



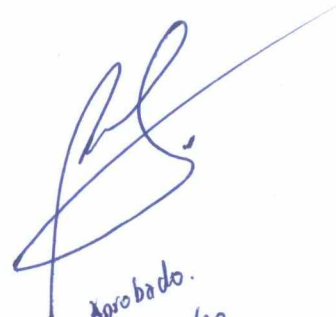
UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UN ROBOT MÓVIL CON SISTEMA DE
RECOLECCIÓN DE OBJETOS CONTROLADO REMOTAMENTE
PARA FINES DIDÁCTICOS**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Joselyn Carolai Burbano Merino
Luis Alberto Cárdenas León
TUTOR: Ing. Ricardo Patricio Manzano Puente M.Sc.

Guayaquil-Ecuador
2022


Aprobado.
2022/09/20

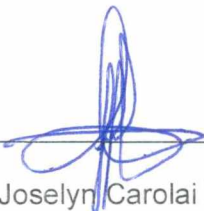
**CERTIFICADO DE RESPONSABILIDAD Y AUDITORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, **Joselyn Carolai Burbano Merino** con documentos de identificación N° **0926016429** y **Luis Alberto Cárdenas León** con documento de identificación N° **0704257906**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 20 de marzo del año 2022

Atentamente,



Joselyn Carolai Burbano Merino

0926016429



Luis Alberto Cárdenas León

0704257906

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Joselyn Carolai Burbano Merino** con documento de identificación N° **0926016429** y **Luis Alberto Cárdenas León** con documento de identificación N° **0704257906**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **DESARROLLO DE UN ROBOT MÓVIL CON SISTEMA DE RECOLECCIÓN DE OBJETOS CONTROLADO REMOTAMENTE PARA FINES DIDÁCTICOS**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo a final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 20 de marzo del año 2022

Atentamente,



Joselyn Carolai Burbano Merino

0926016429



Luis Alberto Cárdenas León

0704257906

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Ricardo Patricio Manzano Puente**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN ROBOT MÓVIL CON SISTEMA DE RECOLECCIÓN DE OBJETOS CONTROLADO REMOTAMENTE PARA FINES DIDÁCTICOS**, realizado por **Joselyn Carolai Burbano Merino** con documento de identificación N° **0926016429** y por **Luis Alberto Cárdenas León** con documento de identificación N° **0704257906**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 20 de septiembre del año 2022

Atentamente,



Ing. Ricardo Patricio Manzano Puente M.Sc.

C.I: 1803850187

ÍNDICE

I.	PROBLEMA	10
II.	JUSTIFICACIÓN	11
III.	OBJETIVOS	12
III-A.	OBJETIVO GENERAL	12
III-B.	OBJETIVOS ESPECIFICOS	12
IV.	MARCO HIPOTÉTICO	13
IV-A.	Hipótesis	13
V.	MARCO TEÓRICO REFERENCIAL	13
V-A.	Introducción	13
V-B.	Historia De La Robótica	13
V-C.	Robots Móviles	14
V-D.	Tipos De Robots Móviles	14
V-D1.	Robot OMRON	14
V-D2.	Robot Quirúrgico Da Vinci	15
V-D3.	Robot Explorador Spirit	15
V-D4.	Robot De Rescate UNAM	15
V-D5.	Brazo Robótico Robotnik	16
V-E.	Estructura Mecánica Del Robot Móvil	16
V-E1.	Morfología Del Robot	16
V-E2.	Estructura Mecánica De Un Robot	17
V-E3.	Cartesiano	18
V-E4.	Cilíndrico	18
V-F.	Componentes Mecánicos	19
V-F1.	Nema 17	19
V-F2.	Modelo del Brazo Robótico en 3D	19
V-F3.	Micro Servomotor SG90	20
V-F4.	Pernos y Tuercas de M3 y M4	20
V-F5.	Piezas de Madera MDF de 8mm	20
V-F6.	Tornillos Autoroscantes	21
V-F7.	Plataforma de Ruedas	21
V-G.	Componentes Eléctricos	22
V-G1.	Batería LI-PO	22
V-H.	Componentes Electrónicos	22
V-H1.	Controlador DRV8825	22
V-H2.	Módulo Bluetooth HC-05	22
V-H3.	Placa Arduino Mega	23
V-H4.	Final De Carrera	23
V-H5.	Sensor Ultrasonido HC-SR04	24
V-I.	La Robótica Orientada A La Educación	24
V-J.	Arduino En La Robótica Educacional	25
V-K.	Apps En El Control De Robots	25

VI. Marco Metodológico	27
VI-A. Diseño De La Investigación	27
VI-B. Materiales Didácticos	27
VI-C. Modalidad De La Investigación	27
VI-D. Tipo De Investigación	27
VI-E. Población Y Muestra	28
VI-F. Técnica De La Investigación	28
VI-G. Instrumentación De La Investigación	28
VII. Marco Procedimental	29
VII-A. Diagrama de flujo de la lógica para la programación del Robot	29
VII-B. Diagrama de la placa electrónica	30
VII-C. Lógica a utilizar en la programación de la placa Arduino	30
VII-D. Estableciendo la lógica de la interfaz a utilizar para el Robot en MIT APP INVENTOR	33
VII-E. Diagrama Electronico de Placa PCB	36
VII-F. Dimensionamiento del motor Nema a utilizar en el proyecto	39
VII-G. Dimensionamiento del Brazo Robótico en el proyecto	40
VII-H. Dimensionamiento de la Batería a utilizar en el proyecto	43
VII-I. Prototipo Ensamblado	44
VIII. Cronograma	48
IX. Presupuesto	49
X. Conclusión y Recomendaciones	50
X-A. Conclusiones	50
X-B. Recomendaciones	50
XI. Anexos	52
XI-A. Código Completo de la Programación en Arduino	52
XI-B. Programación por Bloques completa de la Interfaz de Usuario	63
XI-C. Propuesta de Manual de Robot	65
XI-D. Propuesta de Guía de Prácticas en Laboratorio con Robot Móvil	79
XI-E. Planos de las piezas	80
XI-F. Ensamble de Modelado 3D de Robot Móvil con Brazo Robótico	92

ÍNDICE DE FIGURAS

1.	Modelo Robótico Unimate	13
2.	Modelo Robótico Móvil Shakey	14
3.	Modelo Robótico OMRON	14
4.	Robot Quirúrgico Da Vinci	15
5.	Robot Explorador Spirit	15
6.	Robot de Rescate UNAM	16
7.	Brazo Robótico Robotnik	16
8.	Modelo de las partes Mecánicas de un Robot Móvil	17
9.	Eslabones de un Robot	17
10.	GDL de un Robot	18
11.	Modelo del Robot Cartesiano	18
12.	Modelo del Robot Cilíndrico	19
13.	Motor Nema 17	19
14.	Modelo de Brazo Robótico 3D	19
15.	Micro Servomotor	20
16.	Pernos y Tuercas	20
17.	Madera MDF	21
18.	Tornillo Autorroscante	21
19.	Ruedas Mecanum	21
20.	Batería Li-Po	22
21.	Controlador DRV8825	22
22.	Módulo Bluetooth HC-05	23
23.	Placa Arduinio Mega	23
24.	Final de Carrera	23
25.	Sensor Ultrasónico HC-SR04	24
26.	Propuesta de Diego González	25
27.	Diagrama de flujo del programa a realizar	29
28.	Diagrama de flujo de la placa PCB	30
29.	Programación por Bloque de la función de enlace mediante Bluetooth	33
30.	Programación por Bloque de la función de visualización de botón de movimiento y transmisión de orden de marcha hacia adelante	34
31.	Programación por Bloque de la Función Automática de movimiento	34
32.	Contrucción de la Interfaz	35
33.	Resultado de la Interfaz (1/2)	36
34.	Resultado de la Interfaz (2/2)	36
35.	Diseño de Placa PCB	37
36.	Diagrama de Conexiones	38
37.	Evidencias de Cálculo de Par Motor y Torque de Carga	40
38.	Marcos de Motor Y Especificaciones Típicas para Motores paso a paso	40
39.	Foto de Prototipo 1/4	44
40.	Foto de Prototipo 2/4	45
41.	Foto de Prototipo 3/4	46
42.	Foto de Prototipo 4/4	47
43.	Programación por Bloques de Interfaz (1/2)	63
44.	Programación por Bloques de Interfaz (2/2)	64
45.	Paso1. Sección A y B	67
46.	Paso2. Sección A y B	67
47.	Paso3. Sección A y B	67
48.	Paso4. Sección A y B	68

49.	Paso5. Sección A y B	68
50.	Paso6. Sección C	68
51.	Paso7. Sección C	69
52.	Paso8. Sección C	69
53.	Paso9. Sección D y E	69
54.	Paso10. Sección D y E	70
55.	Paso11. Sección D y E	70
56.	Paso12. Sección D y E	70
57.	Paso13. Sección D y E	71
58.	Paso14. Sección D y E	71
59.	Paso 1. Conexión de las 4 ruedas omnidireccionales, dos ruedas laterales izquierdas y dos ruedas laterales derechas. Sección A y B	71
60.	Paso 2. Montaje de las ruedas. Sección A y B.	72
61.	Paso 3. Montaje de las ruedas con las varillas de aluminio. Sección C y D.	72
62.	Paso 4. Ya puesta la rueda con los 10 rodillos sigue el acople que está hecho de bronce. como se muestra en la figura señalada. Sección A y B	73
63.	Paso 5. Luego, de poner las ruedas con sus acoples vemos que el acople ya tiene puesto un suprimidor. Sección A y B	73
64.	Paso 6. Una vez terminadas las cuatro ruedas explicaremos cómo se va armando el cuerpo del robot. Sección A y B.	73
65.	Paso 7. Como son ruedas omnidireccionales sus direcciones irían de esta manera. Sección A y B	74
66.	Paso 8. Se van colocando cada motor con su ruedas apropiadas como detalla el paso anterior. Sección A y B.	74
67.	Paso 9. Sección A y B.	74
68.	Paso 10. Sección A y B.	75
69.	Paso 11. Sección A y B.	75
70.	Paso 12. Ya puestos los 4 motores, nos toca colocar las ruedas y cada rueda tiene un acople y su prisionero. Primero con una llave allen vas desajustando, luego colocas en la rueda y así sucesivamente con las otras ruedas.	75
71.	Paso 13. Sección A y B.	76
72.	Paso 14. Ya terminado de colocar todas las ruedas con la llave allen, ahora toca armar la tapa de la caja. Sección A y B.	76
73.	Paso 15. Sección A y B.	76
74.	Conexiones a realizar en la placa PCB.	77
75.	Lámina de la Base del Robot	80
76.	Lámina de la Cintura del Robot	81
77.	Lámina de Brazo del Robot(1/3)	82
78.	Lámina de Brazo del Robot(2/3)	83
79.	Lámina de Brazo del Robot(3/3)	84
80.	Lámina de Base de Agarre	85
81.	Lámina de Base de Engranaje(1/2)	86
82.	Lámina de Base de Engranaje(2/2)	87
83.	Lámina de Base de Enlace de Agarre	88
84.	Lámina de Caja	89
85.	Lámina de Acoplador de Eje	90
86.	Lámina de la Tapa del Robot	91
87.	Robot Móvil Ensamblado	92
88.	Robot Móvil Explosionado	92
89.	Brazo Robótico Ensamblado	93
90.	Brazo Robótico Explosionado	93

91.	Modelado Ensamblado	94
92.	Modelado Ensamblado Explosionado	94

I. PROBLEMA

Uno de los factores de riesgo presente en la robótica son sus elevados precios económicos al realizar modelos robóticos, debido a que el costo por los elementos y materiales tienden a ser altos, pero gracias a los beneficios de la robótica, desde ya hace 30 años han ido en constante evolución y estos se han reducido su precio en la actualidad, pudiendo tener su adquisición de una manera más accesible para las empresas y centros educativos que utilizan a los robots como medios de trabajo.

La educación actualmente se toma como una plataforma que sostiene la productividad económica de un país, ofreciéndonos la sugerencia de que con las nuevas tecnologías se ampliara de una manera más diversa las oportunidades de competencias y mejoramiento de la calidad de vida.

El proyecto, que se está realizando en la investigación actual, reducirá de gran manera el gasto en mano de obra, acelerando la productividad de cualquier empresa, reducirá a su vez el costo de compra de los materiales que estructuran un robot, siendo accesible para el nivel económico promedio y reduciendo el riesgo de peligro de una mala práctica en las instituciones comerciales, educadores o trabajadores también debido a que esta problemática se incrementa día a día, siendo necesaria la adopción de nuevas tecnologías para así crear un beneficio y evitar daños colaterales para todos.[3]

Gracias a esto, aumenta la probabilidad de que se utilice invenciones robóticas para mejorar el incremento del conocimiento, convirtiéndose en una pedagogía más utilizada, a pesar de que los estudios realizados en Ecuador indica que es unos de los países en convertirse lentamente a futuro en un 69 por ciento automatizado, siendo así, un importante recurso el tener en centros de aprendizaje, no solo textos y libros sino de una manera real y tangible el conocer las estructuras y conformación de un robot.

“Cada vez se propone destacar cómo la robótica en el aula de clase permite, por una parte, el enriquecer estrategias de aprendizaje como apoyo a la formación integral de los estudiantes, y por otra, es un campo que presenta una demanda creciente en la atención de docentes e investigadores, así como en los estímulos para su investigación, desarrollo y divulgación de parte del estado.”[6]

Se considera como un efecto a producirse el desarrollo de este proyecto en un nivel de aprendizaje con avances para las ciencias, ingenieras, matemáticas, tecnología. Ya que esto ayuda a comprender mejor a los estudiantes el manejo de las nuevas tecnologías al aplicarse en futuras generaciones junto con los nuevos modelos de enseñanza dados a futuro en los colegios, escuelas e instituciones universitarias.

Por lo cual, se propone el diseño de un robot móvil controlado desde una red bluetooth, que tiene como objetivo principal la recolección y posicionamiento de objetos de manera manual y automática, haciendo uso de un smartphone como controlador.

Esto aportaría a la comunidad entera, en este caso el enfoque es el proceso que conlleva la planificación, diseño de la aplicación con su programación y su testeo en tiempo real movilizandoo figuras en 3d, brindando un aporte académico a instituciones educativas como escuelas y colegios, siendo de gran contribución para los estudiantes que a futuro opten por la opción de dedicarse a la robótica.

II. JUSTIFICACIÓN

El desarrollo del presente trabajo de titulación tiene como objetivo dar a conocer los avances tecnológicos en la educación, la cual se centra en dos aspectos: la formación integral de los estudiantes que permite la convivencia con los robots en las escuelas educativas y las herramientas necesarias que consiste en el diseño, análisis y aplicación para su correcta operación. Como lo menciona Muñoz [8], la robótica promueve el desarrollo de habilidades de programación desde una edad escolar temprana, logrando adquirir un rol creativo en el uso de las tecnologías.

Por otro lado, Stephen Hawking, menciona “La inteligencia como la capacidad de adaptarse al cambio”, hace énfasis en los descubrimientos, refiriéndose a esto como un cambio radical. Por eso, ha ido surgiendo diversas formas para que los alumnos adquieran conocimientos. Explorar y conocer sobre la robótica en las instituciones educativas permite el crecimiento individual.

La robótica en el ámbito educativo es considerada una herramienta esencial para adquirir nuevos conocimientos en los estudiantes, ayuda a fomentar la creatividad en ellos y se aprende desde el diseño, la experimentación de proyectos tecnológicos, mediante la interacción y el aporte de sus compañeros. Como lo mencionan López, Losada y Paz [4], “Los sistemas robóticos van ocupando más parcelas de nuestra actividad cotidiana en todos los ámbitos y van superando las rigideces de construcción y programación.”. Esto se evidencia en la curiosidad de los alumnos, que permite potenciar su pensamiento crítico y mejorar sus habilidades mediante el aprendizaje. La robótica en la educación se ven reflejadas diversas áreas, tales como: matemáticas, tecnología, ciencias e ingeniería. Es recomendable que desde temprana edad los niños empiecen a utilizar herramientas como el Lego, figuras en 3d, figuras geométricas, con el fin de que los alumnos comiencen a manipular diversos materiales para desarrollar su creatividad y percepción. Con el transcurso del tiempo, el estudiante se va a familiarizar con las dinámicas de la robótica, por ende, sus retos empiezan a tener dificultad, lo que le llevará aprender hacer circuitos eléctricos, maquetas, diseños, etc. Es allí donde se evidencia la confianza y la capacidad que tiene cada estudiante.

III. OBJETIVOS

III-A. OBJETIVO GENERAL

Desarrollar un robot móvil controlado remotamente con sistema de recolección de objetos que facilite la práctica de robótica para fines educativos.

III-B. OBJETIVOS ESPECIFICOS

- Investigar el tipo de robot móvil utilizado para fines educativos.
- Desarrollar el prototipo mecatrónico del robot, en conjunto con el sistema de recolección de objetos.
- Controlar el robot de forma remota mediante una aplicación móvil.
- Realizar el manual de prácticas de laboratorio con el robot móvil para docentes y estudiantes.

IV. MARCO HIPOTÉTICO

IV-A. Hipótesis

Este proyecto priorizará el aumento del discernimiento de nuevas tecnologías a los docentes y alumnos. Por consiguiente, la robótica educativa creará nuevos modelos de aprendizaje las cuales mejorarán el proceso de adquisición de conocimiento permitiendo que los estudiantes se desarrollen en las áreas de Ciencias, Tecnología y Matemática a medida que se fomentará una pasión para los estudiantes que vean a futuro esta carrera como profesión.

V. MARCO TEÓRICO REFERENCIAL

V-A. Introducción

La robótica es una de las ramas tecnológicas que sirve de utilidad para la vida del ser humano, ya que facilita tareas que represente peligro, cansancio o disminución de eficiencia en calidad y cantidad de producción. La robótica cumple diversos campos como es la ingeniería en automatización, control industrial, ingeniería mecánica e incluso dentro del campo educativo, gracias a las posibilidades de aprendizaje que ofrece la comprensión de esta rama.

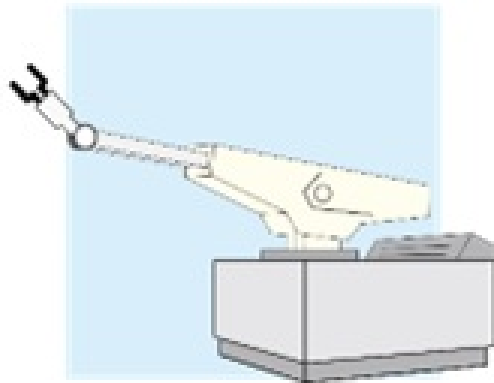
V-B. Historia De La Robótica

Los principios y(u) orígenes de la robótica móvil autómatas e industrial data de la idea misma de Aristóteles inventando herramientas automatizadas, continuando con sus investigaciones Leonardo Da Vinci diseñando un prototipo de caballero mecánico, hasta logra su verdadero desarrollo en la automatización de la robótica a mediados del siglo XX. Se considero un robot móvil autómatas a las decisiones variantes de un operador robótico sin necesidad de supervisión física.

En los años 50 en Inglaterra ELSIE (compañía de electrónica y ciencias), desarrollo el primer modelo de robot móvil de la historia, sus componentes y capacidades eran muy limitados para la época, pero realmente se trataba de un robot electromecánico sensible a la luz con firme y sólido externa e interna en su estructura.

Con el pasar del tiempo en el año 1954 se llevaría a cabo en EE. UU. el que se consideraría el primer robot industrial con estructura de un brazo mecánico hidráulico llamado Unimate especializado en el sector automotriz.

Figura 1. Modelo Robótico Unimate



Nota: El Unimate original constaba de una gran caja computarizada, unida a otra caja que se conectaba a un brazo articulado, con un programa de tareas almacenado en una memoria de tambor.

Ya en los últimos años de los 60 y comienzo de los 70, fue donde se logró un notable cambio en la robótica, en donde se diseñaron prototipos con implementación de sensores de movimientos con luces infrarrojas y cámaras. Unos de los más notables inventos de esa época el robot Shakey en 1966 por el instituto de la universidad de Standford es un hito en la robótica móvil gracias a que su hardware y software permitían entender y comprender su entorno, aunque de forma limitada.

Figura 2. Modelo Robótico Móvil Shakey



Nota: Shakey fue el primer robot móvil de propósito general en ser capaz de razonar sobre sus propias acciones. Fue el primer Robot controlado por Inteligencia Artificial.

En los años 80 y finales de los 90 donde la robótica móvil se expandió por todo el mundo teniendo sus propios prototipos y avances no solo en EE. UU., sino en Asia, Europa, Japón y Suecia.

En la actualidad fabricar robot móvil sufre cambios constantes por las necesidades de las empresas, centros de investigación de cada país y sus características que cada vez son mayores en precisión, Con mejor toma de decisiones teniendo un mantenimiento predilecto de estos inventos.

V-C. Robots Móviles

Un robot móvil es una máquina automática capaz de moverse en un entorno específico. Según (Revista de Robots, 2022) señala que un robot móvil es una combinación de sistemas mecánicos y electrónicos que se trasladan de manera autónoma controlado por medio de un software”. Estos robots son muy útiles ya que hacen tareas que requieren desplazar algo de un lugar a otro. En este apartado describimos el estado de arte de los robots móviles más vistos.

