90°*

En el caso de la cinemática directa, se utiliza la misma herramienta de MATLAB para realizar la simulación. Para verificar la funcionalidad del modelo, se emplearán los valores encontrados de q en la cinemática inversa. Este enfoque permitirá validar la coherencia entre ambos métodos.

De manera similar, se desarrolló una función que incorpora las ecuaciones derivadas, como se muestra en la Figura 27. En esta función, se emplea la transformación homogénea, utilizando los valores correspondientes a la parte del vector de traslación.

En relación con la simulación del robot, se utilizó el modelo previamente diseñado.

La principal diferencia en este código radica en que, en lugar de variar los ejes de coordenadas, se modifican los valores de q, que representan los ángulos de las articulaciones. En la Figura 28 se ilustra que el robot inicia en una posición de 0 y se desplaza con los valores de 10 cm, 0° y 90°.

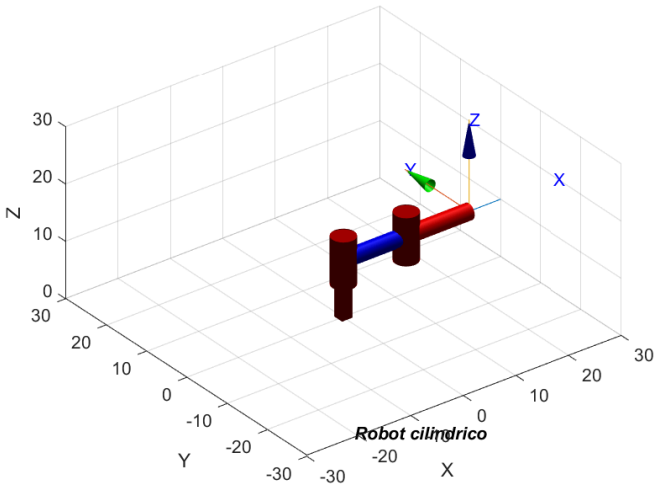


Fig 15. Robot en posición inicial

La Figura 15 ilustra la posición inicial del robot en su configuración de partida. Desde este punto de referencia, el robot se desplazará a lo largo de su trayectoria programada, permitiendo observar cómo las variaciones en los ángulos de las articulaciones afectan su posición en el espacio de trabajo.

Este bloque de código, ilustrado en la Figura 29, simula el movimiento del robot y traza la trayectoria de su efector final. A través de un bucle for, se itera sobre todos los valores en los vectores q1_vals, q2_vals y q3_vals, que representan las configuraciones angulares de las articulaciones en cada iteración. Para cada conjunto de ángulos, se calcula la posición del efector final utilizando la función de cinemática directa. Los valores resultantes de x, y y z se muestran en la consola y se actualiza la visualización del robot en su nueva configuración. Además, se almacena la trayectoria del efector final y se grafica en 3D, permitiendo visualizar su desplazamiento a lo largo del tiempo. La pausa entre cada iteración permite observar la animación del movimiento del robot de manera fluida.

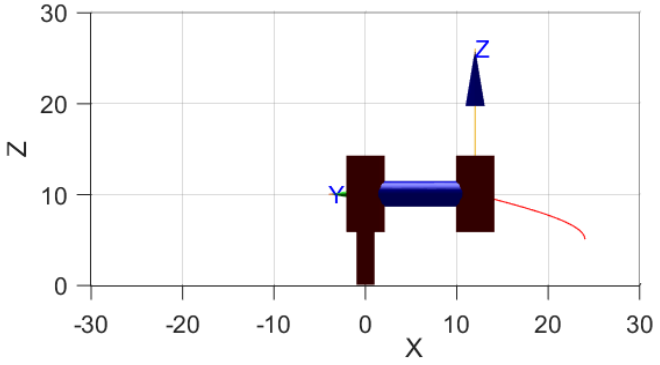


Fig 16. Altura del robot con cinemática directa

Como el valor de q es igual a la altura nos da el valor de 10 cm, por la suma de los 5 cm de la base, como se observa en la Figura 16..

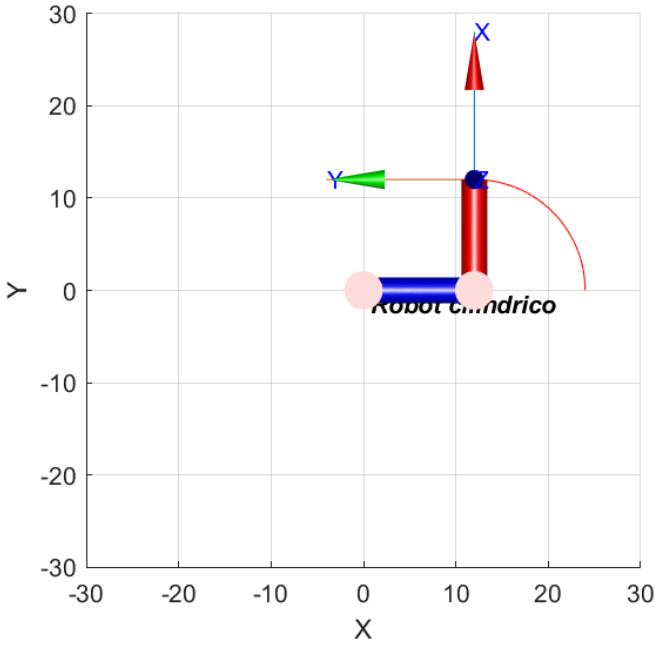


Fig 17. Posición del robot visto desde arriba cinemática directa

A partir de los valores de los ángulos proporcionados, se alcanza la posición final mostrada en la Figura 17. Esta posición coincide con los resultados obtenidos en la cinemática inversa, los resultados son los de la tabla 5:

