



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN MECATRÓNICA**

TEMA:

**“ROBOT MODULAR ENFOCADO A FOMENTAR LA
METODOLOGÍA STEAM EN LA EDUCACIÓN”**



LINEA DE INVESTIGACIÓN: Robótica e inteligencia artificial

AUTOR: Joffre Kevin Túquerres Puma

DIRECTORA: Ing. Luz María Tobar Subía Contenido, MSc.

Ibarra-Ecuador

2025



UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100373937-0		
APELLIDOS Y NOMBRES:	Túquerres Puma Joffre Kevin		
DIRECCIÓN:	Imbabura, Ibarra		
EMAIL:	jktuquerresp@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0998537562

DATOS DE LA OBRA	
TÍTULO:	ROBOT MODULAR ENFOCADO A FOMENTAR LA METODOLOGÍA STEAM EN LA EDUCACIÓN
AUTOR:	Túquerres Puma Joffre Kevin
FECHA:	18/02/2025
SOLO PARA TRABAJOS DE TITULACIÓN	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
ASESOR /DIRECTOR:	Ing. Luz María Tobar Subía Contenido, MSc

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días del mes de febrero de 2025

EL AUTOR:

(Firma).....
Nombre: Joffre Kevin Túquerres Puma



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN
CURRICULAR

Ibarra, 18 de febrero de 2025

Ing. Luz María Tobar Subía Contenido, MSc.

DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Ing. Luz María Tobar Subía Contenido, MSc
C.C.: 1002444204

DEDICATORIA

Con profunda gratitud, dedico este trabajo a mi familia, cuyo apoyo constante ha sido fundamental para alcanzar esta meta. A mis padres, Leonso Túquerres y Maria Puma, les agradezco el invaluable legado de amor, ejemplo y compromiso que me han brindado. Su influencia, así como la de mis hermanos, Oscar, Doris y Jairo, ha moldeado los valores que me definen como persona y profesional.

Además, dedico este logro a mi novia, Anahí Urcuango, quien ha sido mi compañera incondicional a lo largo de todos estos años de estudio. Su aliento constante, comprensión y paciencia, fueron fundamentales para superar ciertas dificultades y alcanzar este triunfo. Gracias, mi amor, por estar siempre a mi lado.

Finalmente, deseo expresar mi gratitud a mi sobrina, Joselyn Molina, cuyo apoyo ha sido fundamental en la elaboración de este proyecto. Su impacto positivo y perspectiva han sido de gran ayuda, por lo que le deseo un futuro lleno de éxitos y alegría.

Joffre Kevin Túquerres Puma

AGRADECIMIENTO

En primer lugar, agradezco profundamente la guía y tutorías brindadas por la MSc. Luz María Tobar Subía Contento, quien con su paciencia y conocimiento me permitió culminar todo el proceso de diseño e investigación que requirió este proyecto de grado. Además, gracias a su apoyo me fue posible conocer a personas influyentes en el área de la educación y creatividad, como fue el caso de la MSc. Mónica Acosta y la MSc. Adriana Aroca quienes me brindaron su tiempo para así poder exponer y validar mi dispositivo robótico dentro de un salón de clase.

Agradezco a mis compañeros de equipo conformado por Marlon Negrete, Edgar Villa y David Villarreal, quienes me han acompañado a lo largo de toda mi vida universitaria. Juntos, hemos navegado por cada semestre, enfrentando desafíos y celebrando triunfos. Agradezco su colaboración, su apoyo y las risas compartidas que hicieron más ligero el camino.

Por último, expreso mi más sincero agradecimiento al Ing. Darío Yépez y al PhD. Xavier Rosero, cuyos conocimientos y experiencias fueron fundamentales para el inicio y desarrollo de este proyecto de innovación en robótica educativa. Su guía y orientación han sido pilares fundamentales en mi formación profesional. Aspiro a poder compartir con ellos, en un futuro, los logros alcanzados gracias a sus enseñanzas.

Joffre Kevin Túquerres Puma

RESUMEN

El limitado uso y conocimiento de herramientas tecnológicas enfocadas al campo de la enseñanza y aprendizaje en centros de educación básica fue el principal motivo para el desarrollo de esta investigación. Con el objetivo de fomentar la metodología de aprendizaje STEAM, se desarrolló un robot modular, el cual establece sus bases teóricas de la investigación documental relacionada con la robótica y al aprendizaje basado en problemas, lo cual genera beneficios tanto a corto como a largo plazo. Con el fin de garantizar que el dispositivo sea eficiente, seguro y atractivo para los niños, se realiza un análisis de criterios, el cual cumple la función de orientar y optimizar la fase de diseño del robot. Cada módulo acoplable al dispositivo principal se encuentra basado en anteriores proyectos relacionados con la robótica educativa, lo que le brinda la capacidad de realizar diversas tareas enfocadas al aprendizaje colaborativo y creativo de los estudiantes. Adicionalmente, el trabajo presenta una propuesta innovadora para el campo de la enseñanza STEAM, debido a que fomenta el desarrollo de habilidades a temprana edad mediante la estimulación de la ciencia y la tecnología dentro de un ambiente en conjunto con el docente encargado.

Palabras clave: STEAM, robótica educativa, aprendizaje, robot modular, tecnología.

ABSTRACT

The limited use and knowledge of technological tools focused on the field of teaching and learning in basic education centers was the main reason for the development of this research. With the aim of promoting the STEAM learning methodology, a modular robot was developed, which establishes its theoretical bases from documentary research related to robotics and problem-based learning, which generates benefits both in the short and long term. In order to guarantee that the device is efficient, safe and attractive for children, a criteria analysis is carried out which fulfills the function of guiding and optimizing the design phase of the robot. Each module attachable to the main device is based on previous projects related to educational robotics, which gives it the ability to perform various tasks focused on the collaborative and creative learning of students. Additionally, the work presents an innovative proposal for the field of STEAM teaching because it allows the development of skills at an early age through the stimulation of science and technology within an environment in conjunction with the teacher in charge.

Keywords: STEAM, educational robotics, learning, modular robot, technology.

ÍNDICE DE CONTENIDOS

IDENTIFICACIÓN DE LA OBRA	II
CONSTANCIAS	II
CERTIFICACIÓN	III
DEDICATORIA	IV
AGRADECIMIENTO	V
RESUMEN	VI
ABSTRACT	VII
ÍNDICE DE CONTENIDOS.....	VIII
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XV
CAPÍTULO I. INTRODUCCIÓN.....	1
1.1 Planteamiento del problema	1
1.2 Objetivo General.....	2
1.3 Objetivos Específicos	2
1.4 Alcance	2
1.5 Justificación	2
CAPÍTULO II. MARCO TEÓRICO.....	4
2.1 Antecedentes.....	4
2.1.1 Internacionales.....	4
2.1.2 Nacionales	5
2.2 Bases Teóricas	7

2.2.1	Principios de la metodología STEAM.....	7
2.2.2	Fundamentos de la metodología STEAM	8
2.2.3	Aprendizaje basado en problemas	9
2.2.4	Futuro de la metodología STEAM	10
2.3	Robótica Educativa.....	10
2.3.1	Robótica educativa en niños	11
2.3.2	Tipos de robots educativos para educación básica.	12
2.3.3	Robótica en los procesos de enseñanza – aprendizaje.....	15
2.4	Robótica modular	16
2.4.1	Componentes de la robótica modular	16
2.4.2	Diseño de robots modulares	17
CAPÍTULO III. MARCO METODOLÓGICO.....		19
3.1	Modelo de investigación.....	19
3.2	Diseño de la investigación.....	19
3.2.1	Fase 1: Adquisición de información	19
3.2.2	Fase 2: Diseño del prototipo.....	20
3.2.3	Fase 3: Pruebas de funcionamiento	22
CAPÍTULO IV. RESULTADOS Y ANÁLISIS.....		23
4.1	Especificaciones del sistema a diseñar	23
4.2	Especificaciones de la solución propuesta.....	23
4.2.1	Placa ESP32 DevKit V1	25
4.2.2	Puente H TB6612FNG	26

4.2.3	Pantalla OLED VGSS I2C	26
4.2.4	Micro motorreductor N20 250:1	27
4.2.5	Buzzer zumbador activo	27
4.2.6	Sensor de línea QTR-1A	28
4.2.7	Transistor 2N2222A NPN.....	28
4.2.8	Baterías de litio y módulo de carga	29
4.2.9	Módulo reductor de voltaje XL4015	29
4.2.10	Diseño de placa PCB	30
4.3	Módulos acoplables	31
4.3.1	Fotorresistencias	32
4.3.2	Sensor ultrasónico	33
4.3.3	Sensor de color	33
4.3.4	Servomotores 180 y 360 grados	34
4.3.5	Diagrama de conexiones.....	34
4.3.6	Cálculos de consumo de corriente	35
4.4	Modalidades y funciones del robot.....	37
4.4.1	Modo seguidor de líneas.....	37
4.4.2	Modo marcador de trazos	37
4.4.3	Modo seguidor de luz	38
4.4.4	Modo evasor de objetos.....	38
4.4.5	Modo lector de colores	39
4.4.6	Modo desplazador de objetos	39

4.5	Programación.....	40
4.5.1	Programación del microcontrolador ESP32	40
4.5.2	Programación del panel de control	41
4.5.3	Diagrama de flujo	43
4.6	Construcción y ensamblaje.....	44
4.6.1	Fabricación de piezas del dispositivo	44
4.6.2	Ensamblaje del robot base	45
4.6.3	Ensamblaje de los módulos acoplables	47
4.7	Análisis de costos	48
4.8	Manual de usuario	50
4.9	Actividades propuestas con el robot modular	50
4.10	Pruebas de campo	55
	CONCLUSIONES.....	61
	RECOMENDACIONES	62
	REFERENCIAS	63
	ANEXOS.....	69
	Anexo A	69
	Anexo B	105
	Anexo C	107
	Anexo D.....	117
	Anexo E	121

ÍNDICE DE FIGURAS

Fig. 2.1 Formulación de un problema para el desarrollo de habilidades	10
Fig. 2.2 Tapete, narración y cartas de movimientos para Bee-Bot	11
Fig. 2.3 Funcionamiento de robot Bee-Bot mediante tablero intuitivo	12
Fig. 2.4 Kit de programación Cubetto para niños	13
Fig. 2.5 Kit Lego Mindstorms	13
Fig. 2.6 MakeBlock mBot robot programable educativo	14
Fig. 2.7 Estructura de Sphero mini y componentes	14
Fig. 2.8 Robot educativo Dash and Dot	15
Fig. 2.9 Frecuencias de uso de los procesos enseñanza - aprendizaje	15
Fig. 2.10 Diagrama de componentes y conexiones de robot publicitario	16
Fig. 4.1 Vista isométrica explosionada de la carcasa del robot en su estado base.	24
Fig. 4.2 Vista isométrica explosionada del robot con sus componentes electrónicos internos.	25
Fig. 4.3 Configuración del controlador dual de motores TB6612FNG	26
Fig. 4.4 Módulo pantalla oled VGSS I2C	26
Fig. 4.5 Micro motorreductor N20 250:1 100RPM 6 V	27
Fig. 4.6 Buzzer o zumbador activo	27
Fig. 4.7 Sensor de línea QTR-1A * 2	28
Fig. 4.8 Configuración de transistor 2N2222A NPN	28
Fig. 4.9 Conexión de baterías de litio a módulo de carga HX-2S-01	29
Fig. 4.10 Módulo reductor de voltaje 5A XL4015	29
Fig. 4.11 Diseño de placa PCB; (a) Esquema eléctrico de conexiones, (b) Visualización 3D de la placa física.....	30
Fig. 4.12 Placa PCB del robot modular.	30

Fig. 4.13 Módulos acoplables al robot; (a) Marcadores de trazos, (b) Seguidor de luz, (c) Evasor de objetos, (d) Detector de color, (e) Grúa.....	31
Fig. 4.14 Esquema de conexiones de para lectura de fotorresistencias.....	33
Fig. 4.15 Módulo sensor ultrasónico HC-SR04	33
Fig. 4.16 Sensor de color TCS3200	33
Fig. 4.17 Servomotores implementados; (a) Tower Pro SG90 180°, (b) Tower Pro MG996R 360°	34
Fig. 4.18 Robot en modalidad de seguidor de líneas.....	37
Fig. 4.19 Robot en modalidad de marcador de trazos.	38
Fig. 4.20 Robot en modalidad de seguidor de luz.	38
Fig. 4.21 Robot en modalidad de evasor de objetos.....	39
Fig. 4.22 Robot en modalidad de lector de colores.	39
Fig. 4.23 Robot en modalidad de desplazador de objetos.	40
Fig. 4.24 Interfaz de programación de Arduino IDE.....	41
Fig. 4.25 Diseño del panel de control del robot.	42
Fig. 4.26 Interfaz de programación de Microsoft Visual Studio.	42
Fig. 4.27 Diagrama de flujo del robot modular.	43
Fig. 4.28 Carcasa impresa del robot base.	44
Fig. 4.29 Tornillo elevador y guías del módulo de grúa.....	45
Fig. 4.30 Ensamblaje de la base inferior del robot.	45
Fig. 4.31 Ensamblaje de la base superior del robot.....	46
Fig. 4.32 Vista interna de conexiones eléctricas del robot.	46
Fig. 4.33 Conexión de módulos acoplables al robot; (a) Seguidor de luz, (b) Evasor de objetos, (c) Detector de color, (d) Grúa.	47

Fig. 4.34 Aplicación del dispositivo robótico enfocado a la enseñanza de figuras geométricas básicas.	55
Fig. 4.35 Aplicación del dispositivo robótico enfocado a la enseñanza de la identificación de cantidades.....	56
Fig. 4.36 Aplicación del dispositivo robótico enfocado al aprendizaje de secuencias con números; (a) Dibujar números, (b) Seguir luz, (c) Seguir líneas.....	58
Fig. 4.37 Aplicación del dispositivo robótico enfocado al reconocimiento de la posición de objetos a través de la lateralidad; (a) Reconocimiento de fichas, (b) Guía del robot.	59

ÍNDICE DE TABLAS

Tabla 4.1 Lista de componentes del dispositivo propuesto.	24
Tabla 4.2 Especificaciones de la placa ESP32 Devkit V1	25
Tabla 4.3 Especificaciones del controlador dual de motores TB6612FNG	26
Tabla 4.4 Especificaciones técnicas del módulo pantalla oled VGSS I2C	27
Tabla 4.5 Lista de componentes de los módulos acoplables.....	32
Tabla 4.6 Especificaciones técnicas de los servomotores implementados	34
Tabla 4.7 Consumo eléctrico del dispositivo.	35
Tabla 4.8 Propiedades del filamento PETG de la marca SUNLU	44
Tabla 4.9 Costos de componentes estándar.....	48
Tabla 4.10 Costos de componentes fabricados por impresión 3D.....	49
Tabla 4.11 Lista de costos total de producción del dispositivo modular.....	50
Tabla 4.12 Matriz de procesos didácticos enfocada al reconocimiento de figuras.....	51
Tabla 4.13 Matriz de procesos didácticos enfocada a la identificación de cantidades. .	52
Tabla 4.14 Matriz de procesos didácticos enfocada a secuencias con números ordinales.	53
Tabla 4.15 Matriz de procesos didácticos enfocada a reconocer la posición de objetos mediante la lateralidad.....	54
Tabla 4.16 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de figuras geométricas.....	55
Tabla 4.17 Resultados de aprendizaje obtenidos por parte de los participantes (primera clase).....	56
Tabla 4.18 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de la identificación de cantidades.....	57

Tabla 4.19 Resultados de aprendizaje obtenidos por parte de los participantes (segunda clase).....	57
Tabla 4.20 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de secuencias con números ordinales.....	58
Tabla 4.21 Resultados de aprendizaje obtenidos por parte de los participantes (tercera clase).....	59
Tabla 4.22 Pruebas de funcionamiento del dispositivo enfocado al reconocimiento de la posición de objetos a través de la lateralidad.	59
Tabla 4.23 Resultados de aprendizaje obtenidos por parte de los participantes (cuarta clase).....	60

CAPÍTULO I. INTRODUCCIÓN

1.1 Planteamiento del problema

La introducción del aprendizaje STEAM en el aula se considera una forma de romper las barreras que existen entre distintas asignaturas, además de ofrecer una enseñanza más profunda y cooperativa entre los estudiantes. Los sectores enfocados a las ciencias, ingeniería, tecnología y las matemáticas son los que generan más empleos y oportunidades de crecimiento económico en la actualidad. Es por ello, que los estudiantes deben esforzarse en desarrollar habilidades enfocadas a la resolución de problemas, creatividad y manejo de nuevas tecnologías, para así poder enfrentar los continuos desafíos presentes en el siglo XXI [1].

La falta de interés por fomentar un sistema de educación basado en el impulso del pensamiento crítico ha generado una disminución constante en países de América Latina en cuanto a temas de innovación tecnológica y emprendimiento entre los años de 2017 hasta 2021[2].

El modelo de educación STEAM se enfoca en desarrollar la creatividad y razonamiento lógico de los estudiantes, impulsando la toma de riesgos los cuales enseñan que todos los errores no deben ser tomados como malos, sino como agentes de ayuda que impulsan su trabajo y aprendizaje, evitando así mantener la falsa ideología de que todas las asignaturas son un conjunto de reglas a memorizar para un examen. Además, este modelo educativo busca crear a los emprendedores del mañana, debido a que alrededor del 65% de los niños que hoy ingresan a la primaria trabajarán en empleos que se encuentran en desarrollo o que aún no existen [3].

La robótica educativa se convierte en una solución valiosa para trabajar en las áreas STEAM, por el hecho de tener un enfoque en varias asignaturas y permitir a los estudiantes aplicar conocimientos teóricos en proyectos prácticos. Sin embargo, el tema

de la robótica no ha sido implementado en su totalidad por parte de una considerable cantidad de escuelas y colegios de América Latina, esto debido a la falta de recursos económicos que estas instituciones poseen y por lo cual es más común encontrar asignaturas y programas relacionados con la robótica en instituciones privadas o de alto prestigio [4].

1.2 Objetivo General

Desarrollar un dispositivo de aprendizaje para fomentar la educación STEAM en centros de educación general básica.

1.3 Objetivos Específicos

- Determinar los parámetros relacionados con robots en un ambiente de educación STEAM.
- Diseñar los mecanismos de movimiento y circuitos del robot empleando herramientas de diseño CAD y CAM.
- Validar el funcionamiento y control del robot modular dentro de un entorno educativo.

1.4 Alcance

La investigación tiene como objetivo el diseño y construcción de un robot modular capaz de desplazarse y realizar diversas acciones establecidas por el usuario, quien, por medio de un tablero didáctico, enviará las instrucciones al mismo. Para el desarrollo de este, se empleará placas “Open Source”, mientras que el tablero constará de fichas de distinto diseño, relacionadas con el razonamiento lógico - matemático.

1.5 Justificación

El diseño de un robot modular enfocado en el modelo educativo STEAM permitirá el desarrollo de las capacidades cognitivas, sensoriales y motrices de los niños/as en centros de educación básica general del cantón Ibarra, tomando énfasis en instituciones

provenientes de sectores rurales, debido a la falta de recursos o materiales didácticos que estos suelen requerir. Es por esta razón, que la implementación de este tipo de dispositivos de innovación representa un aumento significativo en cuanto al nivel y calidad de aprendizaje según menciona el Ministerio de Educación 2021 [5].

Dentro del aspecto económico, la mayoría de los dispositivos robóticos enfocados al área de la educación basada en la metodología STEAM, representan un elevado costo de alrededor de \$750 y \$1200 USD, dependiendo a las acciones y complejidad que estos posean. Además, se resalta su bajo grado de disponibilidad y elevado costo de importación dentro del país. El proporcionar un dispositivo modular evita la necesidad de adquirir un kit completo de robótica, además de brindar la capacidad de agregar o disminuir el nivel de complejidad de programación respecto a la edad o tiempo de aprendizaje del usuario.

Finalmente, el Ministerio de Educación del Ecuador se encuentra implementando una serie de iniciativas para promover la educación basada en la metodología STEAM dentro de escuelas y colegios. Estas iniciativas incluyen la capacitación de docentes y la provisión de recursos, como lo son los robots modulares de costo elevado.

CAPÍTULO II. MARCO TEÓRICO

2.1 Antecedentes

2.1.1 Internacionales

En la Universidad Politécnica de Valencia de España se implementa el kit de robótica LEGO Mindstorms Education EV3 dentro de proyectos STEAM que incorporan herramientas de aprendizaje analíticas, como el aprendizaje cooperativo y la clase inversa. El objetivo de estos proyectos es reforzar el área artística en la educación básica general. Para ello, los estudiantes utilizan los sensores y actuadores del kit de robótica, con el fin de desarrollar sus habilidades creativas y expresivas al momento de diseñar y construir robots artísticos [6].

En la Universidad de Alicante de España se desarrolló un juego de tablero usando robots móviles como fichas interactivas. Estos robots son fáciles de programar, ampliables y flexibles, con la capacidad de detectar las líneas por las que moverse, haciendo uso de sensores y motores, los cuales, gracias a una programación en C y Python, cumplen la función de motivar a los estudiantes a investigar disciplinas como la ingeniería, la física o las matemáticas. Además de buscar el desarrollo del trabajo en equipo o la resolución de problemas aplicables a cualquier sistema complejo de la vida diaria [7].

En la Universidad de la Laguna de España se desarrolló un sistema ludificado enfocado en el uso de la robótica educativa para fomentar el rápido aprendizaje infantil, por medio de actividades basadas en juegos y proyectos, los cuales alimentan tanto el conocimiento como las habilidades motrices de los educados. Este sistema se encuentra compuesto por un robot, tarjetas de programación, una aplicación enfocada a dispositivos móviles y una página web con la cual se busca incentivar la educación desde edades

tempranas haciendo uso de la metodología STEAM como método de educación y adquisición de destrezas [8].

En Uruguay se reconoce el área de la Robótica Educativa como un método eficaz para el desarrollo y enfoque del pensamiento lógico computacional en niños, niñas y adolescentes que se encuentren cruzando la educación básica general. Esto debido a la profunda necesidad de orientar y guiar a la nueva generación de profesionales, destacando así a la programación como una etapa de trabajo en proyectos específicos elegidos por los mismos alumnos. Logrando así cumplir una dinámica educativa secuencial de cuatro palabras: Imaginar, Construir, Diseñar y Programar [9].

En la Universidad de Santander de Colombia se realiza una clase STEAM a estudiantes de séptimo grado de educación. Los alumnos analizan diferentes situaciones por medio de pruebas y simuladores, los cuales, proyectan escenarios con problemas que incentivan su razonamiento lógico matemático. La prueba se basa en diseñar un robot explorador por medio de la metodología STEAM, buscando así fortalecer el interés de los estudiantes en las áreas de Tecnología, Matemáticas, Física y Artística para mantener su creatividad, razonamiento y motivación en la educación a largo plazo [10].

2.1.2 Nacionales

En el Instituto Superior Tecnológico Eloy Alfaro de la ciudad de Esmeraldas se implementó un prototipo de robot móvil para motivar la generación de conocimiento con base en la metodología STEAM. El robot implementa el firmware NodeMCU para realizar su movimiento hacia adelante con giros de derecha, izquierda y alrededor de su eje. Por otro lado, se propuso no aplicar el movimiento hacia atrás para que así el operador llegue a desarrollar aptitudes de orientación espacial, ingenio y razonamiento. Además, se concluye que un gran porcentaje de instituciones de educación general de Esmeraldas,

implementan técnicas de estudio tradicionales para evitar los elevados costos en cuanto a la adquisición de tecnología como medio de enseñanza [11].

La Universidad Técnica del Norte se propuso una metodología enfocada al aprendizaje de la robótica educativa en alumnos de la Unidad Educativa “San Francisco de Asís” de la ciudad Ibarra. Se realizó la construcción de robots con ayuda de componentes electrónicos y partes diseñadas por ellos mediante el uso del computador, todo esto con la finalidad de estimular y desarrollar su pensamiento creativo, además de favorecer un enfoque formal de nuevos conocimientos que llegan a ser claves dentro de la metodología STEAM [12].

En la Universidad de las Fuerzas Armadas de Latacunga se diseñó e implemento un robot social interactivo orientado a la educación STEAM en niños de 8 a 12 años. El proyecto se compone de un robot móvil controlado por medio de una aplicación para dispositivos móviles, la cual consta de un control clásico y por voz, permitiendo así el desplazamiento programado del robot. Esto con el fin de ayudar a los niños a desarrollar de manera más rápida el aprendizaje de materias escolares y la resolución de problemas en diferentes entornos de tipo real [13].

En la Universidad Nacional de Loja se desarrolló el pensamiento lógico computacional de estudiantes que se encuentren cursando el décimo grado de educación general básica en la Unidad Educativa “Santa Mariana de Jesús”. Este proyecto se aplicó mediante el diseño de personajes equipados con placas programables Arduino y servomotores, los cuales permitían la movilidad expresiva del robot humanoide, como brazos, párpados y boca. De esta manera se busca que los alumnos apliquen y desarrollen su imaginación y creatividad, transformando así las materias tradicionales y repetitivas en atractivas e integradoras [14].

En la Universidad Tecnológica Indoamericana de Ambato se aplicó la industria 4.0 como método de enseñanza dentro de la carrera de ingeniería industrial usando un robot que incluye las características principales de dicha área como base para aplicar la metodología STEAM. El robot se basa en un brazo compuesto por partes fabricadas con impresión 3D y una plataforma móvil construida en acero ASTM A36. De esta forma la educación STEAM resalta en la formación profesional de ingeniería y permitiendo así el desarrollo de destrezas ante la solución de problemas que puedan llegar a surgir dentro de industria 4.0 [15].

2.2 Bases Teóricas

2.2.1 Principios de la metodología STEAM.

La metodología STEAM se encuentra enfocada en el desarrollo de las disciplinas de ciencia, tecnología, ingeniería, arte y matemáticas. Este razonamiento fomenta la colaboración entre estudiantes, el pensamiento creativo y la resolución de problemas en un conjunto con sus áreas, las cuales se encuentran representadas por sus siglas en inglés [16].

La metodología STEAM busca que los estudiantes desarrollen un conjunto de habilidades que les permitan abordar y resolver problemas de forma creativa y eficaz. La metodología se basa en la colaboración, el aprendizaje en proyectos y la aplicación de los conocimientos a la vida real.

La metodología STEAM se basa en los siguientes principios [17]:

- **La creatividad y la innovación:** Se busca genera nuevas ideas mientras que la innovación cumple el papel de aplicarlas dentro del área práctica. Ambas habilidades son necesarias para resolver problemas, proporcionando a los estudiantes oportunidades para explorar diferentes disciplinas y perspectivas.

- **El desarrollo de habilidades STEAM:** Las destrezas adquiridas dentro de un entorno de aprendizaje enfocado en la metodología STEAM son fundamentales para el triunfo dentro del siglo XXI. Estos recursos ayudan a los alumnos a prepararse para el éxito y culminación de sus metas y profesiones en la vida.
- **La importancia de las artes:** Las artes ayudan a los estudiantes a comprender conceptos científicos o tecnológicos de una manera más profunda y significativa.
- **La importancia de la colaboración:** La colaboración ayuda a los estudiantes a ver el mundo desde una perspectiva nueva y diferente.

2.2.2 Fundamentos de la metodología STEAM

Los fundamentos de la metodología STEAM se basan en que las disciplinas abarcadas dentro de la ciencia, tecnología e ingeniería están interconectadas. Además, las habilidades y conocimientos que se aprenden en una disciplina son capaces de transferirse a otras sin dificultad alguna [18].

Este enfoque se basa en la creencia de que estas disciplinas están interrelacionadas y que el aprendizaje de una de ellas ayuda a los estudiantes a comprender mejor las demás.

La metodología STEAM se basa en los siguientes fundamentos [19]:

- **Integración:** La metodología STEAM integra las disciplinas para que los estudiantes puedan ver cómo se relacionan entre sí. Esto promueve e incentiva a los estudiantes a comprender y razonar el mundo de una manera más general y eficiente.
- **Aprendizaje activo:** Los estudiantes participan activamente en su propio aprendizaje y entorno, a través de proyectos, experimentos y otras actividades prácticas que refuercen su conocimiento.

- **Problema-Solución:** Los estudiantes tienden a enfrentarse a desafíos y problemas del mundo real, esto con el fin de desarrollar habilidades de razonamiento rápido ante ciertas dificultades, con soluciones de manera creativa e innovadora.
- **Pensamiento crítico:** Se busca desarrollar habilidades de pensamiento crítico, como la capacidad de analizar, evaluar y sintetizar información en el menor tiempo posible.
- **Comunicación:** El estudiante aprende habilidades de comunicación, tanto oral como escrita, para así lograr compartir sus pensamientos e ideas con otros alumnos, ya sea dentro o fuera del aula.
- **Colaboración:** Se busca un entorno colaborativo, enfocado al cumplimiento de objetivos y metas compartidas.

2.2.3 *Aprendizaje basado en problemas*

El aprendizaje basado en problemas (ABP) es considerado como un método de enseñanza en el cual los estudiantes tienden a enfrentarse a un problema o desafío del mundo real. Para resolver el problema, los alumnos deben aplicar sus conocimientos y habilidades en un contexto significativo [20].

El ABP se enfoca en la idea de que los alumnos mejoran su nivel de conocimiento cuando están activamente involucrados en su propio aprendizaje y cuando se enfrentan a desafíos que les retan y motivan a realizar de mejor manera sus actividades. Este tipo de aprendizaje tiene muchos beneficios para los estudiantes, entre ellos [21]:

- **Desarrolla habilidades de resolución de problemas:** El ABP ayuda a los estudiantes a desarrollar habilidades de resolución de problemas, como la capacidad de identificar riesgos, generar soluciones y evaluar resultados mediante la formulación de las preguntas (véase la Fig. 2.1).

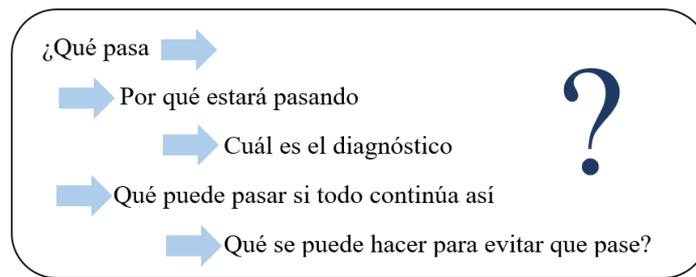


Fig. 2.1 Formulación de un problema para el desarrollo de habilidades [21].

- **Mejora el pensamiento crítico:** El ABP dentro de un ambiente educativo, ayuda a los estudiantes a desarrollar habilidades enfocadas al pensamiento crítico, como es la capacidad de analizar y reconocer información, evaluar argumentos e ideas y llegar a conclusiones basadas en el razonamiento adquirido.
- **Aumenta el interés y la motivación:** Ayuda a aumentar el nivel de interés de los estudiantes, brindando un sano sentido de competencia y satisfacción al cumplir con tareas o asignaciones de preferencia.

2.2.4 Futuro de la metodología STEAM

El futuro de la metodología STEAM es clave dentro de los sistemas educativos a nivel mundial. Cada vez más países y organizaciones se encuentran reconociendo la importancia de esta metodología dentro de la preparación profesional de los estudiantes para su propio éxito en el mundo laboral del siglo XXI.

Dentro de los próximos años, se espera que la metodología STEAM se siga desarrollando y mejorando a una mayor escala, con el fin de integrar aún más disciplinas, como lo son las humanidades. También, se espera que la metodología STEAM sea implementada para abordar problemas de interés social, como lo son cambio climático y la desigualdad [22].

2.3 Robótica Educativa

La robótica educativa es considerada como un sistema de enseñanza interdisciplinar, el cual posibilita que los estudiantes desarrollen habilidades enfocadas a

la enseñanza de la metodología STEAM, por medio de las herramientas científicas que brinda la robótica moderna. Sin embargo, se debe considerar su importancia en ciertas áreas dentro de la lingüística y la creatividad [23].

Durante el periodo de educación superior, los robots poseen un nivel de complejidad considerable, debido a que estos llevan a cabo tareas de programación mucho más avanzadas para sus usuarios.

2.3.1 Robótica educativa en niños

La robótica referente al área de aprendizaje de niños se considera una disciplina que resalta el ámbito educativo. Esta asignación promueve el interés del alumno con la tecnología y la programación, dando así inicio a un aprendizaje dinámico y divertido. Por otra parte, el juego tiende a resaltar dentro del proceso de descubrimiento en el que se entrelazan distintas áreas de estudio: matemáticas, ciencia y programación [24].

La complejidad de la robótica educativa depende de la edad que posee el estudiante. Tomando esto en cuenta, a edades tempranas la enseñanza de la robótica llega a facilitar su aprendizaje, debido a lo llamativo de su funcionamiento e interacción con su operador (véase la Fig. 2.2).

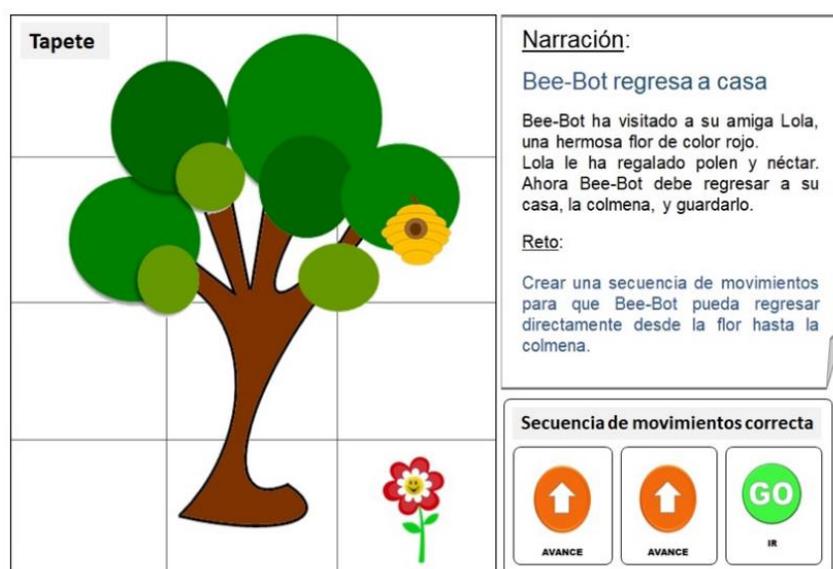


Fig. 2.2 Tapete, narración y cartas de movimientos para Bee-Bot [24].

2.3.2 Tipos de robots educativos para educación básica.

Bee-Bot. Es un robot educativo, adaptado para niños, es clasificado como un dispositivo de interfaz tangible para su usuario debido a que la interacción con el robot se realiza únicamente mediante botones físicos, los cuales se encuentran situados en su parte superior. El movimiento del robot se realiza mediante bloques de programación (véase la Fig. 2.3), los cuales se encuentran disponibles en el tablero que tienen que ver con la trayectoria del robot [25].



Fig. 2.3 Funcionamiento de robot Bee-Bot mediante tablero intuitivo [25].

Cubetto. Es considerado como un dispositivo educativo programable, ya que promueve a los estudiantes el aprendizaje y conocimiento de la programación. Su funcionamiento se basa a partir de la inserción de bloques en diferentes puertos localizados en un tablero de mando (véase la Fig. 2.4). El posicionamiento de los bloques representa acciones específicas que deberá realizar el robot, con ayuda de una placa Arduino y dos motores los cuales realicen giros y desplazamientos [26].



Fig. 2.4 Kit de programación Cubetto para niños [26].

Legó Mindstorms. Es un kit de robótica propuesto por la compañía LEGO, consiste en la construcción de robots con piezas móviles de hasta tres o más ejes de libertad, formando así movimientos programados y autónomos gracias a su módulo controlador (véase la Fig. 2.5), el cual envía las señales de funcionamiento a los demás módulos actuadores y sensores del kit de robótica [27].



Fig. 2.5 Kit Lego Mindstorms [27].

MakeBlock mBot. El robot posee una gran variedad de recursos para aprender robótica educativa y transmitir conocimientos de programación por medio de bloques de manera sencilla e intuitiva. El dispositivo es ideal para niños de 8 a 13 años, debido a que desarrolla la creatividad y mejorar los conocimientos de los niños, presentando un diseño atractivo y moderno (véase la Fig. 2.6) [28].



Fig. 2.6 MakeBlock mBot robot programable educativo [28].

Sphero mini. Dispositivo de tipo compacto, liviano y portable (véase la Fig. 2.7), ideal para personas principiantes en cuanto al aprendizaje de fundamentos básicos de programación basada en bloques. Posee una estructura cuyo interior contiene un giroscopio, un acelerómetro y un conjunto de luces LED de varios colores, las cuales captan órdenes desde un dispositivo móvil, gracias a la aplicación de Sphero Edu donde traza gráficamente las trayectorias de la esfera robótica [29].



Fig. 2.7 Estructura de Sphero mini y componentes [29].

Dash y Dot. Es un dispositivo educativo enfocado a alumnos de a partir de los 6 años. Posee un diseño llamativo y amigable, debido al posicionamiento de un enorme ojo en su módulo central, con el cual le es posible conseguir las primeras nociones de programación (véase la Fig. 2.8). Además, este dispositivo puede llegar a ser usado tanto en casa como en entornos de aprendizaje organizados [30].



Fig. 2.8 Robot educativo Dash and Dot [30].

2.3.3 Robótica en los procesos de enseñanza – aprendizaje

La robótica en los procesos de enseñanza-aprendizaje es considerada como una práctica educativa, la cual utiliza robots enfocados a enseñar a los estudiantes respecto a temas relacionados con la ciencia, tecnología, ingeniería, artes y matemáticas, además de que fomenta la competencia digital de dichos alumnos mediante la estimulación de diversas actividades (véase la Fig. 2.9) [31].



Fig. 2.9 Frecuencias de uso de los procesos enseñanza - aprendizaje [31].

Algunos ejemplos incluyen:

- **Proyectos:** Los estudiantes llegan a poner en práctica el conocimiento adquirido en el aula mediante proyectos, lo cual les brinda una expansión de contenido tanto teórico como metodológico.
- **Actividades:** Las actividades de robótica ayudan a los estudiantes a comprender conceptos básicos de teoría, dando énfasis en temas relevantes del área como el conocimiento de sensores y actuadores, componentes claves dentro de la robótica y su ejecución.

- **Juegos:** Una manera de llamar la atención de los estudiantes a cortas edades se encuentra en la creación de juegos, los cuales fomenten la creatividad de los alumnos por medio de dispositivos robóticos como pilar fundamental de aprendizaje [32].

2.4 Robótica modular

2.4.1 Componentes de la robótica modular

La robótica modular es considerada como un campo de la robótica, el cual se centra en el diseño, programación y la construcción de robots, resaltando la implementación de módulos independientes los unos de los otros para su correcto funcionamiento. Estos módulos llegan a utilizar motores, sensores, controladores o cualquier otro componente electrónico que se necesite para llegar a crear un robot (véase la Fig. 2.10). Los módulos se acoplan entre sí mediante interfaces estandarizadas, desarrollando una amplia gama de robots con diferentes formas y funcionalidades [33].

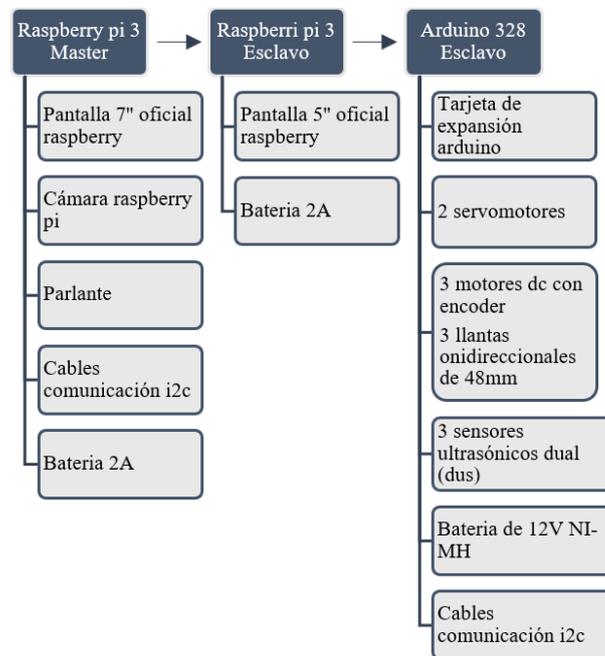


Fig. 2.10 Diagrama de componentes y conexiones de robot publicitario [33].

De entre los componentes que conlleva la robótica modular resaltan:

- **Motores:** Los motores son los componentes cuya funcionalidad se enfoca en el movimiento de casi todo tipo de robots. Además, estos tienden a ser de diferentes tipos, como motores de corriente continua, alterna y motores a pasos.
- **Sensores:** Los robots recopilan información sobre su entorno y estado de funcionamiento a través de los sensores. Además, estos son de diferentes tipos, como sensores de luz, sensores de temperatura, sensores de distancia o sensores de movimiento.
- **Controladores:** Son los componentes que procesan y almacenan la información de los sensores, para así poder enviar una señal de movimiento hacia los actuadores o motores del robot.

Los controladores son de diferentes tipos dependiendo de la complejidad del robot y a las necesidades de su usuario, además de resaltar que la mayoría de estos dispositivos utilizan placas programables como Arduino, Raspberry Pi, ESP32, entre otras.

2.4.2 *Diseño de robots modulares*

El diseño de robots modulares es un proceso complejo debido a su fácil reconfiguración de su estructura. Además, se debe tener en cuenta una serie de factores, como el entorno de aplicación del robot, así como de los componentes disponibles y las restricciones de coste y tiempo que su fabricación conlleva [34].

Es por esto por lo que los pasos básicos para el diseño de robots modulares son los siguientes [35]:

- **Definición de los requisitos:** Se toma en cuenta la aplicación específica que se le dará al robot, incluyendo factores como la forma, el tamaño, la autonomía y el coste de inversión inicial.

- **Selección de componentes:** Dentro de la selección de los componentes se incluyen motores, sensores, controladores y fuentes de alimentación, lo cual permitirá cumplir con las acciones antes planteadas por su usuario.
- **Diseño de la arquitectura:** Implica determinar las conexiones de los diferentes componentes del robot.
- **Diseño de los módulos:** Se diseñan los módulos individuales del robot, respetando los parámetros de la forma, el tamaño y las características de cada módulo por separado.
- **Diseño de la interfaz de usuario:** Por último, se diseña la interfaz de usuario del robot. Esto implica mantener un sistema intuitivo de control y comunicación, ya sea mediante un panel de control o por medio de una aplicación móvil.

CAPÍTULO III. MARCO METODOLÓGICO

3.1 Modelo de investigación

Dentro del desarrollo del presente documento de integración curricular se llega a evidenciar diferentes modelos de investigación implementados, partiendo desde una investigación documental la cual se basa en la consulta de información por medio de recursos de tipo científico como es el caso del impacto y beneficios que conlleva la implementación de la metodología STEAM dentro de un ambiente de enseñanza – aprendizaje.

Además de la información obtenida de manera documental es necesario realizar una investigación de campo para así poder recopilar datos actuales, respecto a la necesidad de adquirir nuevas habilidades de conocimiento y creatividad que brinda la robótica dentro de un ambiente educativo.

Por otro lado, la investigación descriptiva ayuda a la investigación de campo, ya que es la encargada de describir los resultados obtenidos mediante las encuestas realizadas tanto a docentes como a los estudiantes de la unidad educativa.

Por último, la investigación aplicada y experimental se dedican a resolver el problema planteado dentro del trabajo de integración curricular, haciendo uso de la investigación básica y buscando la solución más factible a dicho problema de aprendizaje.

3.2 Diseño de la investigación

Para el desarrollo del presente trabajo de integración curricular, es necesario el cumplir con diferentes actividades, las cuales procuren el correcto orden y validez de los objetivos planteados para el diseño del dispositivo de aprendizaje.

3.2.1 Fase 1: Adquisición de información

Dentro de la primera fase se busca realizar una investigación documental junto con investigación de campo para así poder obtener información con respecto a los

principios y beneficios que conlleva el aplicar la metodología de aprendizaje STEAM hacia estudiantes de educación básica desde edades tempranas.

