

**Universidad Católica de Santa María**

**Facultad de Ciencias e Ingenierías Físicas y Formales**

**Escuela Profesional de Ingeniería Mecatrónica**



**Diseño e implementación de un robot paralelo tipo delta con visión artificial  
para paletizado y discriminación de objetos**

Tesis presentada por el bachiller:

**Llerena Sanchez, Julio Cesar**

**ORCID: 0009-0008-6599-4911**

Para optar el Título Profesional de Ingeniero Mecatrónico

Asesor:

**Mg. Collado Oporto, Christiam Guillermo**

**ORCID: 0000-0002-0529-738X**

**Arequipa– Perú**

**2025**

UCSM-ERP

**UNIVERSIDAD CATÓLICA DE SANTA MARÍA**

**INGENIERIA MECATRONICA**

**TITULACIÓN CON TESIS**

**DICTAMEN APROBACIÓN DE BORRADOR**

Arequipa, 28 de Agosto del 2025

**Dictamen: 013490-C-EPIMMEM-2025**

Visto el borrador del expediente 013490, presentado por:

**2019800431 - LLERENA SANCHEZ JULIO CESAR**

Titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT PARALELO TIPO DELTA CON VISIÓN ARTIFICIAL  
PARA PALETIZADO Y DISCRIMINACIÓN DE OBJETOS**

Nuestro dictamen es:

**APROBADO**

Titulo Profesional/Titulo de Segunda Especialidad/Grado Académico a optar:

**INGENIERO MECATRÓNICO**

**40302702 - QUISPE CCACHUCO MARCELO JAIME  
DICTAMINADOR**



**29277716 - CUADROS MACHUCA JUAN CARLOS  
DICTAMINADOR**



**46292714 - SILES NATES FERNANDO DAVID  
DICTAMINADOR**



# Diseño e implementación de un robot paralelo tipo delta con visión artificial para paletizado y discriminación de objetos

## INFORME DE ORIGINALIDAD

18%

INDICE DE SIMILITUD

17%

FUENTES DE INTERNET

4%

PUBLICACIONES

6%

TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1	<a href="http://laccei.org">laccei.org</a> Fuente de Internet	6%
2	<a href="#">Submitted to Universidad De Cuenca</a> Trabajo del estudiante	2%
3	<a href="http://www.coursehero.com">www.coursehero.com</a> Fuente de Internet	1%
4	<a href="http://tesis.ucsm.edu.pe">tesis.ucsm.edu.pe</a> Fuente de Internet	1%
5	<a href="#">Submitted to Universidad de Pamplona</a> Trabajo del estudiante	1%
6	<a href="http://hdl.handle.net">hdl.handle.net</a> Fuente de Internet	1%
7	<a href="http://repositorio.ucsm.edu.pe">repositorio.ucsm.edu.pe</a> Fuente de Internet	1%
8	<a href="http://www.scribd.com">www.scribd.com</a> Fuente de Internet	<1%

[repositorio.uceva.edu.co](http://repositorio.uceva.edu.co)

## DEDICATORIA

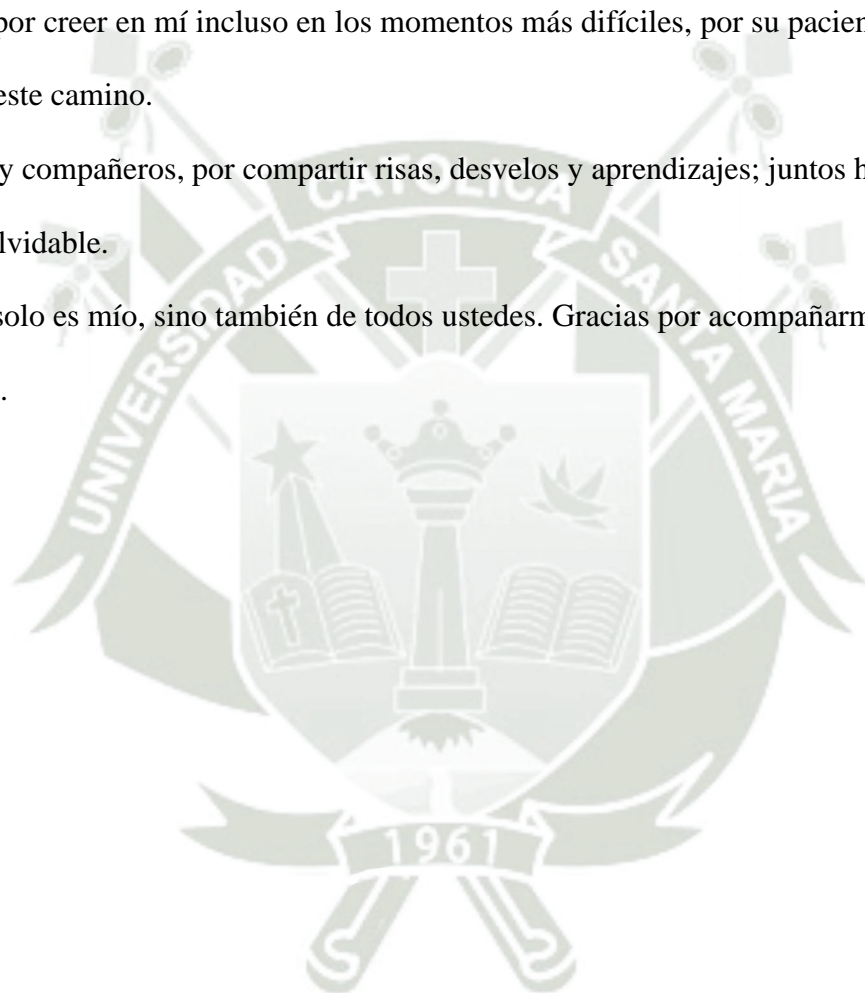
Agradezco a Dios, por ser mi guía en cada paso, por darme la fuerza y la sabiduría necesarias para alcanzar este logro.

A mis padres, por su amor incondicional, sus sacrificios y su ejemplo constante de esfuerzo y perseverancia. Todo lo que soy es reflejo de su apoyo y enseñanzas.

A mi familia, por creer en mí incluso en los momentos más difíciles, por su paciencia y aliento en cada etapa de este camino.

A mis amigos y compañeros, por compartir risas, desvelos y aprendizajes; juntos hicimos que este viaje fuera inolvidable.

Este logro no solo es mío, sino también de todos ustedes. Gracias por acompañarme en este sueño hecho realidad.



## RESUMEN

En el presente documento se plantea el diseño y la implementación de un Robot Paralelo tipo Delta semi-industrial, en el que se aplicó visión artificial con Python en el programa Pycharm donde se discriminaron objetos por tamaño y color, dependiendo de la cantidad de píxeles ocupados y el color en código HSV, es importante mencionar que para un eficiente reconocimiento se trabajó en un ambiente con iluminación controlada; los objetos fueron desplazados en una faja transportadora de 20 cm de ancho donde fueron clasificados en 3 diferentes grupos, considerando un orden de prioridad como el más próximo a salir del proceso como el más importante, para lograr esto se empleó una cámara SD con una resolución de 480 x 640 píxeles, suficiente para nuestro propósito, en la base inferior del robot se instaló un actuador magnético retráctil, el cual será el encargado de recoger cada uno de los objetos. Como aporte importante, se presentó de manera detallada un modelo matemático, la cinemática y dinámica del Robot paralelo tipo delta a modo de ecuaciones de manera que sea adaptable a cualquier diseño, siempre y cuando se respete su estructura física, además, se presentó el proceso de simulación en MATLAB, Arduino y Pycharm junto con planos de toda la estructura metálica, plástica y electrónica evidenciando cada proceso de ensamble, pruebas de actuadores y reconocimiento de objetos.

***Palabras clave:*** Delta, Visión Artificial, Paralelo.

## ABSTRACT

In this document the design and implementation of a semi-industrial Delta type Parallel Robot is proposed, in which artificial vision was applied with Python in the Pycharm program where objects were discriminated by size and color, depending on the number of pixels occupied and the color in HSV code, it is important to mention that for efficient recognition we worked in an environment with controlled lighting; the objects were moved on a 20 cm wide conveyor belt where they were classified into 3 different groups, considering an order of priority as the closest to leaving the process as the most important, to achieve this an SD camera with a resolution of 480 x 640 pixels was used, sufficient for our purpose, a retractable magnetic actuator was installed on the lower base of the robot, which will be in charge of collecting each of the objects in a timely manner and taking it to its corresponding box according to its characteristics. As an important contribution, a detailed mathematical model was presented, the Kinematics and Dynamics of the Delta-type parallel Robot in the form of equations so that it can be adapted to any design as long as its physical structure is respected. In addition, the simulation process in MATLAB, Arduino and Pycharm was presented along with plans of the entire metallic, plastic and electronic structure, showing each assembly process, actuator testing and object recognition.

**Keywords:** Delta, Artificial Vision, Parallel.

## ÍNDICE

DEDICATORIA

RESUMEN

ABSTRACT

INTRODUCCIÓN ..... 1

CAPÍTULO I: MARCO METODOLÓGICO..... 2

1.	DESCRIPCIÓN DEL PROBLEMA .....	2
1.1.	ENUNCIADO.....	2
1.2.	FORMULACIÓN DEL PROBLEMA.....	3
1.3.	ALCANCES DE LA INVESTIGACIÓN .....	3
1.4.	LIMITACIONES DE LA INVESTIGACIÓN .....	3
2.	OBJETIVOS DE LA INVESTIGACION .....	4
2.1.	OBJETIVO GENERAL.....	4
2.2.	OBJETIVOS ESPECÍFICOS.....	4
3.	VARIABLES .....	5
3.1.	VARIABLE DEPENDIENTE .....	5
3.2.	VARIABLE INDEPENDIENTE.....	5
4.	ANTECEDENTES DE LA INVESTIGACIÓN.....	7
5.	NIVEL DE INVESTIGACIÓN .....	9
6.	DISEÑO DE INVESTIGACIÓN.....	9
7.	INSTRUMENTOS DE RECOLECCIÓN DE DATOS .....	9
8.	TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS.....	10
9.	ÁREA DE INVESTIGACIÓN.....	10
10.	SUB ÁREA DE INVESTIGACIÓN .....	10

11.	LÍNEA DE INVESTIGACIÓN .....	10
12.	ASPECTOS ADMINISTRATIVOS .....	10
12.1.	RECURSOS NECESARIOS .....	10
CAPÍTULO II: MARCO TEÓRICO .....		11
1.	ROBÓTICA .....	11
1.1.	DEFINICIÓN DE ROBOT .....	11
1.2.	CLASIFICACIÓN DE LOS ROBOTS .....	11
1.3.	MORFOLOGÍA DEL ROBOT .....	16
1.4.	MÉTODOS DE LOCALIZACIÓN ESPACIAL .....	22
1.5.	CINEMÁTICA DE UN ROBOT .....	23
1.6.	DEFINICIÓN DE ROBOT PARALELO: .....	30
2.	VISIÓN ARTIFICIAL: .....	32
CAPÍTULO III: IMPLEMENTACIÓN .....		35
1.	MODELAMIENTO MATEMÁTICO .....	35
1.1.	CINEMÁTICA DIRECTA .....	35
1.2.	CINEMÁTICA INVERSA .....	37
1.3.	CONTRASTE DE DATOS EN MATLAB .....	44
1.4.	SIMULACION CON TOOL BOX DE ROBÓTICA .....	49
1.5.	DIMENCIONAMIENTO DEL ROBOT .....	52
1.6.	DINAMICA DEL ROBOT .....	55
2.	DISEÑO DEL ROBOT .....	58
2.1.	PARTE PLASTICA .....	58
2.2.	PARTE METÁLICA .....	61
2.3.	PARTE ELECTRÓNICA .....	63

2.4.	PROGRAMA EN ARDUINO .....	67
2.5.	DESARROLLO DE LA VISION ARTIFICIAL.....	69
3.	DISEÑO DE LA FAJA TRASPORTADORA .....	70
3.1.	PARTE PLASTICA.....	70
3.2.	PARTE ELECTRÓNICA.....	75
4.	PRUEBAS Y ENSAYOS.....	76
4.1.	DISEÑO CAD Y PRUEBAS DE FUNCIONAMIENTO FÍSICO .....	76
4.2.	DETERMINACION DEL PUNTO 0 EN LOS MOTORES .....	79
4.3.	PRUEBA DE TRACCION CON UN SOLO BRAZO.....	80
4.4.	ENSAMBLE DE ROBOT Y ESTRUCTURA METALICA .....	80
4.5.	VERIFICACION DE PUNTOS COORDENADOS MANUEALMENTE .....	82
4.6.	VERIFICACION DE TRABAJO CON CAMARA A LAZO CERRADO .....	83
4.7.	PRUEBA DE MOBILIDAD CON FAJA INDEPENDIENTE .....	85
4.8.	COLOR Y POSICION CON LA FAJA EN MOVIMIENTO.....	86
	CONCLUSIONES: .....	88
	RECOMENDACIONES: .....	89
	REFERENCIAS BIBLIOGRÁFICAS .....	90

## ÍNDICE DE FIGURAS

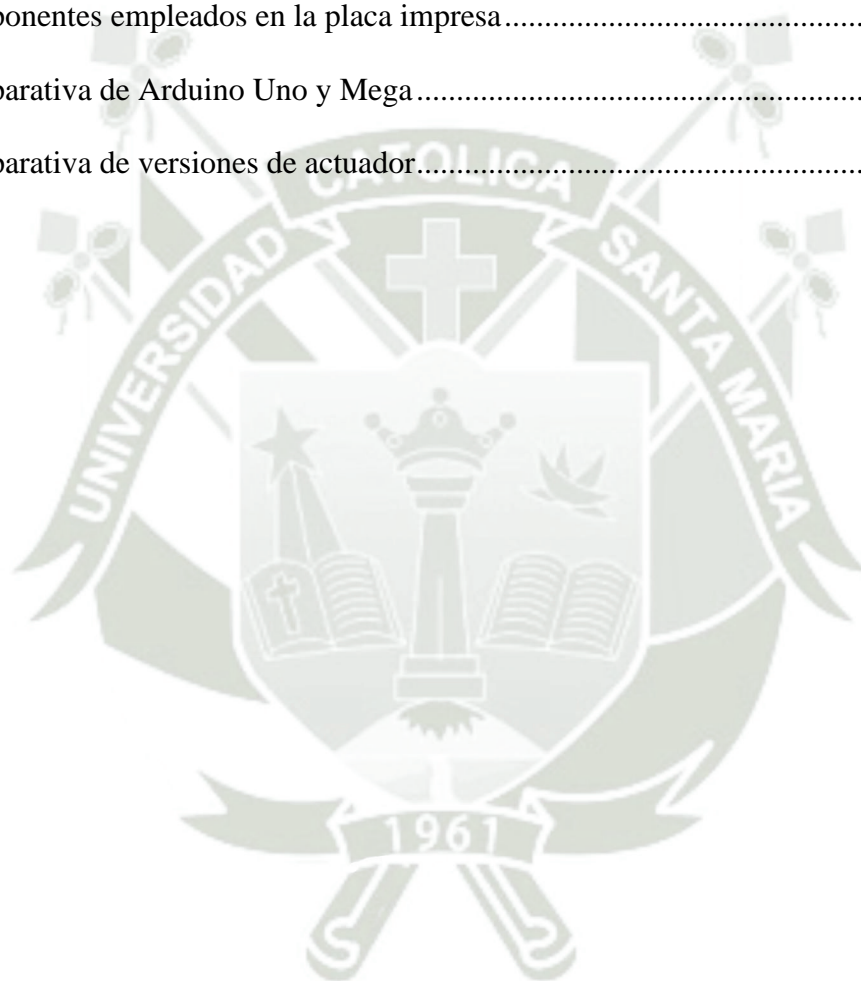
Figura 1:Robot estacionario. ....	12
Figura 2:Robot móvil. ....	12
Figura 3:Robot industrial. ....	13
Figura 4:Robot industrial. ....	13
Figura 5:Robot de servicio. ....	14
Figura 6:Robot de exploración. ....	14
Figura 7:Robot militar. ....	15
Figura 8:Robot de entretenimiento. ....	15
Figura 9:Dron autónomo. ....	16
Figura 10:Actuador del tipo electro - neumático. ....	18
Figura 11:Estructura mecánica. ....	18
Figura 12:Procesamiento de información. ....	19
Figura 13: Fuente de alimentación de 0-30 V y 0-10 A. ....	20
Figura 14: Comunicación Serial. ....	21
Figura 15:Interfaz de usuario. ....	21
Figura 16:Cubierta. ....	22
Figura 17: Denavit Hartenberg de un Robot. ....	23
Figura 18:Cinematica directa. ....	24
Figura 19: Estructura de Matriz de Transformación Homogenea. ....	24
Figura 20: Tabla de DH de un robot: ....	26
Figura 21:Ejemplo de cómo se realizaría la cinemática. ....	27
Figura 22: Ejemplo de modelamiento deferencial. ....	28
Figura 23:Ejemplo de modelamiento dinámico. ....	29

Figura 24: Robot Paralelo de la empresa UIROBOT.....	32
Figura 25: Ejemplo de cómo se aplica la visión artificial en la industria .....	33
Figura 26: Distribución de ángulos por brazo .....	35
Figura 27: DH del Robot Paralelo Tipo Delta.....	36
Figura 28: Triangulo de restricciones del robot .....	45
Figura 29: Iniciar Tool Box de Robótica en la Comand Window .....	49
Figura 30: Representación Digital del Robot respondiendo a su cinemática.....	52
Figura 31: Representacion isometrica del volumen de trabajo .....	54
Figura 32: Vista frontal del volumen de trabajo limitada de $Z = -49$ a $-20$ .....	54
Figura 33: Vista superior del volumen de trabajo limitada a un area de $40 \times 40$ cm .....	55
Figura 34: DCL de la dinámica del robot.....	55
Figura 35: Tensión de Von Mises máxima de 8.24 MPa.....	59
Figura 36: Deformación equivalente a un máximo de 0.016 mm.....	59
Figura 37: Deformación equivalente a un máximo de 0.008 .....	60
Figura 38: Impresión de brazo de robot paralelo .....	61
Figura 39: Impresión de cajas receptoras.....	61
Figura 40: Barillas de aluminio manufacturadas.....	62
Figura 41: Rotulas empernadas a base inferior .....	62
Figura 42: Imagen referencial de ubicación de la base superior del robot.....	63
Figura 43: Vista Normal de la placa impresa .....	64
Figura 44: Vista Real World de la placa impresa.....	65
Figura 45: Vista Artwork de la placa impresa.....	65
Figura 46: Placa impresa quemada.....	66
Figura 47: Placa impresa instalada.....	66

Figura 48: Aplicación de la visión artificial al robot .....	70
Figura 49: Sistema de tracción de la faja transportadora .....	71
Figura 50: Sistema de retorno de la faja transportadora.....	71
Figura 51: Cavidad de la caja reductora para motor paso a paso .....	72
Figura 52: Lateral de la faja transportadora ensamblado .....	73
Figura 53: Eslabones de la faja transportadora ensamblados.....	74
Figura 54: Faja transportadora ensamblada en su totalidad .....	74
Figura 55: Motor paso a paso con caja reductora y catalina de traccion instalados .....	75
Figura 56: Reductor instalado en la faja.....	76
Figura 57: impresión de actuador recolector de objetos .....	77
Figura 58: Prueba de torque de servomotor .....	80
Figura 59: Ensamble de partes plásticas del robot .....	81
Figura 60: Preparación de estructura metálica .....	82
Figura 61: Robot paralelo ensamblado parcialmente y en su totalidad.....	83
Figura 62: Prueba de movilidad a lazo cerrado del robot paralelo.....	84
Figura 63: Representación grafica de los colores en HSB .....	84
Figura 64: Prueba de seguimiento del objeto a tiempo real .....	85

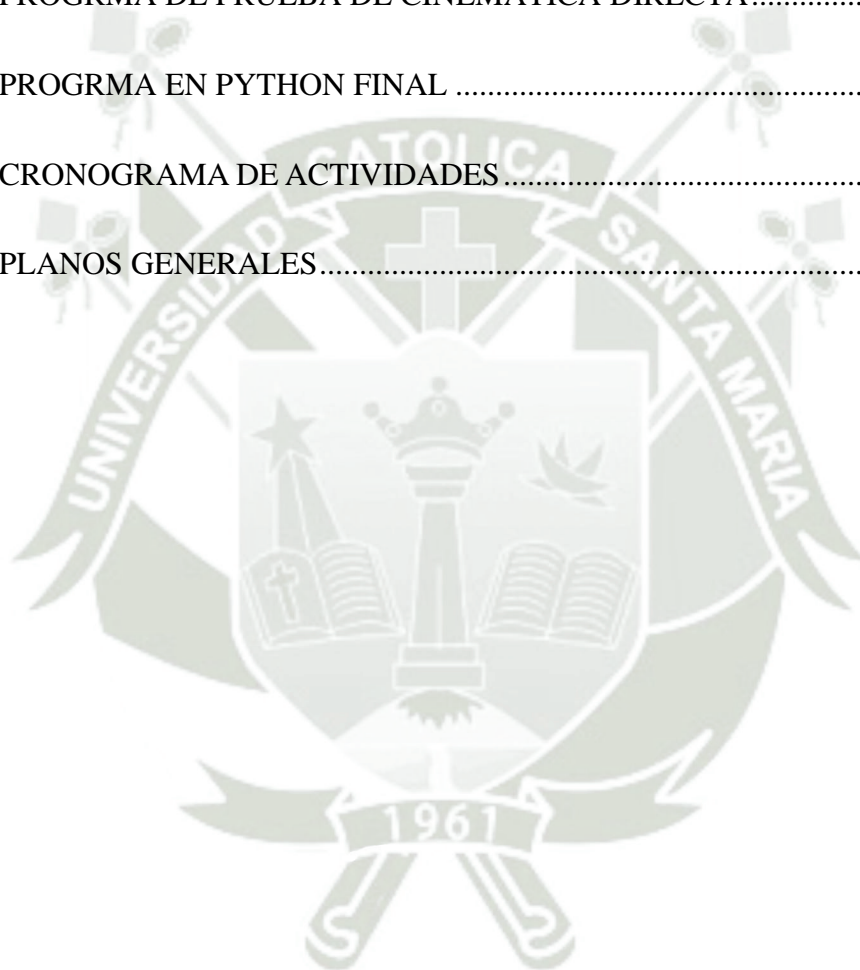
## ÍNDICE DE TABLAS

Tabla 1: Variables dependiente e independiente.....	6
Tabla 2: Recursos necesarios .....	10
Tabla 3: Clasificación de sensores internos de un robot .....	17
Tabla 4: DH del Robot Paralelo Tipo Delta.....	36
Tabla 5: Componentes empleados en la placa impresa.....	67
Tabla 6: Comparativa de Arduino Uno y Mega.....	68
Tabla 7: Comparativa de versiones de actuador.....	78



## ÍNDICE DE ANEXOS

ANEXOS 01: PROGRMA EN ARDUINO FINAL .....	92
ANEXOS 02: PROGRMA EN ARDUINO PARA PRUEBA.....	100
ANEXOS 03: PROGRMA DE PRUEBA DE CINEMATICA DIRECTA CON MTH .....	107
ANEXOS 04: PROGRMA DE PRUEBA DE CINEMATICA DIRECTA.....	109
ANEXOS 05: PROGRMA EN PYTHON FINAL .....	112
ANEXOS 06: CRONOGRAMA DE ACTIVIDADES.....	116
ANEXOS 07: PLANOS GENERALES.....	119



## INTRODUCCIÓN

La automatización industrial continúa evolucionando en la búsqueda de soluciones que ofrezcan mayor eficiencia y versatilidad en tareas de manipulación y ensamblaje. En este contexto, los robots paralelos han ganado protagonismo gracias a su destacada precisión, rapidez operativa y capacidad para manejar cargas significativas. Entre estos, el robot tipo delta ha demostrado ser particularmente efectivo, debido a su estructura cinemática optimizada y su rendimiento en entornos de trabajo exigentes (Merlet, 2006).

Dentro del flujo de producción y logística, el proceso de paletizado representa una etapa clave. La correcta disposición de productos sobre pallets tiene un impacto directo en la eficiencia del almacenamiento, el transporte y en la seguridad global de la cadena de suministro (Zhang, 2017). La implementación de sistemas robóticos en esta etapa puede mejorar notablemente la productividad y reducir errores asociados a la manipulación manual.

Un avance relevante en este campo es la integración de visión artificial en robots paralelos tipo delta. Esta tecnología dota al sistema de percepción visual, permitiéndole detectar, identificar y localizar objetos con alta precisión. La capacidad cognitiva proporcionada por la visión artificial no solo mejora la funcionalidad del robot, sino que también amplía su adaptabilidad para trabajar con productos de diversas formas y tamaños (Siciliano, 2016).

El presente trabajo se estructura en distintos capítulos que exploran el estado actual de la robótica paralela y la visión por computadora, el proceso de diseño y fabricación digital del sistema completo, así como la fase de puesta en marcha, en la que se detallan pruebas y mecanismos de control para asegurar la precisión operativa. Se espera que este estudio contribuya significativamente al desarrollo de soluciones en el ámbito de la automatización industrial, y sirva de base para futuras investigaciones y aplicaciones prácticas en entornos de manufactura.

## CAPÍTULO I: MARCO METODOLÓGICO

### 1. DESCRIPCIÓN DEL PROBLEMA

#### 1.1. ENUNCIADO

En el contexto peruano, actualmente no existe un mercado consolidado que ofrezca soluciones integrales de robots paralelos tipo delta equipados con sistemas de visión artificial. Esta carencia se traduce en una oferta limitada por parte de proveedores nacionales, lo que obliga a las empresas interesadas en incorporar estas tecnologías a asumir el reto de diseñar, ensamblar e implementar por cuenta propia dichos sistemas. En consecuencia, las organizaciones deben invertir recursos adicionales en términos de tiempo, dinero y capacitación técnica especializada para lograr que estos robots funcionen de forma eficiente y adaptada a sus necesidades productivas. Esta situación limita significativamente la adopción de tecnologías de automatización avanzada, especialmente en pequeñas y medianas empresas (PyMEs) que no cuentan con el capital o el personal calificado para integrar robótica y visión computarizada (Corke, 2011). Además, la falta de documentación técnica y guías adaptadas al contexto local impide replicar o escalar soluciones efectivas.

Por ello, se plantea como objetivo principal de esta investigación el desarrollo de una metodología clara para el diseño e implementación de un robot paralelo tipo delta con visión artificial, haciendo énfasis en la flexibilidad de su configuración geométrica y de programación. Este trabajo busca proveer un modelo funcional y adaptable que permita comprender desde los fundamentos matemáticos y cinemáticos (Gosselin, 1990), hasta la integración de hardware y software necesarios para su funcionamiento. Se espera que este enfoque sirva de referencia tanto

académica como práctica, promoviendo el desarrollo local de soluciones robóticas avanzadas.

## **1.2. FORMULACIÓN DEL PROBLEMA**

¿Qué factores permitirán sincronizar de la cinemática del Robot Paralelo tipo Delta con la Visión Artificial para un eficiente reconocimiento de objetos?

## **1.3. ALCANCES DE LA INVESTIGACIÓN**

La automatización de procesos de paletizado es esencial en la industria de manufactura y logística para mejorar la eficiencia, reducir costos y aumentar la calidad de los productos finales, sabiendo que existe una escasez de investigaciones y desarrollos relacionados con robots paralelos tipo Delta y su aplicación en tareas de paletizado, la combinación de robots paralelos tipo Delta y visión artificial es una solución potencialmente eficiente para abordar estos desafíos, ahorrando tiempo y recursos, así como la necesidad de contar con personal capacitado en robótica y visión artificial para el ensamble y configuración necesaria con el propósito de lograr el funcionamiento correcto del Robot clasificando hasta 3 diferentes objetos de diferentes colores, específicamente círculos que serán levantados por el actuador magnético, ya sea que estos estén en paralelo (uno al lado del otro) o en serie (uno detrás de otro), apoyándonos en el modelamiento matemático que también se presentó para el entendimiento de la matemática implicada en un robot paralelo tipo delta y beneficiar a futuras investigaciones.

## **1.4. LIMITACIONES DE LA INVESTIGACIÓN**

En el presente documento se plantea realizar el diseño e implementación completo de un Robot Paralelo tipo Delta, teniendo en cuenta la cinemática directa e inversa para lograr tener su modelo matemático, además su dinámica para calcular las

velocidades de desplazamiento y coordinación del mismo.

En cuanto a la visión artificial, se planea lograr el reconocimiento exitoso de objetos a paletizar por tratamiento digital de imágenes y obtener posiciones y velocidades acertadas de la faja para una correcta precisión de posición de paletización.

Dentro de nuestras limitaciones tenemos:

- La precisión que se pueda llegar a tener por la calidad de materiales manufacturados o la rotación restringida de las rotulas empleadas.
- La velocidad de desplazamiento lineal del robot será de 10 cm/s.
- La velocidad lineal de la faja será de 60 cm/min.
- El área de trabajo de la órbita más baja será de 20 x 20 cm.
- Se alimentará con 7.5 v DC.
- El peso máximo que este puede levantar será de 100 g.
- El trabajo se realiza bajo condiciones adecuadas de iluminación controlada.

## **2. OBJETIVOS DE LA INVESTIGACION**

### **2.1. OBJETIVO GENERAL**

- Diseñar e implementar un Robot Paralelo tipo Delta con visión artificial para la clasificación de objetos empleando paletizado.

### **2.2. OBJETIVOS ESPECÍFICOS**

- Obtener el modelo matemático, la cinemática y dinámica de un Robot paralelo tipo delta.
- Identificar todos los posibles contratiempos que el Procesamiento Digital de Imágenes pueda tener en el sentido que este no confunda los objetos a discriminar con algún agente externo.

- Realizar una simulación en el programa Matlab para probar la sincronización de la cinemática del Robot con movimientos virtuales, además, con Arduino y Pycharm para la visión artificial de manera tal que se pueda adquirir las coordenadas correctas a las que se debe dirigir.
- Probar el robot físicamente, verificando que las coordenadas recibidas corresponden a las reales al momento de dirigirse a un objeto a discriminar y llevándolo a su respectivo lugar, tomando tiempos y variando con diferentes velocidades de desplazamiento.

### 3. VARIABLES

#### 3.1. VARIABLE DEPENDIENTE

Discriminación de objetos.

#### 3.2. VARIABLE INDEPENDIENTE

Visión Artificial.

Tabla 1: Variables dependiente e independiente

VARIABLES	DIMENSION CONPECTUAL	DIMENSIÓN	INDICADORES	RANGO	INSTRUMENTOS
DISCRIMINACIÓN DE OBJETOS	LA PRODUCTIVIDAD SE REFIERE A LA EFICIENCIA CON LA QUE SE UTILIZAN LOS RECURSOS DISPONIBLES PARA ALCANZAR OBJETIVOS Y RESULTADOS ESPECÍFICOS. EN EL CONTEXTO LABORAL Y EMPRESARIAL, LA PRODUCTIVIDAD SUELE MEDIRSE EN TÉRMINOS DE LA RELACIÓN ENTRE LA PRODUCCIÓN (BIENES O SERVICIOS) Y LOS RECURSOS UTILIZADOS PARA GENERAR ESA PRODUCCIÓN, COMO EL TIEMPO, EL ESFUERZO,	PALLET MAGNÉTICO	FORMA	Superficie superior plana	-
			TAMAÑO	< 10 x 5 cm	VERNIER
			PESO	< 200 g	BALANZA
		ACTUADORES NEUMÁTICOS	POTENCIA	Depende del peso y la altura de succión	-
			FORMA DE BOQUILLA	Depende de la forma del pallet	-
			PESO DE LA MANGUERA	< 100 g	BALANZA
		ACTUADORES MECÁNICOS	FUERZA	Depende de la dureza del material a sujetar	-
			FORMA DE LA GARRA	Depende de la forma del pallet	-
			PESO DE LA GARRA	< 300 g	BALANZA
VISIÓN ARTIFICIAL	LA VISIÓN ARTIFICIAL, TAMBIÉN CONOCIDA COMO VISIÓN POR COMPUTADORA, ES UNA RAMA DE LA INTELIGENCIA ARTIFICIAL Y LA INGENIERÍA INFORMÁTICA QUE SE OCUPA DE ENSEÑAR A LAS MÁQUINAS A INTERPRETAR Y COMPRENDER EL MUNDO VISUAL. EL OBJETIVO PRINCIPAL DE LA VISIÓN ARTIFICIAL ES PERMITIR A LAS MÁQUINAS OBTENER INFORMACIÓN A PARTIR DE IMÁGENES O SECUENCIAS DE IMÁGENES, SIMILAR A COMO LO HACEN LOS HUMANOS CON LA VISTA.	INTELIGENCIA ARTIFICIAL	IA DÉBIL (ESTRECHA)	Sin conciencia	-
			IA FUERTE (O GENERAL)	Con conciencia	-
			MEMORIA	<16 GB	RAM
		IMÁGENES	RESOLUCIÓN	480 x 640	PÍXELES
			COLOR	RGB (0-255)	ESCALA NUMÉRICA

#### 4. ANTECEDENTES DE LA INVESTIGACIÓN

Real-time vision-based controller for delta robots, int. j. Intelligent systems technologies and applications, vol. 20, no. 4, 2021

Se presentan dos algoritmos para rastrear objetos en función de sus colores y formas, así como el seguimiento de gestos manuales auto calibrados mediante un dispositivo Kinect. Los algoritmos tienen como objetivo proporcionar un seguimiento eficiente y preciso de objetos y gestos de las manos sin necesidad de una calibración inicial.

El documento destaca las ventajas de utilizar el control basado en visión en aplicaciones industriales, como una mayor eficiencia de producción y una mejor interacción entre los robots y su entorno. También aborda desafíos comunes en el servo visual, incluido el problema del campo de visión y la necesidad de calibración.

Los algoritmos propuestos utilizan un controlador independiente para minimizar el tiempo computacional y una red de control distribuida para lograr control en tiempo real. El documento proporciona detalles sobre la descripción del sistema, el reconocimiento y seguimiento de objetos, el reconocimiento y seguimiento de gestos manuales, el diseño de trayectorias y la programación en tiempo real.

C.A. Peña Cortés, E. Martínez Oviedo, and p. f. Cárdenas Herrera, "Dimensional optimization of a delta parallel robot based on a lower energy consumption," Universidad de Pamplona, Pamplona, Colombia.

En el documento se presenta un enfoque de vanguardia para el control basado en visión en tiempo real para robots delta, aborda desafíos clave y proporciona

capacidades de seguimiento eficientes y precisas. Los algoritmos y el diseño del sistema ofrecen aplicaciones potenciales en diversos entornos industriales.

A. A. L. Pinto and J. A. P. Pinto, "Diseño e implementación de un robot paralelo tipo delta de 3 grados de libertad controlado por medio de lógica difusa para la clasificación de objetos," tesis de ingeniería mecatrónica, Arequipa, Perú, 2018.

El objetivo de esta tesis es desarrollar y operar un robot paralelo delta de tres grados de libertad que se diseñó para emular los movimientos esenciales de clasificación de objetos. Una estrategia de control difuso se utiliza para operar el robot. Para lograr el éxito del proyecto, se realizó un análisis completo que incluyó cinemática, dinámica, mecánica, electrónica, software y control de robot. El comportamiento observado en las evaluaciones prácticas del robot fue la base para elegir y diseñar un controlador difuso adecuado. Como resultado, se desarrollaron programas en lenguaje estructurado que pueden realizar una variedad de tareas, como controlar los movimientos y adquirir datos continuamente.

Smith, J., Johnson, A., & Garcia, M. (2022). Collision diagnosis techniques for parallel delta robots. *robotics and automation conference*, 10(2), 123-135.

Este documento se centra en técnicas de diagnóstico de colisiones para robots delta paralelo. Se han aplicado varios enfoques basados y sin modelos para la detección de colisiones en sistemas robóticos en serie genéricos y vehículos autónomos [8]. Sin embargo, la detección de colisiones en robots delta paralelos es más compleja debido a las interacciones restringidas entre juntas y variables

a través de juntas universales o esféricas en la placa móvil.

Se han propuesto métodos de detección de colisiones basados en modelos, como el método de seguimiento basado en el método de momentos generalizados [3].

Estos métodos evitan las estimaciones de aceleración de la dinámica nominal del robot y la inversión de la matriz de inercia del robot, y pueden abarcar incertidumbres en el sistema. Otro enfoque es el observador de estado extendido (ESO) del marco de control activo de rechazo de perturbaciones (ADRC), que no requiere estimación de aceleración y es adecuado para sistemas robóticos de precisión.

En general, el documento presenta avances en las técnicas de diagnóstico de colisiones para robots delta paralelos, abordando la complejidad de las interacciones restringidas entre variables conjuntas e introduciendo un novedoso modelo de incertidumbre.

## **5. NIVEL DE INVESTIGACIÓN**

- Se plantea una investigación exploratoria debido a los pocos recursos informativos hallados respecto a la temática del modelamiento matemático y la conectividad de entre Python y Arduino para la ejecución de trayectorias y movimientos.

## **6. DISEÑO DE INVESTIGACIÓN**

- La estrategia que se adopta es experimental, ya que se realizarán diferentes pruebas y modelos de cinemática, programas e interfaces de conexión para garantizar el correcto y óptimo funcionamiento según las limitaciones físicas del robot.

## **7. INSTRUMENTOS DE RECOLECCIÓN DE DATOS**

- El instrumento de medición será únicamente la cámara, para observar la posición exacta en unidad de píxeles para posteriormente realizar el escalamiento a unidad de centímetros.

## 8. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

- El procesamiento de datos se hará a través de vectores de posición, provenientes de un arreglo en forma de cadena obtenida en Python, estos datos serán enviados en formato ASCII para su reprocesamiento y separación en Arduino.

## 9. ÁREA DE INVESTIGACIÓN

- Ingeniería y tecnología.

## 10. SUB ÁREA DE INVESTIGACIÓN

- Ingeniería electrónica y mecatrónica.

## 11. LÍNEA DE INVESTIGACIÓN

- Robótica y sistemas automáticos.

## 12. ASPECTOS ADMINISTRATIVOS

### 12.1. RECURSOS NECESARIOS

*Tabla 2: Recursos necesarios*

<b>Materiales</b>	<b>Precio (S/.)</b>
<b>Perfiles de aluminio (perfil 2020 de 1 m) x 6 unidades</b>	70.00
<b>Impresión 3D (Base fija y base móvil, elementos y unión y estructurales (soportes, etc.)</b>	80.00
<b>Servomotores de alta precisión (servo MG996R)</b>	120.00
<b>Cámara USB (Cámara Logitech C270)</b>	150.00
<b>Arduino MEGA</b>	70.00
<b>Cables de conexión y alimentación</b>	10.00
<b>Fuente de alimentación</b>	400.00
<b>Placa impresa</b>	22.30
<b>Estructura metálica</b>	200.00
<b>Impresión 3D (Faja y recolectores)</b>	300.00
<b>TOTAL</b>	<b>S/. 1422.30</b>

- **Considerando costos de servicios de luz y agua, se aumentará un costo de S/20.00 al total, haciendo una nueva cantidad de S/. 1442.30**

## CAPÍTULO II: MARCO TEÓRICO

### 1. ROBÓTICA

#### 1.1. DEFINICIÓN DE ROBOT

Barrientos et al. (2007) dice que “Por robot industrial de manipulación se entiende a una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la posición industrial, ya sea en una posición fija o en movimiento”.

Además, se puede decir que un robot es una máquina autónoma o semiautónoma que combina sensores para la percepción del entorno, sistemas de control para la toma de decisiones y actuadores para la realización de acciones físicas. Los robots son capaces de realizar determinadas tareas o secuencias de acciones de un modo predeterminado o manera flexible.

Un robot paralelo, incluso llamado manipulador paralelo, es un tipo específico de robot en el que varios brazos están conectados en paralelo a una plataforma en movimiento. Estos brazos suelen estar conectados a una base fija y una plataforma móvil para formar una estructura paralela.(Jara et al., 2008)

#### 1.2. CLASIFICACIÓN DE LOS ROBOTS

##### 1.2.1. SEGÚN SU MOVILIDAD

**Robots estacionarios:** Estos los robots se utilizan en entornos donde no se requiere movimiento. No se requiere como, por ejemplo, como líneas de montaje en fábricas. Son reconocidos renombrado por su precisión y coherencia por realizar tareas específicas.(Pinto et al., 2018)

*Figura 1: Robot estacionario.*



*Nota: EDS Robotics.*

**Robots móviles:** Estos robots son versátiles y pueden usarse en una variedad de aplicaciones, desde el móvil son autónomos ya que entregan paquetes. Abarcan hasta robots terrestres que exploran terrenos peligrosos o inaccesibles. (Pinto et al., 2018)

*Figura 2: Robot móvil.*



*Nota: Obtenida en <https://www.robotsmoviles.com/>.*

### 1.2.2. SEGÚN SU APLICACIÓN

- **Robots industriales:** Los robots industriales son esenciales para automatizar los procesos de fabricación porque aumentan la calidad y la eficiencia de la producción. (Jara et al., 2008)

*Figura 3: Robot industrial.*



*Nota: Made in China.*

- **Robots médicos:** En medicina, los robots pueden realizar de forma precisa, cirugías mínimamente invasivas y asistencia ayudar en la atención y rehabilitación del paciente. (García-Murillo et al., 2016)

*Figura 4: Robot industrial.*



*Nota: Obtenido de <https://www.fib.upc.edu/>.*

- **Robots de servicio:** Estos robots están hechos para ayudar a las personas con tareas diarias como limpiar pisos, cuidar a personas mayores o brindar atención al cliente en las tiendas. (Pinto et al., 2018)

*Figura 5: Robot de servicio.*



*Nota: Obtenido de <https://hyundai.pe/vitrina-hyundai/page/6/>*

- **Exploración robótica:** La exploración robótica se utiliza en proyectos de investigación científica como arqueología subacuática, oceanografía y exploración espacial. (Pinto et al., 2018)

*Figura 6: Robot de exploración.*



*Nota: Extraído de Curiosity rover en Marte.*

- **Robots militares:** En aplicaciones militares aplicaciones, los robots pueden realizar tareas que incluyen detonar explosivos, vigilar y reconocer enemigos en zonas de batalla. (Barrientos et al., 2007)

*Figura 7: Robot militar.*



*Nota:* <https://makeblock.com.ar/robots-militares-caracteristicas/>.

- **Robots para entretenimiento:** Estos robots proporcionan diversión y entretenimiento a través de interacciones lúdicas, como los robots educativos. (Barrientos et al., 2007)

*Figura 8: Robot de entretenimiento.*



*Nota:* <https://robots-argentina.com.ar/not/06/1550168.php>.

### 1.2.3. SEGÚN SU APARIENCIA Y ESTRUCTURA

Según Barrientos et al. (2007) tenemos:

- **Robots humanoides:** Estos robots están hechos para parecerse a los humanos y se utilizan con frecuencia en inteligencia artificial.
- **Robots sobre ruedas:** Los robots sobre ruedas se utilizan en aplicaciones que incluyen la búsqueda y el rescate de personas en áreas peligrosas y el transporte de carga en terrenos irregulares.
- **Robots serpiente:** Los robots serpiente son perfectos para moverse en espacios estrechos como tubérculos o panales.
- **Robots que vuelan:** Los drones y otros robots voladores se utilizan en una amplia gama de aplicaciones, desde entregar paquetes hasta inspeccionar infraestructura y tomar fotografías aéreas.

*Figura 9: Dron autónomo*



*Nota: Extraído de Tecnológico de Pabellón de Arteaga:*

### 1.3. MORFOLOGÍA DEL ROBOT

La morfología del robot se refiere a la estructura física y la apariencia exterior del robot, que varía significativamente según el tipo y función del robot. A

continuación, se ofrece una descripción general descripción de las partes y características morfológicas comunes que se pueden encontrar en varios tipos.(Jara et al., 2008)

### 1.3.1. SENSORES

Los robots suelen venir con una variedad de sensores que les permiten percibir sus alrededores. Estos sensores pueden ser cámaras, micrófonos, sensores ultrasónicos, sensores infrarrojos, sensores táctiles y más. La habilidad del robot para percibir e interactuar con su entorno depende de sus sensores, según (Barrientos et al., 2007) los sensores internos de un robot podemos clasificarlos como se muestra en la Tabla 1

*Tabla 3: Clasificación de sensores internos de un robot*

Presencia	Inductivo Capacitivo Efecto Hall Célula Reed	Óptico Ultrasonido Contacto
Posición	Analógicos	Potenciómetros Resolver Sincro Inductosyn LVDT
	Digitales	Encoder absolutos encoder incrementales Regla óptica
Velocidad	Taco generatriz	

Nota: Elaboración propia.

### 1.3.2. ACTUADORES

Según Pinto (Pinto et al., 2018) los actuadores son los encargados de generar el movimiento que posteriormente será transmitido por las articulaciones, existen distintos tipos de actuadores, estos pueden ser

accionados por distintos tipos de energía como: neumática, eléctrica e hidráulica, cada uno de estos presenta sus propias características.