Tabla 5 Resultados de directa

x	*12*
y	*12*
z	*-10*

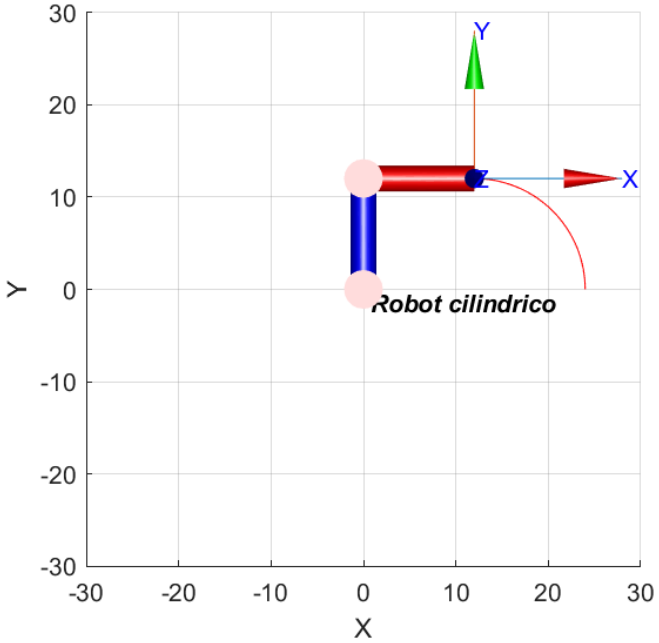


Fig 18. Visto desde arriba cinemática directa, otros valores

Por otro lado si se le dan los otros valores establecidos los cuales son, q1 = 10, q2 = 90° y q3 = -90°, dará el mismo punto pero llegando de diferente forma, como se ve en la Figura 18.

Se obtuvo el mismo resultado de la tabla 6:

Tabla 6 Resultados de directa de diferentes forma

x	*12*
y	*12*
z	*10*

Además, se llevó a cabo la construcción del robot cilíndrico, que se presenta en la Figura 19. Este ensamblaje fue utilizado para la realización de las pruebas experimentales y la evaluación del sistema.

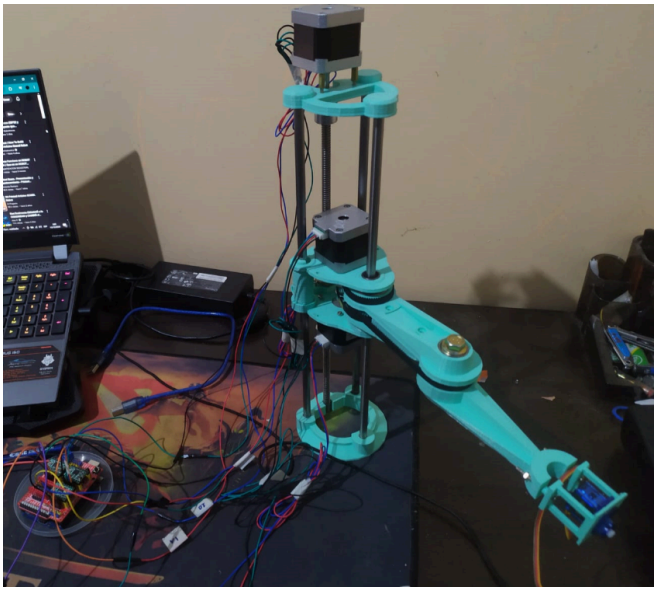


Fig 19. Robot cilíndrico ensamblado

Se llevaron a cabo pruebas experimentales utilizando los ángulos programados para los motores, lo que permitió evaluar parcialmente el desempeño del robot. Sin embargo, debido a dificultades técnicas relacionadas con las correas dentadas, no fue posible realizar pruebas adicionales en esta etapa.

VI. CONCLUSIÓN

Se desarrolló un robot de 4 grados de libertad, aunque se ignoró en gran medida el último motor, ya que solo afectaba el ángulo del efector final. Durante el desarrollo del proyecto, se realizó el modelado 3D del robot, junto con la elaboración de planos y selección de materiales, lo que permitió visualizar el robot final propuesto.

En cuanto al dimensionamiento de los motores utilizados, se presentaron cálculos de torques y relaciones de las correas,

que sirvieron para establecer lo mínimo necesario para el robot.

Desde el punto de vista teórico, se llevaron a cabo los cálculos necesarios para simular el robot, abarcando tanto la cinemática directa mediante transformadas homogéneas como la cinemática inversa utilizando métodos algebraicos y geométricos.

Las simulaciones mostraron que ambos métodos calculados producen resultados coherentes. Se realizaron simulaciones desde diferentes enfoques para verificar que los resultados son consistentes.

También se establecieron los espacios de trabajo del robot, teniendo una movilidad de 180° en ambos brazos y una altura máxima de 17 cm.

Se completó el ensamblaje del robot y se llevaron a cabo pruebas funcionales en los motores, enviando órdenes específicas para ajustar los ángulos de cada uno.

Por otro lado la parte experimental si bien se ha armado el robot la falta de sensores en el mercado entorpece el resultado fnal de este apartado, si bien el robot es funcional no cuenta con una retroalimentación para experimentar de forma correcta

VII. BIBLIOGRAFÍA

[1] G. Rahul y A. Gedam, "Review on Development of Industrial Robotic Arm," Industrial Research & Development, vol. 04, 2017.