- **Actividad 1.1: Investigación respecto a la importancia de fomentar el modelo de educación STEAM desde edades tempranas;** esta actividad recopila información acerca de la aplicación de la metodología STEAM en la actualidad. Resalta la representación de las artes como método de expresión creativa dentro del aula y su importancia a largo plazo.
- **Actividad 1.2: Evaluación de dispositivos robóticos de aprendizaje para centros de educación general básica;** se busca identificar una gama de dispositivos robóticos disponibles en el área de la educación, para resaltar sus beneficios dentro del transcurso de su implementación, hasta su crecimiento a futuro.
- **Actividad 1.3: Realización de entrevistas a docentes referentes a la educación STEAM;** se busca adquirir un enfoque profesional respecto a las necesidades de aprendizaje que los estudiantes requieren dentro de las áreas de matemática, ciencias y artes. Además de identificar posibles soluciones mediante la implementación de la robótica dentro del entorno de clases.

3.2.2 Fase 2: Diseño del prototipo

Su objetivo es el diseño y construcción de un prototipo de robot modular enfocado a fomentar la metodología STEAM, partiendo desde una investigación documental que permita conocer los parámetros y necesidades a cubrir por parte del dispositivo.

- **Actividad 2.1: Determinación de los requisitos de diseño del robot modular;** se establecen los componentes y funciones principales del dispositivo base.
- **Actividad 2.2: Diseño conceptual;** se realiza un boceto base de la forma y funcionamiento del dispositivo, tomando en cuenta las dimensiones aproximadas

de cada módulo, además de procurar un sistema intuitivo y fácil de controlar por parte del estudiante y docente.

- **Actividad 2.3: Selección de componentes electrónicos;** una vez generada la estructura base del dispositivo, se adquieren los componentes electrónicos de control y actuadores que activan y accionan el dispositivo modular. Además, este proceso se realiza tomando en cuenta tanto la disponibilidad como el presupuesto antes planteado.
- **Actividad 2.4: Modelado 3D de piezas a utilizar por módulo;** con ayuda de un software de modelado CAD y las medidas reales de los componentes electrónicos antes adquiridos, se modela las estructuras en 3D de cada módulo que utiliza el dispositivo, tomando en cuenta la red de control y energía que se requiere para cada actuador, procurando así su correcto funcionamiento y movilidad.
- **Actividad 2.5: Selección de materiales;** antes de iniciar con la construcción de los módulos, se selecciona el material más adecuado al ambiente en el que el dispositivo se implementara, por lo que este debe de ser duradero y resistente ante caídas y golpes generados por los estudiantes.
- **Actividad 2.6: Impresión en 3D;** mediante la fácil creación y personalización de piezas que brinda la herramienta de impresión 3D, se obtiene la estructura externa junto con los mecanismos móviles del dispositivo modular.
- **Actividad 2.7: Programación del panel controlador;** se busca programar la conectividad entre los módulos de mando y el panel del dispositivo, además de programar los puertos de lectura y accionamiento de cada componente electrónico, ya sea el caso de sensores o actuadores.

- **Actividad 2.8: Ensamblaje del robot;** una vez realizada la programación del dispositivo, se cierra completamente las carcasas de los módulos para comprobar su compatibilidad y funcionamiento.

3.2.3 Fase 3: Pruebas de funcionamiento

Por último, mediante la investigación experimental se valida el funcionamiento y correcto control del dispositivo modular.

- **Actividad 3.1: Ensayos de funcionamiento;** en esta actividad se realiza las últimas modificaciones de programación al robot, esto con el fin de procurar una interacción fluida entre sus distintos módulos y su usuario.
- **Actividad 3.2: Validación del funcionamiento en una institución de educación básica;** se realiza la implementación del dispositivo en un entorno educativo, esto con el fin de analizar su impacto y reacción por parte de los estudiantes y personal docente.
- **Actividad 3.3: Desarrollo del documento (Trabajo de integración curricular);** se redacta el documento final de integración curricular, destacando los resultados obtenidos y la metodología aplicada a lo largo del desarrollo del proyecto.

CAPÍTULO IV. RESULTADOS Y ANÁLISIS

4.1 Especificaciones del sistema a diseñar

A continuación, se presenta una lista detallada de los criterios y requerimientos a tomar en cuenta dentro de la etapa de diseño del robot modular.

- **Intuitividad:** Las conexiones entre la base principal y los módulos acoplables deberán de ser de fácil ensamblaje, sin la necesidad de hacer uso de pernos o herramientas complejas.
- **Robustez:** La estructura exterior debe soportar leves impactos o caídas accidentales durante su manipulación, sin comprometer la integridad de los sistemas internos.
- **Seguridad:** Los componentes del dispositivo que se encuentren en contacto directo con los niños deberán de evitar la presencia de sustancias tóxicas, alergénicas o irritantes.
- **Valor educativo:** Las actividades que realice el dispositivo deben estar enfocadas al desarrollo de destrezas vistas en niños pertenecientes a unidades de educación general básica.
- **Autonomía:** La duración de la batería debe garantizar que el robot sea utilizado durante toda una sesión de clase, sin generar interrupciones por falta de energía o carga.

4.2 Especificaciones de la solución propuesta

Considerando los requerimientos de diseño, a continuación, se describen cada uno de los elementos que conforman al robot base de la solución propuesta. En la Fig. 4.1 se visualiza los componentes que conforman la carcasa del dispositivo, así como una lista de partes en la Tabla 4.1, la cual detalla su cantidad y material de elaboración. Para una

visualización más detallada del dispositivo se adjuntan los planos de piezas y ensamble presentes en el Anexo A.

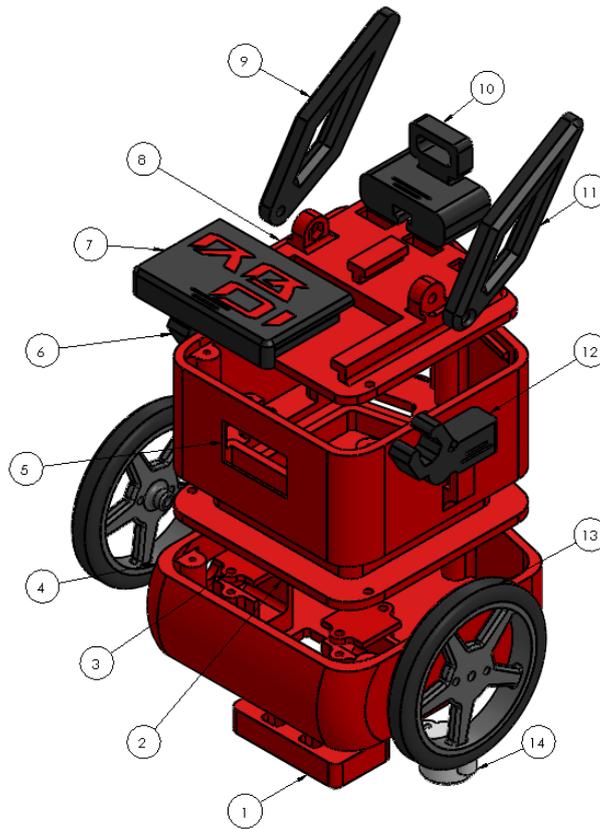


Fig. 4.1 Vista isométrica explosionada de la carcasa del robot en su estado base.

Tabla 4.1 Lista de componentes del dispositivo propuesto.

No.	Descripción	Cantidad	Material
1	Base de sensor de línea	1	PETG
2	Base de separación	1	PETG
3	Tapa motorreductor	2	PETG
4	Chasis base	1	PETG
5	Base superior	1	PETG
6	Brazo derecho	1	PETG
7	Tapa guías frontales	1	PETG
8	Tapa superior	1	PETG
9	Antena derecha	1	PETG
10	Tapa guía posterior	1	PETG
11	Antena izquierda	1	PETG
12	Brazo izquierdo	1	PETG
13	Llanta Pololu 60x8mm	2	-
14	Rueda loca 12 mm	1	-

En la Fig. 4.2 se visualiza el nombre y ubicación de cada componente electrónico que posee el robot.

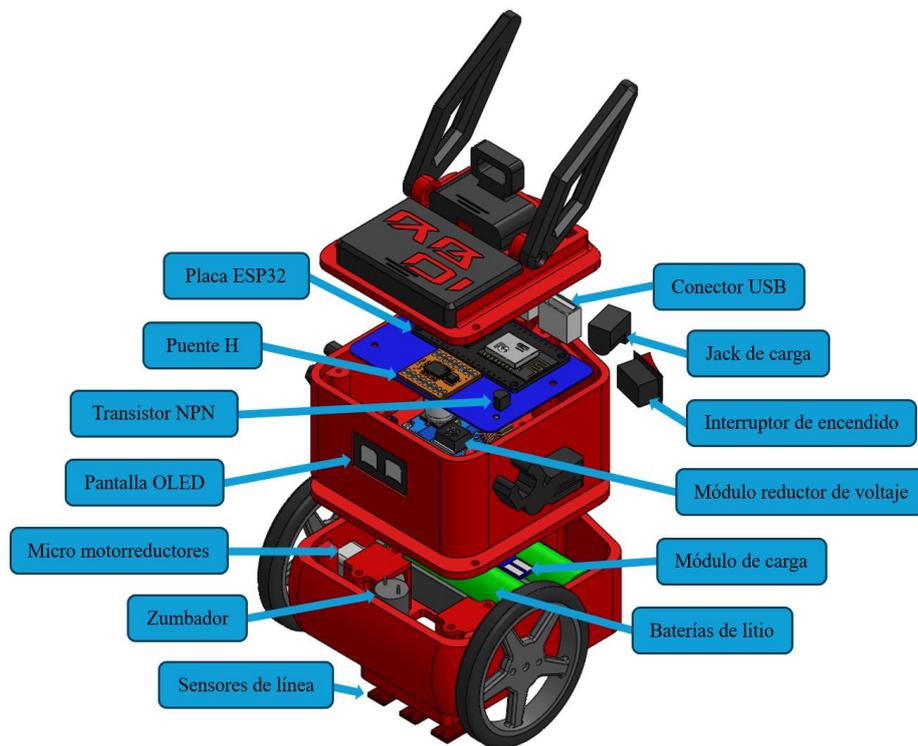


Fig. 4.2 Vista isométrica explotada del robot con sus componentes electrónicos internos.

4.2.1 Placa ESP32 DevKit V1

El microcontrolador ESP32 es el encargado de procesar y activar las diferentes funciones del dispositivo, además de poseer un módulo wifi y Bluetooth, que le facilita conectarse de manera inalámbrica a la aplicación de control del robot. En la Tabla 4.2 se visualiza una lista con algunas de sus características más relevantes.

Tabla 4.2 Especificaciones de la placa ESP32 DevKit V1 [36].

Característica	Descripción
Microcontrolador	ESP32-D0WDQ6
Voltaje de alimentación	3.3 V-5 V
Voltaje de salida regulada	3.3 V
Consumo de corriente	100 mA
Pines de uso general entrada/salida	25 pines GPIO
Memoria Flash	4 MB
SRAM	500KB
Bluetooth	4.0

4.2.2 Puente H TB6612FNG

Para controlar el desplazamiento del robot se ha implementado el módulo controlador TB6612FNG como se muestra en la Fig. 4.3, el cual controla la dirección y velocidad de dos motores DC de forma independiente, esto con ayuda de las salidas PWM de la placa ESP32. En la Tabla 4.3 se visualiza las características más importantes.

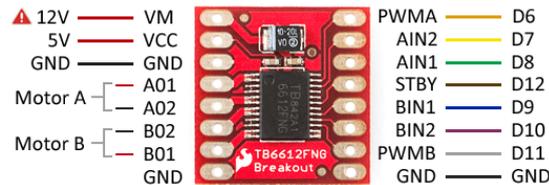


Fig. 4.3 Configuración del controlador dual de motores TB6612FNG [37].

Tabla 4.3 Especificaciones del controlador dual de motores TB6612FNG [37].

Característica	Descripción
Canales	2 (soporta 2 motores DC o 1 motor PAP)
Voltaje de potencia (VM)	5 V – 15 V
Voltaje operación (VCC)	2.7 V – 5.5 V
Corriente de salida continua	1.2 A
Frecuencia máxima de PWM	100 kHz

4.2.3 Pantalla OLED VGSS I2C

Para poder visualizar el cambio de configuración del robot se ha implementado una pantalla oled como se muestra en la Fig. 4.4, la cual por medio de la interfaz I2C le es posible simplificar su conexión a la placa ESP32 con una cantidad mínima de cableado. En la Tabla 4.4 se muestran los datos técnicos de este módulo.



Fig. 4.4 Módulo pantalla oled VGSS I2C [38].

Tabla 4.4 Especificaciones técnicas del módulo pantalla oled VGSS I2C [38].

Característica	Descripción
Driver	SSD1306
Voltaje de operación (VCC)	3.3 V – 5 V
Resolución	128×64 píxeles – 0.96 Pulgadas
Consumo de energía	0.08 W
Colores de pantalla	Azul y amarillo

4.2.4 *Micro motorreductor N20 250:1*

Para desplazar al robot de forma estable se ha utilizado dos micro motorreductores N20 con una relación de transmisión de 250:1 como se muestra en la Fig. 4.5, los cuales mediante una tensión de alimentación de 6 V son capaces de brindar un par motor de 3 kg*cm con un consumo mínimo de corriente de 20 mA [39].



Fig. 4.5 Micro motorreductor N20 250:1 100RPM 6 V [39].

4.2.5 *Buzzer zumbador activo*

El zumbador que se observa en la Fig. 4.6, es utilizado para reproducir diferentes sonidos en diferentes frecuencias programadas, esto con el fin de mejorar la interacción con el usuario. Además, de poseer un bajo consumo energético con un voltaje de funcionamiento de entre 2.7 V a 5.5 V y una corriente máxima de operación de 32 mA [40].



Fig. 4.6 Buzzer o zumbador activo [40].

4.2.6 Sensor de línea QTR-1A

Para que el dispositivo pueda seguir una línea de forma continua, se ha implementado tres sensores de línea QTR-1A como se muestra en la Fig. 4.7, los cuales, mediante un led infrarrojo con un par de fototransistores, son los encargados de enviar el nivel de reflectancia del sensor como un voltaje analógico a un pin de entrada de la placa ESP32. Cada sensor opera con un voltaje de 5 V con un consumo limitado de corriente de aproximadamente 17 mA [41].



Fig. 4.7 Sensor de línea QTR-1A * 2 [41].

4.2.7 Transistor 2N2222A NPN

Para controlar la activación de los seis diodos leds azules que posee el robot se ha implementado un transistor 2N2222A como se muestra en la Fig. 4.8, el cual actúa como un interruptor logrando que fluya una gran corriente a través del colector cuando se detecte el flujo de una pequeña corriente en la base. Este tipo transistor en específico soporta una tensión colector - emisor de hasta 30 V con una corriente máxima de hasta 600 mA [42].

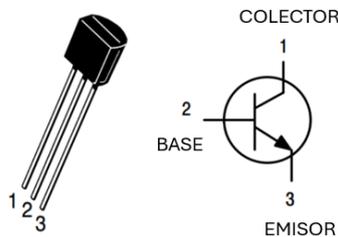


Fig. 4.8 Configuración de transistor 2N2222A NPN [42].

4.2.8 Baterías de litio y módulo de carga

Para poder alimentar a todo el sistema eléctrico principal del robot se ha optado por el uso de dos baterías de litio de 3.7 V a 8.8 Ah recargables, esto con el fin de obtener un valor de voltaje superior a 5 V al mantener una conexión en serie de las baterías, lo que genera un total de 7.4 V, valor suficiente para alimentar tanto sensores como actuadores que posee el robot. Además, se hace uso de un módulo de carga HX-2S-01 para las baterías de litio como se observa en la Fig. 4.9, el cual detiene el flujo de alimentación hacia las mismas en cuanto se ha alcanzado la tensión suficiente de 4.2 V por celda [43].

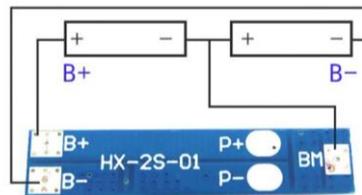


Fig. 4.9 Conexión de baterías de litio a módulo de carga HX-2S-01 [43].

4.2.9 Módulo reductor de voltaje XL4015

Para poder limitar el voltaje de 7.4 V provenientes de la batería del robot, se ha implementado un módulo reductor de voltaje XL4015 como se observa en la Fig. 4.10, el cual cumple la función de regular el voltaje de salida mediante un potenciómetro a 5 V con un flujo de corriente máximo de hasta 5A [44].



Fig. 4.10 Módulo reductor de voltaje 5A XL4015 [44].

4.2.10 Diseño de placa PCB

Para conectar cada componente electrónico que posee el robot internamente, se ha diseñado una placa de circuito impreso con ayuda de la plataforma virtual de Flux IA, la cual genera un esquema de conexiones detallado como se observa en la Fig. 4.11a, además de poder visualizar como se comportara el circuito antes de su fabricación en tiempo real como se visualiza en la Fig. 4.11b.

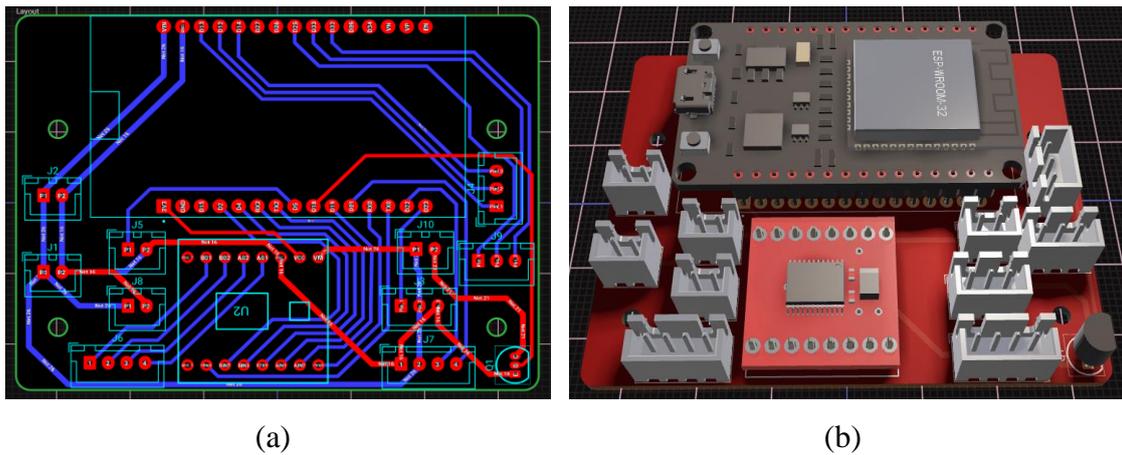


Fig. 4.11 Diseño de placa PCB; (a) Esquema eléctrico de conexiones, (b) Visualización 3D de la placa física.

En la Fig. 4.12 se observa el resultado de la placa PCB obtenida, además de poseer todos sus componentes electrónicos soldados sobre su superficie.

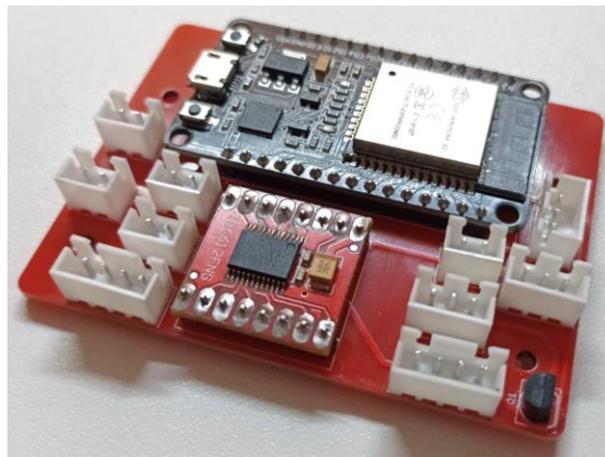


Fig. 4.12 Placa PCB del robot modular.

4.3 Módulos acoplables

Considerando los criterios que deben cumplir los módulos acoplables del robot, a continuación, se describe cada uno de los elementos que estos poseen tanto en su carcasa como en su apartado electrónico. En la Fig. 4.13 se visualiza el diseño de cada módulo generado, así como una lista de componentes presentes en la Tabla 4.5, destacando su cantidad y material de elaboración.

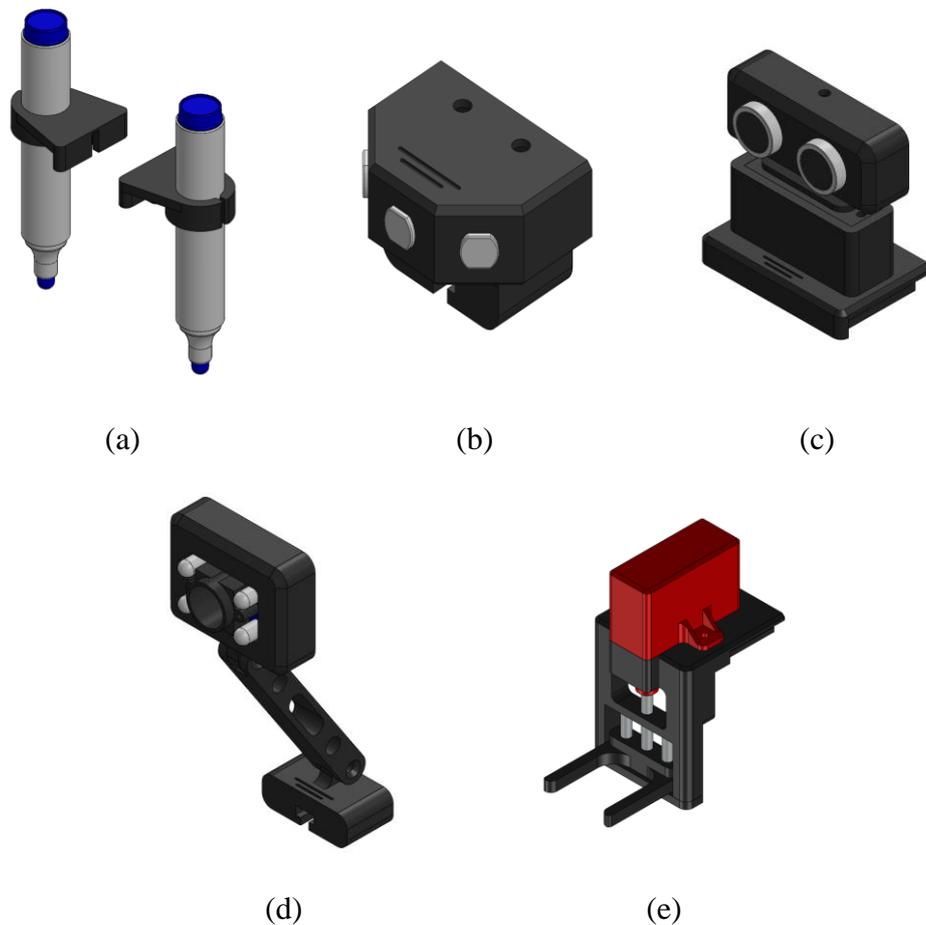


Fig. 4.13 Módulos acoplables al robot; (a) Marcadores de trazos, (b) Seguidor de luz, (c) Evasor de objetos, (d) Detector de color, (e) Grúa.

Tabla 4.5 Lista de componentes de los módulos acoplables.

Módulo	Descripción	Cantidad	Material
A	Base de marcador	2	PETG
	Marcador de tiza líquida	2	-
B	Base de sensor de luz	1	PETG
	Tapa de sensor de luz	1	PETG
	Fotorresistencia	3	-
C	Base rotatoria	1	PETG
	Tapa de base rotatoria	1	PETG
	Carcasa de sensor ultrasónico	1	PETG
	Tapa de carcasa de sensor ultrasónico	1	PETG
	Servomotor 180 grados	1	-
	Sensor ultrasónico	1	-
D	Base de sensor de color	1	PETG
	Soporte de sensor de color	1	PETG
	Carcasa de sensor de color	1	PETG
	Tapa de carcasa de sensor de color	1	PETG
	Sensor de color	1	-
E	Base de grúa	1	PETG
	Tapa servomotor 360 grados	1	PETG
	Acople servomotor 360 grados	1	PETG
	Elevador	1	PETG
	Tapa de grúa	1	PETG
	Guía de grúa	2	Acero
	Tornillo elevador	1	Acero
	Servomotor 360 grados	1	-

4.3.1 Fotorresistencias

Para poder guiar al robot de una manera fluida sin la necesidad de acudir al panel de control del dispositivo, se ha implementado un sistema de seguimiento de luz por medio de tres fotorresistencias. Como se observa en la Fig. 4.14, a medida que la intensidad de luz aumenta, la resistencia LDR disminuye, valor el cual es leído con ayuda de una resistencia de 10 k Ω conectada a un pin análogo de la placa ESP32 [45].

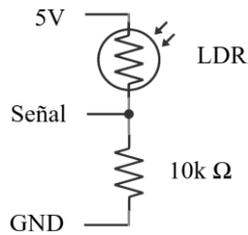


Fig. 4.14 Esquema de conexiones de para lectura de fotorresistencias.

4.3.2 *Sensor ultrasónico*

Para poder evadir objetos, así como resolver laberintos simples se ha implementado un módulo ultrasónico HC-SR04 como se observa en la Fig. 4.15, el cual utiliza ultrasonidos para determinar la distancia en un rango de 2 a 450 cm, con un voltaje de operación de 5 V y una corriente de trabajo de hasta 15 mA [46].



Fig. 4.15 Módulo sensor ultrasónico HC-SR04 [46].

4.3.3 *Sensor de color*

Para poder identificar el color de las fichas interactivas que incluye el robot, se ha implementado un sensor de color TS3200 como se observa en la Fig. 4.16, el cual posee la capacidad de leer hasta 4 colores distintos: rojo, verde, azul y blanco. Para su funcionamiento este sensor requiere de un voltaje de alimentación de 2.7 V a 5.5 V y consumo de corriente de hasta 12 mA [47].



Fig. 4.16 Sensor de color TCS3200 [47].

4.3.4 Servomotores 180 y 360 grados

Para poder rotar al sensor ultrasónico que posee el robot, así como habilitar su mecanismo de grúa, se han implementado dos servomotores como se observa en la Fig. 4.17, los cuales presentan ciertas características diferentes con respecto a su tamaño, torque, y consumo de energía, como se aprecia en la Tabla 4.6.



Fig. 4.17 Servomotores implementados; (a) Tower Pro SG90 180°, (b) Tower Pro MG996R 360° [48].

Tabla 4.6 Especificaciones técnicas de los servomotores implementados [48].

Modelo	Dimensiones	Torque	Peso	Voltaje	Corriente
Tower Pro SG90 180°	22*11.5*27 mm	1.3 kg*cm	9 g	5 V	50 mA
Tower Pro MG996R 360°	40*19*43 mm	13 kg*cm	63 g	5 V	500 mA

4.3.5 Diagrama de conexiones

Una vez descritos los componentes que posee el dispositivo, se realiza el diagrama de conexiones que estos requieren para su correcto funcionamiento, tomando en cuenta la disponibilidad de la fuente de alimentación regulada de 5 V proveniente de las baterías de litio y el módulo reductor de voltaje como se muestra en el Anexo B. Cabe destacar que debido a su estructura modular se ha optado por la implementación de conectores USB, los cuales permitirán la correcta transmisión e intercambio de información entre los módulos acoplables y la placa ESP32.

4.3.6 Cálculos de consumo de corriente

Para poder conocer el tiempo de duración que poseen las baterías acopladas al dispositivo, es necesario calcular el consumo de corriente que llega a requerir cada uno de sus componentes. Si bien los fabricantes suelen especificar la corriente nominal de operación, otros suelen proporcionar la potencia en vatios requerida, por lo que es necesario hacer uso de la ecuación de potencia (Ecuación 4.1).

$$P = I * V \quad \text{(Ecuación 4.1)}$$

Donde:

P: Potencia [Watts].

I: Corriente [Amperios].

V: Voltaje [Voltios].

En la Tabla 4.7 se presentan los valores de corriente calculados por cada elemento electrónico, además de la cantidad de corriente total necesaria para el funcionamiento del dispositivo.

Tabla 4.7 Consumo eléctrico del dispositivo.

Elemento	Corriente [mA]	Voltaje [V]	Cantidad	Corriente Total [mA]
Módulo ESP32 DevKit V1	100	3.3 – 5	1	100
Pantalla oled VGSS 0,96"	16	3.3 – 5	1	16
Buzzer zumbador	32	2.7 – 5.5	1	32
Micro motorreductor 250:1	1200	5 – 6	2	2400
Sensor reflectivo QTR-1RC	17	5	3	51
Led 5 mm normal azul	20	2.4 – 3.7	6	120
Sensor de color TCS3200	12	2.7 – 5.5	1	12
Sensor ultrasónico	15	5	1	15
Servomotor 180 grados	50	4.8 - 6	1	50
Servomotor 360 grados	500	4.8 – 7.2	1	500

Al tratarse de un dispositivo modular, el consumo de energía total se determina tomando en cuenta el módulo acoplado que mayor corriente requiera para su funcionamiento. En este caso, el módulo de grúa el cual se encuentra equipado con un

servomotor de alto torque presenta la demanda energética más significativa. Considerando este factor, el consumo de corriente máximo que llega a requerir el dispositivo es igual a **3.21 mA**, lo que cumple con las características de alimentación provenientes de la fuente de voltaje regulada de 5 V a 5000 mA.

Para determinar con precisión la autonomía del robot, resulta indispensable emplear la Ecuación 4.2. Esta expresión considera tanto la capacidad energética de las baterías como la demanda de corriente del sistema, para así poder calcular de forma precisa el tiempo de funcionamiento del dispositivo.

$$T_{uso} = C_{batería}/C_{dispositivo} \quad (\text{Ecuación 4.2})$$

Donde:

T_{uso} : Tiempo estimado de uso del robot [Horas].

$C_{batería}$: Capacidad energética que almacena la batería [Amperios/Hora].

$C_{dispositivo}$: Consumo de corriente máxima requerida por el dispositivo [Amperios].

Con base a los datos técnicos proporcionados anteriormente, la capacidad de la batería del dispositivo corresponde a 8.800 mAh, con un consumo máximo de corriente de 3.21 mA, lo cual brinda como resultado un tiempo estimado de operación mínimo de **2.73 horas**.

Por último, la Ecuación 4.3 estima el tiempo de carga de las baterías en función de la corriente suministrada por el cargador.

$$T_{carga} = C_{batería}/C_{carga} \quad (\text{Ecuación 4.3})$$

Donde:

T_{carga} : Tiempo de carga de la batería [Horas].

$C_{batería}$: Capacidad energética que almacena la batería [Amperios/Hora].

$C_{dispositivo}$: Capacidad energética que brinda el cargador [Amperios].

Aplicando esta ecuación a los datos proporcionados de las baterías y cargador (8.8 Ah y 2 A), se estima un tiempo de carga de **4.4 horas**.

4.4 Modalidades y funciones del robot

4.4.1 Modo seguidor de líneas

El modo seguidor de líneas, que se muestra en la Fig. 4.18, no requiere la conexión de módulos electrónicos exteriores, esto gracias a la presencia de los tres sensores de línea localizados en la parte inferior del robot, lo que le brinda la capacidad de mantener una ruta estable al momento de desplazarse.



Fig. 4.18 Robot en modalidad de seguidor de líneas.

4.4.2 Modo marcador de trazos

El modo marcador de trazos, que se muestra en la Fig. 4.19, le brinda la capacidad al robot de dibujar su trayectoria al momento de desplazarse con ayuda de ya sea uno o dos marcadores localizados en cada extremo del robot, además de que su control es realizado ya sea mediante el panel de comandos o con ayuda de un nuevo módulo de direccionamiento acoplado en su parte superior.



Fig. 4.19 Robot en modalidad de marcador de trazos.

4.4.3 *Modo seguidor de luz*

Dentro de esta modalidad el desplazamiento del robot es controlado mediante el módulo de sensor de luz acoplado en la parte superior del robot, como se observa en la Fig. 4.20, lo que le posibilita al usuario la capacidad de guiar al dispositivo por diferentes obstáculos con ayuda de una linterna.



Fig. 4.20 Robot en modalidad de seguidor de luz.

4.4.4 *Modo evasor de objetos*

El modo evasor de objetos le brinda la capacidad al robot de movilizarse entre obstáculos haciendo uso del módulo con sensor ultrasónico, como se observa en la Fig. 4.21, el cual mapea y detecta el camino más factible en un rango de 180 grados.



Fig. 4.21 Robot en modalidad de evasor de objetos.

4.4.5 *Modo lector de colores*

Dentro de esta modalidad el robot es capaz de movilizarse realizando diferentes acciones con ayuda del sensor de color ubicado en la parte superior del robot, como se observa en la Fig. 4.22. Para activar al sensor se hace uso de fichas de diferentes colores las cuales darán acceso al usuario a diferentes acciones pregrabadas.



Fig. 4.22 Robot en modalidad de lector de colores.

4.4.6 *Modo desplazador de objetos*

El modo desplazador de objetos le brinda la capacidad al robot de levantar y mover pequeños cubos con ayuda de una grúa localizada en la parte frontal del

dispositivo, como se observa en la Fig. 4.23, este módulo a su vez es manejado mediante la interacción del panel de control y su usuario.

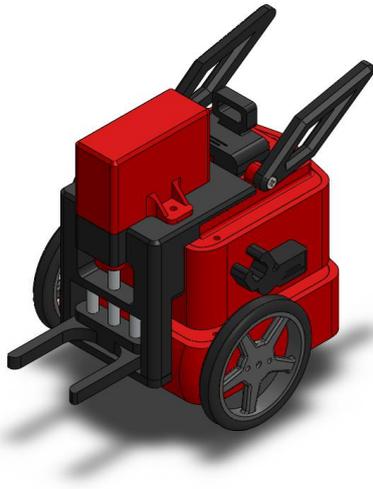


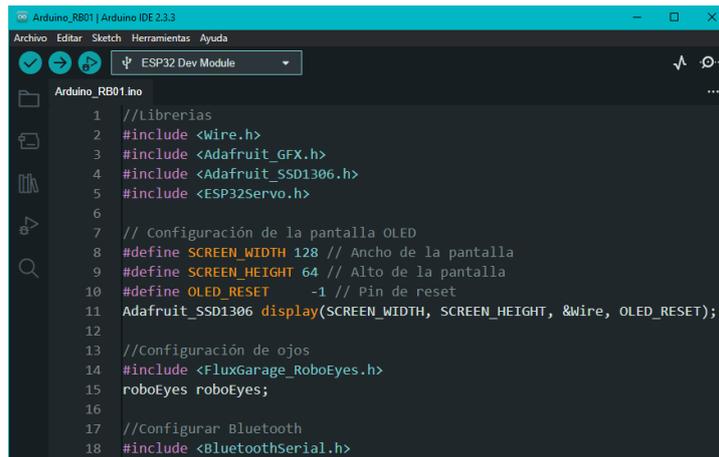
Fig. 4.23 Robot en modalidad de desplazador de objetos.

4.5 Programación

Dentro de la etapa se involucra el desarrollo de un software encargado del control y gestión del robot modular. Este software se encuentra constituido de dos partes principales: la programación del microcontrolador ESP32, que actúa como el cerebro del dispositivo, y la programación del panel de control el cual posibilita la interacción del usuario con el sistema.

4.5.1 Programación del microcontrolador ESP32

Para el desarrollo de la programación del microcontrolador ESP32 correspondiente al Anexo C, se ha empleado el software de Arduino IDE, el cual, mediante el manejo del lenguaje de programación C++, se logra obtener y procesar señales provenientes ya sean del panel de control o de los sensores que posee el dispositivo. En la Fig. 4.24 se observa el entorno de programación empleado.



```
Arduino_RB01.ino
1 //Librerías
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5 #include <ESP32Servo.h>
6
7 // Configuración de la pantalla OLED
8 #define SCREEN_WIDTH 128 // Ancho de la pantalla
9 #define SCREEN_HEIGHT 64 // Alto de la pantalla
10 #define OLED_RESET -1 // Pin de reset
11 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
12
13 //Configuración de ojos
14 #include <FluxGarage_RoboEyes.h>
15 roboEyes roboEyes;
16
17 //Configurar Bluetooth
18 #include <BluetoothSerial.h>
```

Fig. 4.24 Interfaz de programación de Arduino IDE.

La integración de librerías externas en el entorno de Arduino IDE ha permitido adaptar y simplificar el proceso de programación. A continuación, se presentan las librerías implementadas dentro de este entorno:

- Wire.h: Concede la comunicación de la pantalla OLED a través del protocolo I2C.
- Adafruit_GFX.h: Librería básica de gráficas para los controladores de pantallas.
- Adafruit_SSD1306.h: Facilita controlar pantallas OLED de tipo SSD1306.
- ESP32Servo.h: Facilita controlar servomotores usando placas de tipo ESP.
- FluxGarage_RoboEyes.h: Genera y controla la animación de ojos en pantallas.
- BluetoothSerial.h: Admite conectar la placa ESP32 a otros dispositivos bluetooth.

4.5.2 Programación del panel de control

La programación del panel de control del dispositivo correspondiente al Anexo D se desarrolló en el entorno de Microsoft Visual Studio, el cual proporciona un catálogo de herramientas enfocadas al desarrollo de interfaces gráficas, las cuales posibilitan una interacción fácil y sencilla con sus usuarios. Mediante el uso de esta herramienta se generó el diseño base que posee el panel de control del robot, como se observa en la Fig. 4.25.

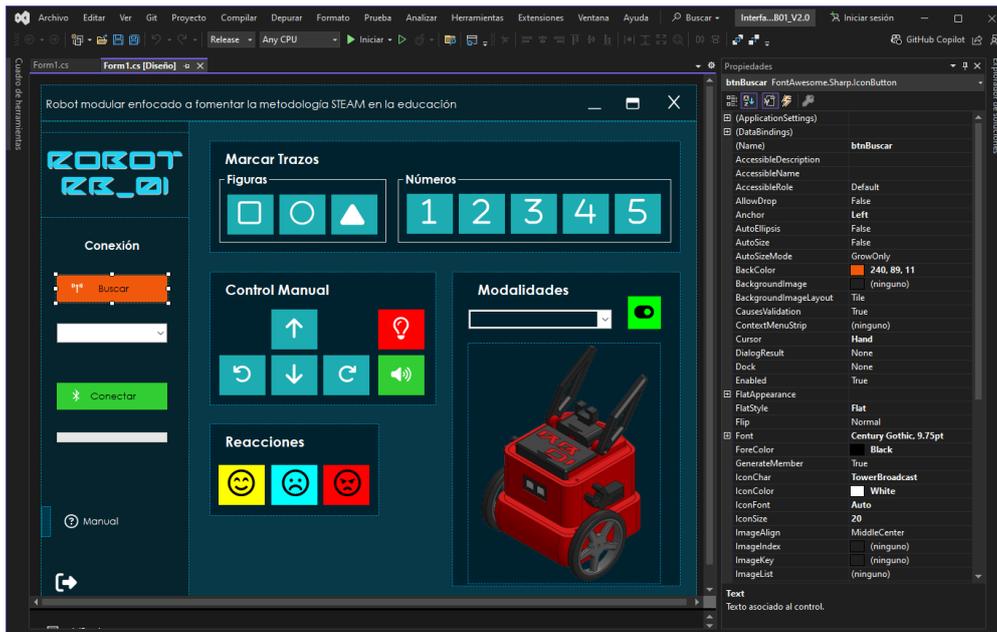


Fig. 4.25 Diseño del panel de control del robot.