*Figura 10: Actuator del tipo electro - neumático.*

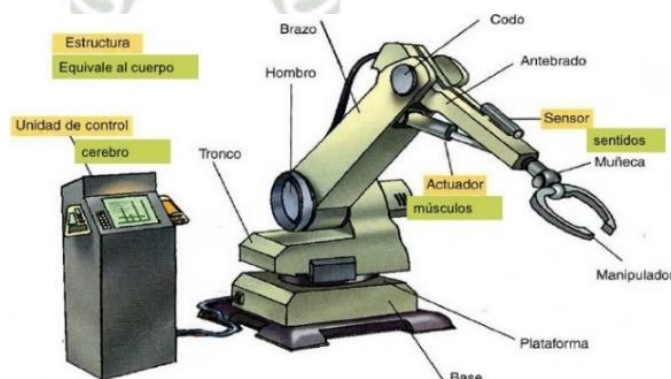


*Nota: Obtenido de <https://www.ebay.com/itm/186577074356>*

### 1.3.3. ESTRUCTURA MECÁNICA

Dependiendo de la solicitud, el diseño mecánico del robot cambiará. Podría ser un brazo articulado en un robot industrial, un chasis en un robot móvil, una estructura humanoide en un robot con forma humana o una estructura flexible en un robot un brazo articulado en forma de serpiente. El diseño de la estructura mecánica morfológica de la estructura permite ciertos movimientos y acciones. (Jara et al., 2008)

*Figura 11: Estructura mecánica.*



*Nota: Obtenido de <https://larobotica999975639.wordpress.com/>.*



### 1.3.6. FUENTE DE ENERGÍA

Para que los robots funcionen es necesaria una fuente de energía. Podría ser una batería, un cable eléctrico u otro tipo de sistema de suministro de energía. La elección de la fuente de la fuente de energía se basa en la movilidad del robot y la duración de la actividad.(Herrera Victor & Sanchez Nestor, 2011)

*Figura 13: Fuente de alimentación de 0-30 V y 0-10 A*



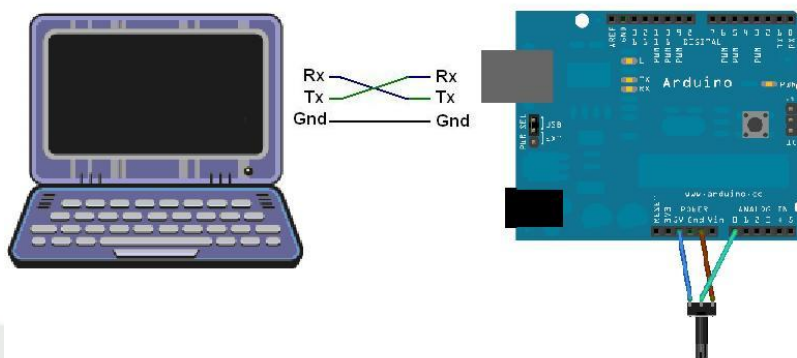
*Nota: Elaboración propia.*

*Nota: Empleada por su rango de corriente capaz de alimentar todos los motores a la vez.*

### 1.3.7. CONECTIVIDAD

Tienen capacidades de comunicación que les permiten interactuar con otros dispositivos o sistemas. Estas implican conectividad a redes Wi-Fi, Bluetooth o comunicación serial.(Herrera Victor & Sanchez Nestor, 2011)

Figura 14: Comunicación Serial



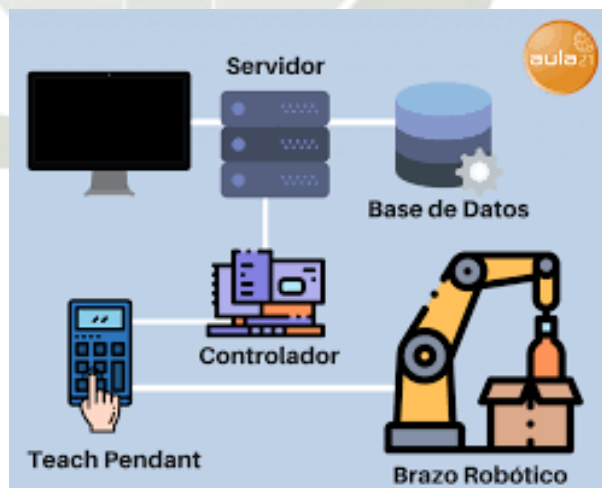
Nota: Extraído de <https://ardubasic.wordpress.com/2012/12/>.

Nota: Comunicación más apropiada, empleada con conexión directa.

### 1.3.8. INTERFAZ DE USUARIO

En algunos casos, los robots pueden tener una interfaz de usuario, como una pantalla táctil, botones o comandos de voz, que permite a los usuarios interactuar directamente con el robot o programarlo. (García-Samartín & Barrientos, 2023)

Figura 15: Interfaz de usuario.



Nota: Extraído de: <https://www.sicma21.com/robots-industriales-tecnologia-y-aplicaciones/>

### 1.3.9. CUBIERTA

(Jara et al., 2008) Nos dice “En muchos robots, se utiliza un cuerpo o cubierta para proteger los componentes internos y darle al robot una apariencia estética o funcional única”.

*Figura 16: Cubierta.*

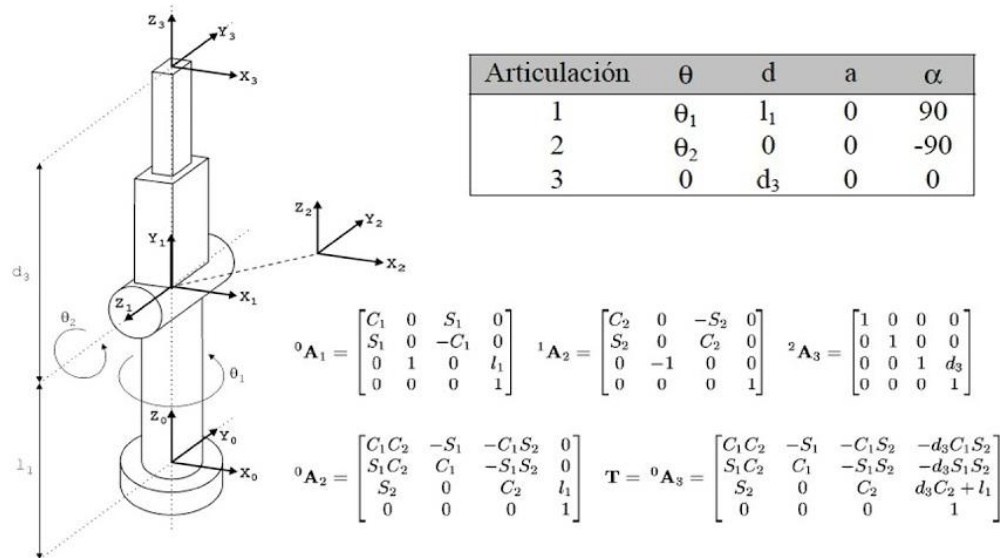


Nota: <https://kuka.my.site.com/s/product/kr-340-r34002/01t1i000000oxujAAA?language=es>

### 1.4. MÉTODOS DE LOCALIZACIÓN ESPACIAL

Según (Barrientos et al., 2007) “Se utilizan herramientas como las matrices de transformación homogénea, la cual es una herramienta clave en robótica y cinematografía que se utiliza para describir la posición y orientación de un sistema o componente móvil en un espacio tridimensional. Es una matriz 4x4 que combina datos sobre la rotación y traducción de un sistema en un solo conjunto de información”.

Figura 17: Denavit Hartenberg de un Robot



Nota: A. Barrientos - Fundamentos de la robótica (1997)

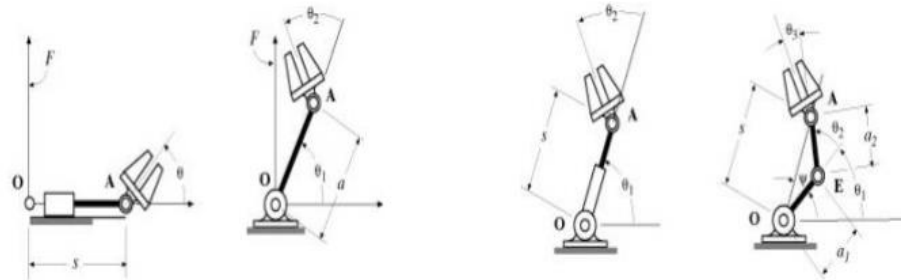
## 1.5. CINEMÁTICA DE UN ROBOT

### 1.5.1. CINEMÁTICA DIRECTA

El enfoque de la cinemática directa es determinar la posición y orientación de un objeto o sistema en relación con magnitudes físicas conocidas, como las longitudes de sus segmentos o las velocidades y aceleraciones de sus partes constituyentes. El uso de la cinematografía directa en robótica e ingeniería implica calcular la posición y orientación final de un robot o sistema mecánico en función de las posiciones y velocidades de cada una de sus partes individuales, como articulaciones o huesos. (Barrientos et al., 2007)

En conclusión, la cinemática directa proporciona una respuesta a la pregunta: "Dadas las condiciones iniciales y de trayectoria de un sistema, ¿qué posición y orientación asume el sistema en tres dimensiones?"

Figura 18: Cinemática directa.



Nota: Extraído de <https://goo.su/lluGd>

Además, esta herramienta también hace uso de la matriz de transformación homogénea

Figura 19: Estructura de Matriz de Transformación Homogénea

$$A_j^k = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Labels: Rotación (points to the top-left 3x3 matrix), Traslación (points to the top-right 3x1 column), Escalado global (points to the bottom-right 1x1 element), Perspectiva (points to the bottom-left 1x3 row).

Nota: Extraído de <https://goo.su/lluGd>

Nota: Empleada para describir movimientos de rotación y desplazamiento de una articulación.

Así mismo, según Barrientos se utiliza el algoritmo de Denavit-Hartenberg (Barrientos et al., 2007).

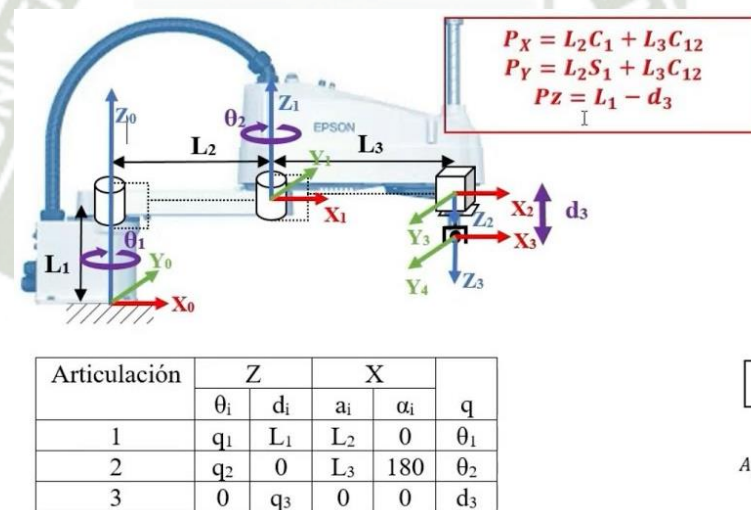
- Numerar los enlaces comenzando con 1 (primer enlace móvil de la cadena) y acabando con n (último enlace móvil). Se numerará como enlace 0 a la base fija del robot.

- Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad y acabando en n).
- Localizar el eje de cada articulación. Si es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- Para  $i$  de 0 a  $n-1$ , situar el eje  $Z_i$ , sobre el eje de la articulación  $i+1$ .
- Situar el origen del sistema de la base ( $S_0$ ) en cualquier punto del eje  $Z_0$ . Los ejes  $X_0$  e  $Y_0$  se situarán de modo que formen un sistema dextrógiro (sentido a las agujas del reloj) con  $Z_0$ .
- Para  $i$  de 1 a  $n-1$ , situar el sistema ( $S_i$ ) (solidario al eslabón  $i$ ) en la intersección del eje  $Z_i$  con la línea normal común a  $Z_{i-1}$  y  $Z_i$ . Si ambos ejes se cortasen se situaría ( $S_i$ ) en el punto de corte. Si fuesen paralelos ( $S_i$ ) se situaría en la articulación  $i+1$ .
- Situar  $X_i$  en la línea normal común a  $Z_{i-1}$  y  $Z_i$ .
- Situar  $Y_i$  de modo que forme un sistema dextrógiro con  $X_i$  y  $Z_i$
- Situar el sistema ( $S_n$ ) en el extremo del robot de modo que  $Z_n$  coincida con la dirección de  $Z_{n-1}$  y  $X_n$  sea normal a  $Z_{n-1}$  y  $Z_n$ .
- Obtener  $\theta_i$  como el ángulo que hay que girar en torno a  $Z_{i-1}$  para que  $X_{i-1}$  y  $X_i$  queden paralelos.
- Obtener  $d_i$  como la distancia, medida a lo largo de  $Z_{i-1}$ , que habría que desplazar ( $S_{i-1}$ ) para que  $X_i$  y  $X_{i-1}$  quedasen alineados.
- Obtener  $a_i$  como la distancia medida a lo largo de  $X_i$  (que ahora coincidiría con  $X_{i-1}$ ) que habría que desplazar el nuevo ( $S_{i-1}$ ) para que su origen coincidiese con ( $S_i$ ).

- Obtener  $\alpha_i$  como el ángulo que habría que girar en torno a  $X_i$  (que ahora coincidiría con  $X_{i-1}$ ), para que el nuevo ( $S_{i-1}$ ) coincidiese totalmente con ( $S_i$ ).
- Obtener las matrices de transformación  ${}^i A_i$ .
- Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot  $T = {}^0 A_1, {}^1 A_2, \dots, {}^{n-1} A_n$ .

La matriz  $T$  define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las  $n$  coordenadas articulares.

Figura 20: Tabla de DH de un robot:



Nota: Extraído de [https://www.youtube.com/watch?v=V\\_IleLJzR44](https://www.youtube.com/watch?v=V_IleLJzR44)

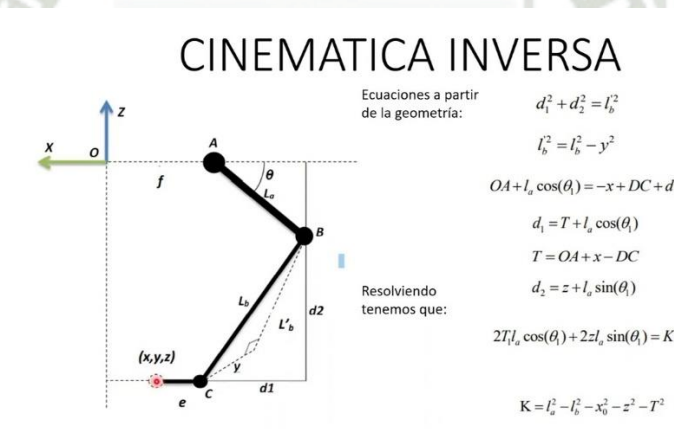
### 1.5.2. CINEMÁTICA INVERSA:

La cinemática inversa es un concepto utilizado en robótica y cinemática que se refiere al proceso de determinar las posiciones o configuraciones de las articulaciones de un robot para lograr una posición y orientación específicas en su extremo o efecto final (como una herramienta o una

pinza) en un espacio de trabajo determinado.(Garcia-Murillo et al., 2016)

El proceso de determinar las configuraciones articulares necesarias para lograr una determinada posición y orientación del efector final (la herramienta o extremo del robot) en el espacio de trabajo se conoce como cinematografía inversa. En otras palabras, dado un punto en el espacio donde se desea la extremidad del robot, la cinematografía inversa calcula las posiciones y ángulos de las articulaciones del robot necesarios para lograr esa ubicación y orientación.

Figura 21: Ejemplo de cómo se realizaría la cinemática inversa en un brazo del robot



Nota: Extraído de <https://www.youtube.com/watch?v=3UUwy99hbzg>.

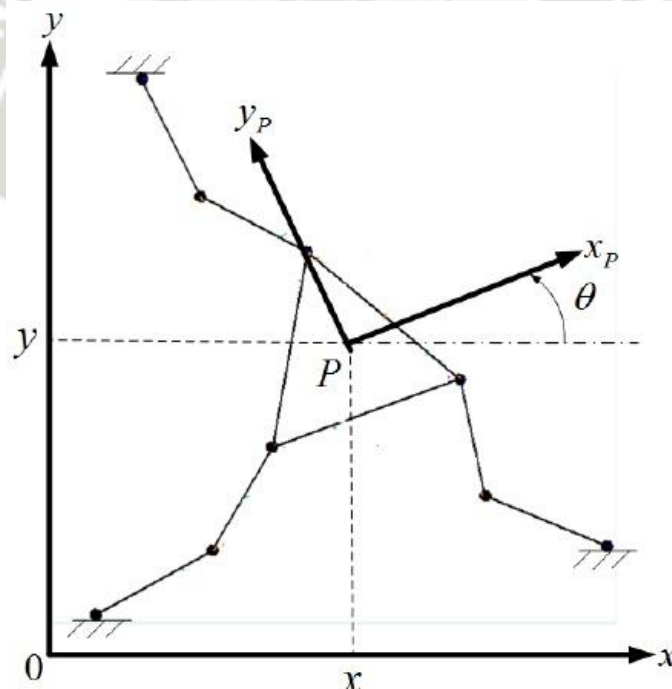
### 1.5.3.MODELO DIFERENCIAL:

Un "modelo diferencial" es un tipo de modelo matemático que se utiliza para describir cómo cambian las variables en relación con el tiempo u otra variable independiente. Estos modelos se basan en ecuaciones diferenciales, que son ecuaciones derivadas que se utilizan para

representar sistemas dinámicos en varias disciplinas como física, ingeniería, biología, economía y otras.

Un modelo diferencial especifica como cambian las variables en respuesta a umbrales de cambio. Los modelos pueden ser básicos o complejos, dependiendo de la naturaleza del sistema que se estudia. Por ejemplo, las ecuaciones diferenciales ordinarias (EDO) describen sistemas en los que solo existe una variable independiente, mientras que las ecuaciones diferenciales parciales (EDP) describen sistemas en los que las variables dependen de muchas variables independientes. (Barrientos et al., 2007)

Figura 22: Ejemplo de modelamiento diferencial



Nota: Extraído de <https://goo.su/XzHv>

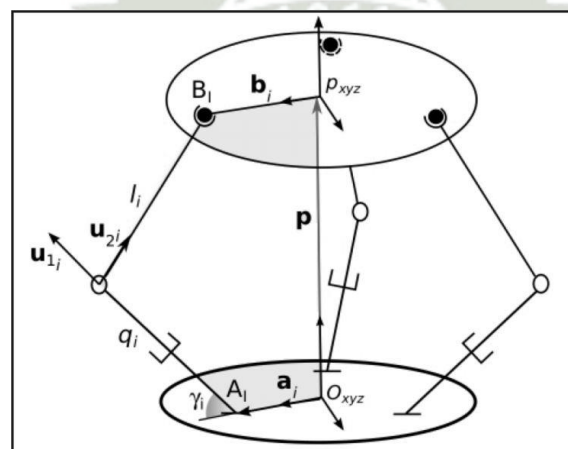
#### 1.5.4. MODELO DINÁMICO:

Según (Pinto et al., 2018) el modelo dinámico se define como:

El modelado dinámico es una técnica utilizada en muchos campos, incluida la ciencia, la ingeniería, la economía y la tecnología de la información, para describir y comprender el comportamiento de sistemas o procesos que cambian con el tiempo. En esencia, implica desarrollar modelos matemáticos o computacionales que capturen la evolución de un sistema a través del tiempo, teniendo en cuenta las interacciones e influencias que pueden cambiar con el paso del tiempo.

El modelado dinámico emplea ecuaciones diferenciales, simulaciones por computadora y otros métodos para describir cómo las variables del sistema cambian con el tiempo y cómo estas variables podrían interconectarse. Estos modelos pueden usarse para predecir el comportamiento futuro del sistema, analizar su estabilidad, optimizar su funcionalidad o tomar decisiones informadas sobre su control y gestión.

Figura 23: Ejemplo de modelamiento dinámico



Nota: Extraído de <https://goo.su/WeRvz>.

El modelo dinámico nos ayudará a determinar el tipo de actuador que necesitamos.

### 1.6. DEFINICIÓN DE ROBOT PARALELO:

Un robot paralelo, también conocido como manipulador paralelo o robot paralelo de cadena cinemática, es un tipo de dispositivo industrial o mecánico que consta de una plataforma móvil que se conecta a una base fija por muchos brazos o cadenas cinemáticas. La característica distintiva de un robot paralelo es que los extremos de los brazos están conectados tanto a la plataforma móvil como a la base fija, formando una estructura paralela.

En un robot paralelo, la plataforma móvil puede moverse en tres dimensiones variando las longitudes de los brazos o cadenas cinemáticas, mientras que la base fija permanece inmóvil. Esta configuración tiene importantes ventajas en términos de precisión, velocidad y capacidad de carga en comparación con otro tipo de robots, como los robots de serie (García-Murillo et al., 2016).

Los robots paralelos se utilizan en una variedad de aplicaciones, incluida la fabricación, la simulación de vuelo, la cirugía robótica, la construcción de maquinaria pesada y la investigación científica. Debido a su capacidad para proporcionar movimientos precisos y controlados, así como a su estabilidad inherente, son adecuados para aplicaciones que requieren un alto nivel de precisión y control.

Según (Pham & Kuo, 2023) tenemos los siguientes componentes en un robot paralelo tipo delta:

1. **Plataforma base:** La estructura fija o estacionaria del robot a la que están conectados los brazos móviles. La plataforma de cimentación proporciona

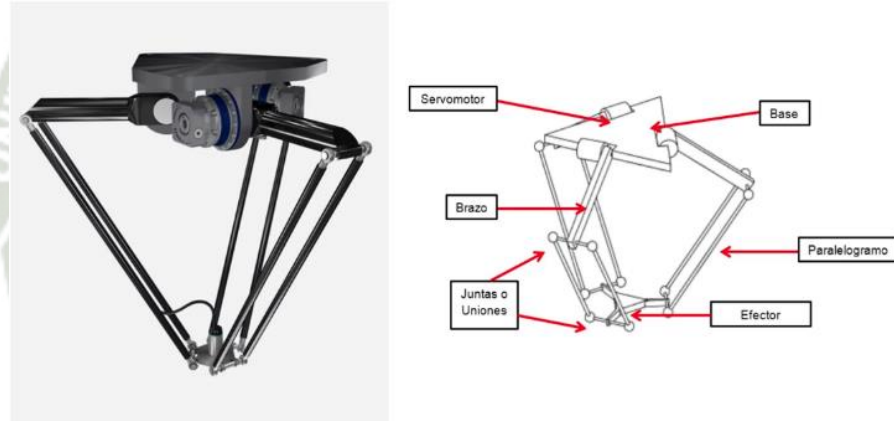
estabilidad al sistema y sirve como punto de referencia para el movimiento.

2. **Brazos móviles:** Los brazos móviles son componentes que se conectan a la plataforma base y se utilizan para realizar determinadas tareas. Dependiendo del diseño del robot, puede haber varios brazos conectados en paralelo o una combinación de brazos y piernas.
3. **Juntas:** Las juntas son los puntos de contacto entre los brazos móviles y la plataforma base. Estas conexiones permiten que los brazos se muevan en diferentes direcciones y ángulos. Los tipos comunes de juntas incluyen juntas rotativas (como las articulaciones esféricas y cardán), juntas prismáticas (que permiten el movimiento lineal) y juntas universales.
4. **Actuadores:** Los actuadores son los componentes encargados de generar movimiento en las articulaciones y, como resultado, en los brazos móviles. Los actuadores pueden ser eléctricos, hidráulicos o neumáticos, según la aplicación y los requisitos de carga.
5. **Sensores:** Los sensores son dispositivos que se utilizan para medir y monitorear muchos factores en el entorno y en el propio robot. Estos sensores pueden incluir sensores de posición, sensores de fuerza, sensores de velocidad, cámaras, sensores de proximidad, etc. Los datos recopilados por los sensores son fundamentales para el control y la retroalimentación del robot.
6. **Controlador:** El controlador es el cerebro del robot paralelo. Procesa la información del sensor y calcula los movimientos necesarios para

completar una determinada tarea. Podría ser un sistema de información con su propio software de control.

7. **Estructura y marco:** La estructura y el marco proporcionan soporte y rigidez al robot. Deben ser lo suficientemente resistentes para soportar las cargas y fuerzas generadas por las operaciones del robot.
8. **Herramienta final o efector:** El efector último es el extremo del brazo móvil que interactúa directamente con la pieza u objeto de trabajo. Puede ser un pasador, una herramienta de soldadura, una herramienta de corte o cualquier otro dispositivo diseñado para realizar determinadas tareas.

Figura 24: Robot Paralelo de la empresa UIROBOT



Nota: Extraído de <https://maquinariaycomponentes.com/robotica/robot-delta/>

## 2. VISIÓN ARTIFICIAL:

La visión artificial es un campo interdisciplinario que se encuentra en la intersección de las ciencias de la información, la ingeniería eléctrica, las matemáticas y la neurociencia. Su principal objetivo es crear sistemas y algoritmos computarizados que permitan a las máquinas "ver" y comprender el mundo visual de la misma manera que lo hacen los humanos. Esta capacidad se logra mediante la

adquisición, procesamiento y análisis de datos visuales, como imágenes y videos, con el objetivo de extraer información significativa, reconocer patrones, reconocer objetos y tomar decisiones basadas en la información visual.(Sharida & Hashlamon, 2021)

*Figura 25: Ejemplo de cómo se aplica la visión artificial en la industria*



*Nota: Extraído de <https://atriainnovation.com/en/blog/computer-vision-and-robots-the-perfect-couple/>*

Según Sharida (Sharida & Hashlamon, 2021) los componentes clave de la visión artificial son:

- a. **Adquisición de datos visuales:** Este paso implica recopilar datos visuales utilizando dispositivos como cámaras digitales, sensores de imagen, cámaras infrarrojas, telémetros láser y otros equipos de detección visual.
- b. **Pre procesamiento de imágenes:** En esta etapa, se utilizan técnicas para mejorar la calidad de las imágenes capturadas, incluidas la eliminación de ruido, la corrección de distorsión y el ajuste de contraste, para preparar los datos visuales para el análisis.
- c. **Segmentación de imágenes:** La segmentación de imágenes implica

subdividir una imagen en regiones más pequeñas o identificar objetos específicos en la imagen. Esto se logra mediante la detección de bordes, colores, texturas y otras características distintivas.

- d. **Extracción de características:** Una vez identificadas las regiones objetivo, se extraen características importantes como formas, colores, texturas y tamaños. Estas características sirven como representaciones numéricas de los objetos en la imagen.
- e. **Reconocimiento de patrón:** En esta etapa se utilizan algoritmos de aprendizaje automático y técnicas de reconocimiento de patrón para analizar características extraídas y tomar decisiones, como la clasificación de objetos o la detección de determinados patrones.
- f. **Interpretación y toma de decisiones:** Interpretar datos visuales implica comprender el significado de la información extraída y tomar decisiones basadas en ella. Esto puede incluir reconocimiento de objetos, detección de anomalías, navegación de robots autónomos y toma de decisiones en aplicaciones médicas, industriales y de seguridad.

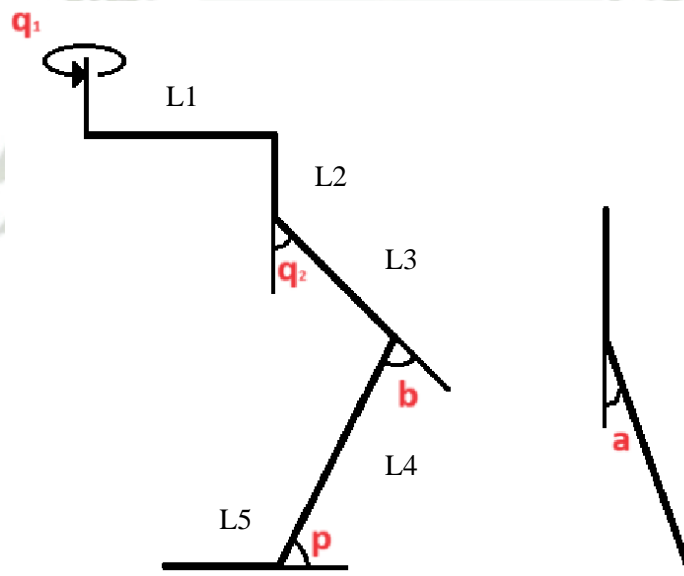
## CAPÍTULO III: IMPLEMENTACIÓN

### 1. MODELAMIENTO MATEMÁTICO

#### 1.1. CINEMÁTICA DIRECTA

Se llevó a cabo la cinemática directa del robot considerando los ángulos y las articulaciones que lo conforman. Para cada brazo se incluyó una articulación rotacional y dos esféricas, estando la articulación rotacional ubicada en la base superior del robot y a su vez siendo la única responsable de generar torque. Para resolver el problema, se utilizó el método de Denavit-Hartenberg, empleando los ángulos especificados en la Figura 26.

Figura 26: Distribución de ángulos por brazo



*Nota: Elaboración propia.*

Planteando así la solución matemática por Denavit Hartenberg en referencia a la Figura 27 con las posiciones y orientaciones de cada articulación que también se ven reflejadas en la Tabla 4 con los movimientos de rotación en Z, desplazamiento en Z, rotación en X y desplazamiento en X (Llerena Julio et al.,

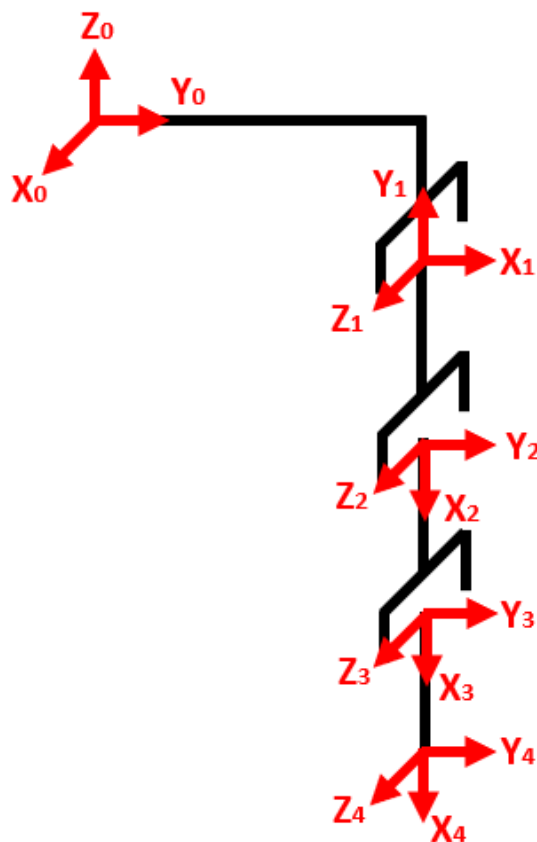
2024).

Tabla 4: DH del Robot Paralelo Tipo Delta

ART	$\theta$	$d$	$a$	$\alpha$
1	$q_1 + \frac{\pi}{2}$	$-L2$	$L1$	$\frac{\pi}{2}$
2	$q_2 - \frac{\pi}{2}$	0	$L3$	0
3	$-b$	$L4 \sin(\alpha)$	$L4 \cos(\alpha)$	0
4	$-p$	0	$L5$	0

Nota: Elaboración propia.

Figura 27:DH del Robot Paralelo Tipo Delta



Nota: Elaboración propia.

## 1.2. CINEMÁTICA INVERSA

La cinemática inversa inicialmente se intentó resolver por medio de matrices de transformación homogénea sin éxito por la complejidad del sistema, por lo que se optó por el método trigonométrico tomando en cuenta los mismos ángulos mencionados en la cinemática directa. Se tomó en consideración un brazo base y posteriormente añadiendo rotaciones de  $120^\circ$  en  $q_1$  para los otros dos brazos.

Ecuaciones que representan el brazo 1:

$$q_1 = 0^\circ \quad [1]$$

$$a = \text{asin}\left(\frac{x}{L_4}\right) \quad [2]$$

$$L_6 = |L_5 + y * \cos(q_1) - x * \sin(q_1)| \quad [3]$$

Es importante señalar que también tenemos restricciones en algunos ángulos dependiendo del área en la que se encuentren como se muestra:

$$s = \sqrt{(L_1 + L_6)^2 + (z + L_2)^2} \quad [4]$$

$$s = \sqrt{(L_1 - L_6)^2 + (z + L_2)^2} \quad [5]$$

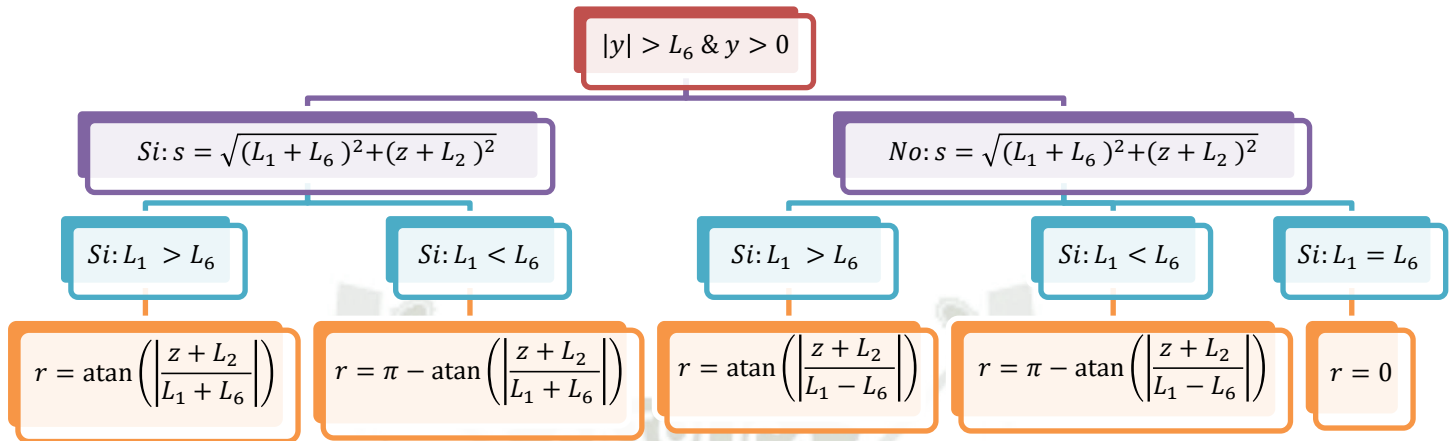
$$r = \text{atan}\left(\left|\frac{z + L_2}{L_1 + L_6}\right|\right) \quad [6]$$

$$r = \pi - \text{atan}\left(\left|\frac{z + L_2}{L_1 + L_6}\right|\right) \quad [7]$$

$$r = \text{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [8]$$

$$r = \pi - \text{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [9]$$

$$r = 0 \quad [10]$$



$$d = \frac{\operatorname{acos}(L_4^2 \cdot \cos^2(a) + s^2 - L_3^2)}{2 \cdot L_4 \cdot \cos(a) \cdot s} \quad [11]$$

$$p = r - d \quad [12]$$

$$q_2 = \pi + \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a) \cdot \cos(p) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a) \cdot \sin(p) - L_2 + z}\right) \quad [13]$$

$$q_2 = \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a) \cdot \cos(p) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a) \cdot \sin(p) - L_2 + z}\right) \quad [14]$$

$$L_4 \cdot \cos(a) \cdot \cos\left(\frac{\pi}{2} - p\right) > -z - L_2$$

$$\textit{Si: } q_2 = \pi + \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a) \cdot \cos(p) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a) \cdot \sin(p) - L_2 + z}\right)$$

$$\textit{No: } q_2 = \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a) \cdot \cos(p) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a) \cdot \sin(p) - L_2 + z}\right)$$

$$u = \text{atan}\left(\frac{\tan(a)}{\cos(p)}\right) \quad [15]$$

$$c = \sqrt{L_4^2 \cdot \sin^2(a) + L_4^2 \cdot \cos^2(a) \cdot \cos^2(p)} \quad [16]$$

Ecuaciones que representan el brazo 2:

$$q_1 = 120^\circ \quad [17]$$

$$i_2 = \left| \text{atan}\left(\frac{y}{x}\right) - 30^\circ \right| \quad [18]$$

$$k_2 = \sqrt{x^2 + y^2} \cdot \cos(i_2) \quad [19]$$

$$a_2 = \text{asin}\left(\frac{\sin(i_2) \cdot \sqrt{x^2 + y^2}}{L_4}\right) \quad [20]$$

$$a_2 = -\text{asin}\left(\frac{\sin(i_2) \cdot \sqrt{x^2 + y^2}}{L_4}\right) \quad [21]$$

$$a_2 = 0 \quad [22]$$

$$\text{Si: } y > \frac{x}{\sqrt{3}}$$

$$a_2 = \text{asin}\left(\frac{\sin(i_2) \cdot \sqrt{x^2 + y^2}}{L_4}\right)$$

$$\text{Si: } y < \frac{x}{\sqrt{3}}$$

$$a_2 = -\text{asin}\left(\frac{\sin(i_2) \cdot \sqrt{x^2 + y^2}}{L_4}\right)$$

$$\text{Si: } y = \frac{x}{\sqrt{3}}$$

$$a_2 = 0$$

$$L_6 = |L_5 + y * \cos(q_1) - x * \sin(q_1)| \quad [23]$$

Es importante señalar que también tenemos restricciones en algunos ángulos dependiendo del área en la que se encuentren como se muestra:

$$s_2 = \sqrt{(L_1 + L_6)^2 + (z + L_2)^2} \quad [24]$$

$$s_2 = \sqrt{(L_1 - L_6)^2 + (z + L_2)^2} \quad [25]$$

$$r_2 = \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 + L_6}\right|\right) \quad [26]$$

$$r_2 = \pi - \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [27]$$

$$r_2 = \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [28]$$

$$r_2 = 0 \quad [29]$$

$$\text{Si: } y > -x \cdot \sqrt{3} + 2 \cdot L_5$$

$$s_2 = \sqrt{(L_1 + L_6)^2 + (z + L_2)^2}$$

$$r_2 = \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 + L_6}\right|\right)$$

$$\text{Si: } y \leq -x \cdot \sqrt{3} + 2 \cdot L_5$$

$$s_2 = \sqrt{(L_1 - L_6)^2 + (z + L_2)^2}$$

$$r_2 = \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right)$$

$$\text{Si: } y \leq -x \cdot \sqrt{3} + 2 \cdot L_5 \text{ \& } L_1 < L_5$$

$$r_2 = \pi - \operatorname{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right)$$

$$\text{Si: } y \leq -x \cdot \sqrt{3} + 2 \cdot L_5 \text{ \& } L_1 = L_5$$

$$r_2 = 0$$

$$d_2 = \frac{\operatorname{acos}(L_4^2 \cdot \cos^2(a_2) + s_2^2 - L_3^2)}{2 \cdot L_4 \cdot \cos(a_2) \cdot s_2} \quad [30]$$

$$p_2 = r_2 - d_2 \quad [31]$$

$$q_2 = \pi + \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_2) \cdot \cos(p_2) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_2) \cdot \sin(p_2) + L_2 + z}\right) \quad [32]$$

$$q_2 = \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_2) \cdot \cos(p_2) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_2) \cdot \sin(p_2) + L_2 + z}\right) \quad [33]$$

$$L_4 \cdot \cos(a_2) \cdot \cos\left(\frac{\pi}{2} - p_2\right) > -z - L_2$$

Si:  $q_2$

$$= \pi + \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_2) \cdot \cos(p_2) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_2) \cdot \sin(p_2) + L_2 + z}\right)$$

No:  $q_2 = \operatorname{atan}\left(\frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_2) \cdot \cos(p_2) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_2) \cdot \sin(p_2) + L_2 + z}\right)$

$$u_2 = \operatorname{atan}\left(\frac{\tan(a_2)}{\cos(p_2)}\right) \quad [34]$$

$$c_2 = \sqrt{L_4^2 \cdot \sin^2(a_2) + L_4^2 \cdot \cos^2(a_2) \cdot \cos^2(p_2)} \quad [35]$$

Ecuaciones que representan el brazo 3:

$$q_1 = 240^\circ \quad [35]$$

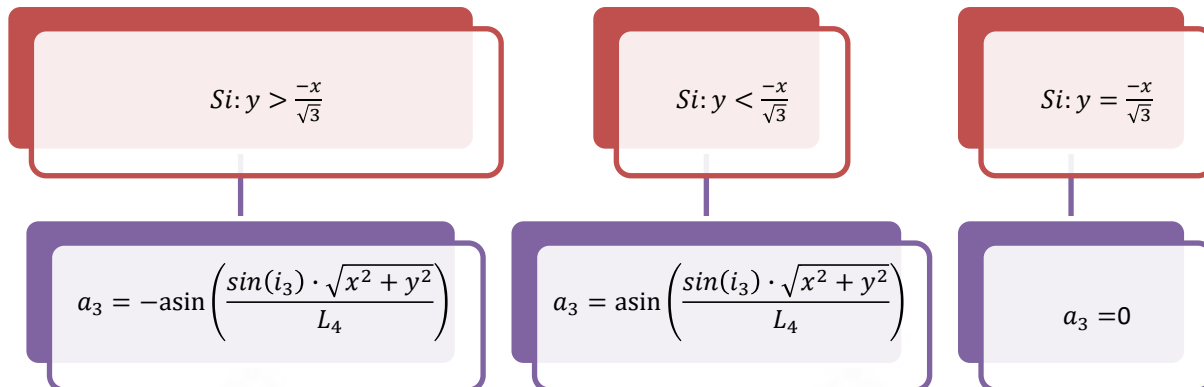
$$i_3 = \left| \operatorname{atan}\left(\frac{y}{x}\right) + 30^\circ \right| \quad [36]$$

$$k_3 = \sqrt{x^2 + y^2} \cdot \cos(i_3) \quad [37]$$

$$a_3 = \operatorname{asin}\left(\frac{\sin(i_3) \cdot \sqrt{x^2 + y^2}}{L_4}\right) \quad [38]$$

$$a_3 = -\operatorname{asin}\left(\frac{\sin(i_3) \cdot \sqrt{x^2 + y^2}}{L_4}\right) \quad [39]$$

$$a_3 = 0 \quad [40]$$



$$L_6 = |L_5 + y * \cos(q_1) - x * \sin(q_1)| \quad [41]$$

Es importante señalar que también tenemos restricciones en algunos ángulos dependiendo del área en la que se encuentren como se muestra:

$$s_3 = \sqrt{(L_1 + L_6)^2 + (z + L_2)^2} \quad [42]$$

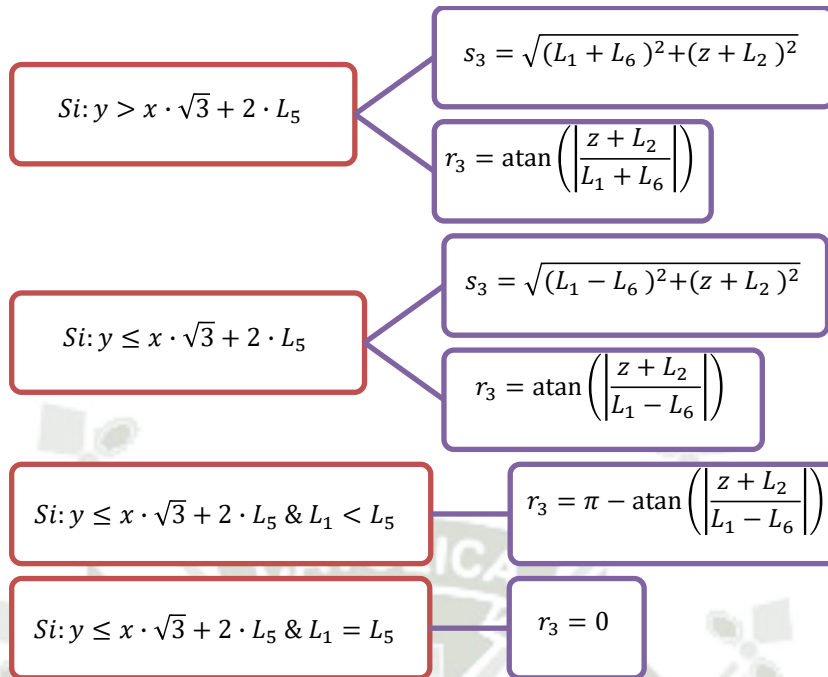
$$s_3 = \sqrt{(L_1 - L_6)^2 + (z + L_2)^2} \quad [43]$$

$$r_3 = \text{atan}\left(\left|\frac{z + L_2}{L_1 + L_6}\right|\right) \quad [44]$$

$$r_3 = \pi - \text{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [45]$$

$$r_3 = \text{atan}\left(\left|\frac{z + L_2}{L_1 - L_6}\right|\right) \quad [46]$$

$$r_3 = 0 \quad [47]$$

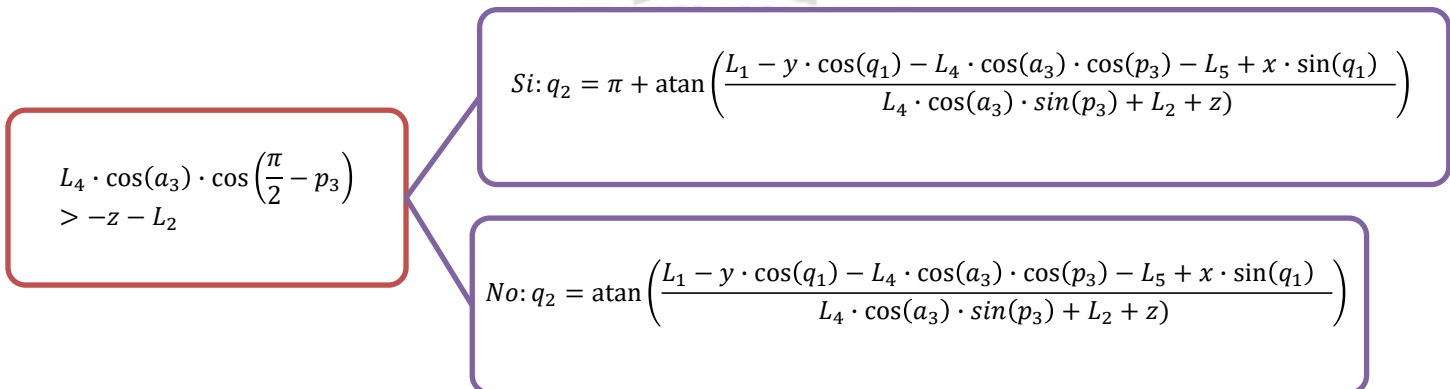


$$d_3 = \frac{\text{acos}(L_4^2 \cdot \cos^2(a_3) + s_3^2 - L_3^2)}{2 \cdot L_4 \cdot \cos(a_3) \cdot s_3} \quad [48]$$

$$p_3 = r_3 - d_3 \quad [49]$$

$$q_2 = \pi + \text{atan} \left( \frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_3) \cdot \cos(p_3) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_3) \cdot \sin(p_3) + L_2 + z} \right) \quad [50]$$

$$q_2 = \text{atan} \left( \frac{L_1 - y \cdot \cos(q_1) - L_4 \cdot \cos(a_3) \cdot \cos(p_3) - L_5 + x \cdot \sin(q_1)}{L_4 \cdot \cos(a_3) \cdot \sin(p_3) + L_2 + z} \right) \quad [51]$$



$$u_3 = \text{atan}\left(\frac{\tan(a_3)}{\cos(p_3)}\right) \quad [52]$$

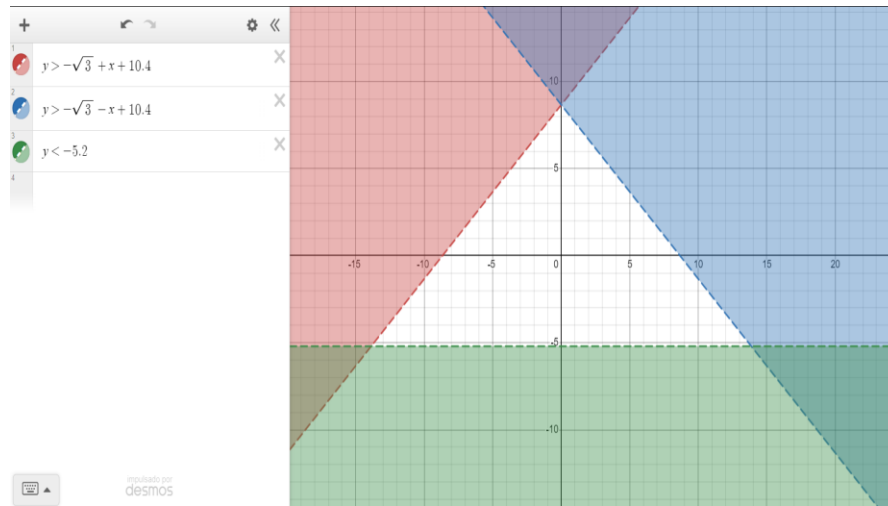
$$c_3 = \sqrt{L_4^2 \cdot \sin^2(a_3) + L_4^2 \cdot \cos^2(a_3) \cdot \cos^2(p_3)} \quad [53]$$

### 1.3. CONTRASTE DE DATOS EN MATLAB

Mediante la utilización de un software especializado como Matlab, se demostró de diferentes maneras la cinemática directa por MTH (Matrices de transformación Homogénea) y por Denavit Hartenberg donde los datos obtenidos debían ser idénticos, seguidamente con los mismos datos obtenidos, estos eran ingresados en las ecuaciones de cinemática inversa, donde precisamente teníamos que obtener los ángulos de rotación insertados al inicio como datos.