[2] A. de Giorgio and L. Wang, "Artificial Intelligence Control in 4D Cylindrical Space for Industrial Robotic Applications," in IEEE Access, vol. 8, pp. 174833-174844, 2020, doi: 10.1109/ACCESS.2020.3026193.

[3] R. Siemasz, K. Tomczuk, y Z. Malecha, "3D printed robotic arm with elements of artificial intelligence," Procedia Computer Science, vol. 176, pp. 3741-3750, 2020. . doi: https://doi.org/10.1016/j.procs.2020.09.013.

[4] N. Barai and S. Manekar, "Review on design and development of intelligent robotic arm," 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2015, pp. 1-3, doi: 10.1109/ISCO.2015.7282389.

[5] Mohammed Abu Qassem, I. Abuhadrous and H. Elaydi, "Modeling and Simulation of 5 DOF educational robot arm," 2010 2nd International Conference on Advanced Computer Control, Shenyang, China, 2010, pp. 569-574, doi: 10.1109/ICACC.2010.5487136.

[6] Murray, R. M., Li, Z., & Sastry, S. (1994). A Mathematical Introduction to Robotic Manipulation. CRC Press.

[7] Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Pearson.

[8] Spong, M. W., & Vidyasagar, M. (2008). Robot Dynamics and Control. Wiley.

[9] Siciliano, B., & Sciavicco, L. (2010). Robotics: Modeling, Planning and Control. Springer.

[10] McGowan, J. (2016). Stepper Motor Basics. In Introduction to Electric Motors.

[11] Athani, V. V. (1997). *Stepper Motors: Fundamentals, Applications, and Design*. New Age International Publishers.

[12] [1] Diario Electrónico Hoy, “Descripción del driver A4988.” [En línea]. Disponible en: https://www.diarioelectronicohoy.com/blog/descripcion-del-driver-a4988. [Accedido: 10-12-2024].

VIII. ANEXO

```
#include <AccelStepper.h>

unsigned long lastTime = 0;

////////////////////////////// q1 ////////////////////////////////////

const int q1stepPin = 2;
const int q1dirPin = 5;

AccelStepper q1Motor(1, q1stepPin, q1dirPin);

///// Transformacion a posicion deseada en mm
double q1DistanceToSteps = 200/2;
int q1position = 10*q1DistanceToSteps;


////////////////////////////// q2 ////////////////////////////////////

const int q2stepPin = 3;
const int q2dirPin = 6;

AccelStepper q2Motor(1, q2stepPin, q2dirPin);

/////// Transformacion a posicion deseada en grados
double q2AngleToSteps = (200.0/360.0)*(62.0/20.0);
int q2position = 0*q2AngleToSteps;

```

Fig 20. Configuración de los pines para los motores

```
void setup() {

  q1Motor.setMaxSpeed(1000);
  q1Motor.setAcceleration(200);
  q1Motor.setCurrentPosition(0);
  ////Orden de movimiento
  q1Motor.moveTo(q1position);


  q2Motor.setMaxSpeed(200);
  q2Motor.setAcceleration(100);
  q2Motor.setCurrentPosition(0);
  ////Orden de movimiento
  q2Motor.moveTo(q2position);

  q3Motor.setMaxSpeed(200);
  q3Motor.setAcceleration(100);
  q3Motor.setCurrentPosition(0);
  ////Orden de movimiento
  q3Motor.moveTo(q3position);

}
```

Fig 21. Configuracion de parametros del motor

```
64  void loop() {
65
66    q1Motor.run();
67    q2Motor.run();
68    q3Motor.run();
69
70  }
```

Fig 22. Ciclo para el funcionamiento de los motores

```
% Longitudes de los eslabones
l2 = 12; % Longitud del primer eslabón rotativo
l3 = 12; % Longitud del segundo eslabón rotativo
elevacion_base = 5; % Elevar el robot 5 cm desde el suelo

% Definición de las articulaciones del robot
L1 = Prismatic('theta', 0, 'a', 0, 'alpha', 0, 'qlim', [0 15]);
L2 = Revolute('d', 0, 'a', l2, 'alpha', 0); % Primer eje rotativ
L3 = Revolute('d', 0, 'a', l3, 'alpha', 0); % Segundo eje rotati

% Ensamblar el manipulador
R = SerialLink([L1, L2, L3], 'name', 'Robot cilindrico');
```

Fig 23. Códigos de los parámetros del robot

```
function [q1, q2, q3] = Cinematica_Inversa(x, y, z)
    % Parámetros del robot
    l1 = 12; % Longitud del primer eslabón (en cm)
    l2 = 12; % Longitud del segundo eslabón (en cm)

    % Cinemática Inversa
    q1 = z;  % El valor de q1 es simplemente la coordenada Z

    % Calculamos q2 y q3 usando las fórmulas proporcionadas
    C3 = (x^2 + y^2 - l1^2 - l2^2) / (2 * l1 * l2);  % C3
    S3 = sqrt(1 - C3^2);  % S3

    q3 = atan2(S3, C3);  % ángulo q3
    q2 = atan2(y, x) - atan2(l2 * sin(q3), l1 + l2 * cos(q3));
end
```

Fig 24. Ecuaciones de cinemática inversa

```
% Definir los valores de x, y, z para simular el movimient
x_vals = linspace(5, 12, 50);  % Generamos valores para el
y_vals = linspace(5, 12, 50);  % Generamos valores para el
z_vals = linspace(elevacion_base, elevacion_base + 5, 50);
```