Una vez realizado el diseño base del panel de control, se agrega funcionalidad y secuencia a cada botón que posee la interfaz gráfica del usuario, esto mediante lenguaje de programación C# como se observa en la Fig.4.26. Además, cabe resaltar que la comunicación que posee esta aplicación con el robot base se realiza mediante caracteres, los cuales activan bucles de acciones pregrabados en la programación de la placa ESP32, por lo que no es necesario mantener un enlace continuo entre el dispositivo y panel de control.

```

155
156     2 referencias
157     private void btnAdelante_Click(object sender, EventArgs e)
158     {
159         serialPort1.WriteLine("A");
160     }
161     2 referencias
162     private void btnReversa_Click(object sender, EventArgs e)
163     {
164         serialPort1.WriteLine("B");
165     }
166     2 referencias
167     private void btnRotarI_Click(object sender, EventArgs e)
168     {
169         serialPort1.WriteLine("P");
170         Thread.Sleep(300);
171         serialPort1.WriteLine("C");
172         serialPort1.WriteLine("W");
173     }
174     2 referencias
175     private void btnRotarD_Click(object sender, EventArgs e)
176     {
177         serialPort1.WriteLine("D");
178     }
179
180     2 referencias
181     private void btnParar_Click(object sender, EventArgs e)
182     {
183         serialPort1.WriteLine("E");
184     }
185
186     2 referencias
187     private void btnReiniciar_Click(object sender, EventArgs e)
188     {
189         serialPort1.WriteLine("R");
190     }
191
192     2 referencias
193     private void btnAyuda_Click(object sender, EventArgs e)
194     {
195         serialPort1.WriteLine("H");
196     }
197
198     2 referencias
199     private void btnSalir_Click(object sender, EventArgs e)
200     {
201         serialPort1.WriteLine("S");
202     }
203
204     2 referencias
205     private void btnInicio_Click(object sender, EventArgs e)
206     {
207         serialPort1.WriteLine("I");
208     }
209
210     2 referencias
211     private void btnFin_Click(object sender, EventArgs e)
212     {
213         serialPort1.WriteLine("F");
214     }
215
216     2 referencias
217     private void btnAyuda_Click(object sender, EventArgs e)
218     {
219         serialPort1.WriteLine("H");
220     }
221
222     2 referencias
223     private void btnSalir_Click(object sender, EventArgs e)
224     {
225         serialPort1.WriteLine("S");
226     }
227
228     2 referencias
229     private void btnInicio_Click(object sender, EventArgs e)
230     {
231         serialPort1.WriteLine("I");
232     }
233
234     2 referencias
235     private void btnFin_Click(object sender, EventArgs e)
236     {
237         serialPort1.WriteLine("F");
238     }
239
240     2 referencias
241     private void btnAyuda_Click(object sender, EventArgs e)
242     {
243         serialPort1.WriteLine("H");
244     }
245
246     2 referencias
247     private void btnSalir_Click(object sender, EventArgs e)
248     {
249         serialPort1.WriteLine("S");
250     }
251
252     2 referencias
253     private void btnInicio_Click(object sender, EventArgs e)
254     {
255         serialPort1.WriteLine("I");
256     }
257
258     2 referencias
259     private void btnFin_Click(object sender, EventArgs e)
260     {
261         serialPort1.WriteLine("F");
262     }
263
264     2 referencias
265     private void btnAyuda_Click(object sender, EventArgs e)
266     {
267         serialPort1.WriteLine("H");
268     }
269
270     2 referencias
271     private void btnSalir_Click(object sender, EventArgs e)
272     {
273         serialPort1.WriteLine("S");
274     }
275
276     2 referencias
277     private void btnInicio_Click(object sender, EventArgs e)
278     {
279         serialPort1.WriteLine("I");
280     }
281
282     2 referencias
283     private void btnFin_Click(object sender, EventArgs e)
284     {
285         serialPort1.WriteLine("F");
286     }
287
288     2 referencias
289     private void btnAyuda_Click(object sender, EventArgs e)
290     {
291         serialPort1.WriteLine("H");
292     }
293
294     2 referencias
295     private void btnSalir_Click(object sender, EventArgs e)
296     {
297         serialPort1.WriteLine("S");
298     }
299
300     2 referencias
301     private void btnInicio_Click(object sender, EventArgs e)
302     {
303         serialPort1.WriteLine("I");
304     }
305
306     2 referencias
307     private void btnFin_Click(object sender, EventArgs e)
308     {
309         serialPort1.WriteLine("F");
310     }
311
312     2 referencias
313     private void btnAyuda_Click(object sender, EventArgs e)
314     {
315         serialPort1.WriteLine("H");
316     }
317
318     2 referencias
319     private void btnSalir_Click(object sender, EventArgs e)
320     {
321         serialPort1.WriteLine("S");
322     }
323
324     2 referencias
325     private void btnInicio_Click(object sender, EventArgs e)
326     {
327         serialPort1.WriteLine("I");
328     }
329
330     2 referencias
331     private void btnFin_Click(object sender, EventArgs e)
332     {
333         serialPort1.WriteLine("F");
334     }
335
336     2 referencias
337     private void btnAyuda_Click(object sender, EventArgs e)
338     {
339         serialPort1.WriteLine("H");
340     }
341
342     2 referencias
343     private void btnSalir_Click(object sender, EventArgs e)
344     {
345         serialPort1.WriteLine("S");
346     }
347
348     2 referencias
349     private void btnInicio_Click(object sender, EventArgs e)
350     {
351         serialPort1.WriteLine("I");
352     }
353
354     2 referencias
355     private void btnFin_Click(object sender, EventArgs e)
356     {
357         serialPort1.WriteLine("F");
358     }
359
360     2 referencias
361     private void btnAyuda_Click(object sender, EventArgs e)
362     {
363         serialPort1.WriteLine("H");
364     }
365
366     2 referencias
367     private void btnSalir_Click(object sender, EventArgs e)
368     {
369         serialPort1.WriteLine("S");
370     }
371
372     2 referencias
373     private void btnInicio_Click(object sender, EventArgs e)
374     {
375         serialPort1.WriteLine("I");
376     }
377
378     2 referencias
379     private void btnFin_Click(object sender, EventArgs e)
380     {
381         serialPort1.WriteLine("F");
382     }
383
384     2 referencias
385     private void btnAyuda_Click(object sender, EventArgs e)
386     {
387         serialPort1.WriteLine("H");
388     }
389
390     2 referencias
391     private void btnSalir_Click(object sender, EventArgs e)
392     {
393         serialPort1.WriteLine("S");
394     }
395
396     2 referencias
397     private void btnInicio_Click(object sender, EventArgs e)
398     {
399         serialPort1.WriteLine("I");
400     }
401
402     2 referencias
403     private void btnFin_Click(object sender, EventArgs e)
404     {
405         serialPort1.WriteLine("F");
406     }
407
408     2 referencias
409     private void btnAyuda_Click(object sender, EventArgs e)
410     {
411         serialPort1.WriteLine("H");
412     }
413
414     2 referencias
415     private void btnSalir_Click(object sender, EventArgs e)
416     {
417         serialPort1.WriteLine("S");
418     }
419
420     2 referencias
421     private void btnInicio_Click(object sender, EventArgs e)
422     {
423         serialPort1.WriteLine("I");
424     }
425
426     2 referencias
427     private void btnFin_Click(object sender, EventArgs e)
428     {
429         serialPort1.WriteLine("F");
430     }
431
432     2 referencias
433     private void btnAyuda_Click(object sender, EventArgs e)
434     {
435         serialPort1.WriteLine("H");
436     }
437
438     2 referencias
439     private void btnSalir_Click(object sender, EventArgs e)
440     {
441         serialPort1.WriteLine("S");
442     }
443
444     2 referencias
445     private void btnInicio_Click(object sender, EventArgs e)
446     {
447         serialPort1.WriteLine("I");
448     }
449
450     2 referencias
451     private void btnFin_Click(object sender, EventArgs e)
452     {
453         serialPort1.WriteLine("F");
454     }
455
456     2 referencias
457     private void btnAyuda_Click(object sender, EventArgs e)
458     {
459         serialPort1.WriteLine("H");
460     }
461
462     2 referencias
463     private void btnSalir_Click(object sender, EventArgs e)
464     {
465         serialPort1.WriteLine("S");
466     }
467
468     2 referencias
469     private void btnInicio_Click(object sender, EventArgs e)
470     {
471         serialPort1.WriteLine("I");
472     }
473
474     2 referencias
475     private void btnFin_Click(object sender, EventArgs e)
476     {
477         serialPort1.WriteLine("F");
478     }
479
480     2 referencias
481     private void btnAyuda_Click(object sender, EventArgs e)
482     {
483         serialPort1.WriteLine("H");
484     }
485
486     2 referencias
487     private void btnSalir_Click(object sender, EventArgs e)
488     {
489         serialPort1.WriteLine("S");
490     }
491
492     2 referencias
493     private void btnInicio_Click(object sender, EventArgs e)
494     {
495         serialPort1.WriteLine("I");
496     }
497
498     2 referencias
499     private void btnFin_Click(object sender, EventArgs e)
500     {
501         serialPort1.WriteLine("F");
502     }
503
504     2 referencias
505     private void btnAyuda_Click(object sender, EventArgs e)
506     {
507         serialPort1.WriteLine("H");
508     }
509
510     2 referencias
511     private void btnSalir_Click(object sender, EventArgs e)
512     {
513         serialPort1.WriteLine("S");
514     }
515
516     2 referencias
517     private void btnInicio_Click(object sender, EventArgs e)
518     {
519         serialPort1.WriteLine("I");
520     }
521
522     2 referencias
523     private void btnFin_Click(object sender, EventArgs e)
524     {
525         serialPort1.WriteLine("F");
526     }
527
528     2 referencias
529     private void btnAyuda_Click(object sender, EventArgs e)
530     {
531         serialPort1.WriteLine("H");
532     }
533
534     2 referencias
535     private void btnSalir_Click(object sender, EventArgs e)
536     {
537         serialPort1.WriteLine("S");
538     }
539
540     2 referencias
541     private void btnInicio_Click(object sender, EventArgs e)
542     {
543         serialPort1.WriteLine("I");
544     }
545
546     2 referencias
547     private void btnFin_Click(object sender, EventArgs e)
548     {
549         serialPort1.WriteLine("F");
550     }
551
552     2 referencias
553     private void btnAyuda_Click(object sender, EventArgs e)
554     {
555         serialPort1.WriteLine("H");
556     }
557
558     2 referencias
559     private void btnSalir_Click(object sender, EventArgs e)
560     {
561         serialPort1.WriteLine("S");
562     }
563
564     2 referencias
565     private void btnInicio_Click(object sender, EventArgs e)
566     {
567         serialPort1.WriteLine("I");
568     }
569
570     2 referencias
571     private void btnFin_Click(object sender, EventArgs e)
572     {
573         serialPort1.WriteLine("F");
574     }
575
576     2 referencias
577     private void btnAyuda_Click(object sender, EventArgs e)
578     {
579         serialPort1.WriteLine("H");
580     }
581
582     2 referencias
583     private void btnSalir_Click(object sender, EventArgs e)
584     {
585         serialPort1.WriteLine("S");
586     }
587
588     2 referencias
589     private void btnInicio_Click(object sender, EventArgs e)
590     {
591         serialPort1.WriteLine("I");
592     }
593
594     2 referencias
595     private void btnFin_Click(object sender, EventArgs e)
596     {
597         serialPort1.WriteLine("F");
598     }
599
600     2 referencias
601     private void btnAyuda_Click(object sender, EventArgs e)
602     {
603         serialPort1.WriteLine("H");
604     }
605
606     2 referencias
607     private void btnSalir_Click(object sender, EventArgs e)
608     {
609         serialPort1.WriteLine("S");
610     }
611
612     2 referencias
613     private void btnInicio_Click(object sender, EventArgs e)
614     {
615         serialPort1.WriteLine("I");
616     }
617
618     2 referencias
619     private void btnFin_Click(object sender, EventArgs e)
620     {
621         serialPort1.WriteLine("F");
622     }
623
624     2 referencias
625     private void btnAyuda_Click(object sender, EventArgs e)
626     {
627         serialPort1.WriteLine("H");
628     }
629
630     2 referencias
631     private void btnSalir_Click(object sender, EventArgs e)
632     {
633         serialPort1.WriteLine("S");
634     }
635
636     2 referencias
637     private void btnInicio_Click(object sender, EventArgs e)
638     {
639         serialPort1.WriteLine("I");
640     }
641
642     2 referencias
643     private void btnFin_Click(object sender, EventArgs e)
644     {
645         serialPort1.WriteLine("F");
646     }
647
648     2 referencias
649     private void btnAyuda_Click(object sender, EventArgs e)
650     {
651         serialPort1.WriteLine("H");
652     }
653
654     2 referencias
655     private void btnSalir_Click(object sender, EventArgs e)
656     {
657         serialPort1.WriteLine("S");
658     }
659
660     2 referencias
661     private void btnInicio_Click(object sender, EventArgs e)
662     {
663         serialPort1.WriteLine("I");
664     }
665
666     2 referencias
667     private void btnFin_Click(object sender, EventArgs e)
668     {
669         serialPort1.WriteLine("F");
670     }
671
672     2 referencias
673     private void btnAyuda_Click(object sender, EventArgs e)
674     {
675         serialPort1.WriteLine("H");
676     }
677
678     2 referencias
679     private void btnSalir_Click(object sender, EventArgs e)
680     {
681         serialPort1.WriteLine("S");
682     }
683
684     2 referencias
685     private void btnInicio_Click(object sender, EventArgs e)
686     {
687         serialPort1.WriteLine("I");
688     }
689
690     2 referencias
691     private void btnFin_Click(object sender, EventArgs e)
692     {
693         serialPort1.WriteLine("F");
694     }
695
696     2 referencias
697     private void btnAyuda_Click(object sender, EventArgs e)
698     {
699         serialPort1.WriteLine("H");
700     }
701
702     2 referencias
703     private void btnSalir_Click(object sender, EventArgs e)
704     {
705         serialPort1.WriteLine("S");
706     }
707
708     2 referencias
709     private void btnInicio_Click(object sender, EventArgs e)
710     {
711         serialPort1.WriteLine("I");
712     }
713
714     2 referencias
715     private void btnFin_Click(object sender, EventArgs e)
716     {
717         serialPort1.WriteLine("F");
718     }
719
720     2 referencias
721     private void btnAyuda_Click(object sender, EventArgs e)
722     {
723         serialPort1.WriteLine("H");
724     }
725
726     2 referencias
727     private void btnSalir_Click(object sender, EventArgs e)
728     {
729         serialPort1.WriteLine("S");
730     }
731
732     2 referencias
733     private void btnInicio_Click(object sender, EventArgs e)
734     {
735         serialPort1.WriteLine("I");
736     }
737
738     2 referencias
739     private void btnFin_Click(object sender, EventArgs e)
740     {
741         serialPort1.WriteLine("F");
742     }
743
744     2 referencias
745     private void btnAyuda_Click(object sender, EventArgs e)
746     {
747         serialPort1.WriteLine("H");
748     }
749
750     2 referencias
751     private void btnSalir_Click(object sender, EventArgs e)
752     {
753         serialPort1.WriteLine("S");
754     }
755
756     2 referencias
757     private void btnInicio_Click(object sender, EventArgs e)
758     {
759         serialPort1.WriteLine("I");
760     }
761
762     2 referencias
763     private void btnFin_Click(object sender, EventArgs e)
764     {
765         serialPort1.WriteLine("F");
766     }
767
768     2 referencias
769     private void btnAyuda_Click(object sender, EventArgs e)
770     {
771         serialPort1.WriteLine("H");
772     }
773
774     2 referencias
775     private void btnSalir_Click(object sender, EventArgs e)
776     {
777         serialPort1.WriteLine("S");
778     }
779
780     2 referencias
781     private void btnInicio_Click(object sender, EventArgs e)
782     {
783         serialPort1.WriteLine("I");
784     }
785
786     2 referencias
787     private void btnFin_Click(object sender, EventArgs e)
788     {
789         serialPort1.WriteLine("F");
790     }
791
792     2 referencias
793     private void btnAyuda_Click(object sender, EventArgs e)
794     {
795         serialPort1.WriteLine("H");
796     }
797
798     2 referencias
799     private void btnSalir_Click(object sender, EventArgs e)
800     {
801         serialPort1.WriteLine("S");
802     }
803
804     2 referencias
805     private void btnInicio_Click(object sender, EventArgs e)
806     {
807         serialPort1.WriteLine("I");
808     }
809
810     2 referencias
811     private void btnFin_Click(object sender, EventArgs e)
812     {
813         serialPort1.WriteLine("F");
814     }
815
816     2 referencias
817     private void btnAyuda_Click(object sender, EventArgs e)
818     {
819         serialPort1.WriteLine("H");
820     }
821
822     2 referencias
823     private void btnSalir_Click(object sender, EventArgs e)
824     {
825         serialPort1.WriteLine("S");
826     }
827
828     2 referencias
829     private void btnInicio_Click(object sender, EventArgs e)
830     {
831         serialPort1.WriteLine("I");
832     }
833
834     2 referencias
835     private void btnFin_Click(object sender, EventArgs e)
836     {
837         serialPort1.WriteLine("F");
838     }
839
840     2 referencias
841     private void btnAyuda_Click(object sender, EventArgs e)
842     {
843         serialPort1.WriteLine("H");
844     }
845
846     2 referencias
847     private void btnSalir_Click(object sender, EventArgs e)
848     {
849         serialPort1.WriteLine("S");
850     }
851
852     2 referencias
853     private void btnInicio_Click(object sender, EventArgs e)
854     {
855         serialPort1.WriteLine("I");
856     }
857
858     2 referencias
859     private void btnFin_Click(object sender, EventArgs e)
860     {
861         serialPort1.WriteLine("F");
862     }
863
864     2 referencias
865     private void btnAyuda_Click(object sender, EventArgs e)
866     {
867         serialPort1.WriteLine("H");
868     }
869
870     2 referencias
871     private void btnSalir_Click(object sender, EventArgs e)
872     {
873         serialPort1.WriteLine("S");
874     }
875
876     2 referencias
877     private void btnInicio_Click(object sender, EventArgs e)
878     {
879         serialPort1.WriteLine("I");
880     }
881
882     2 referencias
883     private void btnFin_Click(object sender, EventArgs e)
884     {
885         serialPort1.WriteLine("F");
886     }
887
888     2 referencias
889     private void btnAyuda_Click(object sender, EventArgs e)
890     {
891         serialPort1.WriteLine("H");
892     }
893
894     2 referencias
895     private void btnSalir_Click(object sender, EventArgs e)
896     {
897         serialPort1.WriteLine("S");
898     }
899
900     2 referencias
901     private void btnInicio_Click(object sender, EventArgs e)
902     {
903         serialPort1.WriteLine("I");
904     }
905
906     2 referencias
907     private void btnFin_Click(object sender, EventArgs e)
908     {
909         serialPort1.WriteLine("F");
910     }
911
912     2 referencias
913     private void btnAyuda_Click(object sender, EventArgs e)
914     {
915         serialPort1.WriteLine("H");
916     }
917
918     2 referencias
919     private void btnSalir_Click(object sender, EventArgs e)
920     {
921         serialPort1.WriteLine("S");
922     }
923
924     2 referencias
925     private void btnInicio_Click(object sender, EventArgs e)
926     {
927         serialPort1.WriteLine("I");
928     }
929
930     2 referencias
931     private void btnFin_Click(object sender, EventArgs e)
932     {
933         serialPort1.WriteLine("F");
934     }
935
936     2 referencias
937     private void btnAyuda_Click(object sender, EventArgs e)
938     {
939         serialPort1.WriteLine("H");
940     }
941
942     2 referencias
943     private void btnSalir_Click(object sender, EventArgs e)
944     {
945         serialPort1.WriteLine("S");
946     }
947
948     2 referencias
949     private void btnInicio_Click(object sender, EventArgs e)
950     {
951         serialPort1.WriteLine("I");
952     }
953
954     2 referencias
955     private void btnFin_Click(object sender, EventArgs e)
956     {
957         serialPort1.WriteLine("F");
958     }
959
960     2 referencias
961     private void btnAyuda_Click(object sender, EventArgs e)
962     {
963         serialPort1.WriteLine("H");
964     }
965
966     2 referencias
967     private void btnSalir_Click(object sender, EventArgs e)
968     {
969         serialPort1.WriteLine("S");
970     }
971
972     2 referencias
973     private void btnInicio_Click(object sender, EventArgs e)
974     {
975         serialPort1.WriteLine("I");
976     }
977
978     2 referencias
979     private void btnFin_Click(object sender, EventArgs e)
980     {
981         serialPort1.WriteLine("F");
982     }
983
984     2 referencias
985     private void btnAyuda_Click(object sender, EventArgs e)
986     {
987         serialPort1.WriteLine("H");
988     }
989
990     2 referencias
991     private void btnSalir_Click(object sender, EventArgs e)
992     {
993         serialPort1.WriteLine("S");
994     }
995
996     2 referencias
997     private void btnInicio_Click(object sender, EventArgs e)
998     {
999         serialPort1.WriteLine("I");
1000    }

```

Fig. 4.26 Interfaz de programación de Microsoft Visual Studio.

4.5.3 Diagrama de flujo

Una vez establecidos los parámetros de programación, en la Fig. 4.27 se muestra el diagrama de flujo el cual representa las funciones y secuencias lógicas que llega a realizar el robot, dando un énfasis en cada una de sus configuraciones accesibles por su usuario.

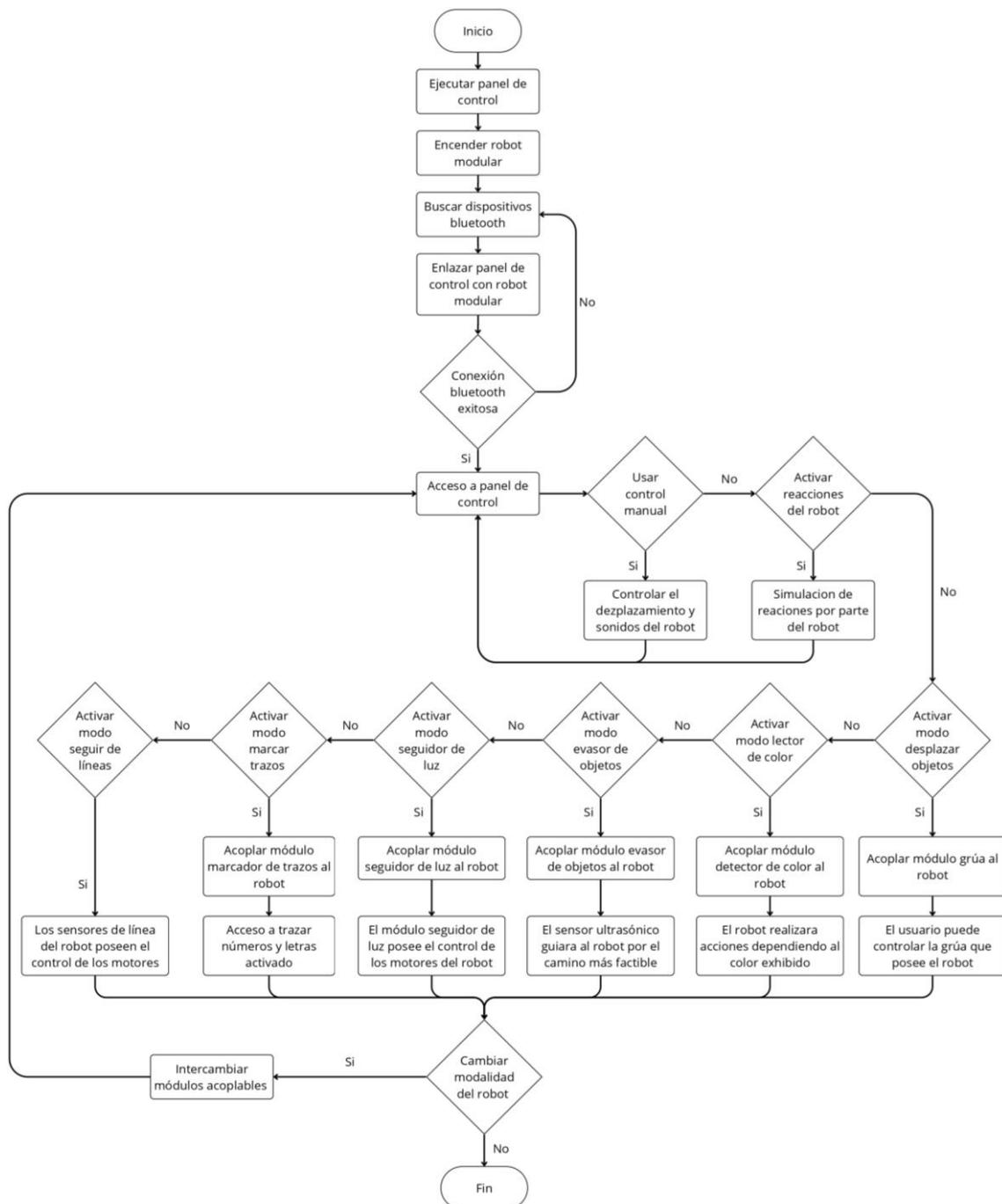


Fig. 4.27 Diagrama de flujo del robot modular.

4.6 Construcción y ensamblaje

Dentro de la etapa de construcción del dispositivo se ha optado por el método de fabricación de piezas por impresión 3D, esto debido a que posibilita el acceso de forma rápida a geometrías complejas que llegan a ser difíciles de obtener por otros métodos. El filamento que conforma la estructura del robot corresponde a PETG de la marca SUNLU, el cual ha sido seleccionado debido a su amplia disponibilidad y sus características mecánicas de impresión, tal como se evidencia en la Tabla 4.8.

Tabla 4.8 Propiedades del filamento PETG de la marca SUNLU [49].

Propiedad	Valor Típico	Unidades	Observación
Densidad	1.27	g/cm ³	Relación peso – tamaño del material.
Módulo de Young	2.2	GPa	Rigidez del material.
Alargamiento a la rotura	30	%	Ductilidad del material.
Resistencia a la tracción	55-65	MPa	Fuerza de ruptura del material.
Temperatura de deformación	85	°C	Flexibilidad del material al calor.
Temperatura de extrusión	230	°C	Temperatura de boquilla de impresión.
Temperatura de cama	65	°C	Temperatura de adherencia de la cama.
Velocidad de impresión	60	mm/s	Velocidad recomendada de impresión.

4.6.1 Fabricación de piezas del dispositivo

Mediante el método de impresión 3D se obtuvieron las piezas que conformaran la estructura base y módulos acoplables del robot como se observa en la Fig. 4.28. Además, para garantizar la resistencia ante impactos y caídas, la carcasa presenta un espesor de pared de entre 2 a 3 mm con una configuración de relleno considerable del 80%.

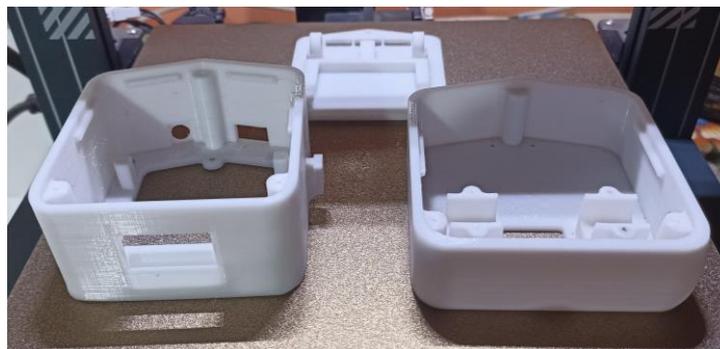


Fig. 4.28 Carcasa impresa del robot base.

Adicionalmente, el módulo de grúa requiere la adaptación de piezas metálicas las cuales cumplen la función de transmitir y guiar la fuerza que esta ejerce sobre los objetos para levantarlos, por lo que es necesario el uso de un tornillo elevador junto con dos guías como se observa en la Fig. 4.29.



Fig. 4.29 Tornillo elevador y guías del módulo de grúa.

4.6.2 Ensamblaje del robot base

Una vez adquiridas las piezas estructurales, así como electrónicas, se inicia con el ensamblaje de la base inferior del robot. Durante esta etapa se toma en cuenta la distribución del cableado además del recubrimiento de conexiones expuestas las cuales llegan a requerir del uso de cinta aislante o materia termo contraíble como se observa en la Fig. 4.30.



Fig. 4.30 Ensamblaje de la base inferior del robot.

Posteriormente, se ensambla la parte superior del robot, insertando y conectando el módulo regulador de voltaje junto con la placa PCB, como se observa en la Fig. 4.31.

Para una visualización más detallada se recomienda visitar el plano de ensamblaje del robot presente en el Anexo A.

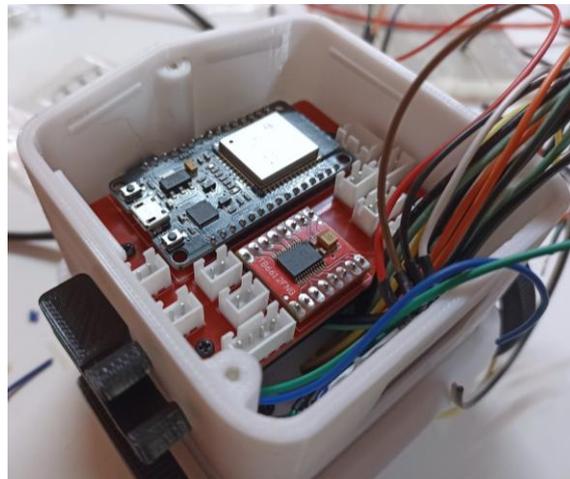


Fig. 4.31 Ensamblaje de la base superior del robot.

Por último, se realiza las conexiones eléctricas del cableado inferior junto con el interruptor de encendido del circuito, además de adaptar las entradas de los puertos USB los cuales permitirán la comunicación entre los módulos acoplables y el robot base, como se observa en la Fig. 4.32. Todo este proceso es realizado procurando mantener el esquema de conexiones presente en el Anexo B.

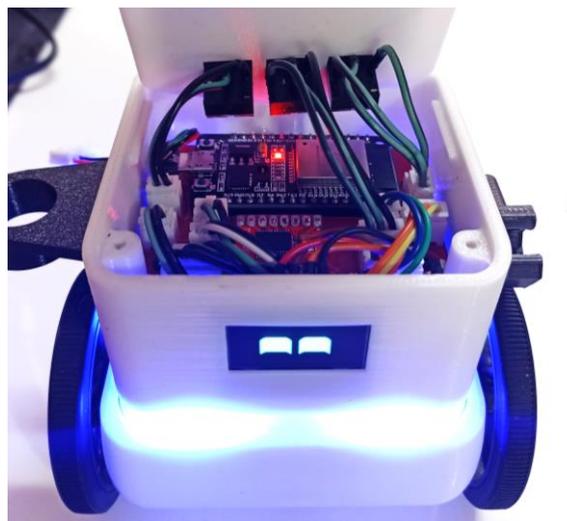


Fig. 4.32 Vista interna de conexiones eléctricas del robot.

4.6.3 Ensamblaje de los módulos acoplables

Para garantizar un correcto ensamblaje de los módulos acoplables, se debe seguir detalladamente los planos de ensamblaje individual incluidos en el Anexo A. Además, se debe tomar en cuenta la integración de los cables USB los cuales mantendrán una conexión estable entre cada módulo y la placa PCB como se observan en la Fig. 4.33.

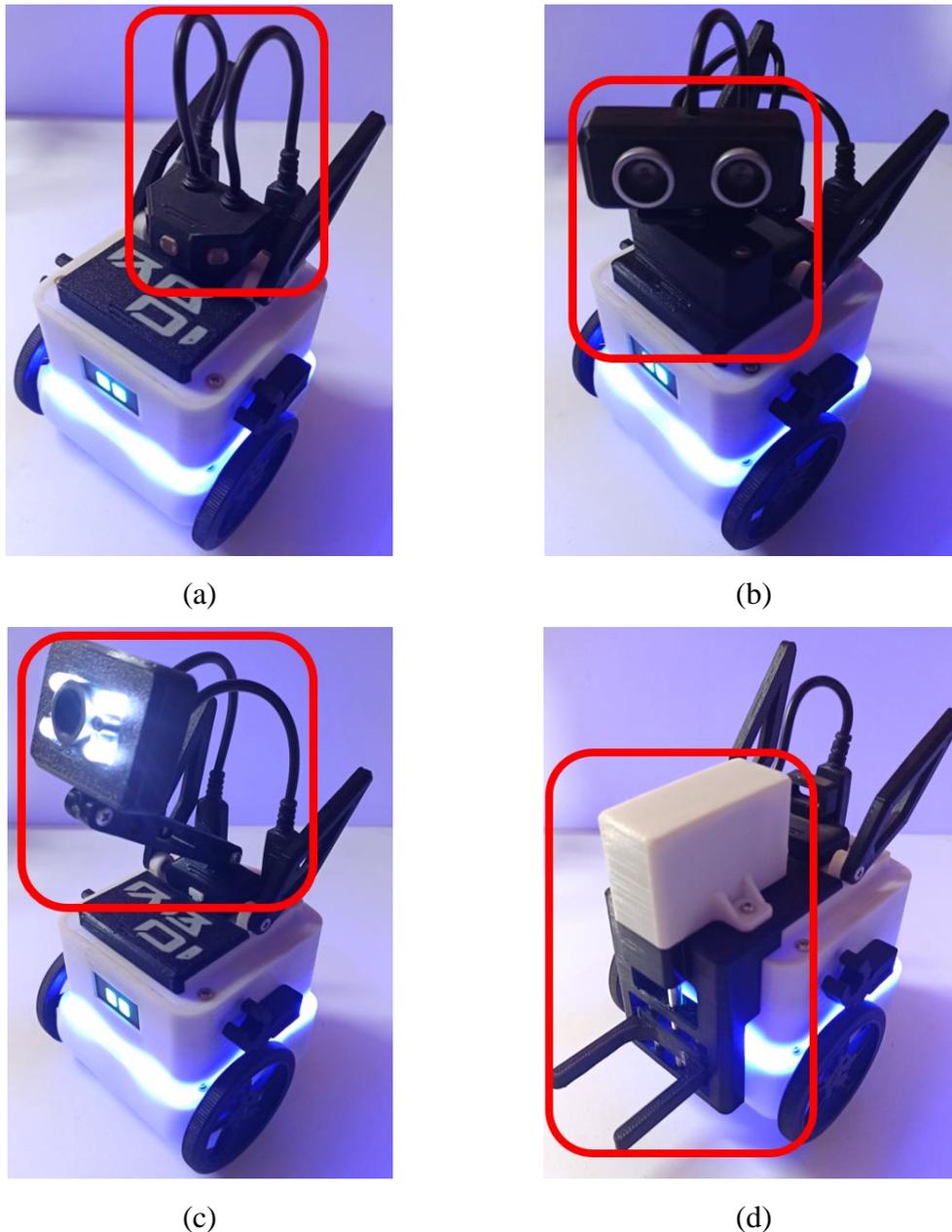


Fig. 4.33 Conexión de módulos acoplables al robot; (a) Seguidor de luz, (b) Evasor de objetos, (c) Detector de color, (d) Grúa.

4.7 Análisis de costos

El costo de producción del dispositivo base junto con sus módulos acoplables se desglosa en dos categorías principales: la adquisición de componentes estándar, detallada en la Tabla 4.9, y la fabricación de piezas elaboradas por impresión 3D, cuyos costos se ven reflejados en la Tabla 4.10.

Tabla 4.9 Costos de componentes estándar.

Componente	Cantidad	Costo Unitario	Costo Total
Módulo ESP32 DevKit V1	1	\$14,99	\$14,99
Puente H TB6612FNG	1	\$5,99	\$5,99
Transistor NPN 2n2222a	1	\$0,08	\$0,08
Pantalla oled VGSS 0,96"	1	\$6,99	\$6,99
Buzzer zumbador	1	\$0,80	\$0,80
Micro motorreductor 250:1 100rpm	2	\$7,99	\$15,98
Sensor reflectivo QTR-1RC	3	\$3,75	\$11,25
Jack DC005 voltaje	1	\$0,20	\$0,20
Módulo reductor de voltaje XL4015	1	\$3,99	\$3,99
Cargador 7.4 V 2A	1	\$6,00	\$6,00
Led 5 mm normal azul	6	\$0,05	\$0,30
Porta pilas 18650	1	\$1,24	\$1,24
Rueda loca metálica 12 mm	1	\$1,99	\$1,99
Par de llantas Pololu 60×8 mm	1	\$7,99	\$7,99
Conector hembra USB	3	\$0,39	\$1,17
Módulo de carga 18650 2 CE	1	\$2,52	\$2,52
Switch mini 2 posiciones	1	\$0,35	\$0,35
Batería de litio 18650 3.7V 8800mAh	2	\$6,99	\$13,98
Placa PCB	1	\$7,00	\$7,00
Sensor de color TCS3200	1	\$12,00	\$12,00
Sensor ultrasónico	1	\$2,00	\$2,00
Marcador de tiza líquida	2	\$0,75	\$1,50
Cable macho USB a USB	4	\$1,00	\$4,00
Servomotor 180 grados	1	\$4,00	\$4,00
Servomotor 360 grados	1	\$7,00	\$7,00
Fotorresistencia	3	\$0,45	\$1,35
Total			\$119,67

El costo de fabricación de piezas impresas en 3D, utilizando filamento PETG, depende principalmente de tres factores: el material de elaboración, el tiempo de impresión y la cantidad de energía consumida durante este proceso. Para este caso, se empleará 1 kg de filamento a un costo de compra de 21 dólares. Además, considerando un costo de consumo eléctrico de 10 centavos de dólar por kWh en Ecuador, se calcula el gasto energético asociado a la impresión.

Tabla 4.10 Costos de componentes fabricados por impresión 3D.

Pieza	Cantidad	Cantidad de material	Tiempo de impresión	Consumo eléctrico	Costo Total
Chasis	1	67g	6.85h	\$0.69	\$2.09
Tapas de motorreductor	1	2g	0.20h	\$0.02	\$0.06
Base sensores de línea	1	8g	0.87h	\$0.09	\$0.26
Separación de nivel	1	25g	2.38h	\$0.24	\$0.76
Base superior	1	59g	6.33h	\$0.63	\$1.87
Tapa superior	1	24g	2.43h	\$0.24	\$0.75
Antenas	1	10g	1.18h	\$0.12	\$0.33
Brazos	1	8g	0.93h	\$0.09	\$0.26
Cubre guía posterior	1	8g	0.93h	\$0.09	\$0.26
Cubre guías frontales	1	11g	1.08h	\$0.11	\$0.34
Soportes marcadores	1	14g	1.51h	\$0.15	\$0.45
Base sensor de luz	1	13g	1.54h	\$0.15	\$0.43
Tapa sensor de luz	1	1g	0.13h	\$0.01	\$0.03
Base rotatoria	1	27g	2.95h	\$0.30	\$0.86
Tapa base rotatoria	1	3g	0.33h	\$0.03	\$0.10
Base sensor ultrasónico	1	10g	0.91h	\$0.09	\$0.30
Tapa sensor ultrasónico	1	5g	0.46h	\$0.05	\$0.15
Base regulable	1	8g	0.90h	\$0.09	\$0.26
Soporte sensor de color	1	9g	0.93h	\$0.09	\$0.28
Base sensor de color	1	10g	0.95h	\$0.10	\$0.31
Tapa sensor de color	1	5g	0.53h	\$0.05	\$0.16
Cubo interactivo	10	46g	3.23h	\$0.32	\$9.98
Total					\$20.28

Finalmente, la Tabla 4.11 detalla el costo total de fabricación del dispositivo modular, incluyendo el uso de licencias de software y el costo de la mano de obra de

ensamblaje. Además, cabe señalar que el fabricar el robot requiere un tiempo estimado de trabajo de 16 horas, lo que implica un mínimo de dos días para su producción.

Tabla 4.11 Lista de costos total de producción del dispositivo modular.

Detalle	Costo
Componentes electrónicos estándar	\$119,67
Componentes fabricados por impresión 3D	\$20.28
Mano de obra	\$47.00
Software de diseño 3D	\$0.00 (Licencia académica)
Software de diseño de circuitos (Flux IA)	\$0.00 (Plan gratuito)
Software para diseño de la interfaz (Visual Studio)	\$0.00 (Licencia académica)
Software de programación del microcontrolador (Arduino IDE)	\$0.00 (Plan gratuito)
Total	\$186.95

4.8 Manual de usuario

Para el uso correcto del dispositivo dentro de un ambiente educativo se ha generado un manual de usuario el cual se encuentra detallado en el Anexo E, esta guía presenta información necesaria del uso y funcionamiento que requiere el dispositivo modular para su correcta ejecución.

4.9 Actividades propuestas con el robot modular

Con el fin de integrar al dispositivo modular en un salón de clase, se han diseñado matrices de procesos didácticos, presentes en las Tablas 4.12, 4.13, 4.14 y 4.15. Estas matrices se centran en el desarrollo de destrezas específicas para niños de primer grado de educación general básica, en concordancia con los lineamientos establecidos por el Ministerio de Educación de Ecuador [50]. Las matrices organizan las actividades de aprendizaje que se llevarán a cabo mediante el uso del dispositivo robótico, en colaboración con el docente asignado.

Tabla 4.12 Matriz de procesos didácticos enfocada al reconocimiento de figuras.

Destreza	Proceso Didáctico	Recursos	Evaluación
	Inicio		Indicador
	Motivación: Adivinanzas.		Señalar y nombrar las figuras geométricas básicas (triángulo, cuadrado, rectángulo y círculo) en objetos presentes en su entorno.
	Tema: Las figuras geométricas en el entorno.		
	Objetivo: Reconocer e identificar figuras geométricas en el entorno.		
	Prerrequisitos: Conocer el vocabulario relacionado con las formas geométricas como: línea recta, curva, largo o corto.		
	Desarrollo		Instrucción
Reconocer figuras geométricas (triángulo, cuadrado y círculo) en objetos del entorno.	Dibujar las tres figuras básicas: cuadrado, círculo y triángulo, sobre el tapete interactivo con ayuda del robot.	Kit de aprendizaje del robot modular.	Con ayuda del robot en su modalidad de dibujante de trazos los niños deberán adivinar la figura que este realice sobre el tapete de trazos, además de mencionar el nombre de un objeto que posea similitud con esta.
	Nombrar y describir las figuras básicas, así como los trazos que las conforman.	Material didáctico de los estudiantes y el aula.	
	Comparar las figuras con elementos que se encuentren en el entorno.		
	Finalización		
	Dibujar un objeto del entorno basándose en una de las figuras geométricas: cuadrado, círculo, rectángulo y triángulo.		
	Reforzar las características de las figuras básicas.		

Tabla 4.13 Matriz de procesos didácticos enfocada a la identificación de cantidades.

Destreza	Proceso Didáctico	Recursos	Evaluación
	Inicio		Indicador
	Motivación: Canciones.		Comprender la representación numérica de cantidades específicas de objetos.
	Tema: Los números en cantidad hasta el 10.		
	Objetivo: Reconocer y representar cantidades de objetos hasta el 10.		
	Prerrequisitos: Comprender la diferencia con relación a cantidad: mucho, poco o ninguno, además de números fundamentales como el uno, dos y tres.		
	Desarrollo		Instrucción
Identificar cantidades y asociarlas con los numerales 1 al 10 y el 0.	Interactuar con cubos relacionando su cantidad con su valor numérico.	Kit de aprendizaje del robot modular.	Con ayuda del robot en su modalidad estándar y sus expresiones, evaluará los diferentes grupos de objetos que formen los niños.
	Agrupar cubos siguiendo una secuencia del 1 al 10 con ayuda del robot en su modalidad de grúa, mientras se va nombrando su valor numérico.	Material didáctico de los estudiantes y el aula.	
	Construir torres de cubos u otros objetos de forma aleatoria para así poder asimilar su cantidad.		
	Finalización		
	Realizar grupos de objetos en diferentes cantidades asignadas por el docente.		
	Reforzar la representación de los números en diferentes cantidades.		

Tabla 4.14 Matriz de procesos didácticos enfocada a secuencias con números ordinales.

Destreza	Proceso Didáctico	Recursos	Evaluación
	Inicio		Indicador
	Motivación: Cuentos.		Identificar secuencias de hasta cinco elementos y asignar correctamente los números ordinales (primero, segundo, tercero, cuarto, quinto) a cada uno.
	Tema: Noción de orden en secuencia del primero al quinto.		
	Objetivo: Comparar posiciones de los objetos en una secuencia.		
	Prerrequisitos: Conocer los números ordinales del uno al cinco.		
	Desarrollo		Instrucción
Utilizar los números ordinales, del primero al quinto, en la ubicación de elementos del entorno.	Dibujar la representación simbólica de los números hasta el 5 sobre el tapete interactivo con ayuda del robot.	Kit de aprendizaje del robot modular.	El docente repartirá los cubos numerados por toda la clase, con el fin de que los niños tracen un camino que cruce por cada cubo siguiendo una secuencia. Posteriormente con ayuda del robot en su modalidad de seguidor de líneas evaluará el cumplimiento de la secuencia antes mencionada.
	Asemejar los símbolos numéricos con secuencias cotidianas.	Material didáctico de los estudiantes y el aula.	
	Interactuar con cubos, los cuales poseen el símbolo de un número marcado en su superficie.		
	Finalización		
	Guiar a los niños junto con el robot en su modalidad seguidor de luz hacia diferentes cubos repartidos en el salón siguiendo una secuencia lógica.		
	Reforzar la representación simbólica de los números y su posición simbólica.		

Tabla 4.15 Matriz de procesos didácticos enfocada a reconocer la posición de objetos mediante la lateralidad.

Destreza	Proceso Didáctico	Recursos	Evaluación	
Reconocer la posición de objetos del entorno: derecha, izquierda.	Inicio		Indicador	
		Motivación: Canciones.	Demostrar la capacidad de identificar y describir la posición de objetos con relación a sí mismo y a otros.	
		Tema: Posición izquierda y derecha en el entorno.		
		Objetivo: Facilita la coordinación de movimientos en el entorno.		
		Prerrequisitos: Conocimiento básico respecto a las partes del cuerpo.		
		Desarrollo		
		Interactuar con fichas, las cuales contienen plasmadas flechas direccionales y partes del cuerpo.	Kit de aprendizaje del robot modular.	El instructor colocará un cubo en el centro de la clase con la finalidad de que los niños guíen al robot en su modalidad evasor de objetos. Los niños deberán cubrir los lados izquierdo y derecho del robot con cartulinas para así poder guiar al robot hasta el objeto antes colocado.
		Identificar lados laterales del cuerpo, así como los objetos que los rodean con ayuda de las fichas y el docente.	Material didáctico de los estudiantes y el aula.	
		Intercambiar las fichas mientras se realizan actividades como saltar o levantar la mano que indique la ficha.		
		Finalización		
	Mencionar y replicar los movimientos que realice el robot al mostrarle diferentes fichas en su modalidad de lector de colores.			
	Reforzar la posición que ocupa los lados: izquierda y derecha en otros objetos.			

4.10 Pruebas de campo

Con el fin de determinar la correcta aplicación del dispositivo modular en un ambiente de aprendizaje STEAM, se han realizado pruebas enfocadas a niños de una institución educativa local. El propósito de estas pruebas es asegurar que el dispositivo cumpla con los requisitos de funcionamiento esperados dentro de las actividades especificadas en las matrices de procesos didácticos.

En la Fig. 4.34, se observa el robot en su modalidad de marcador de trazos, dando énfasis a la enseñanza de figuras geométricas y los trazos que las conforman.



Fig. 4.34 Aplicación del dispositivo robótico enfocado a la enseñanza de figuras geométricas básicas.

La Tabla 4.16 revela una efectividad del dispositivo del 75% en cuanto a la reproducción de figuras geométricas cuadradas. El error registrado en la prueba número 2 se atribuye a irregularidades en la superficie donde se instaló el tapete interactivo, las cuales obstaculizaron el trazado preciso de las cuatro líneas que conforman la figura.

Tabla 4.16 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de figuras geométricas.

Destreza	Número de prueba	Trazar Figura (Círculo)	Trazar Figura (Cuadrado)	Trazar Figura (Triángulo)
Reconocer figuras geométricas (triángulo, cuadrado y círculo) en objetos del entorno.	1	Trazo correcto	Trazo correcto	Trazo correcto
	2	Trazo correcto	Trazo incorrecto	Trazo correcto
	3	Trazo correcto	Trazo correcto	Trazo correcto
	4	Trazo correcto	Trazo correcto	Trazo correcto

Los resultados de la Tabla 4.17 reflejan un porcentaje positivo del 80%, con respecto al reconocimiento inmediato de figuras geométricas por parte de cinco participantes. La selección incorrecta del participante B surge debido a su respuesta inmediata pero errónea, confundiendo el uso y la cantidad de líneas rectas que posee un cuadrado con las de un triángulo.

Tabla 4.17 Resultados de aprendizaje obtenidos por parte de los participantes (primera clase).

Destreza	Participante	Evaluación (Adivinanza)	Resultado Participante	Funcionamiento del dispositivo (Trazar Figura)
Reconocer figuras geométricas (triángulo, cuadrado y círculo) en objetos del entorno.	A	Círculo	Selección correcta	Correcto
	B	Cuadrado	Selección incorrecta	Correcto
	C	Triangulo	Selección correcta	Correcto
	D	Cuadrado	Selección correcta	Correcto
	E	Triangulo	Selección correcta	Correcto

La Fig. 4.35 ilustra las actividades realizadas durante la segunda clase, cuyo objetivo fue la identificación de cantidades.