Es importante mencionar que en la cinemática directa se tenía que evaluar cada brazo por separado con la diferenciación mencionada de  $120^\circ$  en  $q_1$  y respetando áreas límite frente a la que surgían nuevas restricciones, es decir, todas las restricciones del sistema anteriormente mencionadas, se ven afectadas por cada una de las regiones sombreadas vistas en la Figura 28, que corresponden a la geometría de la base inferior, la cual es un triángulo equilátero con una altura media de 5.2 cm

Figura 28: Triangulo de restricciones del robot



Nota: Elaboración propia es el graficador DESMOS.

Realizaremos el siguiente ejemplo para la verificación del programa

$$[q_1, q_2, a, p] = [0, 122.0937, 14.2949, 79.8546]^\circ \rightarrow [X, Y, Z] = [10, 20, -30] \text{ cm:}$$

**MTH**

```
clear all; close all; clc;
L1=15.0;
L2=2.1;
L3=20.2;
L4=40.5;
L5=5.2;
q1=0*pi/180;
q2=122.0937*pi/180;
a=14.2949*pi/180;
p=79.8546*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
e=atan(l/o);
T01=DH(q1+pi/2, -L2, L1, pi/2)
```

```
T01 = 4x4
    0.0000    -0.0000    1.0000    0.0000
    1.0000     0.0000   -0.0000   15.0000
           0     1.0000    0.0000   -2.1000
           0         0         0     1.0000
```

```
T12=DH(q2-pi/2, 0, L3, 0)
```

```
T12 = 4x4
    0.8472   -0.5313     0   17.1130
    0.5313    0.8472     0   10.7324
         0         0   1.0000     0
         0         0     0   1.0000
```

T23=DH(-b,L4\*sin(a),L4\*cos(a),θ)

```
T23 = 4x4
   -0.6722    0.7403     0  -26.3822
   -0.7403   -0.6722     0  -29.0556
         0         0   1.0000  10.0000
         0         0     0   1.0000
```

T34=DH(-p,θ,L5,θ)

```
T34 = 4x4
    0.1761    0.9844     0    0.9160
   -0.9844    0.1761     0   -5.1187
         0         0   1.0000     0
         0         0     0   1.0000
```

```
T02=T01*T12;
T03=T01*T12*T23;
T04=T01*T12*T23*T34
```

```
T04 = 4x4
   -0.0000    0.0000    1.0000   10.0000
   -1.0000    0.0000   -0.0000   20.0000
   -0.0000   -1.0000    0.0000  -30.0000
         0         0         0    1.0000
```

### Denavid Hartenberg

```
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',θ,'a',L3,'alpha',θ);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',θ);
T4=Revolute('d',θ,'a',L5,'alpha',θ);
bot1=SerialLink([T1 T2 T3 T4], 'name', 'Robot_1');
ws=(-100 100 -100 100 -70 20);
bot1.fkine([q1+pi/2 q2-pi/2 -b -p])
```

```
ans =
     0         0         1         10
    -1         0         0         20
     0        -1         0        -30
     0         0         0         1
```

```
function [Q] = paraleloD(x,y,z)
L1=15.0;
L2=2.1;
L3=20.2;
L4=40.5;%40.5
```

```

L5=5.2;
% BRAZO 01
q1=0*pi/180;
a=asin(x/L4);
L6=abs(L5+y*cos(q1)-x*sin(q1));
if abs(y)>L5 && y<0
s=sqrt((L1+L6)^2+(z+L2)^2);
if L1>L6
    r=atan(abs((z+L2)/(L1+L6)));
end
if L6>=L1
    r=pi-atan(abs((z+L2)/(L1+L6)));
end
else
    s=sqrt((L1-L6)^2+(z+L2)^2);
if L1>L6
    r=atan(abs((z+L2)/abs(L1-L6)));
end
if L6>L1
    r=pi-atan(abs((z+L2)/abs(L1-L6)));
end
if L6==L1
    r=0;
end
end
d=acos((L4^2*cos(a)^2+s^2-L3^2)/(2*L4*cos(a)*s));
p=r-d;
if L4*cos(a)*cos(pi/2-p)>-z-L2
    q2=pi+atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-
L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
else
    q2=atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-
L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
end
u=atan(tan(a)/cos(p))*180/pi;
c=sqrt(L4^2*sin(a)^2+L4^2*cos(a)^2*cos(p)^2);
q2=q2*180/pi;
a=a*180/pi;
p=p*180/pi;

% BRAZO 02
q12=120*pi/180;
i2=abs(atan(y/x)-pi/6);
k2=sqrt(x^2+y^2)*cos(i2);
if y>x/sqrt(3)
    a2=asin((sin(i2)*(sqrt(x^2+y^2))/L4));
end
if y<x/sqrt(3)
    a2=-asin((sin(i2)*(sqrt(x^2+y^2))/L4));
end
if y==x/sqrt(3)
    a2=0;
end
end
L6=abs(L5+y*cos(q12)-x*sin(q12));
if y>-sqrt(3)*x+2*L5
    s2=sqrt((L1+L6)^2+(z+L2)^2);

```

```

        r2=atan(abs((z+L2)/(L1+L6)));
    end
    if y<=-sqrt(3)*x+2*L5
        s2=sqrt((L1-L6)^2+(z+L2)^2);
        r2=atan(abs((z+L2)/(L1-L6)));
    end
    if y<=-sqrt(3)*x+2*L5 && L1<L6
        r2=pi-atan(abs((z+L2)/(L1-L6)));
    end
    if y<=-sqrt(3)*x-2*L5 && L6==L1
        r2=0;
    end
    d2=acos((L4^2*cos(a2)^2+s2^2-L3^2)/(2*L4*cos(a2)*s2));
    p2=r2-d2;
    if L4*cos(a2)*cos(pi/2-p2)>-z-L2
        q22=pi+atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-
        L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
    else
        q22=atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-
        L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
    end
    u2=atan(tan(a2)/cos(p2))*180/pi;
    c2=sqrt(L4^2*sin(a2)^2+L4^2*cos(a2)^2*cos(p2)^2);
    q22=q22*180/pi;
    a2=a2*180/pi;
    p2=p2*180/pi;

    % BRAZO 03
    q13=240*pi/180;
    i3=abs(atan(y/x)+pi/6);
    k3=sqrt(x^2+y^2)*cos(i3);
    if y>-x/sqrt(3)
        a3=-asin((sin(i3)*(sqrt(x^2+y^2))/L4));
    end
    if y<-x/sqrt(3)
        a3=asin((sin(i3)*(sqrt(x^2+y^2))/L4));
    end
    if y== -x/sqrt(3)
        a3=0;
    end
    L6=abs(L5+y*cos(q13)-x*sin(q13));
    if y>sqrt(3)*x+2*L5
        s3=sqrt((L1+L6)^2+(z+L2)^2);
        r3=atan(abs((z+L2)/(L1+L6)));
    end
    if y<=sqrt(3)*x+2*L5
        s3=sqrt((L1-L6)^2+(z+L2)^2);
        r3=atan(abs((z+L2)/(L1-L6)));
    end
    if y<=sqrt(3)*x+2*L5 && L1<L6
        r3=pi-atan(abs((z+L2)/(L1-L6)));
    end
    if y>sqrt(3)*x+2*L5 && L1<L6
        r3=pi-atan(abs((z+L2)/(L1+L6)));
    end
    if y<=sqrt(3)*x-2*L5 && L6==L1

```

```

r3=0;
end
d3=acos((L4^2*cos(a3)^2+s3^2-L3^2)/(2*L4*cos(a3)*s3));
p3=r3-d3;
if L4*cos(a3)*cos(pi/2-p3)>-z-L2
q23=pi+atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-
L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
else
q23=atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-
L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
end
u3=atan(tan(a3)/cos(p3))*180/pi;
c3=sqrt(L4^2*sin(a3)^2+L4^2*cos(a3)^2*cos(p3)^2);
q23=q23*180/pi;
a3=a3*180/pi;
p3=p3*180/pi;

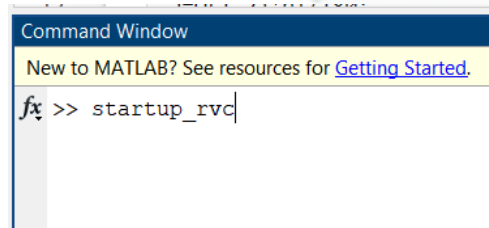
Q=[q2,a,p;q22,a2,p2;q23,a3,p3];
end

```

#### 1.4. SIMULACION CON TOOL BOX DE ROBÓTICA

Teniendo la seguridad de que tanto la cinemática directa como inversa indican el mismo punto cada quien desde su referencia, procedemos a utilizar el Tool Box de robótica el cual iniciaremos desde Matlab, estando en la carpeta rvctools y digitando startup\_rvc en el command window.

*Figura 29: Iniciar Tool Box de Robótica en la Comand Window*



*Nota: Elaboracion propia en el programa MATLAB*

Posteriormente ejecutamos el siguiente programa donde ingresaremos las medidas físicas de robot y las coordenadas XYZ donde queremos que este se ubique, tenga en cuenta que Z es una dimensión negativa ya que parte de la base superior del robot.

```
clear all; close all; clc;
L1=15.0;
L2=2.1;
L3=20.2;
L4=40.5;
L5=5.2;
x=10;
y=20;
z=-30;
[Q] = paraleloD(x,y,z)
```

```
Q = 3x3
    122.0937    14.2949    79.8546
     26.0400    17.7106    14.6393
     60.2542   -33.4441    31.9391
```

```
q2=Q(1,1)*pi/180;
a=Q(1,2)*pi/180;
p=Q(1,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot1=SerialLink([T1 T2 T3 T4],'name','Robot_1');
ws=(-100 100 -100 100 -70 20);
bot1.fkine([pi/2 q2-pi/2 -b -p])
```

```
ans =
     0     0     1     10
    -1     0     0     20
     0    -1     0    -30
     0     0     0     1
```

```
figure(1)
bot1.plot([pi/2 q2-pi/2 -b -p],'workspace',ws,'noname')
hold on
```

```
q2=Q(2,1)*pi/180;
a=Q(2,2)*pi/180;
p=Q(2,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
```

```
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot2=SerialLink([T1 T2 T3 T4],'name','Robot_2');
ws=[-100 100 -100 100 -70 20];
bot2.fkine([7*pi/6 q2-pi/2 -b -p])
```

```
ans =
    0.8660         0    -0.5000         10
    0.5000         0     0.8660         20
         0        -1         0        -30
         0         0         0         1
```

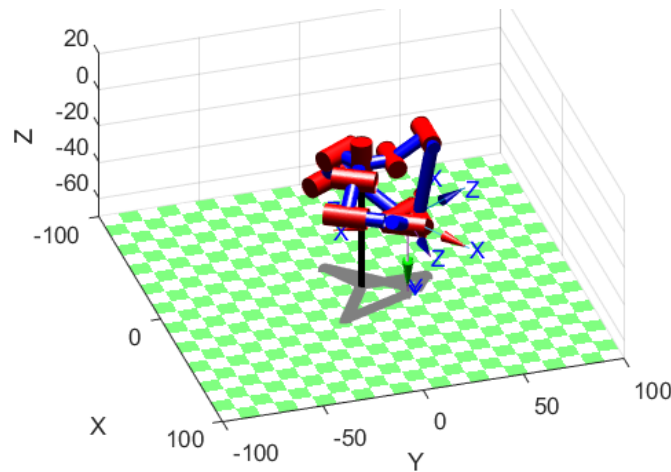
```
bot2.plot([7*pi/6 q2-pi/2 -b -p],'workspace',ws,'noname')
hold on
```

```
q2=Q(3,1)*pi/180;
a=Q(3,2)*pi/180;
p=Q(3,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot3=SerialLink([T1 T2 T3 T4],'name','Robot_3');
ws=[-100 100 -100 100 -70 20];
bot3.fkine([11*pi/6 q2-pi/2 -b -p])
```

```
ans =
   -0.8660         0    -0.5000         10
    0.5000         0    -0.8660         20
         0        -1         0        -30
         0         0         0         1
```

```
bot3.plot([11*pi/6 q2-pi/2 -b -p],'workspace',ws,'noname')
```

Figura 30: Representación Digital del Robot respondiendo a su cinemática



Nota: Elaboración propia en el programa MATLAB.

Lo que podemos apreciar en la Figura 30 es que los 3 brazos convergen en un mismo punto y que todas nuestras MTH coinciden en valores de la última columna que corresponda de a los valores de  $[x;y;z]=[10;20;-30]$  comprobando toda nuestra cinemática. A modo de ensayo se corroboraron varios puntos, obteniendo resultados satisfactorios en todo momento.

### 1.5. DIMENSIONAMIENTO DEL ROBOT

Con la intención de producir un robot semi industrial, se tomó en consideración un área regularmente mediana y el tamaño de los objetos a paletizar, siendo estos de un diámetro 3 cm, y considerando que se pondrán al menos 2 a 3 de estos a la par en una faja transportadora se optó por un área de trabajo máxima de 20 x 20 cm en la órbita más baja, de manera tal que se definieron las dimensiones de robot el función al área de trabajo a logras, tomando en consideración la órbita más baja como 49 cm a partir del eje de los motores hasta la parte baja de la base inferior del robot. En este proceso se obtuvieron como resultado las siguientes dimensiones tentativas distribuidas como se muestra en la Figura 26 y que serán

corroboradas con la dinámica del robot.

$$L1=15.0$$

$$L2=2.1$$

$$L3=20.2$$

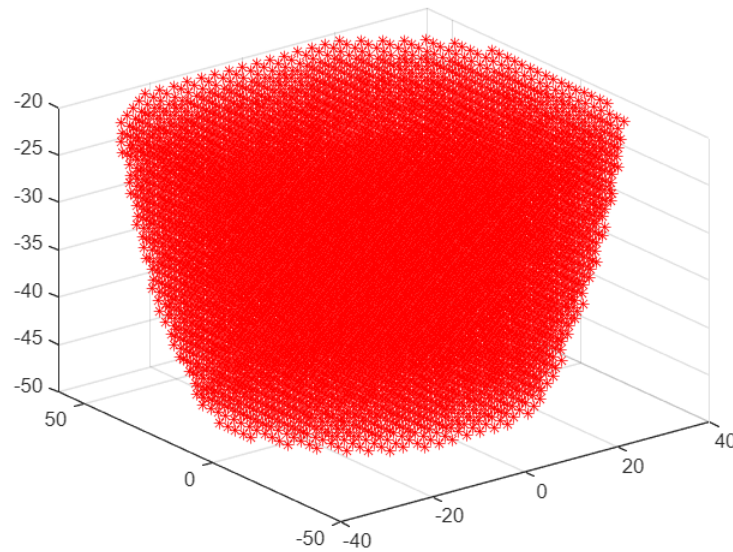
$$L4=40.5$$

$$L5=5.2$$

En cuanto al volumen de trabajo, se realizó un código donde se pretende verificar si el resultado de la posición es un valor real para poder definir con exactitud nuestro volumen de trabajo, se obtuvo una especie de paraboloides elíptico cortado en  $z=-49$  por los motivos explicados anteriormente, pero con un redondeo evidente llegando a  $z=-20$  por las mismas limitaciones físicas del robot como se muestra en las Figuras 31-33.

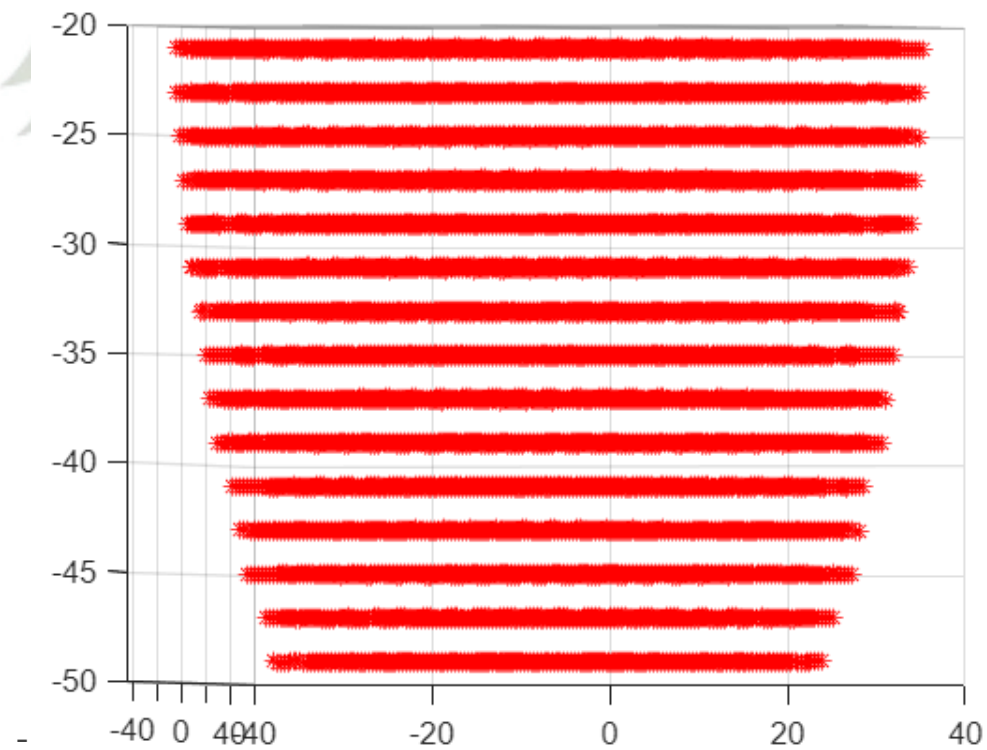
```
clear all; close all; clc;
L1=15.0;
L2=2.1;
L3=20.2;
L4=40.5;
L5=5.2;
for x=-60:3:60
    for y=-60:3:60
        for z=-49:2:-20
            [Q] = paraleloD(x,y,z);
            if isreal(Q(1,3))
                grid on
                plot3(x,y,z,'*r')
                hold on
            end
        end
    end
end
```

Figura 31: Representación isométrica del volumen de trabajo



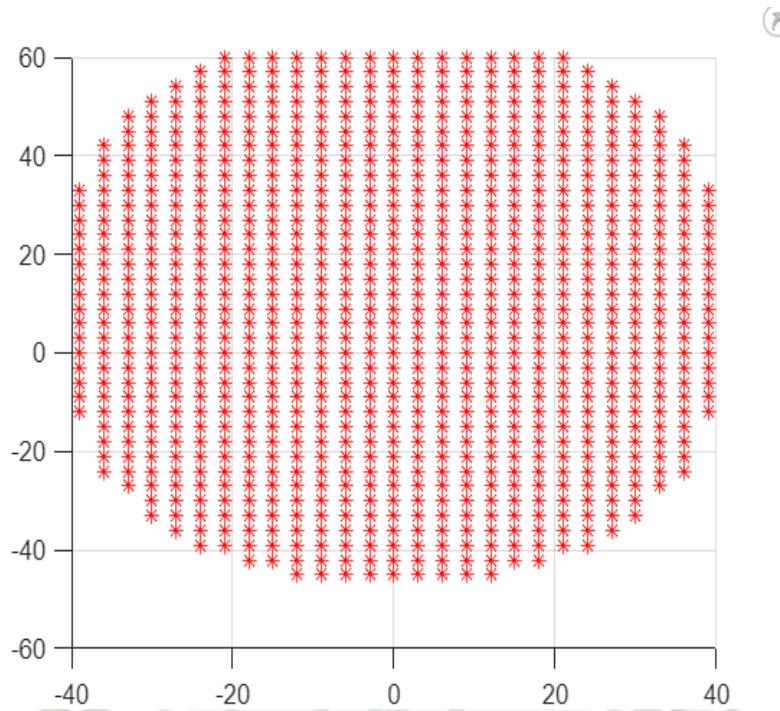
Nota: Elaboración propia en el programa MATLAB.

Figura 32: Vista frontal del volumen de trabajo limitada de  $Z = -49$  a  $-20$



Nota: Elaboración propia en el programa MATLAB.

Figura 33: Vista superior del volumen de trabajo limitada a un área de 40 x 40 cm

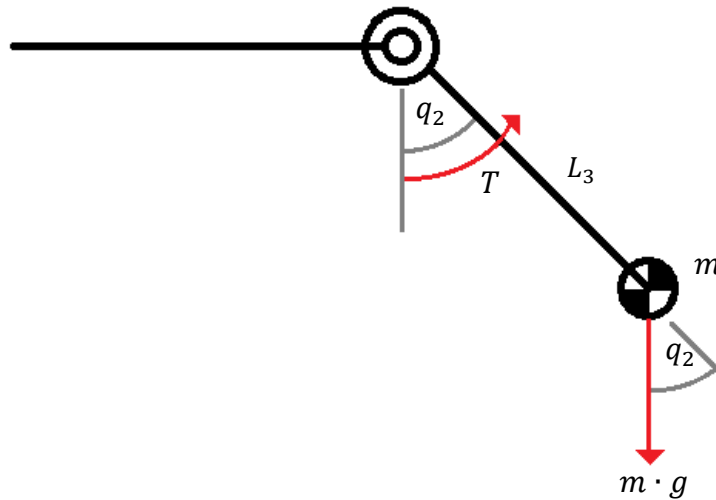


Nota: Elaboración propia en el programa MATLAB.

### 1.6. DINAMICA DEL ROBOT

Primeramente, se realizó el diagrama de cuerpo libre (DLC).

Figura 34: DCL de la dinámica del robot



Nota: Elaboración propia.

### 1.6.1. Método Newton-Euler

$$\sum T = \frac{d}{dt}(I \cdot \omega)$$

$$T = \frac{d}{dt}(I \cdot \dot{q}_2) + m \cdot g \cdot L_3 \cdot \text{Sin}(q_2)$$

$$I = m \cdot L_3^2$$

$$\frac{d}{dt}(I \cdot \dot{q}_2) = \frac{d}{dt}(m \cdot L_3^2 \cdot \dot{q}_2) = m \cdot L_3^2 \cdot \ddot{q}_2$$

$$T = m \cdot L_3^2 \cdot \ddot{q}_2 + m \cdot g \cdot L_3 \cdot \text{Sin}(q_2) \quad [54]$$

### 1.6.2. Método Lagrange

Verificamos las coordenadas en x & y respecto a la referencia que es en este caso la articulación de  $q_2$ . El signo dependerá de la orientación de nuestro sistema realizado en el Denavid Hartenberg.

$$x = L_3 \cdot \text{Sin}(q_2)$$

$$y = -L_3 \cdot \text{Cos}(q_2)$$

$$\dot{x} = L_3 \cdot \text{Cos}(q_2) \cdot \dot{q}_2$$

$$\dot{y} = L_3 \cdot \text{Sin}(q_2) \cdot \dot{q}_2$$

$$E_c = \frac{1}{2} \cdot m \cdot v^2$$

$$v^2 = \dot{x}^2 + \dot{y}^2$$

$$v^2 = (L_3 \cdot \text{Cos}(q_2) \cdot \dot{q}_2)^2 + (L_3 \cdot \text{Sin}(q_2) \cdot \dot{q}_2)^2$$

$$v^2 = L_3^2 \cdot \text{Cos}^2(q_2) \cdot \dot{q}_2^2 + L_3^2 \cdot \text{Sin}^2(q_2) \cdot \dot{q}_2^2$$

$$v^2 = L_3^2 \cdot \dot{q}_2^2$$

$$E_c = \frac{1}{2} \cdot m \cdot L_3^2 \cdot \dot{q}_2^2$$

$$E_p = -m \cdot g \cdot L_3 \cdot \text{Cos}(q_2)$$

$$L = E_c - E_p$$

$$L = \frac{1}{2} \cdot m \cdot L_3^2 \cdot \dot{q}_2^2 + m \cdot g \cdot L_3 \cdot \text{Cos}(q_2)$$

$$T = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}$$

$$\frac{\partial L}{\partial \dot{q}_2} = m \cdot L_3^2 \cdot \dot{q}_2$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) = \frac{d}{dt} (m \cdot L_3^2 \cdot \dot{q}_2) = m \cdot L_3^2 \cdot \ddot{q}_2$$

$$\frac{\partial L}{\partial q_i} = -m \cdot g \cdot L_3 \cdot \text{Sin}(q_2)$$

$$T = m \cdot L_3^2 \cdot \ddot{q}_2 + m \cdot g \cdot L_3 \cdot \text{Sin}(q_2) \quad [55]$$

Considerando de que el servomotor presenta velocidad constante correspondiente a 300°/s, nuestra aceleración sería 0 ( $\ddot{q}_2 = 0$ ) por lo que se redefine la ecuación de torque como:

$$T = m \cdot g \cdot L_3 \cdot \text{Sin}(q_2) \quad [56]$$

Teniendo esto claro se procedió a calcular el torque requerido y el torque máximo entregado, que evidentemente sería con el brazo totalmente estirado en la horizontal y con los siguientes datos:

$$m = 850 \text{ g}$$

$$g = 9.81 \frac{\text{m}}{\text{s}^2}$$

$$L_3 = 20.2 \text{ cm}$$

$$T = 0.7 \text{ kg} \cdot 9.81 \frac{\text{m}}{\text{s}^2} \cdot 0.202 \text{ m} \cdot \text{Sin}(90^\circ)$$

$$T_{\text{requerido}} = 1.68 \text{ N} - m$$

Se realizó la comparativa con el torque ofrecido por el servomotor que se interpretara como la masa que es capaz de levantar a la distancia de 1cm.

$$T_{motor} = 11 \text{ kg} - \text{cm}$$

Si consideramos la distancia a la cual se encuentra la masa tendríamos que a esa distancia sería capaz de levante determinada masa definida como:

$$m_{\text{capaz de levantar}} = \frac{11 \text{ kg} - \text{cm}}{20.2 \text{ cm}} = 545 \text{ g}$$

Por lo tanto, el torque máximo que es capaz de entregarnos a esa distancia sería:

$$T_{\text{maximo entregado}} = 1.08 \text{ N} - \text{m}$$

El torque máximo entregado al ser menor al requerido indica que es necesario que requiramos de un segundo motor que trabajara en paralelo para duplicar el torque aplicado teniendo como resultado un toque que es un 22% más elevado que el requerido, lo que es recomendado.

$$T_{\text{maximo entregado}} = 2.16 \text{ N} - \text{m}$$

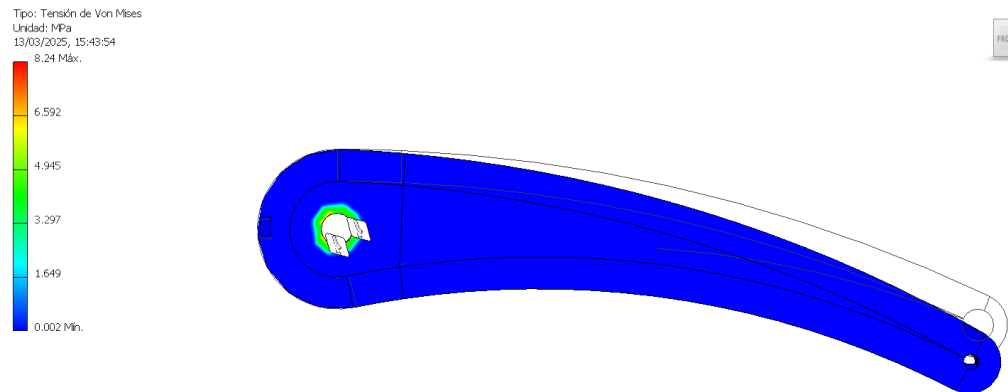
## 2. DISEÑO DEL ROBOT

### 2.1. PARTE PLASTICA

Teniendo finalmente definidas las dimensiones y componentes elegidos se procedió a realizar el modelado de las piezas en inventor por su versatilidad y facilidad de uso, además, se simularon las cargas previamente definidas en el mismo programa y en ANSYS WB para evaluar el material de impresión.

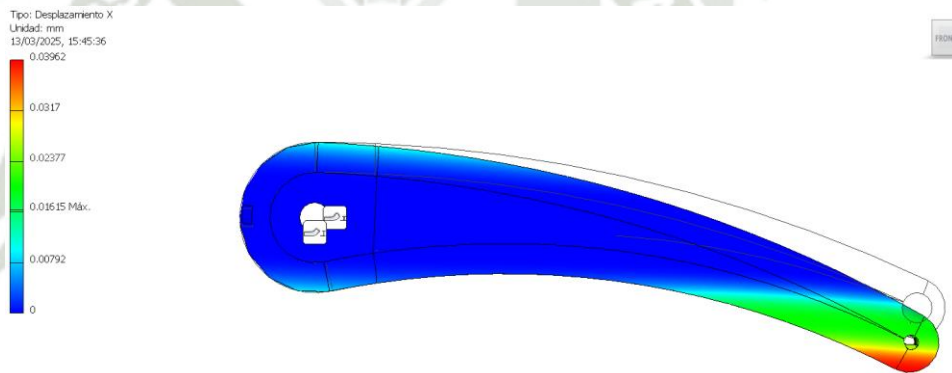
Principalmente se evaluó la deformación para ver si afectaría en la precisión del robot y tensión a la que se vería sometido.

*Figura 35: Tensión de Von Mises máxima de 8.24 MPa*



*Nota: Elaboracin propia en el programa MATLAB.*

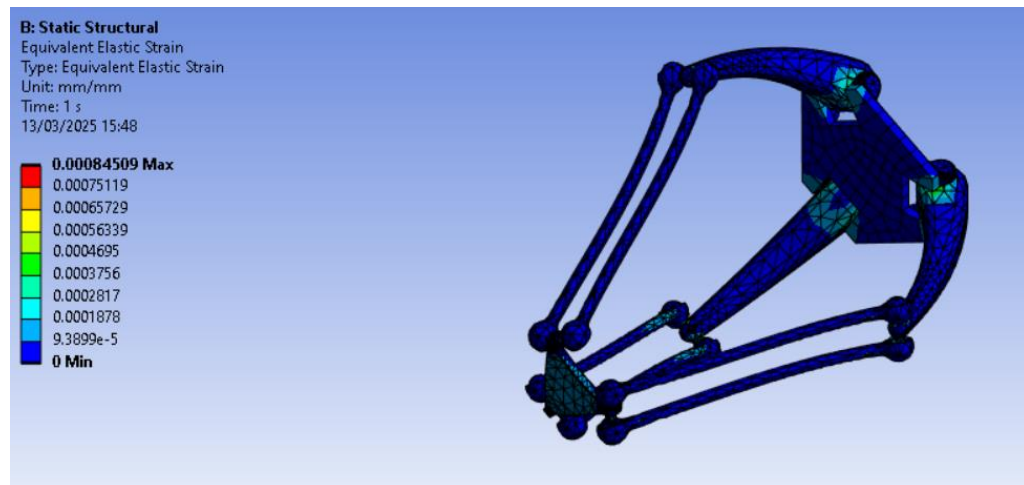
*Figura 36: Deformacin equivalente a un mximo de 0.016 mm*



*Nota: Elaboracin propia en el programa MATLAB.*

En los resultados de la figura 35 se puede apreciar que la tensin de Von Mises mxima es de 8.24 Mpa y se sabe que el lmite elstico del PLA que es el material tentativo es de 55 Mpa por lo que la eleccin es idnea, en cuanto a la deformacin equivalente vista en la figura 36 se ve que solo se deforma 0.039 mm, lo que resulta insignificante para nuestro propsito, por lo tanto, se define el PLA como nuestro material idneo para la manufactura de nuestro robot.

Figura 37: Deformación equivalente a un máximo de 0.008

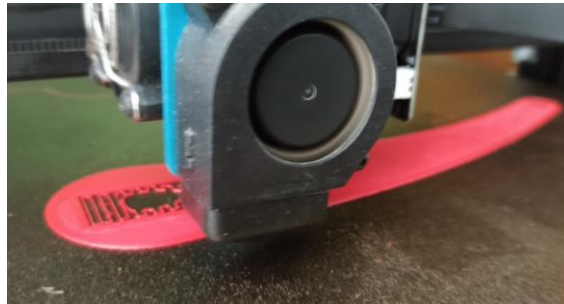


*Nota: Elaboración propia en el programa ANSYS.*

En la figura 37 se puede apreciar la deformación equivalente elástica del robot en conjunto en una posición más alargada con el propósito de evaluar cuanto este se deformará al momento de recoger el objeto y si tendremos problemas, y se ve que esta es de 0.00084 como máximo, lo que garantiza que toda la estructura cumplirá satisfactoriamente su función teniendo como los puntos lejanamente más afectados las uniones con los motores en la base superior del robot.

Las piezas fueron impresas en una impresora 3D Artillery X1 con PLA por su gran resistencia. Dentro de las principales partes se encuentran los 3 brazos que están conectados a los motores, la base superior e inferior, el actuador que cabe resaltar es de diseño propio y los cajones donde se almacenara los objetos.

*Figura 38: Impresión de brazo de robot paralelo*



*Nota: Elaboración propia.*

*Figura 39: Impresión de cajas receptoras.*



*Nota: Elaboración propia.*

## **2.2. PARTE METÁLICA**

En la parte metálica hablaremos de las varillas vistas en la figura 40 que nos servirán como la parte más larga del brazo, este componente se realizó de esta manera por su longitud y resistencia ya que el peso a levantar también se verá influenciado por este material, por lo que el material más económico y liviano

que se puede maquinar es el aluminio, estas piezas fueron maquinadas en el taller de manufactura de la Universidad Católica de Santa María.

*Figura 40: Barillas de aluminio manufacturadas*

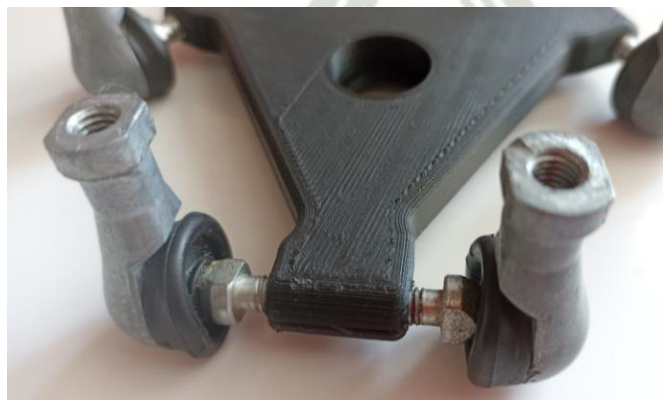


*Nota: Elaboración propia.*

En cuanto a las rotulas vistas en la figura 41, fueron adquiridas a través de mercado libre tras 20 días de espera, estas rotulas fueron determinantes al momento de dimensionar la rosca de las varillas, esta fue M6.

La estructura metálica, fue mandada a manufacturar por un técnico soldador, donde se especificaron medidas, priorizando la ubicación de la cámara y perfiles cuadrados para garantizar el buen soporte para la base superior la cual se sostenía como se muestra en la figura 42.

*Figura 41: Rotulas empernadas a base inferior*



*Nota: Elaboración propia.*

Figura 42: Imagen referencial de ubicación de la base superior del robot



*Nota: Elaboración propia.*

### 2.3. PARTE ELECTRÓNICA

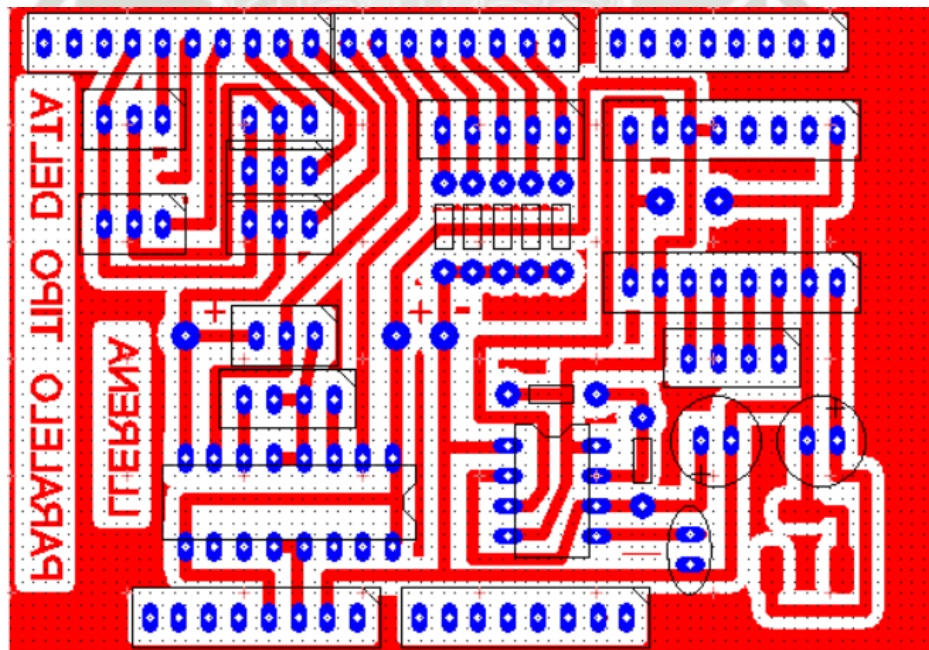
Se conectó los 6 servomotores en paralelo para que reciban todos el mismo voltaje, sin embargo hubieron complicaciones al momento de definir la fuente de alimentación, se probó con diversas fuentes recicladas de DVD, radios o incluso cargadores de teléfonos celulares pero ninguna cubría la demanda de corriente y voltaje solicitado por los servomotores ni la estabilidad necesaria para mantener constante la potencia consumida, se indago respecto a fuentes de alimentación y la fuente MESTEK regulable de 0 a 30 V y 0 -10 A con gran estabilidad de suministro emergentico fue la ideal para poder alimentar a los servomotores con un voltaje de 7.2 V y una corriente de 4 A.

Una vez distribuidos y escogidos los componentes a utilizar se empezó a realizar

la placa impresa donde se conectaron de manera adecuada y ordenada los componentes en el programa PCB WIZARD, se instalaron además puertos adicionales donde es posible conectar más componentes digitales.

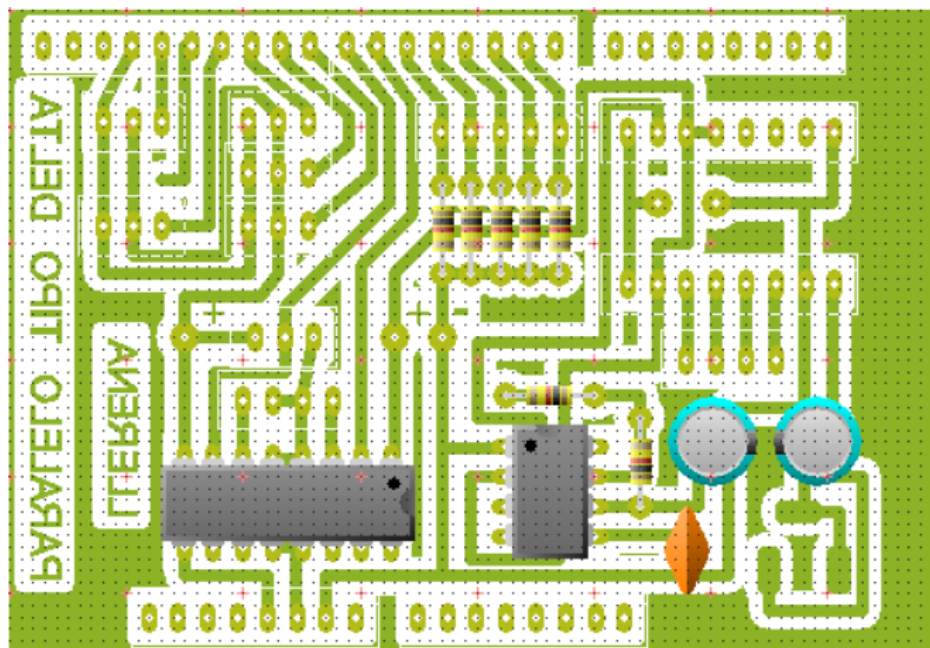
El programa PCB WIZARD nos permite realizar montajes de circuitos con gran precisión, además de ser bastante intuitivo de fácil uso, de esta manera es que podemos obtener diferentes vistas de nuestro circuito, la vista normal que es la que apreciamos en la imagen que dicho sea de paso es en la vista en la que se trabaja inicialmente, tenemos también la vista Real World que es una representación de como terminara nuestro circuito y por último la más importante que es con la que se procederá a realizar la manufactura que es la vista Artwork, que contiene únicamente las pistas que serán pasadas a la baquelita por el proceso de termo transferencia.

*Figura 43: Vista Normal de la placa impresa*



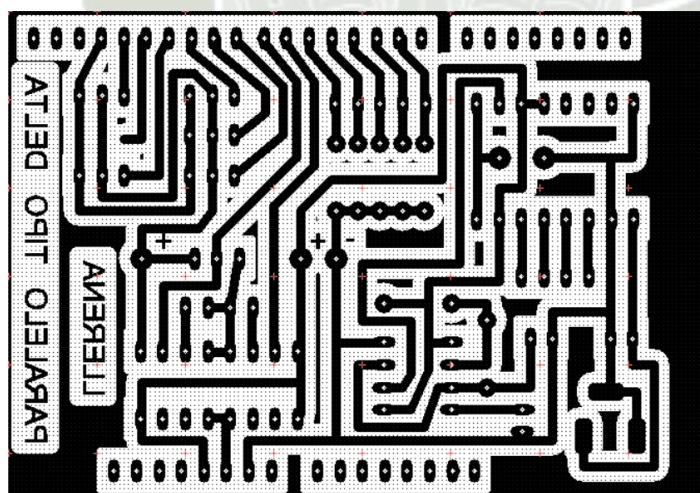
*Nota: Elaboración propia en el programa PCB Wizard.*

Figura 44: Vista Real World de la placa impresa



Nota: Elaboración propia en el programa PCB Wizard.

Figura 45: Vista Artwork de la placa impresa

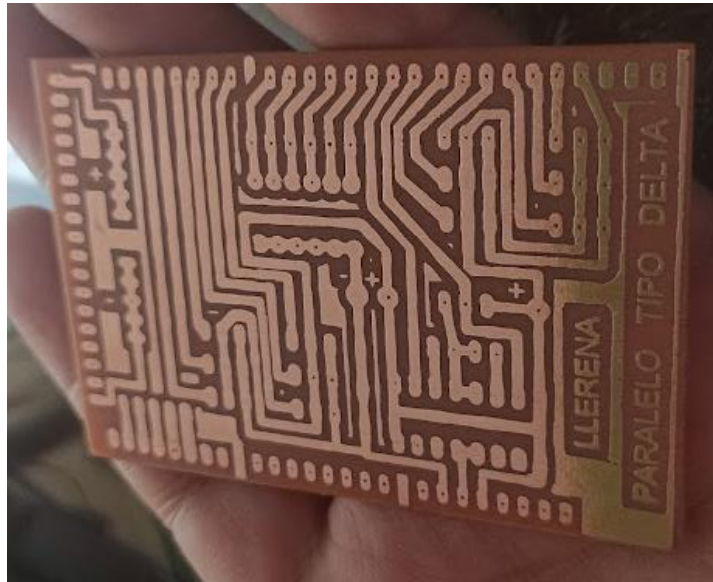


Nota: Elaboración propia en el programa PCB Wizard.