Fig 25. valores del sistemas de coordenadas

```
for i = 1:length(x_vals)
    x = x_vals(i);
    y = y_vals(i);
    z = z_vals(i);  % El valor de Z siempre se eleva desde la base

    % Calcular los valores de q1, q2 y q3 usando la Cinemática Inversa
    [q1, q2, q3] = Cinematica_Inversa(x, y, z);

    % Mostrar los valores calculados
    disp(['q1 = ', num2str(q1)]);
    disp(['q2 = ', num2str(q2)]);
    disp(['q3 = ', num2str(q3)]);

    % Crear el vector q para las articulaciones
    q = [q1, q2, q3];  % Solo los ángulos, no es necesario incluir un eslabón extra

    % Actualizar la visualización del robot
    R.plot(q); % Mostrar el robot en su nueva configuración

    % Obtener la posición del efector final
    efector_final_pos = R.fkine(q).t; % Obtener la posición del efector final

    % Almacenar la posición del efector final
    efector_final_trajectory = [efector_final_trajectory; efector_final_pos(1), efector_final_pos(2), efector_final_pos(3)

    % Graficar el trazo del efector final
    plot3(efector_final_trajectory(:,1), efector_final_trajectory(:,2), efector_final_trajectory(:,3), 'r'); % Trazo en ro

    % Pausa para que se vea la animación
    pause(0.1);  % Puedes ajustar el tiempo para acelerar o ralentizar la animación
end
```

Fig 26. código para simular inversa

```
function [x, y, z] = Cinematica_Directa(q1, q2, q3, l1, l2)
    % Cinemática Directa para el robot cilíndrico
    % Cálculos de la posición del efector final
    x = l1 * cos(q2) + l2 * cos(q2 + q3);
    y = l1 * sin(q2) + l2 * sin(q2 + q3);
    z = q1; % Movimiento prismático
end
```

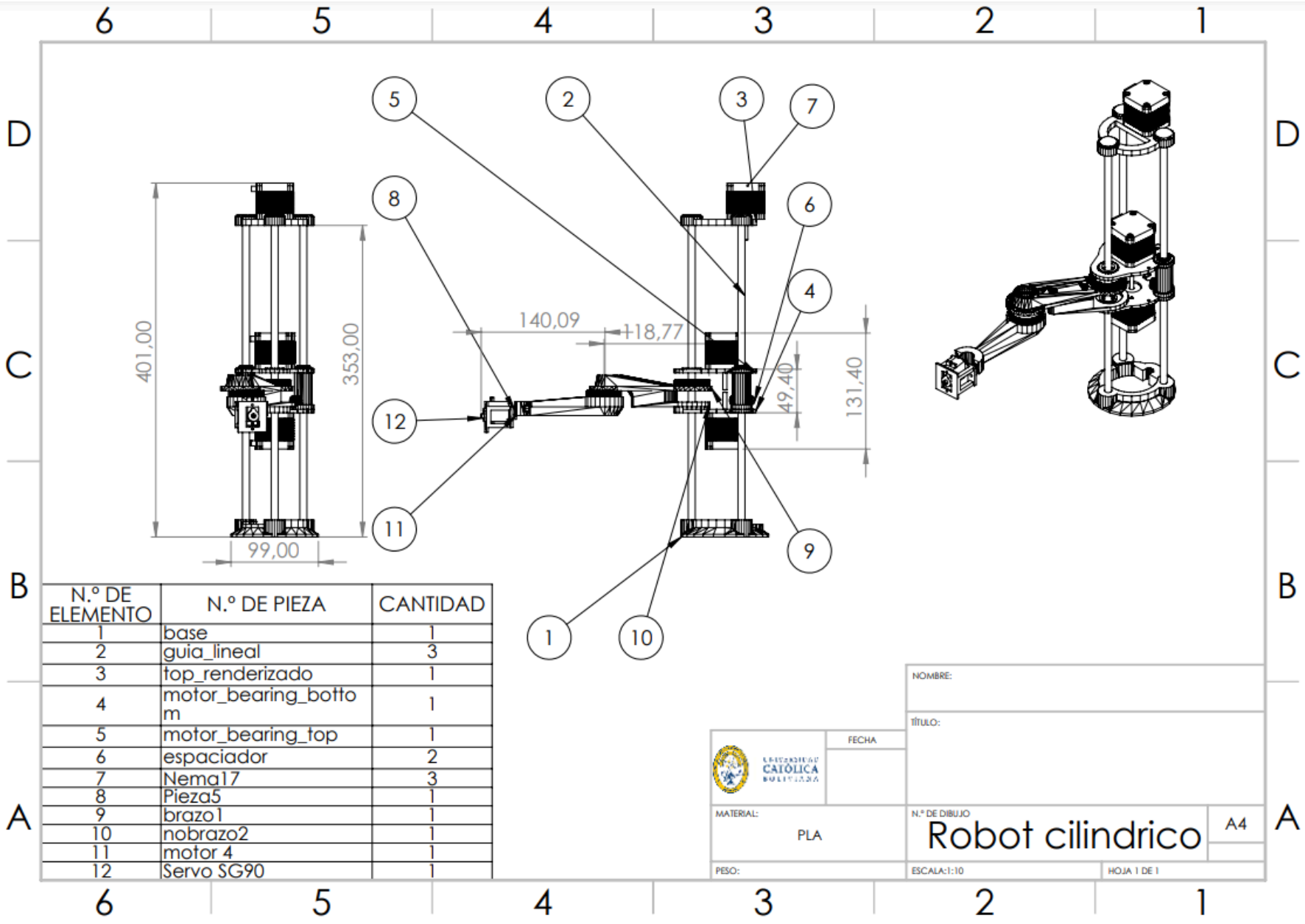
Fig 27. Función de cinemática directa

```
% Definir los valores de q1, q2, q3 para simular el movimie
q1_vals = linspace(elevacion_base, elevacion_base + 5, 50);
q2_vals = linspace(0, 0, 50);  % Valores para el primer áng
q3_vals = linspace(0, pi/2, 50);  % Valores para el segundo
```