Fig. 4.35 Aplicación del dispositivo robótico enfocado a la enseñanza de la identificación de cantidades.

La Tabla 4.18 revela una efectividad del dispositivo del 100% en cuanto a la enseñanza de la identificación de cantidades. El robot en su modalidad de grúa cumplió correctamente con el desplazamiento de cada uno de los cubos numerados hasta el 10, formando así grupos de objetos y siguiendo así una secuencia lógica.

Tabla 4.18 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de la identificación de cantidades.

Destreza	Número de prueba	Desplazar objetos hasta el 5	Desplazar objetos hasta el 10
Identificar cantidades y asociarlas con los numerales 1 al 10 y el 0.	1	Trayectoria correcta	Trayectoria correcta
	2	Trayectoria correcta	Trayectoria correcta
	3	Trayectoria correcta	Trayectoria correcta
	4	Trayectoria correcta	Trayectoria correcta
	5	Trayectoria correcta	Trayectoria correcta

Los resultados presentados en la Tabla 4.19 indican un 80% de acierto en la evaluación de grupos de objetos, aplicada a cinco participantes. La selección incorrecta del participante B se atribuye a la cantidad elevada de objetos propuesta por el docente, lo que generó confusión y llevó al estudiante a formar un grupo de 7 objetos.

Tabla 4.19 Resultados de aprendizaje obtenidos por parte de los participantes (segunda clase).

Destreza	Participante	Evaluación (Grupos de objetos establecidos)	Resultado Participante	Funcionamiento del dispositivo (Expresiones)
Identificar cantidades y asociarlas con los numerales 1 al 10 y el 0.	A	4	Cantidad correcta	Correcto
	B	8	Cantidad incorrecta	Correcto
	C	5	Cantidad correcta	Correcto
	D	3	Cantidad correcta	Correcto
	E	1	Cantidad correcta	Correcto

Para la tercera clase, enfocada en el aprendizaje de secuencias con números ordinales, se utilizaron tres modalidades del dispositivo robótico, como se observa en la Fig. 4.36.

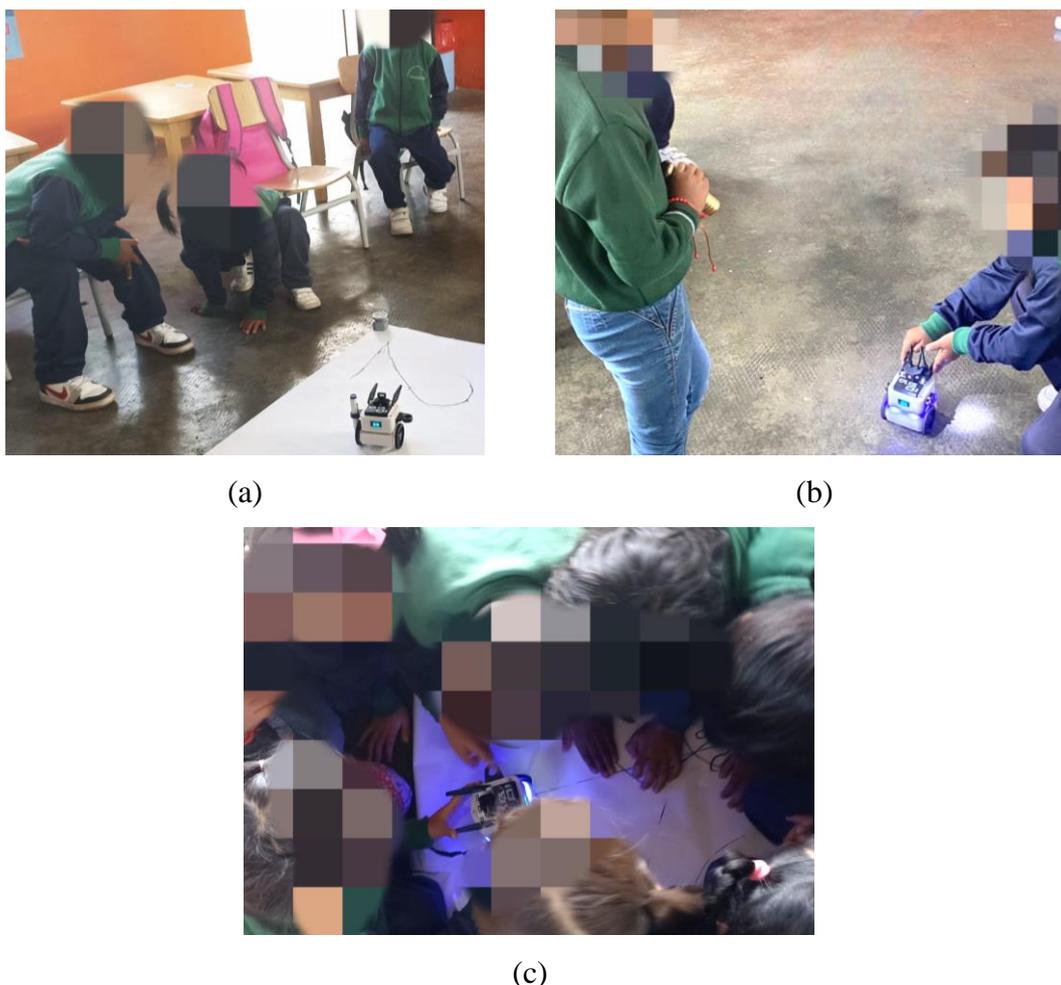


Fig. 4.36 Aplicación del dispositivo robótico enfocado al aprendizaje de secuencias con números; (a) Dibujar números, (b) Seguir luz, (c) Seguir líneas.

Los resultados de la Tabla 4.20 indican una funcionalidad del 80% en el trazado de números por parte del dispositivo. El error registrado en la prueba número 4 se atribuye al trazado incorrecto de las esquinas que conforman este número, además de irregularidades presentes en la superficie donde se instaló el tapate interactivo.

Tabla 4.20 Pruebas de funcionamiento del dispositivo enfocado a la enseñanza de secuencias con números ordinales.

Destreza	Número de prueba	Trazar Números	Seguir Luz
Utilizar los números ordinales, del primero al quinto, en la ubicación de elementos del entorno.	1	Trazo correcto	Trayectoria correcta
	2	Trazo correcto	Trayectoria correcta
	3	Trazo correcto	Trayectoria correcta
	4	Trazo incorrecto	Trayectoria correcta
	5	Trazo correcto	Trayectoria correcta

Los resultados presentados en la Tabla 4.21 indican un 80% de acierto en la evaluación de secuencias. La trayectoria incorrecta realizada por el participante E se atribuye a la complejidad de la secuencia y a la falta de habilidades memorización del estudiante.

Tabla 4.21 Resultados de aprendizaje obtenidos por parte de los participantes (tercera clase).

Destreza	Participante	Evaluación (Realizar secuencias)	Resultado Participante	Funcionamiento del dispositivo (Seguir líneas)
Utilizar los números ordinales, del primero al quinto, en la ubicación de elementos del entorno.	A	1,3,2,4,5	Trayectoria correcta	Correcto
	B	5,4,3,2,1	Trayectoria correcta	Correcto
	C	2,4,3,5,1	Trayectoria correcta	Correcto
	D	3,1,2,5,4	Trayectoria correcta	Correcto
	E	4,3,5,1,2	Trayectoria incorrecta	Correcto

La cuarta clase se centra en el reconocimiento de la posición de objetos a través de la lateralidad, como se observa en la Fig. 4.37.



Fig. 4.37 Aplicación del dispositivo robótico enfocado al reconocimiento de la posición de objetos a través de la lateralidad.

Los resultados de la Tabla 4.22 muestran que el dispositivo en su modalidad de lector de colores posee una funcionalidad del 83.3%, en cuanto al reconocimiento de fichas con direccionamiento derecho. El resultado erróneo reflejado en la prueba número 3 se debe a una lectura incorrecta por parte del sensor de color. Debido a la reflexión lumínica de las fichas con el ambiente, el sensor interpretó erróneamente el color verde como azul.

Tabla 4.22 Pruebas de funcionamiento del dispositivo enfocado al reconocimiento de la posición de objetos a través de la lateralidad.

Destreza	Número de prueba	Mostrar (Ficha Izquierda)	Mostrar (Ficha Derecha)
Reconocer la posición de objetos del entorno: derecha, izquierda.	1	Reconocimiento correcto	Reconocimiento correcto
	2	Reconocimiento correcto	Reconocimiento correcto
	3	Reconocimiento correcto	Reconocimiento incorrecto
	4	Reconocimiento correcto	Reconocimiento correcto
	5	Reconocimiento correcto	Reconocimiento correcto
	6	Reconocimiento correcto	Reconocimiento correcto

Por último, los resultados de la Tabla 4.23 reflejan un porcentaje del 100%, con respecto a las respuestas obtenidas por parte de los 6 participantes con relación al reconocimiento de la posición de objetos a través de la lateralidad.

Tabla 4.23 Resultados de aprendizaje obtenidos por parte de los participantes (cuarta clase).

Destreza	Participante	Evaluación (Identificar posición)	Resultado Participante	Funcionamiento del dispositivo (Evadir)
Reconocer la posición de objetos del entorno: derecha, izquierda.	A	Izquierda	Identificación correcta	Correcto
	B	Derecha	Identificación correcta	Correcto
	C	Izquierda	Identificación correcta	Correcto
	D	Derecha	Identificación correcta	Correcto
	E	Izquierda	Identificación correcta	Correcto
	F	Derecha	Identificación correcta	Correcto

CONCLUSIONES

La investigación documental de los parámetros relacionados con la robótica educativa revela un potencial significativo con relación al uso de un robot modular dentro de un entorno de aprendizaje activo basado en la metodología STEAM. El uso de este dispositivo fomenta el desarrollo de habilidades enfocadas a la resolución de problemas, por medio de un aprendizaje más creativo, colaborativo y centrado en el estudiante.

El diseño y desarrollo de un robot modular que cumpla con los criterios de intuitividad (facilidad de ensamblaje), robustez (resistencia a impactos leves durante la manipulación), seguridad (uso de materiales no tóxicos), valor educativo y autonomía, ha requerido la implementación de herramientas CAD y CAM. Estas herramientas han sido fundamentales para la visualización y el análisis del modelo 3D, además de facilitar las etapas de fabricación y ensamblaje del dispositivo.

Por otro lado, la integración del robot modular al currículo de la Educación General Básica ofrece a los docentes la oportunidad de generar procesos didácticos innovadores para los estudiantes desde una edad temprana. Las matrices generadas se centran en el desarrollo de cuatro destrezas fundamentales para el pensamiento lógico matemático: el reconocimiento de figuras geométricas, la identificación de cantidades, el razonamiento de secuencias y el reconocimiento de posiciones.

Por último, las pruebas de campo realizadas con el dispositivo modular concluyen que los estudiantes (4 a 6 años) de la unidad educativa general básica presentan un mayor nivel de atención en cuanto al aprendizaje. La destreza con la mayor calificación de respuesta del 100% corresponde al reconocimiento la posición de objetos, actividad que se evaluó en parejas con ayuda del dispositivo en su modalidad de evasor de objetos, obteniendo así un dominio total de las nociones de izquierda y derecha por parte de los 6 participantes.

RECOMENDACIONES

Se recomienda realizar detalladamente una matriz de procesos didácticos por cada destreza que se desee impartir dentro de una clase. Esto con el fin de garantizar una correcta integración del dispositivo modular, asegurando que este sea utilizado de manera coherente y estratégica en cada fase del proceso educativo.

En caso de no poseer el suficiente conocimiento en cuanto al uso de herramientas tecnológicas enfocadas al desarrollo del aprendizaje, se sugiere crear programas de capacitación enfocados a la propagación de experiencias y resultados obtenidos a través de la aplicación de la robótica en entornos educativos.

Adicionalmente, el proceso de construcción y ensamblaje del dispositivo modular llega a convertirse en una herramienta pedagógica capaz de fomentar la enseñanza de la robótica en centros de educación superior. La complejidad que posee el desarrollo de este proyecto demanda un nivel de conocimiento y habilidades que promueve el progreso de competencias clave en los estudiantes.

Por último, dado que el dispositivo modular ha sido fabricado en su mayoría mediante la impresión 3D, se recomienda su uso en entornos controlados como lo es un salón de clase. La exposición prolongada a condiciones climáticas adversas, como viento y lluvia, podría comprometer la integridad estructural y electrónica del robot a largo plazo.

REFERENCIAS

- [1] N. A. Narváz Villarreal. “Enfoque Steam como metodología activa del aprendizaje en estudiantes de quinto año de EGB en La Unidad Educativa Juan Montalvo, año lectivo 2022-2023,” B.S. Thesis., 2023. [Online]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/14705>
- [2] D. A. Celis Cuervo y R. A. González Reyes, “Aporte de la metodología Steam en los procesos curriculares”, *Revista Boletín Redipe*, vol. 10, n.o 8, pp. 279-302, ago. 2021, doi: 10.36260/rbr.v10i8.1405.
- [3] D. M. Gray, “¿Cómo prepararse para carreras que aún no existen?”, *Www.ey.com*, 04-jun-2019. [En línea]. Disponible en: https://www.ey.com/es_ec/corporate-responsibility/how-do-you-prepare-for-careers-that-dont-exist-yet.
- [4] K. L. A. Amaya, R. O. A. Rivadeneira, A. M. E. Espino, Z. R. M. Chávez, F. O. Cabrera, y D. Q. de la Torre. “Tecnología educativa para desarrollar la metodología STEAM”, vol. 1. Mar Caribe de Josefrank Pernaleté Lugo, 2023.
- [5] J. V. A. Calle, E. F. Caicedo y P. C. Villavicencio, “Guía de apoyo para los docentes en la implementación de metodología STEM – STEAM”, Ministerio de Educación, 2021. [En línea]. Disponible en: <https://recursos.educacion.gob.ec/red/orientaciones-para-la-aplicacion-del-curriculo-priorizado-con-enfasis/>
- [6] F. Ruiz Vicente, A. Zapatera, N. Montes, y N. Rosillo, “Proyectos STEAM con LEGO Mindstorms para educación primaria en España”, en *Proceedings INNODOCT/18. International Conference on Innovation, Documentation and Education*, València: Editorial Universitat Politècnica de València, nov. 2018. doi: 10.4995/INN2018.2018.8836.
- [7] P. A. Álvarez, “Creación y desarrollo de un juego empleando nuevas tecnologías y robótica”, 2022.
- [8] M. G. da Silva Filgueira y C. S. González González, “PequeBot: Propuesta de un Sistema Ludificado de Robótica Educativa para la Educación Infantil”, 2017. [En línea]. Disponible en: <http://riull.ull.es/xmlui/handle/915/6677>
- [9] J. M. García, “Robótica Educativa. La programación como parte de un proceso educativo.”, *Revista de Educación a Distancia (RED)*, vol. 0, núm. 46, oct. 2015, [En línea]. Disponible en: <https://revistas.um.es/red/article/view/240201>

- [10] J. Ernesto Cárdenas López Renzo Murillo Orrego, “Implementación de la metodología STEAM a través del uso de simuladores para fortalecer los proyectos transversales IE Liceo Andino de la Santísima Trinidad”, 2022.
- [11] A. D. B. Jácome, D. F. Y. Ponce, y C. E. M. Bone, “Implementación de un robot móvil con punto de acceso basado en la metodología steam”, Investigación Tecnológica ISTCT, vol. 2, núm. 1, pp. 10–10, 2020.
- [12] J. C. Segovia Sánchez. “Estrategia metodológica para la enseñanza de la robótica a estudiantes de 3ro de bachillerato en la unidad educativa “San Francisco”, Ibarra, Ecuador”, M.S. Thesis., 2023. [Online]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/14977>
- [13] S. Santo, R. Alexander Valdiviezo Sánchez, y J. Eduardo, “Diseño e implementación de un robot social interactivo orientado a la educación STEAM en niños y niñas de 8 a 12 años por medio de robótica cognitiva para la empresa Artil Robotics SAS de la ciudad de Ambato”.
- [14] L. Andreina y Q. Vera, “Robótica Educativa como estrategia para el desarrollo del pensamiento computacional en los estudiantes de décimo año de Educación General Básica en la Unidad Educativa Particular Santa Mariana de Jesús”.
- [15] J. A. F. Vladimir, “Aplicación de la industria 4.0 en los procesos de enseñanza en la carrera de ingeniería industrial de la Universidad Tecnológica Indoamérica usando metodología STEAM”.
- [16] J. K. Azcaray Fernández, “Metodología para integrar el diseño en un proceso curricular steam a través del uso de las nuevas tecnologías creativas”, Universitat Politècnica de València, Valencia (Spain), 2019. doi: 10.4995/Thesis/10251/125704.
- [17] V. T. Díaz Cedeño, I. M. Salazar Caraballo, y R. López Brito, “Steam: Una breve conceptualización de una metodología orientada al desarrollo de competencias del siglo XXI”, Revista EDUCARE - UPEL-IPB - Segunda Nueva Etapa 2.0, vol. 27, no 2, pp. 73-91, ago. 2023, doi: 10.46498/reduipb.v27i2.1916.
- [18] O. García Fuentes, M. Raposo Rivas, y M. E. Martínez Figueira, “El enfoque educativo STEAM: una revisión de la literatura”, Revista Complutense de Educación, vol. 34, no 1, pp. 191-202, ene. 2023, doi: 10.5209/rced.77261.

- [19] D. F. da C. Martins, N. A. P. Mesquita, y M. J. N. S. Gamboa, “Aprender y crecer con STEAM: una experiencia de diseño en el jardín de infancia”, *Didáctica: Revista de Investigación en Didácticas Específicas*, n.o 10, pp. 21-36, oct. 2021, doi: 10.1344/did.2021.10.21-36.
- [20] M. D. Casa Coila, S. Huatta Pancca, y E. E. Mancha Pineda, “Problem Based Learning as strategy for the development of competences in secondary education students”, *Comunicación: Revista de Investigación en Comunicación y Desarrollo*, vol. 10, n.o 2, pp. 111-121, dic. 2019, doi: 10.33595/2226-1478.10.2.383.
- [21] C. Ciro Aristizabal, “Aprendizaje basado en proyectos (A.B.Pr) como estrategia de enseñanza y aprendizaje en la educación básica y media”. [En línea]. Disponible en: <https://repositorio.unal.edu.co/handle/unal/11717>
- [22] W. Krüger Mariano y A. Chiappe, “Habilidades del siglo XXI y entornos de aprendizaje STEAM: una revisión”, *Revista de Educación a Distancia (RED)*, vol. 21, n.o 68, nov. 2021, doi: 10.6018/red.470461.
- [23] J. V. Viegas-D’Abreu y K. O. Villalba-Condori, “Education and Educative Robotics”, *Revista de Educación a Distancia (RED)*, no 54, jun. 2017, doi: 10.6018/red/54/11.
- [24] Y. A. Caballero-González y A. García-Valcárcel, “¿Aprender con robótica en Educación Primaria? Un medio de estimular el pensamiento computacional”, *Education in the Knowledge Society (EKS)*, vol. 21, p. 15, abr. 2020, doi: 10.14201/eks.22957.
- [25] G. Pérez Buj y P. D. Diago Nebot, “Estudio exploratorio sobre lenguajes simbólicos de programación en tareas de resolución de problemas con Bee-bot/ Exploratory study on symbolic programming languages in problem-solving activities with Bee-bot”, *Magister*, vol. 30, n.o 1, p. 9, dic. 2018, doi: 10.17811/msg.30.1.2018.9-20.
- [26] A. Alsina y Y. Acosta, “Conectando la educación matemática infantil y el pensamiento computacional: aprendizaje de patrones de repetición con el robot educativo programable Cubetto”, *Innovaciones Educativas*, vol. 24, n.o 37, pp. 133-148, jul. 2022, doi: 10.22458/ie.v24i37.4022.

- [27] E. Afari y M. S. Khine, “Robotics as an Educational Tool: Impact of Lego Mindstorms”, *International Journal of Information and Education Technology*, vol. 7, n.o 6, pp. 437-442, 2017, doi: 10.18178/ijiet.2017.7.6.908.
- [28] H. Gómez Rodríguez, “Robótica educativa utilizando el mBot en estudiantes de educación básica”, *RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol. 13, n.o 25, sep. 2022, doi: 10.23913/ride.v13i25.1274.
- [29] M. X. Prado Ortega, R. J. Paucar Córdova, J. W. Valarezo Castro, M. T. Acosta Yela, y K. M. Guaicha Soriano, “Beneficios de la programación por bloques utilizando Sphero mini mediante aprendizaje móvil en la educación superior”, *e-Ciencias de la Información*, jun. 2023, doi: 10.15517/eci.v13i2.54814.
- [30] W.-Y. Huang, C.-F. Hu, y C.-C. Wu, “The Use of Different Kinds of Robots to Spark Student Interest in Learning Computational Thinking”, en *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, IEEE, abr. 2018, pp. 11-16. doi: 10.1109/LaTICE.2018.00-13.
- [31] D. Castillo López, “Las TIC en los procesos de enseñanza-aprendizaje desarrollados por maestros tutores de Educación Primaria en la Región de Murcia”, *Revista Interuniversitaria de Investigación en Tecnología Educativa*, pp. 1-14, dic. 2020, doi: 10.6018/riite.432061.
- [32] A. Lopes Guedes, F. Lopes Guedes, y A. C. Guedes Laimer, “Experiencias de robótica educativa”, *TECHNO REVIEW. International Technology, Science and Society Review /Revista Internacional de Tecnología, Ciencia y Sociedad*, vol. 4, n.o 2, pp. 193-204, jun. 2015, doi: 10.37467/gka-revtechno.v4.887.
- [33] G. Urquiza, J. Gavilanes, Á. Llerena, I. Vaca, y L. Villagómez, “Robot publicitario aplicado a la pedagogía infantil”, *Ciencia*, vol. 21, n.o 1, p. 15, nov. 2019, doi: 10.24133/ciencia.v21i1.1514.
- [34] L. Mena, D. Gil, C. A. Monje, S. Martínez, y C. Balaguer, “Diseño de una articulación de dos grados de libertad para robots modulares”, en *XLIII Jornadas de Automática: libro de actas: 7, 8 y 9 de septiembre de 2022, Logroño (La Rioja)*, Servicio de Publicaciones da UDC, 2022, pp. 743-748. doi: 10.17979/spudc.9788497498418.0743.

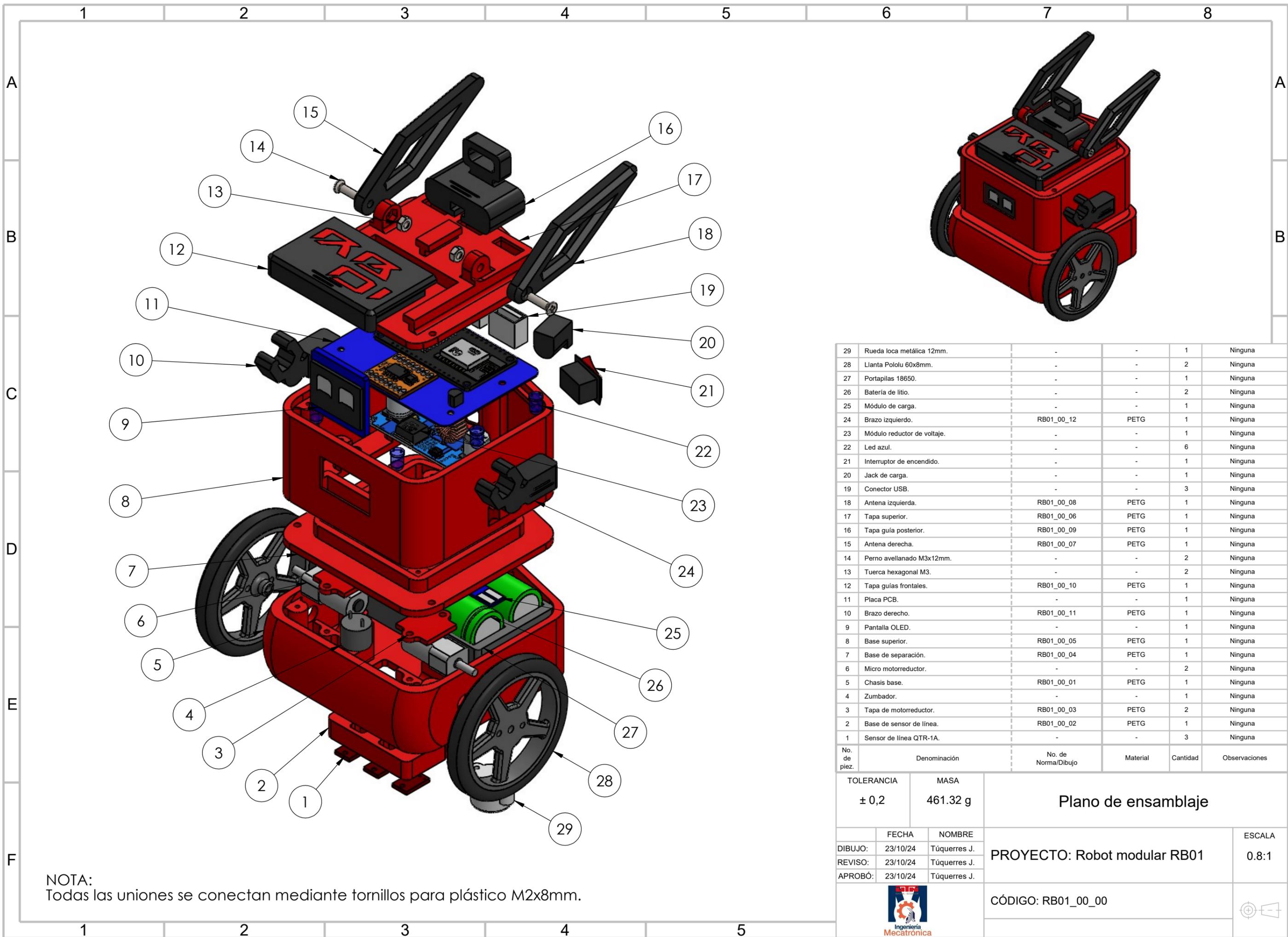
- [35] D. González-Reyes, H. Kim, D. Rubio-Martínez, H. Cervantes-Culebro, y M. Elías-Espinosa, “Metodología de diseño para robots paralelos de cinco eslabones y dos grados de libertad”, *Científica*, vol. 25, n.o 1, pp. 54-70, 2021, doi: 10.46842/ipn.cien.v25n1a05.
- [36] Punto Flotante, “Modulo NODEMCU ESP32 DEVKIT v1 starter kit”, 2022. Disponible en: <https://www.puntoflotante.net/NODEMCU-ESP32-DEVKIT-V1-STARTER-KIT.htm>
- [37] NovaTronic, “TB6612FNG Driver para motor DC”, 2020. Disponible en: <https://novatronic.com/index.php/product/tb6612fng-driver-para-motor-dc/>
- [38] NovaTronic, “Pantalla Lcd Oled Azul 128×64 0.96 I2C”, 2020. Disponible en: <https://novatronic.com/index.php/product/pantalla-lcd-oled-azul-128x64-0-96-i2c/>
- [39] Botland, “Micro motor N20-BT44 250:1 90RPM - 6V”, 2024. Disponible en: <https://botland.store/n20-micro-motors-mp-series-medium-power/12606-micro-motor-n20-bt44-250-1-90rpm-6v-5904422306731.html>
- [40] Electrónica SMD, “Buzzer / Zumbador”, 2024. Disponible en: <https://www.electronicasmd.com/productos/audio/buzzer/>
- [41] NovaTronic, “Sensor reflectivo QTR-1A (2-Pack)”, 2020. Disponible en: <https://novatronic.com/index.php/product/sensor-reflectivo-qtr-1a-2-pack/>
- [42] Tiendatec, “TRANSISTOR 2N2222 NPN”, 2024. Disponible en: <https://www.tiendatec.es/electronica/componentes/transistores/2093-transistor-2n2222-npn-8472496022491.html>
- [43] Tempero Systems, “2S 5A BMS HX-2S-01 Li-ion Battery Protection Module”, 2024. Disponible en: <https://temperosystems.com.au/products/2s-5a-bms-hx-2s-01-li-ion-battery-protection-module/>
- [44] XLSEMI, “Módulo reductor de voltaje 5a xl4015 ajustable step down buck”, 2019. Disponible en: <https://www.alldatasheet.com/datasheet-pdf/view/1134361/XLSEMI/XL4015.html>
- [45] MecatrónicaLATAM, “LDR o fotoresistor o fotoresistencia”, 23 de abril 2021. Disponible: <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/ldr/>

- [46] J. Morgan, “Módulo sensor ultrasónico hcsr-04”, 2014. Disponible en: <https://www.alldatasheet.com/datasheet-pdf/view/1132204/ETC2/HCSR04.html>
- [47] DatasheetHub, “TCS3200 Color Sensor Module With Cap”, 23 de junio 2021. Disponible en: <https://www.datasheethub.com/tcs3200-color-sensor-module-with-cap/>
- [48] Nidek, “Servomotor tower pro sg90 micro 180°”, 2019. Disponible en: <https://www.alldatasheet.es/datasheet-pdf/view/1572383/ETC/SG90.html>
- [49] Sunlu, “PETG and High Speed Matte PETG 3D Printer Filament 1KG”, 15 de mayo 2021. Disponible en: <https://www.sunlu.com/products/over-6kg-bundle-sale-petg-3d-printer-filament-1-75mm-1kg-roll>
- [50] Gob.ec, “Educación general básica – Preparatoria”, Ministerio de Educación, pp.70-89, 2024. Disponible en: <https://educacion.gob.ec/wp-content/uploads/downloads/2016/08/EGB-Preparatoria.pdf>

ANEXOS

Anexo A

Planos del robot modular

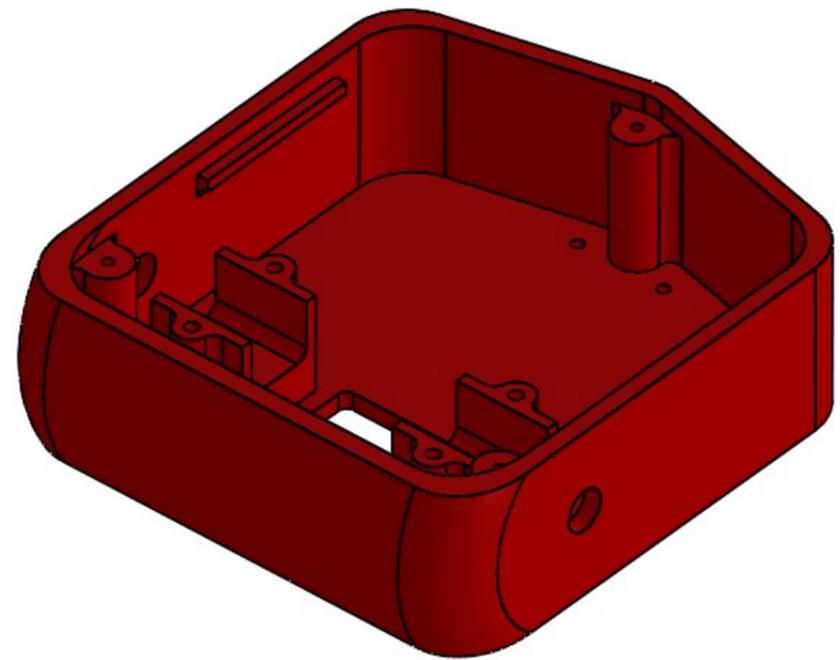
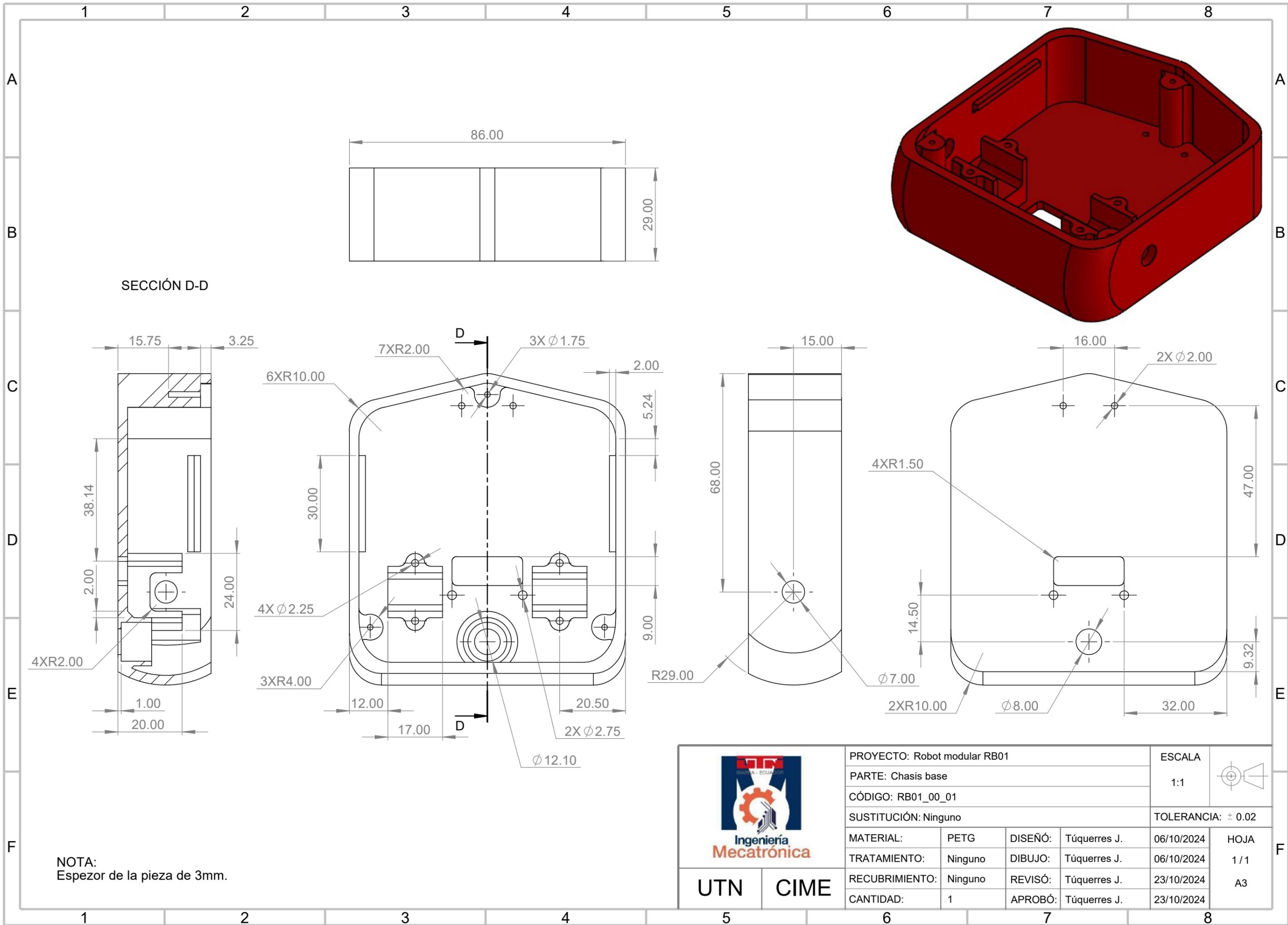


29	Rueda loca metálica 12mm.	-	-	1	Ninguna
28	Llanta Pololu 60x8mm.	-	-	2	Ninguna
27	Portapilas 18650.	-	-	1	Ninguna
26	Batería de litio.	-	-	2	Ninguna
25	Módulo de carga.	-	-	1	Ninguna
24	Brazo izquierdo.	RB01_00_12	PETG	1	Ninguna
23	Módulo reductor de voltaje.	-	-	1	Ninguna
22	Led azul.	-	-	6	Ninguna
21	Interruptor de encendido.	-	-	1	Ninguna
20	Jack de carga.	-	-	1	Ninguna
19	Conector USB.	-	-	3	Ninguna
18	Antena izquierda.	RB01_00_08	PETG	1	Ninguna
17	Tapa superior.	RB01_00_06	PETG	1	Ninguna
16	Tapa guía posterior.	RB01_00_09	PETG	1	Ninguna
15	Antena derecha.	RB01_00_07	PETG	1	Ninguna
14	Perno avellanado M3x12mm.	-	-	2	Ninguna
13	Tuerca hexagonal M3.	-	-	2	Ninguna
12	Tapa guías frontales.	RB01_00_10	PETG	1	Ninguna
11	Placa PCB.	-	-	1	Ninguna
10	Brazo derecho.	RB01_00_11	PETG	1	Ninguna
9	Pantalla OLED.	-	-	1	Ninguna
8	Base superior.	RB01_00_05	PETG	1	Ninguna
7	Base de separación.	RB01_00_04	PETG	1	Ninguna
6	Micro motorreductor.	-	-	2	Ninguna
5	Chasis base.	RB01_00_01	PETG	1	Ninguna
4	Zumbador.	-	-	1	Ninguna
3	Tapa de motorreductor.	RB01_00_03	PETG	2	Ninguna
2	Base de sensor de línea.	RB01_00_02	PETG	1	Ninguna
1	Sensor de línea QTR-1A.	-	-	3	Ninguna
No. de piez.	Denominación	No. de Norma/Dibujo	Material	Cantidad	Observaciones

TOLERANCIA		MASA		Plano de ensamblaje	
± 0,2		461.32 g			
FECHA		NOMBRE		PROYECTO: Robot modular RB01	
DIBUJO: 23/10/24		Túquerres J.			
REVISO: 23/10/24		Túquerres J.			
APROBÓ: 23/10/24		Túquerres J.		ESCALA	
				0.8:1	
				CÓDIGO: RB01_00_00	

NOTA:
Todas las uniones se conectan mediante tornillos para plástico M2x8mm.

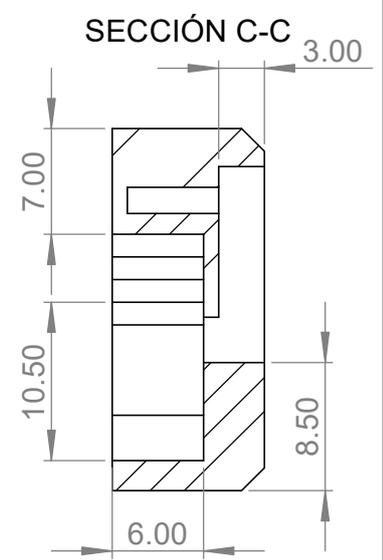
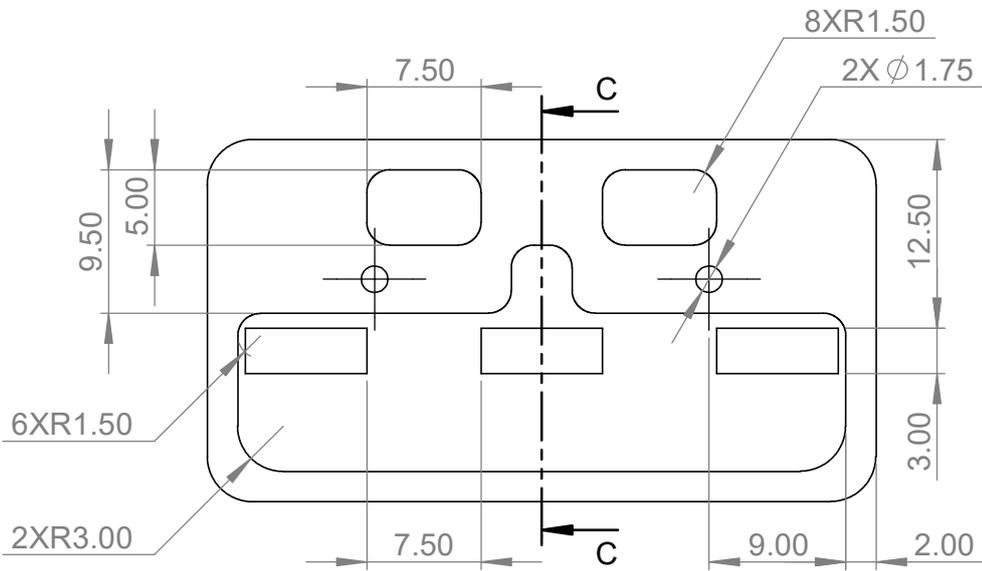
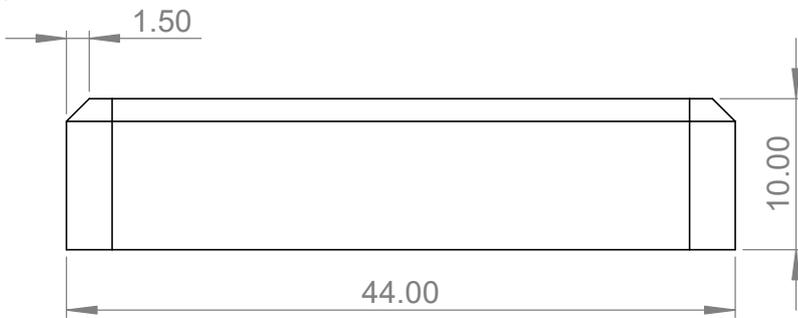
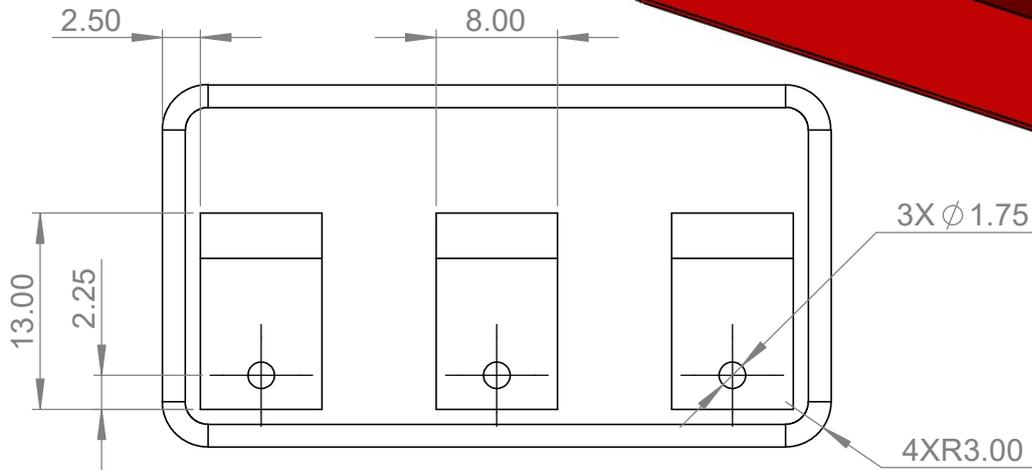
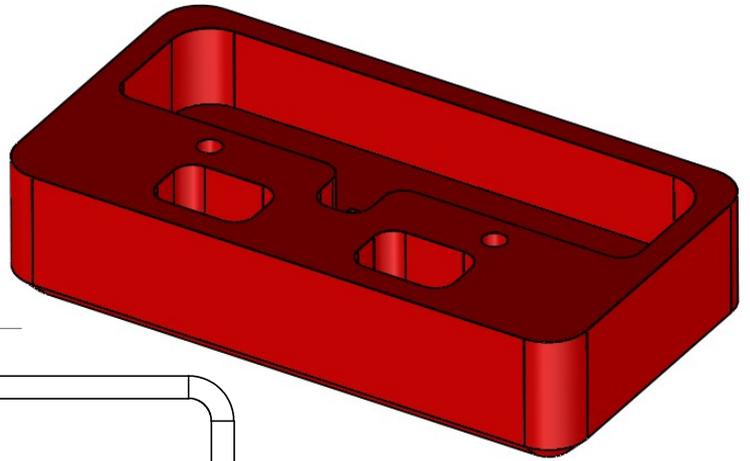




NOTA:
Espesor de la pieza de 3mm.

 Ingeniería Mecatrónica	PROYECTO: Robot modular RB01			ESCALA	 1:1
	PARTE: Chasis base				
	CÓDIGO: RB01_00_01			TOLERANCIA: ± 0.02	
	SUSTITUCIÓN: Ninguno				
MATERIAL:	PETG	DISEÑO:	Túquerres J.	06/10/2024	HOJA 1/1 A3
TRATAMIENTO:	Ninguno	DIBUJO:	Túquerres J.	06/10/2024	
RECUBRIMIENTO:	Ninguno	REVISÓ:	Túquerres J.	23/10/2024	
CANTIDAD:	1	APROBÓ:	Túquerres J.	23/10/2024	
UTN	CIME				

NOTA:
Espezzor de la pieza de 2mm.