Una vez realizado el quemado de la placa obtendremos el resultado visto en la imagen 46 totalmente limpia, lo que procede ahora es realizar el taladrado y

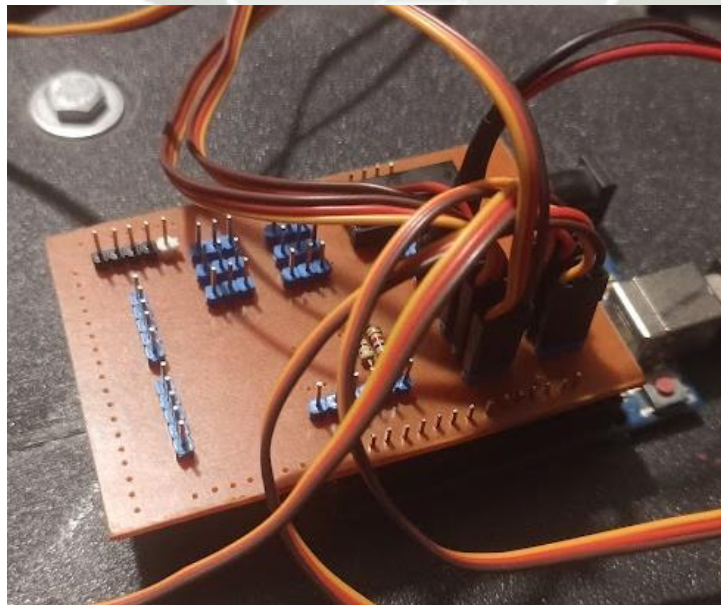
soldado de todos los componentes para tener todo el circuito montado y poderlo acoplar a nuestro Arduino como se muestra en la imagen 47.

*Figura 46: Placa impresa quemada*



*Nota: Elaboración propia.*

*Figura 47: Placa impresa instalada*



*Nota: Elaboración propia.*

Componentes empleados en placa impresa:

*Tabla 5: Componentes empleados en la placa impresa*

<b>Cantidad</b>	<b>Componente</b>	<b>Costo (S/.)</b>
<b>1</b>	CI L293D	7.00
<b>1</b>	CI 555	5.00
<b>7</b>	Resistencias de 1K $\Omega$	0.30
<b>3</b>	Espadines hembra	3.00
<b>1</b>	Espadines macho	1.00
<b>1</b>	Bornera	1.00
<b>1</b>	Entrada de 12 V para placa	1.50
<b>1</b>	Condensador de 100 uf	0.50
<b>1</b>	Condensador de 10 uf	0.50
<b>1</b>	Condensador de 100 nf	0.50
<b>TOTAL</b>		<b>20.30</b>

*Nota: Elaboración propia.*

#### 2.4. PROGRAMA EN ARDUINO

El desarrollo del robot se realizó en diferentes etapas:

- El primer programa se realizó tan solo con un servo para evidenciar que realmente puede ser capaz de levantar el peso mismo de su propio brazo.
- El segundo programa fue la coordinación de servos y establecer el punto 0 para garantizar que todos entiendan cuantos grados debían girar y en qué dirección hacerlo
- En el tercer programa se desarrolló en paralelo a la cinemática inversa contrastando los ángulos ingresados manualmente en el Arduino Mega para ver que efectivamente el punto deseado era el que se veía físicamente en el robot.
- El cuarto programa ya se realizó la cinemática inversa con la seguridad de

que el robot no sufriría fallas ni confusiones de ángulos y ecuaciones.

- Un programa que mencionáremos también es esta sección será el de la faja transportadora que trabajará con el Driver A4988 correspondiente al motor paso a paso.

En el transcurso de las pruebas se notó que en Arduino Uno quedaba corto de memoria, por lo que se tomó la decisión de realizar todo en el Arduino Mega que a su vez nos da una mayor disponibilidad de pines que fueron usados para cada uno de los servos, para los finales de carrera del actuador magnético, para el integrado LM93D (Puente H) , para el CI 555 (Timer usado en el motor paso a paso) y demás puertos de alimentación y puestos libres para posibles componente o indicadores que se puedan añadir en un futuro.

A continuación, se mostrará una comparativa entre Arduino Uno y Arduino Mega para un mayor entendimiento de sus cualidades.

*Tabla 6: Comparativa de Arduino Uno y Mega*

<b>Característica</b>	<b>Arduino Uno</b>	<b>Arduino Mega 2560</b>
<b>Microcontrolador</b>	ATmega328P	ATmega2560
<b>Voltaje de operación</b>	5V	5V
<b>Entradas/Salidas digitales (I/O)</b>	14 (6 PWM)	54 (15 PWM)
<b>Entradas analógicas</b>	6	16
<b>Memoria Flash</b>	32 KB (0.5 KB usado por bootloader)	256 KB (8 KB usado por bootloader)
<b>SRAM</b>	2 KB	8 KB
<b>EEPROM</b>	1 KB	4 KB

*Nota: Elaboración propia.*

## 2.5. DESARROLLO DE LA VISION ARTIFICIAL

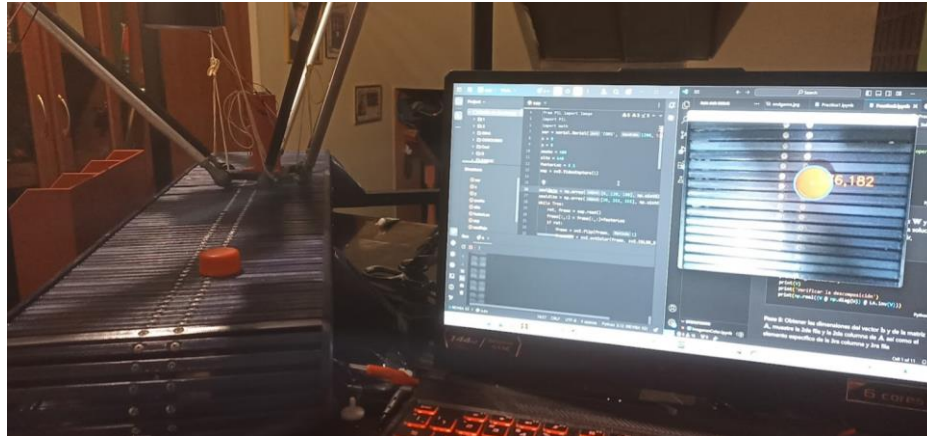
La visión artificial se inició definiendo la resolución de nuestra cámara, probablemente esta sea la parte más interesante de esta implementación, se planteó la idea de tomar cada pixel de la cámara como un punto coordinado en un espacio de 20 x 20 cm, evidentemente este dato fue escalado para trabajarlo de mejor manera y sea entendible para el robot, entonces, el programa hecho en Python visto en el Anexo 5, primeramente define el puerto de comunicación serial y las variables requeridas para armar 6 matrices diferentes, en las que se contendrá un nivel alto y un nivel bajo de cada color escrito de manera matricial, representando la Matriz(H), la Saturación(S) y el Brillo(V) en una matriz 1x3, posterior a esto se hace un escaneo rápido de todos los pixeles de la cámara y comprobando si alguno corresponde al color de alguno de los 3 rangos definidos anteriormente, mencionar también es este tendrá que cumplir condiciones de tamaño en un área de lectura específica que corresponde a 425 pixeles a partir del aparecimiento del objeto.

```
rojoBajo = np.array([0, 50, 150], np.uint8)
rojoAlto = np.array([25, 250, 250], np.uint8)
azulBajo = np.array([115, 0, 100], np.uint8)
azulAlto = np.array([170, 250, 250], np.uint8)
verdeBajo = np.array([35, 40, 140], np.uint8)
verdeAlto = np.array([70, 250, 250], np.uint8)
```

Una vez localizado, a través del puerto serie se enviará la coordenada en formato ASCII para ser recodificado por Arduino este pueda tomar los datos e interpretarlos de manera oportuna. Se determinó que la resolución no requiere ser muy elevada porque habrá un sobre exceso de datos y pixeles a evaluar, por

lo que una cámara de 480 x 640 pixeles fue suficiente para nuestro propósito y evitar el exceso de información.

*Figura 48: Aplicación de la visión artificial al robot*



*Nota: Elaboración propia.*

Como se aprecia en la figura 48, la cámara capta en exceso la luz, por lo que, en ocasiones, resulta que el proceso de lectura sea dificultoso, por lo que se recomienda que esto se trabaje en un ambiente con iluminación controlada y en función a la iluminación redefinir las matrices de color, de lo contrario estaremos teniendo malas lecturas, más adelante en el presente documento se profundizará respecto al tema en las pruebas y ensayos realizados.

### **3. DISEÑO DE LA FAJA TRASPORTADORA**

#### **3.1. PARTE PLASTICA**

El material utilizado para la elaboración de estos componentes fue PLA de color azul de la marca ESUN, por su longitud la faja transportadora fue impresa por tramos, así como los 126 eslabones que la componen, sus 4 patas, 3 retenedores inferiores y 4 superiores. Se diseñó además una catalina que engranará con cada uno de los eslabones como se ve en la figura 49.

*Figura 49: Sistema de tracción de la faja transportadora*



*Nota: Elaboración propia.*

Así como se tenía una catalina para la tracción, se añadió una catalina de retorno para evitar atascos al final de la faja ya que estaría a tensión y se le complicaría dar la vuelta sin ayuda, es por esa razón que se coloca esta catalina con rodamientos para que sirva de guía y pueda retornar la faja con normalidad sin sufrir atascos o sobre esfuerzos y desfases al motor paso a paso tal como se ve en la figura 50.

*Figura 50: Sistema de retorno de la faja transportadora*



*Nota: Elaboración propia.*

También fue necesario el diseño de una caja reductora vista en la figura 51 para

el motor paso a paso ya que su velocidad era excesiva para nuestro propósito, para esta se tomaron tolerancias y consideraciones para la sujeción directa del motor mediante bastones huecos por los que pasaban pernos de manera tal que este se sujetara de manera segura al motor y sin desalineamientos de ningún tipo.

*Figura 51: Cavidad de la caja reductora para motor paso a paso*



*Nota: Elaboración propia.*

Como se mencionó anteriormente, las guías para la faja fueron impresas por tramos debido a su longitud con una estructura interna triangular para asegurar su rigidez y optimización del diseño, teniendo en cuenta los puntos de fijación a la estructura como se ve en la imagen 52 los círculos concéntricos con una cruz inscrita en la circunferencia de mayor dimensión. De esta manera se buscó tener una unión firme y a la vez capaz de ser impresa en 3D debido a la complejidad del diseño, de otra manera esta hubiera salido con excesivos defectos.

Toda esta estructura fue empernada con un par de pernos punta broca por ambos extremos para garantizar la integridad del material y la precisión de los huecos en el metal para tener ambos lados perfectamente paralelos para que no

tengamos problemas con los eslabones o con atracos inesperados por una mala alineación.

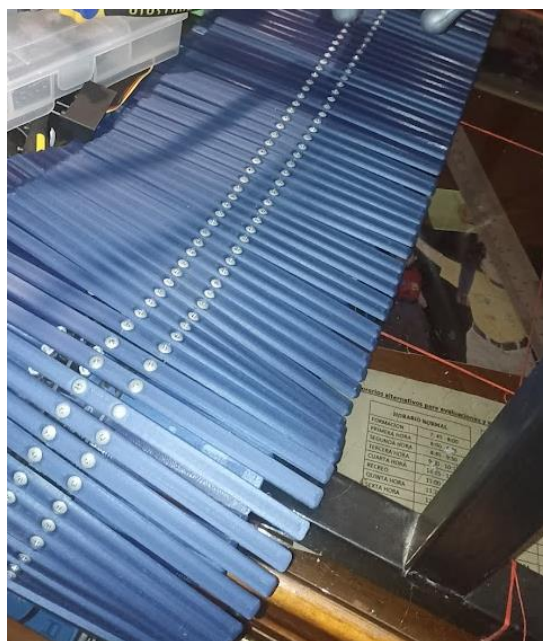
*Figura 52: Lateral de la faja transportadora ensamblado*



*Nota: Elaboración propia.*

En cuanto a los eslabones vistos en la figura 53 estos fueron emperrados a una cadena plástica que será la encargada de guiar a estos junto con las guías laterales para que se produzca un movimiento lineal y controlado, cada eslabón estuvo separa de las guías laterales 1.5 mm para que no este no se atore y para reducir la fricción entre eslabones y guías, un detalle a mencionar de la faja es de que presenta una masa de 950 g, lo que no complica el proceso pero quizá sea algo que se pueda mejorar a futuro.

*Figura 53: Eslabones de la faja transportadora ensamblados*



*Nota: Elaboración propia.*

Una vez se tengan todos los componentes se procedió al ensamble completo de la faja donde se aprecia uniformidad y solidez, que es precisamente es lo que se buscaba como se aprecia en la imagen 54.

*Figura 54: Faja transportadora ensamblada en su totalidad*



*Nota: Elaboración propia.*

### 3.2. PARTE ELECTRÓNICA

Inicialmente para las pruebas se realizaron con motorreductor para comprobar su movilidad, sin embargo, lo que se busca es tener una velocidad controlada por lo que se cambió este por un motor paso a paso, instalado a un extremo de la faja de donde saldrá su cable de alimentación hacia la placa principal donde se encuentra también su driver A4988. Cuando se colocó el motor paso a paso se vio que su velocidad era demasiada alta por lo que se tomó la decisión de diseñar el reductor de velocidad mencionado con anterioridad. Este consta de 3 etapas de reducción iguales, las cuales nos garantizaran una velocidad lineal de 19 cm/min, evidentemente cuando se diseñó el reductor se pensó en la lubricación que este tendría y la reducción del sonido y posible desgaste prematuro de los engranajes, por esa razón dentro de este se incorporó grada de engranajes que redujo el ruido y a su vez claramente el desgaste porque ya no chocaran plástico con plástico, alargando la vida útil de este componente.

*Figura 55: Motor paso a paso con caja reductora y catalina de traccion instalados*



*Nota: Elaboración propia.*

Figura 56: Reductor instalado en la faja



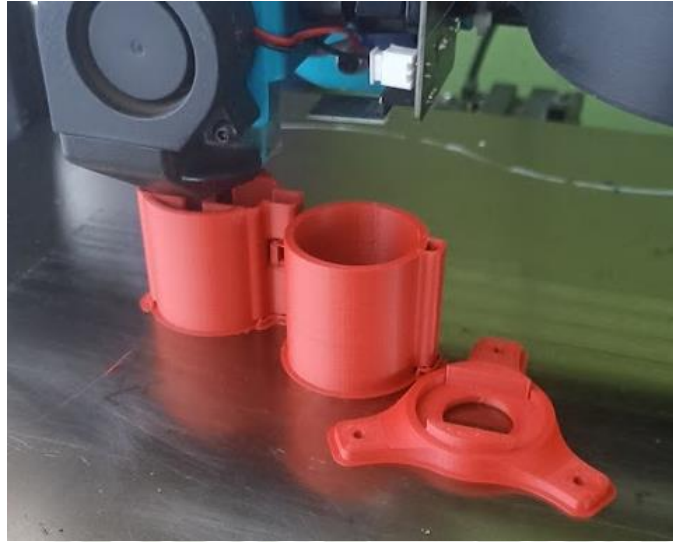
*Nota: Elaboración propia.*

#### 4. PRUEBAS Y ENSAYOS

##### 4.1. DISEÑO CAD Y PRUEBAS DE FUNCIONAMIENTO FÍSICO

Esta etapa nos ayudó a determinar la geometría adecuada y a verificar zonas de posibles colisiones en la trayectoria del robot según su área de trabajo, además se vio los detalles que se requería para la fabricación del actuador con finales de carrera mecánicos, de manera tal que este no tuviera ninguna obstrucción al momento de accionarse y sus cables salgan en un solo sentido para no entorpecer el proceso.




Figura 57: impresión de actuador recolector de objetos

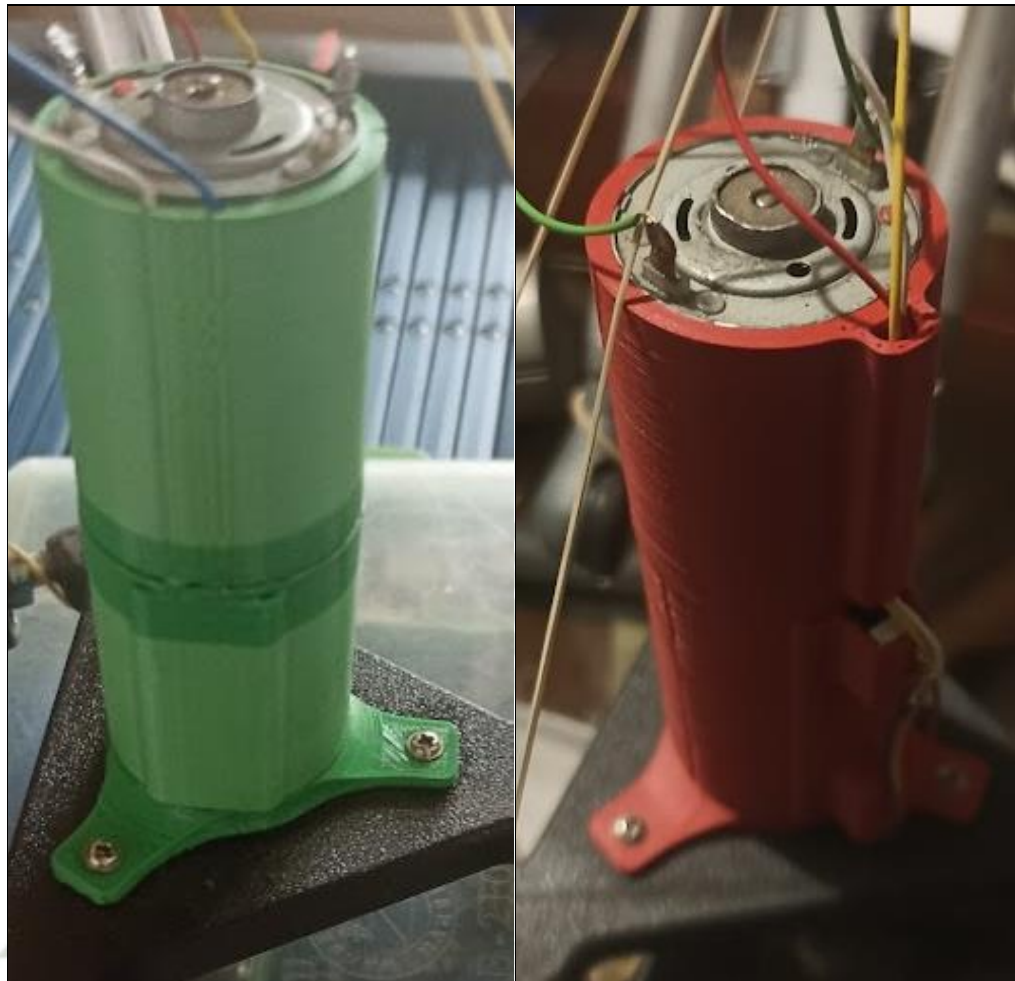


*Nota: Elaboración propia.*

En el caso del actuador se dieron 2 versiones, inicialmente se planteó con un final de carrera con pulsador, pero se tuvo muchos problemas porque cuando este impactaba en el pulsador, el tonillo que tenía el motor terminaba amarrándose y no permitía la retracción del actuador, por lo que se tomó la decisión de hacerlo con finales de carrera con lengüeta de paso, lo que facilitó el trabajo ya que no necesitaba llegar a impactar para detenerse, únicamente estar paralelo al accionador de los finales de carrera para poder lograr que el tornillo quede libre.

Tabla 7: Comparativa de versiones de actuador

ACTUADOR	
VERSION 01	VERSION 02
	
	
	



*Nota: Elaboración propia.*

#### **4.2. DETERMINACION DEL PUNTO 0 EN LOS MOTORES**

Esta tarea se realizó de manera sencilla realizando un programa inicial donde todos los servos serían colocados en  $0^\circ$  y colocando su cabezal en la misma posición, posteriormente se probó de  $0$  a  $180^\circ$  verificando que todos giraban coordinadamente, finalmente se emparejaron y cada servo trabajaría a la par para mover el robot completo, hacer la aclaración de que como se trabajaban en pares los servos, un servo tendría que trabajar de  $0$  a  $180^\circ$  y otro de  $180$  a  $0^\circ$  para que estos sean colocados en espejo y muevan un solo brazo. Si lo analizamos bien, nuestro punto 0 sería  $90^\circ$  ya que es este el que nos permitirá volver y avanzar

según sea el caso. Es por esa razón que a un motor se le suma y al otro se le resta.

#### 4.3. PRUEBA DE TRACCION CON UN SOLO BRAZO

Esta prueba que se ve en la figura 58 que se realizó fue para corroborar los cálculos de la dinámica del robot donde el juego de motores era capaz de levantar el peso de todos los brazos, para no vernos afectados con problemas de torque.

Esta prueba se realizó con el voltaje máximo de trabajo del servomotor MG996R que es 7.2 V y una corriente de 2A.

*Figura 58: Prueba de torque de servomotor*



*Nota: Elaboración propia.*

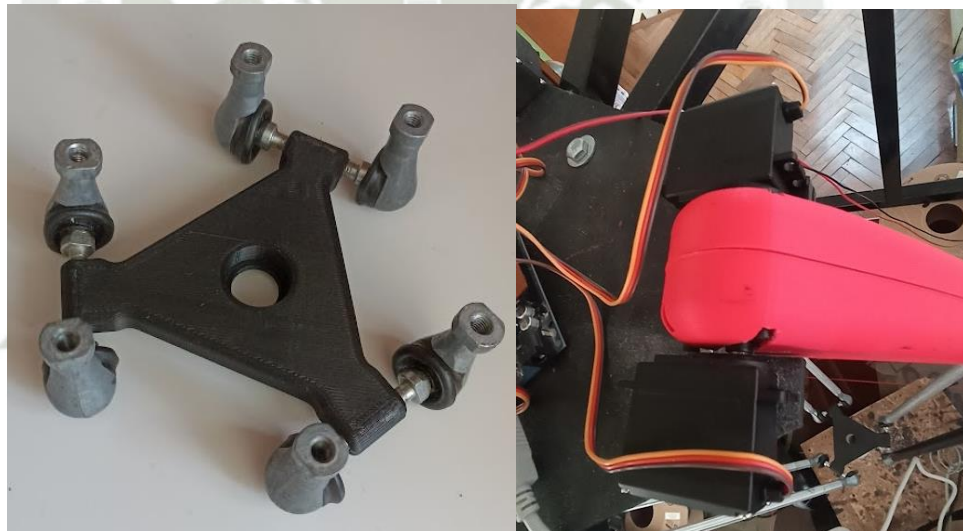
#### 4.4. ENSAMBLE DE ROBOT Y ESTRUCTURA METALICA

Hasta este punto ya se tiene definido dimensiones y forma de todo el robot, por lo que se imprimió la parte plástica en una impresora 3D Artillery X1 y se mandó a fabricar la estructura metálica, con excepción de la ubicación exacta del robot lo que se haría por cuenta propia para garantizar que el robot este en el lugar y

posición adecuada.

Cada uno de los brazos fue ensamblado cuidando que todos tengan idéntica dimensión para garantizar una buena precisión, junto con el ángulo correcto para cada motor para evitar sobre esfuerzos, para instalar las rotulas se dejó un hueco con la altura media de la rosca M6 para que esta pueda ser capaz de atornillarse con facilidad, pero a su vez que quede rígido ya que la misma rosca de la rótula crearía su canal en el interior.

*Figura 59: Ensamble de partes plásticas del robot*



*Nota: Elaboración propia.*

Se ubico la base superior equidistantemente con los 3 perfiles cuadrados para que este esté centrado y se procedió a perforar primeramente la base superior del robot, luego los perfiles cuadrados como se ve en la figura 60 para finalmente pasarle un perno de 3/8" con sus respectivas volandas para que quede bien sujeto a la estructura y no tengamos complicaciones en la precisión del robot y se emperno al soporte metálico 3 puntos equidistantes, finalmente se instaló el actuador en la base inferior cuidando que se encuentre centrado y sin ninguna

obstrucción para su libre movimiento.

*Figura 60: Preparación de estructura metálica*



*Nota: Elaboración propia.*

#### **4.5. VERIFICACION DE PUNTOS COORDENADOS MANUEALMENTE**

Solo con el robot montado se procedió a verificar su funcionamiento con ayuda de unos hilos como se muestra en la figura 61 que se colocaron con ayuda de una regla para garantizar su precisión con 4 puntos de referencia, seguidamente en el programa de Arduino de cinemática inversa se colocaron uno por uno los puntos de referencia para verificar que el robot realmente alcanzaba esos puntos y hacer los ajustes requerido en el caso no se obtenga un buen resultado, este ejercicio también nos ayudó a verificar los límites físicos reales que podía alcanzar el robot, se determinó un área de trabajo de 20 x 20 cm en la órbita más baja que es a nivel de la faja transportadora.

Figura 61: Robot paralelo ensamblado parcialmente y en su totalidad



Nota: Elaboración propia.

#### 4.6. VERIFICACION DE TRABAJO CON CAMARA A LAZO CERRADO

Una vez esclarecido que el robot es capaz de dirigirse a cualquier punto de su área de trabajo con una precisión de  $\pm 5\text{mm}$ , se introdujo la cámara y colocó una base blanca para tener mayor uniformidad, se realizaron pruebas con una pelota, la cual era movida manualmente como se ve en la figura 62, teniendo el robot como objetivo seguir dicha pelota siempre y cuando esté dentro del área de trabajo, de esta manera verificábamos que no existan puntos ciegos para la cámara ni para la cinemática, dicho de otra manera, el robot era capaz de dirigirse a cualquier punto dentro del área de trabajo sin complicación ni errores matemáticos.

Se realizaron pruebas con diferentes objetos de diferentes tamaños y colores para que de esta forma podamos limitar más las características del objeto que

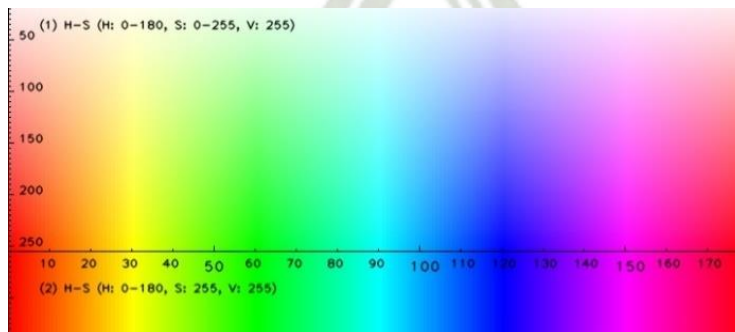
queremos reconocer, el tamaño de estos fue medido por la cantidad de pixeles que ocupaban y el color en HSB (color, saturación y brillo) según la figura 63.

*Figura 62: Prueba de movilidad a lazo cerrado del robot paralelo*



*Nota: Elaboración propia.*

*Figura 63: Representación grafica de los colores en HSB*



*Nota: Obtenida de una respuesta en <https://stackoverflow.com/>.*

En la figura 64 se puede apreciar cómo es de que los pixeles son leídos realizando un seguimiento bastante preciso y rápido, y justamente por esta razón es de que dentro del código se presenta una condición que consiste en que si el objeto identificado no se mueve un promedio de 25 pixeles a la redonda, no se realizara una nueva lectura, de esta manera se evitó problemas de vibraciones en el robot que se tuvo inicialmente, estas eran producidas por el exceso de datos enviados al robot, de manera tal que al mínimo cambio se realizaba un cálculo nuevo y el robot empezaba a cambiar rápidamente de ángulos y por consecuente a calentar de sobremanera los motores.

*Figura 64: Prueba de seguimiento del objeto a tiempo real*



*Nota: Elaboración propia.*

#### **4.7. PRUEBA DE MOBILIDAD CON FAJA INDEPENDIENTE**

Antes de realiza run proceso convino se optó por primeramente tener a la faja en óptimas condiciones, inicialmente se probó con un motorreductor simple de 5 v y sin una catalina de retorno, sin embargo, se sabía que se tenía que tener una velocidad constante y controlada, por lo que se decidió cambiar el motor por un motor paso a paso, el cual está controlado por su driver A4988, cabe resaltar que para que este opere era necesario que le lleguen pulsos altos y bajos constantemente pero esto no podíamos añadirlo como subrutina en nuestro

programa de Arduino porque se perdían datos y tiempo, entonces se decidió tomar el motor paso a paso como un circuito a parte gobernando su velocidad con un timer 555 en modo monoestable para obtener pulsos continuos.

Un problema que se observó en este proceso fue de que la faja se atoraba en las partes del retorno por la tracción y la fricción que se generaba, para evitar este problema se optó por la utilización de una catalina de retorno para evitar el apiñamiento de los eslabones de la faja, además de lubricar con aceite de máquina.

Cuando se planteó el uso de un motor paso a paso, evidentemente se notó que la velocidad de este era excesivamente alta, por lo que necesariamente se tenía que adaptar un reductor de velocidad para este, que también se diseñó e imprimió en 3D junto con su caja, fue una rección en 3 etapas donde también se le añadió grasa de engranajes para mejorar la fricción que sufría el diseño y de esa manera tener una velocidad lenta y controlada para una mejor predicción de la posición del objeto en función a la velocidad lineal de la faja de 19 cm/min.

#### **4.8. COLOR Y POSICION CON LA FAJA EN MOVIMIENTO**

Cuando se realizó el programa en Python, se impuso un condicional donde delimita el área de la cámara, que corresponde a una franja con la altura del objeto a reconocer de tal manera de que cuando este aparezca completamente en la cámara sea leído una sola vez y más aún gracias a la condición hecha en Arduino donde se especifica que una vez ingresada una lectura, esta no se volverá a realizar a menos que el objeto se desplace más de 22 y 29 pixeles en el eje Y & X respectivamente que corresponde a 1 cm en la vida real.

Al tener la faja a velocidad constante, es fácil determinar la posición de los objetos con tan solo calcular el tiempo que le toma al robot desplazarse, de tal manera que ya contamos con tiempos exactos de desplazamiento.

Es importante mencionar que la iluminación de los objetos tiene que ser controlada, para evitar variaciones en las tonalidades de color y este pase por alto los objetos que pasan por la cámara.



## CONCLUSIONES:

- PRIMERA: Se logró diseñar e implementar un robot paralelo tipo Delta integrado con un sistema de visión artificial para la clasificación de objetos mediante un proceso de paletizado. El sistema incluye un actuador magnético retráctil que permitió la recolección y depósito de objetos en tres ubicaciones distintas, cumpliendo satisfactoriamente con las restricciones de velocidad de faja de 60 cm/min, el área de trabajo de 20 x 20 cm y la capacidad de cargar una masa de 100 g.
- SEGUNDA: Se obtuvo de manera satisfactoria el modelo matemático completo del robot paralelo tipo Delta, desarrollando las ecuaciones correspondientes a la cinemática directa e inversa, así como a su dinámica. Estas formulaciones fueron estructuradas de forma general, permitiendo su adaptación a cualquier robot Delta con configuración geométrica similar.
- TERCERA: A través del uso de Python y bibliotecas de visión artificial, se implementó un sistema capaz de discriminar objetos por tamaño a partir de la cantidad de píxeles detectados en la imagen y por color, utilizando el modelo HSV con rangos establecidos para cada tono. Este proceso resultó eficiente y confiable en condiciones controladas.
- CUARTA: Se realizaron simulaciones de posicionamiento en distintos puntos dentro del volumen de trabajo del robot utilizando MATLAB. Dichas simulaciones fueron validadas física y matemáticamente, arrojando resultados consistentes y satisfactorios con el comportamiento esperado del sistema robótico con una tolerancia de  $\pm 1$  cm.
- QUINTA: La integración de visión artificial con un robot paralelo tipo Delta demostró ser una solución viable y eficiente para tareas de clasificación y paletizado en entornos automatizados. Además, la estructura modular del sistema, el uso de código abierto (Python y Arduino) y la documentación técnica generada durante el proyecto ofrecen una base replicable para futuras implementaciones en la industria nacional.

**RECOMENDACIONES:**

- Se debe tomar en cuenta que la iluminación sea en un ambiente controlado, es decir, sin variaciones de intensidad en la luz, para garantizar que los colores sean inconfundibles en diferentes ambientes.
- Es recomendable realizar calibraciones periódicas del sistema de visión artificial para mantener la precisión en la identificación de tamaños y colores. Esto incluye el ajuste de parámetros como enfoque, balance de blancos, brillo y contraste, así como la actualización de los rangos HSV utilizados.
- Se recomienda integrar protocolos de seguridad como botones de parada de emergencia, límites de velocidad en movimientos y sensores de presencia humana. Esto asegurará una operación segura del sistema en entornos industriales, especialmente si se trabaja en proximidad con operarios.
- Para asegurar la robustez del sistema en condiciones reales, se aconseja realizar pruebas constantes con objetos variados en tamaño, forma y color. Estas pruebas permitirán detectar posibles fallos del sistema de visión o del algoritmo de posicionamiento, y facilitarán su corrección y mejora continua.
- Para garantizar el funcionamiento correcto se recomienda a futuro implementar un sistema de control a lazo cerrado donde se corrijan trayectorias y posiciones de los objetos.

**REFERENCIAS BIBLIOGRÁFICAS**

- Barrientos, A., Peñin Luis, Balaguer Carlos, & Aracil Rafael. (2007). *Fundamentos de la Robótica*.
- Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*.
- Garcia-Murillo, M. A., Sanchez-Alonso, R. E., & Gallardo-Alvarado, J. (2016). Kinematics and dynamics of a translational parallel robot based on planar mechanisms. *Machines*, 4(4). <https://doi.org/10.3390/machines4040022>
- García-Samartín, J. F., & Barrientos, A. (2023). Kinematic Modelling of a 3RRR Planar Parallel Robot Using Genetic Algorithms and Neural Networks. *Machines*, 11(10), 952. <https://doi.org/10.3390/machines11100952>
- Gosselin, C. , & A. J. (1990). *A comparative study of mechanical architectures for parallel robotic manipulators*. *Robotics and Autonomous Systems*. 131-150.
- Herrera Victor, & Sanchez Nestor. (2011). *ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL*.
- Ingeniería Electromecánica, C. DE, Chango Wilmer Wladimir Toapanta Guachi Edwin Leonardo, C., & Corrales Bonilla Johnatan Israel, I. (2023). *UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS-CIYA*.
- Jara, C. A., Sabater, J. M., Azorín, J. M., García, N. M., Pérez, C., Saltaren, R., & Yime, E. (2008). *ANÁLISIS DEL ESPACIO DE TRABAJO DE UN ROBOT PARALELO 3RRR*.
- Llerena Julio, Ccachuco Marcelo, & Cesar Pio. (2024). *Design and Implementation of a Delta-type Parallel Robot with Artificial Vision to for object tracking*.
- Merlet, J.-P. (2006). *Parallel Robots*.
- Pham, P. C., & Kuo, Y. L. (2023). Model-based and model-free collision detection and

identification for a parallel Delta robot with uncertainties. *Control Engineering*

*Practice*, 139. <https://doi.org/10.1016/j.conengprac.2023.105663>

Pinto, L., Alonso, A., Pinto, P., & Augusto, J. (2018). «DISEÑO E IMPLEMENTACIÓN DE UN ROBOT PARALELO TIPO DELTA DE 3 GRADOS DE LIBERTAD CONTROLADO POR MEDIO DE LÓGICA DIFUSA PARA LA CLASIFICACIÓN DE OBJETOS» Tesis presentada por los Bachilleres.

Sharida, A., & Hashlamon, I. (2021). Real-time vision-based controller for delta robots. *International Journal of Intelligent Systems Technologies and Applications*, 20(4), 271. <https://doi.org/10.1504/ijista.2021.10045532>

Siciliano, B. , & K. O. (Eds. ). (2016). *Springer Handbook of Robotics* (2nd ed.).

Zhang, Z. , D. H. , & Z. J. (2017). *Application of industrial robots in palletizing and packaging. International Journal of Advanced Manufacturing Technology*. 2131-2144.

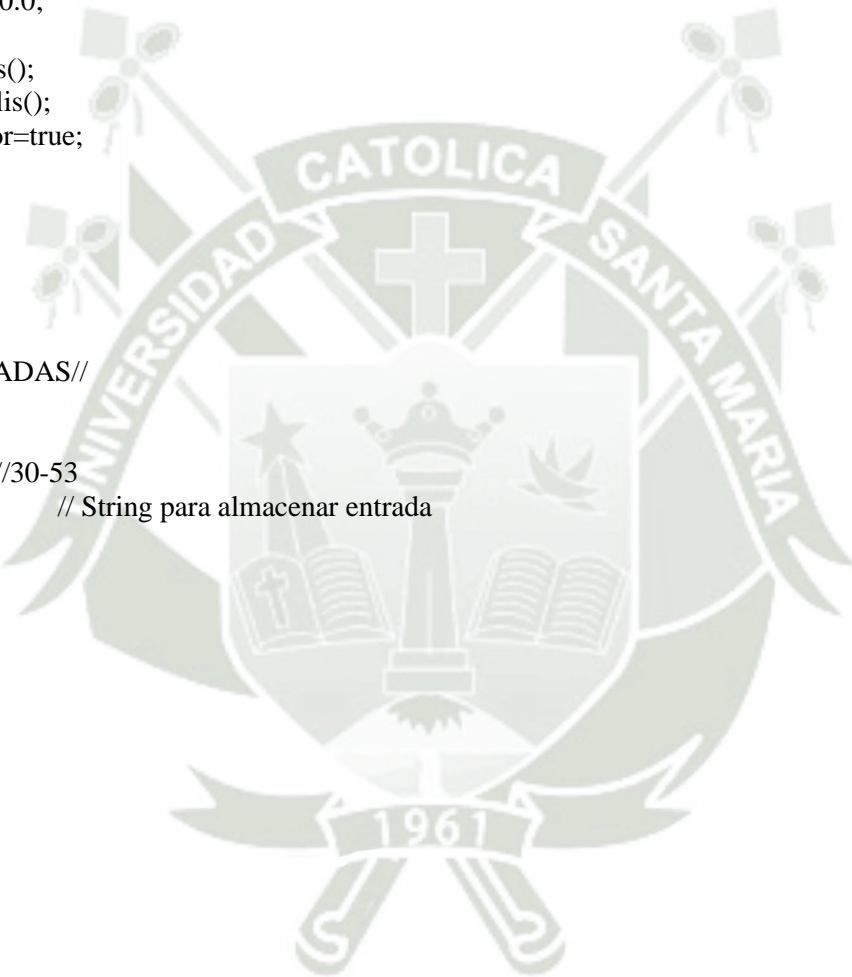


**ANEXOS 01: PROGRAM EN ARDUINO FINAL**

```

#include <Servo.h>
#include <Wire.h>
String px;
String py;
String color;
String tmp;
int pos;
int posC;
int posT;
int ncolor;
float tiempo;
float tiempoa=0.0;
int At;
float del=millis();
float mov=millis();
bool habilitador=true;
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
// COORDENADAS//
float x=0.0;
float y=0.00;
float z=-40.0; //30-53
String cor=""; // String para almacenar entrada
//////////
float L1=10.6;
float L2=2.1;
float L3=20.2;
float L4=40.5;
float L5=5.2;
float L6;
float q1;
float q2;
float q3;
float q12;
float q13;
float pi=3.14;
float a;
float s;
float p;
float r;
float d;
float u;
float c;
float i2;
float i3;
float k2;
float k3;
float d2;

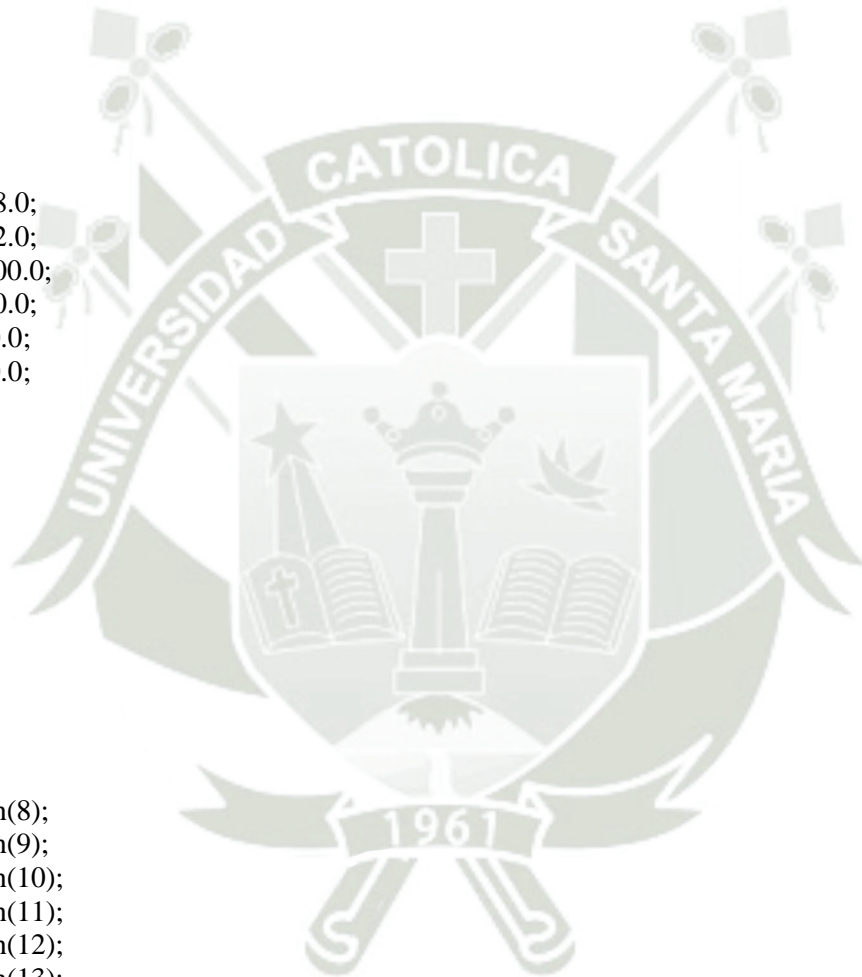
```



```
float d3;
float a2;
float a3;
float s2;
float s3;
float r2;
float r3;
float p2;
float p3;
float q22;
float q23;
float u2;
float u3;
float c2;
float c3;
float Q;
```

```
int q221a=98.0;
int q222a=82.0;
int q231a=100.0;
int q232a=90.0;
int q21ia=90.0;
int q22ia=90.0;
float q221;
float q222;
float q231;
float q232;
float q21i;
float q22i;
```

```
int ps=4;
int pii=5;
int p1=6;
int p22=7;
void setup() {
  servo1.attach(8);
  servo2.attach(9);
  servo3.attach(10);
  servo4.attach(11);
  servo5.attach(12);
  servo6.attach(13);
  pinMode(ps,INPUT);
  pinMode(pi,INPUT);
  pinMode(p1,OUTPUT);
  pinMode(p2,OUTPUT);
  Serial.begin(700);
  sube();
}
void loop(){
  while (Serial.available()) {
    lectura();
    habilitador=true;
```



```

if(habilitador==true){
if(tiempo>=4.0){
del=millis();
if(tiempo>=16.0){
while(millis()-del<12000){}
}else{
while(millis()-del<16000-tiempo*1000.0){}
}
x=15.0;
tiempoa=0;
}else{
x=43.5-2.47*(12.0+(tiempo+tiempoa));
tiempoa=tiempo;
}
}

```

```

punto();
switch (ncolor){
case 1:
baja();
x=0.0;
y=0.00;
z=-30.0;
punto();
x=-12.0;
y=23.00;
z=-30.0;
punto();
sube();
x=0.0;
y=0.00;
z=-35.0;
punto();
break;
case 2:
baja();
x=0.0;
y=0.00;
z=-30.0;
punto();
x=0.0;
y=23.00;
z=-30.0;
punto();
sube();
x=0.0;
y=0.00;
z=-35.0;
punto();
break;
case 3:
baja();
x=0.0;

```





```

if (L6>=L1){
    r=pi-atan(abs((z+L2)/(L1+L6)));
} else{
    s=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
if (L1>L6){
    r=atan(abs((z+L2)/abs(L1-L6)));
}
if (L6>L1){
    r=pi-atan(abs((z+L2)/abs(L1-L6)));
}
if (L6==L1){
    r=0.0;
}
}
d=acos((L4*L4*cos(a)*cos(a)+s*s-L3*L3)/(2.0*L4*cos(a)*s));
p=r-d;
if (L4*cos(a)*cos(pi/2-p)>-z-L2){
    q2=pi+atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
} else{
    q2=atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
}
u=atan(tan(a)/cos(p))*180.0/pi;
c=sqrt(L4*L4*sin(a)*sin(a)+L4*L4*cos(a)*cos(a)*cos(p)*cos(p));
q2=q2*180.0/pi;
a=a*180.0/pi;
p=p*180.0/pi;

// BRAZO 02
q12=120.0*pi/180.0;
i2=abs(atan(y/x)-pi/6.0);
k2=sqrt(x*x+y*y)*cos(i2);
if (y>x/sqrt(3.0)){
    a2=asin((sin(i2)*(sqrt(x*x+y*y))/L4));
}
if (y<x/sqrt(3.0)){
    a2=-asin((sin(i2)*(sqrt(x*x+y*y))/L4));
}
if (y==x/sqrt(3.0)){
    a2=0.0;
}
L6=abs(L5+y*cos(q12)-x*sin(q12));
if (y>-sqrt(3)*x+2*L5){
    s2=sqrt((L1+L6)*(L1+L6)+(z+L2)*(z+L2));
    r2=atan(abs((z+L2)/(L1+L6)));
}
if (y<=-sqrt(3)*x+2*L5){
    s2=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
    r2=atan(abs((z+L2)/(L1-L6)));
}
if (y<=-sqrt(3)*x+2*L5 && L1<L6){
    r2=pi-atan(abs((z+L2)/(L1-L6)));
}
    
```

```

if (y<=-sqrt(3.0)*x-2.0*L5 && L6==L1){
    r2=0.0;
}
d2=acos((L4*L4*cos(a2)*cos(a2)+s2*s2-L3*L3)/(2.0*L4*cos(a2)*s2));
p2=r2-d2;
if (L4*cos(a2)*cos(pi/2-p2)>-z-L2){
    q22=pi+atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
}else{
    q22=atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
}
u2=atan(tan(a2)/cos(p2))*180.0/pi;
c2=sqrt(L4*L4*sin(a2)*sin(a2)+L4*L4*cos(a2)*cos(a2)*cos(p2)*cos(p2));
q22=q22*180.0/pi;
a2=a2*180.0/pi;
p2=p2*180.0/pi;

// BRAZO 03
q13=240.0*pi/180.0;
i3=abs(atan(y/x)+pi/6.0);
k3=sqrt(x*x+y*y)*cos(i3);
if (y>-x/sqrt(3.0)){
    a3=-asin((sin(i3)*(sqrt(x*x+y*y))/L4));
}
if (y<-x/sqrt(3.0)){
    a3=asin((sin(i3)*(sqrt(x*x+y*y))/L4));
}
if (y==x/sqrt(3.0)){
    a3=0.0;
}
L6=abs(L5+y*cos(q13)-x*sin(q13));
if (y>sqrt(3.0)*x+2.0*L5){
    s3=sqrt((L1+L6)*(L1+L6)+(z+L2)*(z+L2));
    r3=atan(abs((z+L2)/(L1+L6)));
}
if (y<=sqrt(3.0)*x+2.0*L5){
    s3=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
    r3=atan(abs((z+L2)/(L1-L6)));
}
if (y<=sqrt(3.0)*x+2.0*L5 && L1<L6){
    r3=pi-atan(abs((z+L2)/(L1-L6)));
}
if (y<=sqrt(3.0)*x-2.0*L5 && L6==L1){
    r3=0.0;
}
d3=acos((L4*L4*cos(a3)*cos(a3)+s3*s3-L3*L3)/(2.0*L4*cos(a3)*s3));
p3=r3-d3;
if (L4*cos(a3)*cos(pi/2-p3)>-z-L2){
    q23=pi+atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
}else{
    q23=atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
}
u3=atan(tan(a3)/cos(p3))*180.0/pi;