V-D. Tipos De Robots Móviles

Para identificarse con un robot móvil debemos tener en cuenta que tienen diversas capacidades y características que diferencian a cada una de estas invenciones. Hay distintos tipos de movilidad en ruedas modelos orugas y con patas las cuales se mostrarán en ejemplos a continuación:

V-D1. Robot OMRON: El robot móvil OMRON se especializa en el desempeño de transporte, entrega y enrutamiento de cargas u objetos con geo-localizador y diseño con ruedas para la movilización.

Figura 3. Modelo Robótico OMRON



Nota: El robot colaborativo, aporta destreza, flexibilidad y tiene la capacidad de resolver tareas.

V-D2. *Robot Quirúrgico Da Vinci*: Este robot fue diseñado para realizar cirugías invasivas en operaciones como el cáncer y extracción de tumores. Consta de precisión en sus brazos robóticos articulados, una cámara en 3D, un agarre sostenible y fuerte para maniobrar los objetos quirúrgicos.

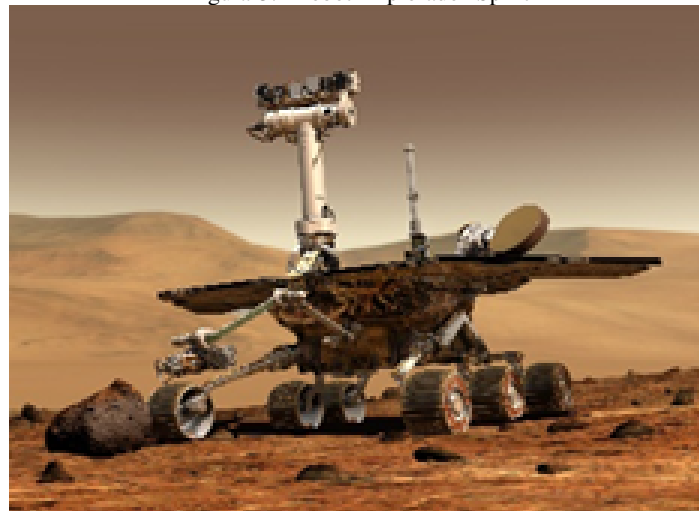
Figura 4. Robot Quirúrgico Da Vinci



Nota: Es un robot esclavo, que obedece al cirujano a la vez que aumenta su capacidad para operar con precisión y destreza, reduciendo el temblor y proporcionando una visión excepcionalmente clara de la anatomía del paciente.

V-D3. *Robot Explorador Spirit*: Unos de los primeros robots móviles creados por la Nasa para el estudio del planeta Marte, sus características es la emisión de video, desplazamiento a todo terreno mediante una banda transportadora con ruedas y agarre de objetos.

Figura 5. Robot Explorador Spirit



Nota: Es el primero de los dos robots que forma parte del Programa de Exploración de Marte de la NASA.

V-D4. *Robot De Rescate UNAM*: Es un robot de rescate capaz de desplazarse en terrenos extremos, moverse de manera automática para la localización de víctimas potenciales de cualquier desastre o accidentes que consta de herramientas, luces infrarrojas, cámara de video y ruedas resistentes a cualquier ambiente.

Figura 6. Robot de Rescate UNAM



Nota: Los robots de rescate están diseñados especialmente para ayudar en la búsqueda y el rescate de personas tras un desastre.

V-D5. Brazo Robótico Robotnik: Con un sistema de agarre multieje capaz de la acomodación y suspensión de objetos, tareas automátatas con precisión especialmente utilizado para partidas de ajedrez.

Figura 7. Brazo Robótico Robotnik

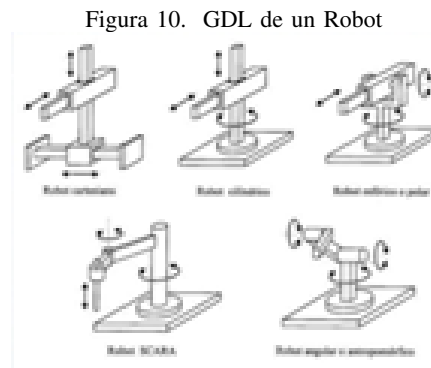


Nota: Un brazo robótico es un dispositivo programable cuyas funciones principales y comportamiento se asemejan a las de un brazo humano.

V-E. Estructura Mecánica Del Robot Móvil

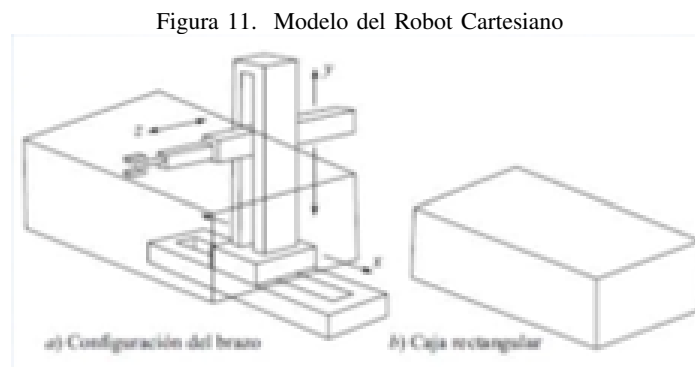
V-E1. Morfología Del Robot: Un robot está formado por los siguientes elementos: estructura mecánica, transmisiones, sistema de accionamiento, sistema sensorial, sistema de control y elementos terminales.

Las combinaciones más usadas son la que representa en la siguiente imagen donde se ve las tres principales articulaciones del robot, que son las más importantes porque son como la base principal que le da posición al robot en un punto central.



Nota: Son los grados de libertad de un robot.

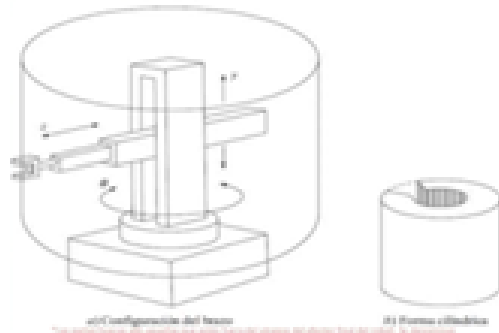
V-E3. Cartesiano: De modo cartesiano, es cuando el brazo del robot se mueve en las direcciones de las coordenadas X_1 , Y_1 y Z_1 como el sistema de coordenadas cartesianas rectangulares como se ve en la imagen siguiente. Este modelo se lo conoce como robot cartesiano y la base del robot tiene la forma de un prisma rectangular.



Nota: Un robot cartesiano es una serie múltiple de ejes que se desplazan cada uno de ellos de forma lineal, y entre los diferentes ejes conforman ángulos rectos unos respecto de otros.

V-E4. Cilíndrico: Cilíndrico, es cuando el brazo del robot tiene una articulación de revolución y dos prismáticas, los puntos que pueden alcanzar pueden ser convenientes especificados con coordenadas cilíndricas.

Figura 12. Modelo del Robot Cilíndrico

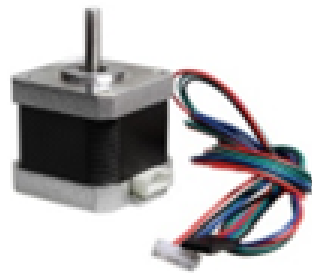


Nota: Es un robot cuyos ejes forman un sistema de coordenadas cilíndricas.

V-F. Componentes Mecánicos

V-F1. *Nema 17*: Nema 17, es un motor muy preciso que se utiliza para hacer varias aplicaciones diferentes. Con este motor paso a paso se podrá controlar el giro de su eje con mucha precisión y así obtendrá el movimiento de tu máquina o robot.

Figura 13. Motor Nema 17



Nota: Motor de precisión, cuando se enfatiza movimientos lentos y controlables.

V-F2. *Modelo del Brazo Robótico en 3D*: Hay muchas formas y estilos de modelos de brazos robóticos en 3D, pero la mayoría funcionan con el mismo principio de generar movimientos. Los brazos robóticos son únicos en el sentido de que no están limitados por su tamaño y, como tales, ocupan pocas dimensiones en comparación con otros modelos de máquinas similares como las modelos cartesianas que son las impresoras en 3D ya son grande para un área de trabajo y limitadas por un sistema de coordenadas específicas para su movimiento.

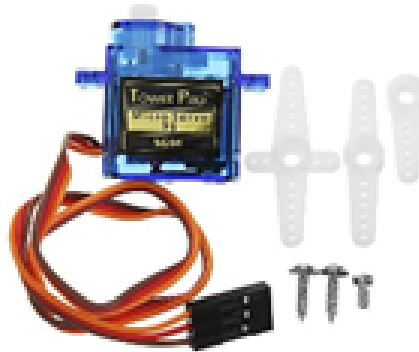
Figura 14. Modelo de Brazo Robótico 3D



Nota: El diseño 3D ayuda a tener una mejor idea del producto final.

V-F3. *Micro Servomotor SG90*: Es un motor que permite un control de precisión en posición angular ya que puede rotar hasta 180°, su voltaje de operación es de los 4.8 a 6VDC. Este modelo está confirmado por 3 brazos, 3 tornillos y cuenta con un cable de hasta 25cm convirtiéndose en un excelente actuador rotativo de cualquier diseño robótico a implementarse.

Figura 15. Micro Servomotor



Nota: Ideal para movimientos no tan demandantes con alto control en sus grados de libertad.

V-F4. *Pernos y Tuercas de M3 y M4*: Son elementos de vital importancia dentro de la tecnología ya que su utilización es para unir piezas de forma permanente o temporal ya que permiten el ensamble y desmontable de todo tipo de prototipo a realizar ya que la mayoría de las estructuras se unen con pernos y tuercas.

Figura 16. Pernos y Tuercas



Nota: Muy utilizados en estructuras de todo tipo.

V-F5. *Piezas de Madera MDF de 8mm*: Las piezas de madera MDF se utilizan normalmente para trabajos y completar varios tipos de acabados siendo de contextura resistente, puedes realizar la personalización de acabado de la madera a tu gusto adaptándose al uso que quieras darle al tablero MDF.

Figura 17. Madera MDF



Nota: Material resistente y económico dependiendo del acabado.

V-F6. *Tornillos Autoroscantes:* Es una varilla maciza de acero cincado con rosca en su diseño tiene una forma de broca. se utiliza normalmente para enroscar el metal sin la necesidad de hacer un orificio al metal ya que crea su propia rosca y se fija en los objetos, chapas o elementos de metales y es recomendable taladrar primero con una broca del tamaño apropiado por sus distintos grosores, cabezas y modelos como avellanados, hexagonales y cilíndricos.

Figura 18. Tornillo Autorroscante



Nota: Agujera la superficie mientras se fija en ella.

V-F7. *Plataforma de Ruedas:* La plataforma rodante es una base que se desplaza sobre ruedas y que habitualmente cuenta con un tirador o asa para reducir el esfuerzo en el transporte de la carga. Se presenta como una herramienta que se adapta a variedad de usos y situaciones que requieren del transporte de paquetes y mercancías sobre ruedas.

Figura 19. Ruedas Mecanum



Nota: Su ventaja se encuentra en la alta tracción.

V-G. Componentes Eléctricos

V-G1. *Batería LI-PO*: Es una batería compuesta por (litio y polímero) siendo de modelo recargable utilizado a prestaciones superiores a otras como las NiCd y NiHm almacenando grandes cantidades de energía y fabricando se a medidas personalizadas para su uso.

Figura 20. Batería Li-Po



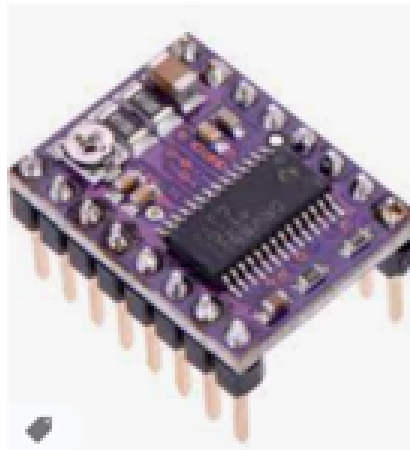
Nota: Dispositivo para la energización de circuitos

V-H. Componentes Electrónicos

V-H1. *Controlador DRV8825*: El DRV8825 es un controlador de motor paso a paso que tiene una mayor capacidad de corriente, también se lo conoce como A4988 de Allegro por su uso en máquinas CNC en miniatura. El DRV8825 es ideal para usar en aplicaciones de impresión 3D y otras máquinas CNC de tamaño medio.

En este caso el chip DRV8825 este puesto sobre una placa modular se puede insertar en shields para Arduino o también se puede colocar sobre un protoboard para realizar pruebas directas con el chip.

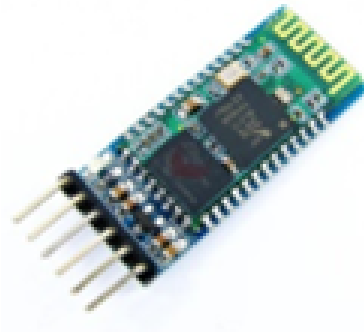
Figura 21. Controlador DRV8825



Nota: Un dispositivo de protección extra para estructuras móviles.

V-H2. *Módulo Bluetooth HC-05*: El módulo Bluetooth HC-05 permite conectar el prototipo robotico con Arduino a un Smartphone, celular o PC de forma inalámbrica, se conecta de forma directa a los pines seriales de los microcontroladores, está placa incluye un regulador de 3.3v Permitiendo alimentar al módulo con un voltaje de 3.6v hasta 6v. Con la facilidad de operación y transmisión transparente a los programadores.

Figura 22. Módulo Bluetooth HC-05



Nota: Su tamaño compacto es una ventaja al momento de aplicarlo.

V-H3. Placa Arduino Mega: Arduino mega es una placa que contiene todo lo necesario para dar soporte a los microcontroladores es ideal para proyectos ya que permite controlar grandes cantidades de sensores contando con conexión USB, conector de alimentación, botón de reinicio más de 54 pines de entrada y salida digital, 16 entradas analógicas de 4 UARTS Y muchas características para el buen uso del control robótico.

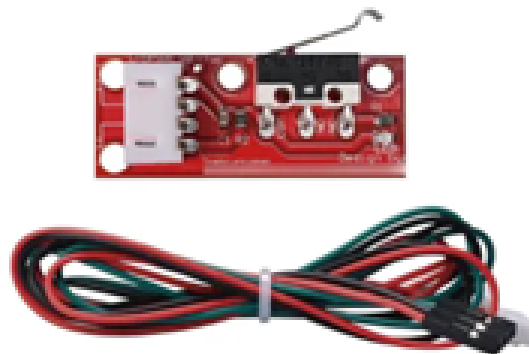
Figura 23. Placa Arduinio Mega



Nota: Tarjeta programable que permite aplicaciones automatizadas.

V-H4. Final De Carrera: Este tipo de sensor se acciona mecánicamente, cuando un objeto entra en contacto directo con el sensor de final de carrera a una determinada posición este envía una señal que puede ser aprovechada por otros dispositivos para realizar respectivos comportamientos.

Figura 24. Final de Carrera



Nota: Sensor mecánico usado para limitar el movimiento físico de un mecanismo.

V-H5. *Sensor Ultrasonido HC-SR04*: Este sensor es utilizado para medir distancias de objetos con respecto al sensor, con una distancia efectiva entre unos 2cm a 450cm. Funciona de tal manera que el emisor piezoeléctrico emite pulsos de ultrasonido que viajan en el aire y al chocar contra algún tipo de objeto regresan, tomando el tiempo que tardan los pulsos en volver el microcontrolador calcula la distancia estimada del objeto.

Figura 25. Sensor Ultrasonico HC-SR04



Nota: El sensor ultrasónico mide distancias cortas.

V-I. *La Robótica Orientada A La Educación*

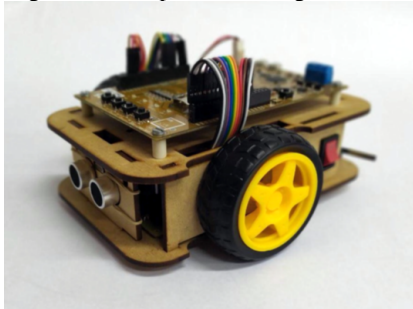
La robótica como tal era parte del sílabo de carreras específicas de educación superior, gracias al avance de la tecnología y educación, la complejidad de esta rama fue volviéndose más asequible a grados intermedios y básico, como indica el Ministerio de Educación de Argentina [9], la educación se encuentra en un constante desafío debido a los saltos agigantados de la tecnología.

Es una propuesta muy atractiva desde una perspectiva educativa por los amplios temas que conlleva, ciencias, matemáticas, física y tecnología, no quedando corta solo en un análisis teórica, sino llevando cada uno de los conceptos a su aplicación en el diseño, desarrollo y programación de los robots, llegando así a estar acorde con los postulados ofrecidos por la UNESCO.[5]

Además de pulir y reforzar los conocimientos en las áreas científicas, el mayor beneficio de implementar estas prácticas en los estudiantes es incentivar al estudiante que sea participe en un rol activo como diseñador y constructor de su propio aprendizaje, moldee su creación al entorno haciendo uso de su imaginación en conjunto de herramientas físicas y de simulación para innovar.[5]

Los robots móviles son una herramienta didáctica y atractiva, debido al entusiasmo que causa en los estudiantes y a la diversidad de disciplinas tecnológicas que involucran, tales como: mecatrónica, electrónica, eléctrica, mecánica, sensores, control, procesamiento de señales e imágenes, comunicación inalámbrica, programación, inteligencia artificial, entre otras. Como menciona Martín en su trabajo, en primordial atraer estudiantes y nuevas generaciones con proyectos que sean fáciles de entender, pero no escatimen en el potencial académico que pueden ofrecer, siguiendo los pasos que le motivó a presentar su robot móvil en el 2019 “EduRoMAA”, es primordial seguir ofreciendo a la comunidad alternativas de aprendizaje sobre sistemas que en un principio podrían considerar complejos, pero que con el tiempo, lo vean como algo con una facilidad relativa para entretenerse creando en un fin de semana. [7]

Figura 26. Propuesta de Diego González



Por supuesto que esta práctica no se puede realizar en un solo ciclo escolar, por lo que es necesario segmentar el nivel teórico que se imparte, por un lado, para evitar abrumar al estudiante, y por otro, para distribuir en una escala de menor a mayor cuáles son los requerimientos académicos previos para adentrar al estudiante por completo en la robótica.

Desde el nivel primario, se enseña al estudiante a reconocer y familiarizarse con dispositivos electrónicos y computarizados que ya existen en su día a día, llevándolos por el camino de comprender que tipos de información son las que utilizan, como la traducen para aplicarla en sus acciones y que comportamientos internos ocurren para realizarlas. Con esto, el alumno es abierto a imaginar en que otros tipos de situaciones aquellos mecanismos pueden ser implementados o despertar la curiosidad en animarlos a descubrir que otras formas estaban antes invisibles ante sus ojos. A través de los años en los niveles superiores de educación, ahora viendo de una manera distinta el mundo que les rodea en conjunto con los conocimientos académicos adquiridos, serán capaces de traer innovación a las diferentes áreas en las que decidan desempeñarse. [10]

En el Ecuador, aunque algunas instituciones contemplan a esta rama dentro de su sílabo de trabajo, aún el país como tal está lejos de implementarlo como un estándar nacional, aun así, gracias a kits económicos preparados por los distribuidores de estos implementos, permite que cada vez más instituciones se animen a tratar de impartir talleres donde los alumnos puedan instruirse en un nivel básico de programación robótica.

V-J. Arduino En La Robótica Educativa

En tiempos actuales, las herramientas educativas de robótica están más al alcance que nunca, siendo por ejemplo el software de uso libre “Scratch” permite un entorno de juego/aprendizaje al estudiante, enseñándole uno de los métodos adecuados para el programador principiante, lo que es la programación por bloques, dando paso al entendimiento de la lógica de programación, métodos simples de accionamiento de órdenes y bucles, páginas dedicadas al entorno de la programación robótica hacen uso de lenguaje “Arduino” como una manera amigable de enlazar la programación con la electrónica, permitiendo hacer una realidad avanzadas. [12]

El lenguaje Arduino como tal es un software de uso libre, diseñado para ser entendido y escrito con facilidad, con una gran comunidad que cada día crece más, no solo en número, sino también en cooperación, gracias a que cuenta con unas de las librerías online mas grandes de la red. La cual todos pueden subir sus aportes y cualquiera puede hacer uso de los códigos dependiendo de sus necesidades. A un nivel de marca supo dejar su huella igualmente debido a una gama importante de controladores y dispositivos en los cuáles probar, usar y armar proyectos simples o complejos con ellos, desde ventas por unidad o kits completos para propósitos didácticos, sin duda Arduino marcó un antes y después para toda persona o institución interesada en descubrir el mundo de la robótica.

V-K. Apps En El Control De Robots

El ser humano siempre encuentra nuevas e innovadoras formas de ayudarse en sus actividades diarias, el campo de la robótica no es la excepción, en sus inicios estos tenían una programación estática en la cual solo cumplían las acciones dentro de su programación y para algo nuevo esta debía ser reingresada, si se buscaba algún tipo de manejo manual, las radio señales o directamente cables cumplían esta función, algo rústico para los estándares de hoy, con esto claro, en la actualidad avances como lo son el bluetooth y el wifi expanden más el abanico de posibilidades,

debido que permite controlar diversos dispositivos usando funciones que se encuentran en smartphones por defecto, estas funciones no cuentan con una conexión directa a dispositivos electrónicos que reciben estas señales porque requieren una programación previa, aquí es donde entra a escena una herramienta de uso libre desarrollada por el “Massachusetts Institute of Technology” llamada MIT APP INVENTOR, esta herramienta da la posibilidad a programadores principiantes o avanzados a que puedan desarrollar aplicaciones de control o análisis para su smartphone, siempre y cuando estos tengan como sistema operativo Android o iOS. Dentro de la propia herramienta el programador tiene acceso a programación por bloques, diseño de la pantalla de su aplicación, desde botones para enviarles señales de cualquier tipo o recibirlas; hasta imágenes o figuras para decorarla, como herramienta para proyectos o enseñanza didáctica, tiene un buen atractivo para el educador o alguien que busque crear sus propias invenciones.