PROYECTO: Robot modular RB01

PARTE: Base de sensor de línea

CÓDIGO: RB01_00_02

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

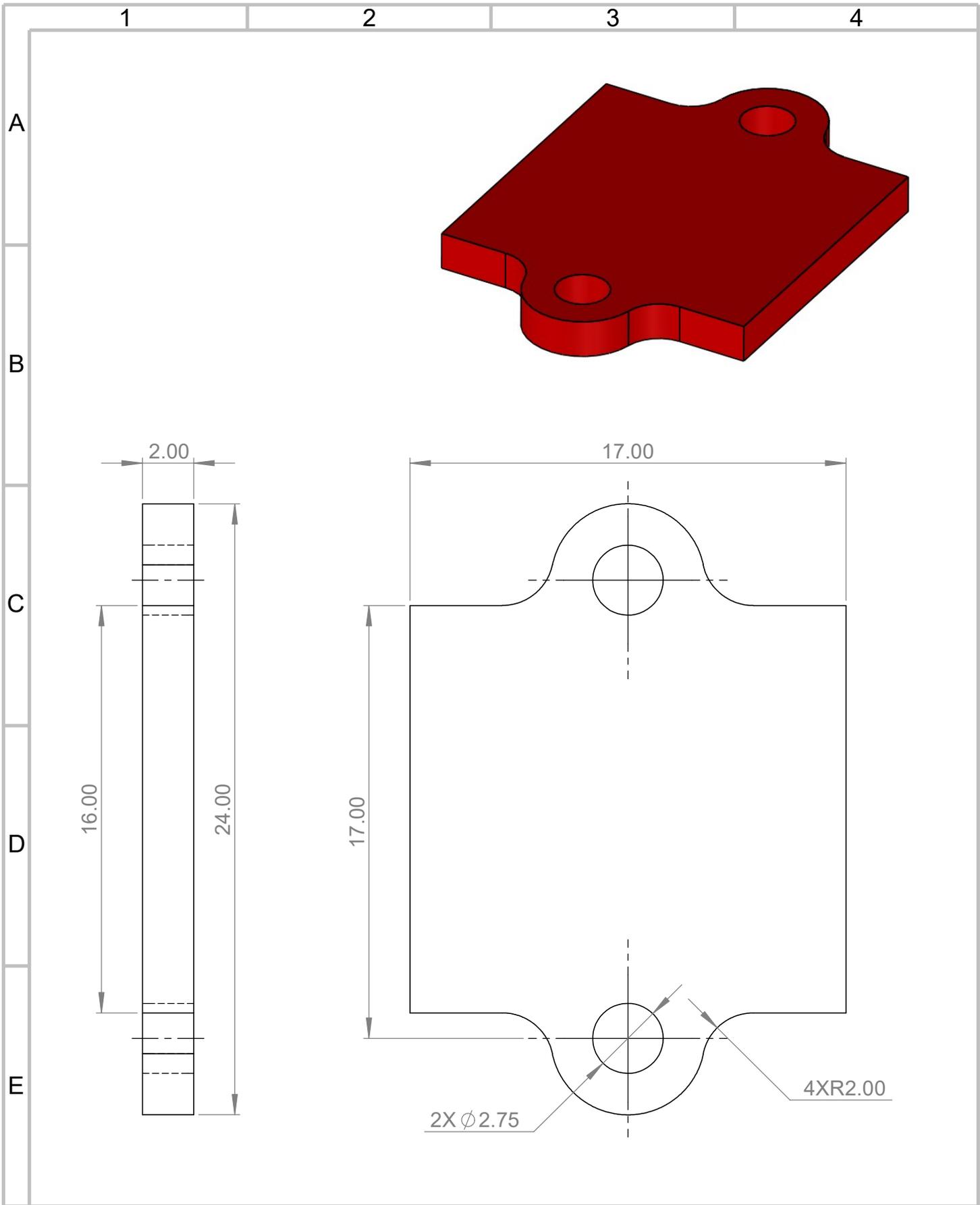
A4

06/10/2024

06/10/2024

23/10/2024

23/10/2024



Ingeniería
Mecatrónica

PROYECTO: Robot modular RB01

PARTE: Tapa motorreductor

CÓDIGO: RB01_00_03

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑÓ:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

2

APROBÓ:

Túquerres J.

ESCALA

5:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

A4

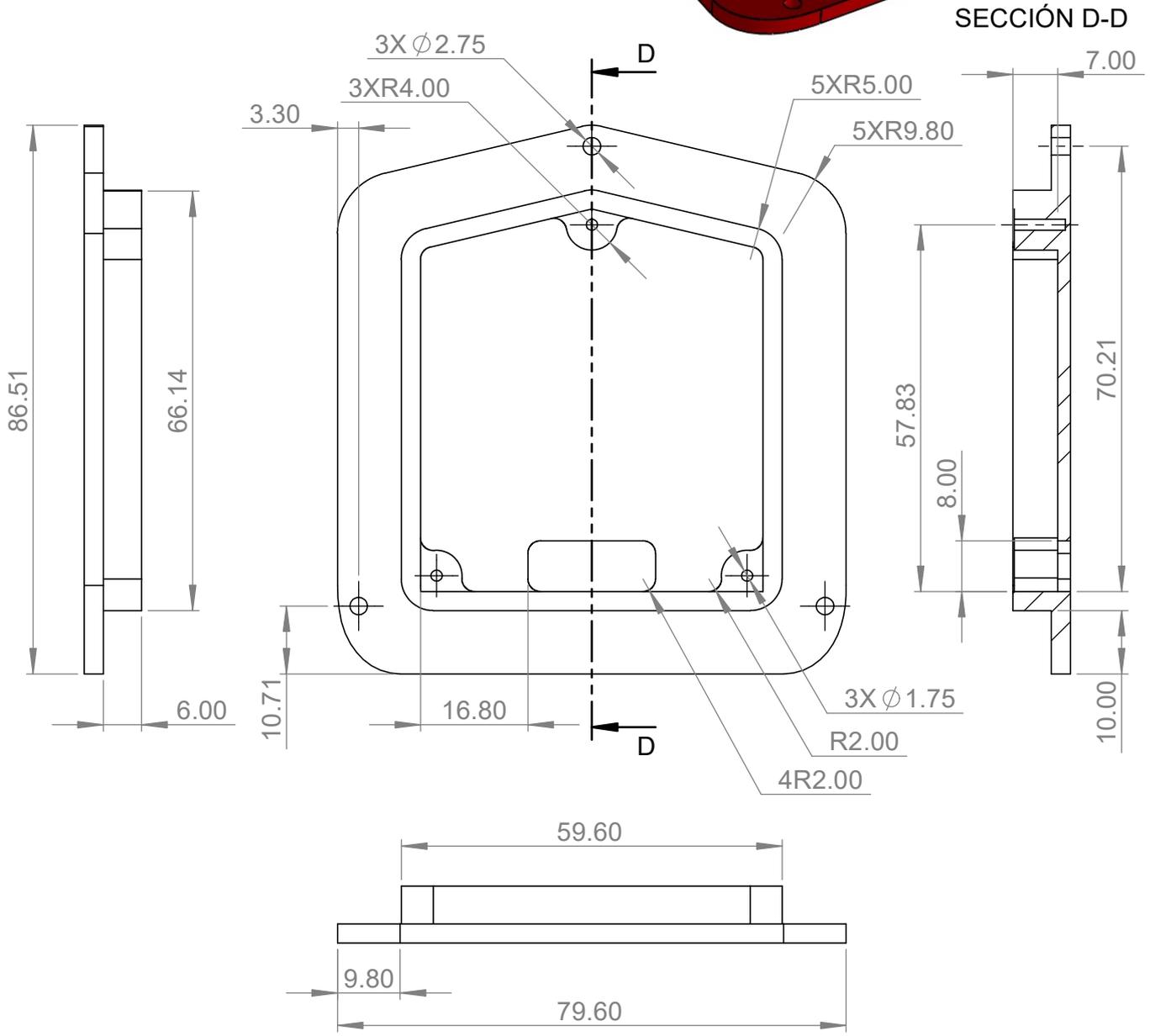
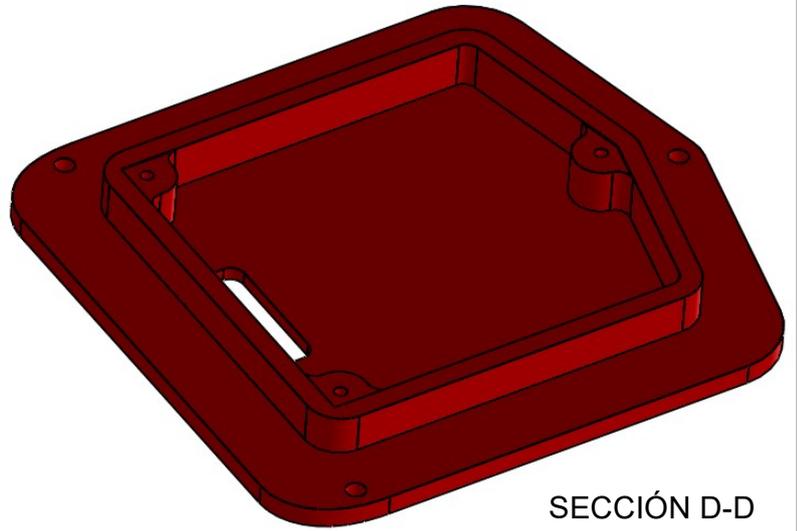
06/10/2024

06/10/2024

23/10/2024

23/10/2024

NOTA:
Espezor de la pieza de 3mm.



PROYECTO: Robot modular RB01

PARTE: Base de separación

CÓDIGO: RB01_00_04

SUSTITUCIÓN: Ninguno

MATERIAL: PETG

TRATAMIENTO: Ninguno

RECUBRIMIENTO: Ninguno

CANTIDAD: 1

DISEÑO: Túquerres J.

DIBUJO: Túquerres J.

REVISÓ: Túquerres J.

APROBÓ: Túquerres J.

ESCALA

1:1

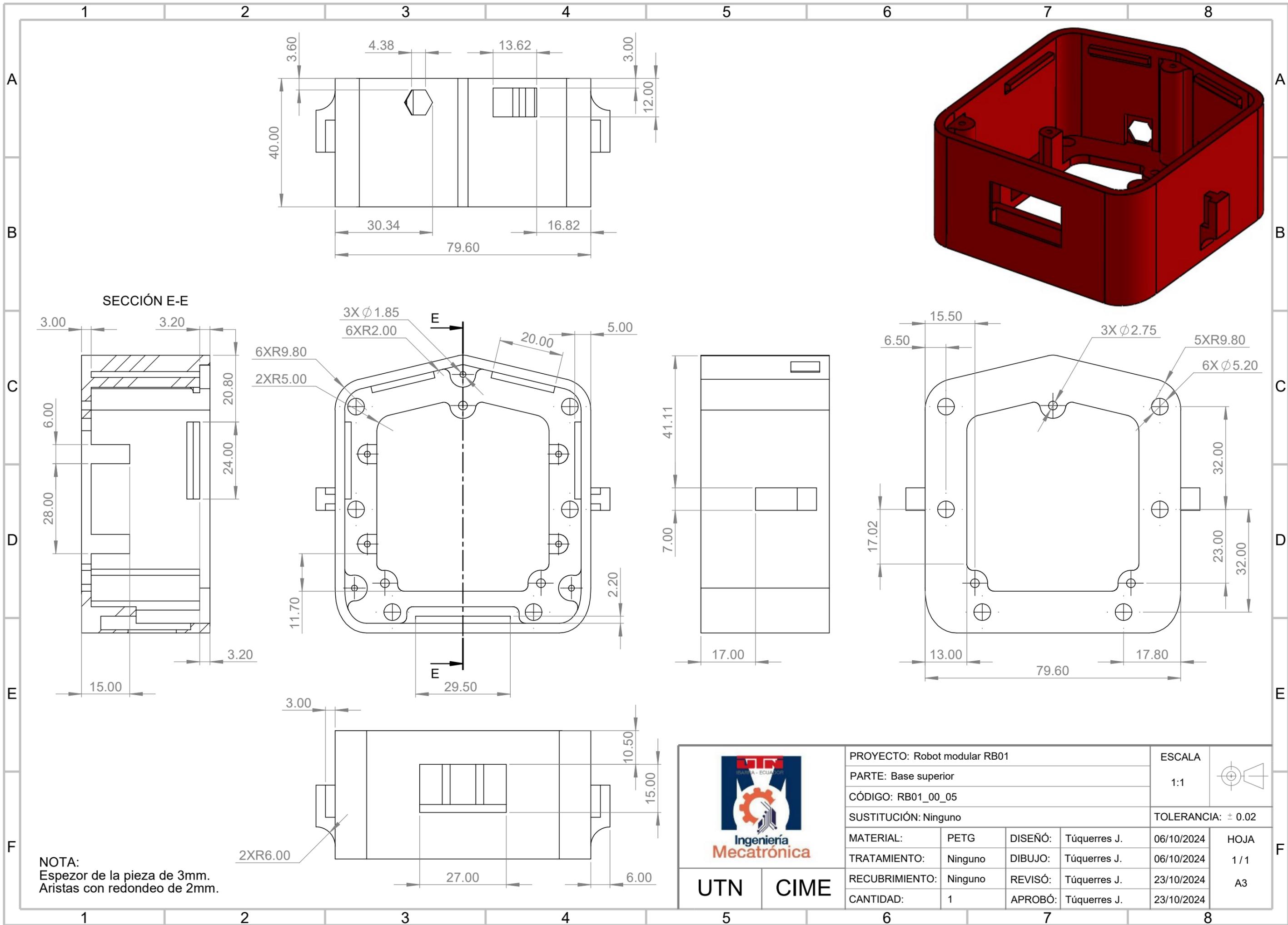


TOLERANCIA: ± 0.02

UTN

CIME

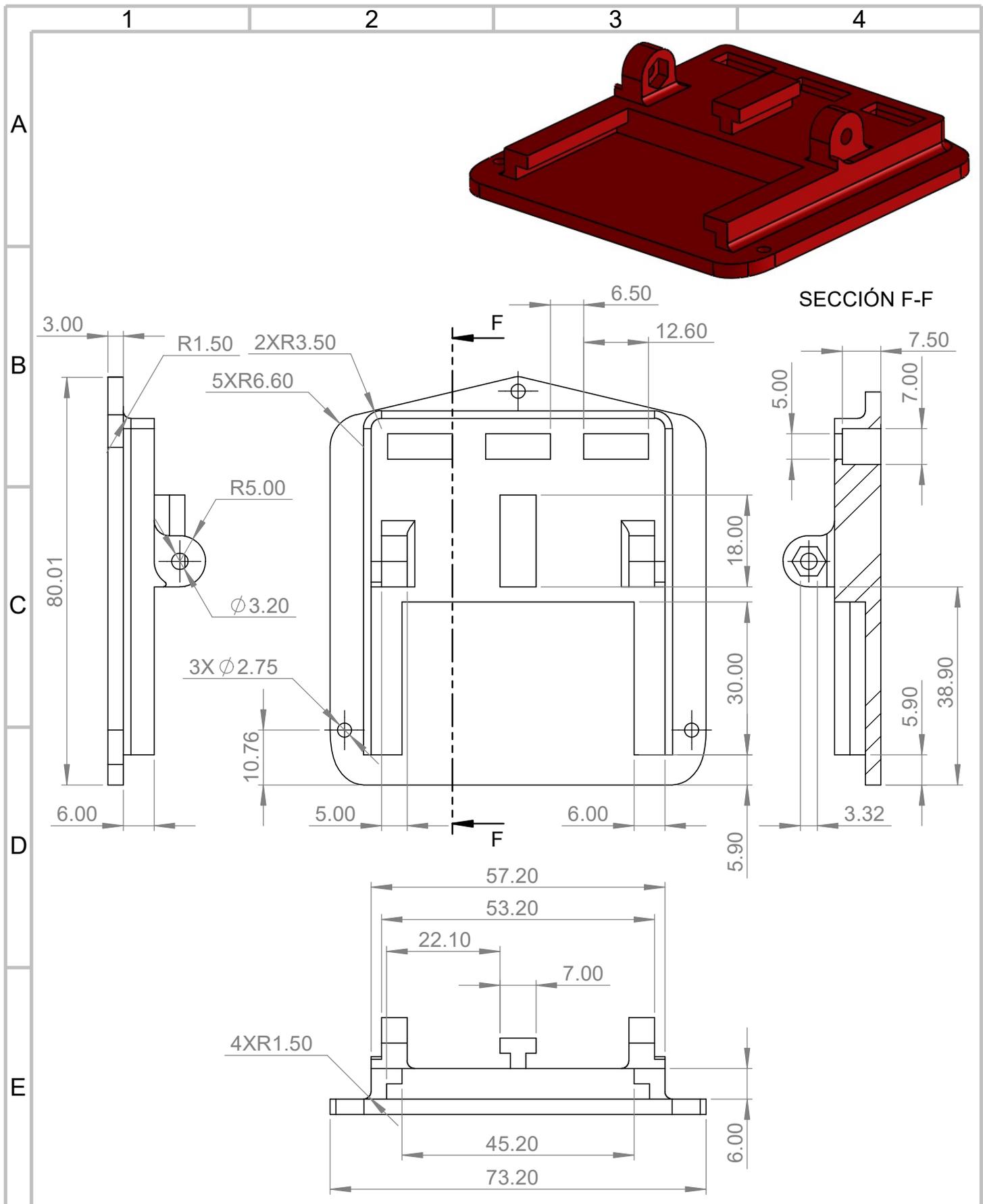
HOJA
1 / 1
A4



NOTA:
 Espesor de la pieza de 3mm.
 Aristas con redondeo de 2mm.

UTN
 Ingeniería
 Mecatrónica

PROYECTO: Robot modular RB01				ESCALA	
PARTE: Base superior				1:1	
CÓDIGO: RB01_00_05				TOLERANCIA: ± 0.02	
SUSTITUCIÓN: Ninguno					
MATERIAL:	PETG	DISEÑO:	Túquerres J.	06/10/2024	HOJA
TRATAMIENTO:	Ninguno	DIBUJO:	Túquerres J.	06/10/2024	1/1
RECUBRIMIENTO:	Ninguno	REVISÓ:	Túquerres J.	23/10/2024	A3
CANTIDAD:	1	APROBÓ:	Túquerres J.	23/10/2024	



Ingeniería
Mecatrónica

PROYECTO: Robot modular RB01

PARTE: Tapa superior

CÓDIGO: RB01_00_06

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

A4

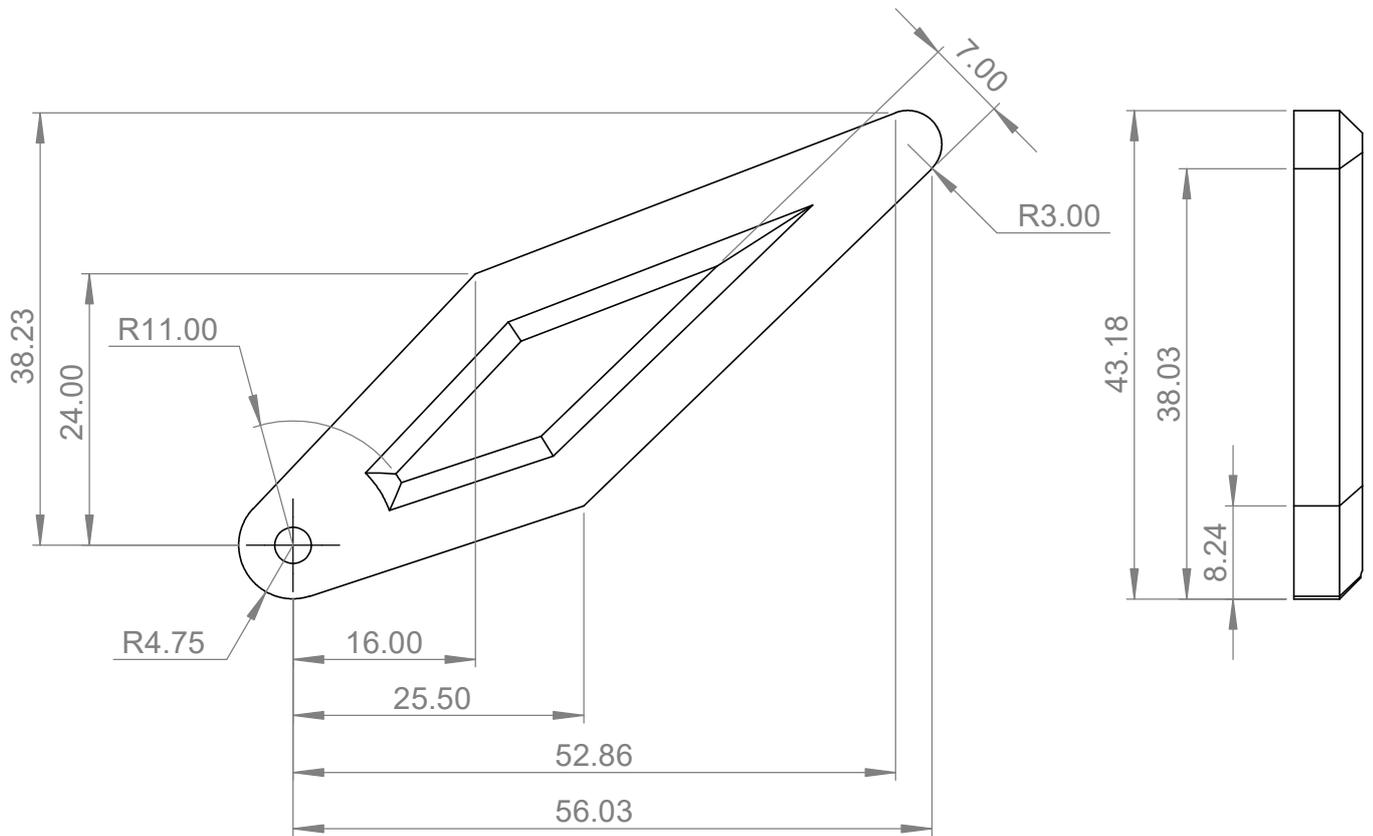
06/10/2024

06/10/2024

23/10/2024

23/10/2024

NOTA:
Chafilanes interno y externo de 2mm.



Ingeniería
Mecatrónica

PROYECTO: Robot modular RB01

PARTE: Antena derecha

CÓDIGO: RB01_00_07

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

A4

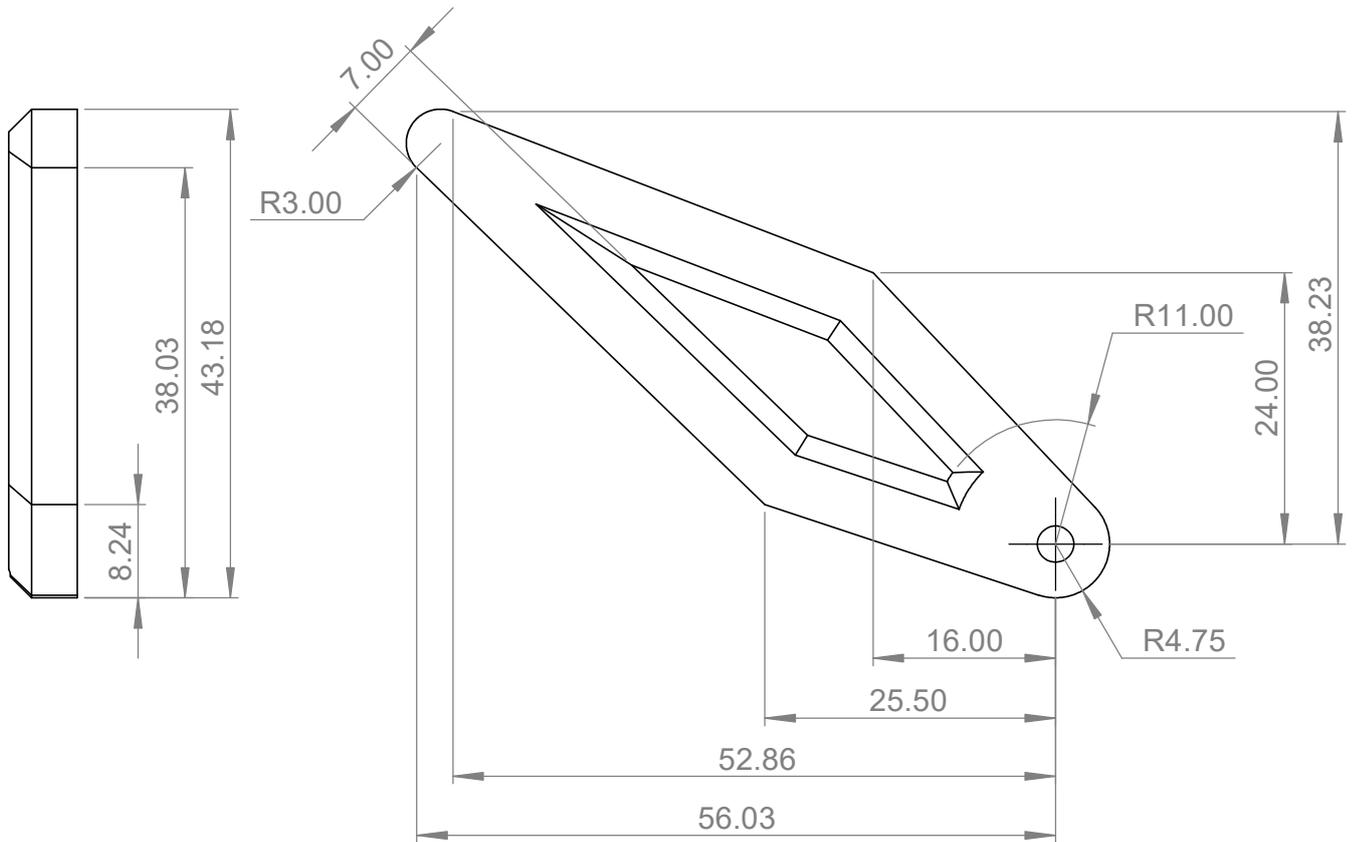
06/10/2024

06/10/2024

23/10/2024

23/10/2024

NOTA:
Chafilanes interno y externo de 2mm.



PROYECTO: Robot modular RB01

PARTE: Antena izquierda

CÓDIGO: RB01_00_08

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

06/10/2024

HOJA

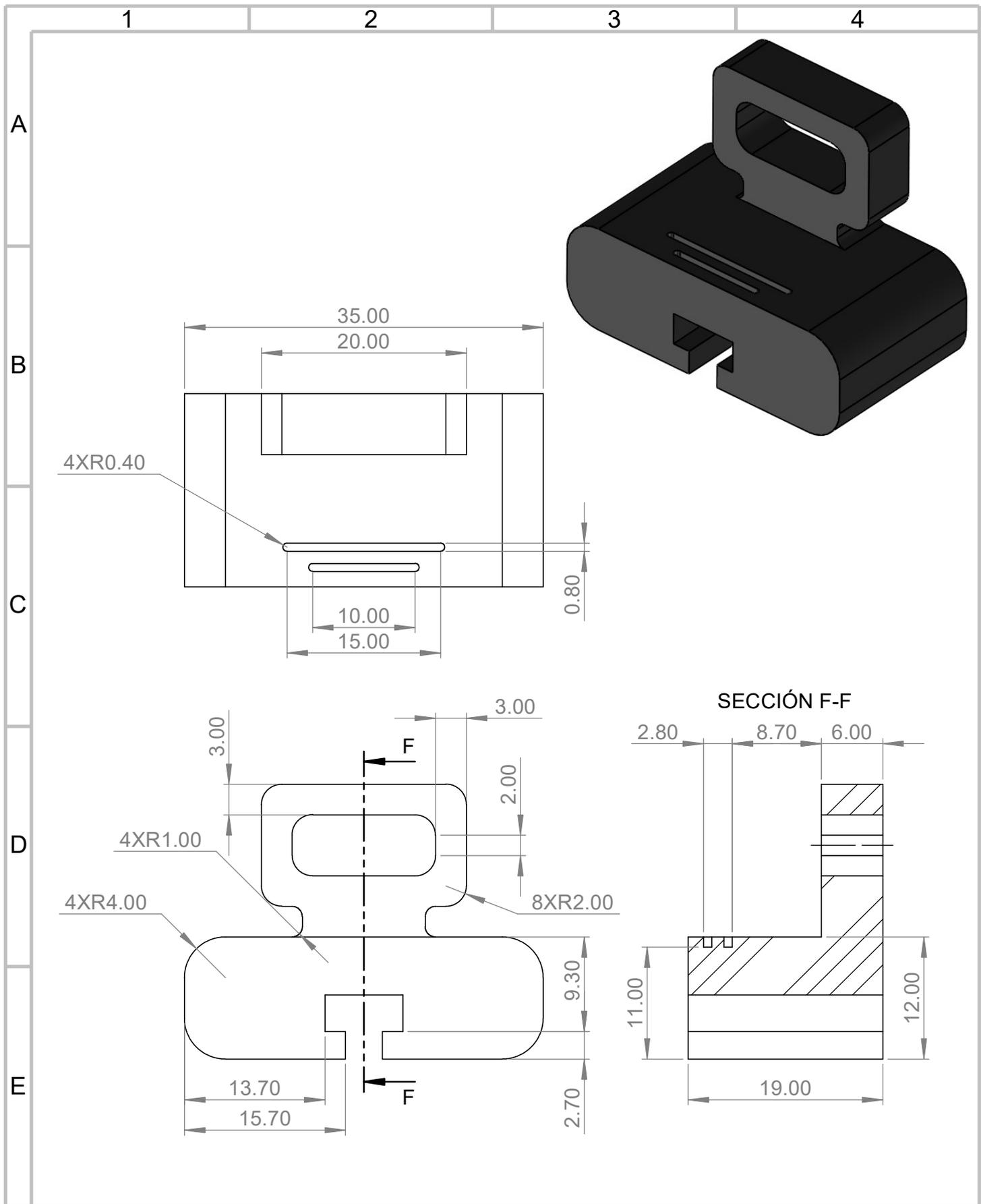
06/10/2024

1 / 1

23/10/2024

A4

23/10/2024



PROYECTO: Robot modular RB01

ESCALA

PARTE: Tapa guía posterior

2:1



CÓDIGO: RB01_00_09

SUSTITUCIÓN: Ninguno

TOLERANCIA: ± 0.02

MATERIAL:

PETG

DISEÑO:

Túquerres J.

06/10/2024

HOJA

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

06/10/2024

1 / 1

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

23/10/2024

A4

CANTIDAD:

1

APROBÓ:

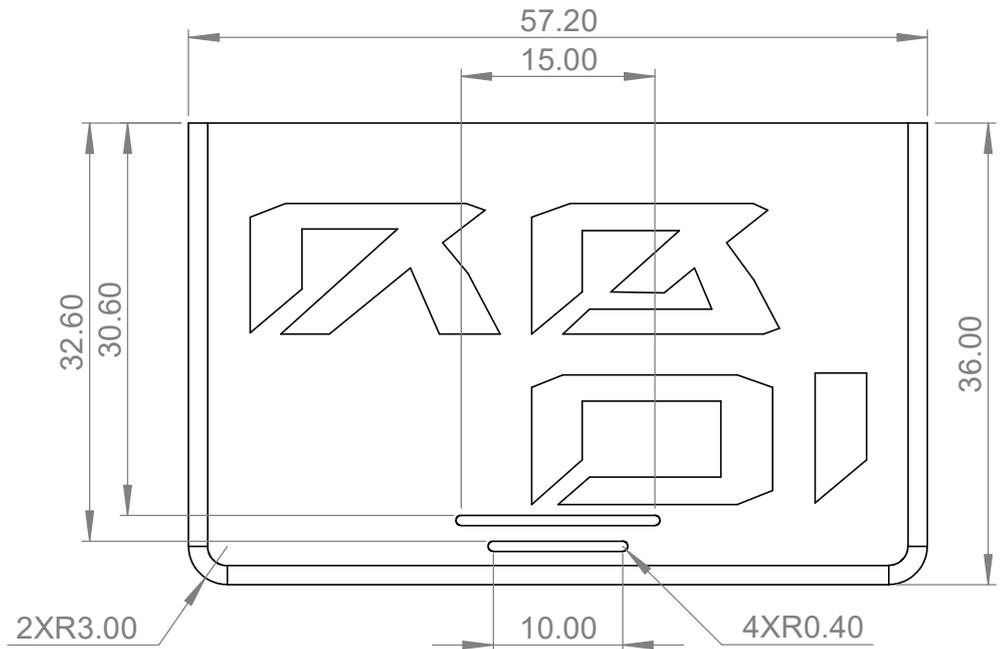
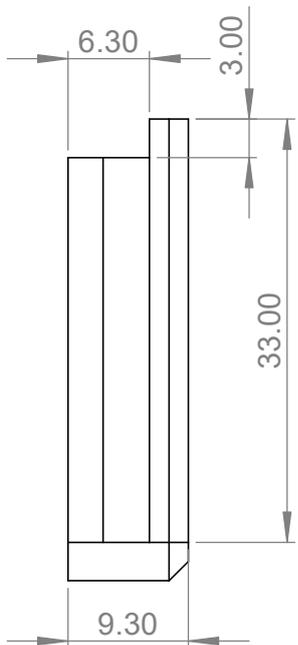
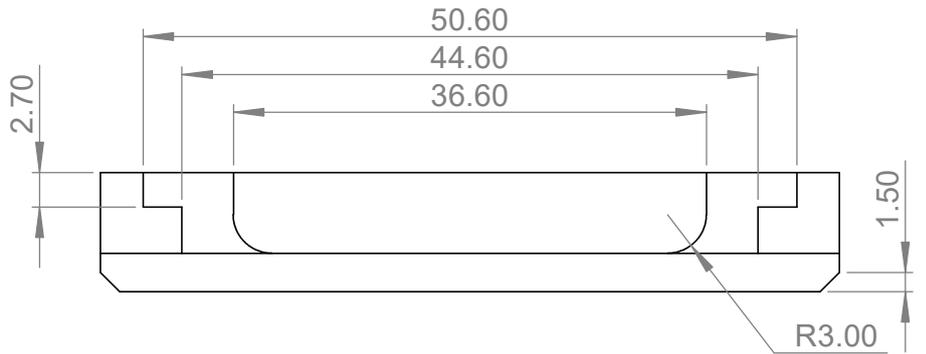
Túquerres J.

23/10/2024

UTN

CIME

NOTA:
Tipografía tipo futurista de 10.20mm.



PROYECTO: Robot modular RB01

PARTE: Tapa guías frontales

CÓDIGO: RB01_00_10

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

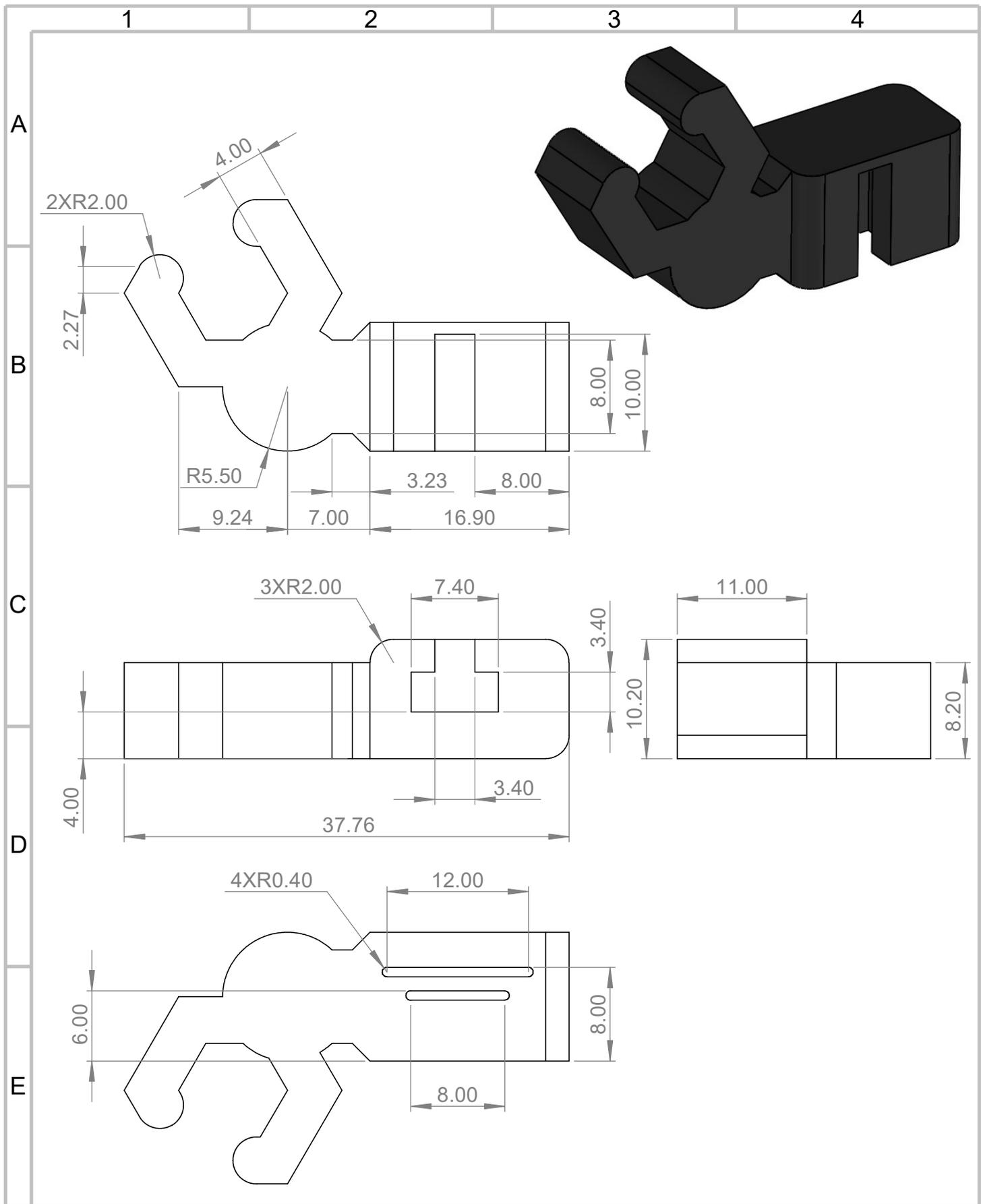
A4

06/10/2024

06/10/2024

23/10/2024

23/10/2024



PROYECTO: Robot modular RB01

ESCALA

PARTE: Brazo derecho

1:1



CÓDIGO: RB01_00_11

SUSTITUCIÓN: Ninguno

TOLERANCIA: ± 0.02

MATERIAL:

PETG

DISEÑO:

Túquerres J.

06/10/2024

HOJA

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

06/10/2024

1 / 1

UTN

CIME

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

23/10/2024

A4

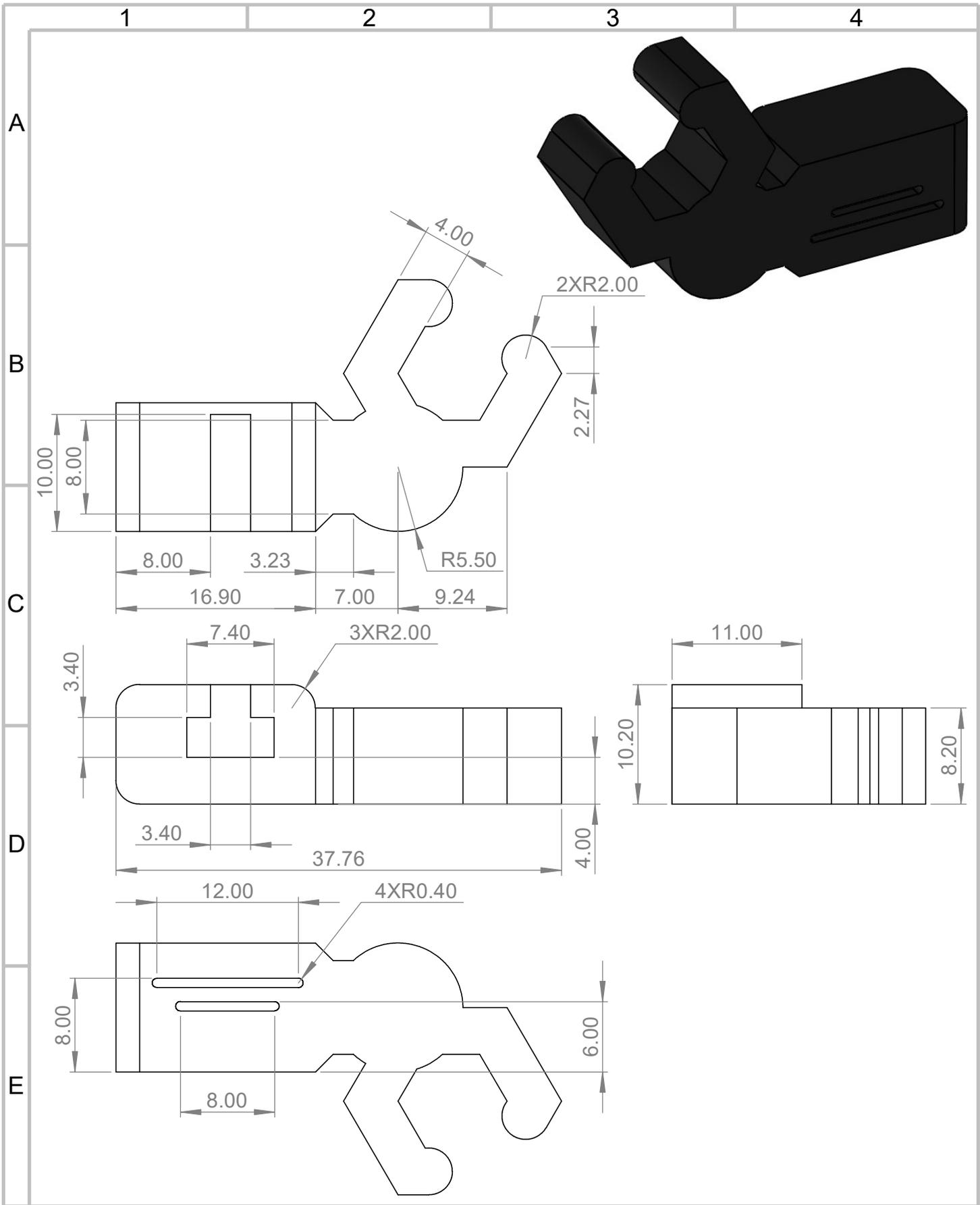
CANTIDAD:

1

APROBÓ:

Túquerres J.