Fig 28. Valores variables de los ángulos

```
for i = 1:length(q1_vals)
    q1 = q1_vals(i);
    q2 = q2_vals(i);
    q3 = q3_vals(i);

    % Calcular las posiciones del efector final usando la Cinemática Directa
    [x, y, z] = Cinematica_Directa(q1, q2, q3, l2, l3);

    % Mostrar los valores calculados
    disp(['x = ', num2str(x)]);
    disp(['y = ', num2str(y)]);
    disp(['z = ', num2str(z)]);

    % Crear el vector q para las articulaciones
    q = [q1, q2, q3];

    % Actualizar la visualización del robot
    R.plot(q); % Mostrar el robot en su nueva configuración

    % Almacenar la posición del efector final
    efector_final_trajectory = [efector_final_trajectory; x, y, z];

    % Graficar el trazo del efector final
    plot3(efector_final_trajectory(:,1), efector_final_trajectory(:,2), efector_final_trajectory(:,3), 'r');

    % Pausa para que se vea la animación
    pause(0.1); % Puedes ajustar el tiempo para acelerar o ralentizar la animación
end
```

Fig 29. código para simular directa



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	base	1
2	guia_lineal	3
3	top_renderizado	1
4	motor_bearing_botto m	1
5	motor_bearing_top	1
6	espaciador	2
7	Nema17	3
8	Pieza5	1
9	brazo1	1
10	nobrazo2	1
11	motor 4	1
12	Servo SG90	1

NOMBRE:

TÍTULO:

FECHA

MATERIAL: PLA

N.º DE DIBUJO: Robot cilindrico

A4

PESO:

ESCALA:1:10

HOJA 1 DE 1

Fig 30. planos del robot cilindrico ensamblado

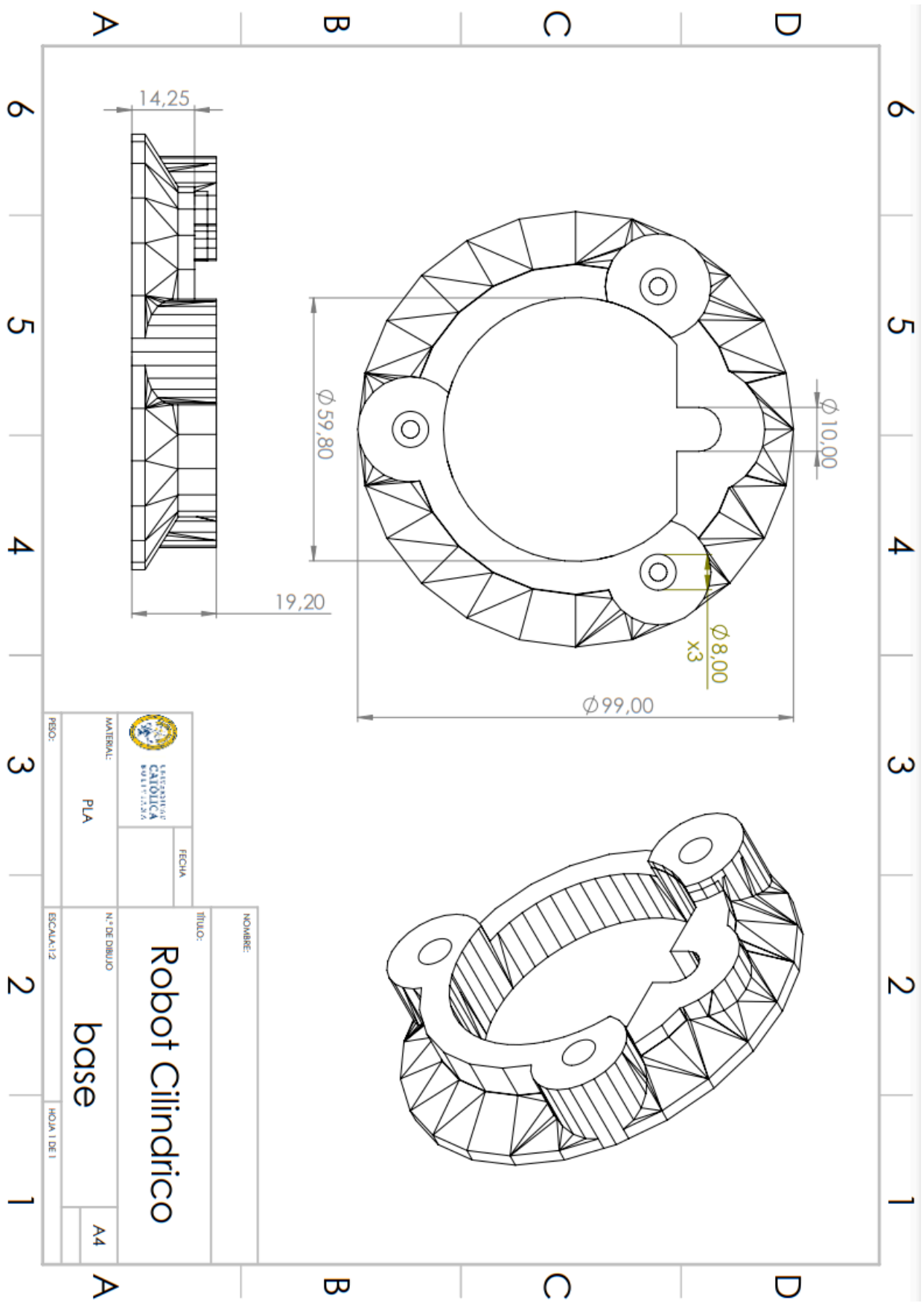


Fig 31. Planos de la base