VI. MARCO METODOLÓGICO

VI-A. *Diseño De La Investigación*

El diseño de investigación es la acción de que manera será aplicado el conjunto de estrategia y metodología definida previamente al desarrollo de una investigación. Como Hernández menciona que “Su interés se centra en explicar por qué ocurre un fenómeno y en qué condiciones se manifiesta” [11]. Generalmente se llama este diseño como un plan o estructura de investigación, con el fin de tener una estrategia para la aplicación del proyecto de prototipos de robótica para un fin didáctico, buscando la manera más eficiente para su desarrollo y aplicación en todos los campos educativos.

VI-B. *Materiales Didácticos*

La robótica móvil se la utiliza para diversos entornos y posee varias actividades que le facilita hacer tareas manuales y autónomas. Para llevar a cabo el funcionamiento del robot, se lo realiza mediante una aplicación móvil de un celular, siendo está gratuita que nos permite el manejo y el control del mismo. Posee motores tanto para el brazo como para las ruedas omnidireccionales, estos motores dotan las capacidades del robot y pueden ser modificadas. Asimismo, realiza actividades de acuerdo a su tipo de traslado, es decir la manera que deseamos que se desplace de un punto específico a otro. El enfoque que adaptamos al robot es para un ambiente estudiantil, donde podemos encontrar: laboratorio de robótica, kits electrónicos, herramientas y materiales necesarios para la construcción del robot. El método que utiliza para transportarse es terrestre, para lo cual, el piso debe ser de caucho o se puede usar tapetes que contengan una pequeña fricción, puesto que las ruedas omnidireccionales, no permite usar cualquier piso liso. Las actividades que realiza el robot puede ser la clasificación de traslado de objetos con colores, trasladar objetos pequeños elaborados por una impresión en 3d y colocarlo en su sitio indicado, asimismo, podemos controlar los movimientos de manera automática del robot, lo que se desea hacer en cada paso, guardar esos movimientos y accionarlos, etc. [10]

VI-C. *Modalidad De La Investigación*

Para la investigación se enfoca en dos modelos siendo el cuantitativo y cualitativo. La metodología cuantitativa consiste en ser una investigación científica que incluye descripción, medición y explicación de un fenómeno (Álvarez, 2011). Por otro lado, es una representativa experimental que se hace en una población o fenómeno objeto de estudio. Los datos obtenidos son numéricos y estadísticos, siendo simuladores de gráficos o tablas con el objetivo de cuantificar, siendo beneficioso para el análisis del proyecto mediante la recolección de datos dependiendo el mercado educativo que se desea aplicar.

La metodología cualitativa es un espacio con variedad de disciplinas que aporta en la producción de investigación porque contiene una variedad métodos para la con selección y análisis de datos con información detallada.[11]

Esta parte se enfoca de un método fácil para interpretar, describir y comprender datos recopilados mediante la observación de entrevistas, notas de campo o descripciones. En el caso el proyecto mostrara los resultados del uso de los robots pedológicos mediante las entrevistas y notas de campo que se realiza durante la investigación.

VI-D. *Tipo De Investigación*

Existe una variedad de tipo de investigación, pero el proyecto se basa en el estudio transversal, siendo una investigación observacional que recolecta para describir variables y sus incidencias. Los datos obtenidos se darán a conocer durante el periodo de estudio. Estas variables darán a conocer la población de muestra durante su periodo y como resultado se reconocerá el comportamiento del mercado educativo al tener esta tecnología innovadora para su uso.

VI-E. Población Y Muestra

La población es el conjunto de elemento sobre lo que se va a investigar, en este caso la unidad educativa, que tendrá un grupo de observadores que sería la muestra, siendo esta el promedio del estudio. Esto será con un tipo de población real siendo con un grupo tangible, y la muestra sería aleatorio simple con elemento en la lista al lazar siendo más homogéneo y eficaz. Por otro lado, se realiza un muestreo no probabilístico. “Generalmente es sobre la base de un buen conocimiento de los estratos de la población y/o de los individuos más representativos.^o .adecuados” para los fines de la investigación” [1]. Durante este periodo el estudio se hará en “Universidad Politécnica Salesiana” y se analiza el muestreo con los miembros de ese lugar.

VI-F. Técnica De La Investigación

Es el conjunto de herramienta y procedimiento utilizado para tener la información y conocimiento dependiendo la metodología seleccionada. Se utiliza la técnica de campo siendo una observación directa al objetivo del estudio, es decir el método de observación para comprender. La otra técnica es la entrevista para análisis del comportamiento siendo en un ambiente de forma oral y personalizada.

VI-G. Instrumentación De La Investigación

Los instrumentos es el elemento que ayuda a extraer la información [2]. Se distingue esto en forma y contenido, es decir que se establece la forma como información de datos empíricos y como contenido la serie de indicadores que ayuda para el método de observación mediante preguntas.

Un instrumento es el cuestionario, donde se recoleta la información mediante los datos de tabulación con su construcción de preguntas abiertas y cerradas, con esto se podrá obtener la información previa para la ejecución de problema.

VII. MARCO PROCEDIMENTAL

VII-A. Diagrama de flujo de la lógica para la programación del Robot

Figura 27. Diagrama de flujo del programa a realizar

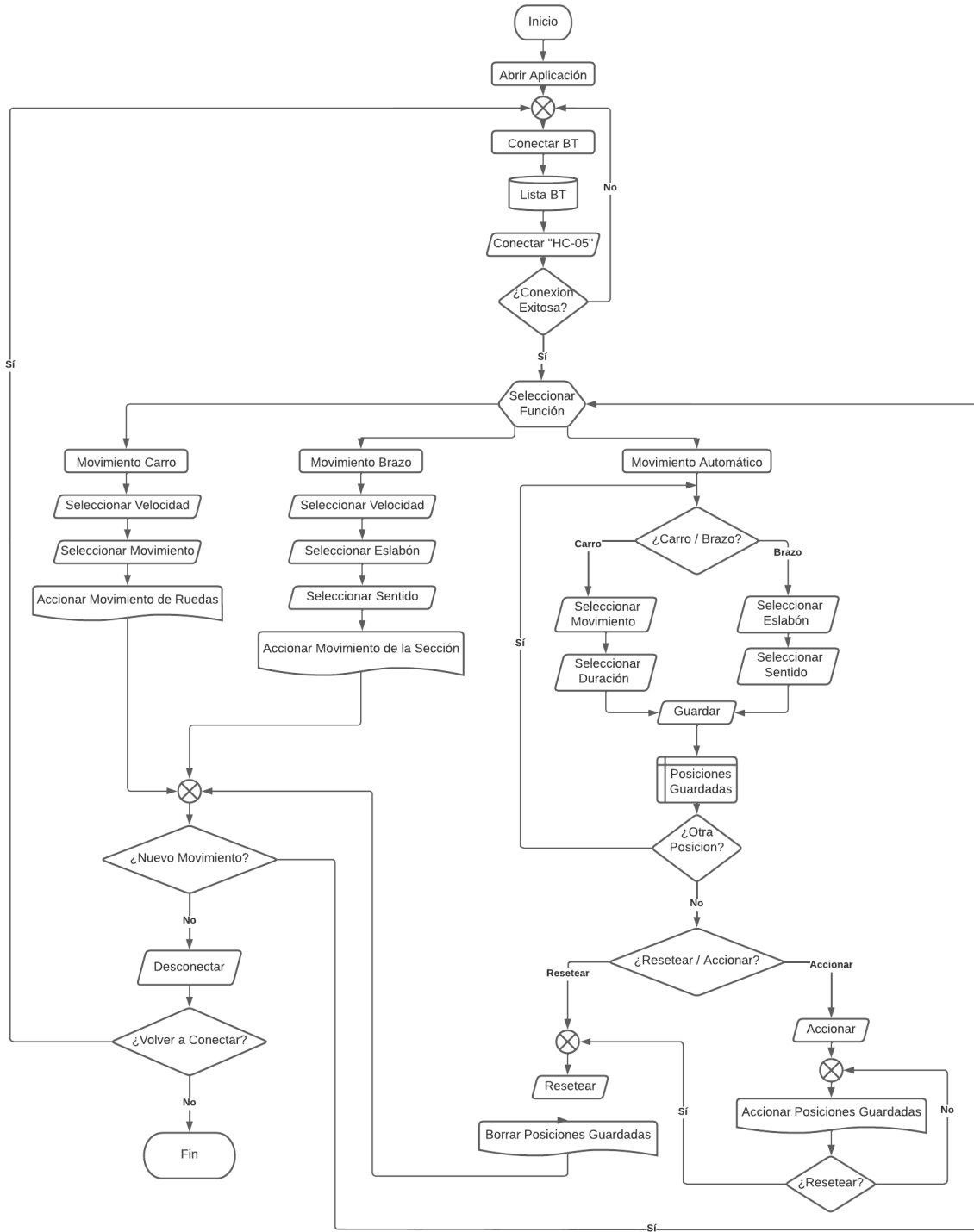
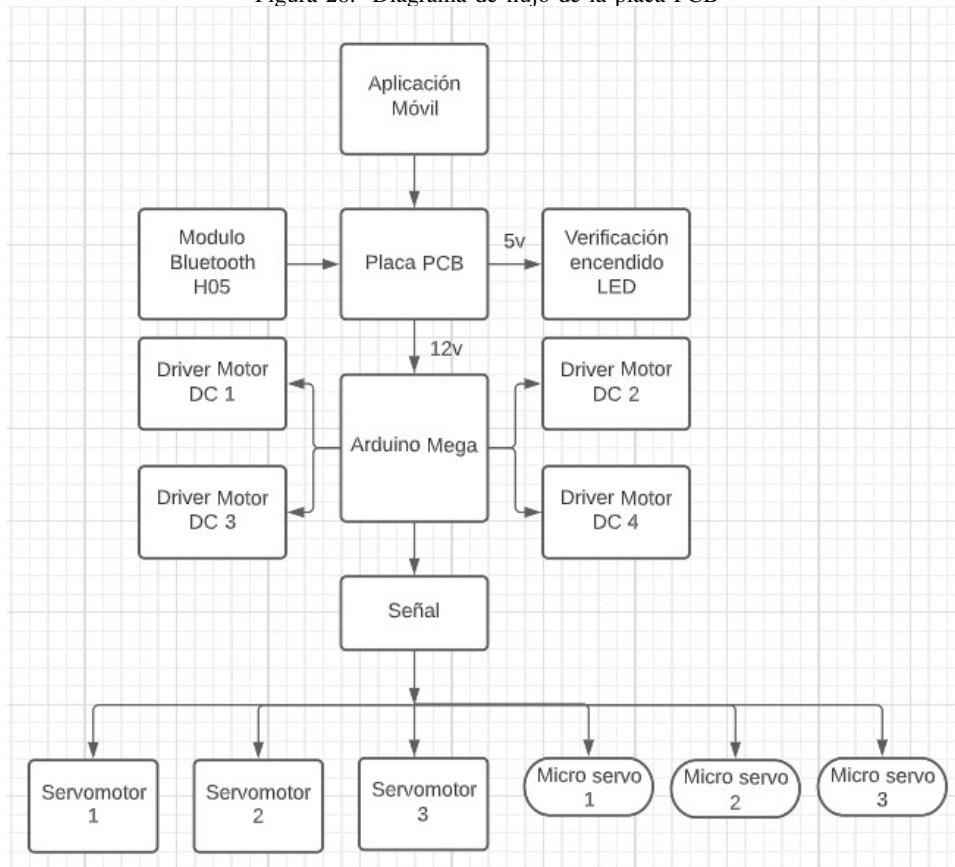


Figura 28. Diagrama de flujo de la placa PCB



VII-B. Diagrama de la placa electrónica

VII-C. Lógica a utilizar en la programación de la placa Arduino

Como existen en muchos compiladores de programación, no hace falta programar todo desde cero, la existencia de librerías en la que basan las funciones que facilitan el control de los procesos, en el proyecto al contar con motores paso a paso y servomotores, las librerías de estos dos son imprescindibles.

Dentro del entorno es necesario introducir dichas librerías con el comando #include

Introducción de Librerías

```

1 #include <AccelStepper.h>
2 #include <Servo.h>
          
```

La estructura típica de un programa en arduino es la llamada de librerías, definir las variables a utilizar y por ultimo configurar las dos secciones que configuran el programa como tal "Void Setup" & "Void Loop". Cada tipo de variable posee su propia estructura a seguir para una correcta lectura de la misma por parte del compilador y el programa.

Ejemplo Definir Motores Paso a Paso

```
1 // Se definen los motores paso a paso y los pines a utilizar
2 AccelStepper LeftBackWheel(1, 46, 48); // (Driver, STEP, DIR) - Motor1
3 AccelStepper LeftFrontWheel(1, 52, 50); // Motor2
4 AccelStepper RightBackWheel(1, 32, 34); // Motor3
5 AccelStepper RightFrontWheel(1, 38, 36); // Motor4\\
```

Al momento de establecer las variables y emplear, también se deben aclarar las matrices que servirán para guardar datos, como posiciones y velocidades.

Definición de Variables para Motor Paso a Paso

```
1 int wheelSpeed = 1500;
2 int lbw[50], lfw[50], rbw[50], rfw[50]; // Matriz para guardar posiciones/pasos
3 int speedDelay = 20;
4 int index = 0;
5 int dataIn;
6 int m = 0;
```

Dentro del primer bucle, "Void Setup", como el nombre indica, en este espacio las variables establecidas y matrices obtienen sus valores iniciales.

Ejemplo Establecer valores iniciales de motores paso a paso

```
1 LeftFrontWheel.setMaxSpeed(3000);
2 LeftBackWheel.setMaxSpeed(3000);
3 RightFrontWheel.setMaxSpeed(3000);
4 RightBackWheel.setMaxSpeed(3000);
```

Definición de Variables a manera de Salida

```
1 pinMode(led1, OUTPUT);
2 pinMode(led2, OUTPUT);
3 pinMode(led3, OUTPUT);
```

El segundo y último bucle "Void Loop", es el programa de control como tal, en el cual van todas las funciones que afectan las variables y en él recae la activación y el apagado de nuestros implementos.

Funcion de Comunicación BlueTooth

```
1 if (Serial3.available() > 0) \{\n2   dataIn = Serial3.read();
```

Es posible establecer funciones de comunicación con el módulo bluetooth

Con la funcion de comunicación se combina con la funcion `if` para que dependiendo de que orden reciba, actúa acorde a la lógica de control, como es activar uno de los servomotores

Funcion de Comunicación BlueTooth para activar servomotor

```
1   if (dataIn == 16) {\n2     m = 16;\n3   }\n4   // Movimiento del servo 1 en direccion positiva\n5   while (m == 16) {\n6     if (Serial3.available() > 0) {\n7       m = Serial3.read();\n8     }\n9     servo01.write(servo1PPos);\n10    servo1PPos++;\n11    delay(speedDelay);\n12  }
```

VII-D. Estableciendo la lógica de la interfaz a utilizar para el Robot en MIT APP INVENTOR

Figura 29. Programación por Bloque de la función de enlace mediante Bluetooth

```
when BluetoothList . BeforePicking
do set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

when BluetoothList . AfterPicking
do if call BluetoothClient1 . Connect
    address BluetoothList . Selection
then set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

when Clock1 . Timer
do if BluetoothClient1 . IsConnected
then set Connected . Text to "Conectado"
    set Connected . TextColor to blue
if not BluetoothClient1 . IsConnected
then set Connected . Text to "Desconectado"
    set Connected . TextColor to red

when Disconnect . Click
do call BluetoothClient1 . Disconnect

when BluetoothClient1 . BluetoothError
functionName message
do call BluetoothClient1 . Disconnect
if not BluetoothClient1 . IsConnected
then set Connected . Text to "Desconectado"
    set Connected . TextColor to red
```

Se observa en el primer bucle el programa llama una lista de direcciones bluetooth desde el dispositivo móvil, consecuentemente se habilita la selección de una dirección que se encuentre en dicha lista, existe un bucle de tipo timer para tener una confirmación visual si se ha logrado con éxito conectar con el dispositivo bluetooth, siendo el HC-05, de lo contrario permanecerá en estado de "Desconectado", por último tenemos una condicional para cuando se requiera desconectar del dispositivo, siendo en forma de un botón "Desconectar"

Figura 30. Programación por Bloque de la función de visualización de botón de movimiento y transmisión de orden de marcha hacia adelante

```

when RightForward .Click
do
  if RightForward .Image = RightForward_0.png
  then
    set RightForward .Image to RightForward_1.png
    set Left .Image to Left_0.png
    set Right .Image to Right_0.png
    set Backward .Image to Backward_0.png
    set LeftForward .Image to LeftForward_0.png
    set Forward .Image to Forward_0.png
    set LeftBackward .Image to LeftBackward_0.png
    set RightBackward .Image to RightBackward_0.png
    set RotateLeft .Image to RotateLeft_0.png
    set RotateRight .Image to RotateRight_0.png
    call BluetoothClient1 .Send1ByteNumber
    number 3
  else
    set RightForward .Image to RightForward_0.png
    call BluetoothClient1 .Send1ByteNumber
    number 0
  
```

En el bucle existe un establecimiento de que sin importar cuando se accione el botón de marcha hacia adelante, no afectará a la visualización del resto de botones direccionales, aun cuando se presione una sola vez, mandará una señal de 3 al módulo bluetooth, activando dicho movimiento, de lo contrario, manda una señal de 0 para desactivarlo.

Figura 31. Programación por Bloque de la Función Automática de movimiento

```

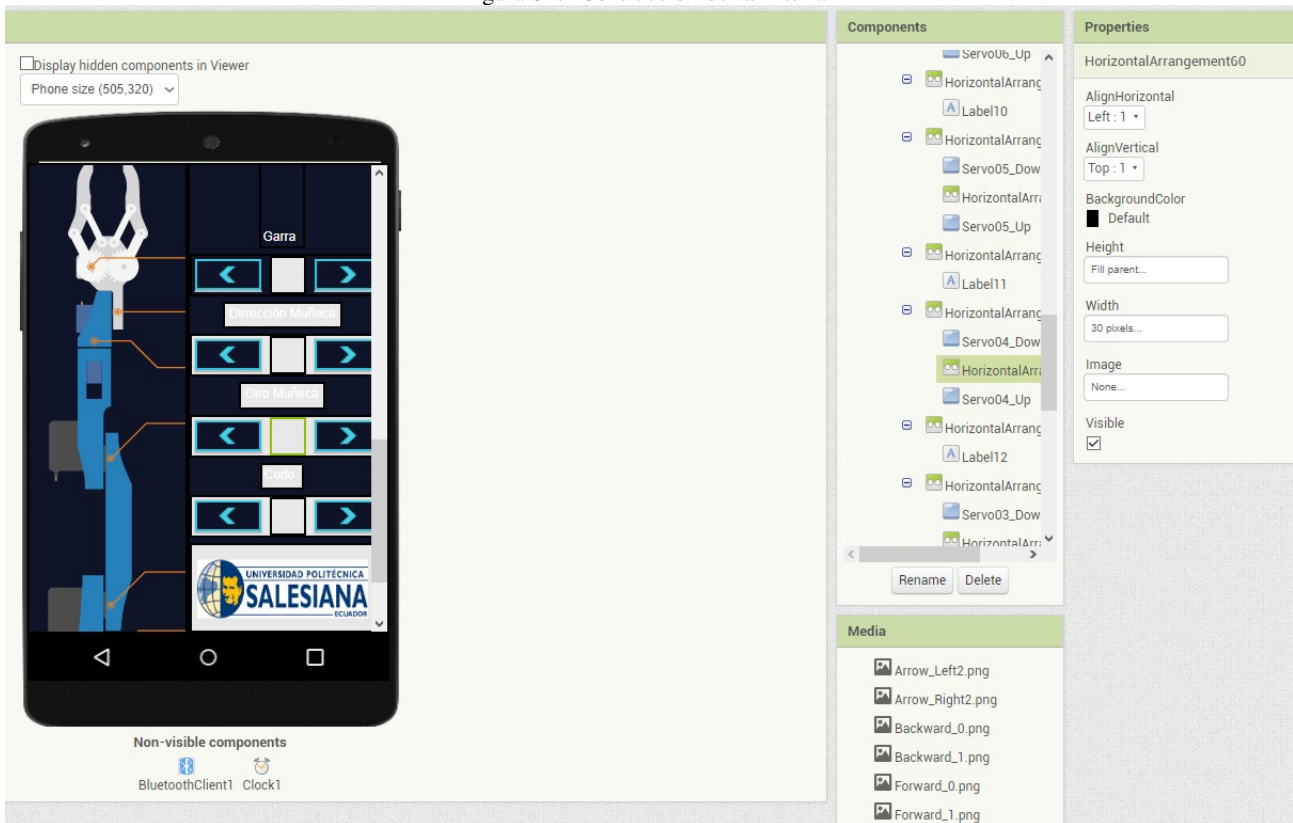
when Save .Click
do
  call BluetoothClient1 .Send1ByteNumber
  number 12
  set Positions .Text to Positions .Text + 1

when Reset .Click
do
  call BluetoothClient1 .Send1ByteNumber
  number 13
  set Run .BackgroundColor to #cccccc
  set Positions .Text to 0

when Run .Click
do
  if Run .Text = Corriendo
  then
    set Run .Text to Pausado
    call BluetoothClient1 .Send1ByteNumber
    number 14
    set Run .BackgroundColor to #00ffff
  else
    set Run .Text to Corriendo
    call BluetoothClient1 .Send1ByteNumber
    number 15
  
```

El movimiento automático funciona de tal manera que al presionar el botón de guardar en la aplicación, se envía el movimiento o posición a una lista para luego ser utilizada, de ser necesario, es posible borrar todos los datos de la lista con la función de RESET, cuando se tiene todas las posiciones y movimientos deseados, se puede accionar en el orden que se guardaron mediante el botón ACCIONAR en la aplicación.