23/10/2024



PROYECTO: Robot modular RB01

PARTE: Brazo izquierdo

CÓDIGO: RB01_00_12

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

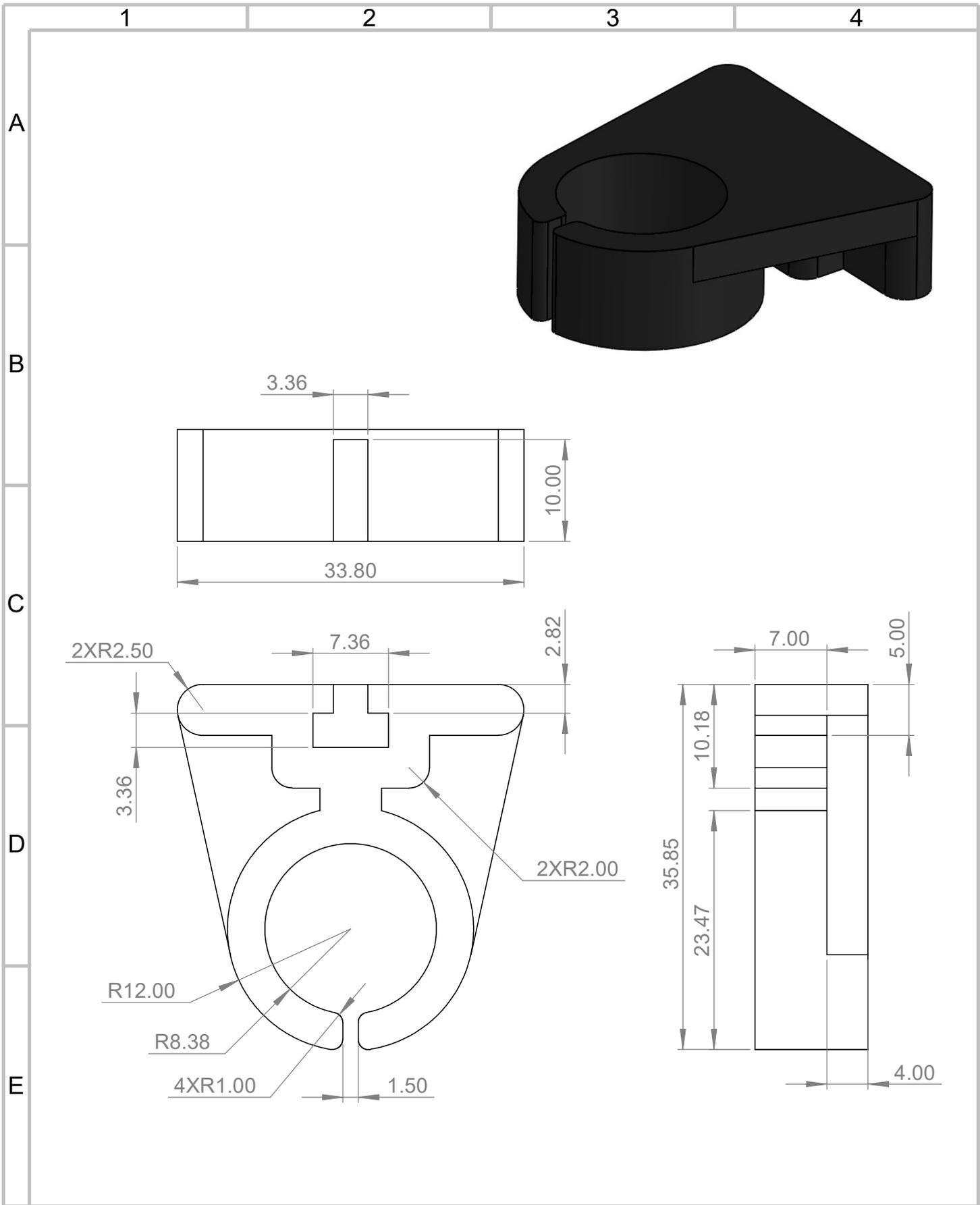
A4

06/10/2024

06/10/2024

23/10/2024

23/10/2024



Ingeniería
Mecatrónica

PROYECTO: Robot modular RB01

PARTE: Base de marcador

CÓDIGO: RB01_00_13

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

2

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

08/10/2024

HOJA

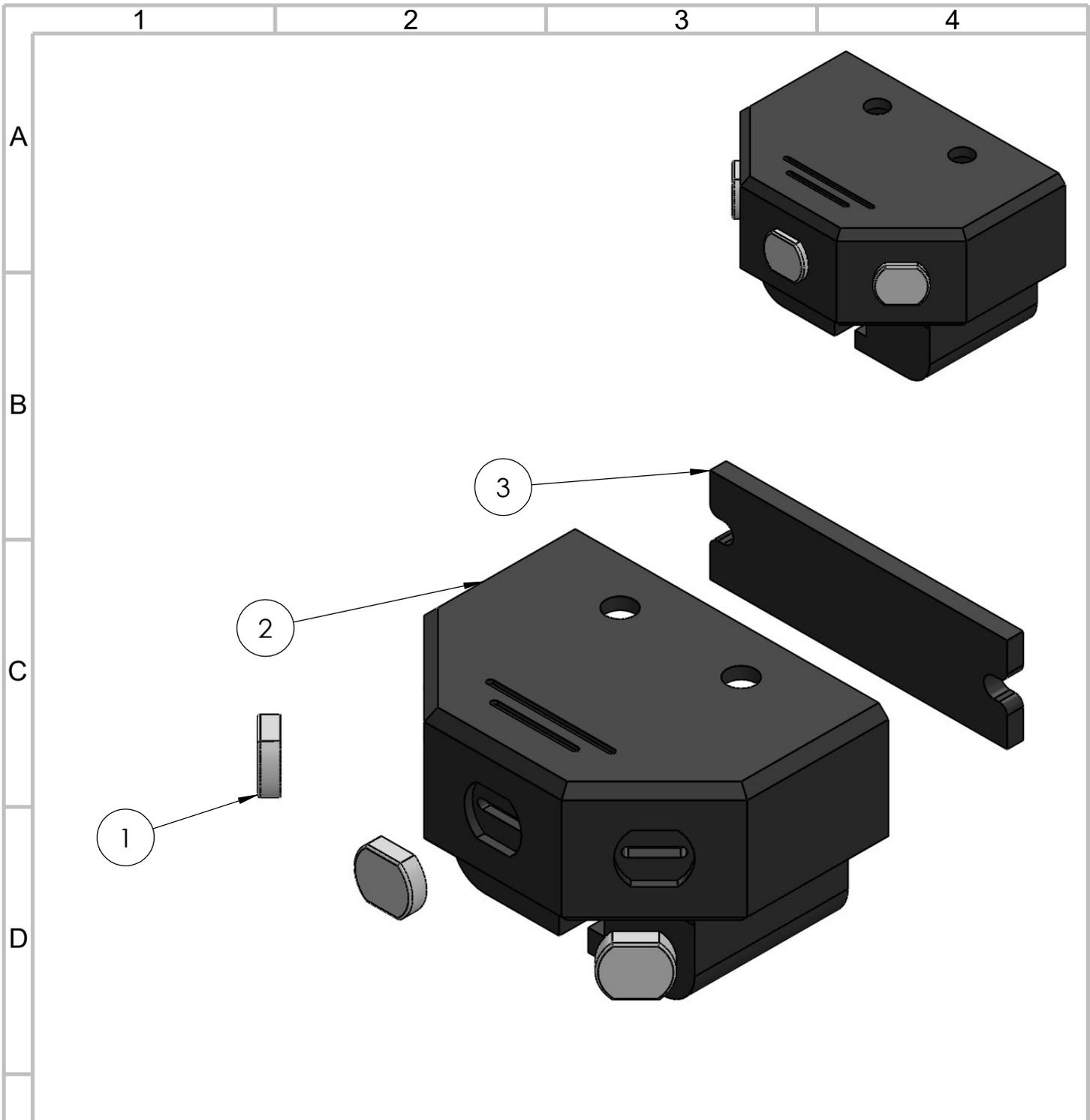
08/10/2024

1 / 1

23/10/2024

A4

23/10/2024



NOTA:
Todas las uniones se conectan mediante tornillos para plástico M2x8mm.

No. de pieza	Denominación	No. de Norma/Dibujo	Material	Cantidad	Observaciones
3	Tapa de sensor de luz.	RB01_01_02	PETG	1	Ninguna
2	Base de sensor de luz.	RB01_01_01	PETG	1	Ninguna
1	Fotorresistencia.	-	-	3	Ninguna

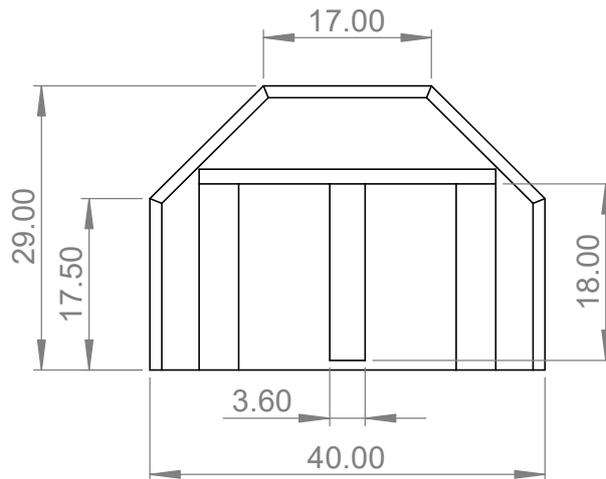
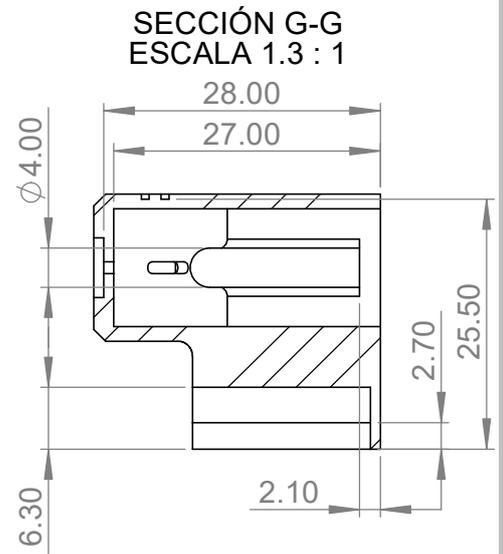
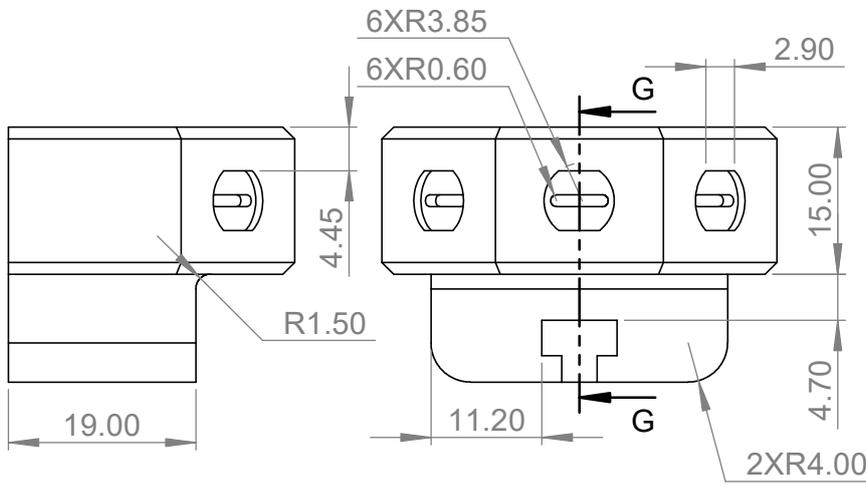
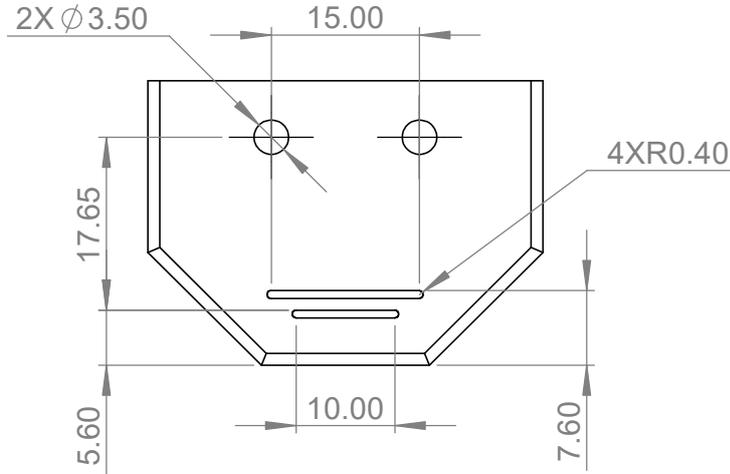
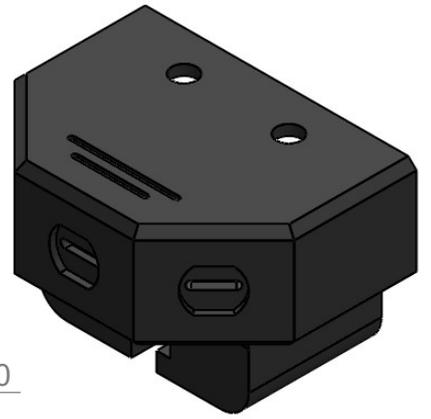
TOLERANCIA
± 0,2

MASA
16.47 g

Plano de ensamblaje

			PROYECTO: Módulo seguidor de luz	ESCALA 2:1
DIBUJO:	FECHA: 23/10/24	NOMBRE: Túquerres J.		
REVISO:	FECHA: 23/10/24	NOMBRE: Túquerres J.		
APROBÓ:	FECHA: 23/10/24	NOMBRE: Túquerres J.		
			CÓDIGO: RB01_01_00	

NOTA:
Chafanes externos de 1.20mm.



PROYECTO: Módulo seguidor de luz

ESCALA

PARTE: Base de sensor de luz

1:1

CÓDIGO: RB01_01_01

SUSTITUCIÓN: Ninguno

TOLERANCIA: ± 0.02

MATERIAL:

PETG

DISEÑO:

Túquerres J.

10/10/2024

HOJA

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

10/10/2024

1 / 1

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

23/10/2024

A4

CANTIDAD:

1

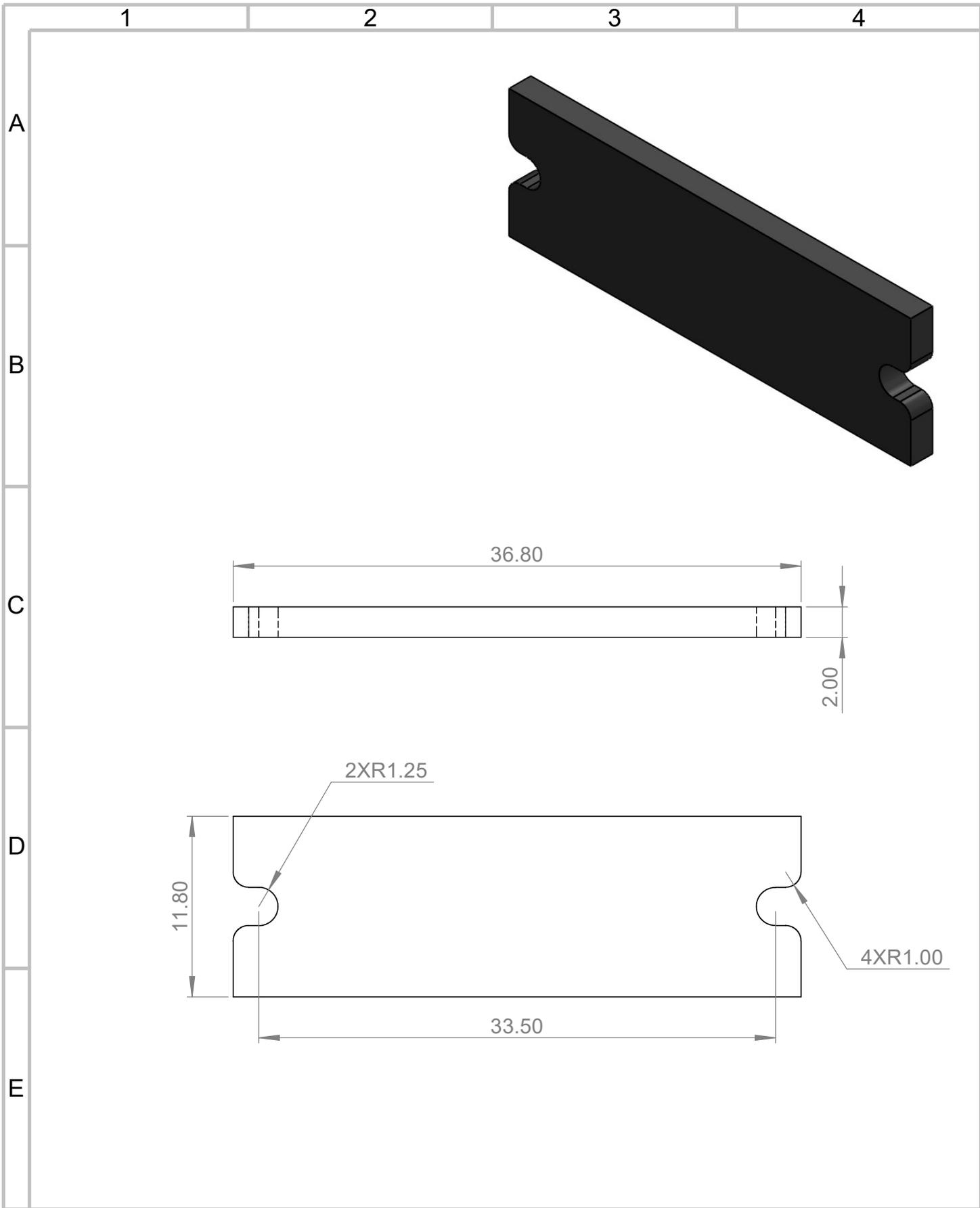
APROBÓ:

Túquerres J.

23/10/2024

UTN

CIME



Ingeniería
Mecatrónica

PROYECTO: Módulo seguidor de luz

PARTE: Tapa de sensor de luz

CÓDIGO: RB01_01_02

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

10/10/2024

HOJA

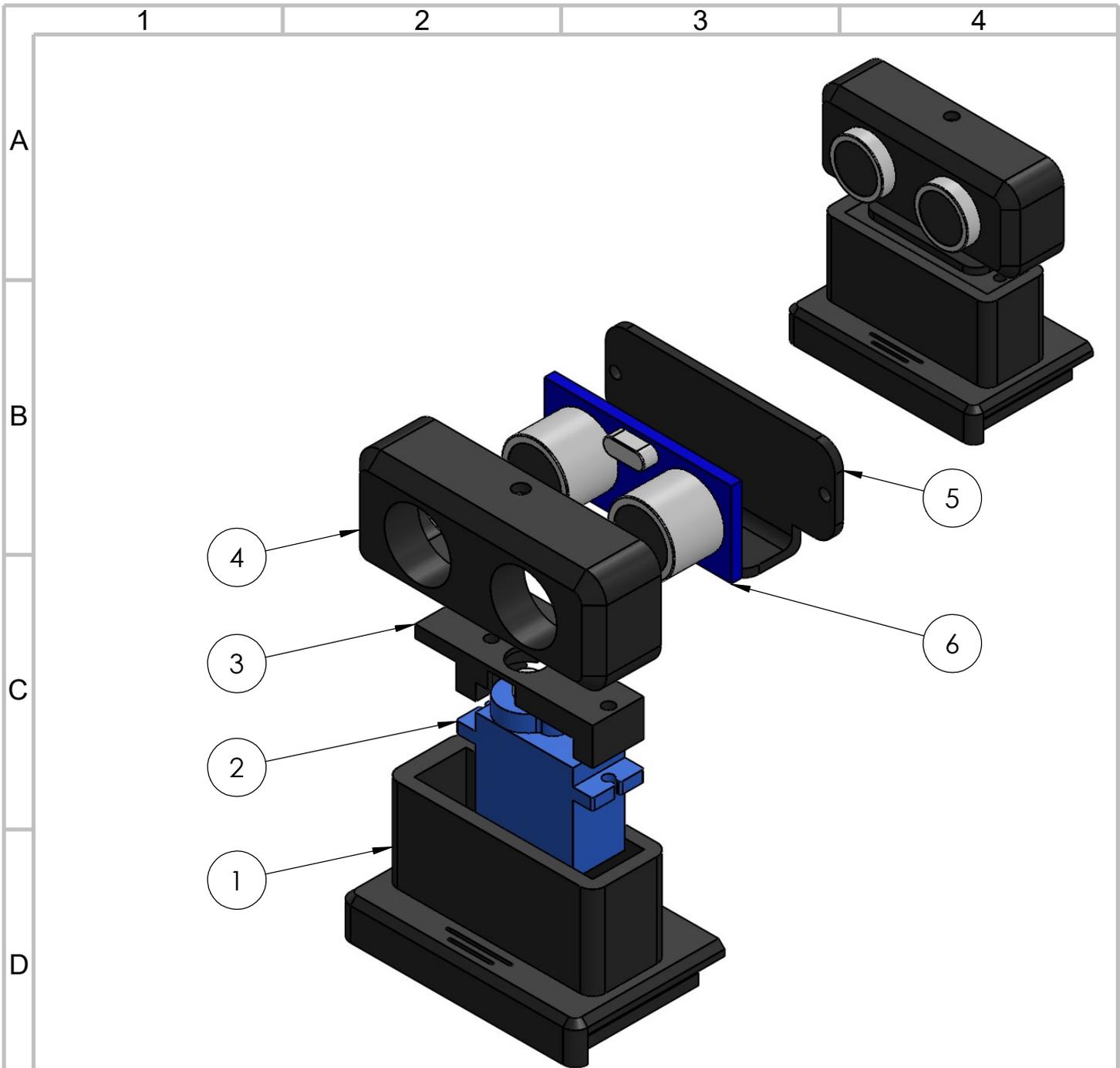
10/10/2024

1 / 1

23/10/2024

A4

23/10/2024



NOTA:
Todas las uniones se conectan mediante tornillos para plástico M2x8mm.

6	Sensor ultrasónico.	-	-	1	Ninguna
5	Tapa de carcasa de sensor ultrasónico.	RB01_02_04	PETG	1	Ninguna
4	Carcasa de sensor ultrasónico.	RB01_02_03	PETG	1	Ninguna
3	Tapa de base rotatoria.	RB01_02_02	PETG	1	Ninguna
2	Servomotor 180 grados.	-	-	1	Ninguna
1	Base rotatoria.	RB01_02_01	PETG	1	Ninguna
No. de piez.	Denominación	No. de Norma/Dibujo	Material	Cantidad	Observaciones

TOLERANCIA
 $\pm 0,2$

MASA
68.77 g

Plano de ensamblaje

	FECHA	NOMBRE
DIBUJO:	23/10/24	Túquerres J.
REVISO:	23/10/24	Túquerres J.
APROBÓ:	23/10/24	Túquerres J.

PROYECTO: Módulo evasor de objetos

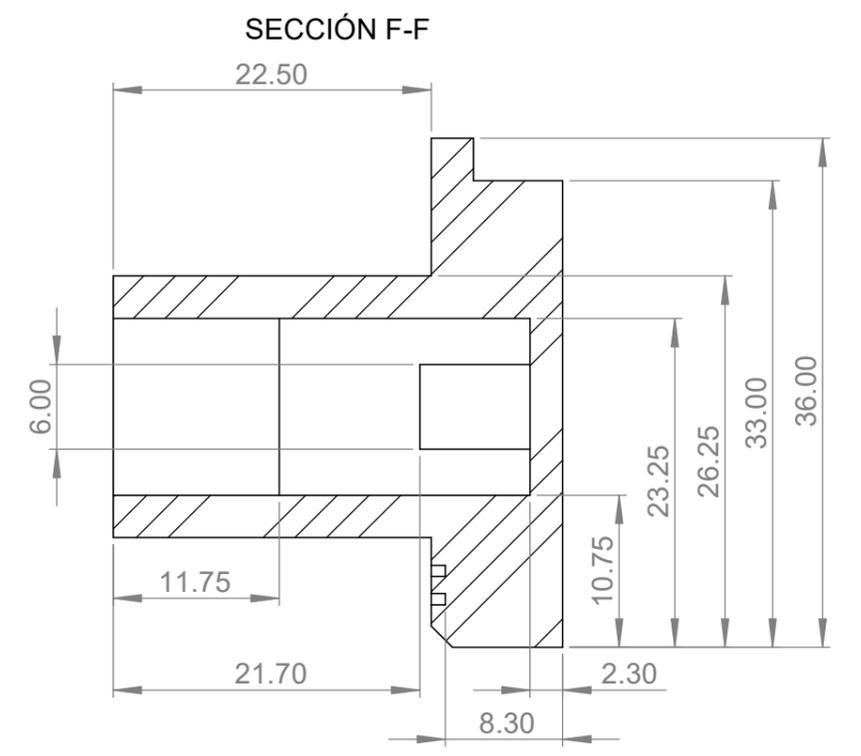
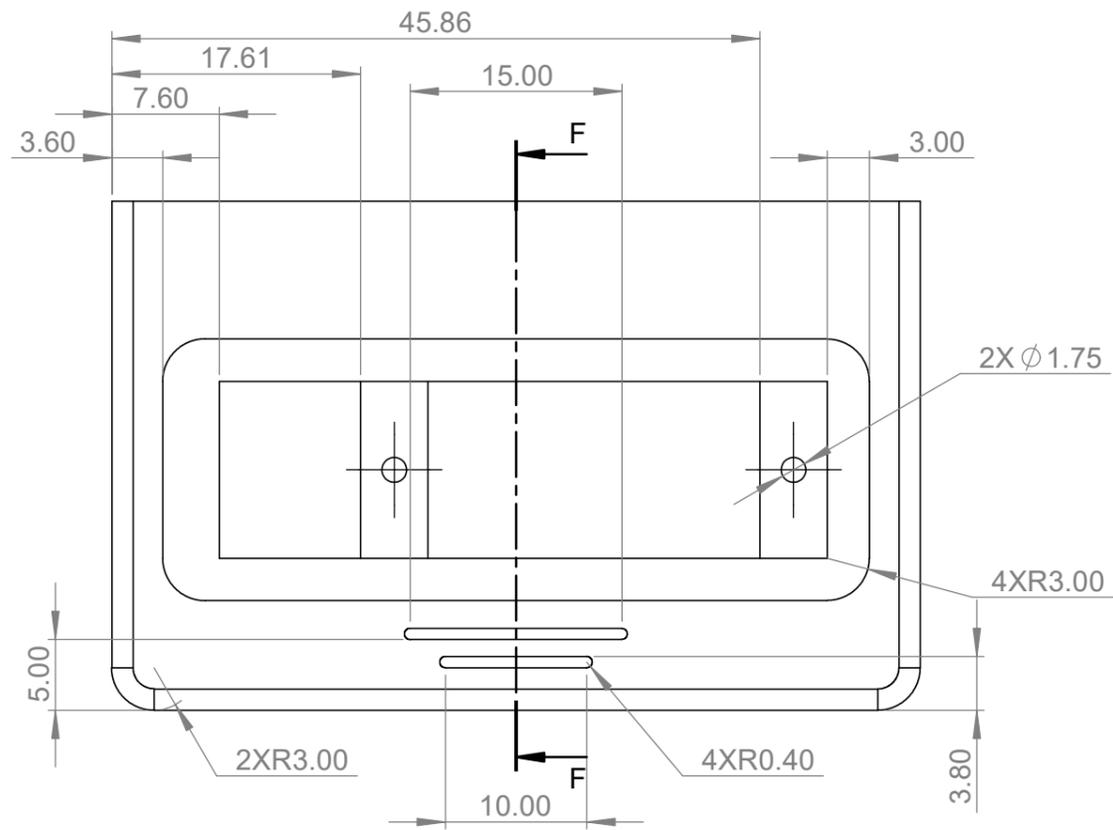
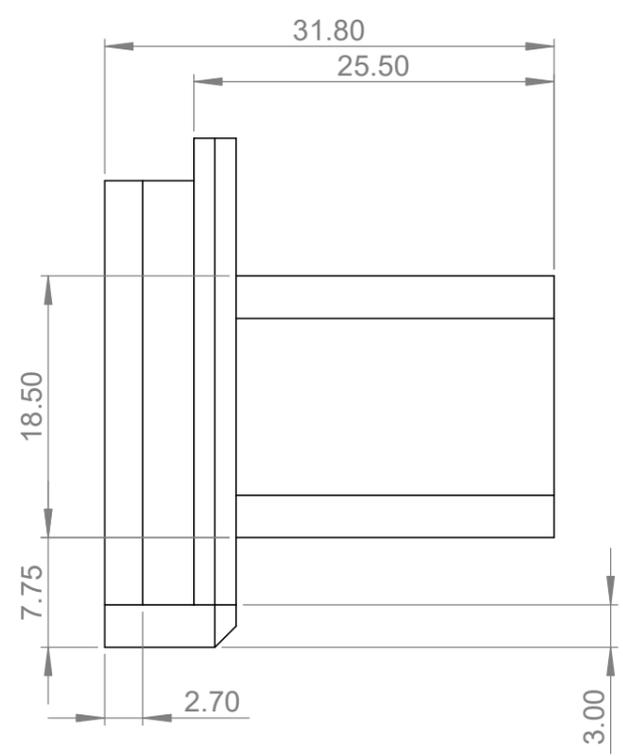
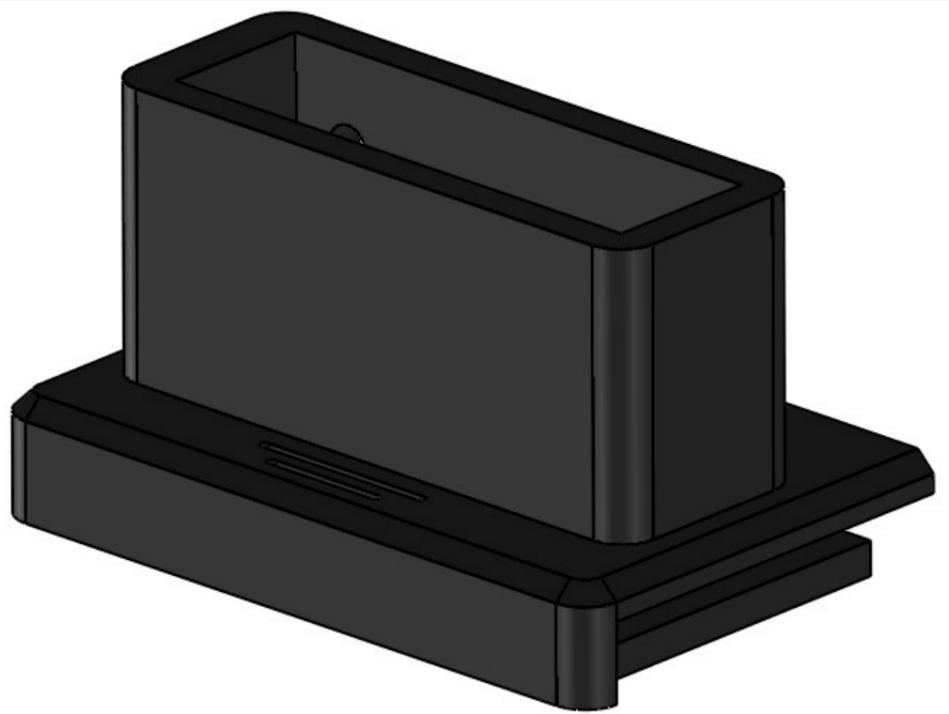
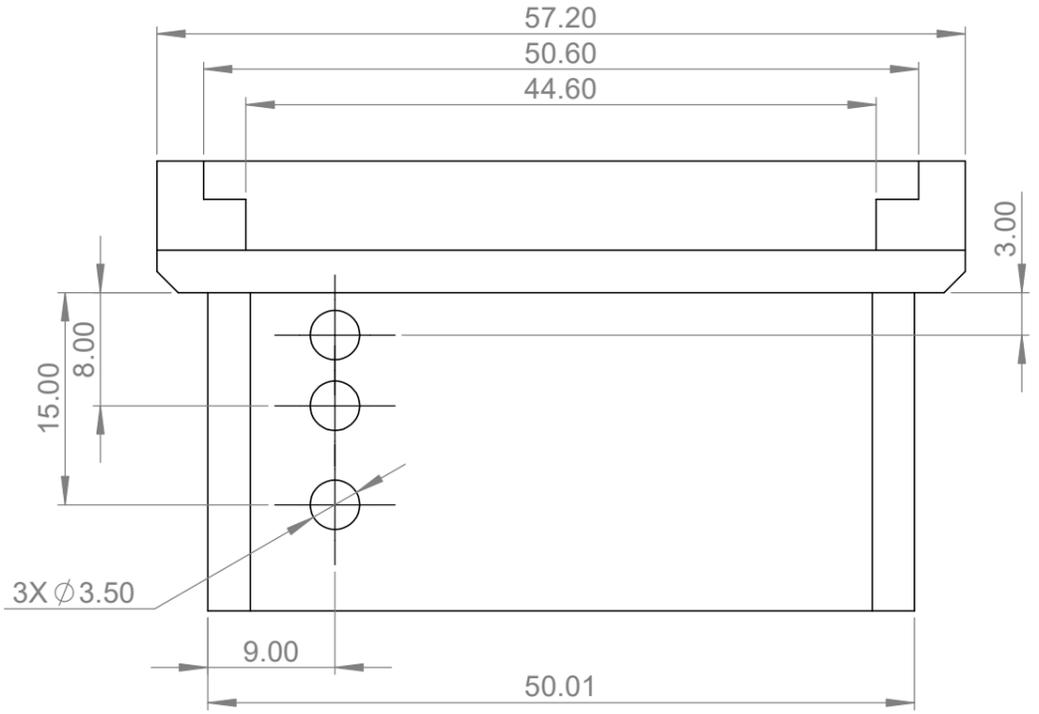
ESCALA
1:1



CÓDIGO: RB01_02_00

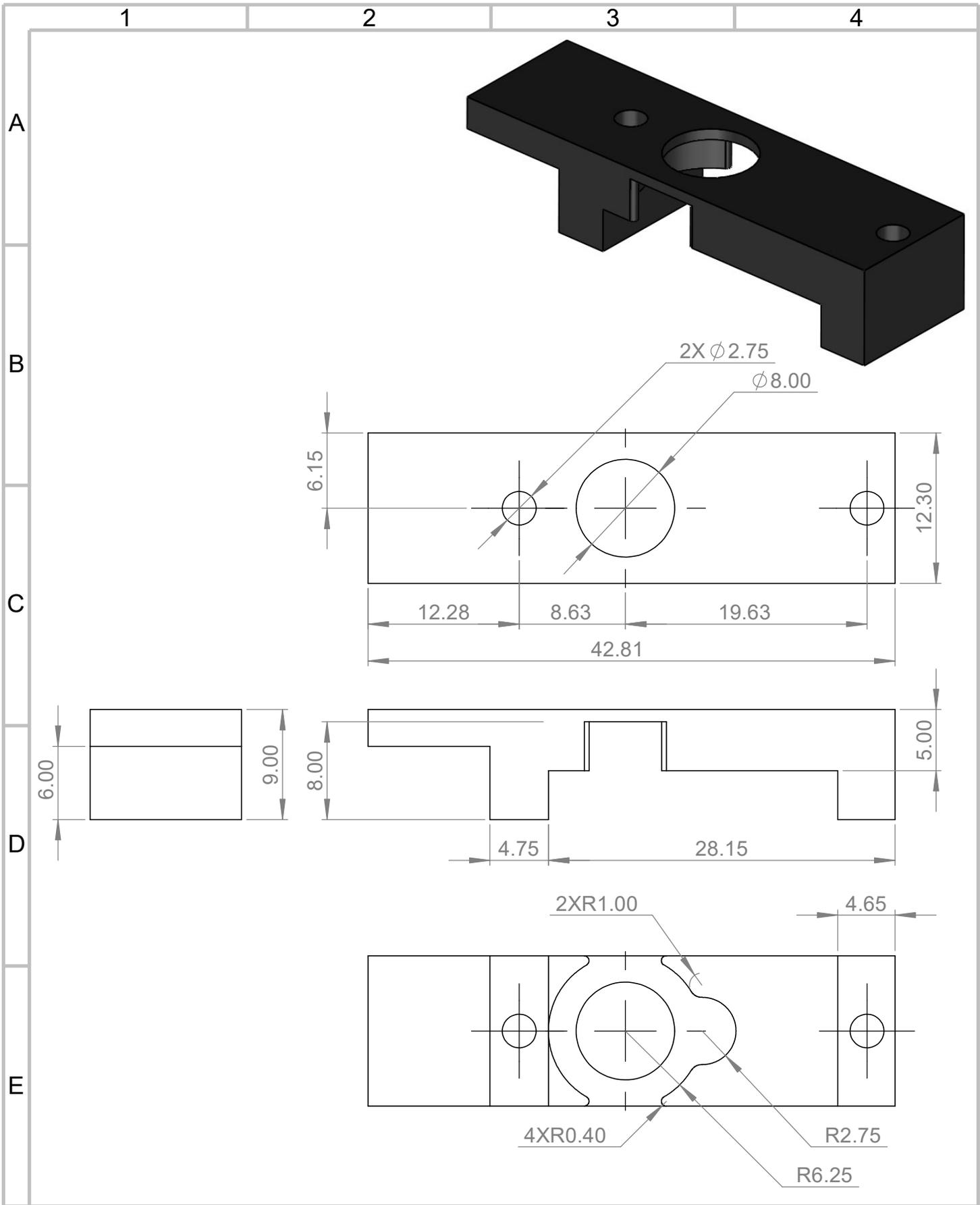


NOTA:
Chafilán externo de 1.50mm.



UTN CIME

PROYECTO: Módulo evasor de objetos				ESCALA	
PARTE: Base rotatoria				2:1	
CÓDIGO: RB01_02_01				TOLERANCIA: ± 0.02	
SUSTITUCIÓN: Ninguno				HOJA	
MATERIAL:	PETG	DISEÑO:	Túquerres J.	12/10/2024	1 / 1
TRATAMIENTO:	Ninguno	DIBUJO:	Túquerres J.	12/10/2024	A3
RECUBRIMIENTO:	Ninguno	REVISÓ:	Túquerres J.	23/10/2024	
CANTIDAD:	1	APROBÓ:	Túquerres J.	23/10/2024	



Ingeniería
Mecatrónica

PROYECTO: Módulo evasor de objetos

PARTE: Tapa de base rotatoria

CÓDIGO: RB01_02_02

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

A4

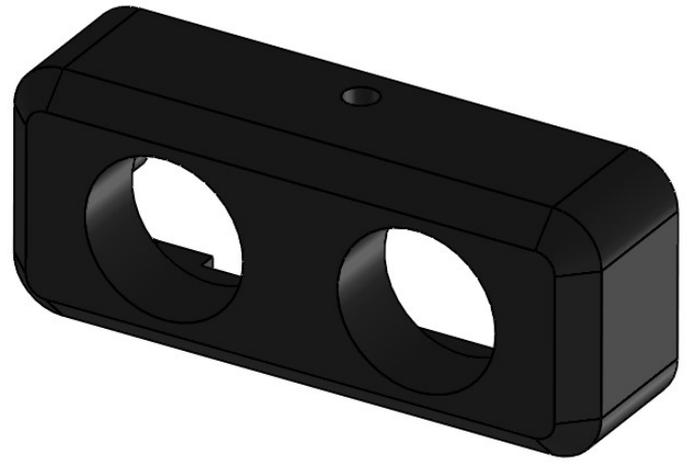
12/10/2024

12/10/2024

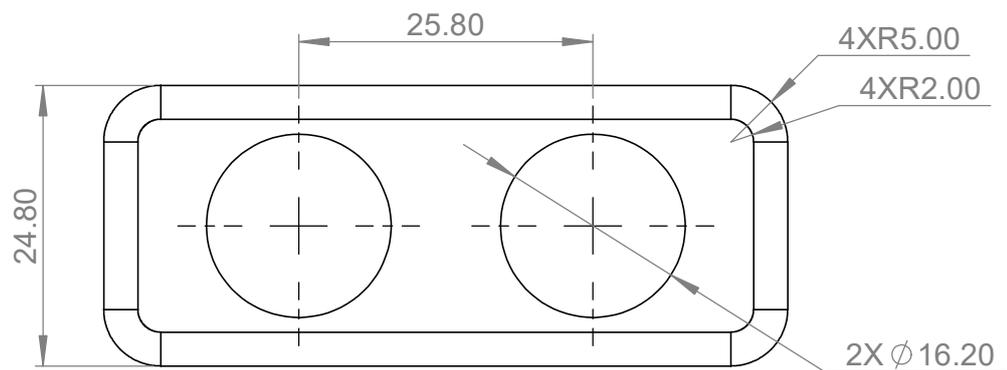
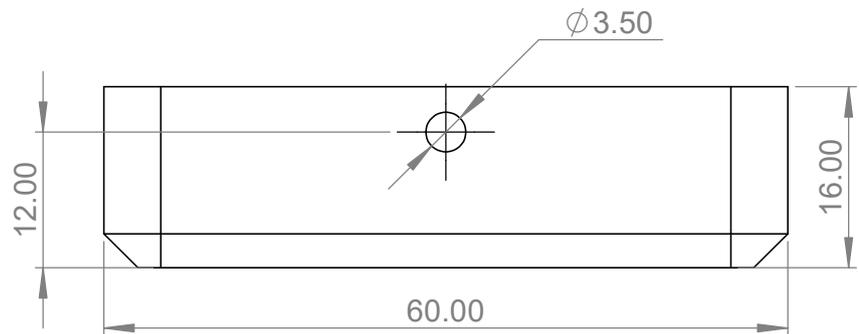
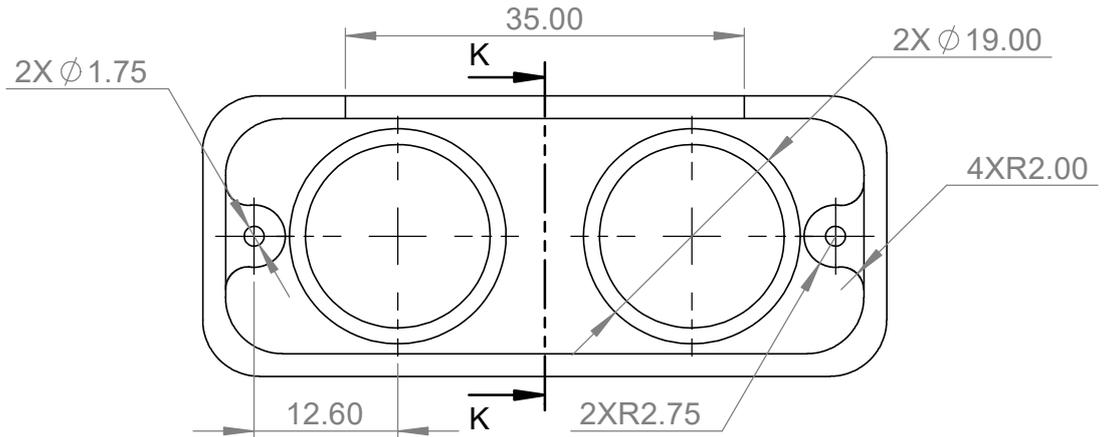
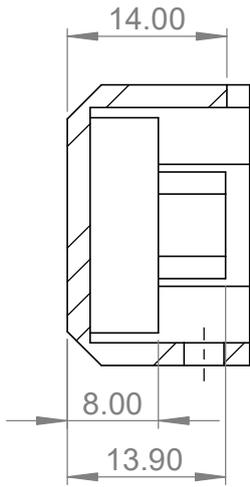
23/10/2024

23/10/2024

NOTA:
 Espesor de la carcasa de 2mm.
 Chaflán externo de 3mm.