```

```

c3=sqrt(L4*L4*sin(a3)*sin(a3)+L4*L4*cos(a3)*cos(a3)*cos(p3)*cos(p3));
q23=q23*180.0/pi;
a3=a3*180.0/pi;
p3=p3*180.0/pi;
q221=98.0+q22-90;
q222=82.0-q22+90;
q231=100.0+q23-90;
q232=90.0-q23+90;
q21i=90.0-q2+90;
q22i=90.0+q2-90;
for(int i=0;i<=200;i++){
servo1.write(q221a+i*((q221-q221a)/200.0));//sube para subir
servo2.write(q222a+i*((q222-q222a)/200.0));//baja para subir
servo3.write(q231a+i*((q231-q231a)/200.0));//baja para subir
servo4.write(q232a+i*((q232-q232a)/200.0));//sube para subir
servo5.write(q21ia+i*((q21i-q21ia)/200.0));//sube para subir
servo6.write(q22ia+i*((q22i-q22ia)/200.0));//baja para subir
mov=millis();
while(millis()-mov<3.0){}
}
q221a=q221;
q222a=q222;
q231a=q231;
q232a=q232;
q21ia=q21i;
q22ia=q22i;
}
void sube(){
while(digitalRead(ps)==LOW){
digitalWrite(p1,LOW);
digitalWrite(p22,HIGH);
}
digitalWrite(p1,LOW);
digitalWrite(p22,LOW);
}
void baja(){
while(digitalRead(pii)==LOW){
digitalWrite(p1,HIGH);
digitalWrite(p22,LOW);
}
digitalWrite(p1,LOW);
digitalWrite(p22,LOW);
}
}

```



**ANEXOS 02: PROGRAM EN ARDUINO PARA PRUEBA**

```

#include <Servo.h>
#include <Wire.h>
String px;
String py;
int pos;
Servo servo1;Servo servo2;
Servo servo3;Servo servo4;
Servo servo5;Servo servo6;
// COORDENADAS//
float x=0.0;
float y=24.00;
float z=-35.0; //30-53
String cor="";
float L1=15.0;
float L2=2.1;
float L3=20.2;
float L4=40.5;
float L5=5.2;
float L6;
float q1;
float q2;
float q3;
float q12;
float q13;
float pi=3.14;
float a;float s;
float p;float r;
float d;float u;
float c;float i2;
float i3;float k2;
float k3;float d2;
float d3;float a2;
float a3;float s2;
float s3;float r2;
float r3;float p2;
float p3;float q22;
float q23;float u2;
float u3;float c2;
float c3;float Q;
  int q221a=85.0;
  int q222a=85.0;
  int q231a=88.0;
  int q232a=96.0;
  int q21ia=105.0;
  int q22ia=85.0;
float q221;float q222;
float q231;float q232;
float q21i;float q22i;
int ps=4;
int pii=5;
int p1=6;
int p22=7;

```



```
void setup() {
  servo1.attach(8);
  servo2.attach(9);
  servo3.attach(10);
  servo4.attach(11);
  servo5.attach(12);
  servo6.attach(13);
  pinMode(ps,INPUT);
  pinMode(pi,INPUT);
  pinMode(p1,OUTPUT);
  pinMode(p2,OUTPUT);
  Serial.begin(1200);
  sube();
}
void loop(){
  x=0.0; y=0.00; z=-39.0;
  punto();
  baja();
  x=0.0; y=0.00; z=-30.0;
  punto();
  x=0.0; y=23.00; z=-30.0;
  punto();
  sube();
  x=0.0; y=0.00; z=-30.0;
  punto();
  x=0.0; y=0.00; z=-39.0;
  punto();
  baja();
  x=0.0; y=0.00; z=-30.0;
  punto();
  x=15.0; y=23.00; z=-30.0;
  punto();
  sube();
  x=0.0; y=0.00; z=-30.0;
  punto();
  x=0.0; y=0.00; z=-39.0;
  punto();
  baja();
  x=0.0; y=0.00; z=-30.0;
  punto();
  x=-15.0; y=23.00; z=-30.0;
  punto();
  sube();
  x=0.0; y=0.00; z=-30.0;
  punto();
  while (Serial.available()) {
    lectura();
  }
}
void lectura(){
  float pxf=x;
  float pyf=y;
  cor=Serial.readStringUntil('\n');
```

```

pos=cor.indexOf(',');
py=cor.substring(0,pos);
px=cor.substring(pos+1);
x=px.toFloat();
y=py.toFloat();
if(abs(px-x)>0.5||abs(pyf-y)>0.5){
x=(x*40.0/480.0)-20.0;
y=(-y*40.0/640.0)+20.0;
}
pxf=x;
pyf=y;
}
void punto(){
// BRAZO 01
q1=0.0*pi/180.0;
a=asin(x/L4);
L6=abs(L5+y*cos(q1)-x*sin(q1));
if (abs(y)>L5 && y<0.0){
s=sqrt((L1+L6)*(L1+L6)+(z+L2)*(z+L2));
if (L1>L6){
r=atan(abs((z+L2)/(L1+L6)));
}
if (L6>=L1){
r=pi-atan(abs((z+L2)/(L1+L6)));
} else{
s=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
if (L1>L6){
r=atan(abs((z+L2)/abs(L1-L6)));
}
if (L6>L1){
r=pi-atan(abs((z+L2)/abs(L1-L6)));
}
if (L6==L1){
r=0.0;
} }
d=acos((L4*L4*cos(a)*cos(a)+s*s-L3*L3)/(2.0*L4*cos(a)*s));
p=r-d;
if (L4*cos(a)*cos(pi/2-p)>-z-L2){
q2=pi+atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
} else{
q2=atan((L1-y*cos(q1)-L4*cos(a)*cos(p)-L5+x*sin(q1))/(L4*cos(a)*sin(p)+L2+z));
}
}
u=atan(tan(a)/cos(p))*180.0/pi;
c=sqrt(L4*L4*sin(a)*sin(a)+L4*L4*cos(a)*cos(a)*cos(p)*cos(p));
q2=q2*180.0/pi;
a=a*180.0/pi;
p=p*180.0/pi;
// BRAZO 02
q12=120.0*pi/180.0;
i2=abs(atan(y/x)-pi/6.0);
k2=sqrt(x*x+y*y)*cos(i2);
if (y>x/sqrt(3.0)){

```

```

a2=asin((sin(i2)*(sqrt(x*x+y*y))/L4));
}
if (y<x/sqrt(3.0)){
a2=-asin((sin(i2)*(sqrt(x*x+y*y))/L4));
}
if (y==x/sqrt(3.0)){
a2=0.0;
}
L6=abs(L5+y*cos(q12)-x*sin(q12));
if (y>-sqrt(3)*x+2*L5){
s2=sqrt((L1+L6)*(L1+L6)+(z+L2)*(z+L2));
r2=atan(abs((z+L2)/(L1+L6)));
}
if (y<=-sqrt(3)*x+2*L5){
s2=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
r2=atan(abs((z+L2)/(L1-L6)));
}
if (y<=-sqrt(3)*x+2*L5 && L1<L6){
r2=pi-atan(abs((z+L2)/(L1-L6)));
}
if (y<=-sqrt(3.0)*x-2.0*L5 && L6==L1){
r2=0.0;
}
d2=acos((L4*L4*cos(a2)*cos(a2)+s2*s2-L3*L3)/(2.0*L4*cos(a2)*s2));
p2=r2-d2;
if (L4*cos(a2)*cos(pi/2-p2)>-z-L2){
q22=pi+atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
}else{
q22=atan((L1-y*cos(q12)-L4*cos(a2)*cos(p2)-L5+x*sin(q12))/(L4*cos(a2)*sin(p2)+L2+z));
}
u2=atan(tan(a2)/cos(p2))*180.0/pi;
c2=sqrt(L4*L4*sin(a2)*sin(a2)+L4*L4*cos(a2)*cos(a2)*cos(p2)*cos(p2));
q22=q22*180.0/pi;
a2=a2*180.0/pi;
p2=p2*180.0/pi;

// BRAZO 03
q13=240.0*pi/180.0;
i3=abs(atan(y/x)+pi/6.0);
k3=sqrt(x*x+y*y)*cos(i3);
if (y>-x/sqrt(3.0)){
a3=-asin((sin(i3)*(sqrt(x*x+y*y))/L4));
}
if (y<-x/sqrt(3.0)){
a3=asin((sin(i3)*(sqrt(x*x+y*y))/L4));
}
if (y==x/sqrt(3.0)){
a3=0.0;
}
L6=abs(L5+y*cos(q13)-x*sin(q13));
if (y>sqrt(3.0)*x+2.0*L5){
s3=sqrt((L1+L6)*(L1+L6)+(z+L2)*(z+L2));

```

```

    r3=atan(abs((z+L2)/(L1+L6)));
}
if (y<=sqrt(3.0)*x+2.0*L5){
    s3=sqrt((L1-L6)*(L1-L6)+(z+L2)*(z+L2));
    r3=atan(abs((z+L2)/(L1-L6)));
}
if (y<=sqrt(3)*x+2*L5 && L1<L6){
    r3=pi-atan(abs((z+L2)/(L1-L6)));
}
if (y<=sqrt(3.0)*x-2.0*L5 && L6==L1){
    r3=0.0;
}
d3=acos((L4*L4*cos(a3)*cos(a3)+s3*s3-L3*L3)/(2.0*L4*cos(a3)*s3));
p3=r3-d3;
if (L4*cos(a3)*cos(pi/2-p3)>-z-L2){
    q23=pi+atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
}else{
    q23=atan((L1-y*cos(q13)-L4*cos(a3)*cos(p3)-L5+x*sin(q13))/(L4*cos(a3)*sin(p3)+L2+z));
}
u3=atan(tan(a3)/cos(p3))*180.0/pi;
c3=sqrt(L4*L4*sin(a3)*sin(a3)+L4*L4*cos(a3)*cos(a3)*cos(p3)*cos(p3));
q23=q23*180.0/pi;
a3=a3*180.0/pi;
p3=p3*180.0/pi;
q221=85.0-q22+90; q222=85.0+q22-90;
q231=88.0-q23+90; q232=96.0+q23-90;
q21i=105.0+q2-90; q22i=85.0-q2+90;
for(int i=0;i<=265;i++){
    servo1.write(q221a+i*((q221-q221a)/265.0));//sube para subir
    servo2.write(q222a+i*((q222-q222a)/265.0));//baja para subir
    servo3.write(q231a+i*((q231-q231a)/265.0));//baja para subir
    servo4.write(q232a+i*((q232-q232a)/265.0));//sube para subir
    servo5.write(q21ia+i*((q21i-q21ia)/265.0));//sube para subir
    servo6.write(q22ia+i*((q22i-q22ia)/265.0));//baja para subir
    delay(1);
}
q221a=q221; q222a=q222;
q231a=q231; q232a=q232;
q21ia=q21i; q22ia=q22i;
}
void sube(){
    while(digitalRead(ps)==LOW){
        digitalWrite(p1,LOW);
        digitalWrite(p22,HIGH);
    }
    digitalWrite(p1,LOW);
    digitalWrite(p22,LOW);
}
void baja(){
    while(digitalRead(pii)==LOW){
        digitalWrite(p1,HIGH);
        digitalWrite(p22,LOW);
    }
}

```

```
}  
digitalWrite(p1,LOW);  
digitalWrite(p22,LOW);  
}
```



**ANEXOS 03: PROGRMA DE PRUEBA DE CINEMATICA DIRECTA CON MTH**



## MTH

```
clear all; close all; clc;
L1=15.0;L2=2.1;
L3=20.2;L4=40.5;
L5=5.2;
q1=0*pi/180;q2=122.0937*pi/180;
a=14.2949*pi/180;p=79.8546*pi/180;
b=q2-p+pi/2;l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
e=atan(l/o);
T01=DH(q1+pi/2,-L2,L1,pi/2);T12=DH(q2-pi/2,0,L3,0);
T23=DH(-b,L4*sin(a),L4*cos(a),0);T34=DH(-p,0,L5,0);
T02=T01*T12;T03=T01*T12*T23;
T04=T01*T12*T23*T34
```

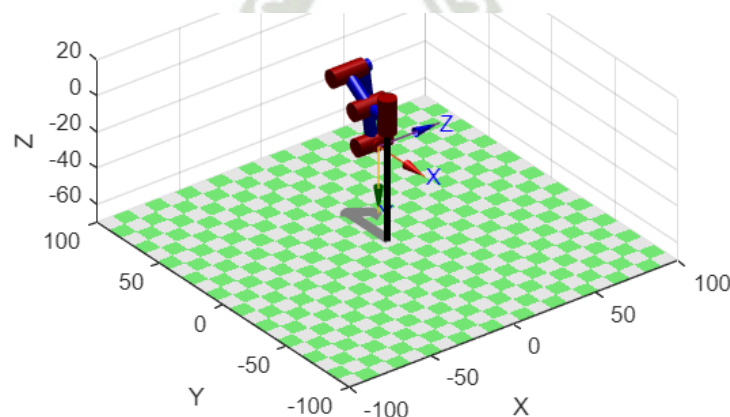
```
T04 = 4x4
    -0.0000    0.0000    1.0000    10.0000
    -1.0000    0.0000   -0.0000    20.0000
    -0.0000   -1.0000    0.0000   -30.0000
         0         0         0         1.0000
```

### Denavid Hartenberg

```
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot1=SerialLink([T1 T2 T3 T4],'name','Robot_1');
ws=[-100 100 -100 100 -70 20];
bot1.fkine([q1+pi/2 q2-pi/2 -b -p])
```

```
ans =
     0         0         1         10
    -1         0         0         20
     0        -1         0        -30
     0         0         0         1
```

```
bot1.plot([q1+pi/2 q2-pi/2 -b -p],'workspace',ws,'noname')
```



**ANEXOS 04: PROGRAMA DE PRUEBA DE CINEMATICA DIRECTA**



```
clear all; close all; clc;
L1=15.0;L2=2.1;
L3=20.2;L4=40.5;
L5=5.2;
x=20;
y=20;
z=-49;
[Q] = paraleloD(x,y,z)
```

```
Q = 3x3
    52.9074    29.5924    80.3346
   -31.4602    10.4136    48.1458
    21.9506   -42.4216    70.3973
```

```
q2=Q(1,1)*pi/180;
a=Q(1,2)*pi/180;
p=Q(1,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot1=SerialLink([T1 T2 T3 T4],'name','Robot_1');
ws=[-100 100 -100 100 -70 20];
bot1.fkine([pi/2 q2-pi/2 -b -p])
```

```
ans =
     0         0         1        20
    -1         0         0        20
     0        -1         0       -49
     0         0         0         1
```

```
figure(1)
bot1.plot([pi/2 q2-pi/2 -b -p],'workspace',ws,'noname');
hold on

q2=Q(2,1)*pi/180;
a=Q(2,2)*pi/180;
p=Q(2,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
```

```
bot2=SerialLink([T1 T2 T3 T4], 'name', 'Robot_2');
ws=[-100 100 -100 100 -70 20];
bot2.fkine([7*pi/6 q2-pi/2 -b -p])
```

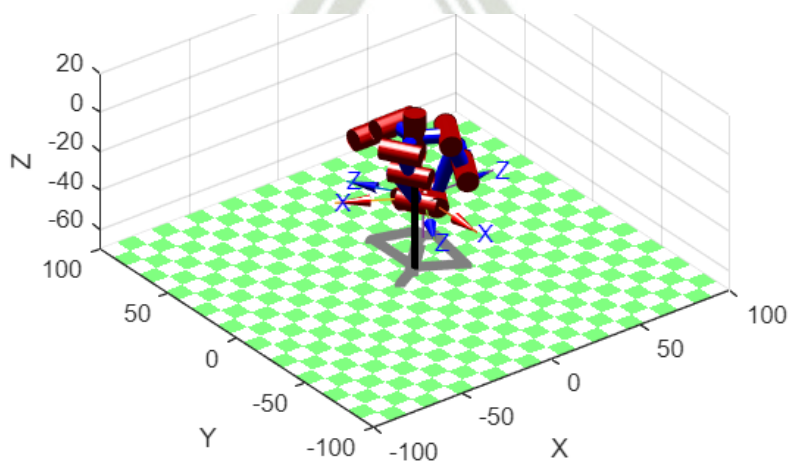
```
ans =
    0.8660     0    -0.5000     20
    0.5000     0     0.8660     20
         0    -1         0    -49
         0     0         0     1
```

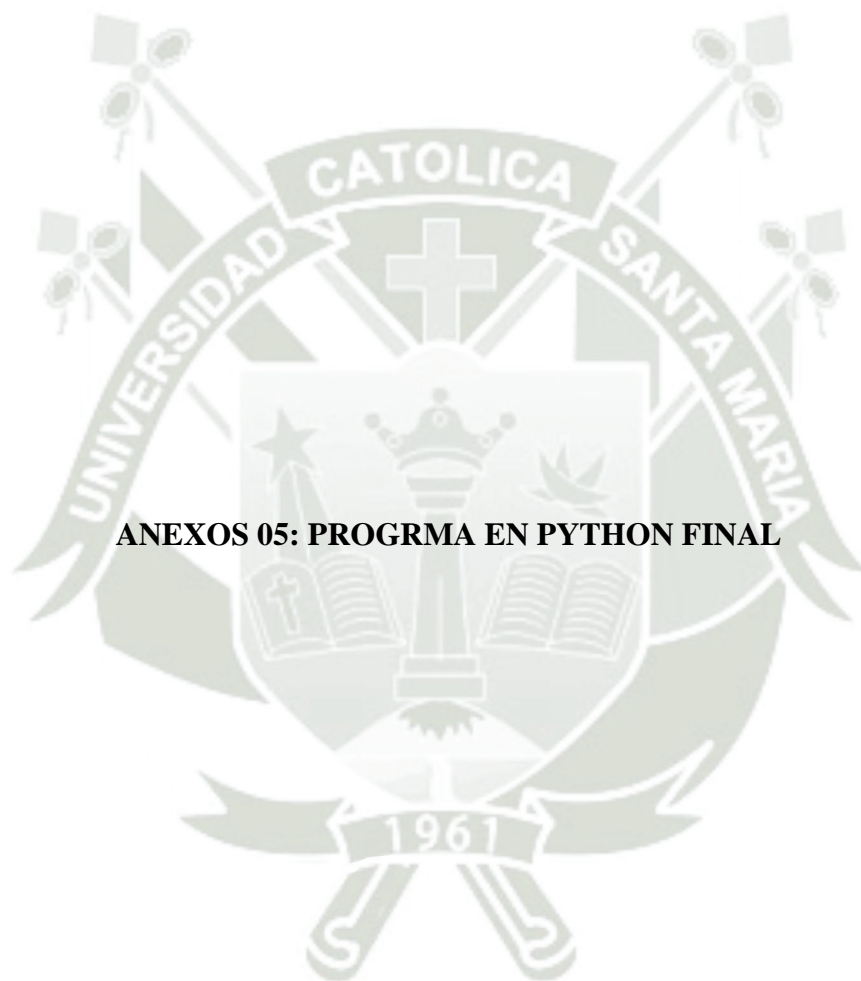
```
bot2.plot([7*pi/6 q2-pi/2 -b -p], 'workspace',ws, 'noname');
hold on
```

```
q2=Q(3,1)*pi/180;
a=Q(3,2)*pi/180;
p=Q(3,3)*pi/180;
b=q2-p+pi/2;
l=L4*sin(a);
o=L1+L3*sin(q2)-L4*cos(p)*cos(a);
T1=Revolute('d',-L2,'a',L1,'alpha',pi/2);
T2=Revolute('d',0,'a',L3,'alpha',0);
T3=Revolute('d',L4*sin(a),'a',L4*cos(a),'alpha',0);
T4=Revolute('d',0,'a',L5,'alpha',0);
bot3=SerialLink([T1 T2 T3 T4], 'name', 'Robot_3');
ws=[-100 100 -100 100 -70 20];
bot3.fkine([11*pi/6 q2-pi/2 -b -p])
```

```
ans =
   -0.8660     0    -0.5000     20
    0.5000     0    -0.8660     20
         0    -1         0    -49
         0     0         0     1
```

```
bot3.plot([11*pi/6 q2-pi/2 -b -p], 'workspace',ws, 'noname')
```





**ANEXOS 05: PROGRMA EN PYTHON FINAL**

```
import cv2
import numpy as np
import serial
import time
ser = serial.Serial('COM4', 700, timeout=0)## Lectura serial
x = 0
y = 0
inicio=time.time() # inicio del tiempo serial
pxfr = x
pyfr = y
pxfc = x
pyfc = y
pxfv = x
pyfv = y
factorLuz = 0.8
cap = cv2.VideoCapture(0) ## Numero de cámara empleada
rojoBajo = np.array([0, 50, 150], np.uint8) ## Color Intensidad Brillos
rojoAlto = np.array([25, 250, 250], np.uint8)
azulBajo = np.array([115,0,100], np.uint8)
azulAlto = np.array([170, 250, 250], np.uint8)
verdeBajo = np.array([35, 40, 140], np.uint8)
verdeAlto = np.array([70, 250, 250], np.uint8)
while True:
    ret, frame = cap.read()
    data = frame[:,:]
    frame[:,:]=data*factorLuz ## A todos los piexeles se les regula la
    cantidad de luz presente
    ##ROJO#####
    if ret:
        frame = cv2.rotate(frame, 1)
        frameHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        mascaraR = cv2.inRange(frameHSV, rojoBajo, rojoAlto)
        contornosR, _ = cv2.findContours(mascaraR, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        cv2.drawContours(frame, contornosR, -1, (0, 0, 255), 2)

        for c in contornosR:
            area = cv2.contourArea(c)
            if area > 9000:
                fin = time.time()
                M = cv2.moments(c)
                if M["m00"] == 0:
                    M["m00"] = 1
                x = int(M["m10"] / M["m00"])
                y = int(M["m01"] / M["m00"])
                cor = (str(x)+","+str(y)+"C1T"+str(fin-inicio)+"\n")
                if(y>425):
                    if abs(pxfr - x) > 15 or abs(pyfr - y) > 15:
                        pxfr = x
                        pyfr = y
                        ser.write(cor.encode('ascii'))
                        print(cor)
                        inicio = time.time()
                        cv2.circle(frame, (x, y), 7, (0, 0, 255), -1)
                        font = cv2.FONT_HERSHEY_SIMPLEX
                        cv2.putText(frame, '{}{}'.format(x, y), (x + 10, y), font,
1.2, (0, 0,0), 3, cv2.LINE_AA)
                        nuevoContornoR = cv2.convexHull(c)
```

```
cv2.drawContours(frame, [nuevoContornoR], 0, (0, 0, 255),
3)

## Celeste
if ret:
    frameHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mascara = cv2.inRange(frameHSV, azulBajo, azulAlto)
    contornos, _ = cv2.findContours(mascara, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    cv2.drawContours(frame, contornos, -1, (255, 0, 0), 2)
    for c in contornos:
        area = cv2.contourArea(c)
        if area > 9000: ## Area del objeto
            fin = time.time()
            M = cv2.moments(c)
            if M["m00"] == 0:
                M["m00"] = 1
            x = int(M["m10"] / M["m00"])
            y = int(M["m01"] / M["m00"])
            cor = str(x) + "," + str(y) + "C2T"+str(fin-inicio)+ "\n"
            if (y > 425): ## Posicion de la lectura
                if abs(pxfc - x) > 15 or abs(pyfc - y) > 15:
                    pxfc = x
                    pyfc = y
                    ser.write(cor.encode('ascii'))
                    print(cor)
                    inicio = time.time()
            cv2.circle(frame, (x, y), 7, (255, 0, 0), -1)
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(frame, '{}{}'.format(x, y), (x + 10, y), font,
1.2, (0, 0, 0), 3, cv2.LINE_AA)
            nuevoContorno = cv2.convexHull(c)
            cv2.drawContours(frame, [nuevoContorno], 0, (255, 0, 0), 3)

## Verde
if ret:
    frameHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mascaraV = cv2.inRange(frameHSV, verdeBajo, verdeAlto)
    contornosV, _ = cv2.findContours(mascaraV, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    cv2.drawContours(frame, contornosV, -1, (0, 255, 0), 2)
    for c in contornosV:
        area = cv2.contourArea(c)
        if area > 9000:
            fin = time.time()
            M = cv2.moments(c)
            if M["m00"] == 0:
                M["m00"] = 1
            x = int(M["m10"] / M["m00"])
            y = int(M["m01"] / M["m00"])
            cor = str(x) + "," + str(y) + "C3T"+str(fin-inicio)+ "\n"
            if (y > 425):
                if abs(pxfv - x) > 15 or abs(pyfv - y) > 15:
                    pxfv = x
                    pyfv = y
                    ser.write(cor.encode('ascii'))
                    print(cor)
                    inicio = time.time()
```

```
cv2.circle(frame, (x, y), 7, (0, 255, 0), -1)
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(frame, '{}{}'.format(x, y), (x + 10, y), font,
1.2, (0, 0, 0), 3, cv2.LINE_AA)
nuevoContornoV = cv2.convexHull(c)
cv2.drawContours(frame, [nuevoContornoV], 0, (0, 255, 0),
3)

# cv2.imshow('mascaraAzul', mascara)
cv2.imshow('frame', frame)
#time.sleep(0.04)

if cv2.waitKey(1) & 0xFF == ord('s'):
    ser.close()
    break
cap.release()
cv2.destroyAllWindows()
```





**ANEXOS 06: CRONOGRAMA DE ACTIVIDADES**

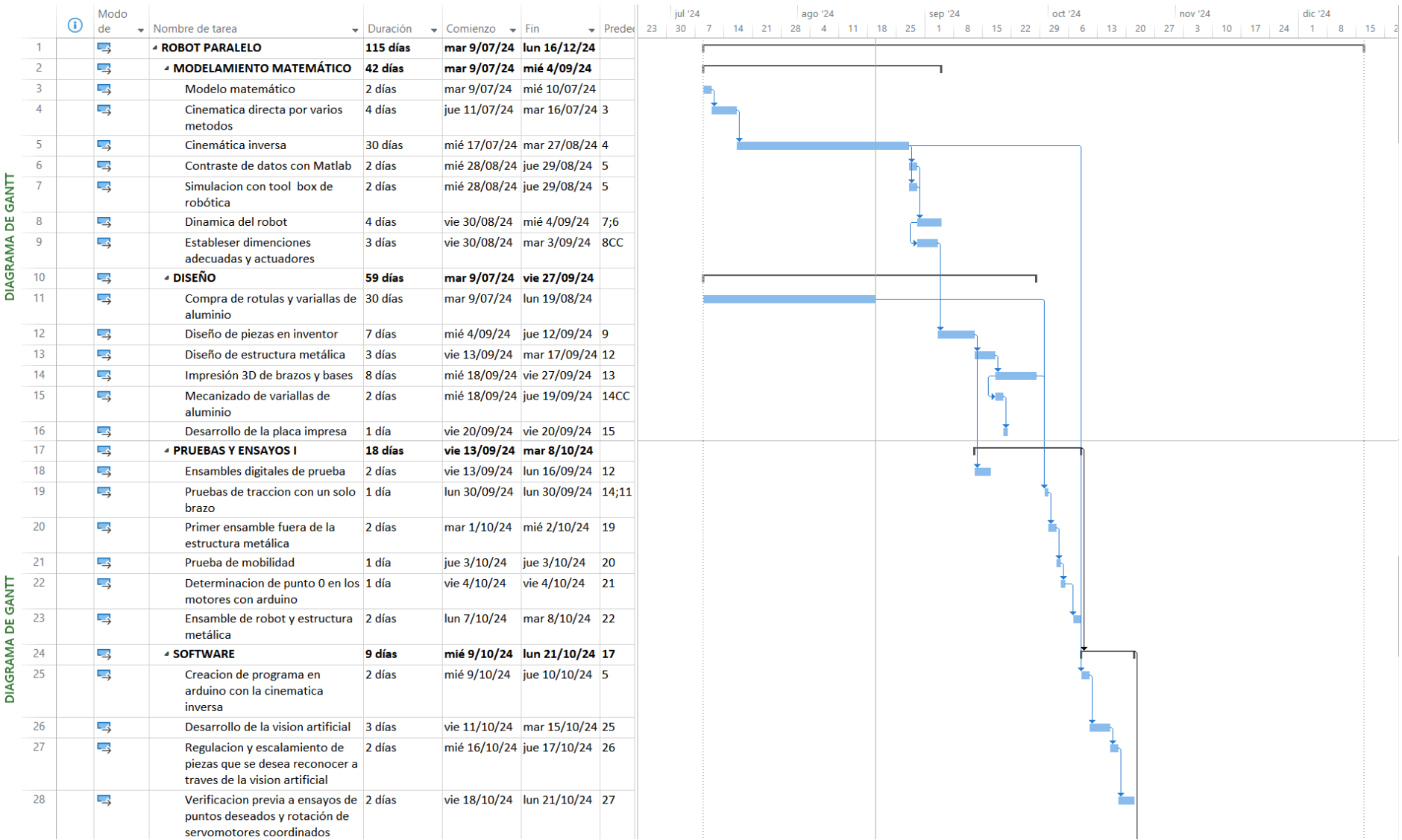
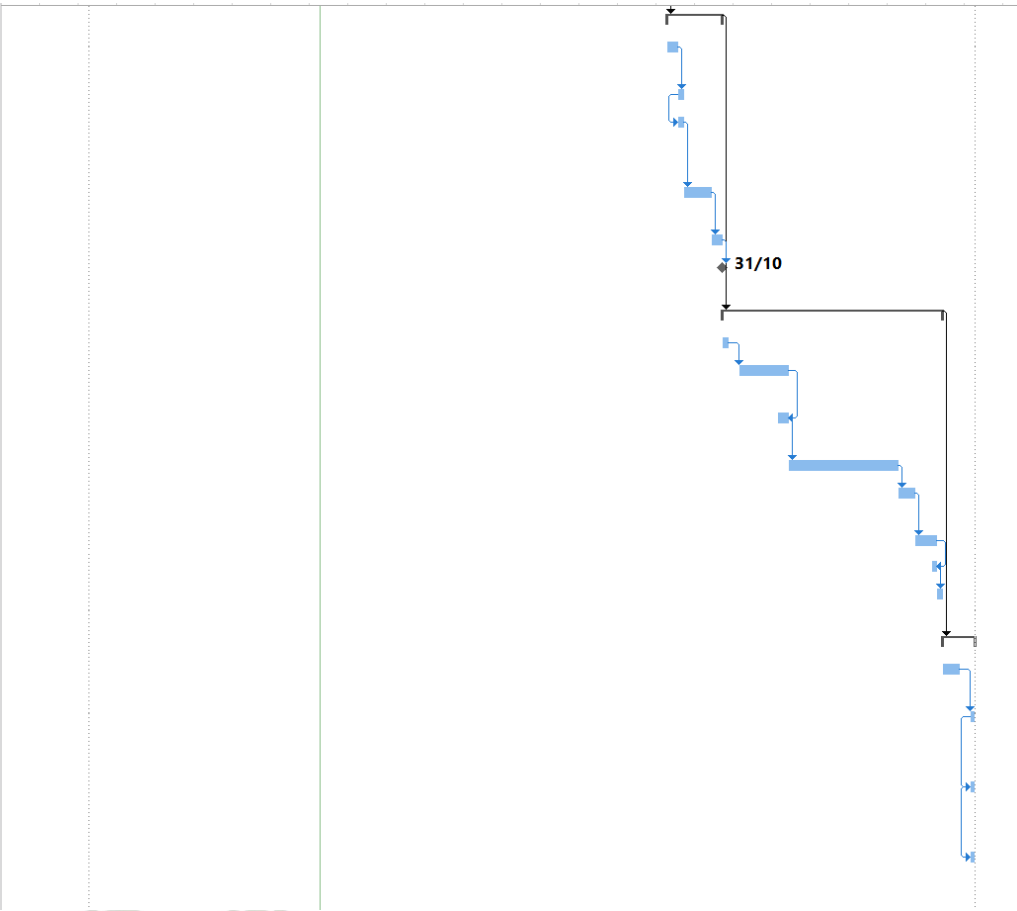


DIAGRAMA DE GANTT

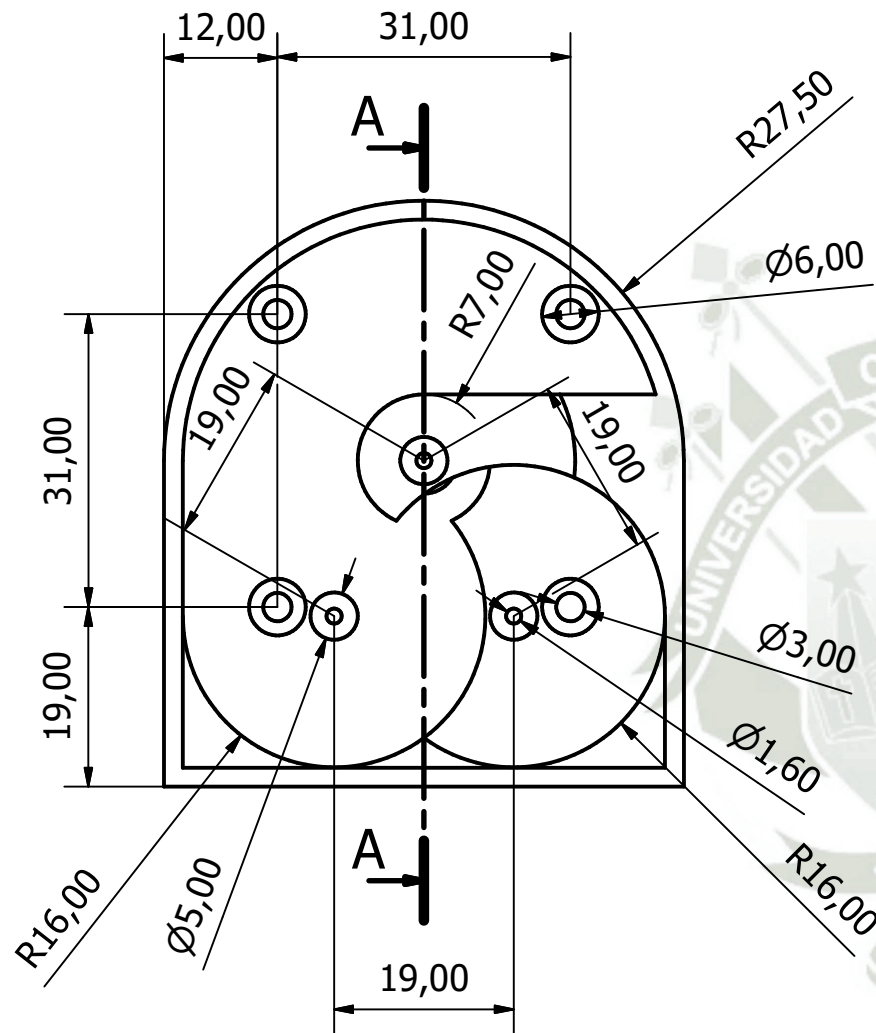
DIAGRAMA DE GANTT

29	→	<b>PRUEBAS Y ENSAYOS II</b>	<b>8 días</b>	<b>mar 22/10/24</b>	<b>jue 31/10/24</b>	<b>24</b>
30	→	Enviar puntos coordenados manualmente al robot	2 días	mar 22/10/24	mié 23/10/24	
31	→	Verificar la precisión	1 día	jue 24/10/24	jue 24/10/24	30
32	→	Verificar que la orientación de la cámara sea idéntica al plano coordinado asignado	1 día	jue 24/10/24	jue 24/10/24	31CC
33	→	Incluir la cámara para que trabaje en lazo cerrado	3 días	vie 25/10/24	mar 29/10/24	32
34	→	Realizar pruebas de movilidad	2 días	mié 30/10/24	jue 31/10/24	33
35	→	El robot es capaz de seguir objetos con precisión	0 días	jue 31/10/24	jue 31/10/24	34
36	→	<b>FAJA TRASPORTADORA</b>	<b>28 días</b>	<b>vie 1/11/24</b>	<b>mar 10/12/24</b>	<b>29</b>
37	→	Diseño de depósitos en inventor	1 día	vie 1/11/24	vie 1/11/24	
38	→	Diseño de piezas de la faja en inventor	7 días	lun 4/11/24	mar 12/11/24	37
39	→	Ensamble de verificación en Inventor	2 días	lun 11/11/24	mar 12/11/24	38FF
40	→	Impresión de piezas en 3D	14 días	mié 13/11/24	lun 2/12/24	39
41	→	Ensamble de faja (120 eslabones independientes)	3 días	mar 3/12/24	jue 5/12/24	40
42	→	Ensamble y fijación de estructura	2 días	vie 6/12/24	lun 9/12/24	41
43	→	Instalación de motor	1 día	lun 9/12/24	lun 9/12/24	42FF
44	→	Incorporación de la faja a la estructura	1 día	mar 10/12/24	mar 10/12/24	43
45	→	<b>PRUEBAS Y ENSAYOS III</b>	<b>4 días</b>	<b>mié 11/12/24</b>	<b>lun 16/12/24</b>	<b>36</b>
46	→	Prueba y ajustes de movilidad de a faja independiente al robot	3 días	mié 11/12/24	vie 13/12/24	
47	→	Encuadre de cámara y reescalado a dimensiones de la faja	1 día	lun 16/12/24	lun 16/12/24	46
48	→	Prueba de coordinación de faja y robot ajustando la altura pertinente	1 día	lun 16/12/24	lun 16/12/24	47CC
49	→	Reconocimiento de eje x de la ubicación del objeto transportado sin diferencia de color	1 día	lun 16/12/24	lun 16/12/24	48CC

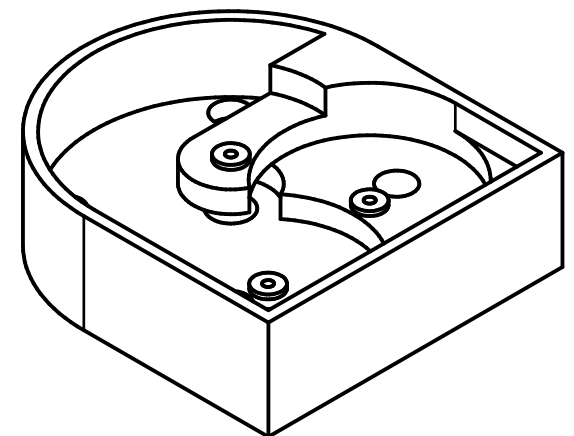
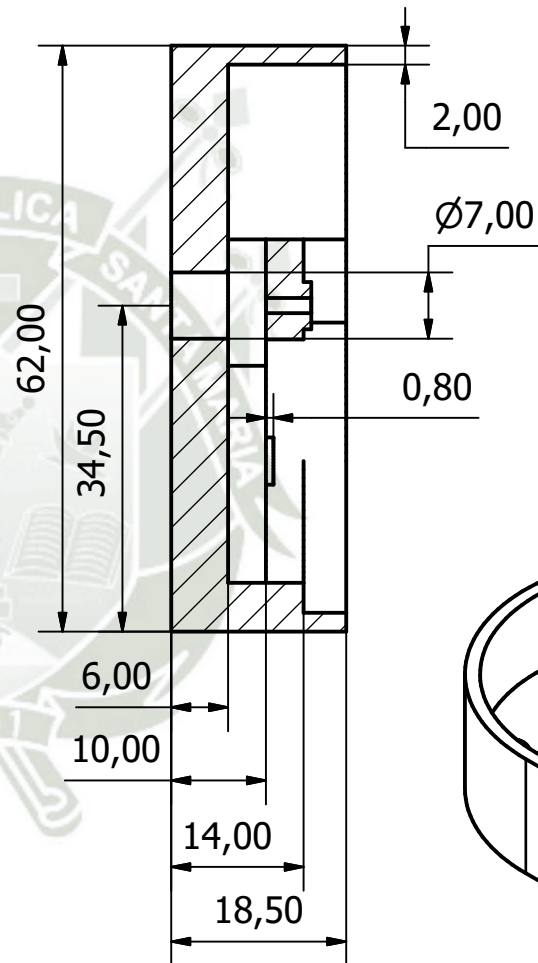




**ANEXOS 07: PLANOS GENERALES**



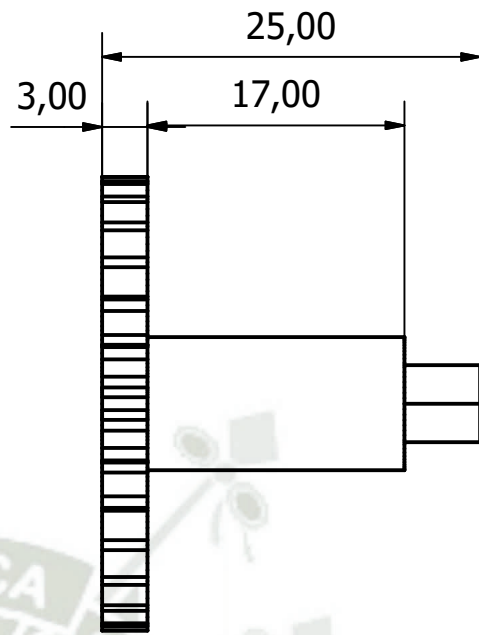
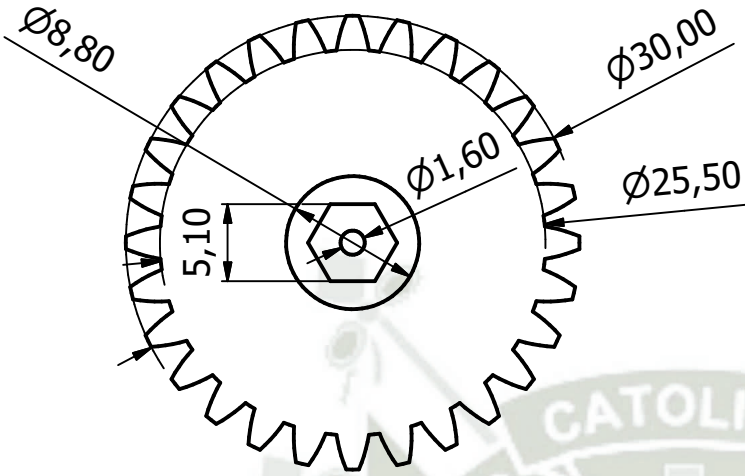
A-A ( 1 : 0.8 )



Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/12/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>BASE DEL REDUCTOR</b>		Página: 1/25

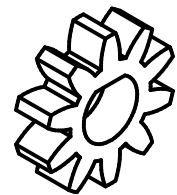
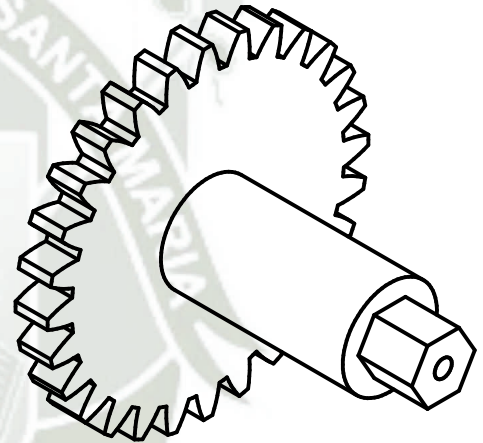
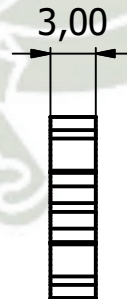
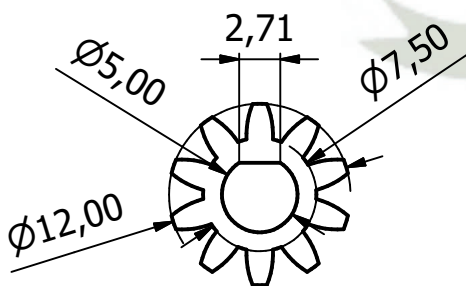
DATOS DE ENGRANAJE DE SALIDA