Figura 32. Contrucción de la Interfaz



Dentro del entorno de MIT APP INVENTOR se cuenta con diferentes tipo de figuras, listas y opciones de texto para armar la visualización de la aplicación, los sliders son la mejor opción para controlar los ángulos de los servos, y considerando una mejor percepción por parte del usuario, se incluyen flechas guía, el logotipo y el dibujo del brazo es un detalle meramente estético pero importante al momento de considerar el público al que se encuentra orientado

La interfaz cuenta con botones con la función de conectarse y desconectarse del módulo bluetooth, seguidos por los respectivos botones de movimiento para acabar la primera sección con un slider de velocidad y tres botones que guardan, esta acción y resetean las matrices de movimientos posiciones para que el robot pueda realizarlos en secuencia.

En la segunda sección de la interfaz de usuario se encuentran los controles de movimiento para cada eslabón del brazo robótico, un slider adicional para indicar la velocidad del movimiento y un diagrama representativo de a eslabón pertenece en el brazo real.

Figura 33. Resultado de la Interfaz (1/2)

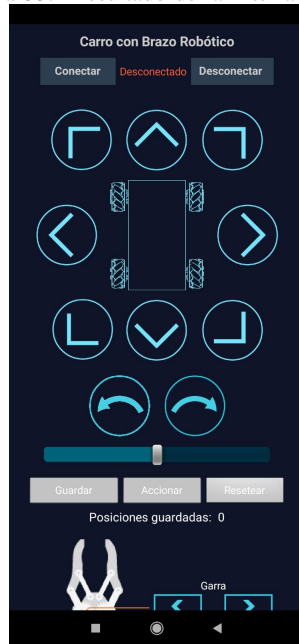
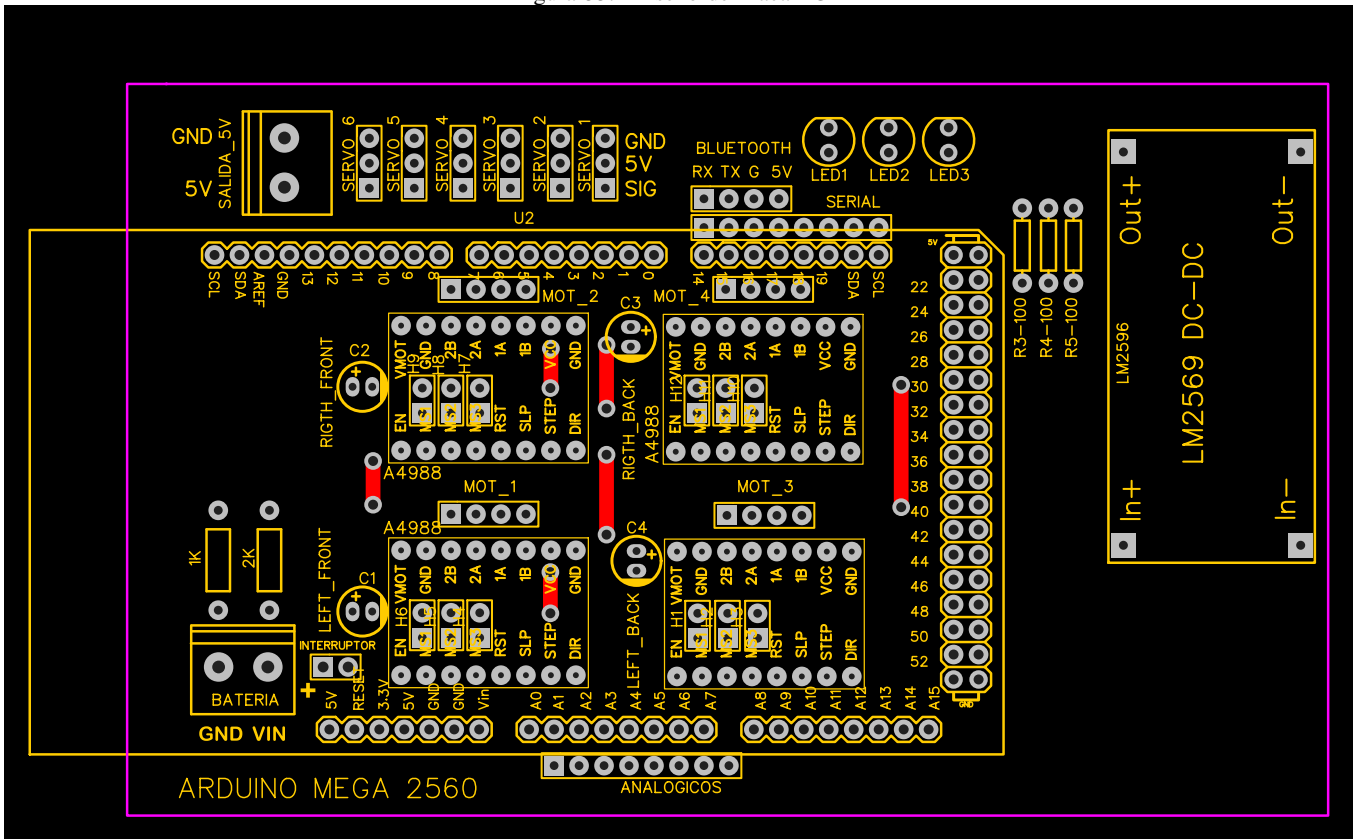


Figura 34. Resultado de la Interfaz (2/2)



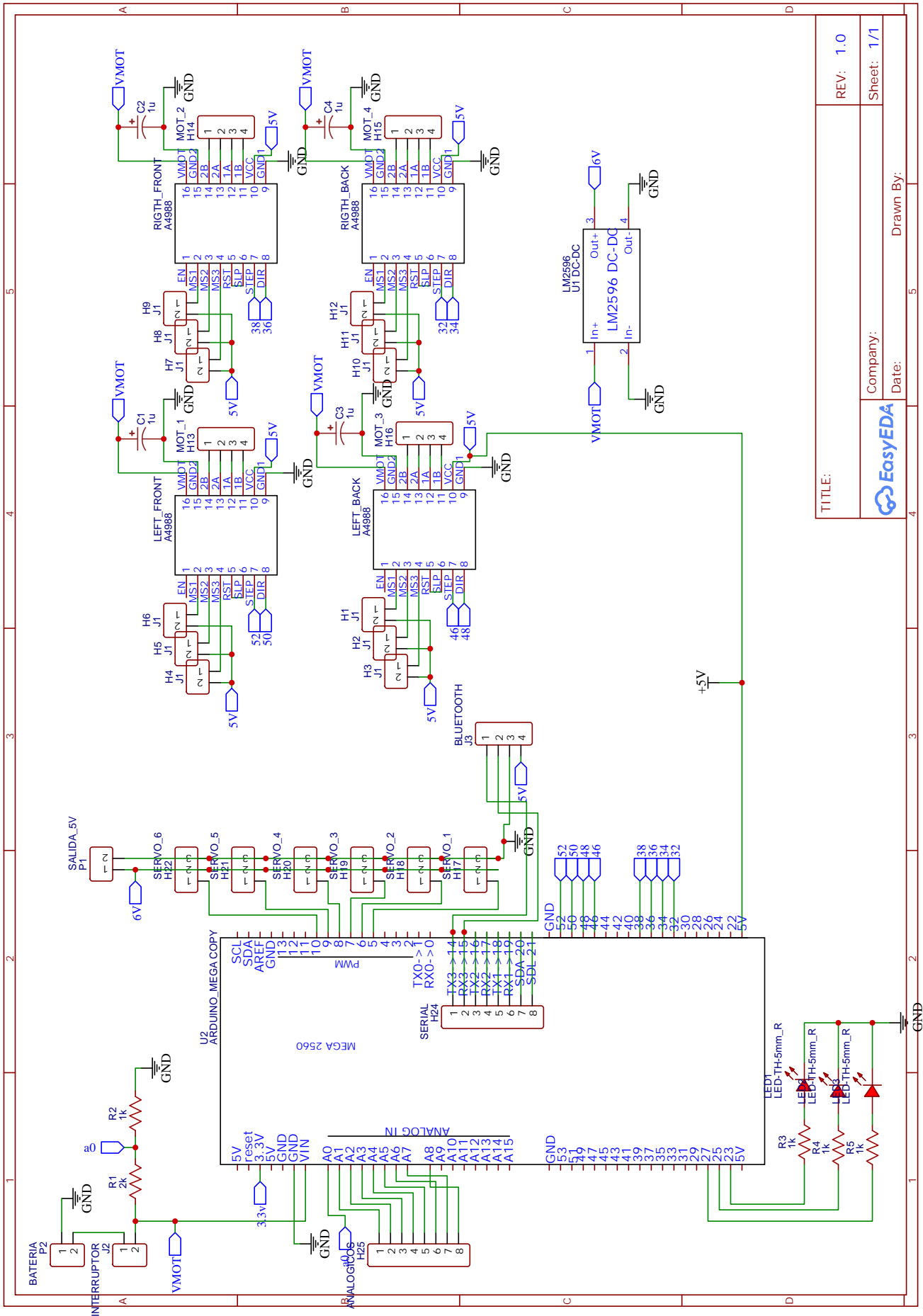
VII-E. Diagrama Electronico de Placa PCB

Figura 35. Diseño de Placa PCB



Analizando la placa PCB, su posición será sobrepuesta en la placa Arduino Mega, en el diseño del cuadro con margen morado representa la placa PCB como tal, y el recuadro amarillo representa la placa Arduino Mega, desde de la esquina inferior izquierda podemos visualizar que tenemos una entrada, la misma que esta diseñada para recibir el voltaje de la batería lipo que entrega 11.1V y esta puede aguantar hasta una carga de 12V, continuando por el lado inferior izquierdo central, tenemos los pines de la placa que irán a las respectivas entradas y salidas de la placa Arduino Mega, en la parte central es donde se distribuyen cada driver que esta controlado por los motores paso a paso Nema 17 con conjunto con sus respectivos capacitores para contrarrestar cualquier posible caída de voltaje, hacia el lado derecho dentro del espacio de la placa Arduino se encuentran los pines de comunicación, seguido de estos tenemos el espacio para un regulador de voltaje, para convertir el voltaje de entrada a 5V, por ultimo en la parte superior tendremos pines de conexión que facilitan el control del brazo robótico del proyecto.

Figura 36. Diagrama de Conexiones



TITLE:	Company:	Drawn By:
	EasyEDA	
REV: 1.0	Date:	Sheet: 1/1

En el diagrama de conexiones se hace uso de 2 voltajes, una de entrada de 11.1v; El cual es usado mayormente por los motores paso a paso en conjunto con el sistema de ventilación y un voltaje que se regula en la placa de 5v para la placa arduino, servomotores del brazo robótico como lo son los MG996R y SG90, e indicadores led de encendido. El regulador de voltaje es LM2596, que puede manejar hasta 4 amperio de corriente. Los 6 servos del brazo consumen un rango de hasta 3 amperio de corriente, siendo una carga manejable para el dispositivo, en caso de algún tipo de sobrecalentamiento, el sistema de ventilación envía una constante ráfaga de aire a través de la placa. Los motores paso a paso cuentan con su respectivo driver, los cuales tuvieron que ser calibrados a su previo uso y cuentan con un disipador de calor en caso de sobrecargas, un dato importante a tomar en cuenta con los drivers es la función que cumple cada pin, conectar la alimentación en pines incorrectos daña de manera inmediata dicho driver, para una mejor manipulación de la conexiones, los pines de la placa PCB esta alineados con los de la placa Arduino Mega

VII-F. Dimensionamiento del motor Nema a utilizar en el proyecto

Mediante el uso de una balanza se realizó la medición del peso de las piezas del proyecto, con el fin de encontrar el motor nema adecuado sin sobredimensionarlo o sobrecargarlo. A continuación se detalla los datos que fueron necesarios en el cálculo:

Peso del Carro	2580 gr
Diametro Externo de la Rueda	76.4 mm
Peso de la Rueda	105 gr
N# de Ruedas	4
Coefficiente de friccion	0.2
Eficiencia del Sistema	80 %
Factor de Seguridad	1.5

Los valores de peso deben ser llevados a masa antes de poder ser utilizados en posteriores cálculos:

$$(m)Carro = \frac{2580gr}{9,81 \frac{m}{s^2}} = 262,99gr$$

$$(m)Rueda = \frac{105gr}{9,81 \frac{m}{s^2}} = 10,70gr$$

Estos valores fueron introducidos en un software libre uso para calcular los valores de par motor y torque que podrían afectar al sistema.

Figura 37. Evidencias de Cálculo de Par Motor y Torque de Carga

Sizing Results			
Load Inertia	J_L	=	0.4115 [kg·m ²]
Required Speed	V_m	=	4.190 [r/min]
Required Torque	T	=	2.414 [N·m]
Acceleration Torque	T_a	=	0.1806 [N·m]
Load Torque	T_L	=	1.429 [N·m]
Required Stopping Accuracy	$\Delta\theta$	=	7.543 [deg]

El torque de carga esta en 1.429 N*m, cada motor nema tiene un valor de placa que se puede consultar, como la tabla a continuación:

Figura 38. Marcos de Motor Y Especificaciones Típicas para Motores paso a paso

Frame Size	Diameter (mm)	Typical Torque Range for a Stepper Motor (Nm)	Typical Speed Range for a Stepper Motor (RPM)
NEMA 8	20	0.01 - 0.04	0-1000
NEMA 11	28	0.06 - 0.12	0-1000
NEMA 14	35	0.05 - 0.5	0-1000
NEMA 16	39	0.1 - 0.25	0-1000
NEMA 17	43	0.2 - 1	0-1000
NEMA 23	57	0.5 - 3	0-1000
NEMA 24	60	1.2 - 4.6	0-1000
NEMA 34	86	3 - 12	0-1000
NEMA 42	102	12 - 20	0-1000

Los motores Nema 17 tienen un valor de torque entre 0.2Nm y 1Nm, al tomar en cuenta las 4 ruedas resulta que el torque de carga necesario para el sistema se encuentra dimensionando correctamente para la elección.

VII-G. Dimensionamiento del Brazo Robótico en el proyecto

Lo primero a considerar es el torque del brazo, debido que el mismo constantemente durante su uso será sometido a esta fuerza física.

$$\tau = (F)(d)(\sin\theta)$$

Donde (F) es la fuerza aplicada, (d) es la distancia del punto de giro y theta el ángulo de aplicación de la fuerza, siendo máximo para un ángulo de 90 grados y nulo para un ángulo de 0 grados.

$$\tau = (mg)(d)(\sin\theta)$$

La fuerza se define como el peso de la masa sujeta al servo, esta es por tanto la impresión 3D, el peso de los otros servos y la carga

$$\tau = (mg)(d)$$

Se considera el torque máximo, la masa está dada por la densidad del material PLA por el volumen de la impresión. Del modelo 3D se obtienen las distancias y masa de las impresiones 3D.

Eslabón del Brazo Robótico	Peso
Base	89 gramos
Brazo 1	84 gramos
Brazo 2	73 gramos
Brazos 3	21 gramos
Base de la Garra	20 gramos
Garra	27 gramos

Los servos utilizados desde las hojas de datos del fabricante tienen estas características:

Modelo	Peso	Torque Máximo
MG996R	55 gramos	11 kg cm
SG90	14.7 gramos	2.5 kg cm

Conforme se aumentan elementos 3D y servos, aumenta el peso y por tanto el toque que van a realizar los motores desde el eslabón de la garra hasta la base. Los cálculos se resumen a continuación:

Torque para eslabón de Garra

$$\tau = (27\text{gramos}) * (7,8\text{cm}) = 210,6\text{gcm}/1000\text{kg} = 0,210\text{kgcm}$$

Torque para eslabón de Base de Garra

$$\tau = (61,7\text{gramos}) * (6,5\text{cm}) = 401,05\text{gcm}/1000\text{kg} = 0,401\text{kgcm}$$

Torque para eslabón de Brazo 3

$$\tau = (97,4\text{gramos}) * (5,5\text{cm}) = 535,7\text{gcm}/1000\text{kg} = 0,536\text{kgcm}$$

Torque para eslabón de Brazo 2

$$\tau = (185,1\text{gramos}) * (11,3\text{cm}) = 2091,63\text{grcm}/1000\text{kg} = 2,092\text{kgcm}$$

Torque para eslabón de Brazo 1

$$\tau = (324,1\text{gramos}) * (15,8\text{cm}) = 5120,78\text{grcm}/1000\text{kg} = 5,121\text{kgcm}$$

Torque para eslabón de Base

$$\tau = (468,1\text{gramos}) * (5\text{cm}) = 2340,5\text{grcm}/1000\text{kg} = 2,341\text{kgcm}$$

Eslabón del Brazo Robótico	Peso Total	Distancia	Torque Máximo
Garra	27 gramos	7.8 cm	0.210 kg cm
Base de garra	61.7 gramos	6.5 cm	0.401 kg cm
Brazo 3	97.4 gramos	5.5 cm	0.536 kg cm
Brazo 2	185.1 gramos	11.3 cm	2.092 kg cm
Brazo 1	324.1 gramos	15.8 cm	5.121 kg cm
Base	468.1 gramos	5 cm	2.341 kg cm

En definitiva, los motores pueden aguantar el peso de la estructura del brazo robótico, el peso máximo que se puede añadir de cargar al brazo se calcula de la siguiente manera:

$$\tau = (\text{peso})x(\text{distancia}) = (\text{peso.brazo} + \text{peso.carga})x(\text{distancia})$$

$$\begin{aligned} \text{peso}_c &= \frac{\tau_{\text{maximo}}}{\text{distancia}} - \text{peso}_b \\ \text{peso}_c &= \frac{2,5\text{kgcm}}{7,8\text{cm}} - 27\text{gramos} \\ \text{peso}_c &= 320,5\text{gramos} - 27\text{gramos} \\ \text{peso}_c &= 293,5\text{gramos} \end{aligned}$$

VII-H. Dimensionamiento de la Batería a utilizar en el proyecto

Mediante el uso de una fuente de poder dispuesta por la universidad se llegó a realizar pruebas de cuanto amperaje utilizaba el circuito una vez se instalaron todos los implementos necesarios y que corrieran sin novedades, resultando en la medición de un consumo de 1.5A en el pico del uso del circuito, siguiendo un dimensionamiento del 30 % se obtuvo lo siguiente:

$$D(30\%) = (1,5A) + (1,5A)(0,3) = 1,5A + 0,45A = 1,95A$$

Se redondea a 2A, esta es la base para analizar la capacidad de la batería, no exagerando en cuestión de espacio, se implementa una batería lipo con las siguientes características:

Voltaje	11.1V
Celdas	3
Capacidad	4000mA

Con los datos obtenidos se puede calcular el tiempo mínimo de descarga que tiene el robot en su totalidad

$$TiempoDescarga = \frac{AmperiosHora}{ConsumoCorriente} = \frac{4000mAh}{2000mA} = 2horas$$

VII-I. Prototipo Ensamblado

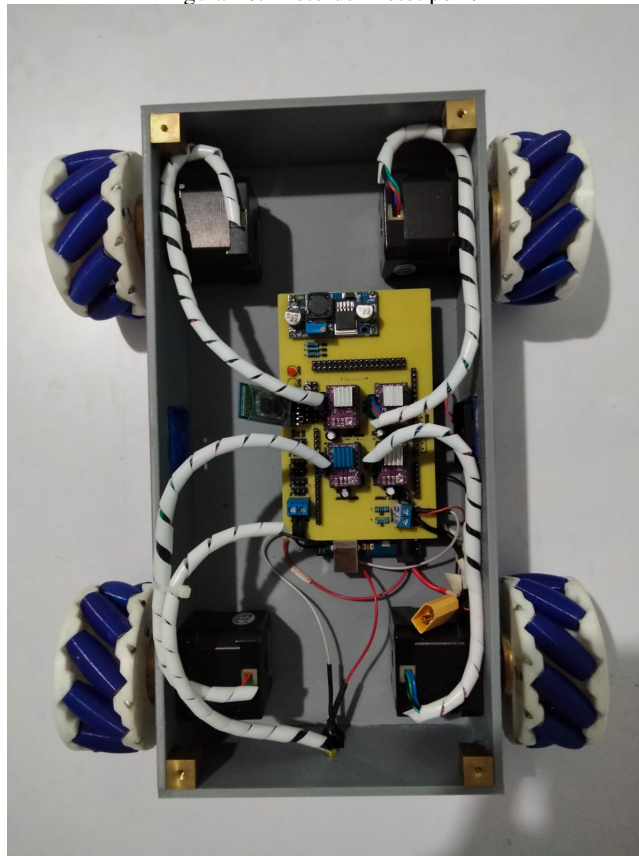
A continuación se muestra el ensamble del prototipo con las piezas impresas y hechas con sus respectivas conexiones.