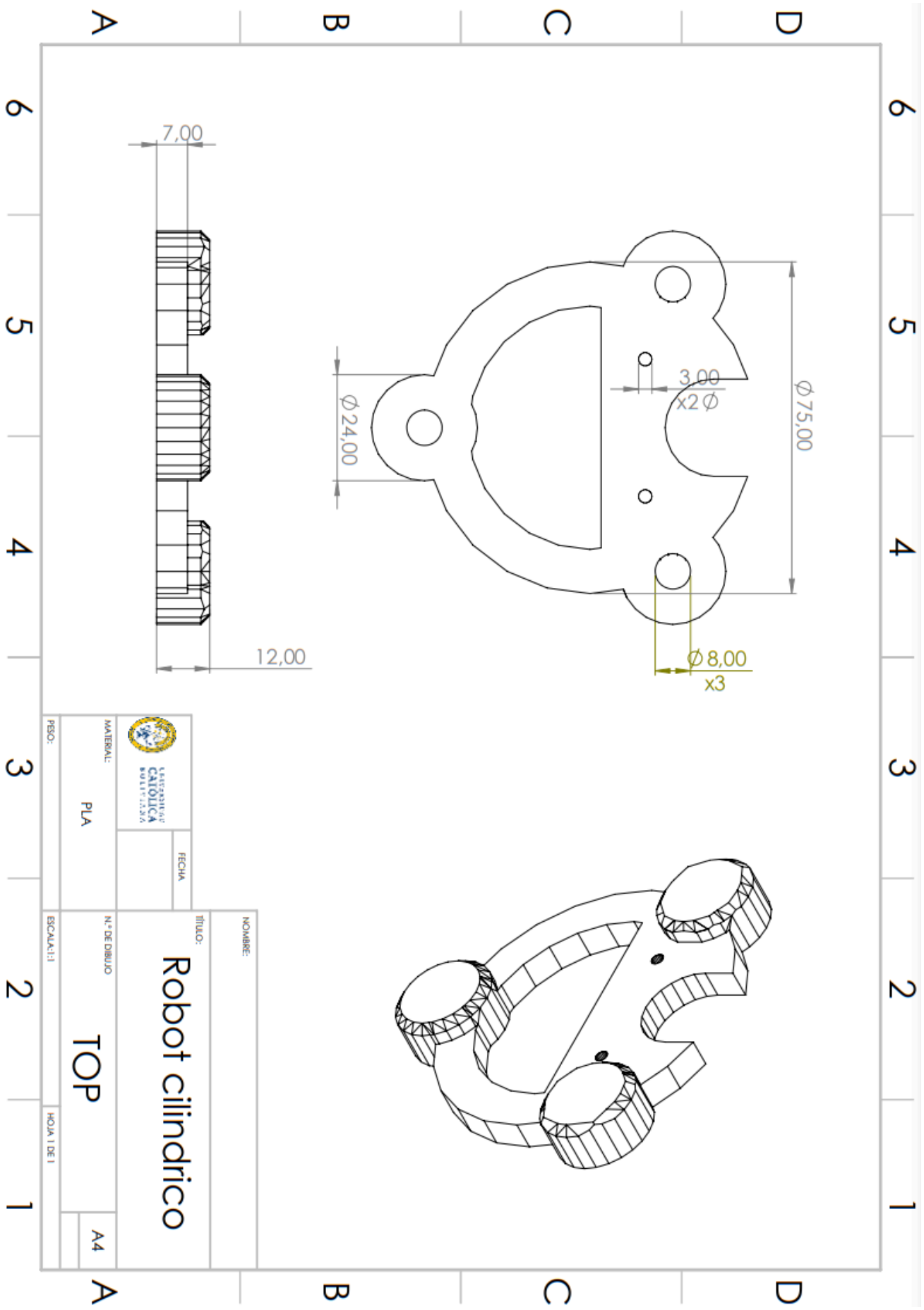


Fig 32. Planos de la parte superior

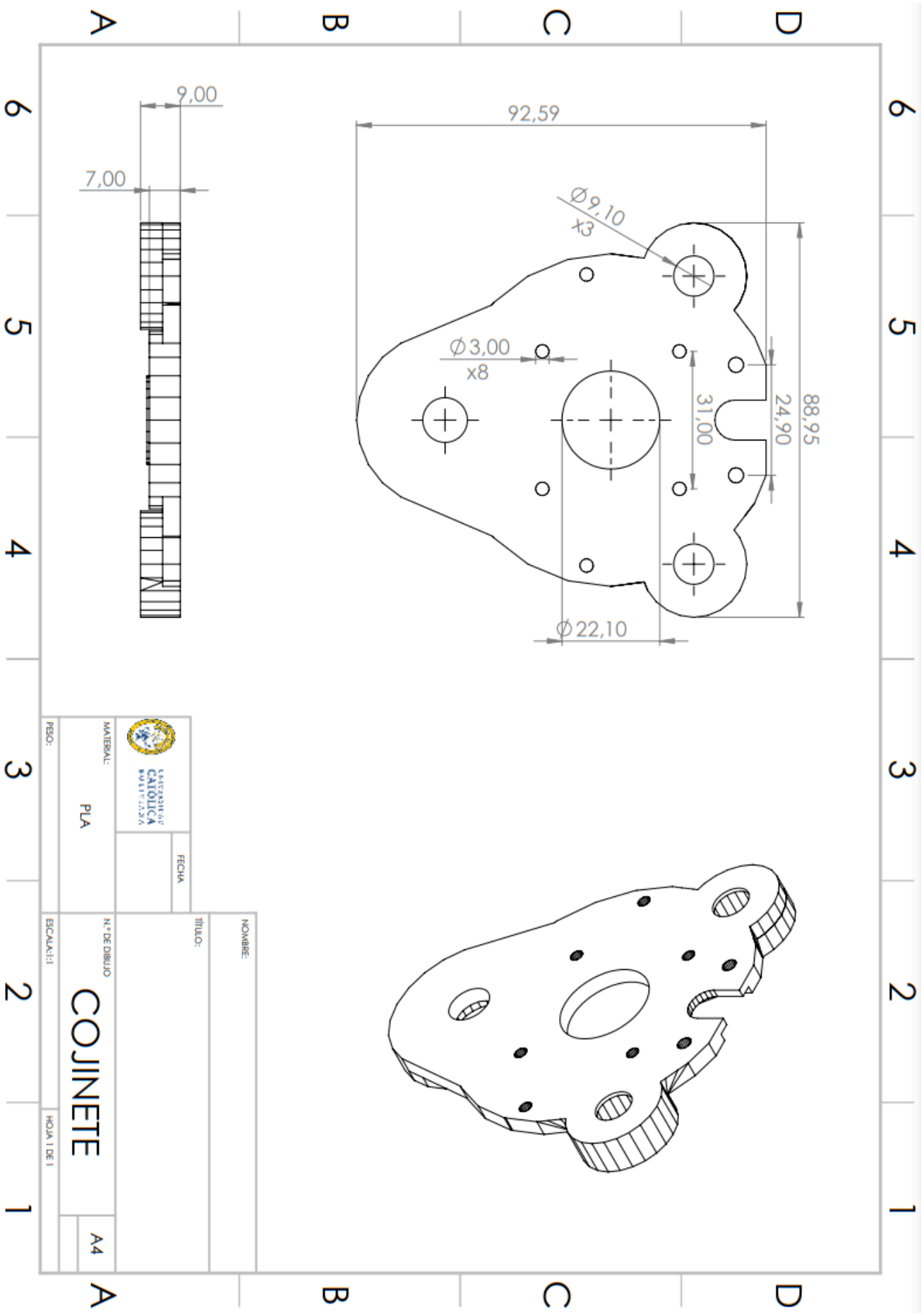


Fig 33. Cojinete para los motores