Z	28
Modulo	1
De	30 mm



DATOS DE ENGRANAJE DE ENTRADA

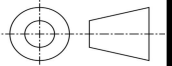
Z	10
Modulo	1
De	12 mm



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
29/12/2024



Universidad Católica  
de Santa María

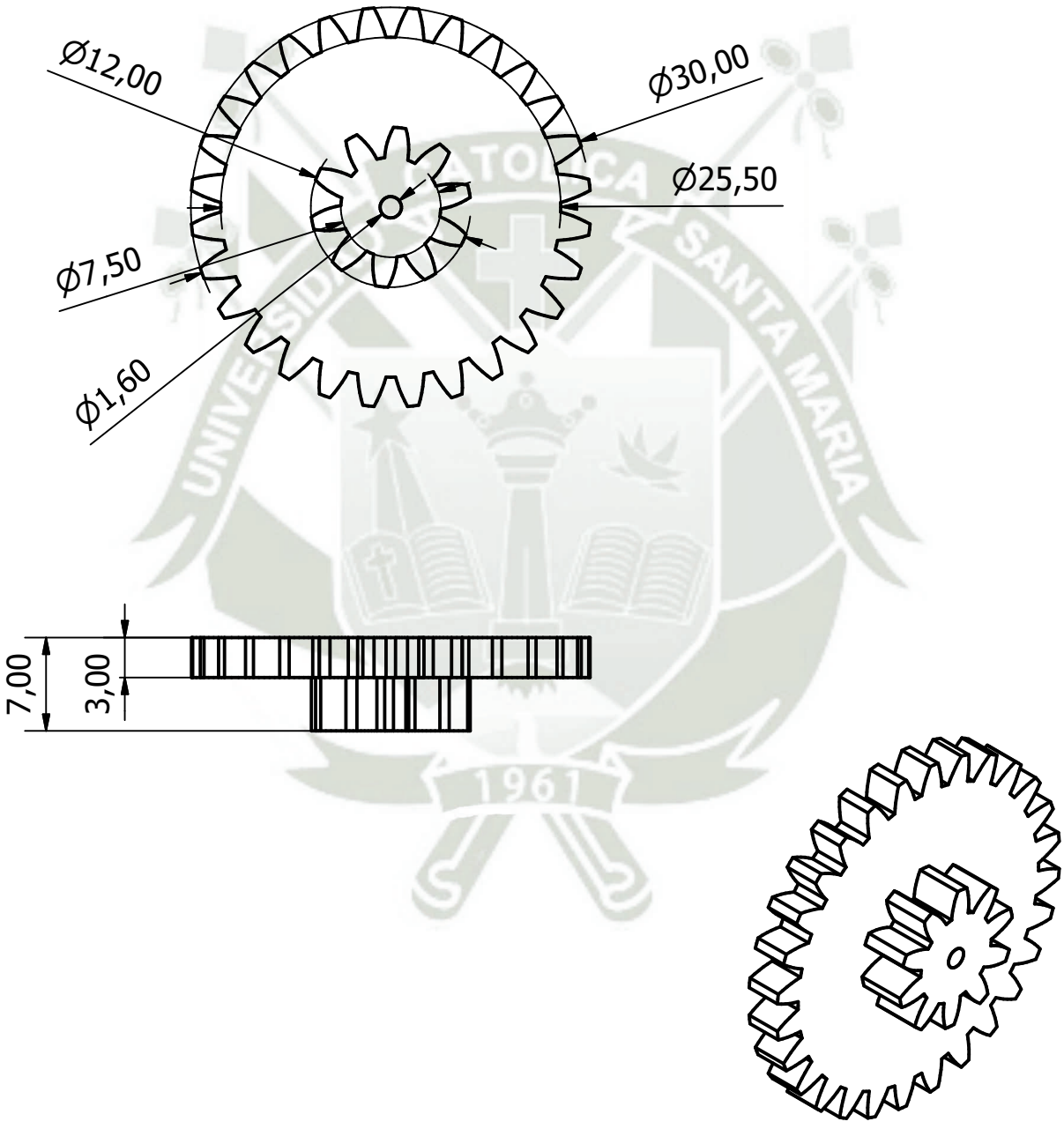
FAJA TRASPORTADORA


ENGRANAJE DE ENTRADA Y SALIDA

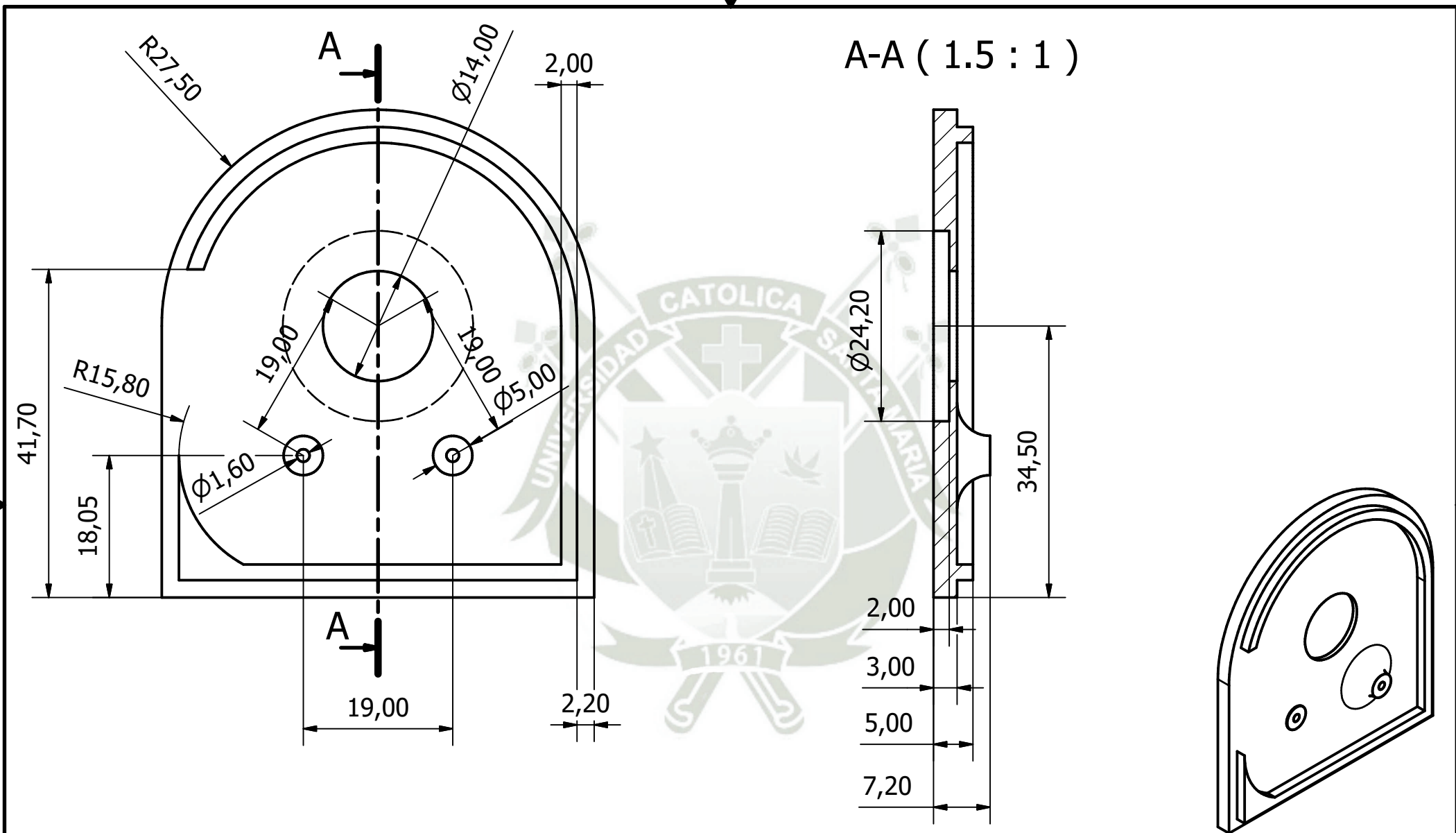
Página:  
2/25

Escala:  
2 : 1

DATOS DE ENGRANAJE DE SALIDA



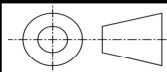
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/12/2024	
 <b>Universidad Católica de Santa María</b>	FAJA TRASPORTADORA		
	ENGRANAJE DOBLE	Página: 3/25	Escala: 2 : 1



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
29/12/2024



**Universidad Católica  
de Santa María**

FAJA TRASPORTADORA

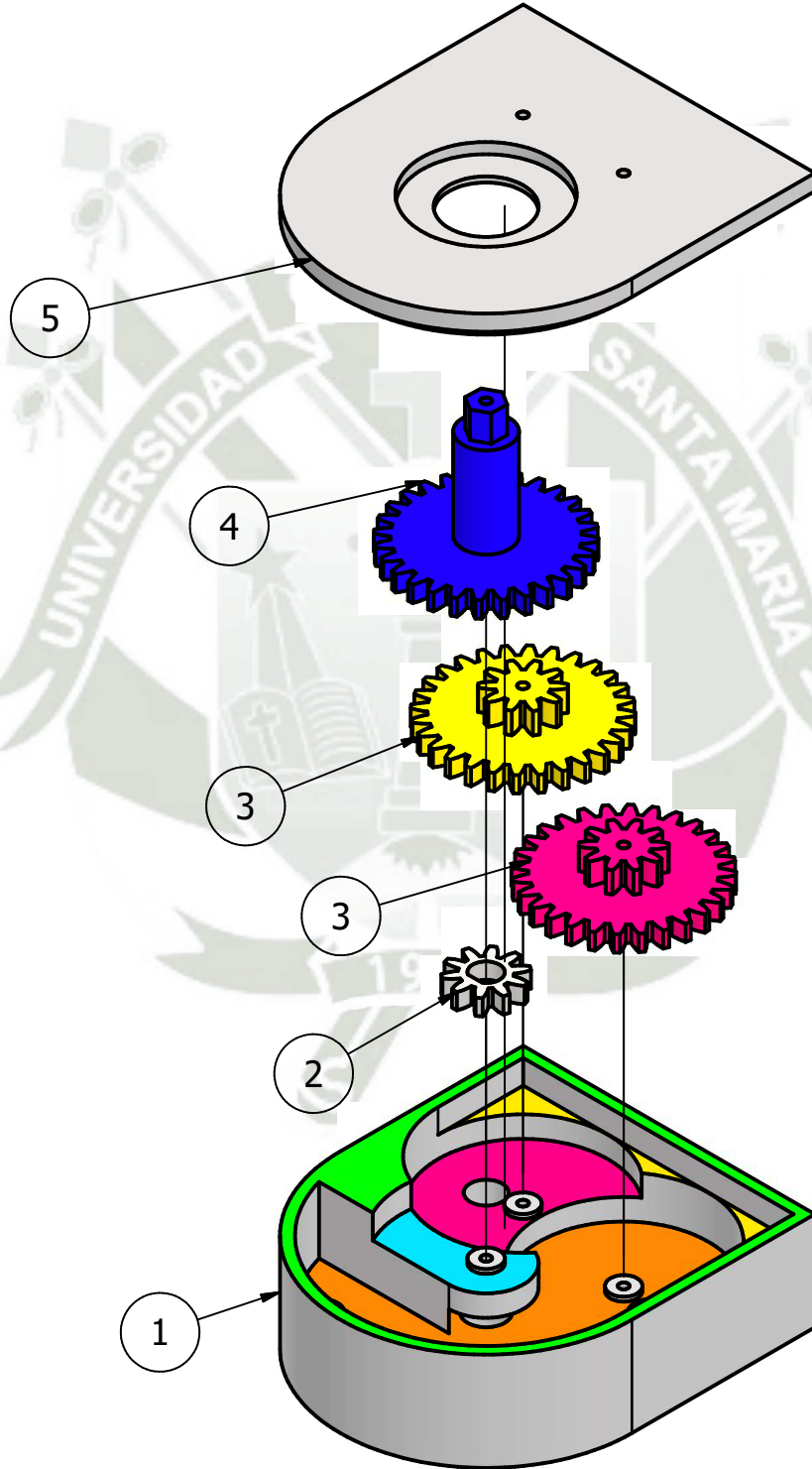
TAPA DEL REDUCTOR

Página:  
4/25

Escala:  
1.5 : 1

LISTA DE PIEZAS

ELEMENTO	CANTIDAD	PIEZA	HOJA
1	1	Base del reductor	1
2	1	Engranaje de entrada	2
3	2	Engrane doble	3
4	1	Engranaje de salida	2
5	1	Tapa del reductor	4



Dibujado por:

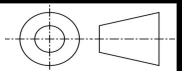
LLERENA SANCHEZ JULIO CESAR

Revisado por:

Ing. Christiam Collado Oporto

Fecha:

1/01/2025



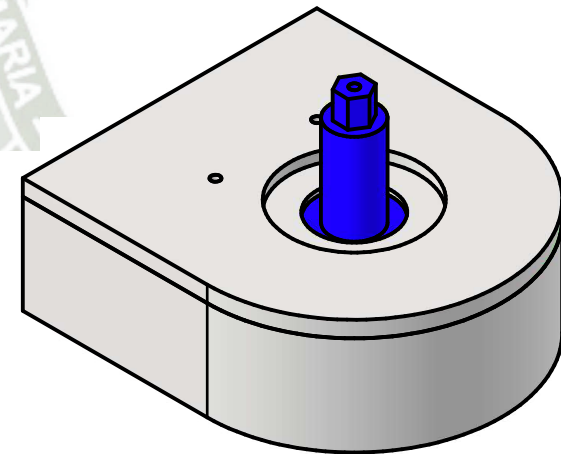
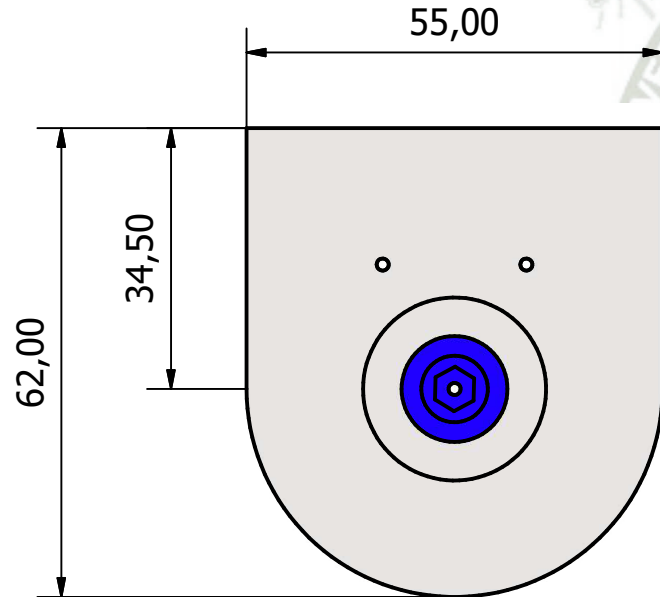
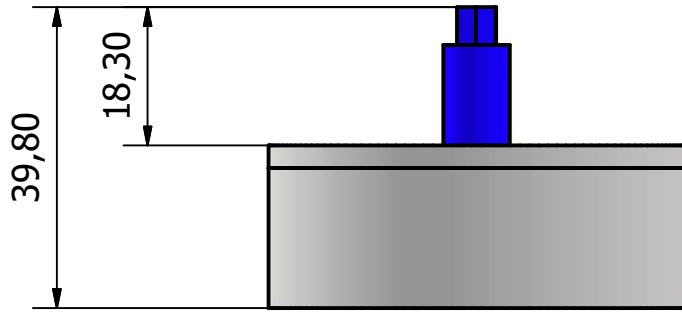
Universidad Católica  
de Santa María

FAJA TRANSPORTADORA

EXPLOSION DE CAJA REDUCTORA

Página:  
5/25

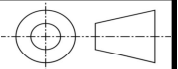
Escala:  
1 : 1



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
1/01/2025



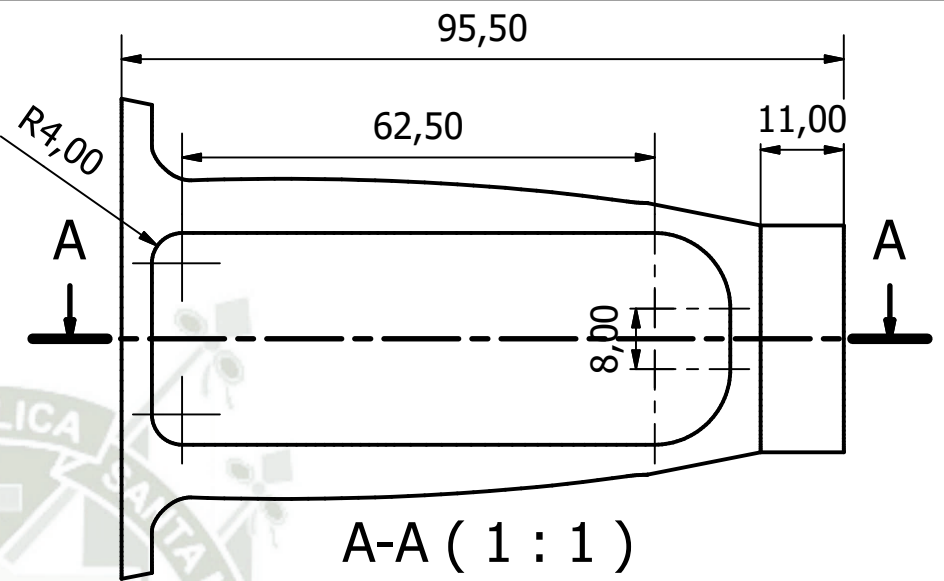
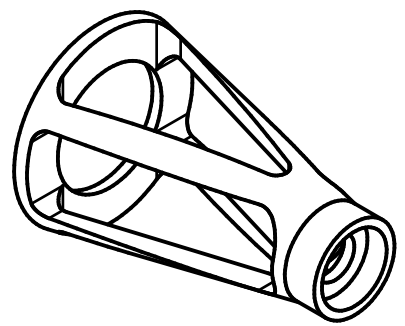
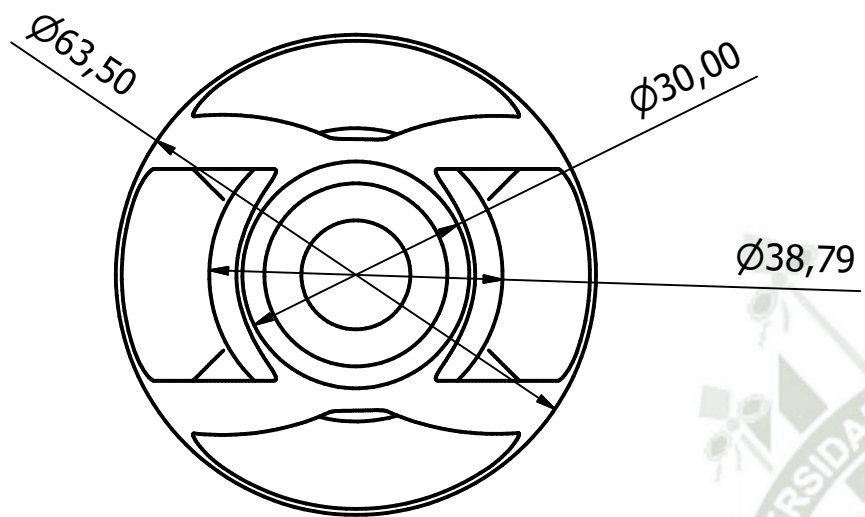
Universidad Católica  
de Santa María

FAJA TRASPOTADORA

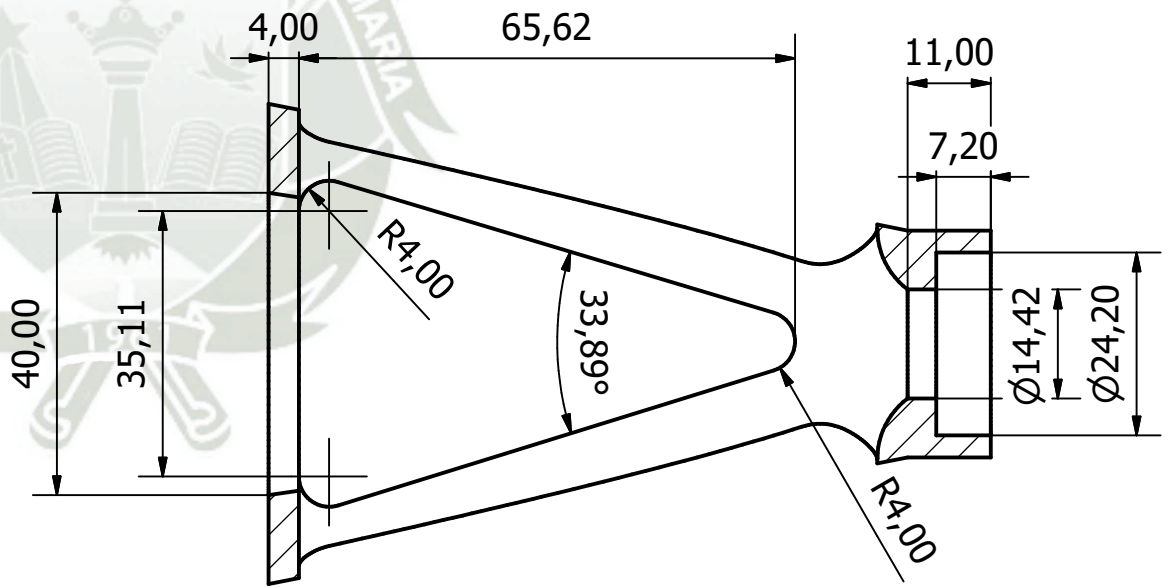
ENSAMBLE DE CAJA REDUCTORA

Página:  
6/25

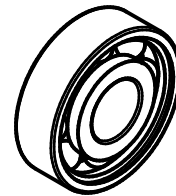
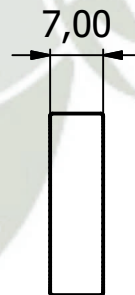
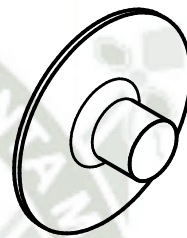
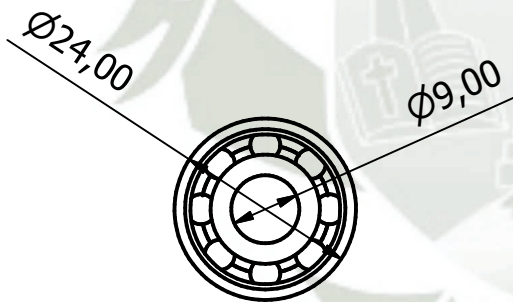
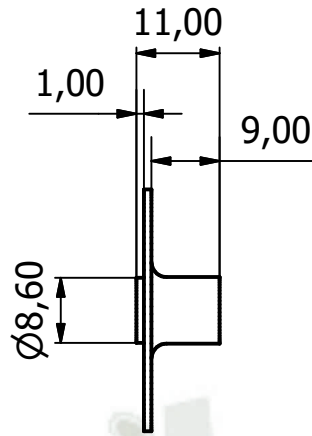
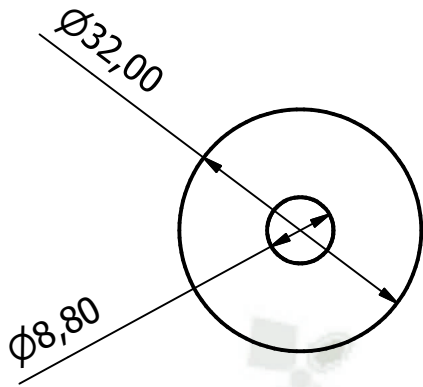
Escala:  
1 : 1



A-A (1 : 1)



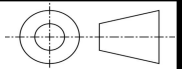
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 18/09/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>SUJETADOR DE RODAMIENTO</b>	Página: 7/25	Escala: 1 : 1



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
10/01/2025



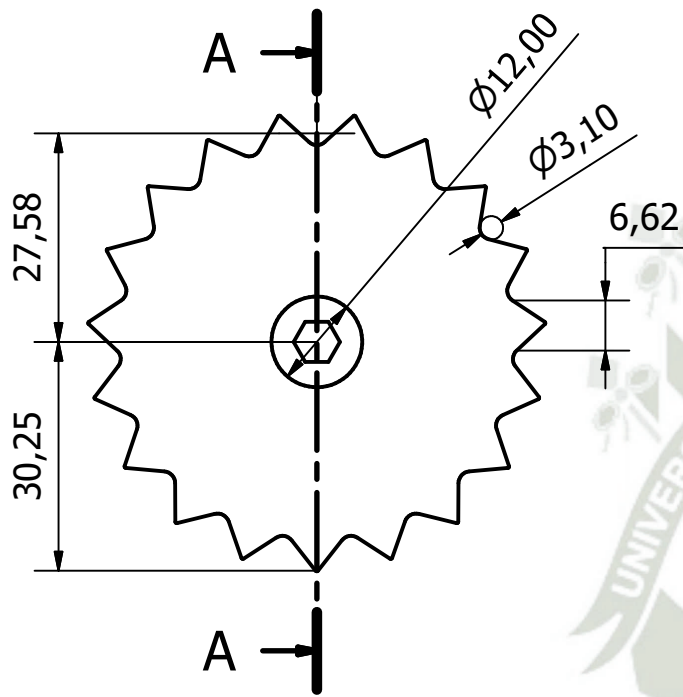
**Universidad Católica  
de Santa María**

FAJA TRASPORTADORA

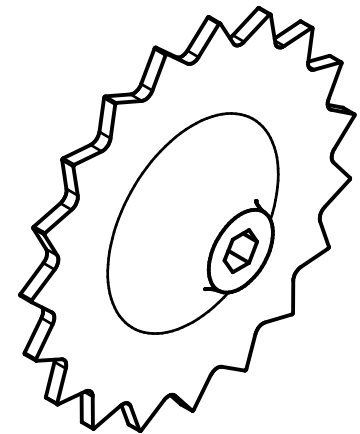
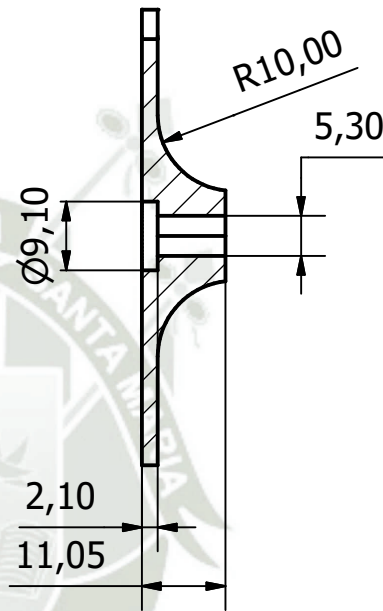
RODAMIENTO Y ACOPLE

Página:  
8/25

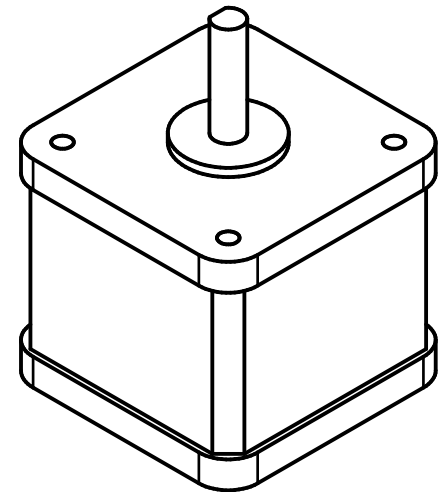
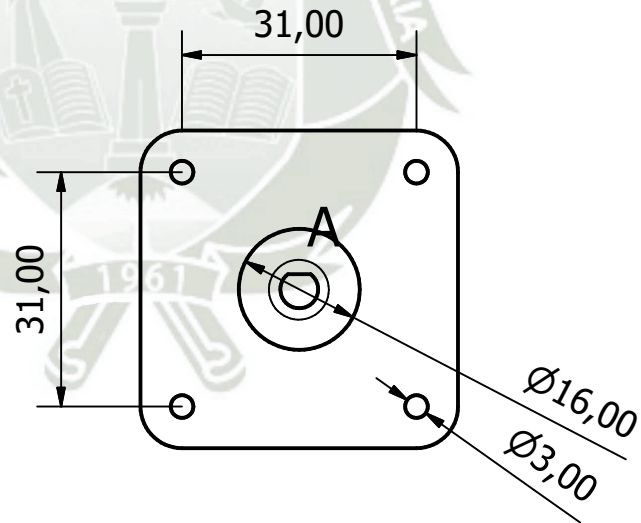
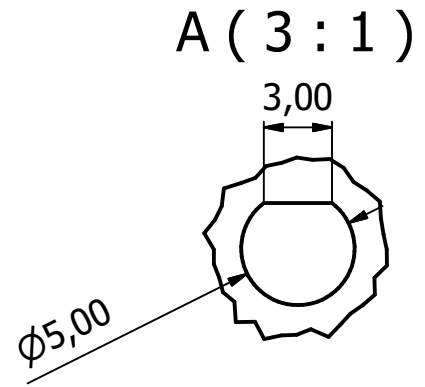
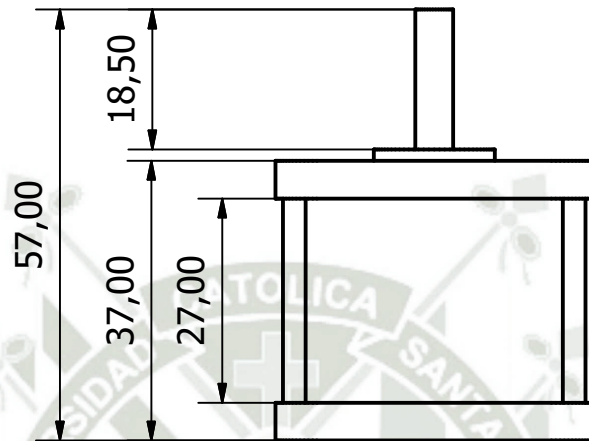
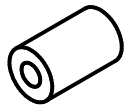
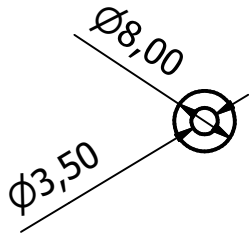
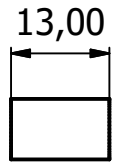
Escala:  
1 : 1



A-A ( 1 : 1 )



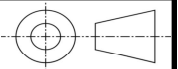
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 5/01/2025	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	CATALINA DE TRACCION	Página: 9/25	Escala: 1 : 1



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
17/03/2025



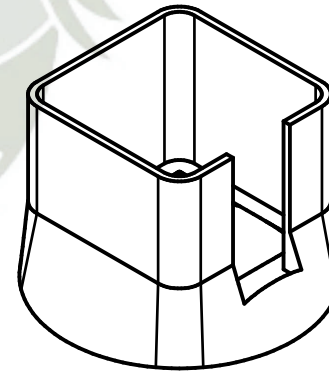
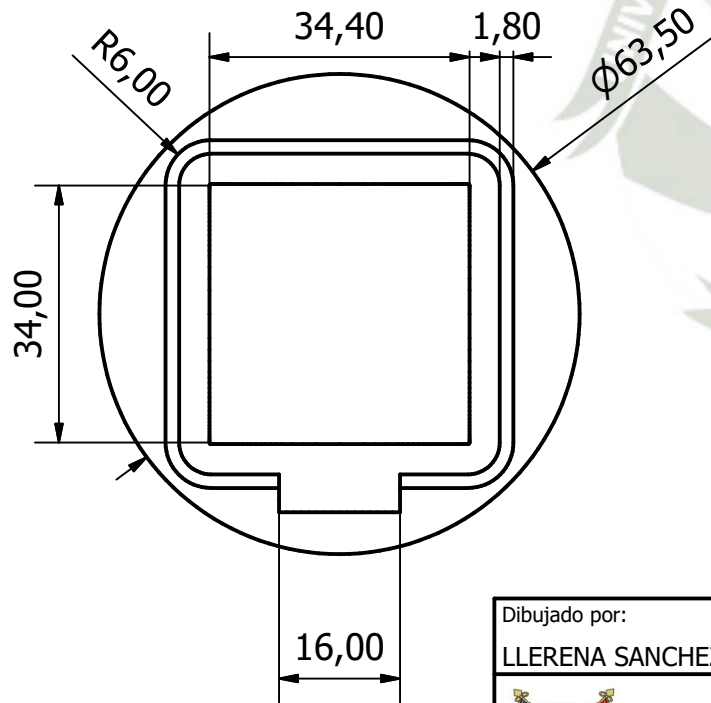
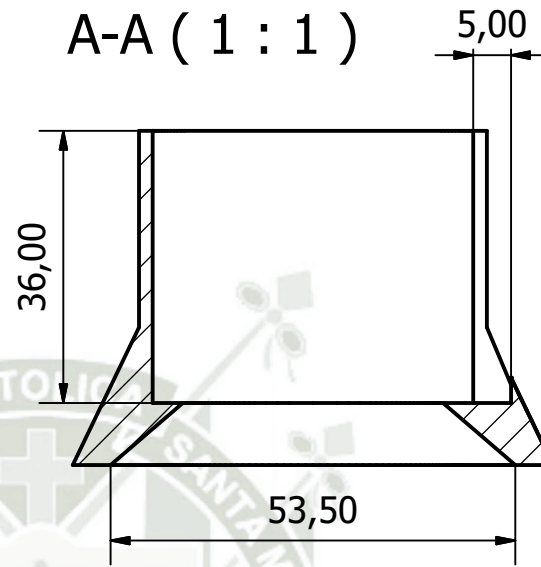
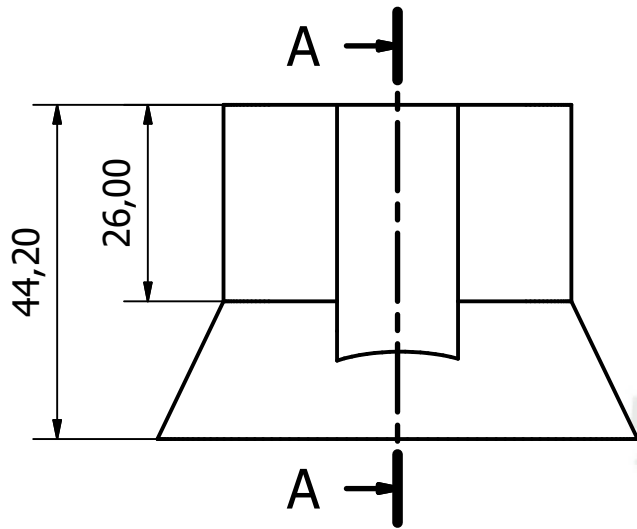
Universidad Católica  
de Santa María

FAJA TRASPORTADORA

SEPARADORES Y MOTOR PASO A PASO

Página:  
10/25

Escala:  
1 : 1

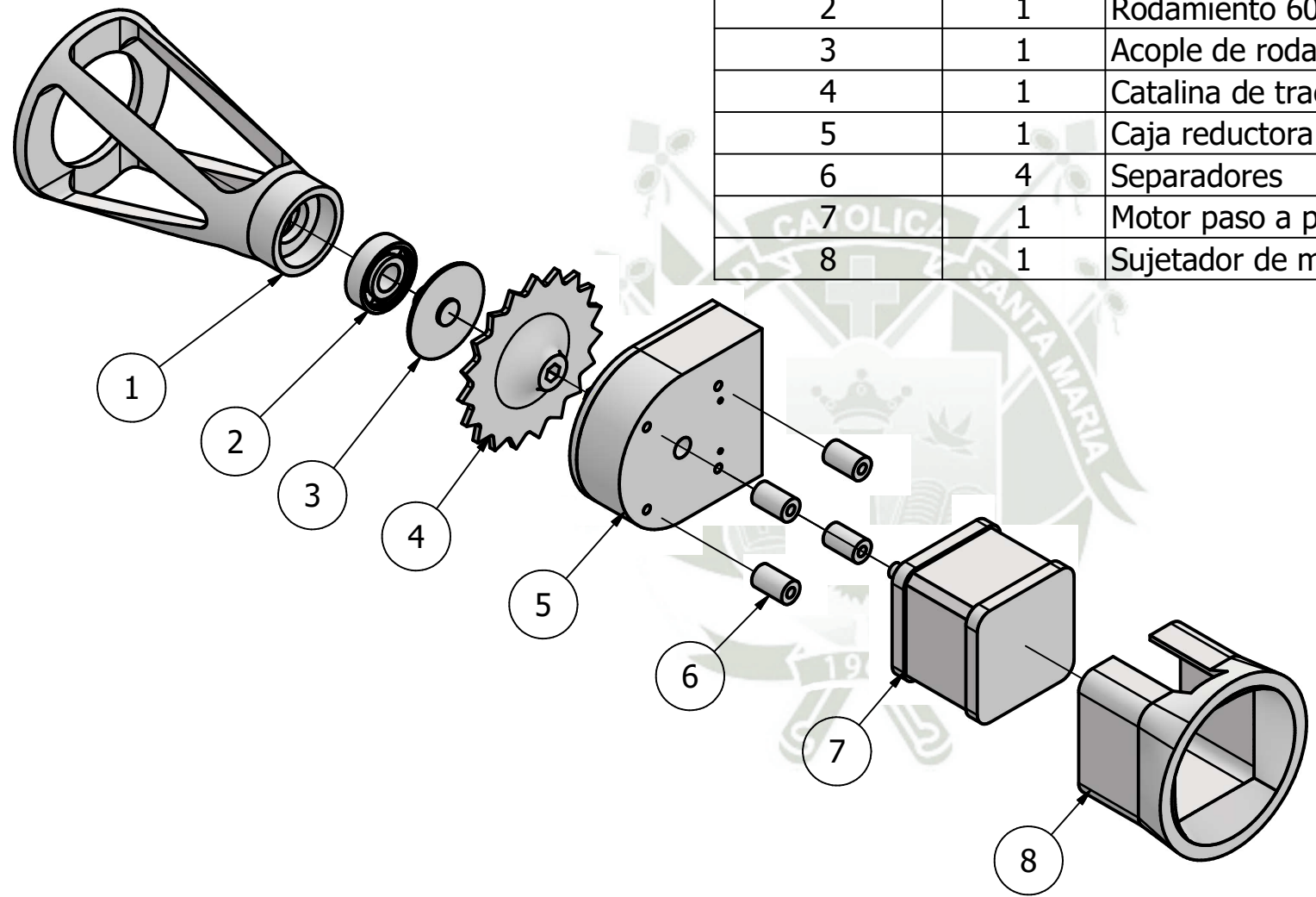


Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 18/09/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>SUJETADOR DE MOTOR</b>		Página: 11/25      Escala: 1 : 1



### LISTA DE PIEZAS

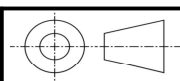
ELEMENTO	CANTIDAD	PIEZA	HOJA
1	1	Sujetador de rodamiento	7
2	1	Rodamiento 609	8
3	1	Acople de rodamiento con catalina	8
4	1	Catalina de traccion	9
5	1	Caja reductora	6
6	4	Separadores	10
7	1	Motor paso a paso NEMA 17	10
8	1	Sujetador de motor	11



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
29/07/2024

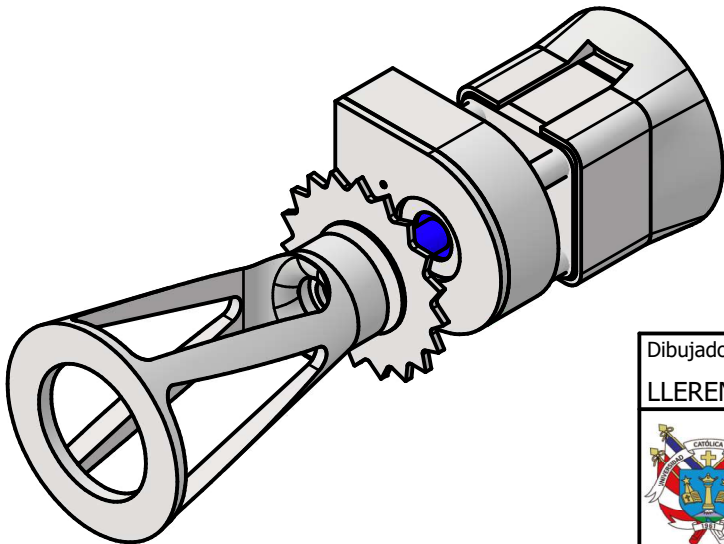
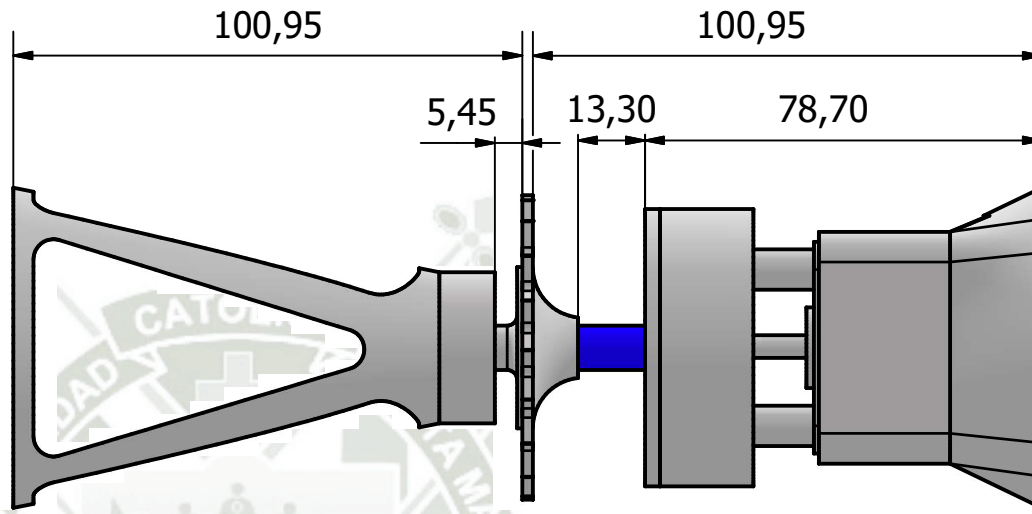
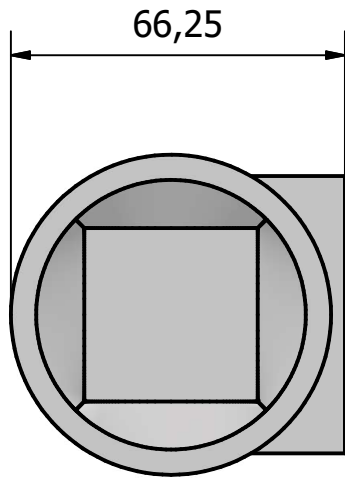


FAJA TRASPORTADORA

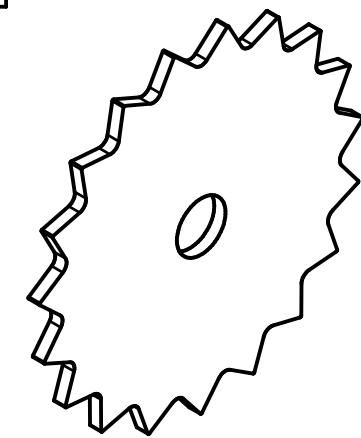
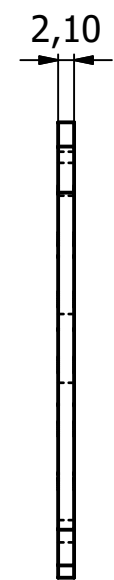
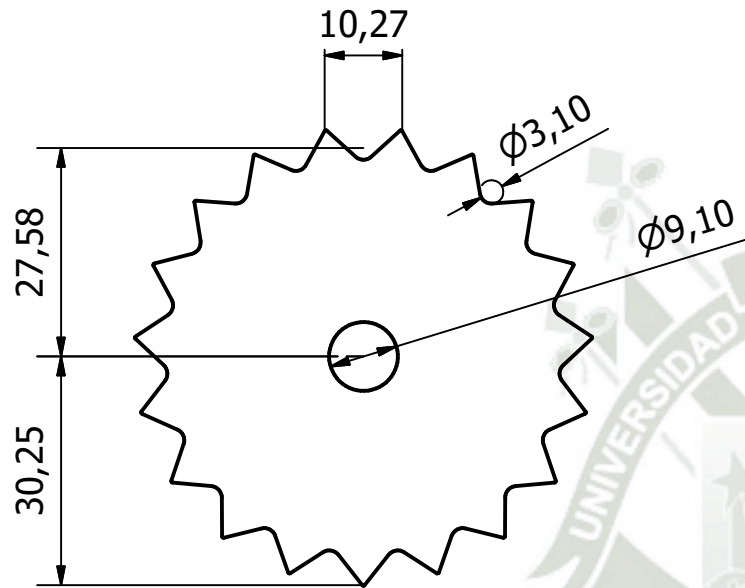
EXPLOSION DE SISTEMA DE TRACCION

Página: 12/25 Escala: 1 : 2





Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/07/2024	
 <b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>ENSAMBLE DE SISTEMA DE TRACCION</b>	Página: 13/25	Escala: 1 : 1.5