Figura 39. Foto de Prototipo 1/4



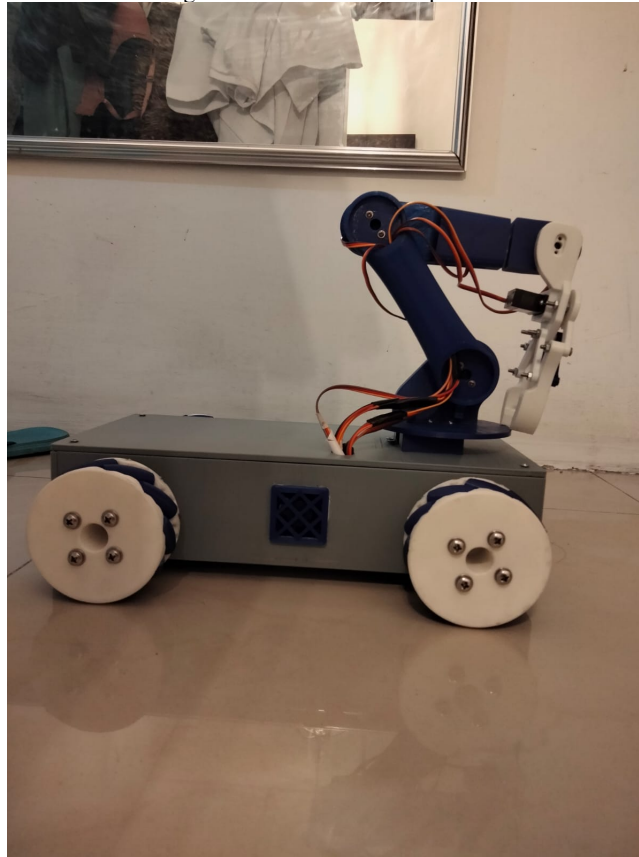
Observamos, el modelo de un brazo robótico con 6 grados de libertad, diseñado en SolidWorks e impreso en 3d con un relleno de 40 por ciento de PLA. Se controlara el brazo remotamente con una aplicación móvil. Sus colores son azul con blanco y realizara la función de recolección de objetos como: juegos geométricos, letras, numero y todo en impresión 3d.

Figura 40. Foto de Prototipo 2/4



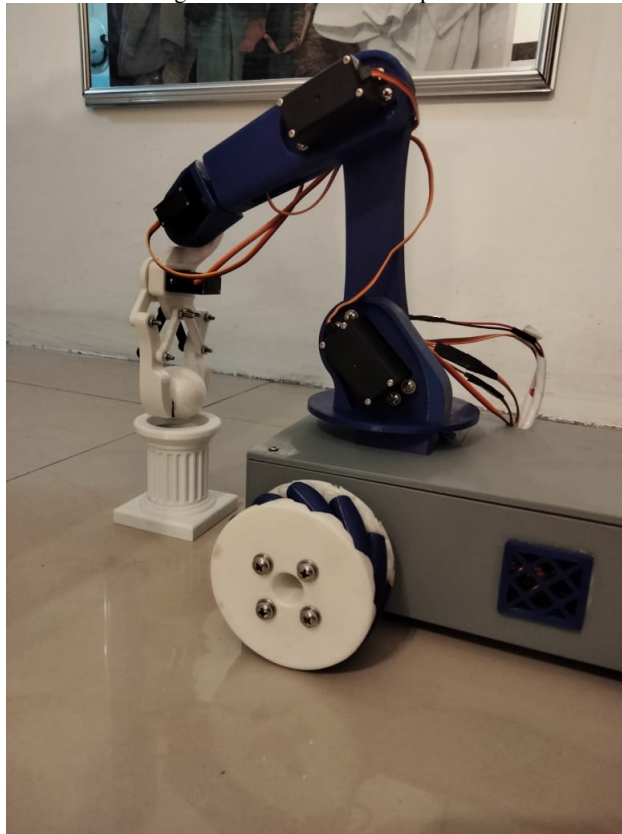
A continuación podemos ver en la imagen el esqueleto del robot móvil con sus 4 acoples que se utiliza para colocarlo en cada rueda omnidireccional además, tiene un motor paso a paso ya que, es una gran opción para trasladarse hasta cierta distancia sea de manera manual o automática, diseñado en SolidWorks e impreso en 3d con un relleno de 40 por ciento de PLA. Tiene 10 movimientos para desplazarse y es una de las partes más principales del robot.

Figura 41. Foto de Prototipo 3/4



En la siguiente sección podemos observar el prototipo ya ensamblado con todas sus piezas impresas en 3d, los 4 acoples de bronce y sus pernos que conllevan cada pieza.

Figura 42. Foto de Prototipo 4/4



Por último, en esta imagen se aprecia una de las actividades que se logra hacer con el robot, el ejemplo que se presento fue que con las pinzas del brazo del robot permite el agarre de una pelota para posteriormente ubicarla en una bandeja.

VIII. CRONOGRAMA

SEDE - GUAYAQUIL		CRONOGRAMA DE TRABAJO																					
		AUTOR 1		JOSELYN CAROLAI BURBANO MERINO										AUTOR 2		LUIS ALBERTO CARDENAS LEON							
CAMPUS - CENTENARIO		UNIVERSIDAD POLITÉCNICA SALESIANA																					
CARRERA - MECATRÓNICA		DESARROLLO DE UN ROBOT MÓVIL CON SISTEMA DE RECOLECCIÓN DE OBJETOS CONTROLADO REMOTAMENTE																					
TEMA DE TRABAJO DE TITULACIÓN		Meses/Semanas																					
N	ACTIVIDADES	may-09		jun-09		jul-09		ago-09		sep-04													
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4						
1	Desarrollo de documentación científica del proyecto																						
2	Adquisición de componentes para la fabricación de la máquina																						
3	Modelado, cálculo mecánico y diseño en 3D del prototipo																						
4	Ensamblaje de prototipo del Robot																						
5	Diseñar e imprimir la placa PCB																						
6	Programación del Robot																						
7	Implementación del software con el Robot																						
8	Pruebas del brazo robótico																						
9	Análisis de resultados																						
7	Revisión calidad del producto final																						
10	Presentación de proyecto																						

IX. PRESUPUESTO

GASTOS DEL ROBOT				
MATERIAL O ELEMENTO	UNIDAD DE MEDIDA	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
MOTOR PASO A PASO - NEMA 17	UNIDAD	\$ 13,00	4	\$ 52,00
CONTROLADOR PASO A PASO DRV8825	UNIDAD	\$ 4,50	4	\$ 18,00
SERVOMOTOR MG996R	UNIDAD	\$ 9,00	3	\$ 27,00
MICROSERVOMOTOR SG90	UNIDAD	\$ 3,50	3	\$ 10,50
MÓDULO BLUETOOTH HC-05	UNIDAD	\$ 7,50	1	\$ 7,50
BATERÍA LI-PO	UNIDAD	\$ 26,00	1	\$ 26,00
PLACA ARDUINO MEGA 2560 REV3 [A00067]	UNIDAD	\$ 29,00	1	\$ 29,00
PLACA PCB PERSONALIZADA	UNIDAD	\$ 15,00	1	\$ 15,00
DISIPADOR DE CALOR ALUMINIO	UNIDAD	\$ 0,50	4	\$ 2,00
VENTILADOR DE 12 V CC PEQUEÑO	UNIDAD	\$ 5,00	1	\$ 5,00
KIT DE TORNILLO, PERNOS, TUERCAS Y ARANDELAS	UNIDAD	\$ 6,00	1	\$ 6,00
BRUJIYA SUPER BONDER	UNIDAD	\$ 2,00	4	\$ 8,00
LED	UNIDAD	\$ 0,05	2	\$ 0,10
CAPACITORES	UNIDAD	\$ 0,50	6	\$ 3,00
RESISTENCIAS	UNIDAD	\$ 0,10	10	\$ 1,00
JUEGO DE JUMPER	UNIDAD	\$ 3,00	3	\$ 9,00
ROLLO DE ALAMBRE DE ACERO DE 3 mm	UNIDAD	\$ 1,75	1	\$ 1,75
BROCA FORSTNER DE 25 mm	UNIDAD	\$ 0,30	1	\$ 0,30
TALADRO	UNIDAD	\$ 38,00	1	\$ 38,00
SIERRA DE MESA	UNIDAD	\$ 0,30	1	\$ 0,30
PLA	UNIDAD	\$ 30,00	2	\$ 60,00
CALIBRADOR	UNIDAD	\$ 1,60	1	\$ 1,60
ALICATE CORTE FRIO	UNIDAD	\$ 4,96	1	\$ 4,96
LIJA UNIÓN HIERRO 060	UNIDAD	\$ 0,75	1	\$ 0,75
PINZA CORTADOR DE ALAMBRES	UNIDAD	\$ 6,80	1	\$ 6,80
FUNDA DE AMARRAS TRANSPARENTES	UNIDAD	\$ 1,50	1	\$ 1,50
ROLLO DE FILAMENTO	UNIDAD	\$ 30,00	2	\$ 60,00
CREALITY ENDER 6	UNIDAD	\$ 650,00	1	\$ 650,00
AGUA	GLOBAL	\$ 1,00	10	\$ 10,00
TRANSPORTE	UNIDAD	\$ 5,00	10	\$ 50,00
REFRIGERIO	UNIDAD	\$ 4,00	10	\$ 40,00
MATERIALES DE OFICINA	GLOBAL	\$ 5,00	10	\$ 50,00
TOTAL GASTOS				\$ 1.195,06

X. CONCLUSIÓN Y RECOMENDACIONES

X-A. Conclusiones

- Las investigaciones de los prototipos del brazo robótico móvil en el ambiente escolar, nos arrojó un resultado muy favorable en la tasa de desarrollo educacional, debido a que, muchas instituciones ya han implementado modelos tecnológicos para mayor aprendizaje en los niños y jóvenes. Y este proyecto ayudará al costo, siendo económicamente accesible para las instituciones educativas y necesario para entender el funcionamiento aplicado en el arquetipo de ciencia robótica.
- Se logró desarrollar un modelo robótico, el cual está caracterizado por el correcto funcionamiento de la parte del brazo y sus movimientos en el sentido de los puntos cardinales y su sistema de recolección, soportando objetos de pesos ligeros y ofreciendo una movilidad omnidireccional.
- La aplicación se logró enlazar con éxito al módulo bluetooth del robot. El cual nos permite manejar cada acción del robot, sea desde las ruedas omnidireccionales y su brazo con agarre incorporado.
- La propuesta del manual del prototipo, sirven como material complementario tanto para el docente y el estudiante que necesite saber su utilización, conexiones y funcionamiento o se requiera intervenir en algún tipo de mantenimiento, por el lado del usuario las prácticas son cortas y sencillas de entender, pero tienen una amplia capacidad de aumentar su complejidad según la necesidad del docente.

X-B. Recomendaciones

- El dimensionamiento de la batería afecta directamente la duración, pero, también conlleva un mayor peso que tomar en cuenta, la relación entre duración deseada y capacidad de carga es primordial a tomar en cuenta.
- Posibles mantenimientos y aperturas del robot para visualizar el funcionamiento interno no deben realizarse sin supervisión de un tutor o profesor asignado para evitar daños en el equipo.
- Levantar cargas mayores a los que el brazo puede soportar podría dañar o reducir el tiempo de vida útil del mismo, su aplicación debe ser meramente didáctica.
- Las piezas impresas afectan directamente al rendimiento del robot en general, debido a que un mayor relleno por pieza impresa mejora su calidad y dureza, pero viene a costa de un mayor peso, exigiendo más al sistema

REFERENCIAS

- [1] J. BILBAO y P. ESCOBAR, *Investigación y Educación Superior*. Estados Unidos: LuLu, 2020.
- [2] C. GARAY, *Técnicas e Instrumentos de Investigación*. Panamá: Universidad de Panamá, jul. de 2020.
- [3] J. GONZALEZ, «La robotica educativa como recurso tecnológico innovador para potenciar el razonamiento lógico, la creatividad y el aprendizaje significativo en la asignatura de matemáticas para los niños del segundo año de educación básica de la escuela Lauro Damerval,» Tesis de Grado, Universidad Nacional de Loja, Loja,Loja,Ecuador, Diciembre de 2015.
- [4] J. LÓPEZ, D. LOSADA, R. SANZ y E. PAZ, «Sistema de Vigilancia de edificios basado en robots móviles,» en *Actas del Workshop Robot'09*, ResearchGate, ed., Vigo, 2009, págs. 1-9.
- [5] L. LÓPEZ, *Robótica educativa: recuperando la alegría por el aprendizaje y la investigación por la ciencia y la tecnología*. Quito,Pichincha,Ecuador: Universidad San Francisco de Quito, sep. de 2013.
- [6] P. LÓPEZ y H. SOSA, «Aprendizaje con Robótica, algunas experiencias,» *Educación*, págs. 43-63, 2013.
- [7] D. G. Martín BAUDINO, «Robot móvil de diseño abierto y bajo costo con fines didácticos basado en la EduCIAA,» en *Robot móvil de diseño abierto y bajo costo confines didácticos basado en la EduCIAA*, C. de Investigacion en Informatica para la Ingenieria, ed., Argentina: Facultad Regional Cordoba, Universidad Tecnol´ogica Nacional, nov. de 2019, págs. 1-6.
- [8] A. MUÑOZ e Y. GONZÁLEZ, «Robótica para desarrollar el pensamiento computacional en Educación Infantil,» *Comunicar*, págs. 63-72, 2019.
- [9] M. RIPANI, *Programación y robótica: Objetivos de aprendizaje para la educación básica*. Buenos Aires,Argentina: Ministerio de Eduación de Argentina, 2017.
- [10] A. RUBÍO, «Los bloques legos como recurso didáctico para la enseñanza de las cuentas anuales,» Tesis de Diplomado, Universidad Internacional de Rioja, Bilbao,Vizcaya,España, mayo de 2016.
- [11] R. SAMPIERI, C. COLLADO y M. BAPTISTA, *Metodología de la Investigación*. Ciudad de México,México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V., 2014.
- [12] D. Tavares, «La robótica como herramienta para el desarrollo de capacidad para aprender a ser,convivir y aprender a aprender,» Tesis de Diplomado, Universidad Andina Simón Bolívar, Quito,Pichincha,Ecuador, 2019.

XI. ANEXOS

XI-A. Código Completo de la Programación en Arduino

A continuación se introduce el código completo usado, con comentarios en la funciones más importantes usadas para explicar su funcionamiento

```
1 // Programa de Control Carro con Brazo Robotico
2 // Autores: Luis Cardenas - Josselyn Burbano
3
4 // Se incluyen las librerias necesarias
5 #include <AccelStepper.h>
6 #include <Servo.h>
7
8 // Se definen servos a utilizar
9 Servo servo01;
10 Servo servo02;
11 Servo servo03;
12 Servo servo04;
13 Servo servo05;
14 Servo servo06;
15
16 // Se definen los motores paso a paso y los pines a utilizar
17 AccelStepper LeftBackWheel(1, 46, 48); // (Driver, STEP, DIR) - Motor1
18 AccelStepper LeftFrontWheel(1, 52, 50); // Motor2
19 AccelStepper RightBackWheel(1, 32, 34); // Motor3
20 AccelStepper RightFrontWheel(1, 38, 36); // Motor4
21
22 // Se definen leds
23 #define led1 23
24 #define led2 25
25 #define led3 27
26
27 int wheelSpeed = 1500;
28
29 int lbw[50], lfw[50], rbw[50], rfw[50]; //Matriz para guardar posiciones/pasos
30
31 int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // Posicion actual
32 int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // Posicion
  anterior
33 int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50
  ]; //Para guardar posiciones/pasos
34 int speedDelay = 20;
35 int index = 0;
36 int dataIn;
37 int m = 0;
38
39 void setup() {
40 //Se establecen valores iniciales para los motores
41 LeftFrontWheel.setMaxSpeed(3000);
42 LeftBackWheel.setMaxSpeed(3000);
43 RightFrontWheel.setMaxSpeed(3000);
44 RightBackWheel.setMaxSpeed(3000);
45 pinMode(led1, OUTPUT);
46 pinMode(led2, OUTPUT);
47 pinMode(led3, OUTPUT);
48 servo01.attach(5);
49 servo02.attach(6);
50 servo03.attach(7);
51 servo04.attach(8);
52 servo05.attach(9);
53 servo06.attach(10);
54 Serial3.begin(9600); // Valor predeterminado de lectura para el modulo bluetooth
```

```

55 Serial3.setTimeout(5);
56 delay(20);
57 Serial.begin(9600);
58
59 //Mueve el robot a la posicion inicial
60 servo1PPos = 90;
61 servo01.write(servo1PPos);
62 servo2PPos = 90;
63 servo02.write(servo2PPos);
64 servo3PPos = 90;
65 servo03.write(servo3PPos);
66 servo4PPos = 90;
67 servo04.write(servo4PPos);
68 servo5PPos = 90;
69 servo05.write(servo5PPos);
70 servo6PPos = 90;
71 servo06.write(servo6PPos);
72 }
73
74 void loop() {
75 // Chequea informacion de entrada
76 if (Serial3.available() > 0) {
77     dataIn = Serial3.read(); // Lee la informacion
78
79     if (dataIn == 0) {
80         m = 0;
81     }
82     if (dataIn == 1) {
83         m = 1;
84     }
85     if (dataIn == 2) {
86         m = 2;
87     }
88     if (dataIn == 3) {
89         m = 3;
90     }
91     if (dataIn == 4) {
92         m = 4;
93     }
94     if (dataIn == 5) {
95         m = 5;
96     }
97     if (dataIn == 6) {
98         m = 6;
99     }
100    if (dataIn == 7) {
101        m = 7;
102    }
103    if (dataIn == 8) {
104        m = 8;
105    }
106    if (dataIn == 9) {
107        m = 9;
108    }
109    if (dataIn == 10) {
110        m = 10;
111    }
112    if (dataIn == 11) {
113        m = 11;
114    }
115    if (dataIn == 12) {
116        m = 12;
117    }

```

```

118     if (dataIn == 14) {
119         m = 14;
120     }
121     if (dataIn == 16) {
122         m = 16;
123     }
124     if (dataIn == 17) {
125         m = 17;
126     }
127     if (dataIn == 18) {
128         m = 18;
129     }
130     if (dataIn == 19) {
131         m = 19;
132     }
133     if (dataIn == 20) {
134         m = 20;
135     }
136     if (dataIn == 21) {
137         m = 21;
138     }
139     if (dataIn == 22) {
140         m = 22;
141     }
142     if (dataIn == 23) {
143         m = 23;
144     }
145     if (dataIn == 24) {
146         m = 24;
147     }
148     if (dataIn == 25) {
149         m = 25;
150     }
151     if (dataIn == 26) {
152         m = 26;
153     }
154     if (dataIn == 27) {
155         m = 27;
156     }
157
158     // Mueve la base del robot
159     if (m == 4) {
160         moveSidewaysLeft();
161     }
162     if (m == 5) {
163         moveSidewaysRight();
164     }
165     if (m == 2) {
166         moveForward();
167     }
168     if (m == 7) {
169         moveBackward();
170     }
171     if (m == 3) {
172         moveRightForward();
173     }
174     if (m == 1) {
175         moveLeftForward();
176     }
177     if (m == 8) {
178         moveRightBackward();
179     }
180     if (m == 6) {

```

```

181     moveLeftBackward();
182 }
183 if (m == 9) {
184     rotateLeft();
185 }
186 if (m == 10) {
187     rotateRight();
188 }
189
190 if (m == 0) {
191     stopMoving();
192 }
193
194 // Velocidad de las ruedas del robot
195 if (dataIn > 30 & dataIn < 100) {
196     wheelSpeed = dataIn * 20;
197 }
198
199 // Movimiento del servo 1 en direccion positiva
200 while (m == 16) {
201     if (Serial3.available() > 0) {
202         m = Serial3.read();
203     }
204     servo01.write(servo1PPos);
205     servo1PPos++;
206     if (servo1PPos >= 180){
207         servo1PPos = 180;
208     }
209     Serial.println(servo1PPos);
210     delay(speedDelay);
211 }
212 // Movimiento del servo 1 en direccion negativa
213 while (m == 17) {
214     if (Serial3.available() > 0) {
215         m = Serial3.read();
216     }
217     servo01.write(servo1PPos);
218     servo1PPos--;
219     if (servo1PPos <= 0){
220         servo1PPos = 0;
221     }
222     Serial.println(servo1PPos);
223     delay(speedDelay);
224 }
225 // Movimiento del servo 2
226 while (m == 19) {
227     if (Serial3.available() > 0) {
228         m = Serial3.read();
229     }
230     servo02.write(servo2PPos);
231     servo2PPos++;
232     if (servo2PPos >= 180){
233         servo2PPos = 180;
234     }
235     Serial.println(servo2PPos);
236     delay(speedDelay);
237 }
238 }
239 while (m == 18) {
240     if (Serial3.available() > 0) {
241         m = Serial3.read();
242     }
243     servo02.write(servo2PPos);