SECCIÓN K-K
 ESCALA 1.5 : 1



PROYECTO: Módulo evasor de objetos

PARTE: Carcasa de sensor ultrasónico

CÓDIGO: RB01_02_03

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

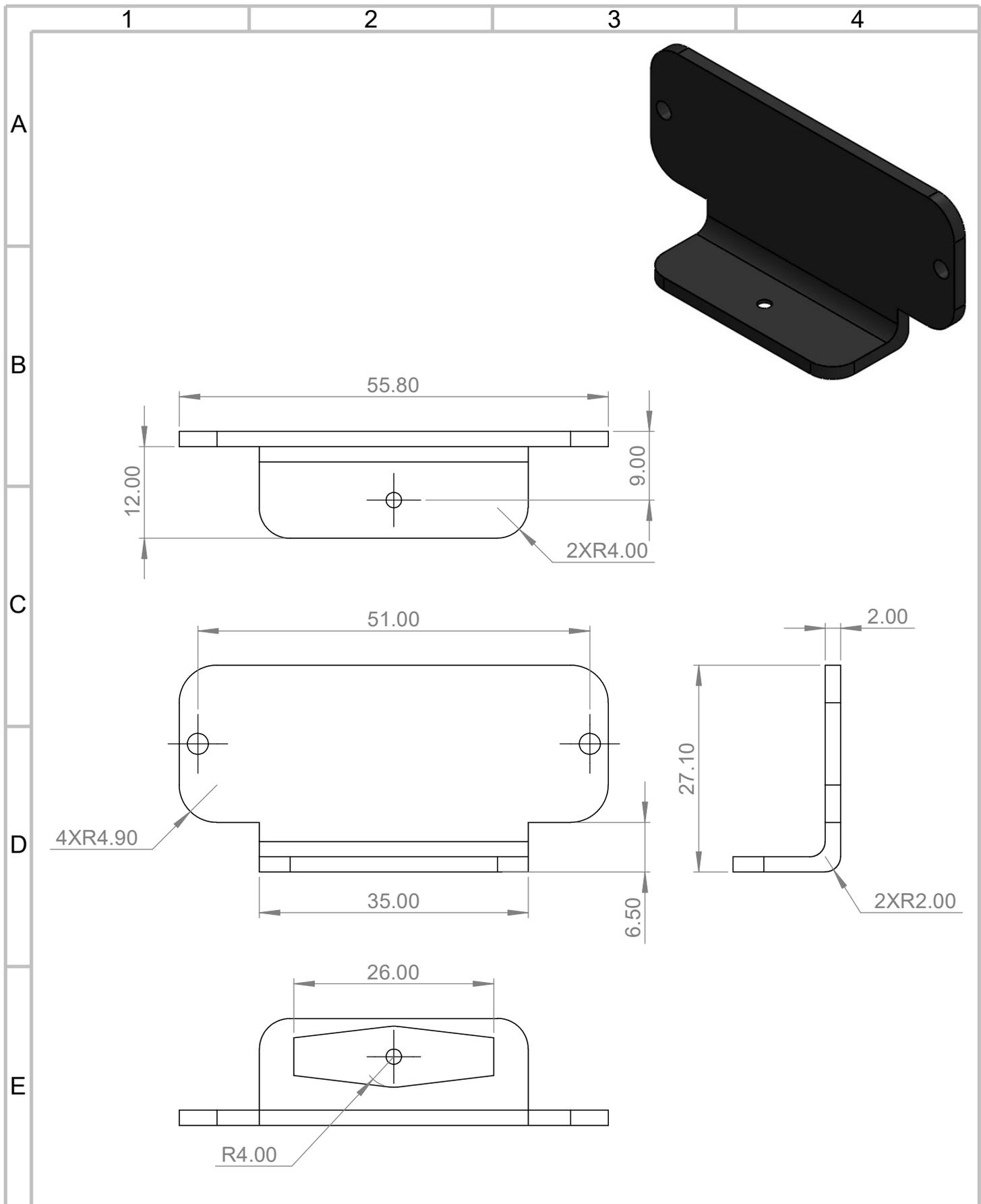
A4

12/10/2024

12/10/2024

23/10/2024

23/10/2024



PROYECTO: Módulo evasor de objetos
 PARTE: Tapa de carcasa de sensor ultrasónico
 CÓDIGO: RB01_02_04
 SUSTITUCIÓN: Ninguno

ESCALA
1:1



TOLERANCIA: ± 0.02

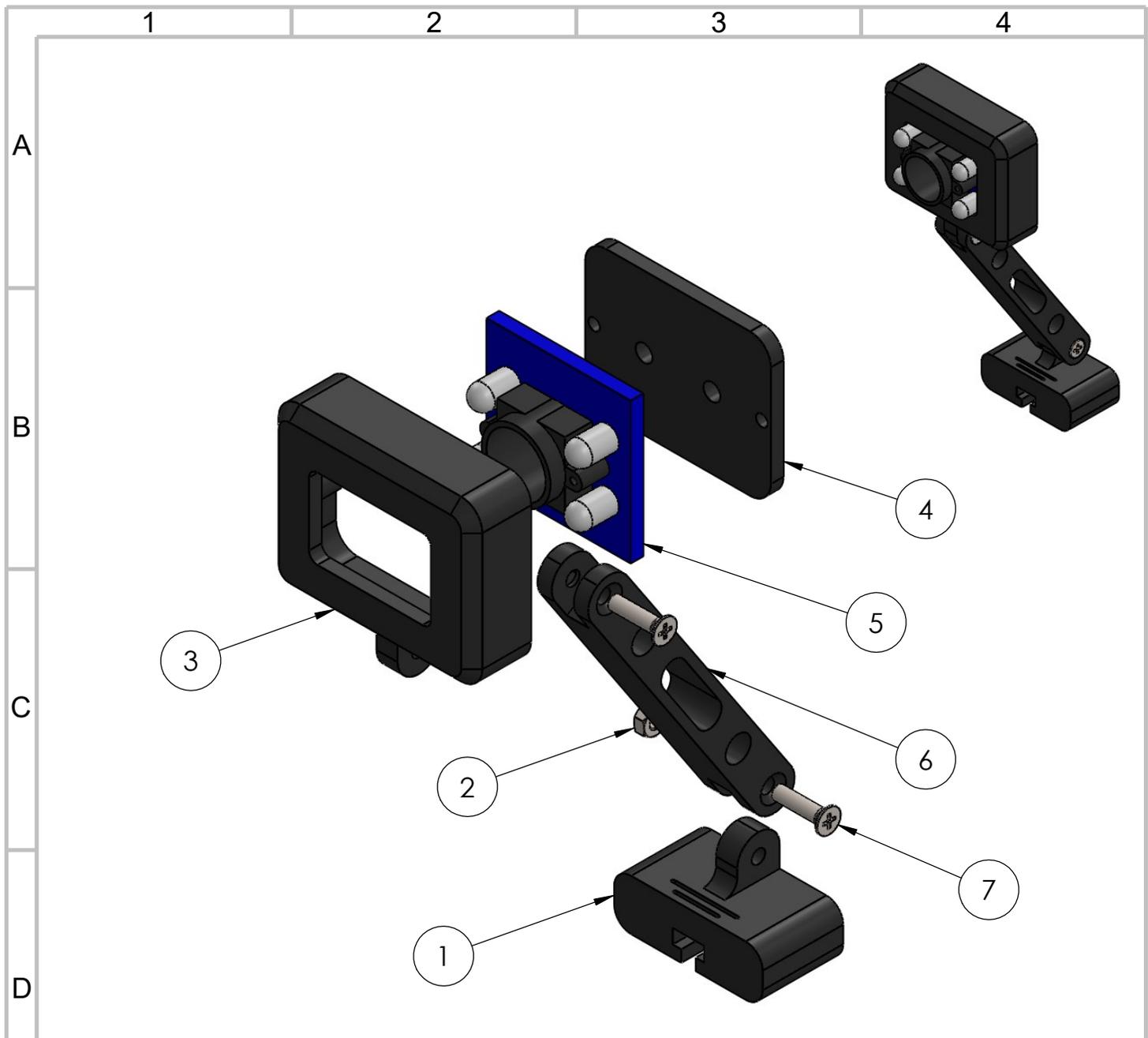
MATERIAL:	PETG	DISEÑO:	Túquerres J.
TRATAMIENTO:	Ninguno	DIBUJO:	Túquerres J.
RECUBRIMIENTO:	Ninguno	REVISÓ:	Túquerres J.
CANTIDAD:	1	APROBÓ:	Túquerres J.

12/10/2024
12/10/2024
23/10/2024
23/10/2024

HOJA
1 / 1
A4

UTN

CIME



NOTA:
Las uniones de la tapa se conectan mediante tornillos para plástico M2x8mm.

7	Perno avellanado M3x12mm.	-	-	2	Ninguna
6	Soporte de sensor de color.	RB01_03_02	PETG	1	Ninguna
5	Sensor de color.	-	-	1	Ninguna
4	Tapa de carcasa de sensor de color.	RB01_03_04	PETG	1	Ninguna
3	Carcasa de sensor de color.	RB01_03_03	PETG	1	Ninguna
2	Tuerca hexagonal M3.	-	-	2	Ninguna
1	Base de sensor de color.	RB01_03_01	PETG	1	Ninguna
No. de pieza.	Denominación	No. de Norma/Dibujo	Material	Cantidad	Observaciones

TOLERANCIA
 $\pm 0,2$

MASA
41.76 g

Plano de ensamblaje

	FECHA	NOMBRE
DIBUJO:	23/10/24	Túquerres J.
REVISO:	23/10/24	Túquerres J.
APROBÓ:	23/10/24	Túquerres J.

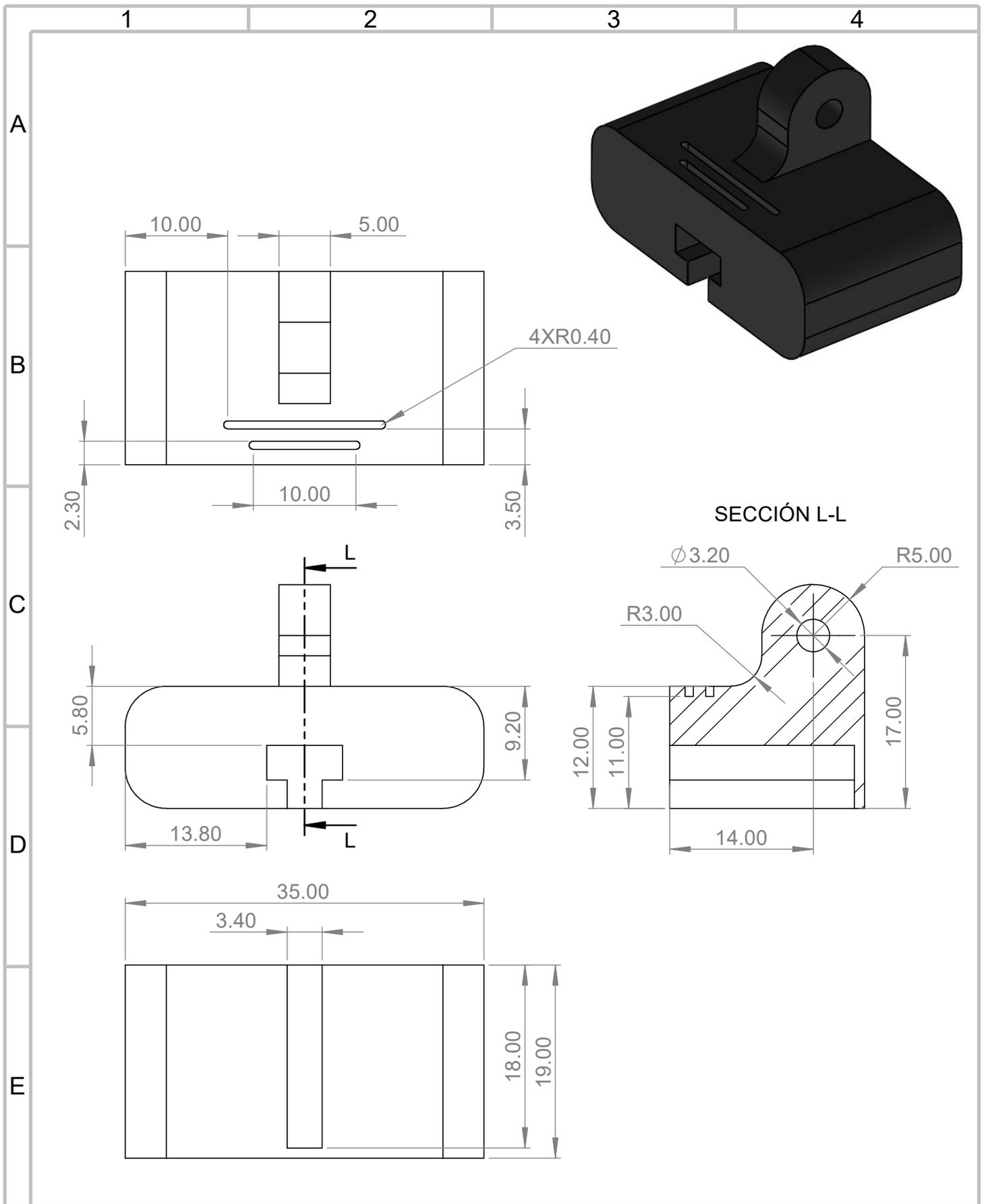
PROYECTO: Módulo detector de color

ESCALA
1:1



CÓDIGO: RB01_03_00





Ingeniería
Mecatrónica

PROYECTO: Módulo detector de color

PARTE: Base de sensor de color

CÓDIGO: RB01_03_01

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

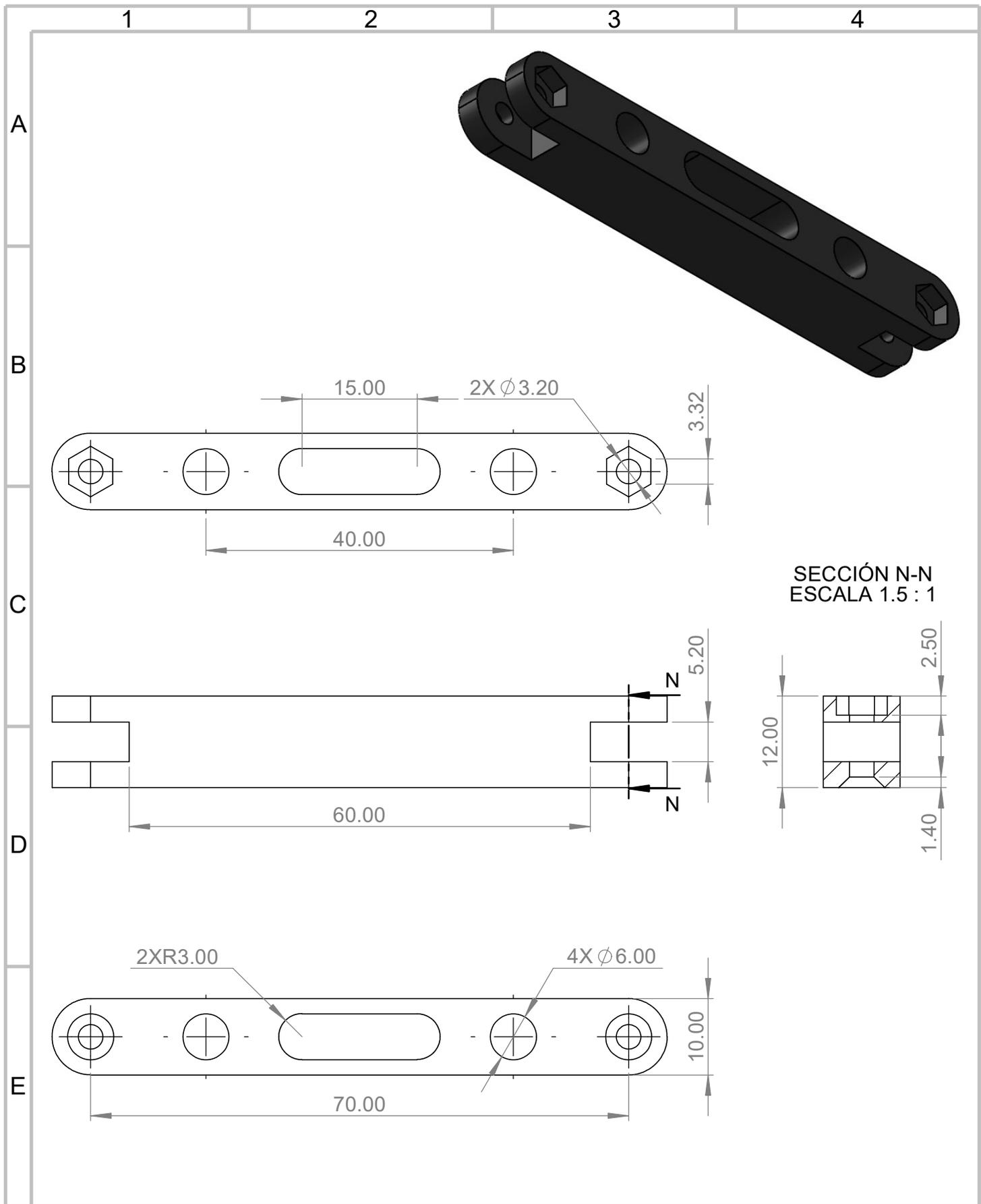
CIME

HOJA

1 / 1

A4

14/10/2024
14/10/2024
23/10/2024
23/10/2024



SECCIÓN N-N
ESCALA 1.5 : 1



Ingeniería
Mecatrónica

PROYECTO: Módulo detector de color

PARTE: Soporte de sensor de color

CÓDIGO: RB01_03_02

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

14/10/2024

HOJA

14/10/2024

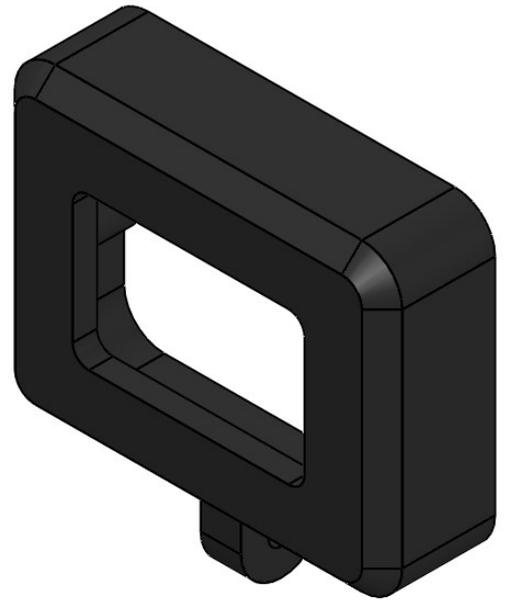
1 / 1

23/10/2024

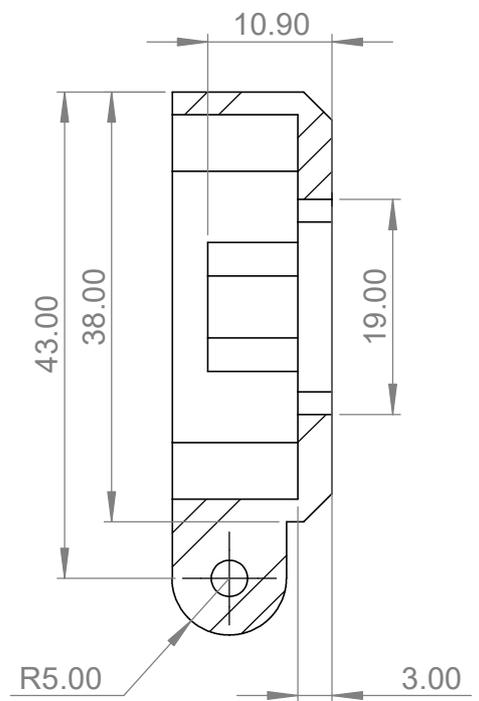
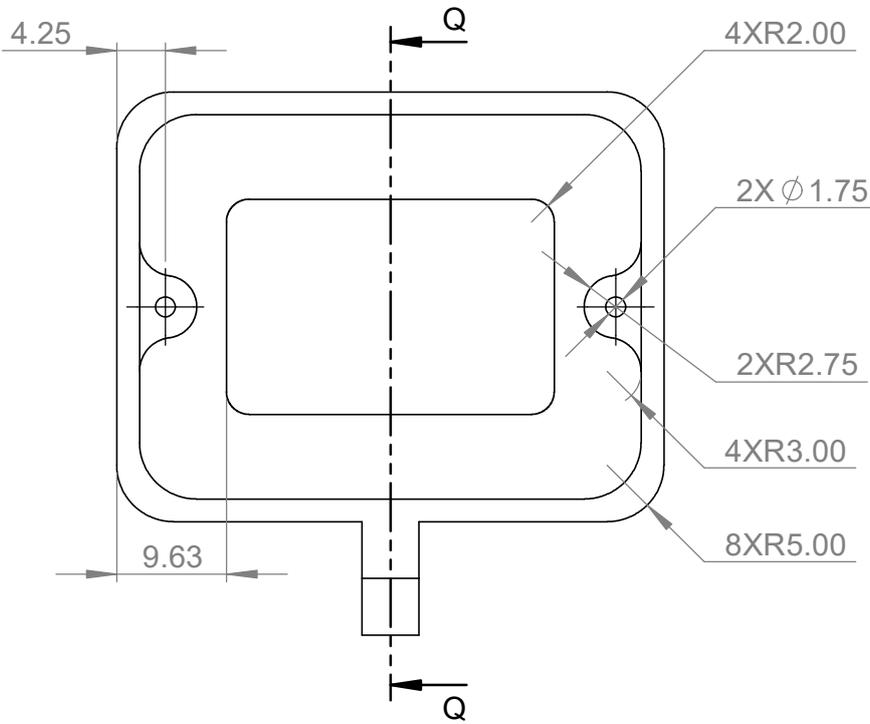
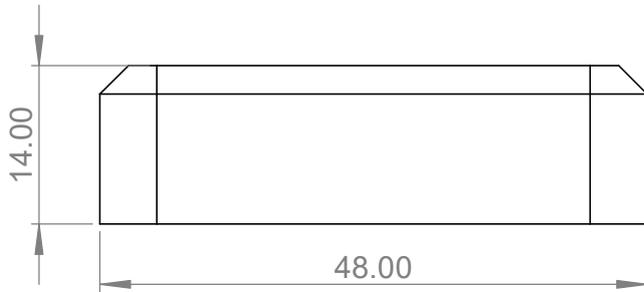
A4

23/10/2024

NOTA:
 Espesor de la carcasa de 2mm.
 Chaflán externo de 2.50mm.



SECCIÓN Q-Q
 ESCALA 1.5 : 1



PROYECTO: Módulo detector de color

ESCALA

PARTE: Carcasa de sensor de color

1:1



CÓDIGO: RB01_03_03

SUSTITUCIÓN: Ninguno

TOLERANCIA: ± 0.02

MATERIAL:

PETG

DISEÑO:

Túquerres J.

14/10/2024

HOJA

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

14/10/2024

1 / 1

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

23/10/2024

A4

CANTIDAD:

1

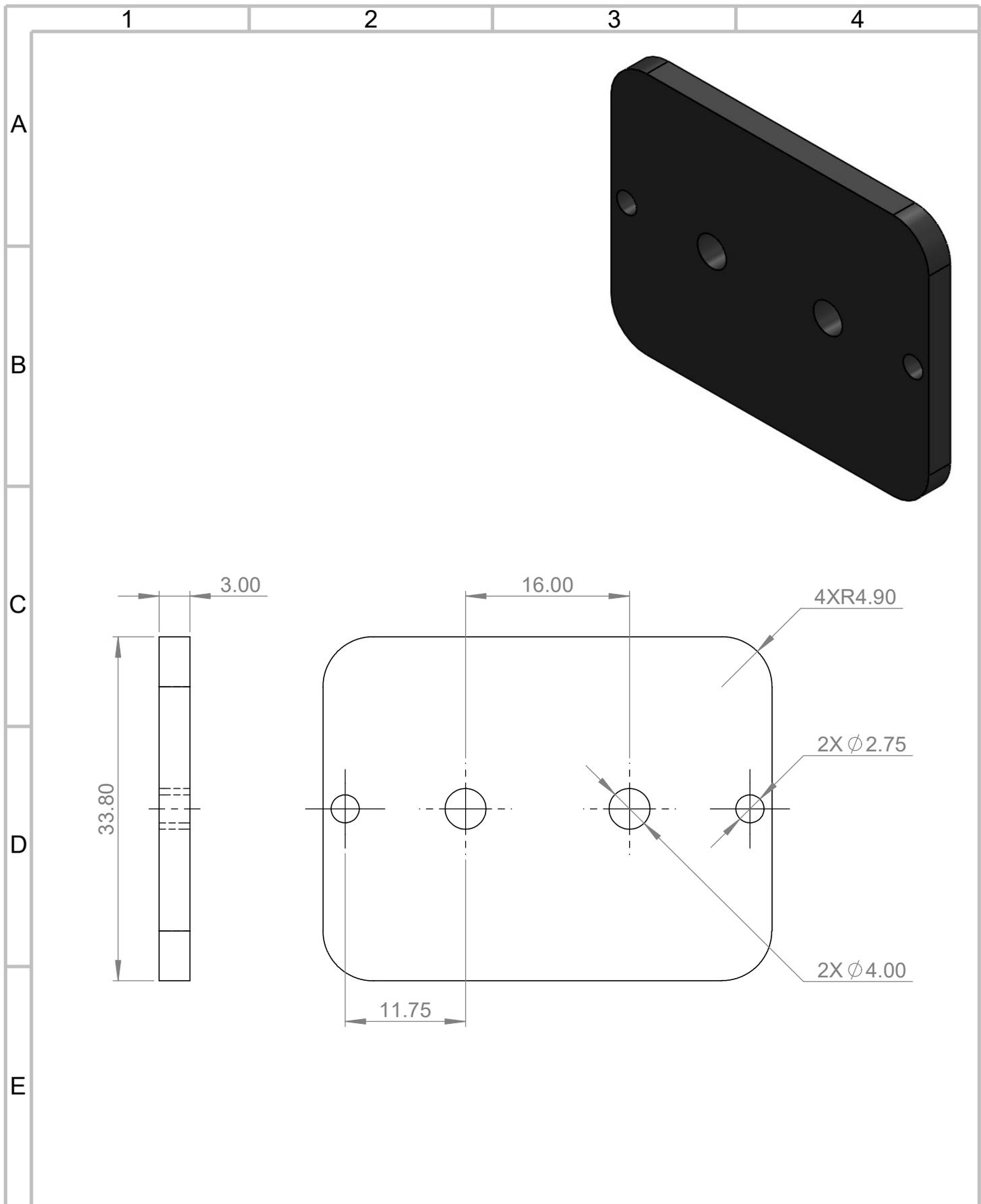
APROBÓ:

Túquerres J.

23/10/2024

UTN

CIME



PROYECTO: Módulo detector de color

PARTE: Tapa de carcasa de sensor de color

CÓDIGO: RB01_03_04

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

14/10/2024

HOJA

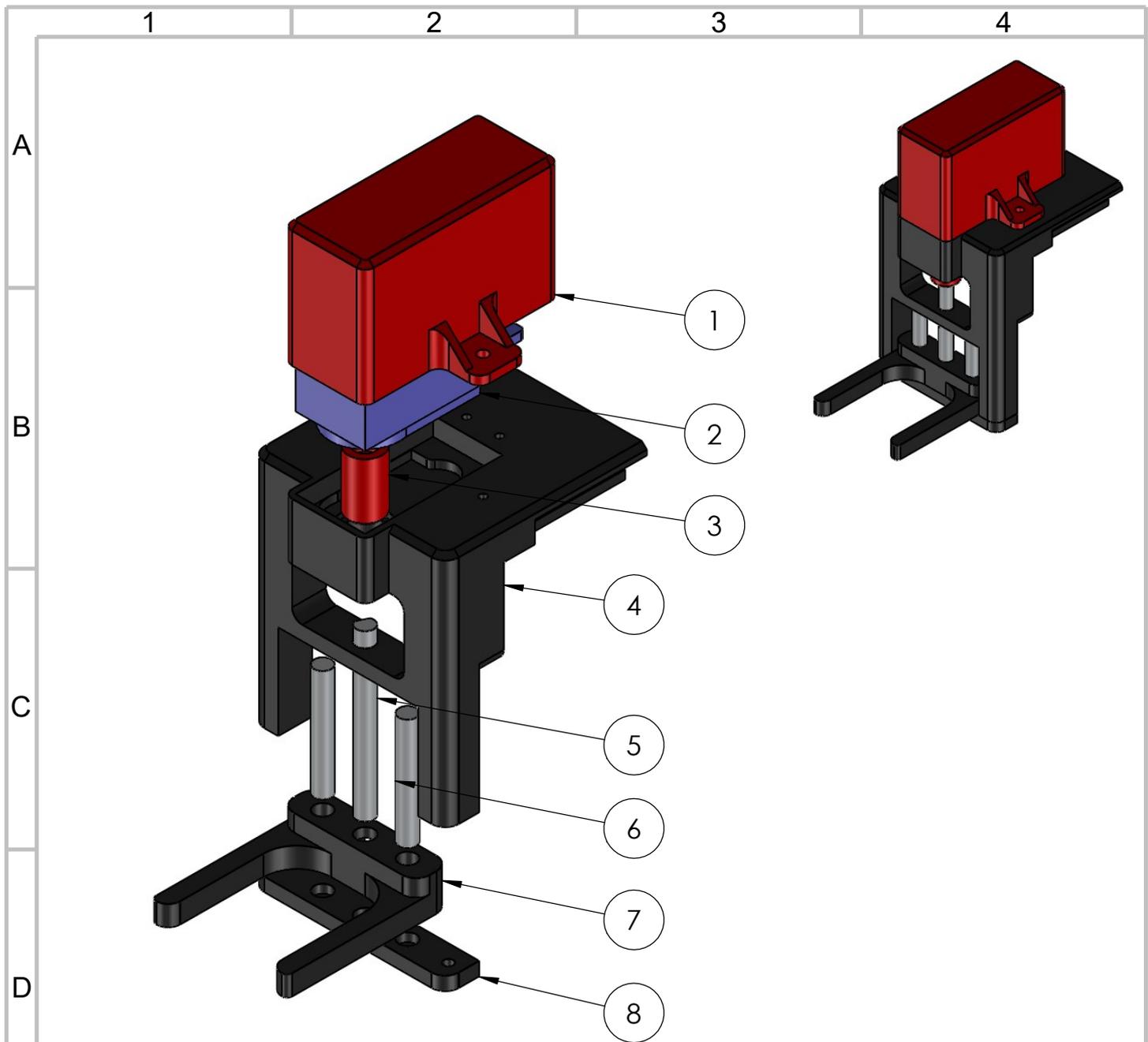
14/10/2024

1 / 1

23/10/2024

A4

23/10/2024



NOTA:
Todas las uniones se conectan mediante tornillos para plástico M2x8mm.

8	Tapa de grúa.	RB01_04_03	PETG	1	Ninguna
7	Elevador.	RB01_04_05	PETG	1	Ninguna
6	Guía de grúa.	RB01_04_06	Acero	2	Ninguna
5	Tornillo elevador.	RB01_04_07	Acero	1	Ninguna
4	Base de grúa.	RB01_04_01	PETG	1	Ninguna
3	Acople servomotor 360 grados.	RB01_04_04	PETG	1	Ninguna
2	Servomotor 360 grados.	-	-	1	Ninguna
1	Tapa servomotor 360 grados.	RB01_04_02	PETG	1	Ninguna
No. de piez.	Denominación	No. de Norma/Dibujo	Material	Cantidad	Observaciones

TOLERANCIA
 $\pm 0,2$

MASA
159.70 g

Plano de ensamblaje

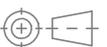
	FECHA	NOMBRE
DIBUJO:	23/10/24	Túquerres J.
REVISO:	23/10/24	Túquerres J.
APROBÓ:	23/10/24	Túquerres J.

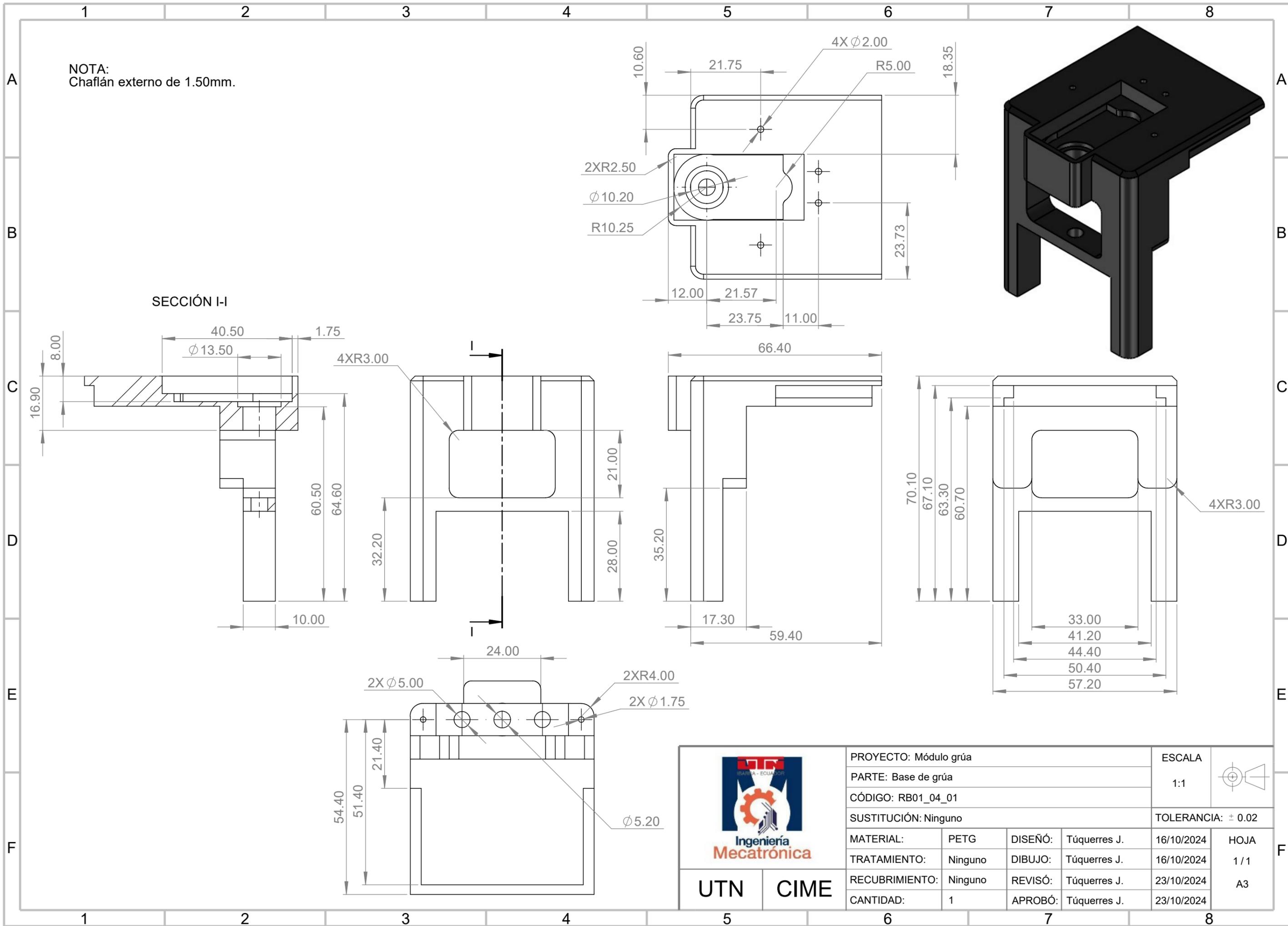
PROYECTO: Módulo grúa

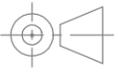
ESCALA
0.8:1



CÓDIGO: RB01_04_00

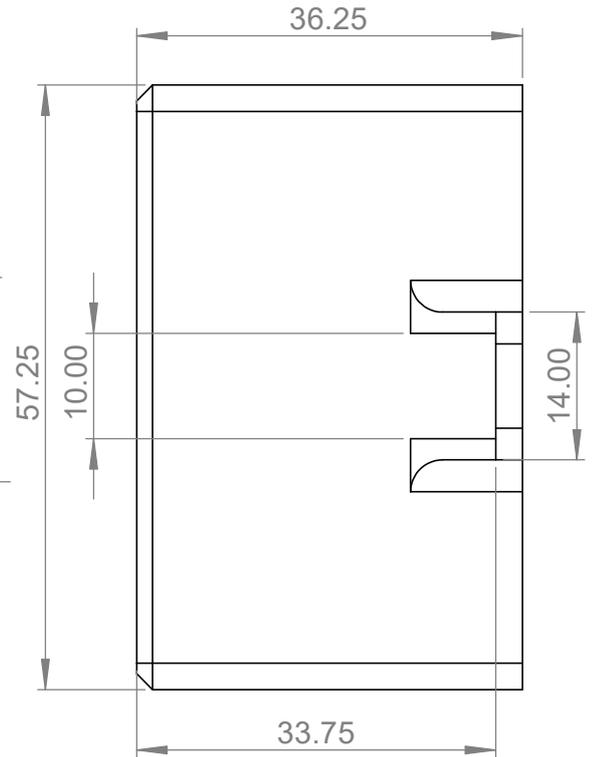
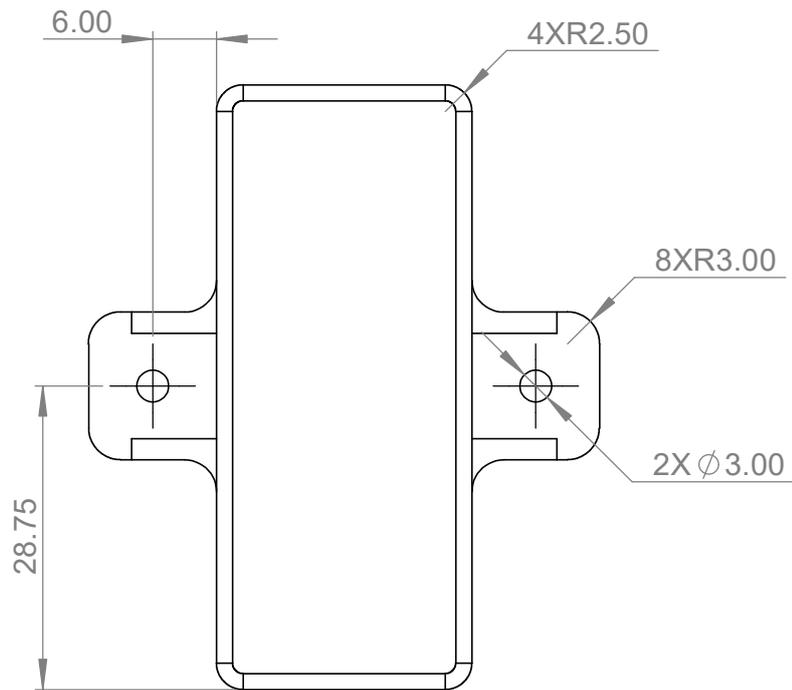
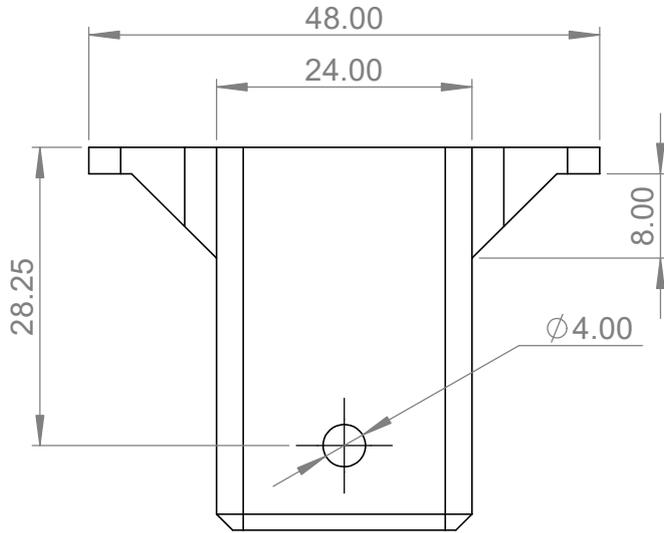
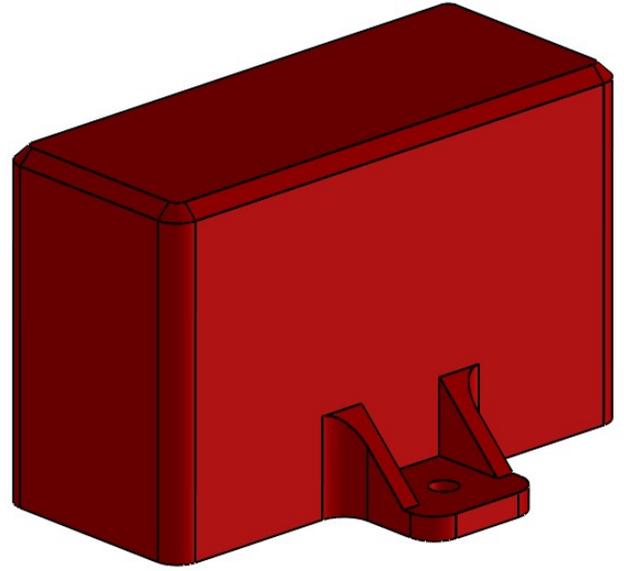




 Ingeniería Mecatrónica	PROYECTO: Módulo grúa			ESCALA 1:1	
	PARTE: Base de grúa				
	CÓDIGO: RB01_04_01			TOLERANCIA: ± 0.02	
	SUSTITUCIÓN: Ninguno				
	MATERIAL: PETG	DISEÑO: Túquerres J.	16/10/2024	HOJA	
	TRATAMIENTO: Ninguno	DIBUJO: Túquerres J.	16/10/2024	1 / 1	
	RECUBRIMIENTO: Ninguno	REVISÓ: Túquerres J.	23/10/2024	A3	
	CANTIDAD: 1	APROBÓ: Túquerres J.	23/10/2024		

UTN **CIME**

NOTA:
Espezor de la pieza de 1.50mm.