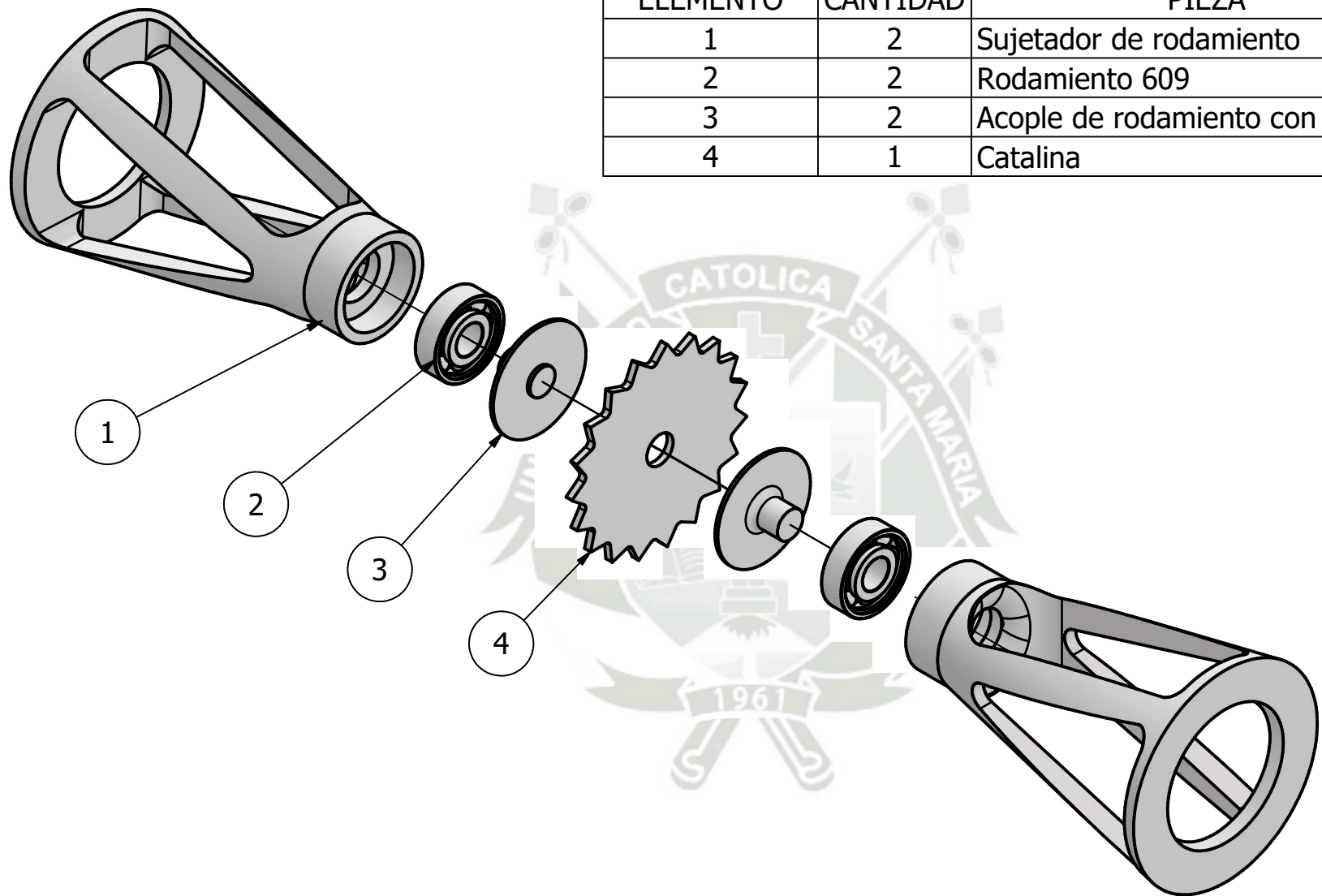


Dibujado por: LLERENA SANCHEZ JULIO CESAR		Revisado por: Ing. Christiam Collado Oporto		Fecha: 5/01/2025			
		<b>FAJA TRASPORTADORA</b>					
		CATALINA				Página: 14/25	Escala: 1 : 1



LISTA DE PIEZAS

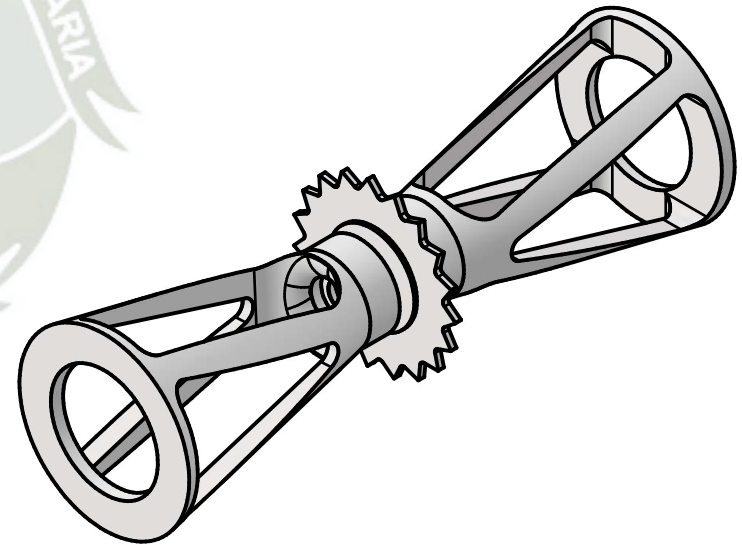
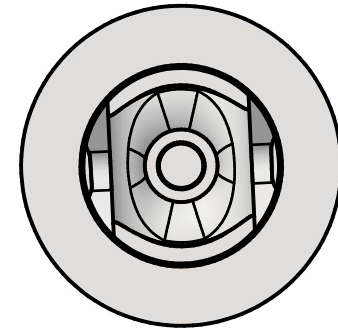
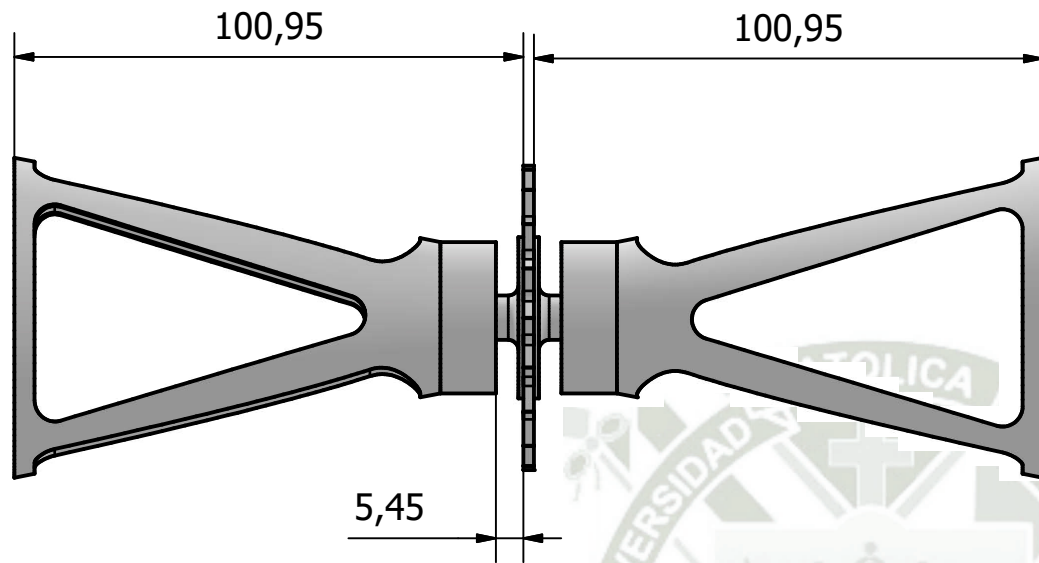
ELEMENTO	CANTIDAD	PIEZA	HOJA
1	2	Sujetador de rodamiento	7
2	2	Rodamiento 609	8
3	2	Acople de rodamiento con catalina	8
4	1	Catalina	14

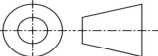



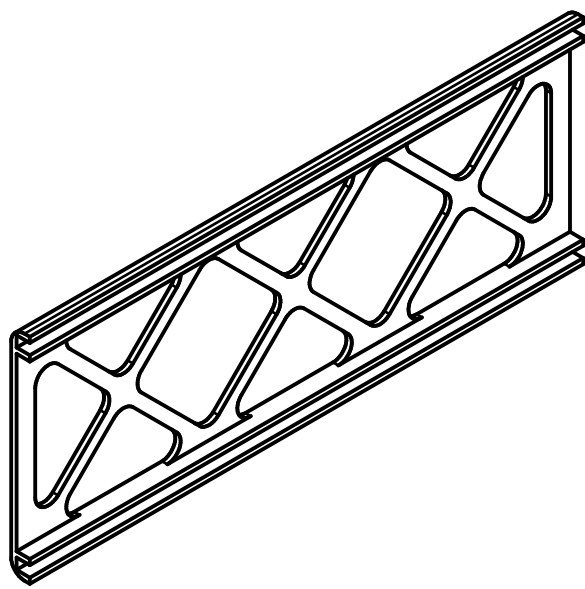
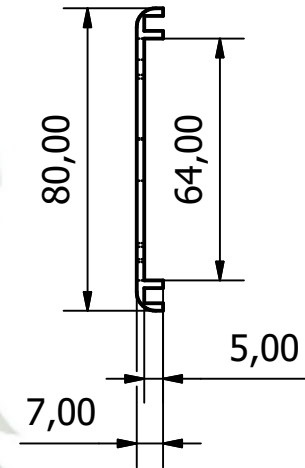
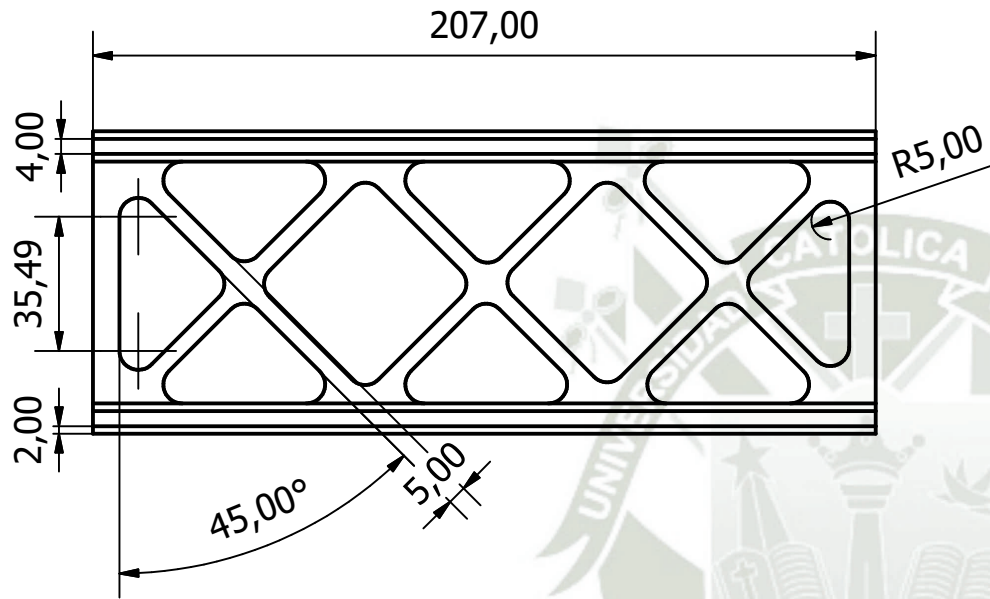
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/07/2024	
--	--	----------------------	--

	FAJA TRASPOTADORA	
	EXPLOSION DE SISTEMA DE RETORNO	Página: 15/25 Escala: 1 : 1.5

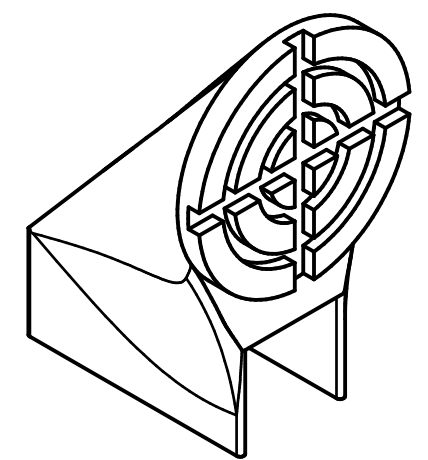
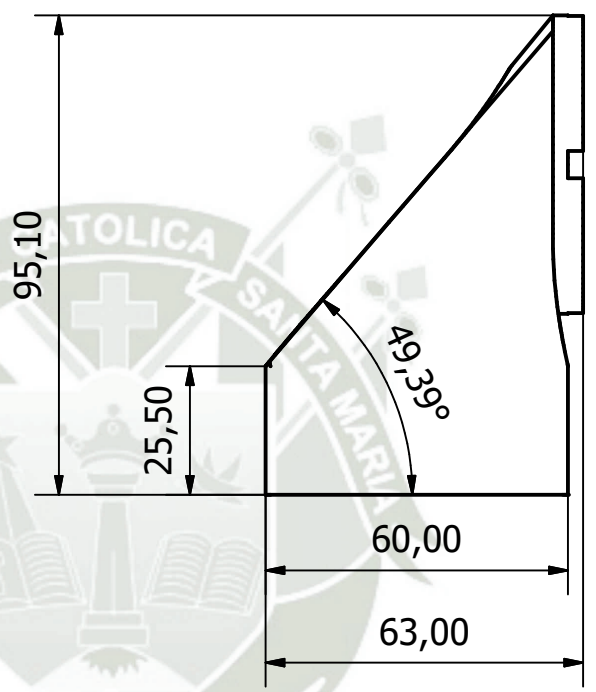
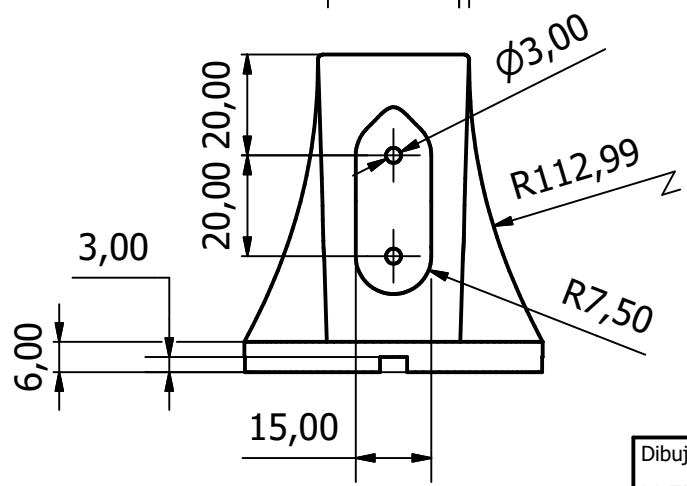
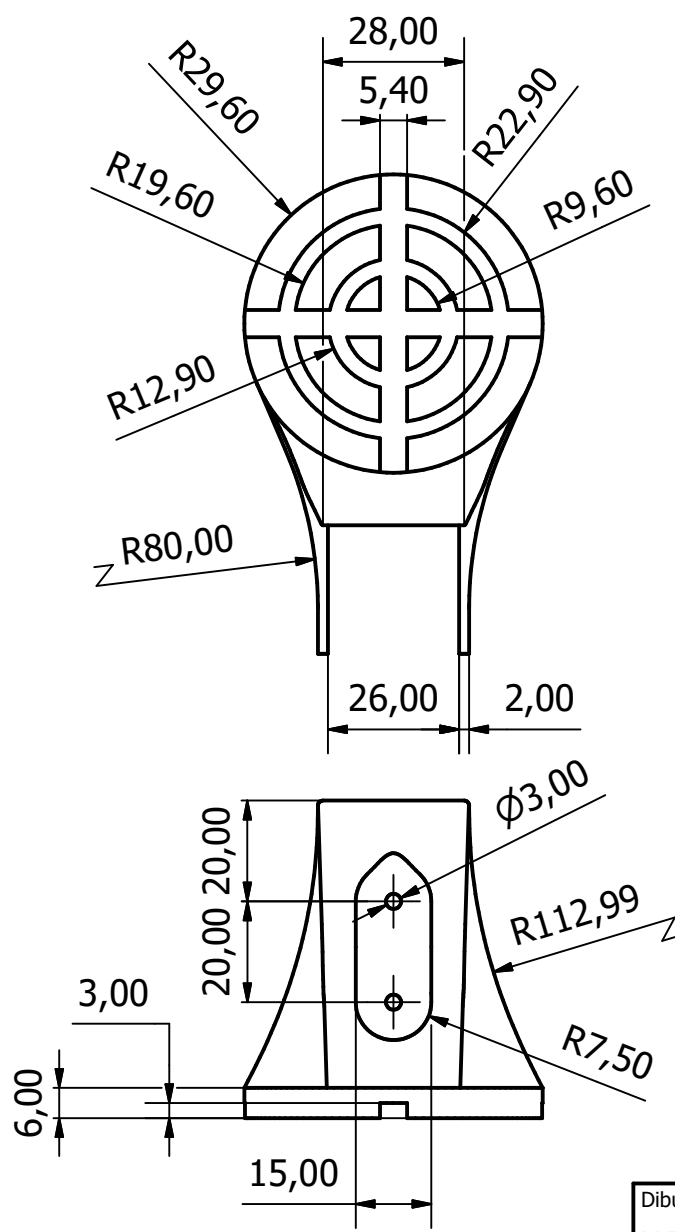




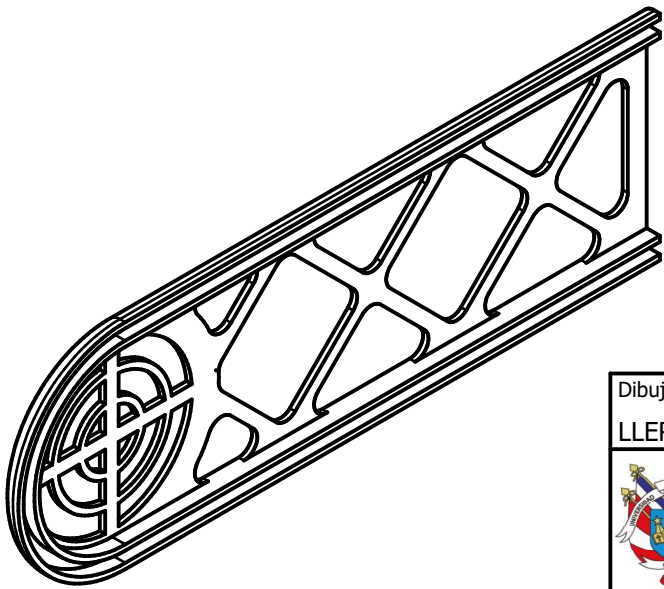
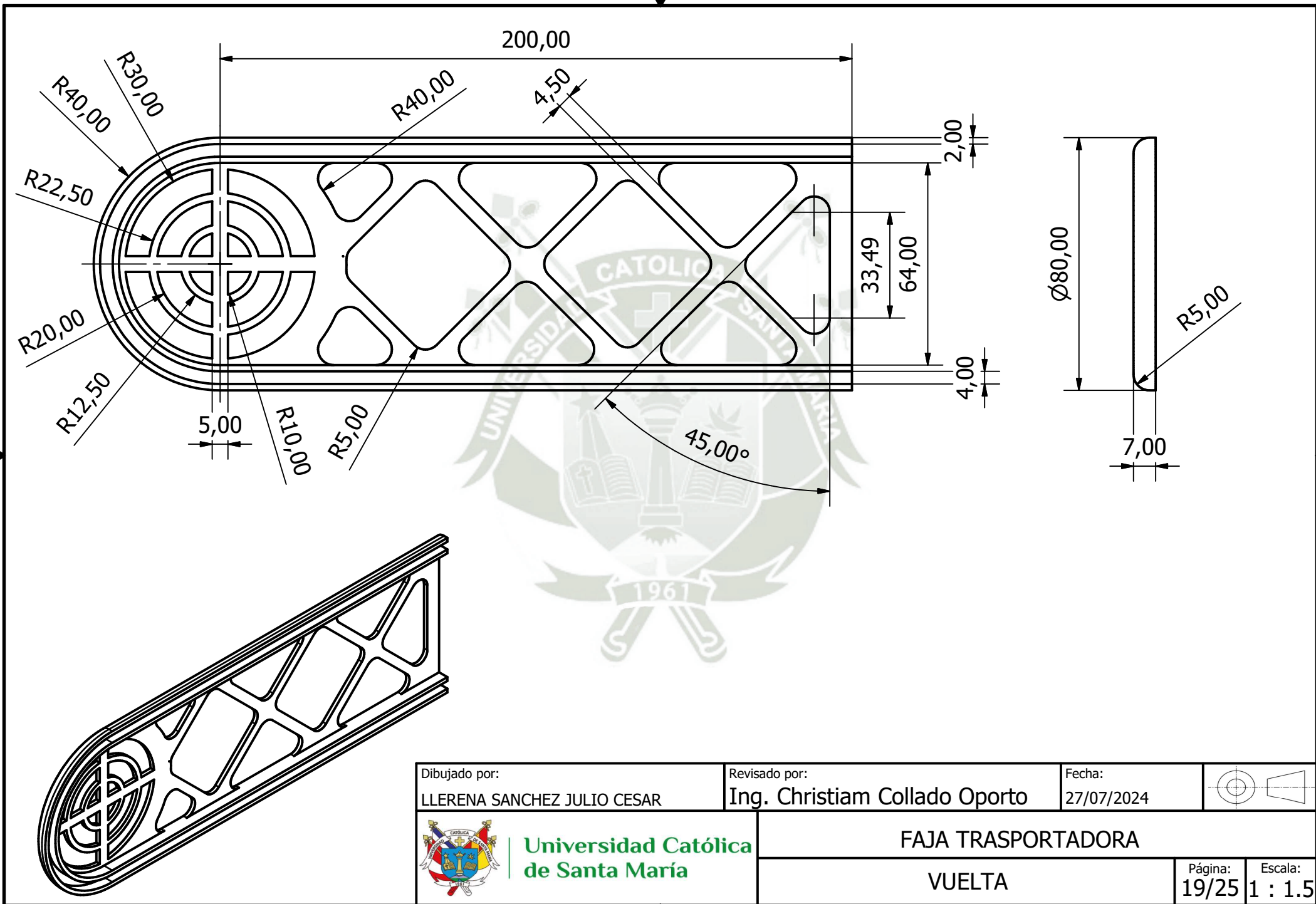
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/07/2024	
 <b>Universidad Católica de Santa María</b>	<b>FAJA TRASPOTADORA</b>		
<b>ENSAMBLE DE SISTEMA DE RETORNO</b>		Página: 16/25	Escala: 1 : 1.5

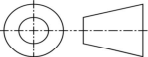



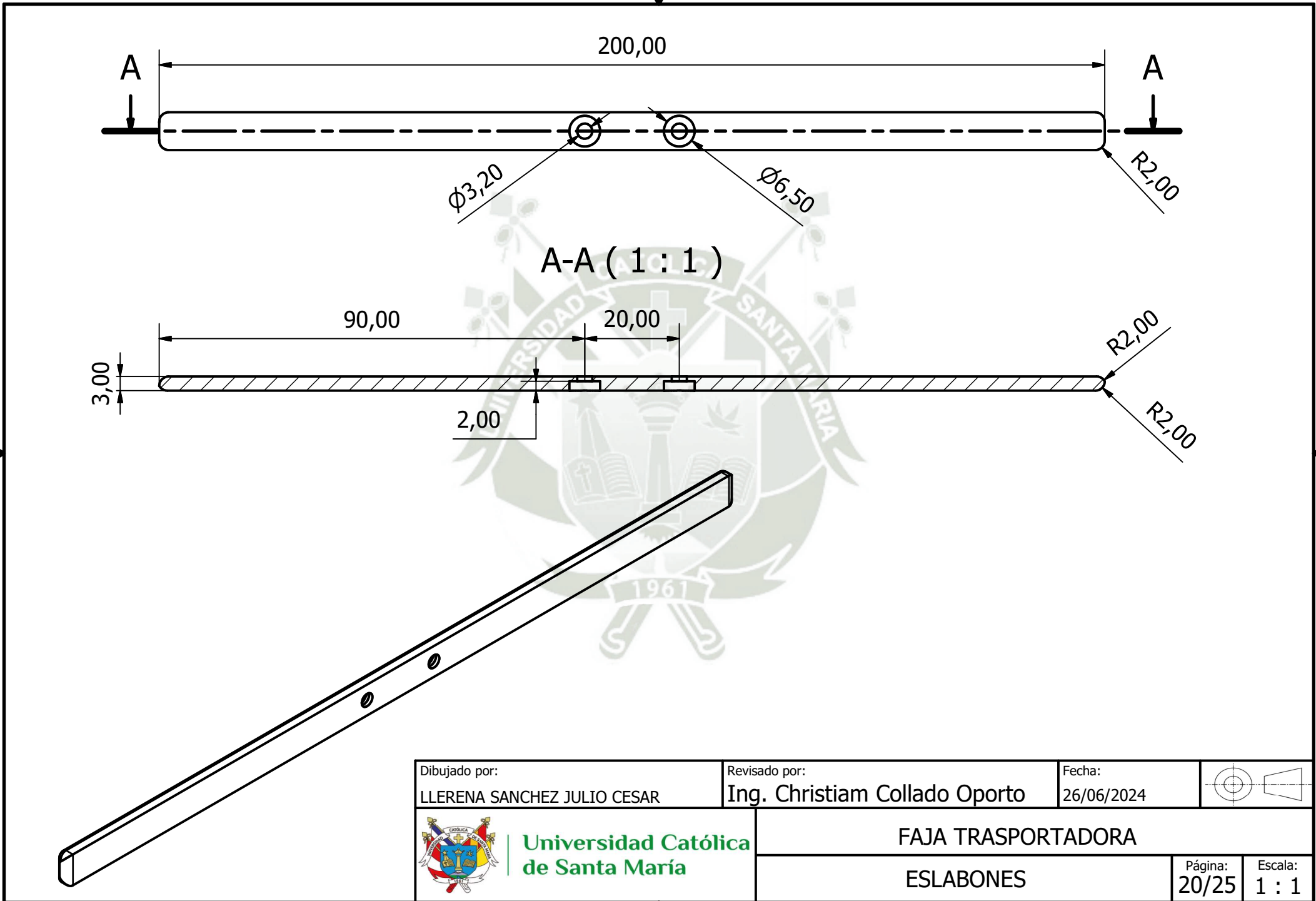
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 27/07/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>MITADES</b>	Página: <b>17/25</b>	Escala: <b>1 : 2</b>



Dibujado por: LLERENA SANCHEZ JULIO CESAR		Revisado por: Ing. Christiam Collado Oporto		Fecha: 29/07/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>			Página: 18/25	Escala: 1 : 1.5
	<b>SUJETADOR DE PERFILES</b>				



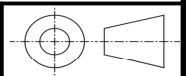
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 27/07/2024	
 <b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>VUELTA</b>	Página: 19/25	Escala: 1 : 1.5



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
26/06/2024



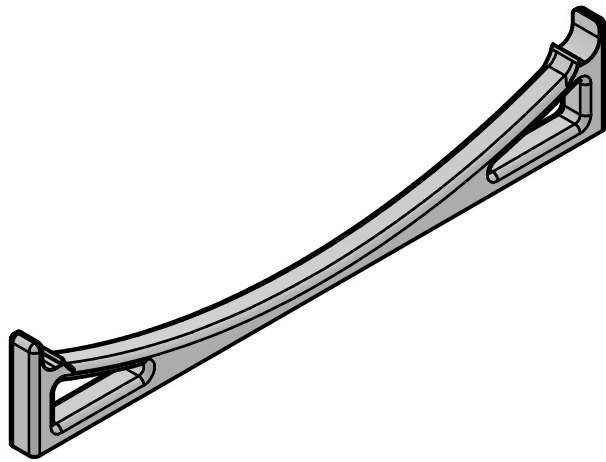
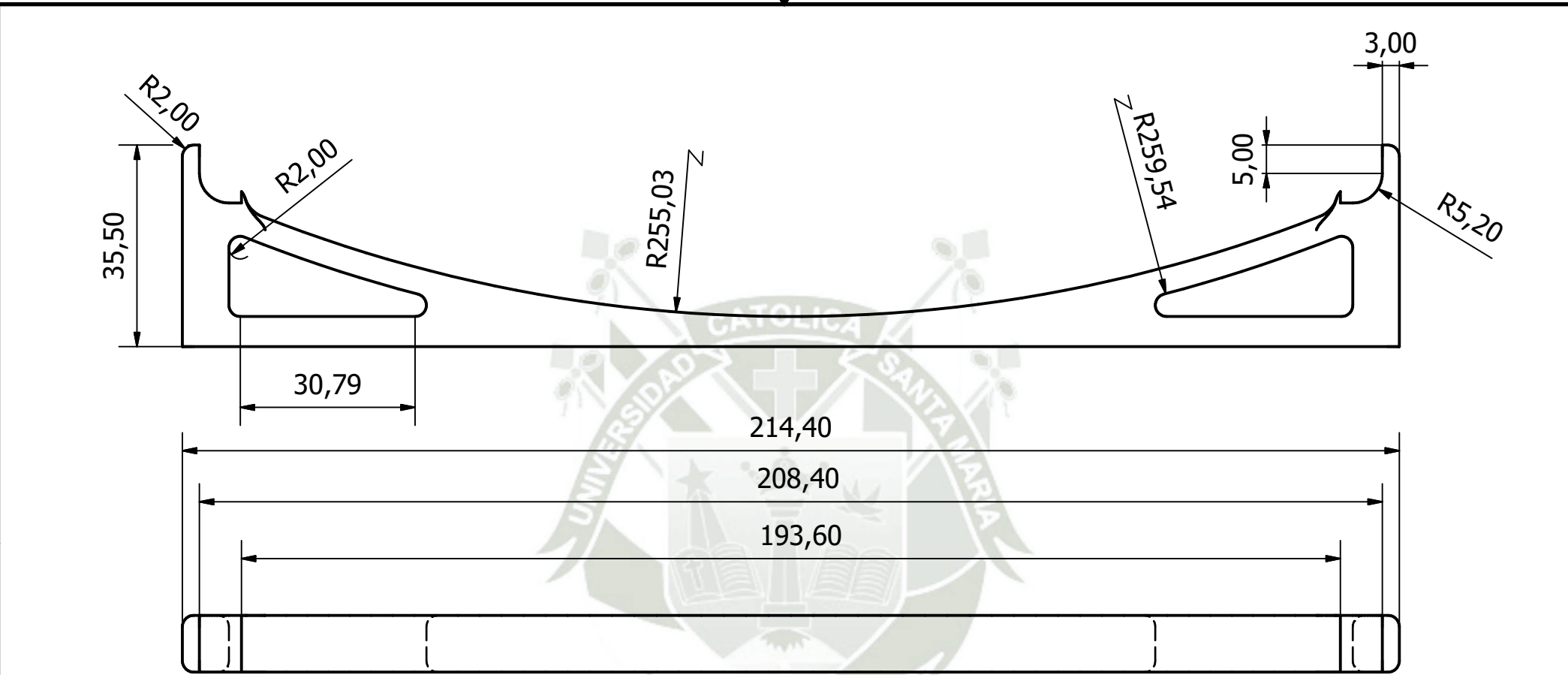
**Universidad Católica  
de Santa María**

**FAJA TRASPORTADORA**

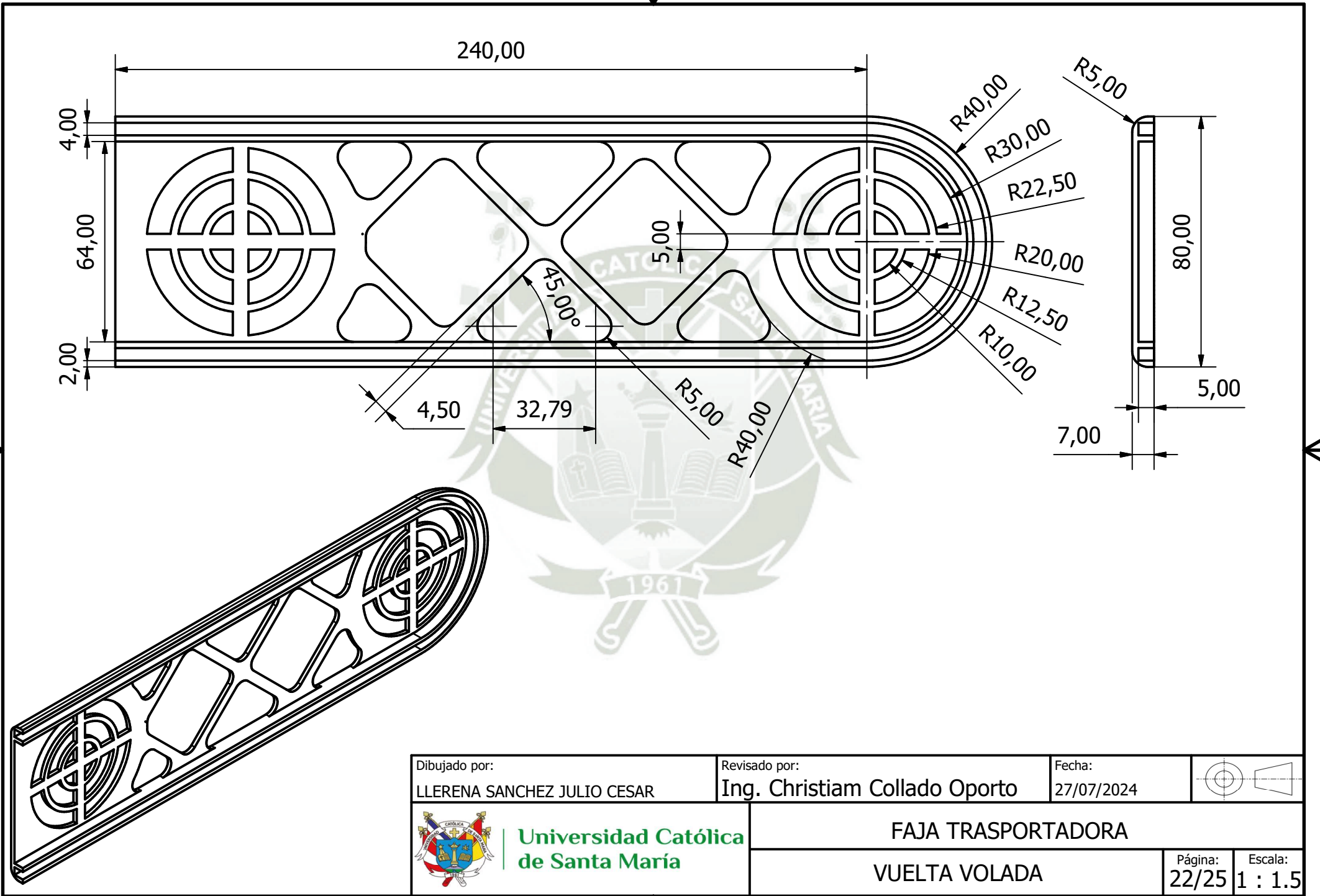
**ESLABONES**

Página:  
**20/25**

Escala:  
**1 : 1**



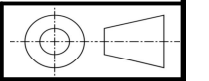
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 30/07/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>CRUZETA</b>	Página: 21/25	Escala: 1 : 1



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
27/07/2024



**Universidad Católica  
de Santa María**

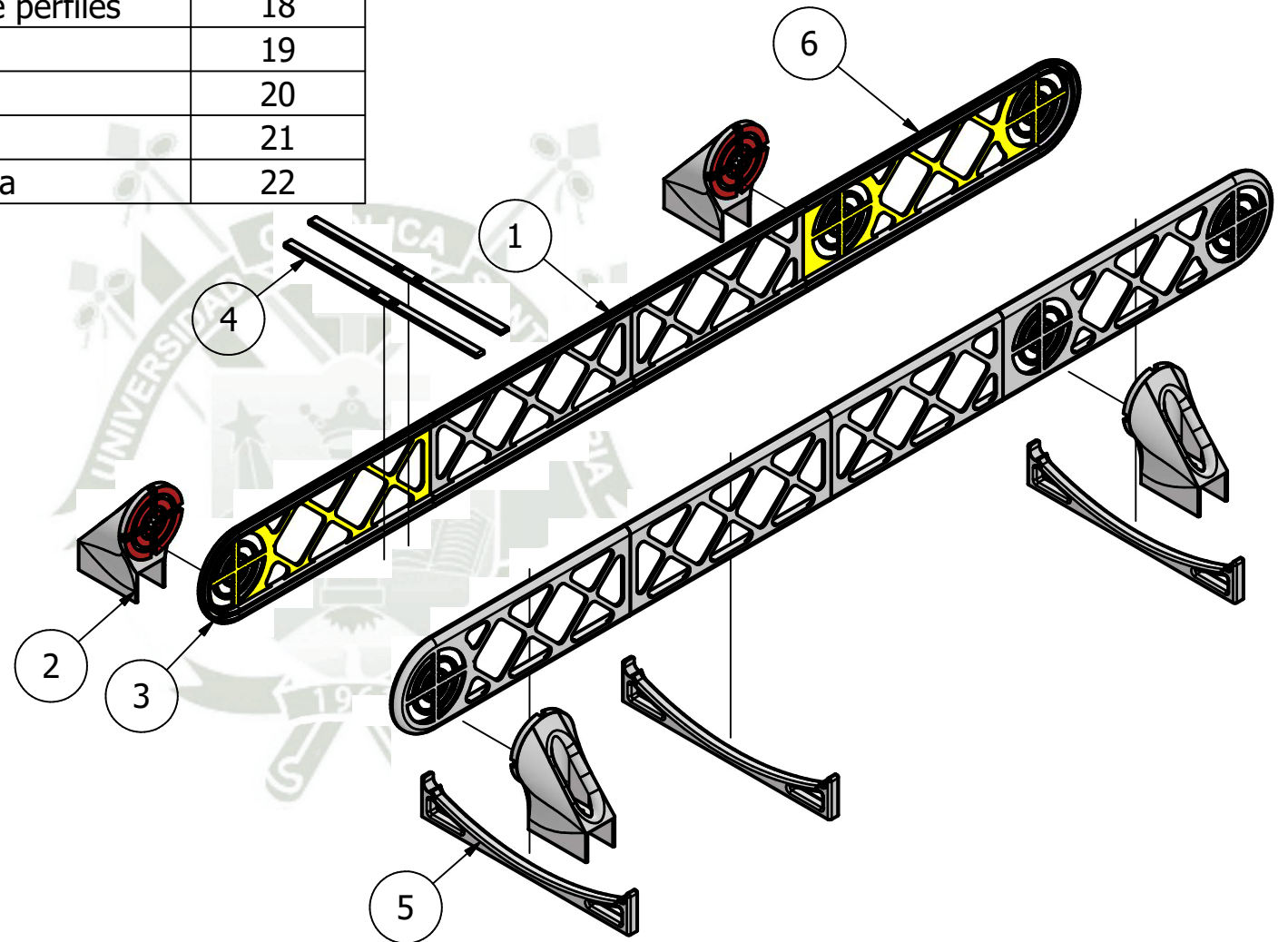
**FAJA TRASPORTADORA**

**VUELTA VOLADA**

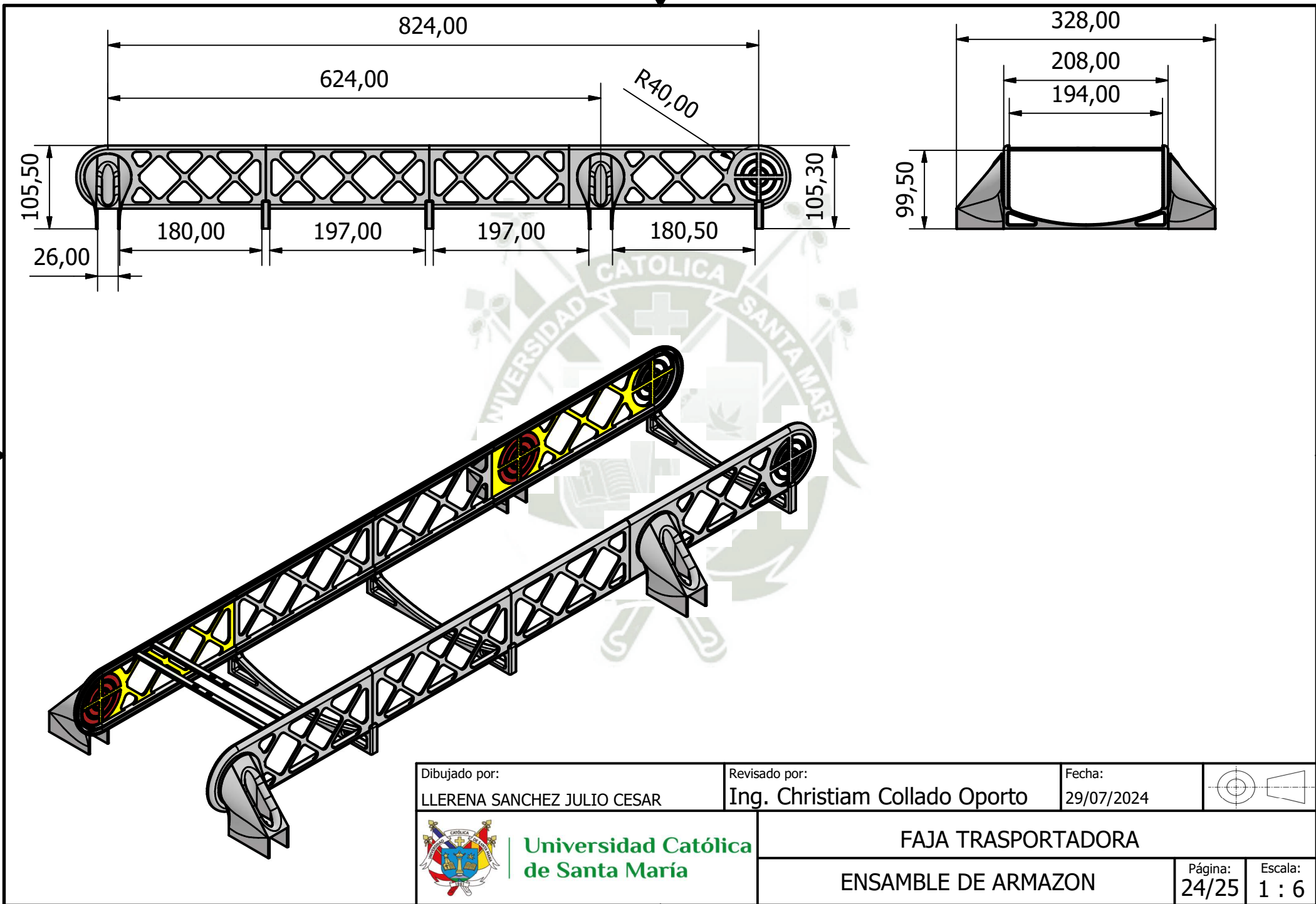
Página:  
**22/25** Escala:  
**1 : 1.5**

LISTA DE PIEZAS

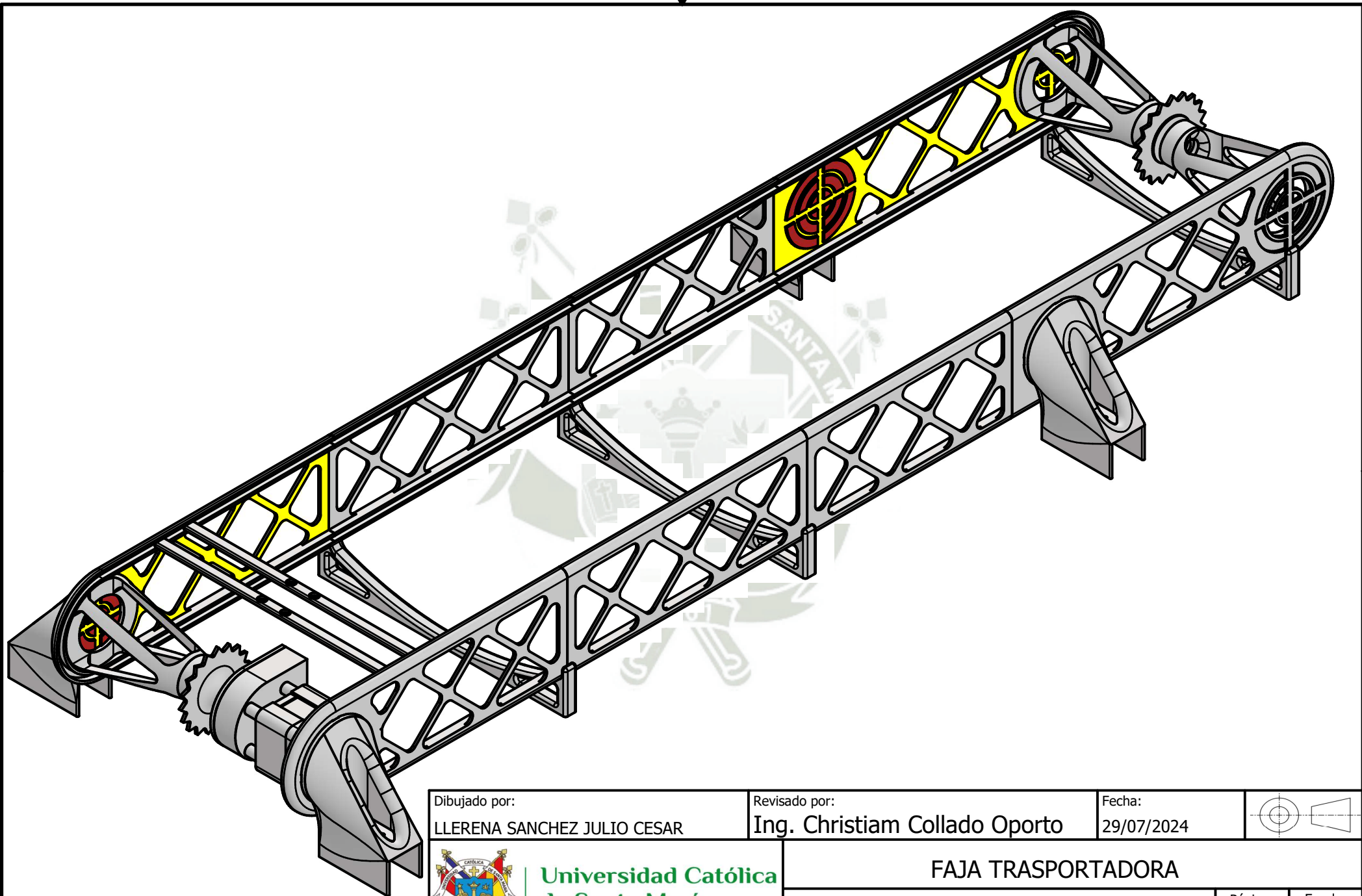
ELEMENTO	CANTIDAD	PIEZA	HOJA
1	4	Mitades	17
2	4	Sujetador de perfiles	18
3	2	Vuelta	19
4	126	Eslabones	20
5	3	Cruzeta	21
6	2	Vuelta volada	22



Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/07/2024	
	<b>FAJA TRASPORTADORA</b>		
	EXPLOSION DE ARMAZON		Página: 23/25      Escala: 1 : 5



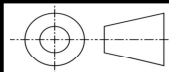
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 29/07/2024	
<b>Universidad Católica de Santa María</b>	<b>FAJA TRASPORTADORA</b>		
	<b>ENSAMBLE DE ARMAZON</b>		Página: 24/25