```

```

244     servo2PPos--;
245     if (servo2PPos <= 0){
246         servo2PPos = 0;
247     }
248     Serial.println(servo2PPos);
249     delay(speedDelay);
250 }
251 }
252 // Movimiento del servo 3
253 while (m == 20) {
254     if (Serial3.available() > 0) {
255         m = Serial3.read();
256     }
257     servo03.write(servo3PPos);
258     servo3PPos++;
259     if (servo3PPos >= 180){
260         servo3PPos = 180;
261     }
262     Serial.println(servo3PPos);
263     delay(speedDelay);
264 }
265 while (m == 21) {
266     if (Serial3.available() > 0) {
267         m = Serial3.read();
268     }
269     servo03.write(servo3PPos);
270     servo3PPos--;
271     if (servo3PPos <= 0){
272         servo3PPos = 0;
273     }
274     Serial.println(servo3PPos);
275     delay(speedDelay);
276 }
277 // Movimiento del servo 4
278 while (m == 23) {
279     if (Serial3.available() > 0) {
280         m = Serial3.read();
281     }
282     servo04.write(servo4PPos);
283     servo4PPos++;
284     if (servo4PPos >= 180){
285         servo4PPos = 180;
286     }
287     Serial.println(servo4PPos);
288     delay(speedDelay);
289 }
290 while (m == 22) {
291     if (Serial3.available() > 0) {
292         m = Serial3.read();
293     }
294     servo04.write(servo4PPos);
295     servo4PPos--;
296     if (servo4PPos <= 0){
297         servo4PPos = 0;
298     }
299     Serial.println(servo4PPos);
300     delay(speedDelay);
301 }
302 // Movimiento del servo 5
303 while (m == 25) {
304     if (Serial3.available() > 0) {
305         m = Serial3.read();
306     }

```

```

307     servo05.write(servo5PPos);
308     servo5PPos++;
309     if (servo5PPos >= 180){
310         servo5PPos = 180;
311     }
312     Serial.println(servo5PPos);
313     delay(speedDelay);
314 }
315 while (m == 24) {
316     if (Serial3.available() > 0) {
317         m = Serial3.read();
318     }
319     servo05.write(servo5PPos);
320     servo5PPos--;
321     if (servo5PPos <= 0){
322         servo5PPos = 0;
323     }
324     Serial.println(servo5PPos);
325     delay(speedDelay);
326 }
327 }
328 // Movimiento del servo 6
329 while (m == 26) {
330     if (Serial3.available() > 0) {
331         m = Serial3.read();
332     }
333     servo06.write(servo6PPos);
334     servo6PPos++;
335     if (servo6PPos >= 180){
336         servo6PPos = 180;
337     }
338     Serial.println(servo6PPos);
339     delay(speedDelay);
340 }
341 while (m == 27) {
342     if (Serial3.available() > 0) {
343         m = Serial3.read();
344     }
345     servo06.write(servo6PPos);
346     servo6PPos--;
347     if (servo6PPos <= 0){
348         servo6PPos = 0;
349     }
350     Serial.println(servo6PPos);
351     delay(speedDelay);
352 }
353 }
354 // Si el slider de la velocidad del brazo llega a cambiar
355 if (dataIn > 101 & dataIn < 250) {
356     speedDelay = dataIn / 10; // Modifica la velocidad del servo (delay)
357 }
358 }
359 // Si se presiona el boton "Guardar"
360 if (m == 12) {
361     // Si es el primer guardado, coloca los motores en posicion 0
362     if (index == 0) {
363         LeftBackWheel.setCurrentPosition(0);
364         LeftFrontWheel.setCurrentPosition(0);
365         RightBackWheel.setCurrentPosition(0);
366         RightFrontWheel.setCurrentPosition(0);
367     }
368     lbw[index] = LeftBackWheel.currentPosition(); // Guarda la posicion en la matriz
369     lfw[index] = LeftFrontWheel.currentPosition();

```



```

370     rbw[index] = RightBackWheel.currentPosition();
371     rfw[index] = RightFrontWheel.currentPosition();
372
373     servo01SP[index] = servo1PPos; // Guarda la posicion en la matriz
374     servo02SP[index] = servo2PPos;
375     servo03SP[index] = servo3PPos;
376     servo04SP[index] = servo4PPos;
377     servo05SP[index] = servo5PPos;
378     servo06SP[index] = servo6PPos;
379     index++; // Incrementa el index de la matriz
380     m = 0;
381 }
382
383 // Si se presiona el boton "Accionar"
384 if (m == 14) {
385     runSteps();
386
387     // Si se presiona el boton "Resetear"
388     if (dataIn != 14) {
389         stopMoving();
390         memset(lbw, 0, sizeof(lbw)); // Resetea los valores de la matriz a 0
391         memset(lfw, 0, sizeof(lfw));
392         memset(rbw, 0, sizeof(rbw));
393         memset(rfw, 0, sizeof(rfw));
394         memset(servo01SP, 0, sizeof(servo01SP)); // Resetea los valores de la matriz a 0
395         memset(servo02SP, 0, sizeof(servo02SP));
396         memset(servo03SP, 0, sizeof(servo03SP));
397         memset(servo04SP, 0, sizeof(servo04SP));
398         memset(servo05SP, 0, sizeof(servo05SP));
399         memset(servo06SP, 0, sizeof(servo06SP));
400         index = 0; // Index a 0
401     }
402 }
403 }
404 LeftFrontWheel.runSpeed();
405 LeftBackWheel.runSpeed();
406 RightFrontWheel.runSpeed();
407 RightBackWheel.runSpeed();
408
409 // Monitoreo del voltaje de la bateria
410 int sensorValue = analogRead(A0);
411 float voltage = sensorValue * (5.0 / 1023.00) * 3; // Convierte los valores de lectura de
412 // 5v a 12v
413 // Serial.println(voltage);
414 // Si el voltaje es menor a 11v se activa el led rojo
415 if (voltage > 11) {
416     digitalWrite(led1, HIGH);
417     digitalWrite(led2, LOW);
418     digitalWrite(led3, LOW);
419 }
420 if (voltage > 10 and voltage < 11) {
421     digitalWrite(led1, LOW);
422     digitalWrite(led2, HIGH);
423     digitalWrite(led3, LOW);
424 }
425 if (voltage < 10) {
426     digitalWrite(led1, LOW);
427     digitalWrite(led2, LOW);
428     digitalWrite(led3, HIGH);
429 }
430 void moveForward() {
431     LeftFrontWheel.setSpeed(wheelSpeed);

```

```

432 LeftBackWheel . setSpeed ( wheelSpeed );
433 RightFrontWheel . setSpeed ( wheelSpeed );
434 RightBackWheel . setSpeed ( wheelSpeed );
435 }
436 void moveBackward () {
437     LeftFrontWheel . setSpeed ( -wheelSpeed );
438     LeftBackWheel . setSpeed ( -wheelSpeed );
439     RightFrontWheel . setSpeed ( -wheelSpeed );
440     RightBackWheel . setSpeed ( -wheelSpeed );
441 }
442 void moveSidewaysRight () {
443     LeftFrontWheel . setSpeed ( wheelSpeed );
444     LeftBackWheel . setSpeed ( -wheelSpeed );
445     RightFrontWheel . setSpeed ( -wheelSpeed );
446     RightBackWheel . setSpeed ( wheelSpeed );
447 }
448 void moveSidewaysLeft () {
449     LeftFrontWheel . setSpeed ( -wheelSpeed );
450     LeftBackWheel . setSpeed ( wheelSpeed );
451     RightFrontWheel . setSpeed ( wheelSpeed );
452     RightBackWheel . setSpeed ( -wheelSpeed );
453 }
454 void rotateLeft () {
455     LeftFrontWheel . setSpeed ( -wheelSpeed );
456     LeftBackWheel . setSpeed ( -wheelSpeed );
457     RightFrontWheel . setSpeed ( wheelSpeed );
458     RightBackWheel . setSpeed ( wheelSpeed );
459 }
460 void rotateRight () {
461     LeftFrontWheel . setSpeed ( wheelSpeed );
462     LeftBackWheel . setSpeed ( wheelSpeed );
463     RightFrontWheel . setSpeed ( -wheelSpeed );
464     RightBackWheel . setSpeed ( -wheelSpeed );
465 }
466 void moveRightForward () {
467     LeftFrontWheel . setSpeed ( wheelSpeed );
468     LeftBackWheel . setSpeed ( 0 );
469     RightFrontWheel . setSpeed ( 0 );
470     RightBackWheel . setSpeed ( wheelSpeed );
471 }
472 void moveRightBackward () {
473     LeftFrontWheel . setSpeed ( 0 );
474     LeftBackWheel . setSpeed ( -wheelSpeed );
475     RightFrontWheel . setSpeed ( -wheelSpeed );
476     RightBackWheel . setSpeed ( 0 );
477 }
478 void moveLeftForward () {
479     LeftFrontWheel . setSpeed ( 0 );
480     LeftBackWheel . setSpeed ( wheelSpeed );
481     RightFrontWheel . setSpeed ( wheelSpeed );
482     RightBackWheel . setSpeed ( 0 );
483 }
484 void moveLeftBackward () {
485     LeftFrontWheel . setSpeed ( -wheelSpeed );
486     LeftBackWheel . setSpeed ( 0 );
487     RightFrontWheel . setSpeed ( 0 );
488     RightBackWheel . setSpeed ( -wheelSpeed );
489 }
490 void stopMoving () {
491     LeftFrontWheel . setSpeed ( 0 );
492     LeftBackWheel . setSpeed ( 0 );
493     RightFrontWheel . setSpeed ( 0 );
494     RightBackWheel . setSpeed ( 0 );

```

```

495 }
496
497 // Modo Automatico para correr los pasos guardados
498 void runSteps() {
499   while (dataIn != 13) { // Acciona los pasos una y otra vez hasta que el bot n "
Resetear" sea presionado"
500     for (int i = 0; i <= index - 2; i++) { // Corre cada uno de los pasos(index)
501       if (Serial3.available() > 0) { // Chequea informacion de entrada
502         dataIn = Serial3.read();
503         if ( dataIn == 15) { // Si el boton "Pausado" es presionado
504           while (dataIn != 14) { // Espera hasta que el boton "Corriendo" vuelva a
ser presionado
505             if (Serial3.available() > 0) {
506               dataIn = Serial3.read();
507               if ( dataIn == 13) {
508                 break;
509               }
510             }
511           }
512         }
513         // Si el slider es cambiado
514         if (dataIn > 100 & dataIn < 150) {
515           speedDelay = dataIn / 10; // Cambia la velocidad del servo (delay)
516         }
517         // Velocidad de las ruedas
518         if (dataIn > 30 & dataIn < 100) {
519           wheelSpeed = dataIn * 10;
520           dataIn = 14;
521         }
522       }
523       LeftFrontWheel.moveTo(lfw[i]);
524       LeftFrontWheel.setSpeed(wheelSpeed);
525       LeftBackWheel.moveTo(lbw[i]);
526       LeftBackWheel.setSpeed(wheelSpeed);
527       RightFrontWheel.moveTo(rfw[i]);
528       RightFrontWheel.setSpeed(wheelSpeed);
529       RightBackWheel.moveTo(rbw[i]);
530       RightBackWheel.setSpeed(wheelSpeed);
531
532       while (LeftBackWheel.currentPosition() != lbw[i] & LeftFrontWheel.currentPosition()
!= lfw[i] & RightFrontWheel.currentPosition() != rfw[i] & RightBackWheel.
currentPosition() != rbw[i]) {
533         LeftFrontWheel.runSpeedToPosition();
534         LeftBackWheel.runSpeedToPosition();
535         RightFrontWheel.runSpeedToPosition();
536         RightBackWheel.runSpeedToPosition();
537       }
538       // Servo 1
539       if (servo01SP[i] == servo01SP[i + 1]) {
540       }
541       if (servo01SP[i] > servo01SP[i + 1]) {
542         for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
543           servo01.write(j);
544           delay(speedDelay);
545         }
546       }
547       if (servo01SP[i] < servo01SP[i + 1]) {
548         for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
549           servo01.write(j);
550           delay(speedDelay);
551         }
552       }
553     }

```

```

554 // Servo 2
555 if (servo02SP[i] == servo02SP[i + 1]) {
556 }
557 if (servo02SP[i] > servo02SP[i + 1]) {
558     for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
559         servo02.write(j);
560         delay(speedDelay);
561     }
562 }
563 if (servo02SP[i] < servo02SP[i + 1]) {
564     for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
565         servo02.write(j);
566         delay(speedDelay);
567     }
568 }
569
570 // Servo 3
571 if (servo03SP[i] == servo03SP[i + 1]) {
572 }
573 if (servo03SP[i] > servo03SP[i + 1]) {
574     for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
575         servo03.write(j);
576         delay(speedDelay);
577     }
578 }
579 if (servo03SP[i] < servo03SP[i + 1]) {
580     for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
581         servo03.write(j);
582         delay(speedDelay);
583     }
584 }
585
586 // Servo 4
587 if (servo04SP[i] == servo04SP[i + 1]) {
588 }
589 if (servo04SP[i] > servo04SP[i + 1]) {
590     for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
591         servo04.write(j);
592         delay(speedDelay);
593     }
594 }
595 if (servo04SP[i] < servo04SP[i + 1]) {
596     for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
597         servo04.write(j);
598         delay(speedDelay);
599     }
600 }
601
602 // Servo 5
603 if (servo05SP[i] == servo05SP[i + 1]) {
604 }
605 if (servo05SP[i] > servo05SP[i + 1]) {
606     for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
607         servo05.write(j);
608         delay(speedDelay);
609     }
610 }
611 if (servo05SP[i] < servo05SP[i + 1]) {
612     for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
613         servo05.write(j);
614         delay(speedDelay);
615     }
616 }

```

```

617
618 // Servo 6
619 if (servo06SP[i] == servo06SP[i + 1]) {
620 }
621 if (servo06SP[i] > servo06SP[i + 1]) {
622     for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
623         servo06.write(j);
624         delay(speedDelay);
625     }
626 }
627 if (servo06SP[i] < servo06SP[i + 1]) {
628     for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
629         servo06.write(j);
630         delay(speedDelay);
631     }
632 }
633 }
634 }
635 }

```

Listing 1. Programación en Arduino

Figura 44. Programación por Bloques de Interfaz (2/2)

The image displays 23 code blocks for an interface program, organized into several groups:

- Block 1:** A 'when clicked' block for 'RotarLeft' that sets 'RotacionLeft' to 0 and 'RotacionLeft' to 1, then sends a 'RotacionLeft' message to 'BluetoothClient1'.
- Block 2:** A 'when clicked' block for 'RotarRight' that sets 'RotacionRight' to 0 and 'RotacionRight' to 1, then sends a 'RotacionRight' message to 'BluetoothClient1'.
- Block 3:** A 'when speed platform position changed' block that sends a 'Position' message to 'BluetoothClient1'.
- Block 4:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 5:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 6:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 7:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 8:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 9:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 10:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 11:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 12:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 13:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 14:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 15:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 16:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 17:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 18:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 19:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 20:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 21:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.
- Block 22:** A 'when clicked' block for 'RotarUp' that sets 'RotacionUp' to 0 and 'RotacionUp' to 1, then sends a 'RotacionUp' message to 'BluetoothClient1'.
- Block 23:** A 'when clicked' block for 'RotarDown' that sets 'RotacionDown' to 0 and 'RotacionDown' to 1, then sends a 'RotacionDown' message to 'BluetoothClient1'.

XI-C. Propuesta de Manual de Robot

MANUAL DE USUARIO DE UN ROBOT MÓVIL CON SISTEMA DE RECOLECCIÓN DE OBJETOS CONTROLADO REMOTAMENTE PARA FINES DIDÁCTICOS CONTENIDO

1. Advertencia
2. Ensamblaje del Robot
3. Conexiones del Robot

Advertencia

- Este robot está programado para realizar funciones de manera manual y automática. Es conveniente que el usuario primero lea el manual antes de montar al robot.
- No utilizar el robot en pisos con rugosidad, preferible usarlo en pisos de caucho o tapetes.
- Se recomienda probar primero el robot antes de usarlo y revisar que esté en perfectas condiciones.

Ensamblaje del Robot

Antes de realizar el ensamblaje del robot, se especifica una lista de secciones caracterizando como va estructurado el robot.

- Sección 1. Partes del Robot Móvil
- Sección 2. Partes del Brazo robótico
- Sección 3. Lista de Herramientas
- Sección 4. Lista de Componentos electrónicos
- Sección 5. Ensamblaje del Robot

Sección 1 - Partes del Robot Móvil

- Caja.
- Rodillo.
- Acople de Eje.
- Base de la rueda izuierda.
- Base de la rueda derecha.
- Rueda lado izquierdo.
- Rueda lado derecho.

Sección 2 - Partes del Brazo Robótico

- Sección Brazo (1/3).
- Sección Brazo (2/3).
- Sección Brazo (3/3).
- Engranaje (1/2).
- Engranaje (2/2).
- Base de Agarre.
- Cintura.
- Pinza.

Sección 3 - Lista de Herramientas

Lista	Cantidad
Tornillo M2 x 6mm	6
Tornillo M3 x 3mm	12
Arandelas	12
Perno M4 x 50mm	16
Perono con cabeza allen M3 x 20mm	7
Tuercas M3 x 3mm	35
Pernos M3 x 3mm	24
Perno M4 x 5mm	3
Tornillo M2 x 4mm	3
Varillas de Aluminio M3 x 40mm	16
Desarmadores	#
Juego de llaves allen	#

Sección 4 - Lista de Componentes Electrónicos

Lista	Cantidad
Micro Servo	3
Servo Motor	3
Motor Paso a Paso Nema 17	4
Placa Arduino Mega	1
Placa PCB personalizada	1
Módulo Bluetooth HC-05	1
Driver para motor paso a paso	4
Batería Lipo 4A	1
Interruptor	1
Led	1
Ventilador de 12v	1
Cargador para batería lipo de 3 celdas	1
Juegos de cables jumper	#

Sección 5.1 - Ensamblaje del Brazo Robótico

Luego de obtener las herramientas necesarias para el robot y los componentes electrónicos. Se explicara paso a paso cómo se va armando el robot. Primero empezaremos a armar el brazo del robot y luego el robot con sus 4 ruedas omnidireccional.

Estas son todas las piezas para el brazo robótico. A Continuación se mostrará paso a paso cómo se va armando.

Tabla de los pernos, tornillos y arandelas que se van a utilizar para el brazo robótico

Sección	Lista	Cantidad
A. Servo Motor	Tornillo M3 x 3 mm	12
B. Servo Motor	Arandelas	12
C. Micro Servo	Tornillo M2 x 6 mm	6
D. Montaje Agarre	Pernos con cabeza allen M3 x 20 mm	7
E. Montaje Agarre	Tuercas M3 x 3 mm	7

Nota: Esta tabla es una guía para usar los pernos de cada sección del brazo.

Figura 45. Paso1. Sección A y B



Figura 46. Paso2. Sección A y B



Figura 47. Paso3. Sección A y B



Figura 48. Paso4. Sección A y B



Figura 49. Paso5. Sección A y B



Figura 50. Paso6. Sección C



Figura 51. Paso7. Sección C



Figura 52. Paso8. Sección C



Figura 53. Paso9. Sección D y E

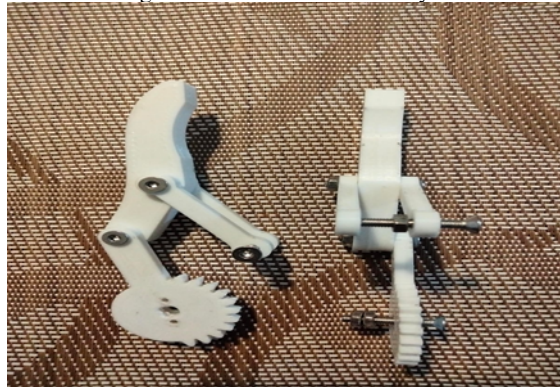


Figura 54. Paso10. Sección D y E



Figura 55. Paso11. Sección D y E



Figura 56. Paso12. Sección D y E



Figura 57. Paso13. Sección D y E



Figura 58. Paso14. Sección D y E



Sección 5.2 - Ensamblaje del Cuerpo Móvil Robótico

Sección	Lista	Cantidad
A. Rueda y cuerpo del robot	Pernos M3 X 45 mm	16
B. Rueda y cuerpo del robot	Tuercas M3	16
C. Ruedas	Varillas de Aluminio de M3 x 40 mm	16
D. Ruedas	Arandelas	40

Nota: La tabla es una guía para usar los pernos de cada sección del robot móvil

Figura 59. Paso 1. Conexión de las 4 ruedas omnidireccionales, dos ruedas laterales izquierdas y dos ruedas laterales derechas. Sección A y B

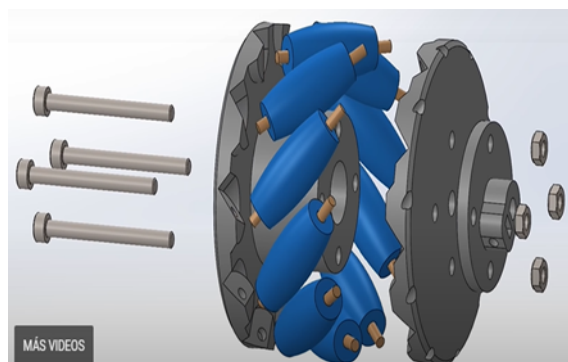


Figura 60. Paso 2. Montaje de las ruedas. Sección A y B.



Figura 61. Paso 3. Montaje de las ruedas con las varillas de aluminio. Sección C y D.