Ingeniería
Mecatrónica

PROYECTO: Módulo grúa

PARTE: Tapa servomotor 360 grados

CÓDIGO: RB01_04_02

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

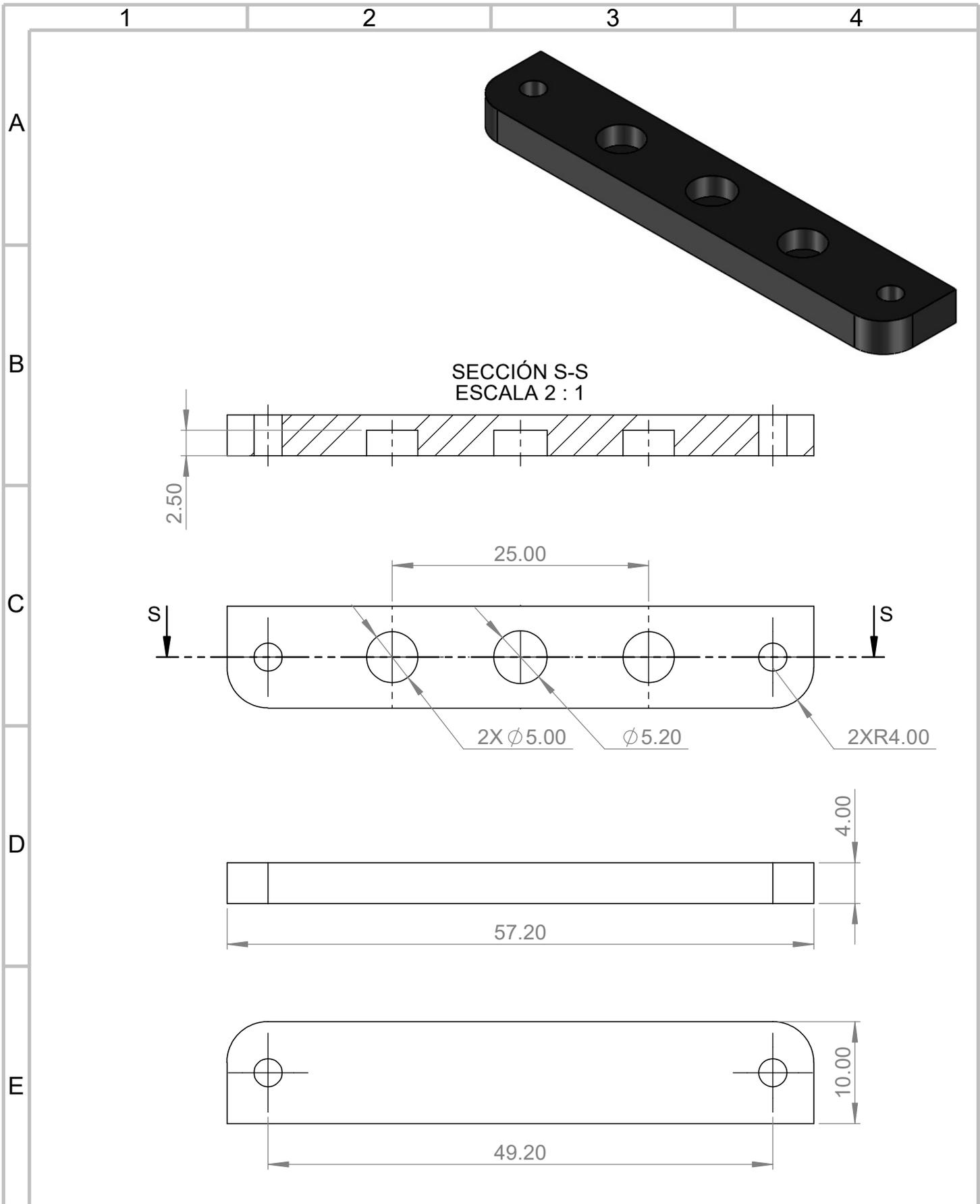
A4

16/10/2024

16/10/2024

23/10/2024

23/10/2024



Ingeniería
Mecatrónica

PROYECTO: Módulo grúa

PARTE: Tapa de grúa

CÓDIGO: RB01_04_03

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑÓ:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

1:1



TOLERANCIA: ± 0.02

UTN

CIME

16/10/2024

HOJA

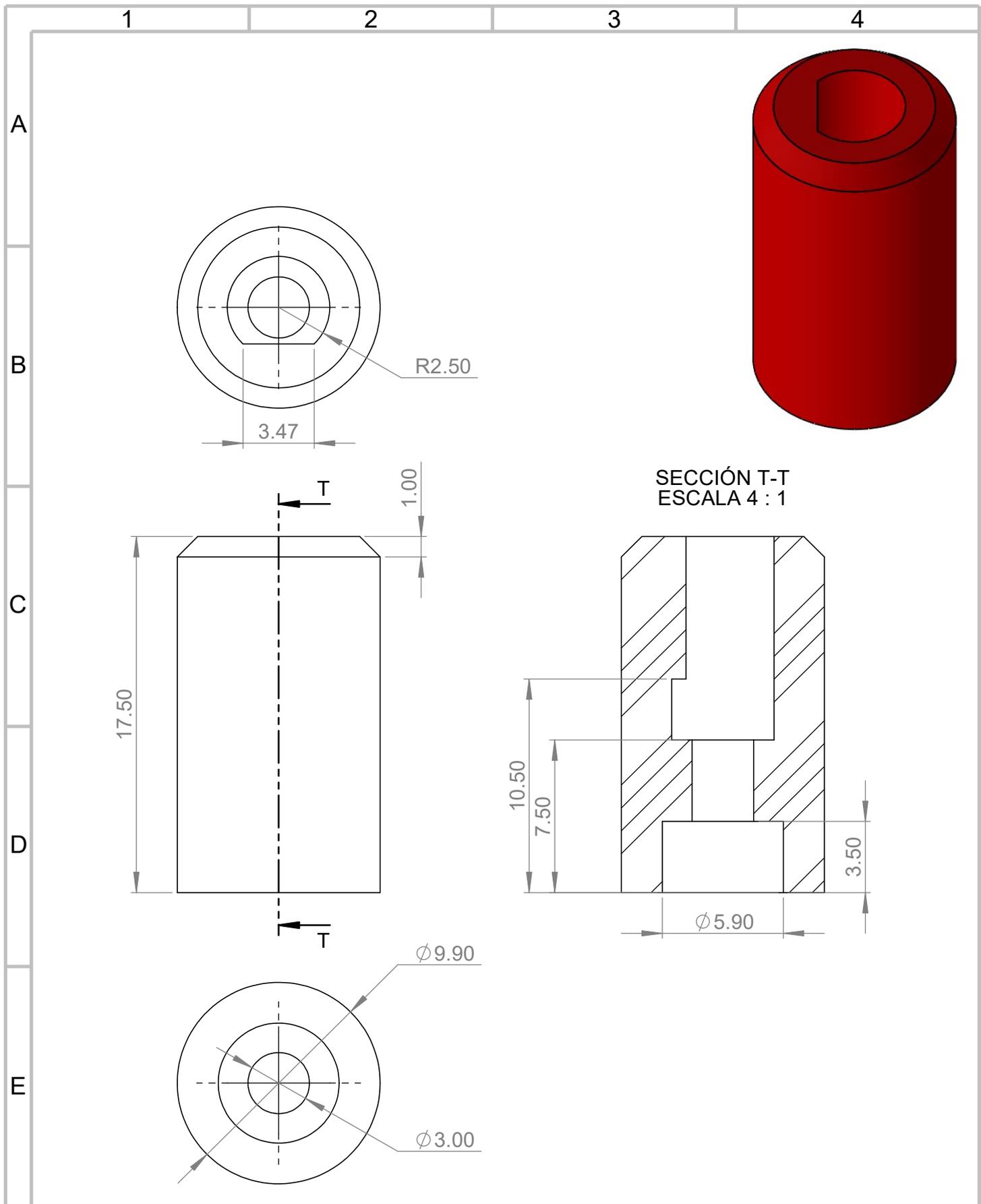
16/10/2024

1 / 1

23/10/2024

A4

23/10/2024



SECCIÓN T-T
ESCALA 4 : 1



PROYECTO: Módulo grúa

PARTE: Acople servomotor 360 grados

CÓDIGO: RB01_04_04

SUSTITUCIÓN: Ninguno

MATERIAL:

PETG

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

HOJA

1 / 1

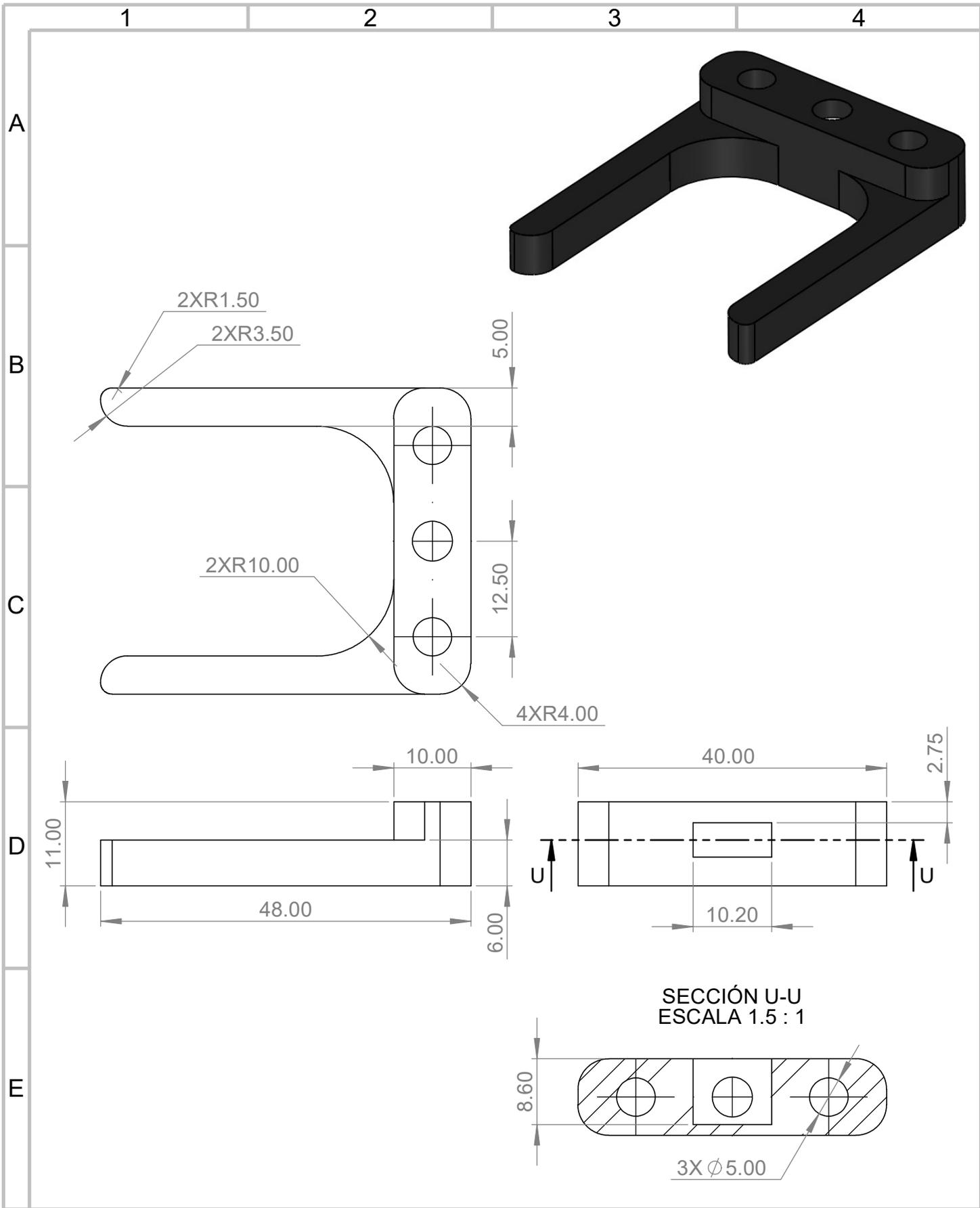
A4

16/10/2024

16/10/2024

23/10/2024

23/10/2024



Ingeniería
Mecatrónica

PROYECTO: Módulo grúa

PARTE: Elevador

CÓDIGO: RB01_04_05

SUSTITUCIÓN: Ninguno

MATERIAL: PETG

TRATAMIENTO: Ninguno

RECUBRIMIENTO: Ninguno

CANTIDAD: 1

DISEÑO: Túquerres J.

DIBUJO: Túquerres J.

REVISÓ: Túquerres J.

APROBÓ: Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

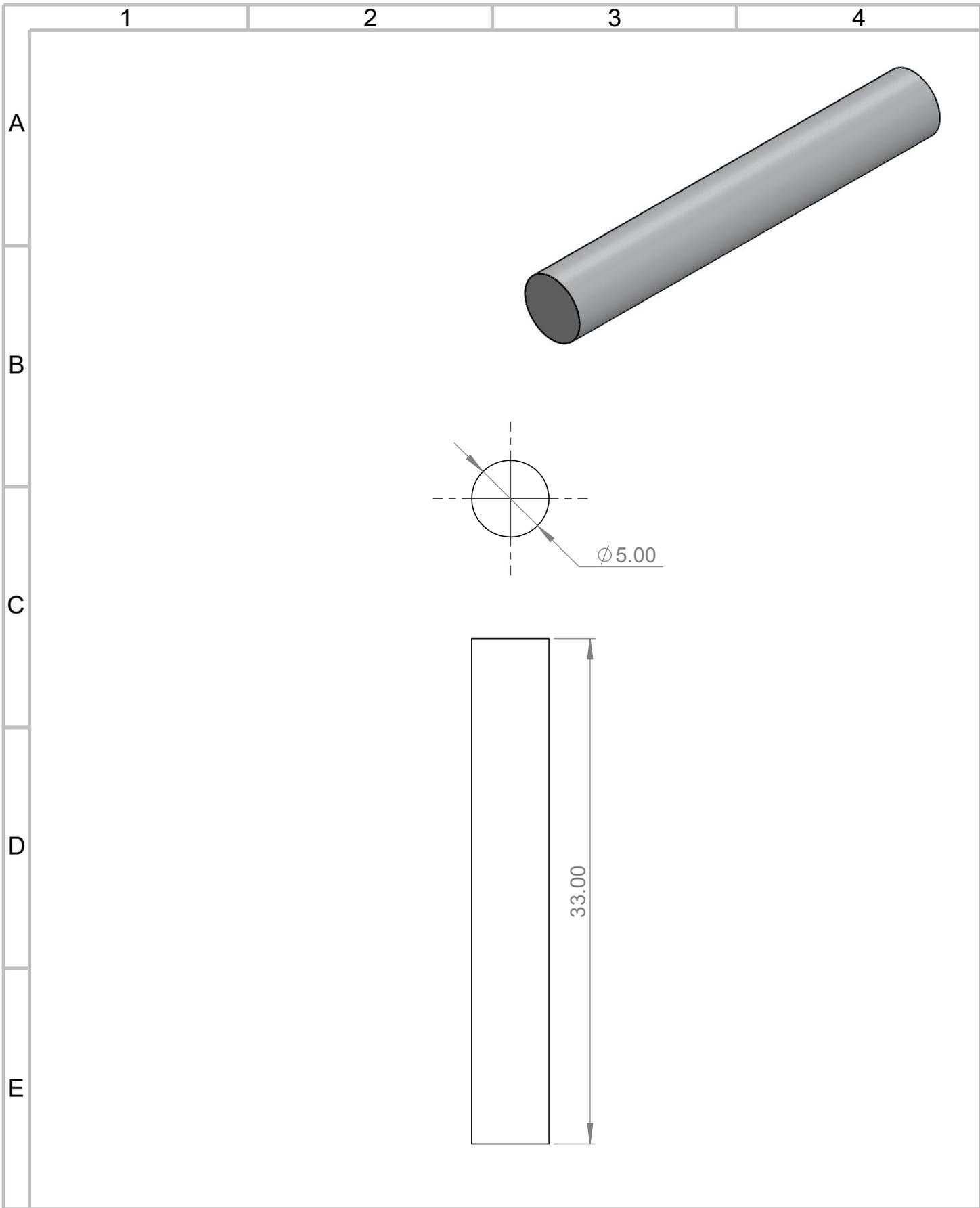
HOJA

1 / 1

A4

UTN

CIME



Ingeniería
Mecatrónica

PROYECTO: Módulo grúa

PARTE: Guía de grúa

CÓDIGO: RB01_04_06

SUSTITUCIÓN: Ninguno

MATERIAL:

Acero

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

2

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

16/10/2024

HOJA

16/10/2024

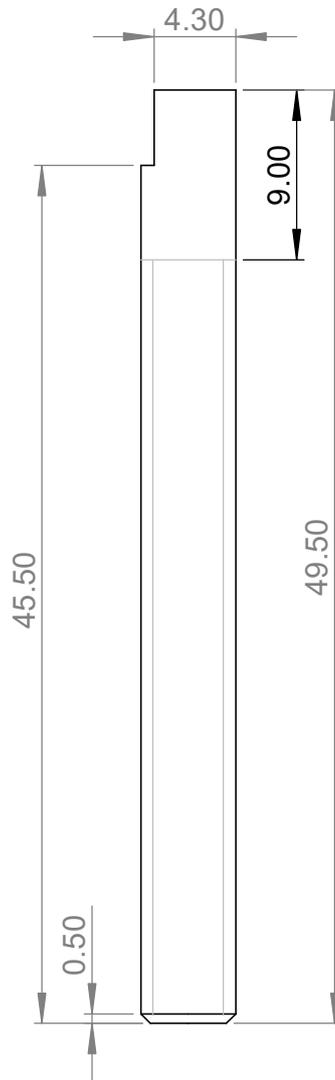
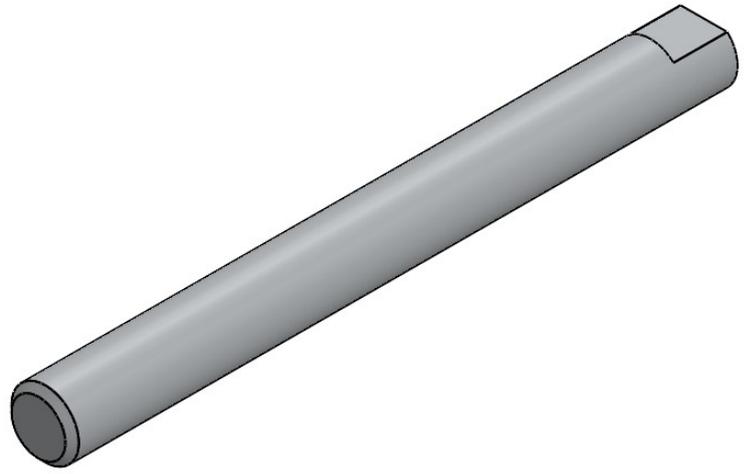
1 / 1

23/10/2024

A4

23/10/2024

NOTA:
Tornillo base M5X1.50.



Ingeniería
Mecatrónica

PROYECTO: Módulo grúa

PARTE: Tornillo elevador

CÓDIGO: RB01_04_07

SUSTITUCIÓN: Ninguno

MATERIAL:

Acero

DISEÑO:

Túquerres J.

TRATAMIENTO:

Ninguno

DIBUJO:

Túquerres J.

RECUBRIMIENTO:

Ninguno

REVISÓ:

Túquerres J.

CANTIDAD:

1

APROBÓ:

Túquerres J.

ESCALA

2:1



TOLERANCIA: ± 0.02

UTN

CIME

16/10/2024

HOJA

16/10/2024

1 / 1

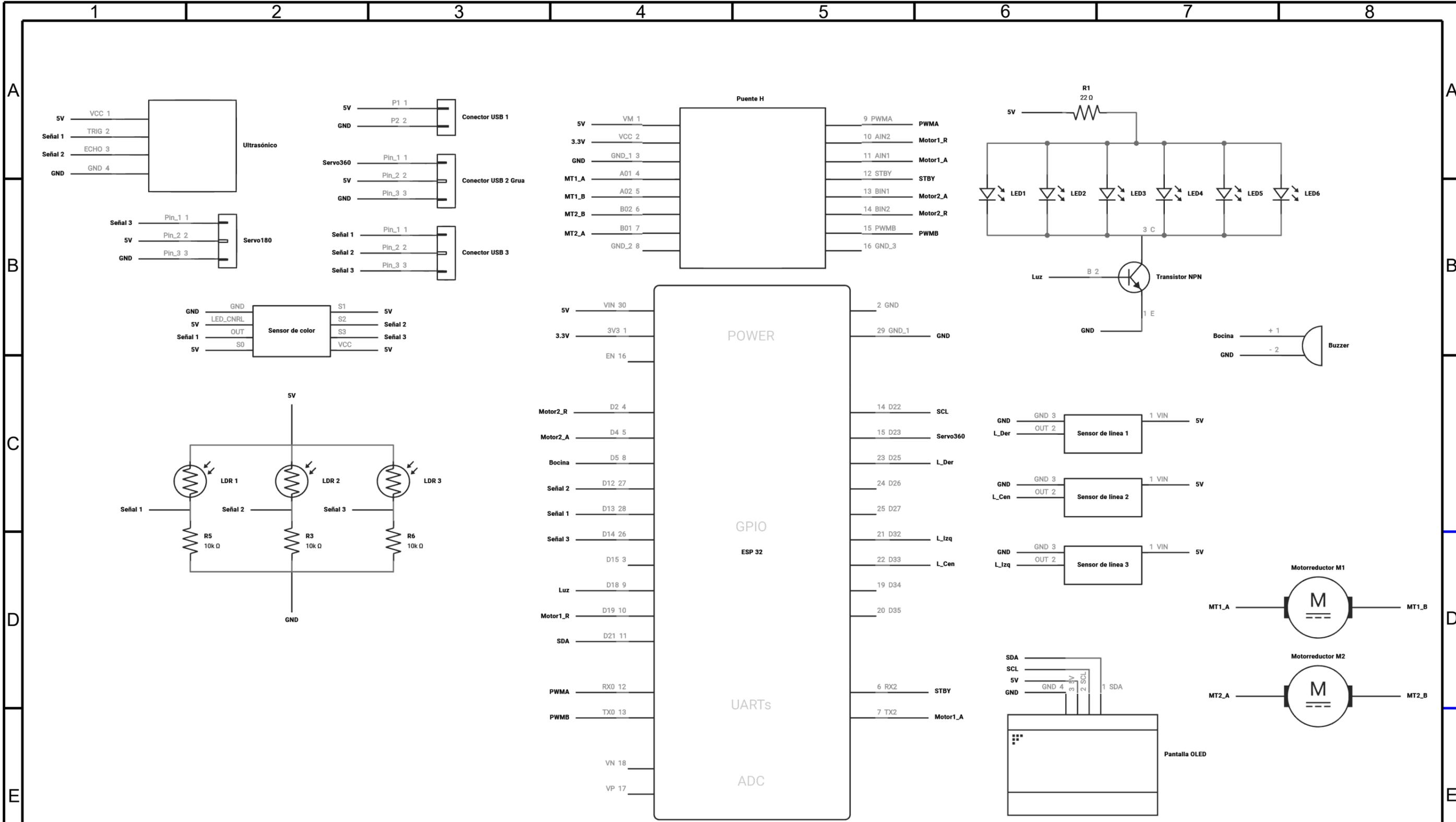
23/10/2024

A4

23/10/2024

Anexo B

Plano de conexiones



ESQUEMA ELÉCTRICO			Plano de conexiones	
Flux IA			PROYECTO: Robot modular RB01	HOJA
	FECHA	NOMBRE		1:1
DIBUJO:	23/10/24	Túquerres J.		
REVISO:	23/10/24	Túquerres J.		
APROBÓ:	23/10/24	Túquerres J.		
			CÓDIGO: RB01_PCE_00	TAMAÑO
				A3

Anexo C

Código de programación del microcontrolador ESP32

```
//Librerias
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP32Servo.h>

// Configuración de la pantalla
OLED

#define SCREEN_WIDTH 128 //
Ancho de la pantalla

#define SCREEN_HEIGHT 64 //
Alto de la pantalla

#define OLED_RESET -1 // Pin
de reset

Adafruit_SSD1306
display(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire,
OLED_RESET);

//Configuración de ojos

#include
<FluxGarage_RoboEyes.h>
roboEyes roboEyes;

//Configuración Bluetooth

#include <BluetoothSerial.h>

#if
!defined(CONFIG_BT_ENABLED)
||
!defined(CONFIG_BLUEDROID_
ENABLED)

#error Bluetooth is not enabled!
Please run `make menuconfig` to
and enable it

#endif

//Configurar nombre del dispositivo
BluetoothSerial SerialBT;

const String deviceName =
"RB_01";

//Servomotores
Servo servo1;
Servo servo2;

//Motores
int Motor2_R = 4;
int Motor2_A = 2;
int STBY = 16;
int Motor1_A = 19;

int Motor1_R = 17;
int PWMA = 3;
int PWMB = 1;

//Indicadores
int Luz = 18;
int Bocina = 5;

//Variables de modalidad seguidor
de líneas
int lineaIz = 33;
int lineaCen = 32;
int lineaDer = 25;

int val1=0;
int val2=0;
int val3=0;

int estado = 0;

//Conector USB de señales
int S1 = 13;
int S2 = 12;
int S3 = 14;

//Variables de modalidad seguidor
de luz
float luzCe = 0;
float luzDe = 0;
float luzIz = 0;
int luzIzq = 0;
int luzCen = 0;
int luzDer = 0;

//Variables de sensor ultrasónico
int duracion;
int distancia;

//Variables de sensor de color
int R_Min = 5;
int R_Max = 38;
int G_Min = 4;
int G_Max = 42;
int B_Min = 4;
int B_Max = 35;
int Red = 0;

int Green = 0;
int Blue = 0;
int redValue;
int greenValue;
int blueValue;
int Frequency;

void setup() {
Serial.begin(115200);

if(!display.begin(SSD1306_SWITC
HCAPVCC, 0x3C)) { // El segundo
parámetro varia según la dirección
I2C

Serial.println(F("Fallo en la
asignacion de la pantalla SSD"));

}

pinMode(Motor1_A, OUTPUT);
//Salidas del microcontrolador

pinMode(Motor1_R, OUTPUT);
pinMode(Motor2_A, OUTPUT);
pinMode(Motor2_R, OUTPUT);
pinMode(STBY, OUTPUT);

pinMode(lineaIz,INPUT);
pinMode(lineaCen,INPUT);
pinMode(lineaDer,INPUT);

pinMode(Bocina, OUTPUT);
pinMode(Luz, OUTPUT);
pinMode(PWMA, OUTPUT);
pinMode(PWMB, OUTPUT);

digitalWrite(Motor1_A, LOW);
digitalWrite(Motor1_R, LOW);
digitalWrite(Motor2_A, LOW);
digitalWrite(Motor2_R, LOW);

digitalWrite(Bocina, LOW);
digitalWrite(Luz, LOW);
digitalWrite(STBY, HIGH);
analogWrite(PWMA, 200);
analogWrite(PWMB, 200);

SerialBT.begin(deviceName);

Serial.println(deviceName +
"Conectado!...");
```

```

// Pantalla y sonido de inicio
robot_Encendido();

// Configuración de pantalla con
ojos
robot_Ojos();
}

void loop(){
  roboEyes.update();    //Ojos
interactivos
  if (SerialBT.available())
  {
    char datos = SerialBT.read();
    Serial.write(datos);
    switch(datos){
      case 'A':
        robot_Adelante();
        break;
      case 'B':
        robot_Reversa();
        break;
      case 'C':
        robot_Izquierda();
        break;
      case 'D':
        robot_Derecha();
        break;
      case 'F':
        robot_Feliz();
        break;
      case 'G':
        robot_Enojado();
        break;
      case 'H':
        robot_Triste();
        break;
      case 'I':
        robot_Bocina();
        break;
      case 'J':
        digitalWrite(Luz, HIGH);
        break;
      case 'K':
        digitalWrite(Luz, LOW);
        break;
      case 'a':
        robot_Conectado();
        break;
      case 'b':
        robot_Desconectado();
        break;
      case 'X':
        roboEyes.setMood(HAPPY);
        roboEyes.setPosition(S);
        break;
      case 'Y':
        roboEyes.setMood(ANGRY);
        roboEyes.setPosition(S);
        break;
      case 'Z':
        roboEyes.setMood(TIRED);
        roboEyes.setPosition(S);
        break;
      case 'W':
        roboEyes.setMood(DEFAULT);
        roboEyes.setPosition(S);
        break;
      case 'P':
        roboEyes.setMood(DEFAULT);
        roboEyes.setPosition(SE);
        break;
      case 'Q':
        roboEyes.setMood(DEFAULT);
        roboEyes.setPosition(SW);
        break;
      case 'L':
        robot_Seguidor_de_lineas();
        break;
      case 'M':
        robot_Marcador_de_trazos();
        break;
      case 'N':
        robot_Seguidor_de_luz();
        pinMode(S1,INPUT);
        pinMode(S2,INPUT);
        pinMode(S3,INPUT);
        break;
      case 'R':
        robot_Evasor_de_objetos();
        servo1.attach(S1);
        pinMode(S2,INPUT);
        pinMode(S3,OUTPUT);
        break;
      case 'S':
        robot_Lector_de_colores();
        pinMode(S2, OUTPUT);
        pinMode(S1, OUTPUT);
        pinMode(S3, INPUT);
        digitalWrite(Luz, LOW);
        break;
      case 'T':
        robot_Desplazador_de_objetos();
        servo2.attach(23);
        break;
      case 'c':
        robot_dibujar_cuadrado();
        break;
      case 'd':
        robot_dibujar_circulo();
        break;
      case 'e':
        robot_dibujar_triangulo();
        break;
      case 'f':
        robot_dibujar_1();
        break;
      case 'g':
        robot_dibujar_2();
        break;
      case 'h':
        robot_dibujar_3();
        break;
      case 'i':
        robot_dibujar_4();
        break;
      case 'j':
        robot_dibujar_5();
        break;
      case 'k':
        servo2.write(180);
        delay(4000);
        servo2.write(90);
        break;
      case 'l':
        servo2.write(0);
    }
  }
}

```



```

digitalWrite(Motor1_R, HIGH);
delay(500);
motores_apagados();
digitalWrite(Motor1_A, HIGH);
delay(500);
motores_apagados();
digitalWrite(Motor1_R, HIGH);
delay(500);
motores_apagados();
digitalWrite(Motor1_A, HIGH);
delay(500);
motores_apagados();
digitalWrite(Motor1_R, HIGH);
delay(500);
motores_apagados();
}
break;
}
}
void robot_Adelante(){
roboEyes.setPosition(S);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(300);
motores_apagados();
}
void robot_Reversa(){
roboEyes.setPosition(S);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(300);
motores_apagados();
}
void robot_Izquierda(){
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(100);
motores_apagados();
}
void robot_Derecha(){
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(100);
motores_apagados();
}
}
void robot_Bocina(){
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
}
void robot_Encendido(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2); // Tamaño
del texto
display.setCursor(35,0); // Posición
inicial del texto
display.println("Robot");
display.setTextSize(4);
display.setCursor(8,28);
display.println("RB_01");
display.display();
tone(Bocina, 2000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
digitalWrite(Luz, LOW);
delay(100);
tone(Bocina, 2000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
delay(3000);
}
void robot_Conectado(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(0,0);
display.println("Estado: ");
}
void robot_Desconectado(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2); // Tamaño
del texto
display.setCursor(0,0); // Posición
inicial del texto
display.println("Estado: ");
display.setTextSize(2);
display.setCursor(0,25);
display.println("Desconectado!");
display.display();
tone(Bocina, 2000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
digitalWrite(Luz, LOW);
delay(200);
tone(Bocina, 1000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
delay(2000);
}
void motores_apagados(){
digitalWrite(Motor1_A, LOW);
}
display.setTextSize(2);
display.setCursor(0,28);
display.println("Conectado!");
display.display();
tone(Bocina, 1000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
digitalWrite(Luz, LOW);
delay(200);
tone(Bocina, 2000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
delay(2000);
}
}
void motores_apagados(){
digitalWrite(Motor1_A, LOW);
}

```



```

digitalWrite(Motor1_R, HIGH);
delay(200);
motores_apagados();
digitalWrite(Motor1_A, HIGH);
delay(200);
motores_apagados();
digitalWrite(Motor2_R, HIGH);
delay(200);
motores_apagados();
digitalWrite(Motor2_A, HIGH);
delay(200);
motores_apagados();
tone(Bocina, 2000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
digitalWrite(Luz, LOW);
delay(200);
tone(Bocina, 1000);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
}

void robot_Seguidor_de_lineas(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
display.setCursor(25,20);
display.println("Seguir");
display.setTextSize(2);
display.setCursor(25,42);
display.println("Lineas");
display.display();
tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);
}

void robot_Marcador_de_trazos(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
display.setCursor(25,20);
display.println("Marcar");
display.setTextSize(2);
display.setCursor(25,42);
display.println("Trazos");
display.display();
tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);}

void robot_Seguidor_de_luz(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
display.setCursor(25,18);
display.println("Seguir");
display.setTextSize(2);
display.setCursor(40,42);
display.println("Luz");
display.display();
tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);}

void robot_Evasor_de_objetos(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
}

display.setCursor(25,18);
display.println("Evadir");
display.setTextSize(2);
display.setCursor(20,42);
display.println("Objetos");
display.display();
tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);}

void robot_Lector_de_colores(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
display.setCursor(35,18);
display.println("Leer");
display.setTextSize(2);
display.setCursor(20,42);
display.println("Colores");
display.display();
tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);}

void
robot_Desplazador_de_objetos(){
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(35,0);
display.println("Modo");
display.setTextSize(2);
display.setCursor(10,18);
display.println("Desplazar");
display.setTextSize(2);
display.setCursor(20,42);
display.println("Objetos");
display.display();
}

```

```

tone(Bocina, 1500);
delay(100);
noTone(Bocina);
delay(2000);
}
void robot_dibujar_cuadrado(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2000);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(1250);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2000);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(1300);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2000);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(1320);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2200);
motores_apagados();
delay(300);
tone(Bocina, 2000);

delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_circulo(){
digitalWrite(Luz, LOW);
delay(100);
analogWrite(PWMA, 190);
analogWrite(PWMB, 120);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(15000);
motores_apagados();
delay(300);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
delay(200);
analogWrite(PWMA, 200);
analogWrite(PWMB, 200);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_triangulo(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2600);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(1750);
motores_apagados();
delay(300);

digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2600);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(1750);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2600);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2750);
motores_apagados();
delay(300);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_1(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_A, HIGH);
delay(1300);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(2800);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
delay(1200);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
}

```

```

delay(200);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2500);
motores_apagados();
delay(300);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
delay(100);
}
void robot_dibujar_2(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(2200);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(900);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(2200);
motores_apagados();
delay(300);
analogWrite(PWMB, 100);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(2300);
motores_apagados();
delay(300);
analogWrite(PWMA, 200);
analogWrite(PWMB, 200);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_3(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_R, HIGH);
delay(300);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(2900);
motores_apagados();
delay(300);
analogWrite(PWMA, 105);
analogWrite(PWMB, 190);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(7000);
motores_apagados();
delay(300);
analogWrite(PWMA, 200);
analogWrite(PWMB, 200);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_4(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(1500);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(200);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(1500);
motores_apagados();
delay(300);
digitalWrite(Motor2_R, HIGH);
delay(1200);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);
delay(1300);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(3200);
motores_apagados();
delay(300);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
}

```

```

noTone(Bocina);
digitalWrite(Luz, HIGH);
}
void robot_dibujar_5(){
digitalWrite(Luz, LOW);
delay(100);
digitalWrite(Motor1_R, HIGH);
delay(300);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
delay(3000);
motores_apagados();
delay(300);
digitalWrite(Motor2_R, HIGH);
delay(1200);
motores_apagados();
delay(300);
digitalWrite(Motor1_R, HIGH);
digitalWrite(Motor2_R, HIGH);

delay(1600);
motores_apagados();
delay(300);
digitalWrite(Motor2_R, HIGH);
delay(1200);
motores_apagados();
delay(300);
digitalWrite(Motor1_A, HIGH);
digitalWrite(Motor2_A, HIGH);
delay(1300);
motores_apagados();
delay(300);
tone(Bocina, 2000);
delay(100);
noTone(Bocina);
delay(200);
tone(Bocina, 1000);
delay(100);
noTone(Bocina);
digitalWrite(Luz, HIGH);
}

int getRed() {
digitalWrite(S2,LOW);
digitalWrite(S1,LOW);
Frequency = pulseIn(S3, LOW);
return Frequency;
}

int getGreen() {
digitalWrite(S2,HIGH);
digitalWrite(S1,HIGH);
Frequency = pulseIn(S3, LOW);
return Frequency;
}

int getBlue() {
digitalWrite(S2,LOW);
digitalWrite(S1,HIGH);
Frequency = pulseIn(S3, LOW);
return Frequency;
}

```

Anexo D

Código de programación del panel de control

```
using System;
using System.Drawing;
using System.IO.Ports;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using System.Threading;

namespace Interfaz_RB01_V2._0
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            btnConectar.Enabled = true;
            btnBuscar.Enabled = true;
            btnOff.Visible = false;
            pgbConectar.Value = 0;

            string[] portList =
            SerialPort.GetPortNames();
            cmbPuertos.Items.AddRange(portList);

            //Mover panel
            [DllImport("user32.DLL",
            EntryPoint = "ReleaseCapture")]
            private extern static void
            ReleaseCapture();

            [DllImport("user32.DLL",
            EntryPoint = "SendMessage")]
            private extern static void
            SendMessage(System.IntPtr hWnd,
            int wMsg, int wParam, int lParam);

            private void
            PanelTitulo_MouseDown(object
            sender, MouseEventArgs e)
            {
                ReleaseCapture();

                SendMessage(this.Handle,
                0x112, 0xf012, 0);
            }

            private void
            btnCerrar_Click(object sender,
            EventArgs e)
            {
                Application.Exit();
            }

            private void
            btnMaximizar_Click(object sender,
            EventArgs e)
            {
                this.WindowState =
                FormWindowState.Maximized;

                btnMaximizar.Visible =
                false;
                btnRestaurar.Visible = true;
            }

            private void
            btnRestaurar_Click(object sender,
            EventArgs e)
            {
                this.WindowState =
                FormWindowState.Normal;

                btnRestaurar.Visible = false;
                btnMaximizar.Visible = true;
            }

            private void
            btnMinimizar_Click(object sender,
            EventArgs e)
            {
                this.WindowState =
                FormWindowState.Minimized;
            }

            private void
            btnSalir_Click(object sender,
            EventArgs e)
            {
                Application.Exit();
            }

            private void
            btnBuscar_Click(object sender,
            EventArgs e)
            {
                string[] ports =
                SerialPort.GetPortNames();

                cmbPuertos.DataSource =
                ports;

                private void
                btnConectar_Click(object sender,
                EventArgs e)
                {
                    try
                    {
                        if (!serialPort1.IsOpen)
                        {
                            serialPort1.PortName =
                            cmbPuertos.Text;

                            serialPort1.BaudRate =
                            Convert.ToInt32("115200");

                            serialPort1.Open();

                            serialPort1.WriteLine("a");

                            pgbConectar.Value =
                            100;

                            btnConectar.BackColor
                            = Color.Red;

                            btnConectar.Text = "
                            Desconectar";

                            btnBuscar.Enabled =
                            false;
                            cmbPuertos.Enabled =
                            false;
                            btnAdelante.Enabled =
                            true;
                            btnReversa.Enabled =
                            true;
                            btnRotarD.Enabled =
                            true;
                            btnRotarI.Enabled =
                            true;
                            btnBocina.Enabled =
                            true;
                            btnLuzOff.Enabled =
                            true;
                            btnLuz.Enabled = false;
                            btnFeliz.Enabled = true;
                            btnTriste.Enabled =
                            true;
                        }
                    }
                }
            }
        }
    }
}
```



```

        {
            btnReversa_Click(sender,
e);
        }
        else if (e.KeyCode ==
Keys.A)
        {
            btnRotarI_Click(sender,
e);
        }
        else if (e.KeyCode ==
Keys.D)
        {
            btnRotarD_Click(sender,
e);
        }
        else if (e.KeyCode ==
Keys.R)
        {
            btnBocina_Click(sender,
e);
        }
        }
        private void
cmbModalidades_SelectedIndexCha
nged(object sender, EventArgs e)
        {
            string modalidad =
cmbModalidades.SelectedItem.ToSt
ring();
            switch (modalidad)
            {
                case "Seguidor de líneas":
serialPort1.WriteLine("E");

serialPort1.WriteLine("L");

                btnOn.Enabled = true;
                btnOn.Visible = true;

                pictureRobot.Image =
Properties.Resources.Img_Lineas;
                break;
                case "Marcador de
trazos":
serialPort1.WriteLine("E");

serialPort1.WriteLine("M");

                btnOn.Enabled = true;
                btnOn.Visible = true;
                pictureRobot.Image =
Properties.Resources.Img_Trazos;
                break;
                case "Seguidor de luz":
serialPort1.WriteLine("E");

serialPort1.WriteLine("N");

                btnOn.Enabled = true;
                btnOn.Visible = true;
                pictureRobot.Image =
Properties.Resources.Img_Luz;
                break;
                case "Evasor de objetos":
serialPort1.WriteLine("E");

serialPort1.WriteLine("R");

                btnOn.Enabled = true;
                btnOn.Visible = true;
                pictureRobot.Image =
Properties.Resources.Img_Evasor;
                break;
                case "Lector de colores":
serialPort1.WriteLine("E");

serialPort1.WriteLine("S");

                btnOn.Enabled = true;
                btnOn.Visible = true;
                pictureRobot.Image =
Properties.Resources.Img_Color;
                break;
                case "Desplazador de
objetos":
serialPort1.WriteLine("E");

serialPort1.WriteLine("T");

                btnOn.Enabled = true;
                btnOn.Visible = true;
                pictureRobot.Image =
Properties.Resources.Img_Grua;
                break;
            }
        }
        private void
btnOn_Click(object sender,
EventArgs e)
        {
            string modalidad =
cmbModalidades.SelectedItem.ToSt
ring();
            if (modalidad == "Seguidor
de líneas")
            {
                serialPort1.WriteLine("U");

                btnOn.Visible = false;
                btnOff.Enabled = true;
                btnOff.Visible = true;
            }
            else if (modalidad ==
"Marcador de trazos")
            {
                btnCuadrado.Enabled =
true;
                btnCirculo.Enabled = true;
                btnTriangulo.Enabled =
true;
                btn1.Enabled = true;
                btn2.Enabled = true;
                btn3.Enabled = true;
                btn4.Enabled = true;
                btn5.Enabled = true;
                btnOn.Visible = false;
                btnOff.Enabled = true;
                btnOff.Visible = true;
            }
            else if (modalidad ==
"Seguidor de luz")
            {
                serialPort1.WriteLine("O");

                btnOn.Visible = false;
                btnOff.Enabled = true;
                btnOff.Visible = true;
            }
            else if (modalidad ==
"Evasor de objetos")
            {
                serialPort1.WriteLine("V");

                btnOn.Visible = false;
                btnOff.Enabled = true;
            }
        }
    }
}

```


Anexo E

Manual de usuario del dispositivo

Manual de usuario

ROBOT MODULAR ENFOCADO A FOMENTAR LA METODOLOGÍA STEAM EN LA EDUCACIÓN

Propósito: Proporcionar al usuario una guía la cual facilite la comprensión de los usos y funcionamiento que posee el dispositivo modular, así como brindar instrucciones con relación a las actividades que este puede desarrollar dentro de un entorno educativo.

Introducción: El dispositivo modular posee la capacidad de desplazarse y realizar diversas acciones depende a su composición modular y configuración de estado, los cuales son seleccionados a través de su tablero de control. El diseño de este panel presenta diferentes comandos relacionados al aprendizaje y razonamiento lógico matemático que puede llegar a presentarse dentro de centros de educación general básica.

Características del dispositivo

El robot modular "RB_01", posee la capacidad de operar en cinco modalidades distintas. Cada modalidad requiere del acoplamiento y conexión de diferentes módulos en su sección superior, lo que le permite realizar una amplia gama de acciones controladas por el usuario mediante su acceso al panel de control.

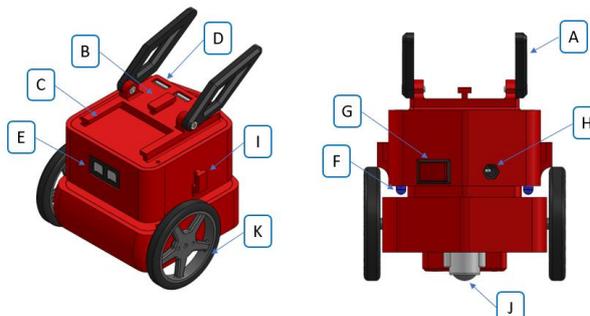
El software que contiene el panel de control posee compatibilidad con sistemas Windows, además de no requerir de pasos adicionales dentro de su instalación.



Características del dispositivo (Estructura)

Lista de elementos

- A. Antenas
- B. Guía posterior
- C. Guía frontal
- D. Conectores USB
- E. Pantalla OLED
- F. Luces LED
- G. Interruptor de encendido
- H. Jack de carga
- I. Guías laterales
- J. Rueda loca
- K. Llantas



Características del dispositivo (Panel de control)

Lista de componentes

- A. Panel de búsqueda de dispositivos disponibles.
- B. Botón de conexión con el dispositivo.
- C. Manual de usuario.
- D. Panel de graficas disponibles.
- E. Panel de control manual del dispositivo.
- F. Panel de reacciones (feliz, triste, enojado).
- G. Panel de control de grúa .
- H. Listado de modalidades del dispositivo.
- I. Grafica de ensamble del dispositivo.



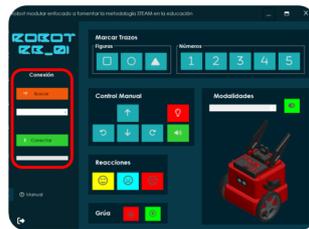
Encendido y conexión del dispositivo

Para iniciar el funcionamiento del robot, se debe activar el interruptor rojo localizado en la parte posterior del dispositivo. Esta acción activará la pantalla, mostrando el nombre del robot, además de producir un parpadeo por parte de los indicadores LED.



Posteriormente se debe buscar al dispositivo mediante el panel de control el cual permitirá establecer su conexión a través de bluetooth. Una vez enlazado con el dispositivo el usuario posee acceso a todas las modalidades que este posee siempre y cuando se encuentre ensamblado según indique la guía.

Panel de conexión del dispositivo



Modalidad - Seguidor de líneas

La modalidad de seguimiento de líneas corresponde a la forma más básica del robot. Esto debido a que no requiere del ensamblaje de módulos electrónicos externos para su funcionamiento.

Esta modalidad le permite al usuario guiar al robot hacia un objetivo en específico, haciendo uso de cinta adhesiva de color negro la cuales cumple la función de simular un camino el cual seguirá el dispositivo.



Modalidad - Marcador de trazos

Dentro de esta modalidad el usuario puede realizar graficas basadas en trazos, las cuales son generadas mediante el acoplamiento de dos bases laterales encargadas de sostener un marcador de tiza líquida en cada extremo.

Mediante el panel de control el usuario posee la opción de graficar de forma automática tanto figuras geométricas (marcador izquierdo) como secuencias numéricas del 1 al 5 (marcador derecho).



Modalidad - Seguidor de luz

Mediante esta modalidad el movimiento del robot está controlado por medio de tres sensores de luz acoplados en su parte superior. Estos sensores le permiten al usuario guiar al dispositivo a través de obstáculos utilizando una fuente de luz como guía.

Además, este módulo es considerado como el primero en requerir de una conexión eléctrica por lo que es necesario revisar el código de colores que este posee en cada cable para así poder enlazarlo con su entrada USB correspondiente.



Modalidad - Evasor de objetos

Dentro de esta modalidad el desplazamiento del robot es controlado mediante un sensor ultrasónico ubicado en la parte superior del dispositivo, lo que le brinda al dispositivo la capacidad de evadir obstáculos a una distancia máxima de 25 cm.

El módulo posee un servomotor de 180 grados lo que le permite escanear y elegir la ruta más factible en su trayectoria, además de que para su uso es recomendable colocar objetos los cuales se encuentren a una altura igual o superior a la del sensor.



Modalidad - Lector de colores

El modo lector de colores le permite al dispositivo acceder a diferentes acciones pre grabadas por el usuario. Estas acciones son efectuadas gracias al escaneo de fichas las cuales poseen diferentes colores y símbolos en su superficie.

El sensor de color que incorpora este módulo domina la capacidad de reconocer entre cuatro tipos de colores diferentes, además de que posee un alcance de lectura recomendado de hasta 5cm.



Modalidad - Desplazador de objetos

Dentro de esta modalidad el robot es capaz de levantar y movilizar pequeños cubos numerados con ayuda de una grúa acoplada en su parte frontal.

El control de este módulo es realizado mediante el panel de control el cual además de accionar la extensión de grúa cumple la función de direccionar al robot hacia una posición deseada.