Dibujado por:  
LLERENA SANCHEZ JULIO CESAR

Revisado por:  
Ing. Christiam Collado Oporto

Fecha:  
29/07/2024



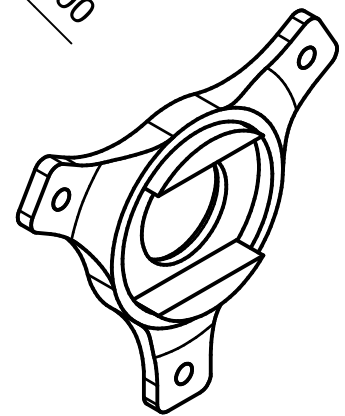
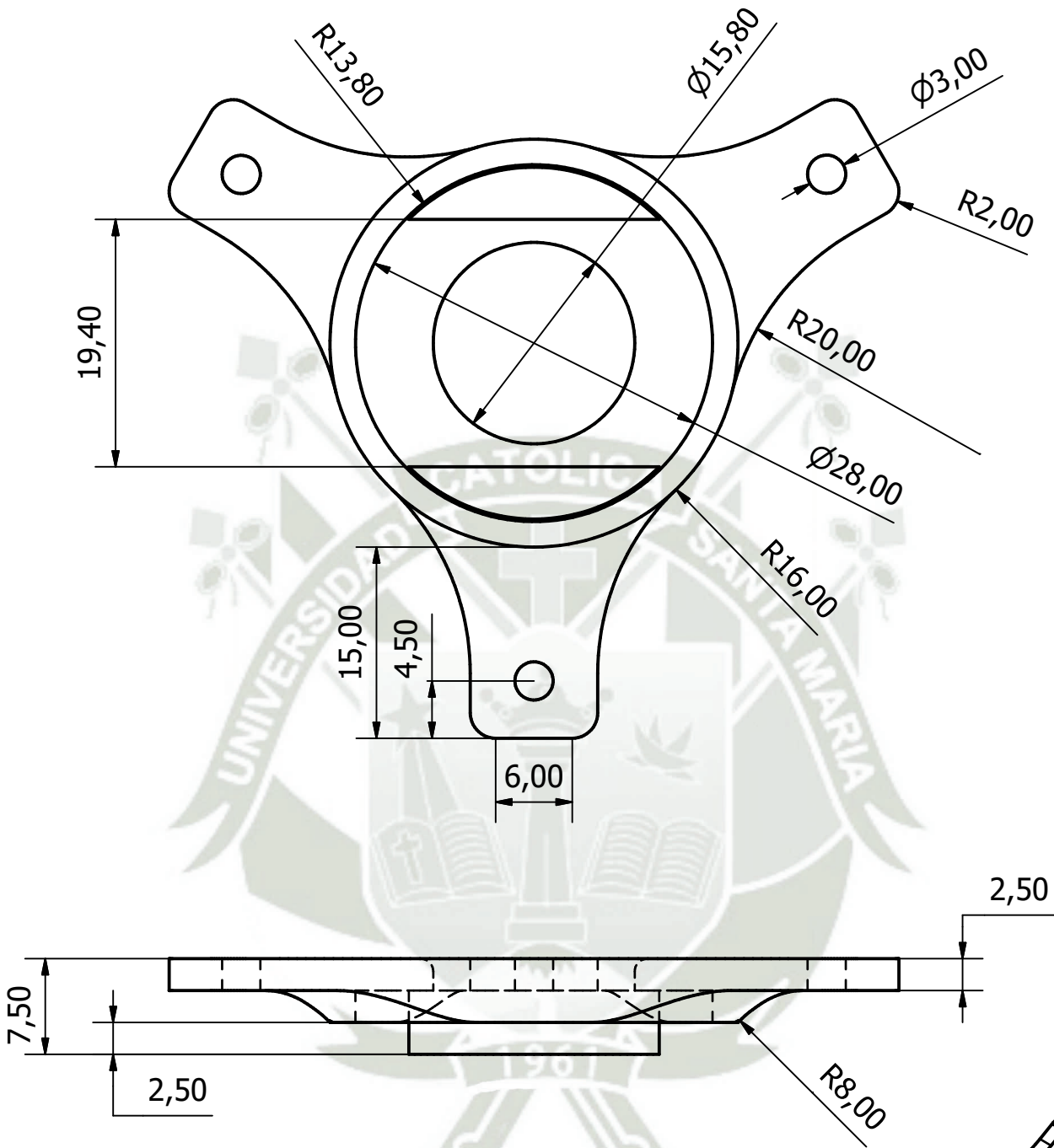
Universidad Católica  
de Santa María


FAJA TRASPOTADORA

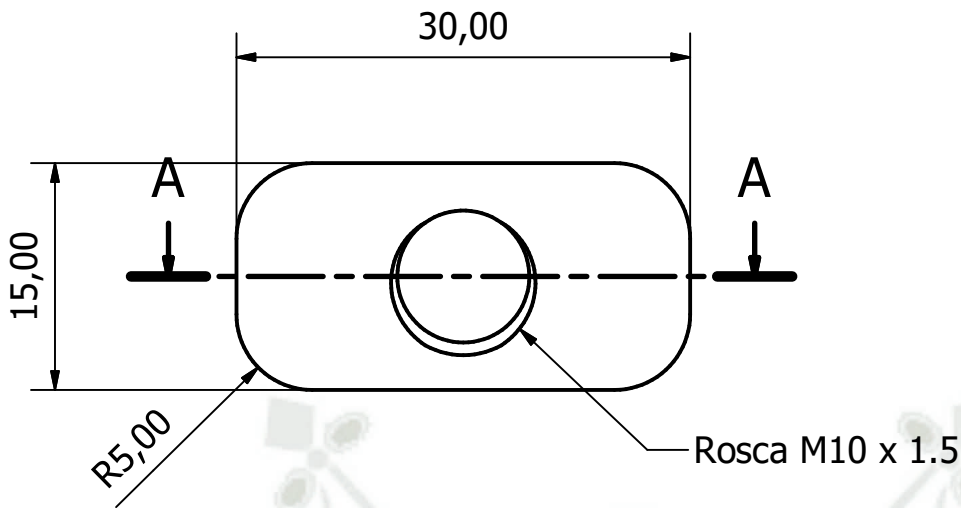
ENSAMBLE GENERAL DE LA FAJA

Página:  
25/25

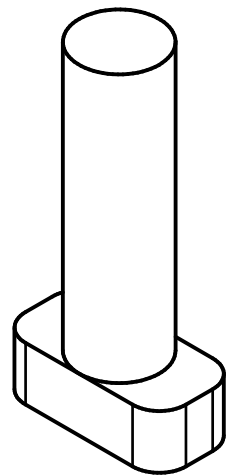
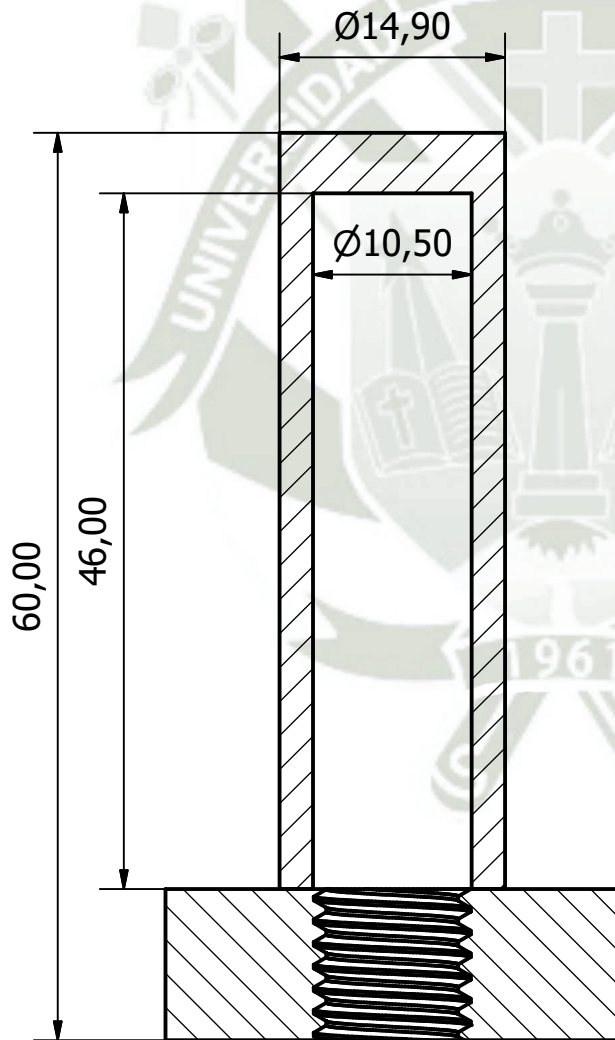
Escala:  
1 : 3



Dibujado LLERENA SANCHEZ JULIO CESAR	Revisado: Ing. Christiam Collado Oporto	Fecha: 10/08/2024	
 <b>Universidad Católica de Santa María</b>	<b>ACTUADOR MAGNETICO</b>		
	<b>BASE DEL ACTUADOR</b>	Página: 1/6	Escala: 2:1



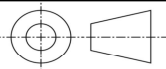
A-A ( 2 : 1 )



Dibujado  
LLERENA SANCHEZ JULIO CESAR

Revisado:  
Ing. Christiam Collado Oporto

Fecha:  
10/08/2024



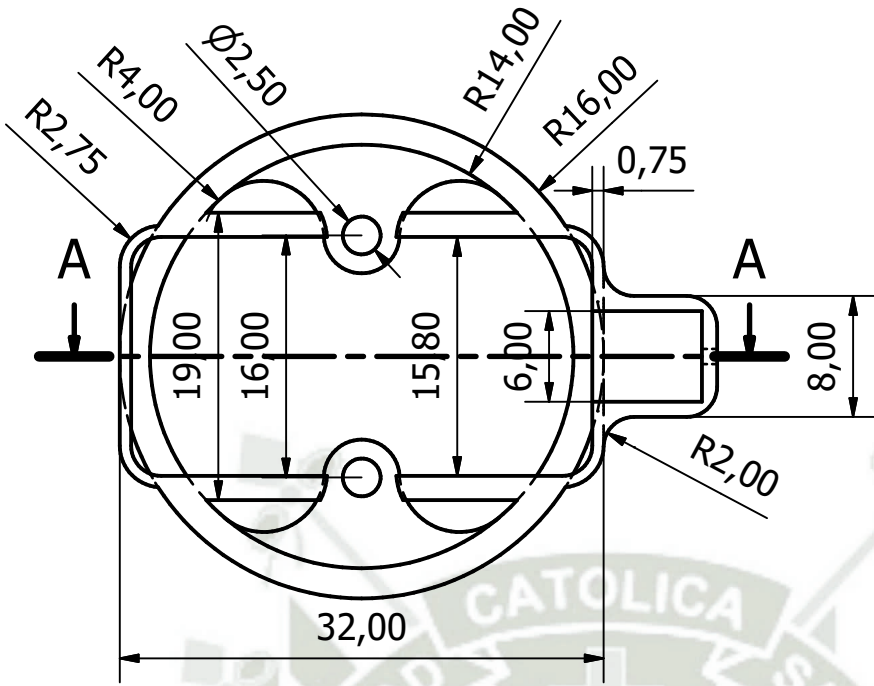
Universidad Católica  
de Santa María

ACTUADOR MAGNETICO

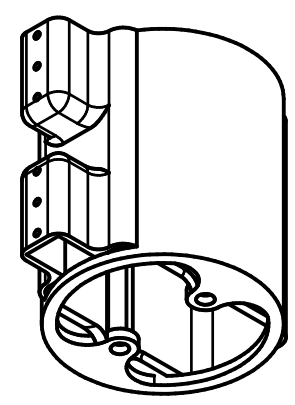
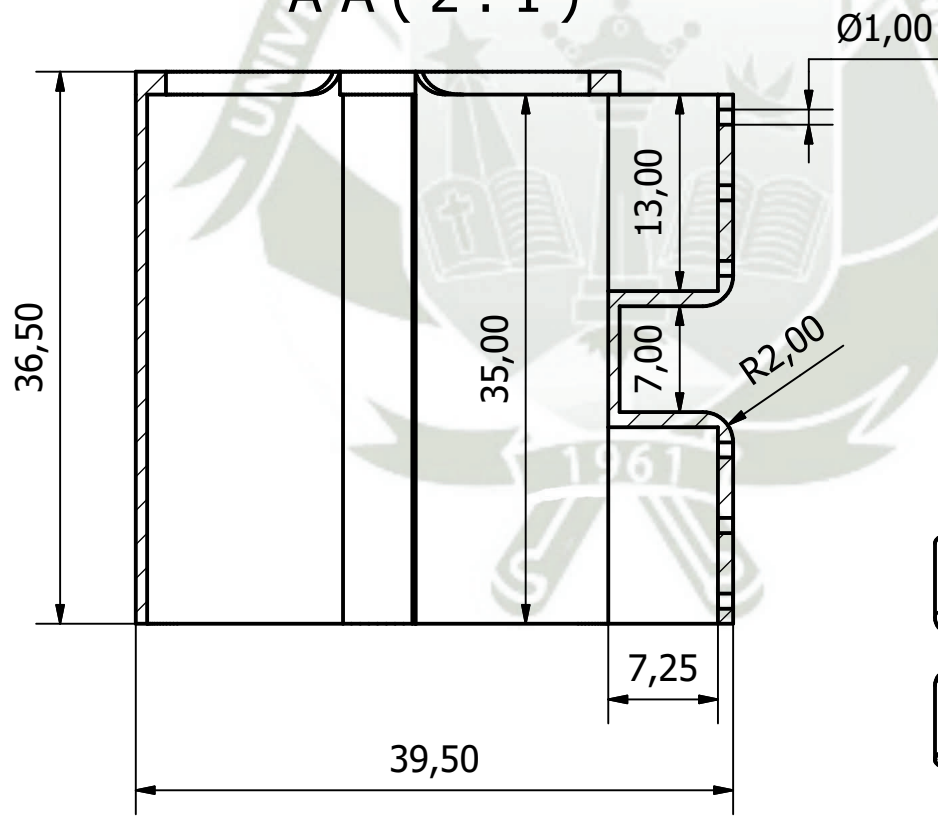
VÁSTAGO


Página:  
2/6

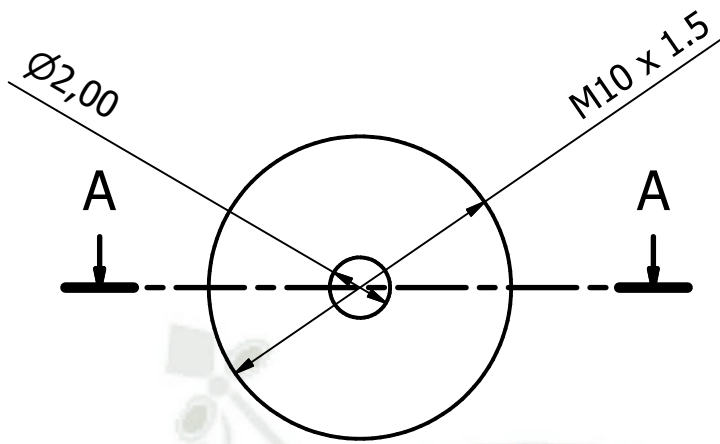
Escala:  
2:1



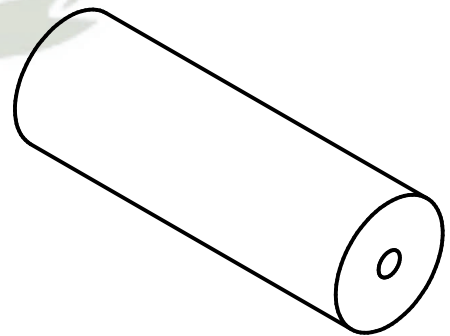
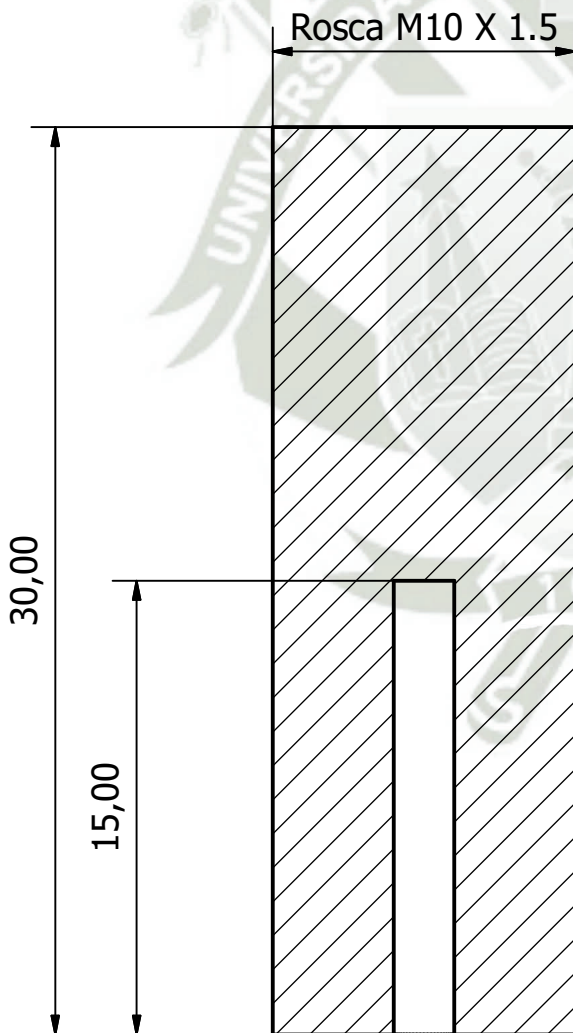
A-A ( 2 : 1 )




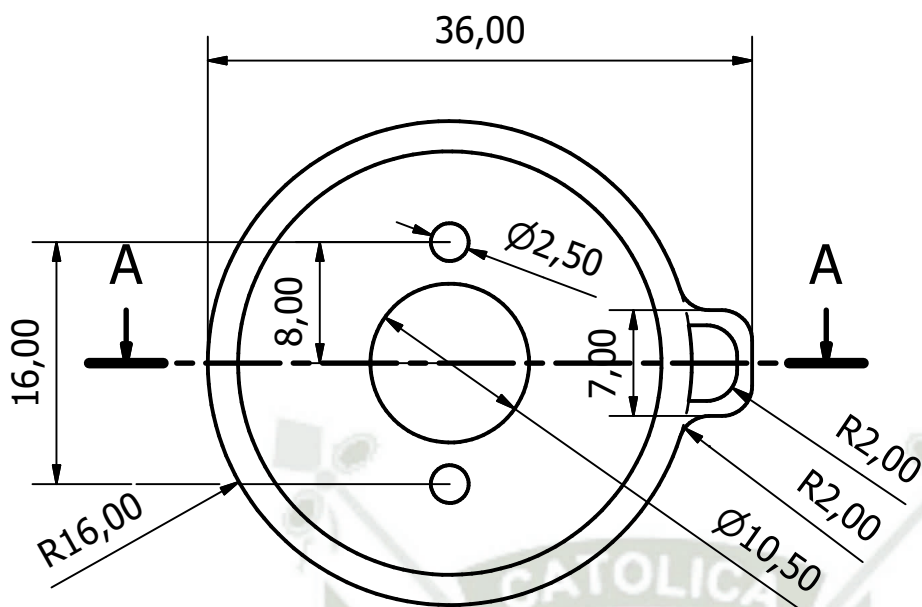
Dibujado LLERENA SANCHEZ JULIO CESAR	Revisado: Ing. Christiam Collado Oporto	Fecha: 10/08/2024	
 <b>Universidad Católica de Santa María</b>	<b>ACTUADOR MAGNETICO</b>		
	CAVIDAD DE VÁSTAGO	Página: 3/6	Escala: 2:1



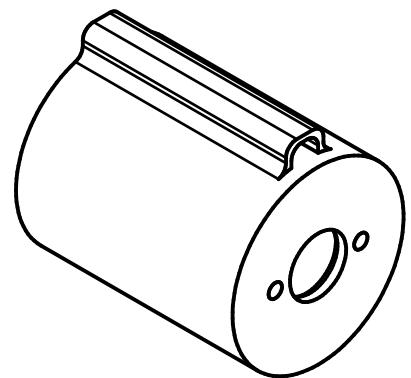
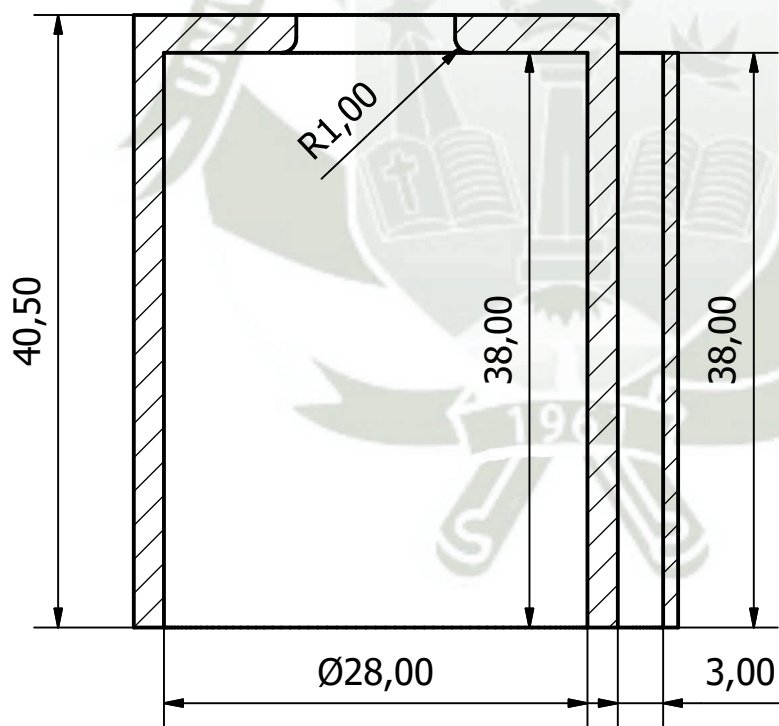
A-A ( 4 : 1 )



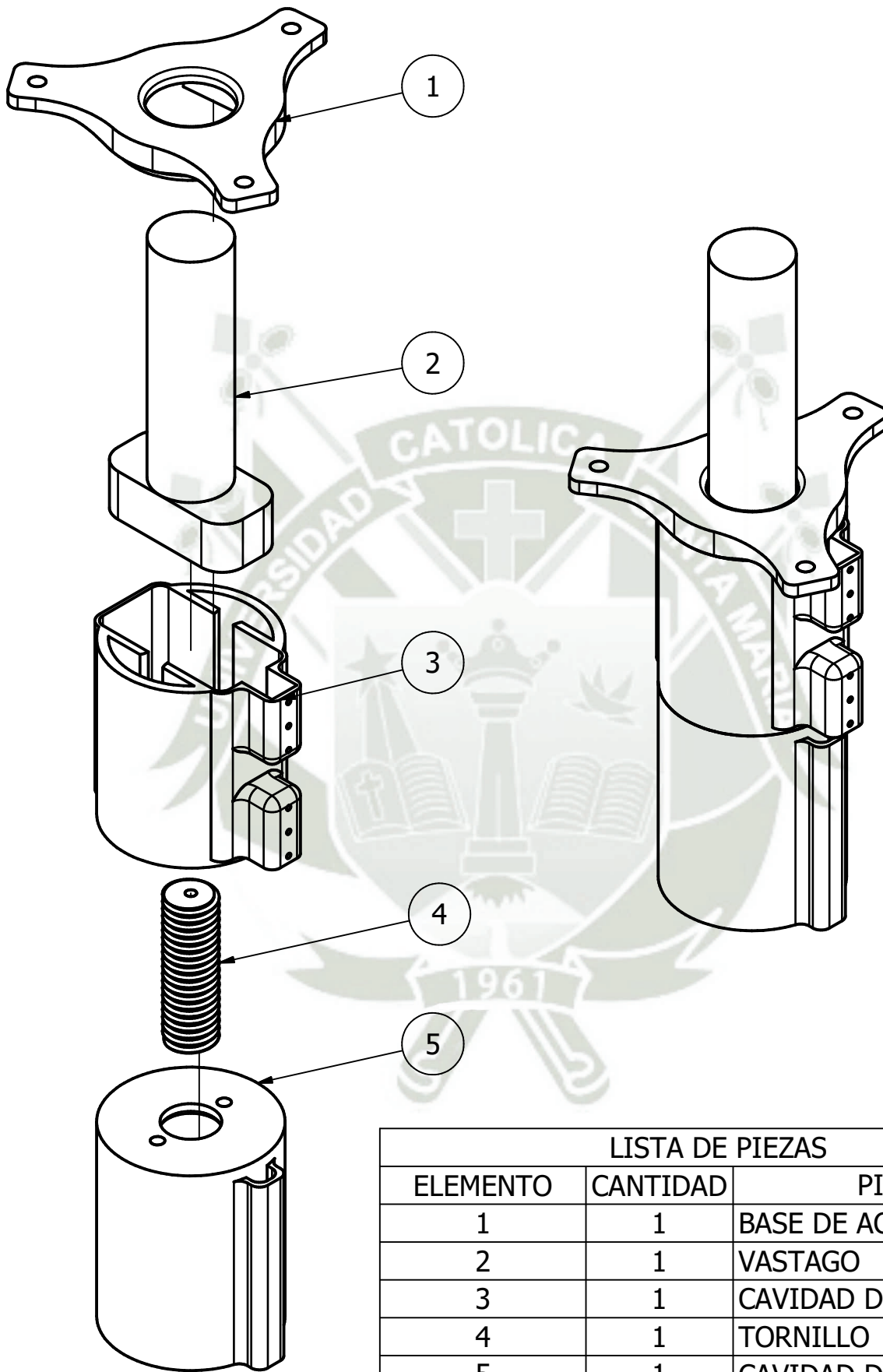
Dibujado LLERENA SANCHEZ JULIO CESAR	Revisado: Ing. Christiam Collado Oporto	Fecha: 10/08/2024	
 <b>Universidad Católica de Santa María</b>	<b>ACTUADOR MAGNETICO</b>		
	<b>TORNILLO</b>	Página: <b>4/6</b>	Escala: <b>4:1</b>



A-A (2 : 1)



Dibujado LLERENA SANCHEZ JULIO CESAR	Revisado: Ing. Christiam Collado Oporto	Fecha: 10/08/2024	
	ACTUADOR MAGNETICO		
	CAVIDAD DE MOTOR	Página: 5/6	Escala: 2:1



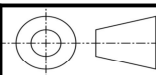
LISTA DE PIEZAS

ELEMENTO	CANTIDAD	PIEZA
1	1	BASE DE ACTUADOR
2	1	VASTAGO
3	1	CAVIDAD DE VASTAGO
4	1	TORNILLO
5	1	CAVIDAD DE MOTOR

Dibujado  
LLERENA SANCHEZ JULIO CESAR

Revisado:  
Ing. Christiam Collado Oporto

Fecha:  
10/08/2024



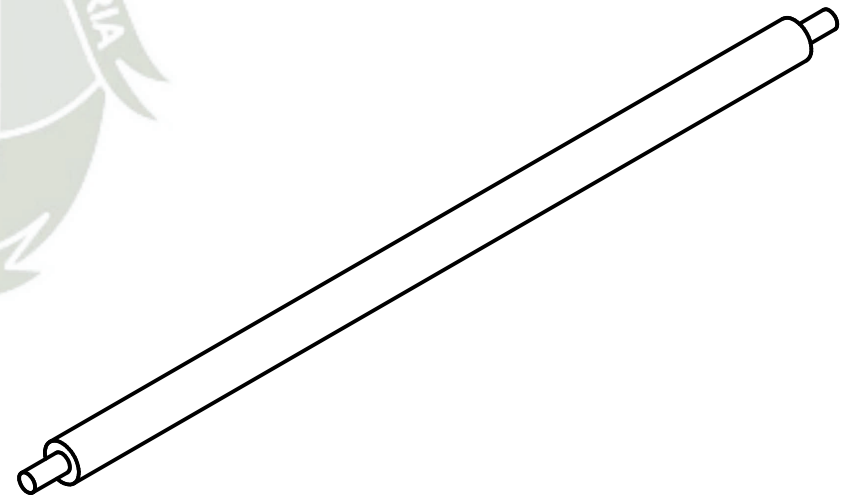
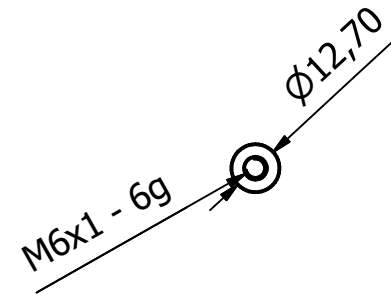
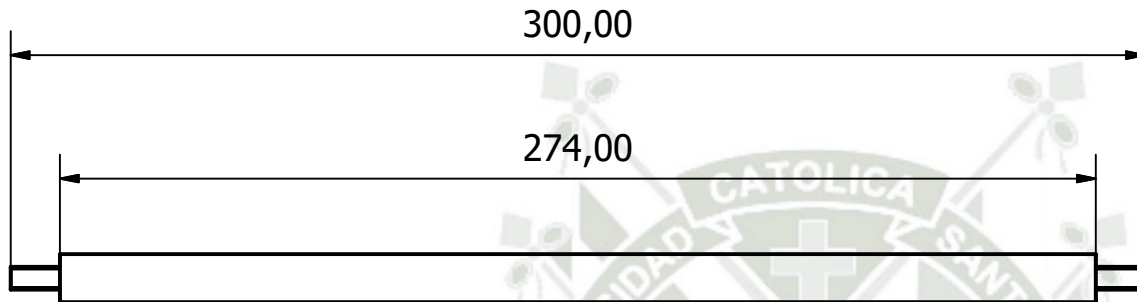
**Universidad Católica  
de Santa María**

ACTUADOR MAGNETICO

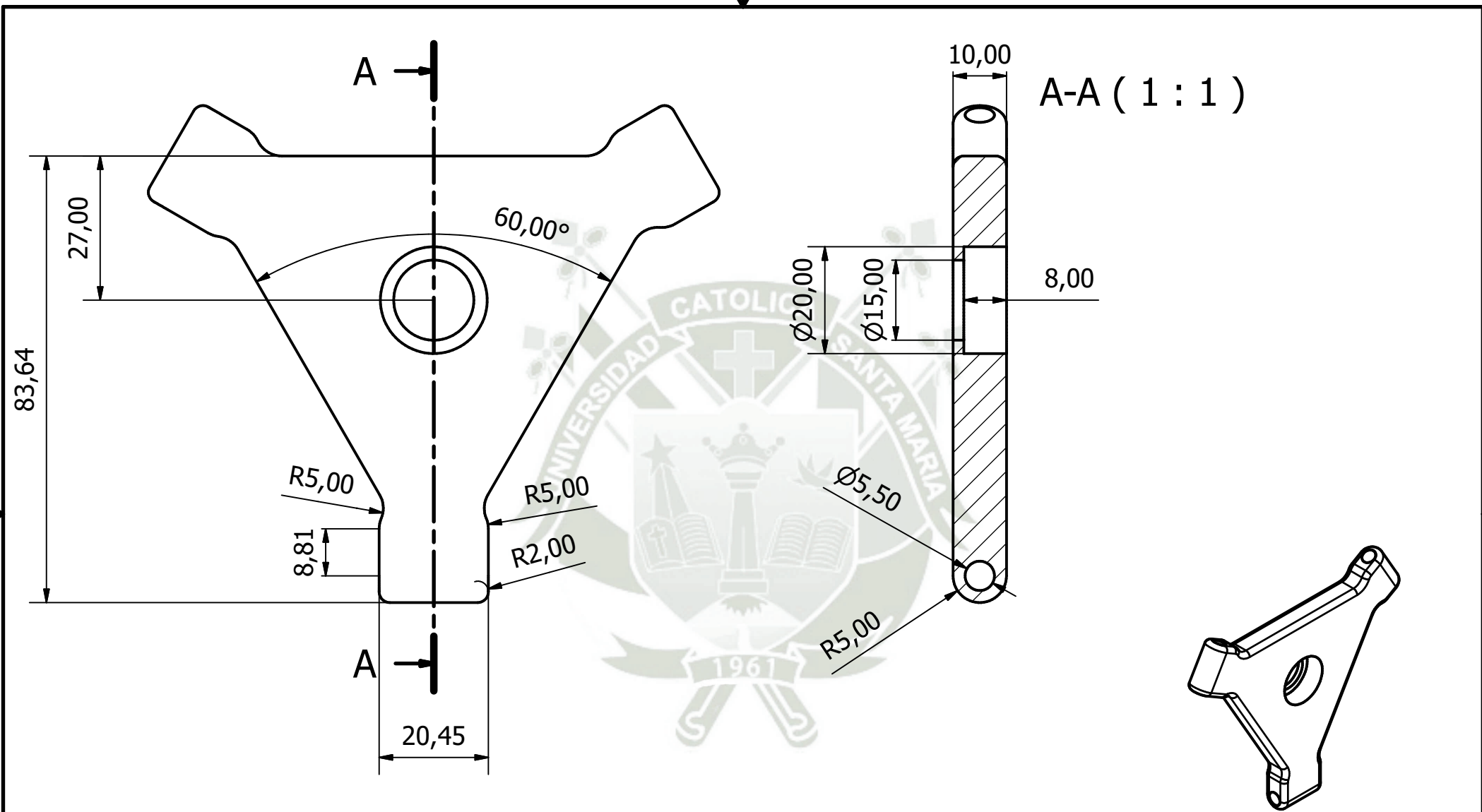
PLANO DE DESPIECE

Página:  
6/6

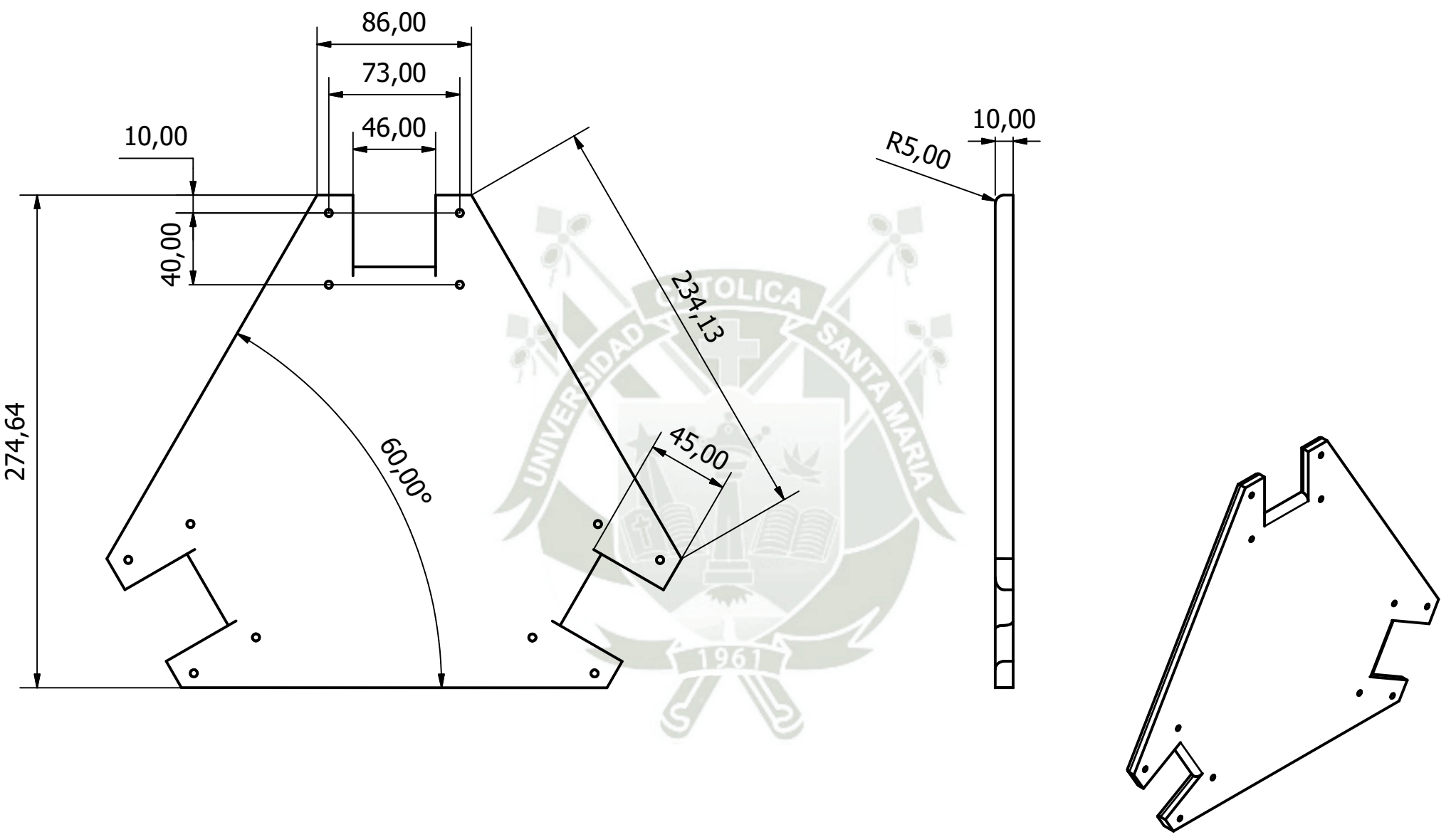
Escala:  
1:1




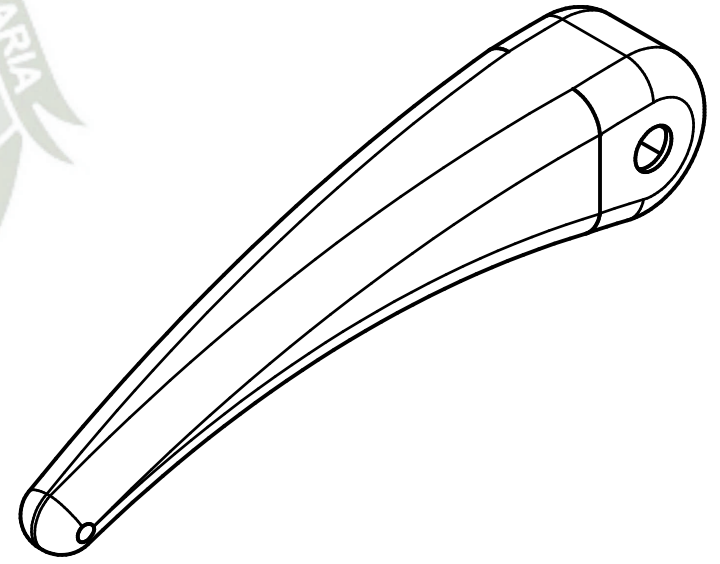
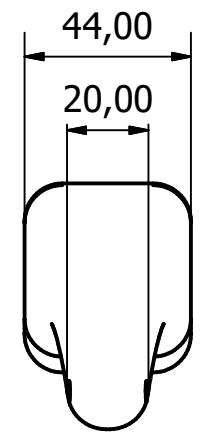
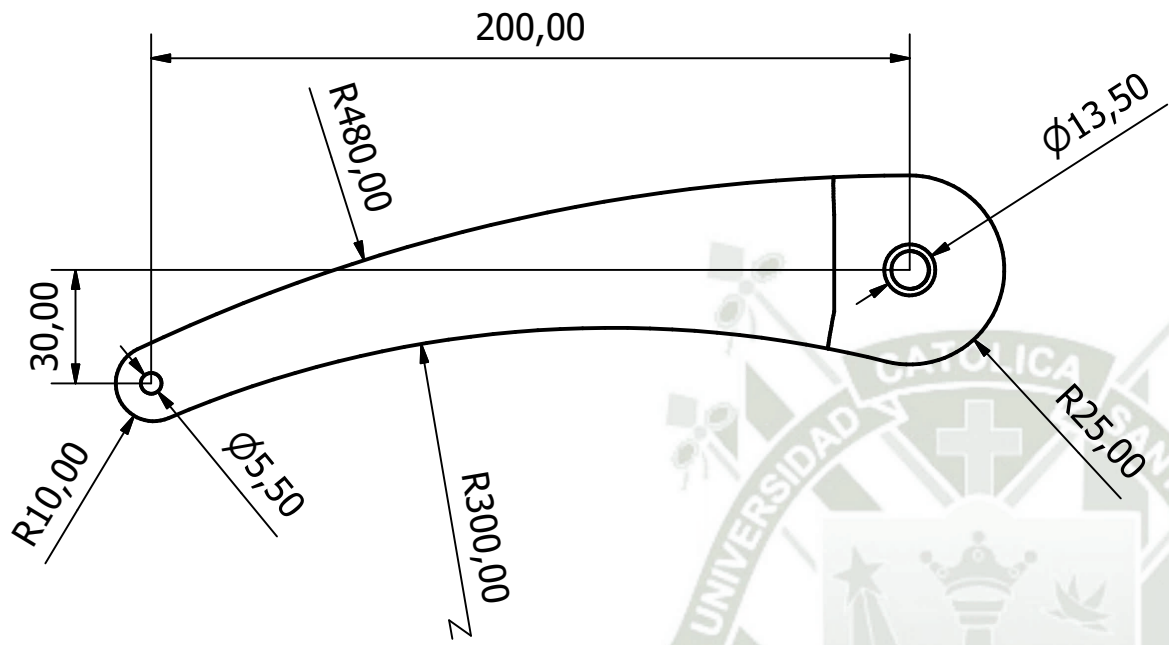
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 10/09/2023	
<b>Universidad Católica de Santa María</b>	<b>ROBOT PARALELO</b>		
	VARILLA ROSCADA	Página: 1/5	Escala: 1 : 2



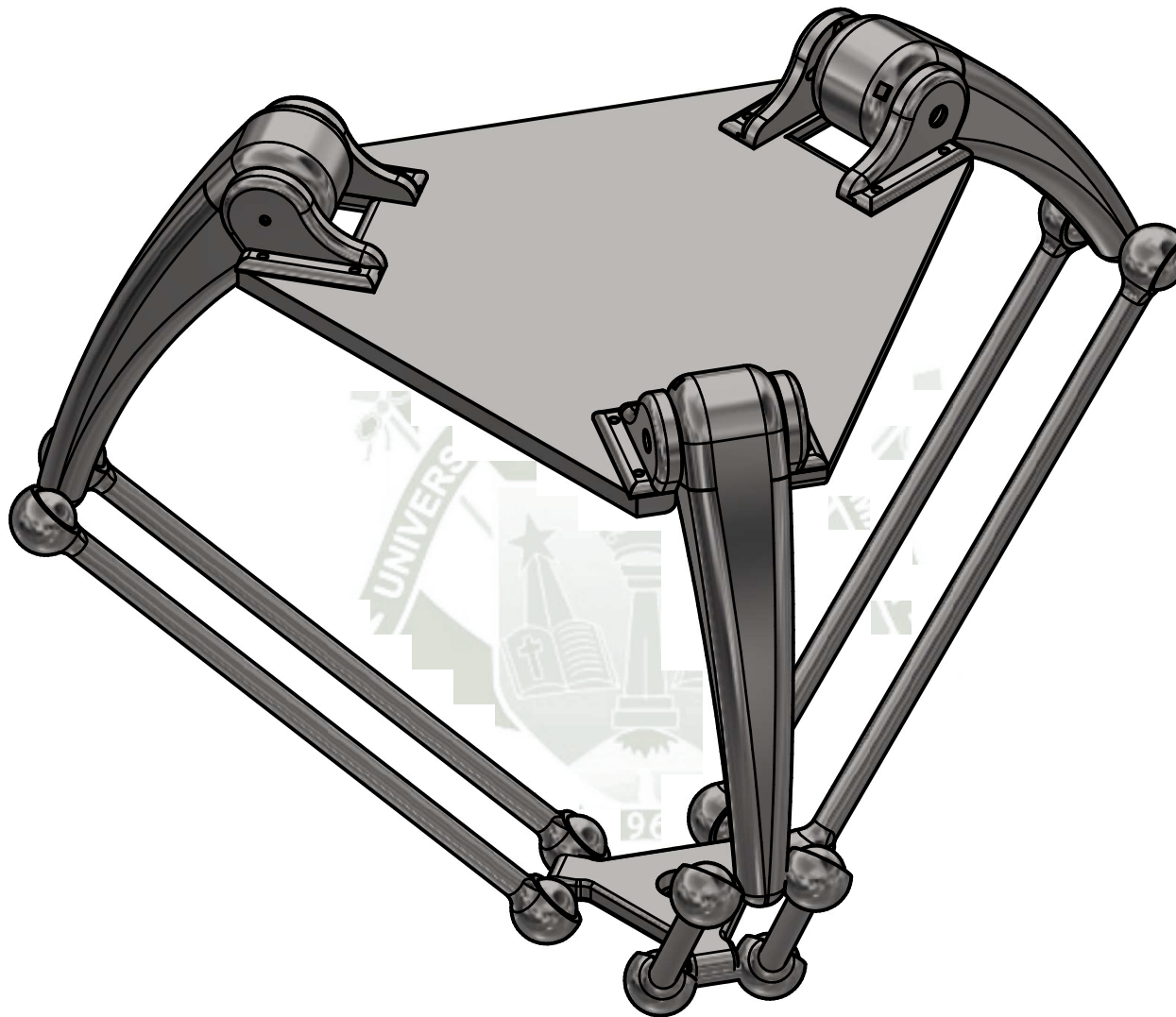
Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 2/09/2023	
<b>Universidad Católica de Santa María</b>	<b>ROBOT PARALELO</b>		
	BASE INFERIOR	Página: 2/5	Escala: 1 : 1



Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 2/09/2023	
 <b>Universidad Católica de Santa María</b>	<b>ROBOT PARALELO</b>		
	<b>BASE SUPERIOR</b>	Página: <b>3/5</b>	Escala: <b>1 : 3</b>



Dibujado por: LLERENA SANCHEZ JULIO CESAR	Revisado por: Ing. Christiam Collado Oporto	Fecha: 2/09/2023	
<b>Universidad Católica de Santa María</b>	<b>ROBOT PARALELO</b>		
	<b>BRAZO</b>	Página: 4/5	Escala: 1 : 2



Dibujado por:

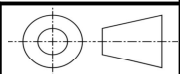
LLERENA SANCHEZ JULIO CESAR

Revisado por:

Ing. Christiam Collado Oporto

Fecha:

3/09/2023



Universidad Católica  
de Santa María

ROBOT PARALELO

ENSAMBLE

Página:  
5/5

Escala:  
1 : 3