Figura 62. Paso 4. Ya puesta la rueda con los 10 rodillos sigue el acople que está hecho de bronce. como se muestra en la figura señalada. Sección A y B

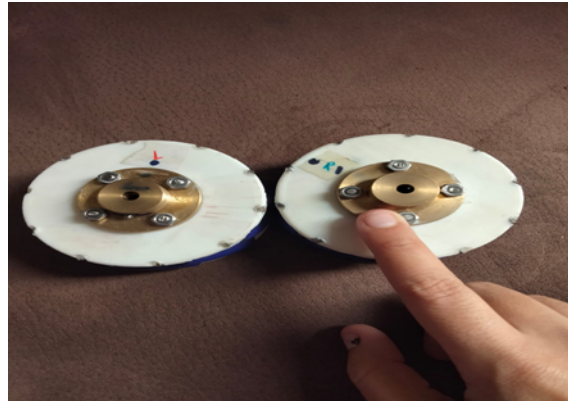


Figura 63. Paso 5. Luego, de poner las ruedas con sus acoples vemos que el acople ya tiene puesto un suprimidor. Sección A y B



Figura 64. Paso 6. Una vez terminadas las cuatro ruedas explicaremos cómo se va armando el cuerpo del robot. Sección A y B.



Figura 65. Paso 7. Como son ruedas omnidireccionales sus direcciones irían de esta manera. Sección A y B

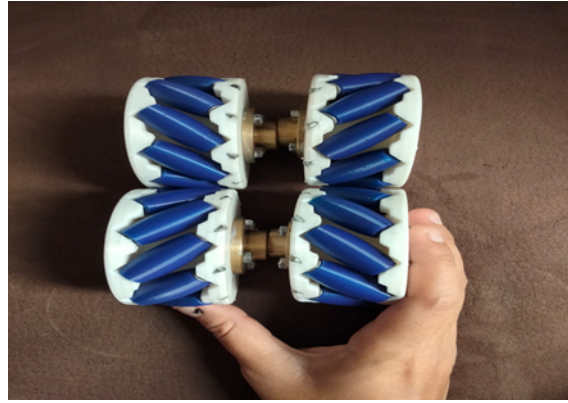


Figura 66. Paso 8. Se van colocando cada motor con su ruedas apropiadas como detalla el paso anterior. Sección A y B.



Figura 67. Paso 9. Sección A y B.

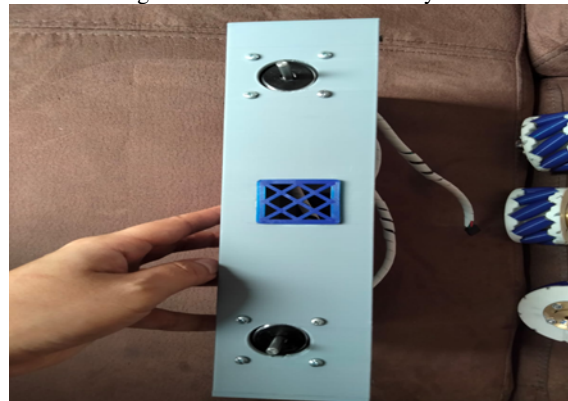


Figura 68. Paso 10. Sección A y B.



Figura 69. Paso 11. Sección A y B.



Figura 70. Paso 12. Ya puestos los 4 motores, nos toca colocar las ruedas y cada rueda tiene un acople y su prisionero. Primero con una llave allen vas desajustando, luego colocas en la rueda y así sucesivamente con las otras ruedas.



Figura 71. Paso 13. Sección A y B.

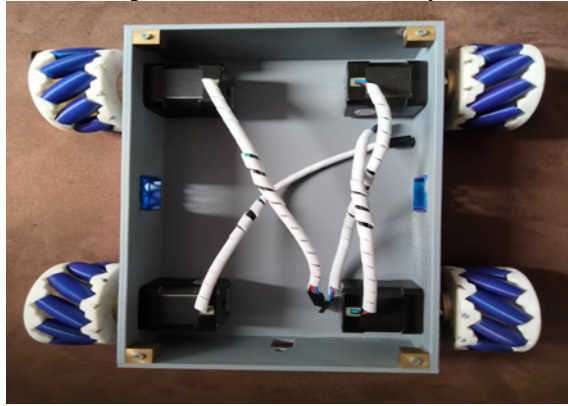


Figura 72. Paso 14. Ya terminado de colocar todas las ruedas con la llave allen, ahora toca armar la tapa de la caja. Sección A y B.

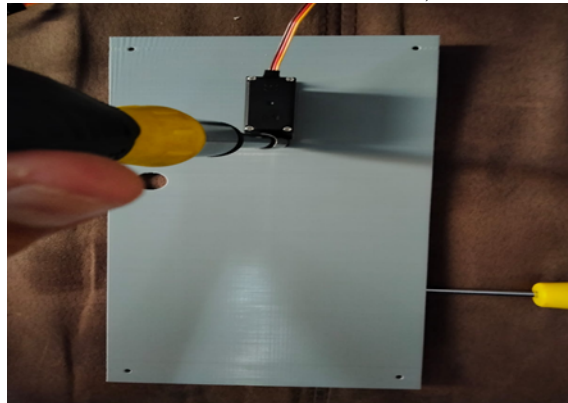
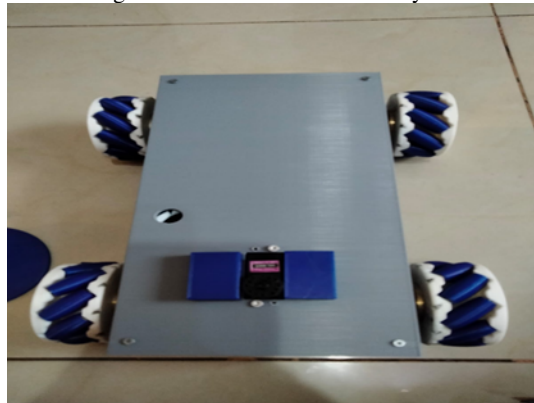
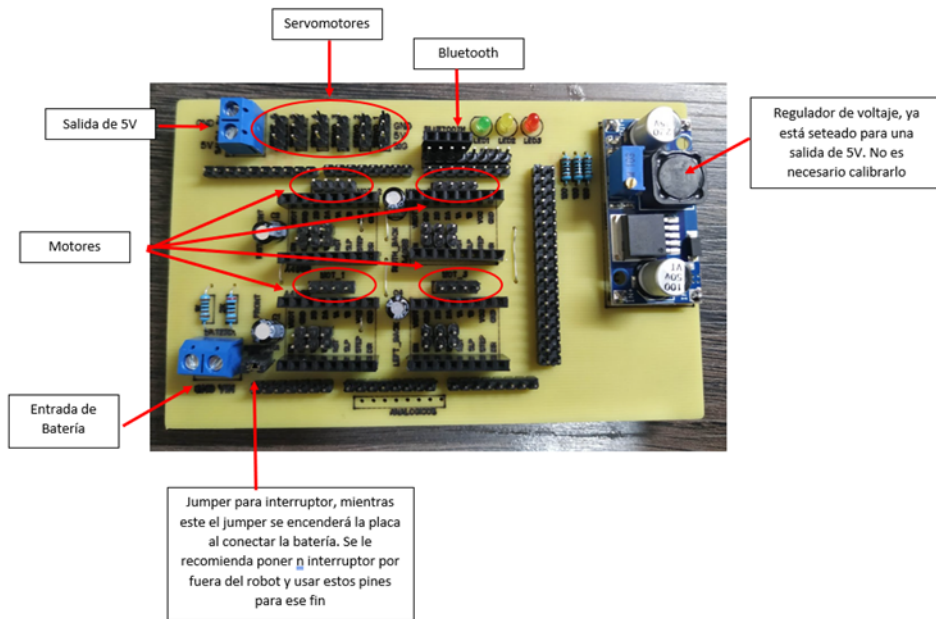


Figura 73. Paso 15. Sección A y B.



Conexiones del Robot

Figura 74. Conexiones a realizar en la placa PCB.



Guía de Instrucciones

Tabla de Instrucciones Para Brazo Robótico

Partes del Brazo	Activación	Función
Cadera	Boton Derecha	Girar la cadera hacia la Derecha
Cadera	Boton Izquierda	Girar la cadera hacia la Izquierda
Hombro	Boton Derecha	Inclinar el homnro hacia Arriba
Hombro	Boton Izquierda	Inclinar el homnro hacia Abajo
Codo	Boton Derecha	Mueve el codo hacia Arriba
Codo	Boton Izquierda	Mueve el codo hacia Abajo
Giro Muñeca	Boton Derecha	Giro Horario
Giro Muñeca	Boton Izquierda	Giro Anti - Horario
Dirección Muñeca	Boton Derecha	Movimiento Muñeca hacia arriba
Dirección Muñeca	Boton Izquierda	Movimiento Muñeca hacia abajo
Garra	Boton Derecha	Cierra la Pinza
Garra	Boton Izquierda	Abre la Pinza

Tabla de Instrucciones Para Plataforma Móvil

Dirrección de las Ruedas	Activación	Función
Adelante	Boton Adelante	Movimiento Adelante
Atrás	Boton Atrás	Movimiento Atrás
Derecha	Boton Derecha	Movimiento Lado Derecha
Izquierda	Boton Izquierda	Movimiento Lado Izquierda
Diagonal Adelante Derecho	Boton Diagonal Superior Derecho	Movimiento diagonal adelante derecho
Diagonal Adelante Izquierdo	Boton Diagonal Superior Izquierdo	Movimiento diagonal adelante izquierdo
Diagonal Atras Izquierdo	Boton Diagonal Inferior Izquierdo	Movimiento diagonal atras izquierdo
Diagonal Atras Derecho	Boton Diagonal Inferior Derecho	Movimiento diagonal atras derecho
Giro Derecho en su propio eje	Boton de Giro Derecho	Giro Completo Derecho
Giro Izquierdo en su propio eje	Boton de Giro Izquierdo	Giro Completo Izquierdo

Tabla de Instrucciones para Uso Automático

Activación	Función
Botón Guardar	Guarda los pasos de cada posición del robot
Botón Accionar	Acciona los pasos guardados del robot
Botón Resetear	Borra los pasos anteriores y empieza de cero

El Robot Móvil con Brazo Robótico tiene dos opciones para controlarlo que son: modo manual y modo automático.

Actividad 1. Movimiento Lineal Directo y Manipulación básica de objetos

Como actividad introductoria, el objetivo de esta actividad es familiarizar al usuario con el robot, con movimientos lineales de cuerpo móvil y uso simple para agarrar con suelte de una pelotita ligera.

1. Paso.
 - El robot rueda hacia adelante.
2. Paso.
 - El robot para unos segundos y sobre una base 1 hay una pelotita.
3. Paso.
 - Agarra la pelotita, luego pasa unos segundos.
4. Paso.
 - Abre el agarre y deja la pelotita.

Actividad 2. Movimiento Manual con curvas y Manipulación con posicionamiento de objetos

Como actividad siguiente, el objetivo de esta actividad es ayudar al usuario a usar más funciones con el robot, con el robot fuera del alcance lineal directo con el objeto inicial el usuario tendrá que maniobrar con los controles para llegar esta vez a una letra ligera, luego no solo agarrará dicha letra, sino que la transportará y posicionará a su gusto en otra zona diferente a la original.

1. Paso.
 - Llevar el robot hasta la letra.
2. Paso.
 - El robot para unos segundos y sobre una base hay una letra
3. Paso.
 - Agarra la letra, en caso de caer dicha letra, manipular el brazo para una mejor posición de agarre.
4. Paso.
 - Mover la letra a una nueva zona y depositarla ahí.

Actividad 3. Movimiento Automático

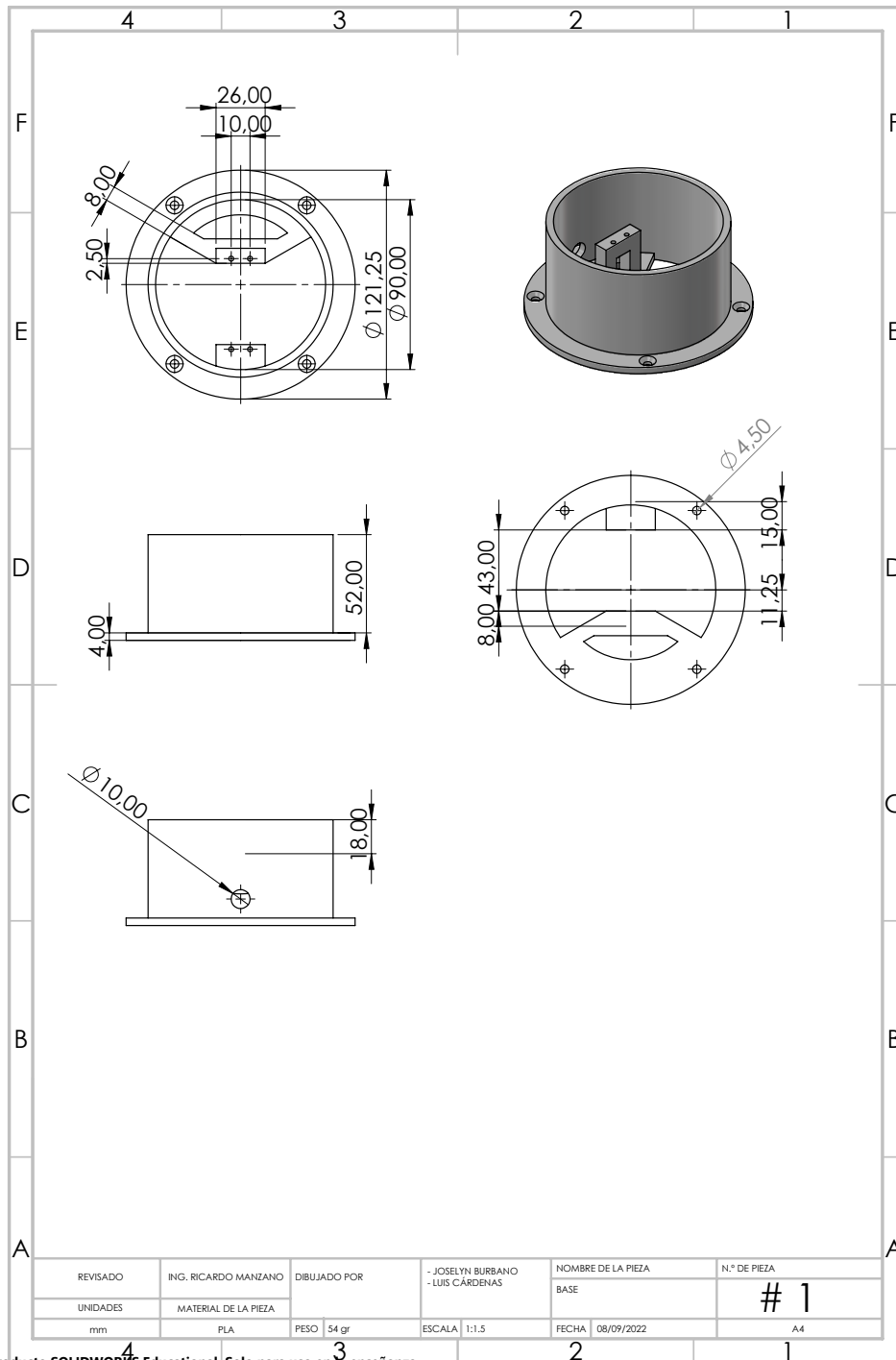
Como actividad sugerida Final, el objetivo de esta actividad es ayudar al usuario a comprender la función automática del robot, haciendo uso de guardar posiciones tanto de cuerpo móvil como de brazo, pueden lograr una gran cantidad de diferentes combinaciones para propósitos varios

1. Paso.
 - El tutor debe establecer un objetivo para el robot
2. Paso.
 - El usuario trata de armar un ciclo que cumpla con dicho propósito, para ello hace uso de los botones "Guardar", o "Resetear" para volver a armar el ciclo
3. Paso.
 - Con el ciclo ya armado, haciendo uso del botón "Accionar", el robot de manera automática realiza el ciclo
4. Paso.
 - Se verifica si el ciclo cumple o sino se vuelve a intentar hasta lograrlo.

XI-E. Planos de las piezas

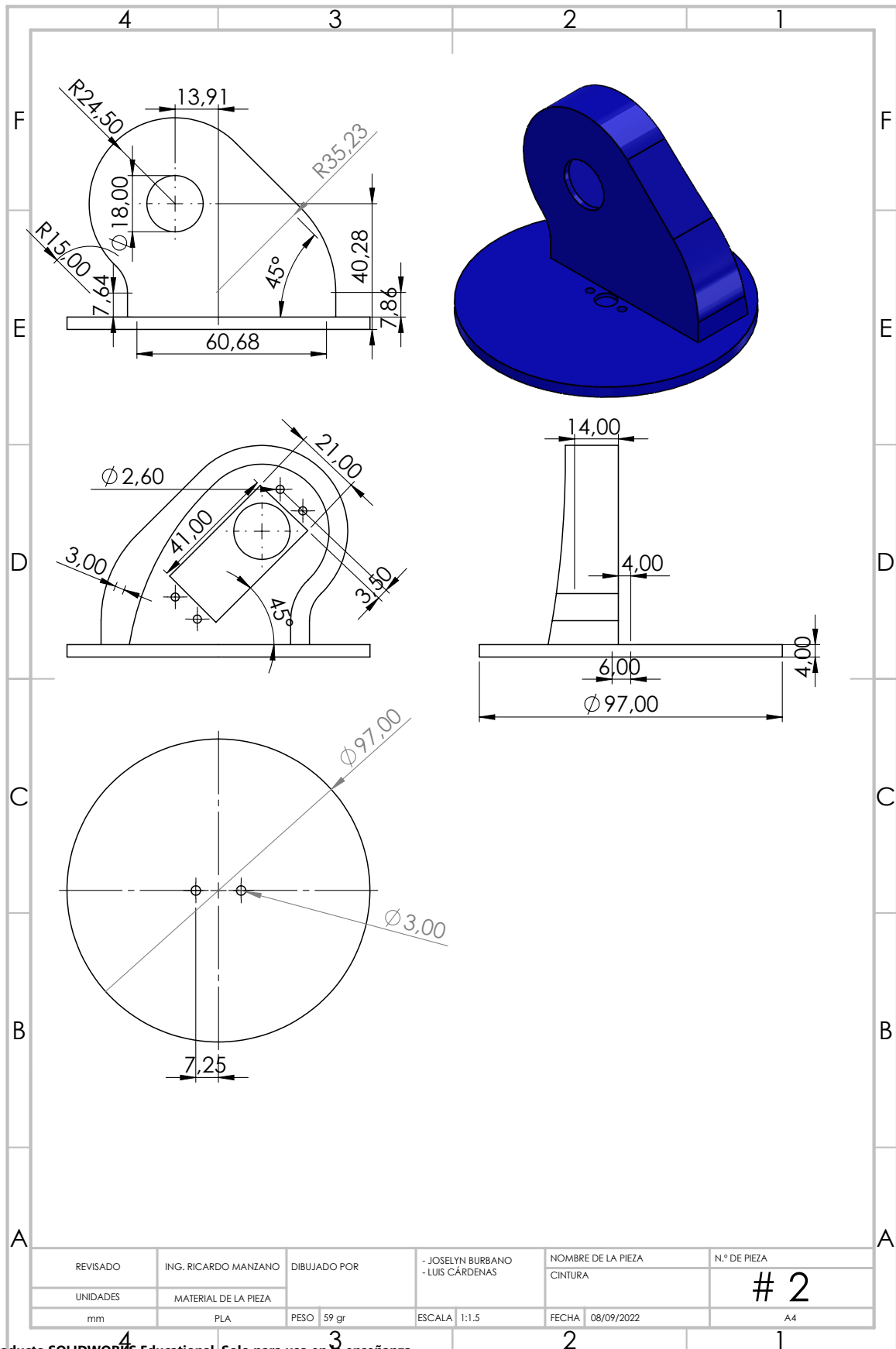
Láminas con detalles y medidas a tomar en cuenta para la creación de las piezas