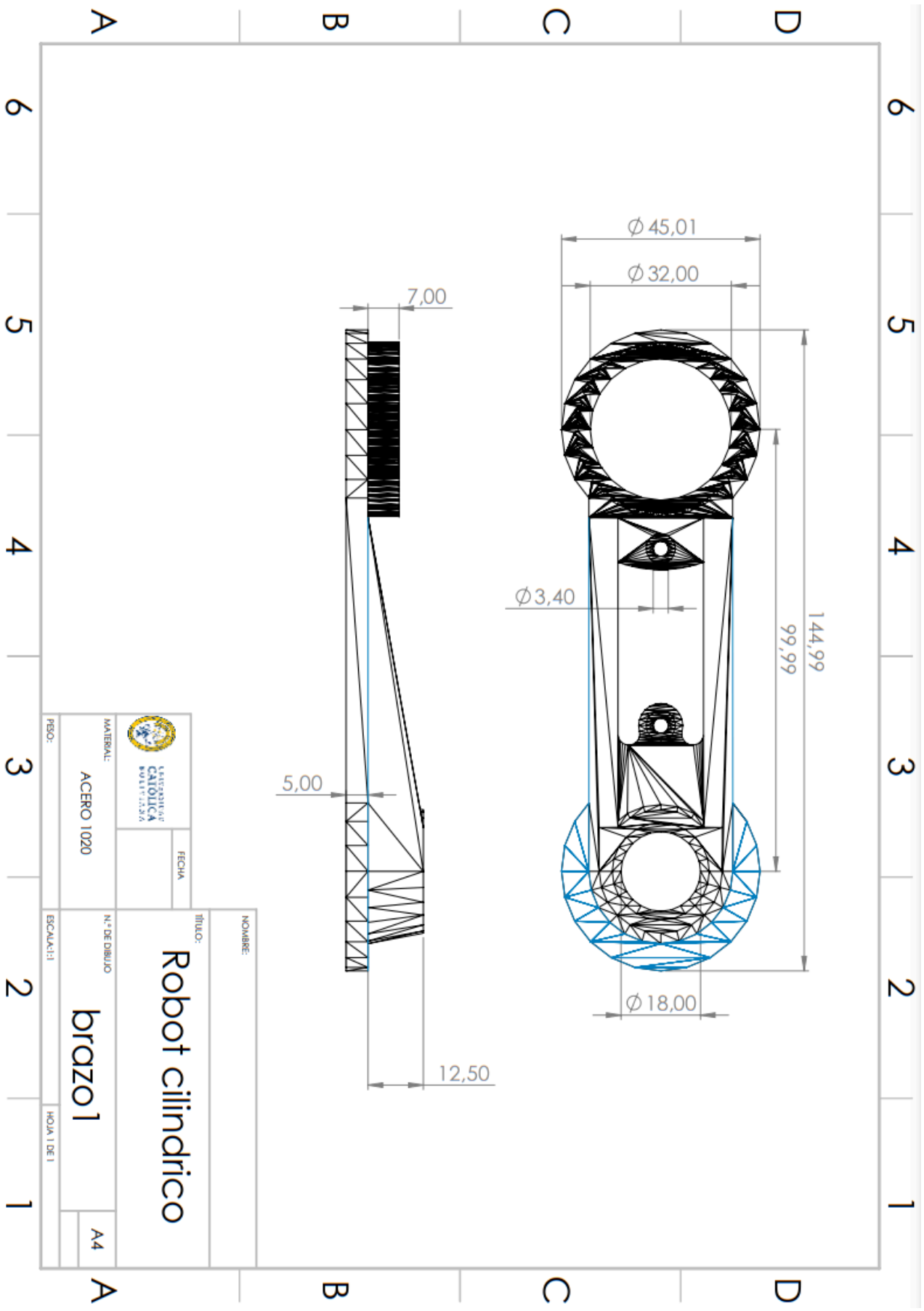


Fig 34. Planos del brazo 1

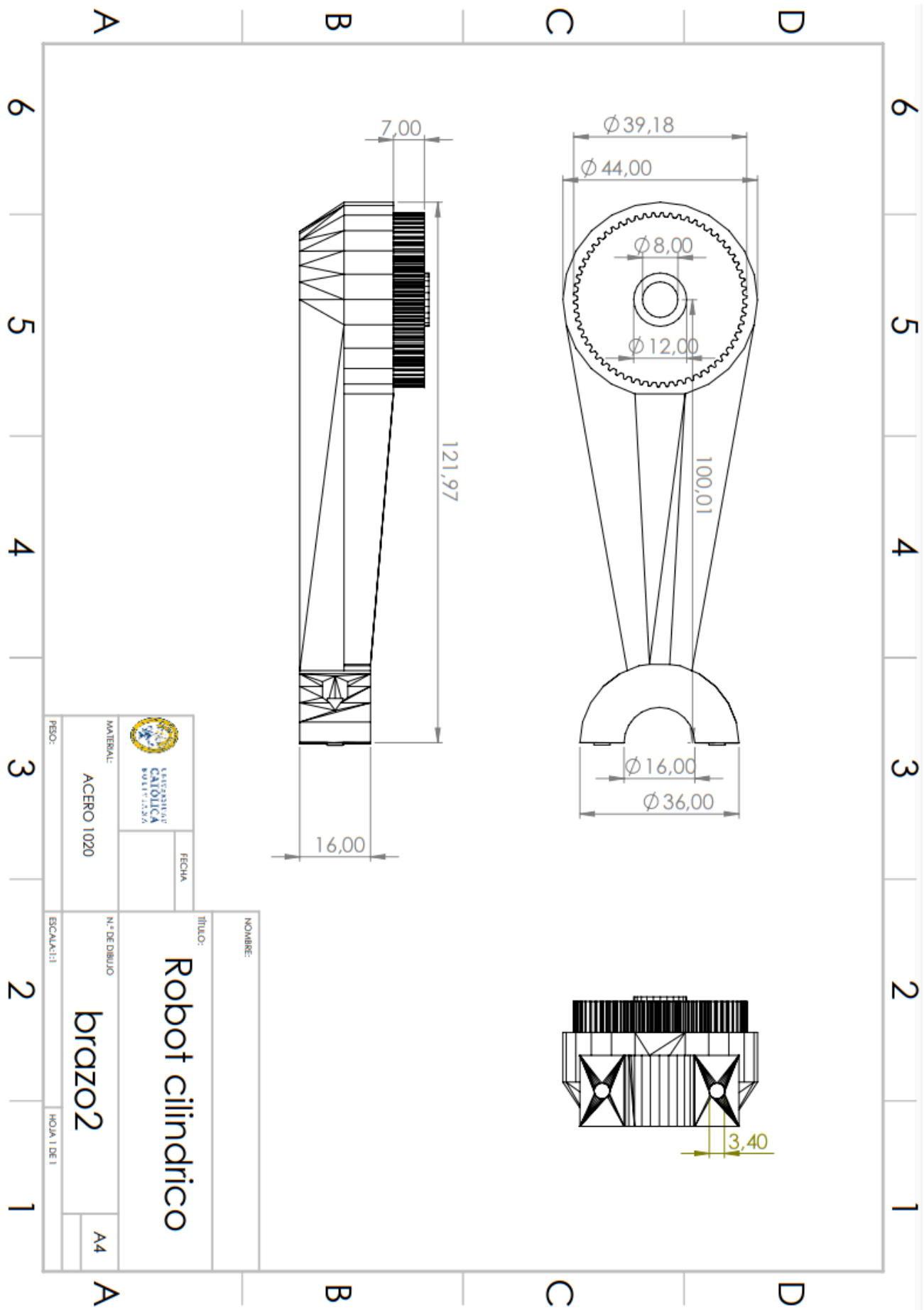


Fig 35. Planos del brazo 2