Figura 75. Lámina de la Base del Robot



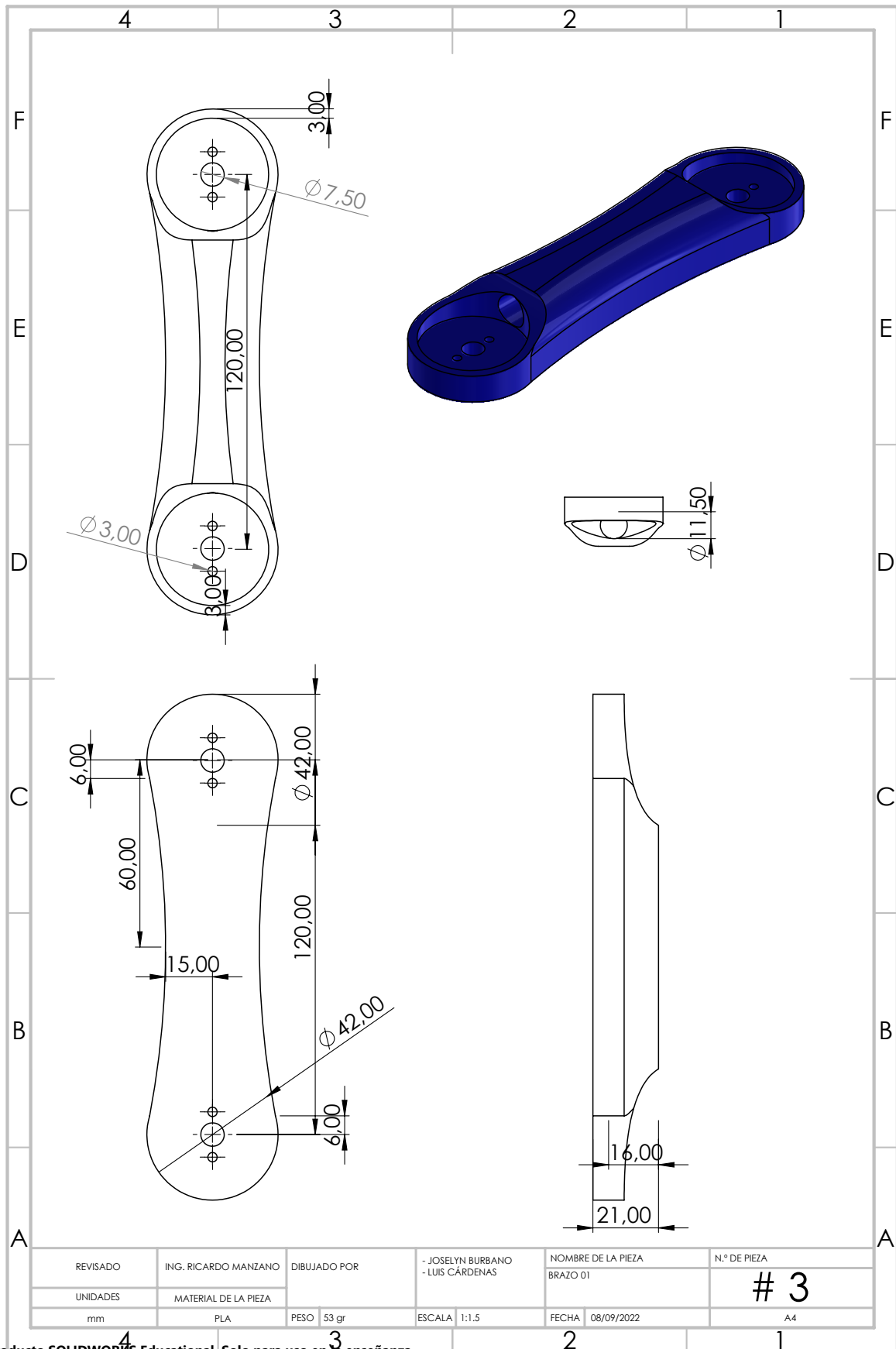
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 76. Lámina de la Cintura del Robot



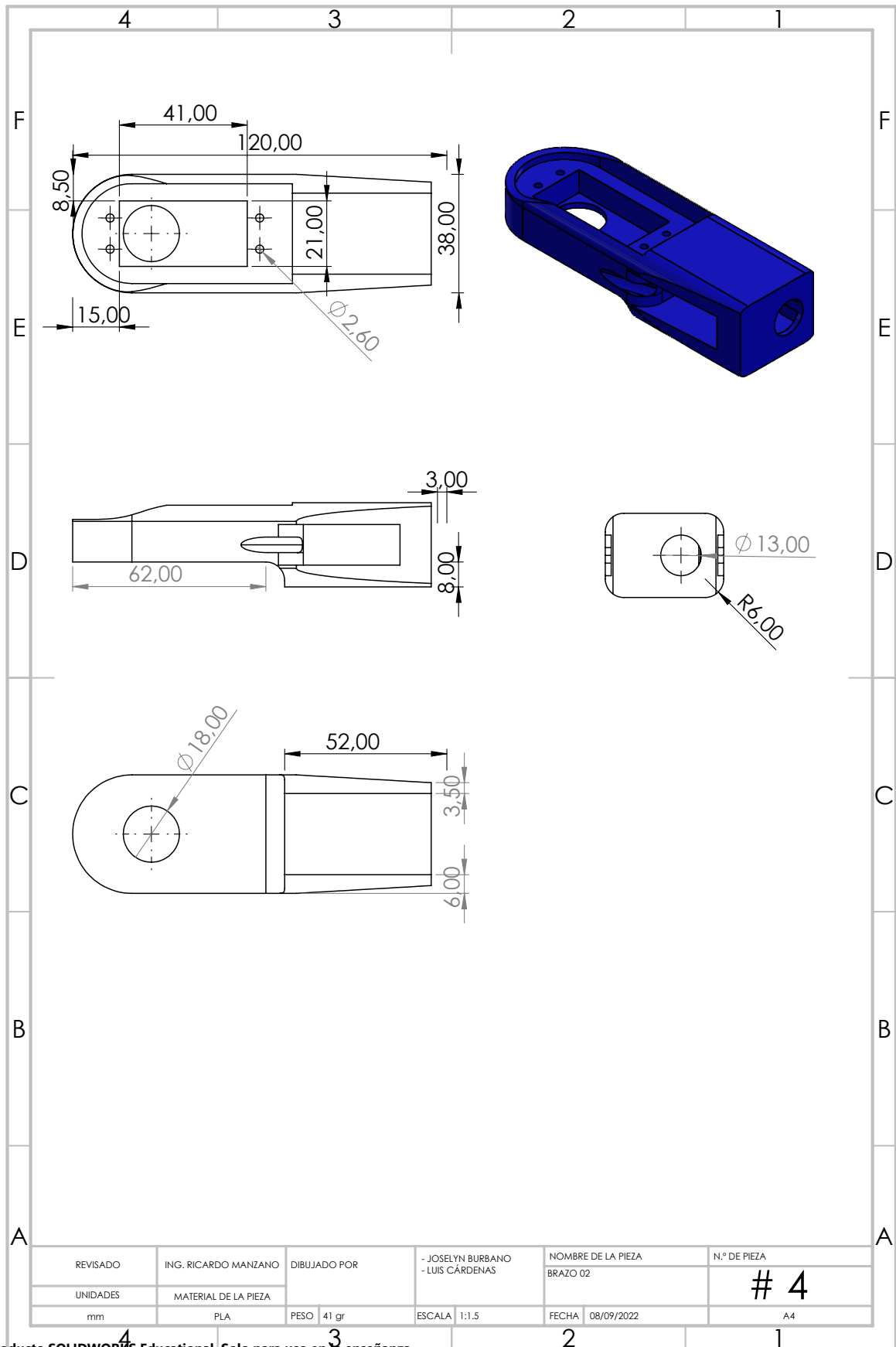
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 77. Lámina de Brazo del Robot(1/3)



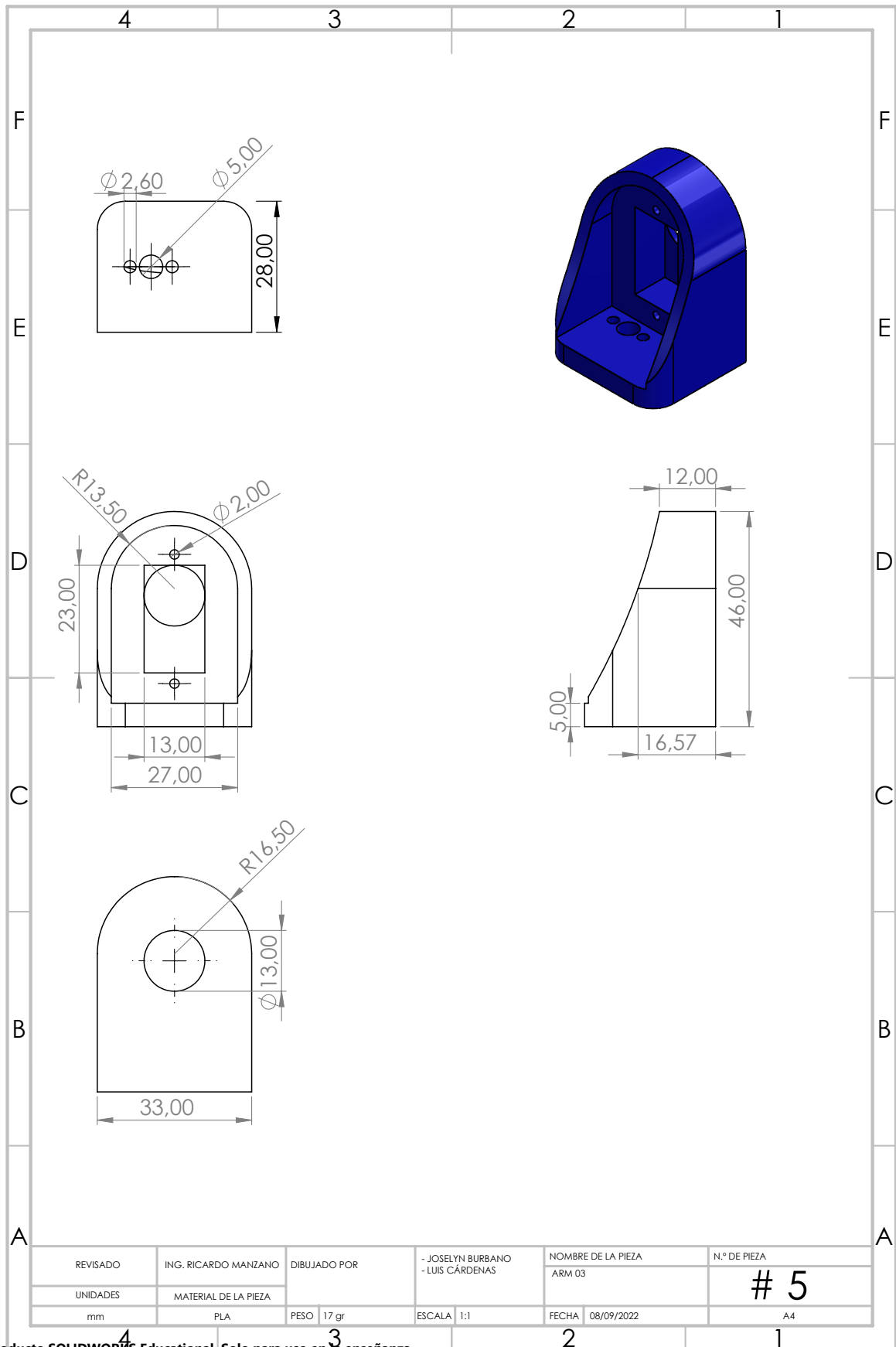
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 78. Lámina de Brazo del Robot(2/3)



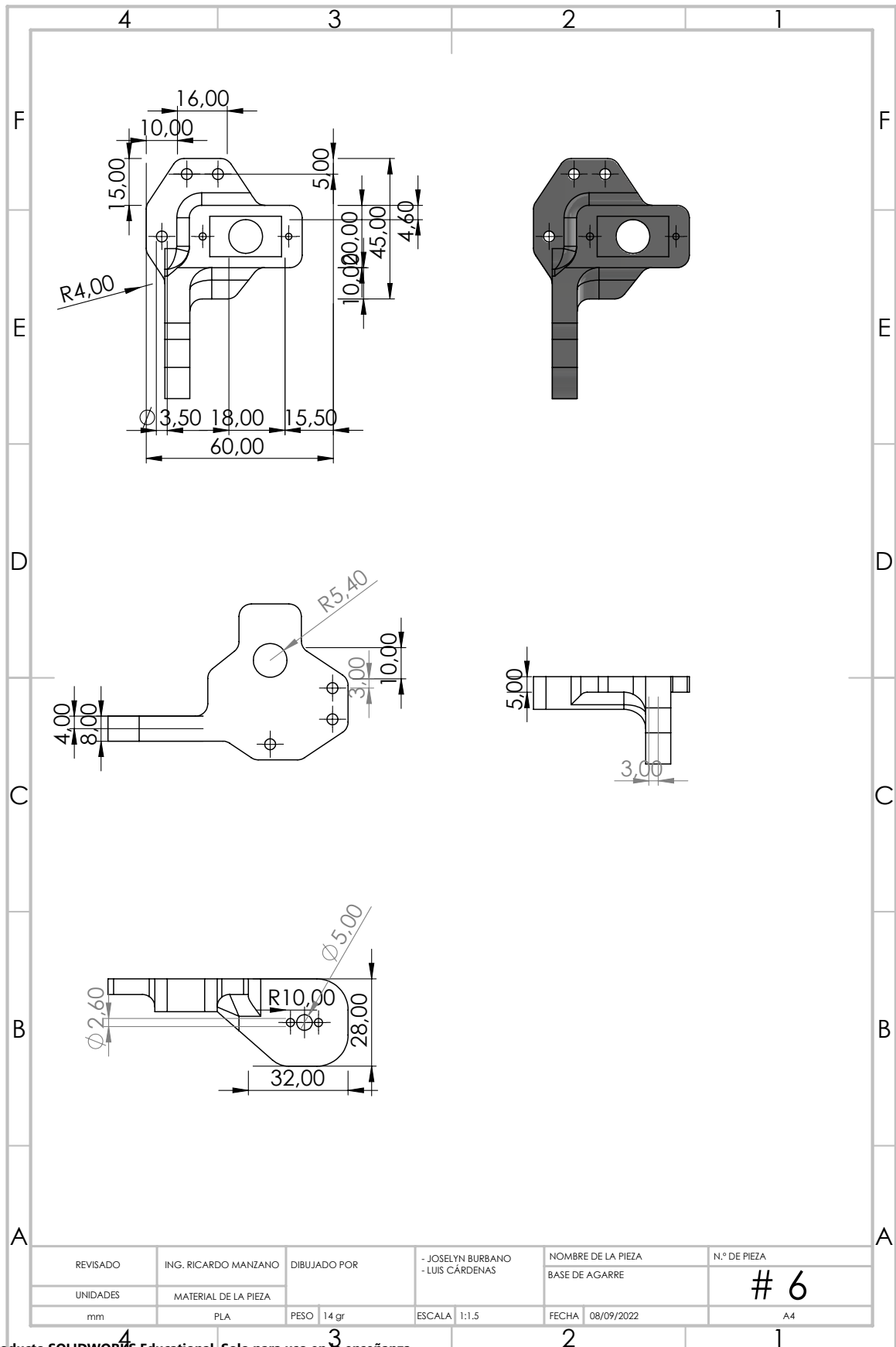
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 79. Lámina de Brazo del Robot(3/3)



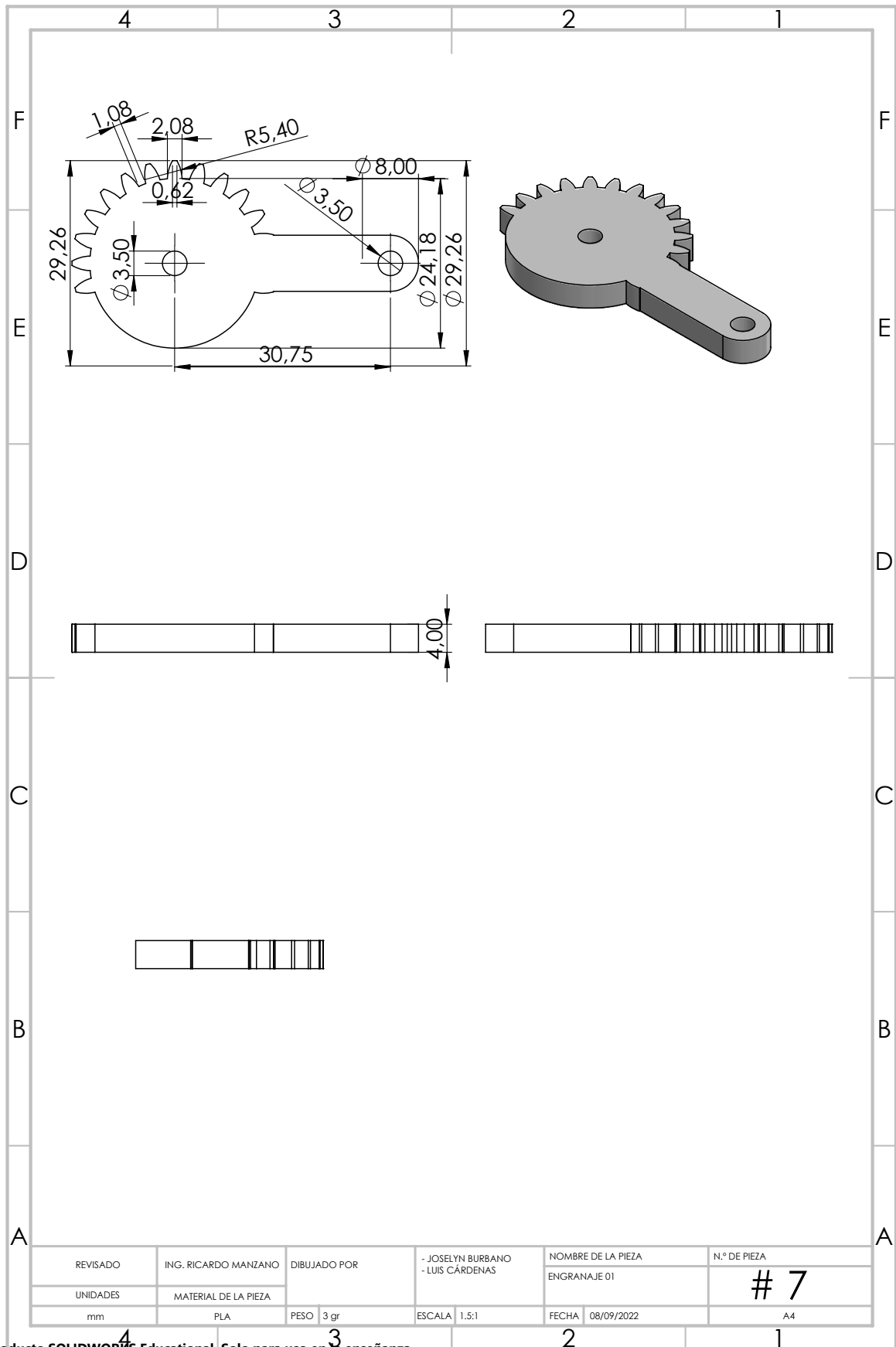
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 80. Lámina de Base de Agarre



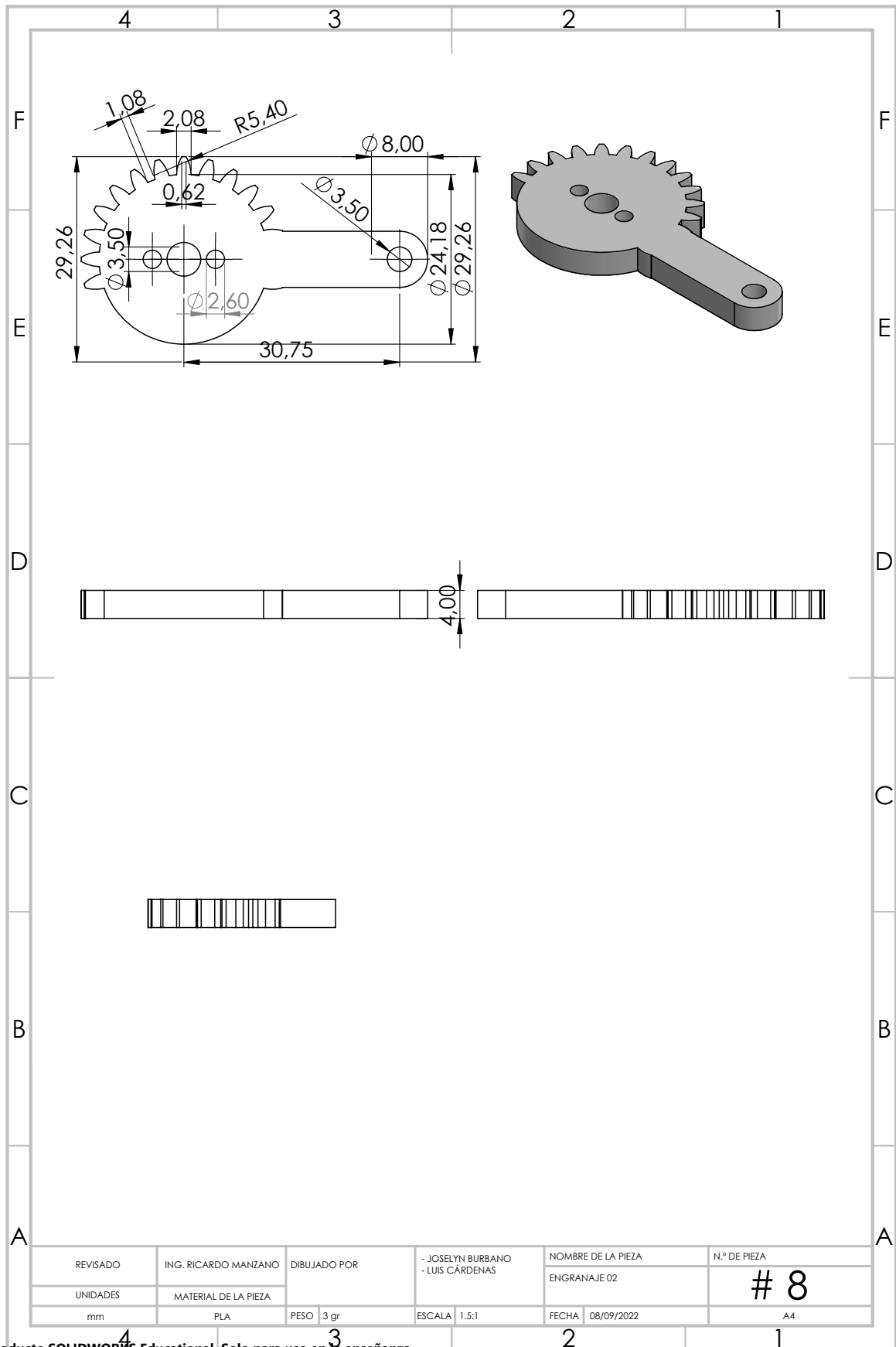
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 81. Lámina de Base de Engranaje(1/2)



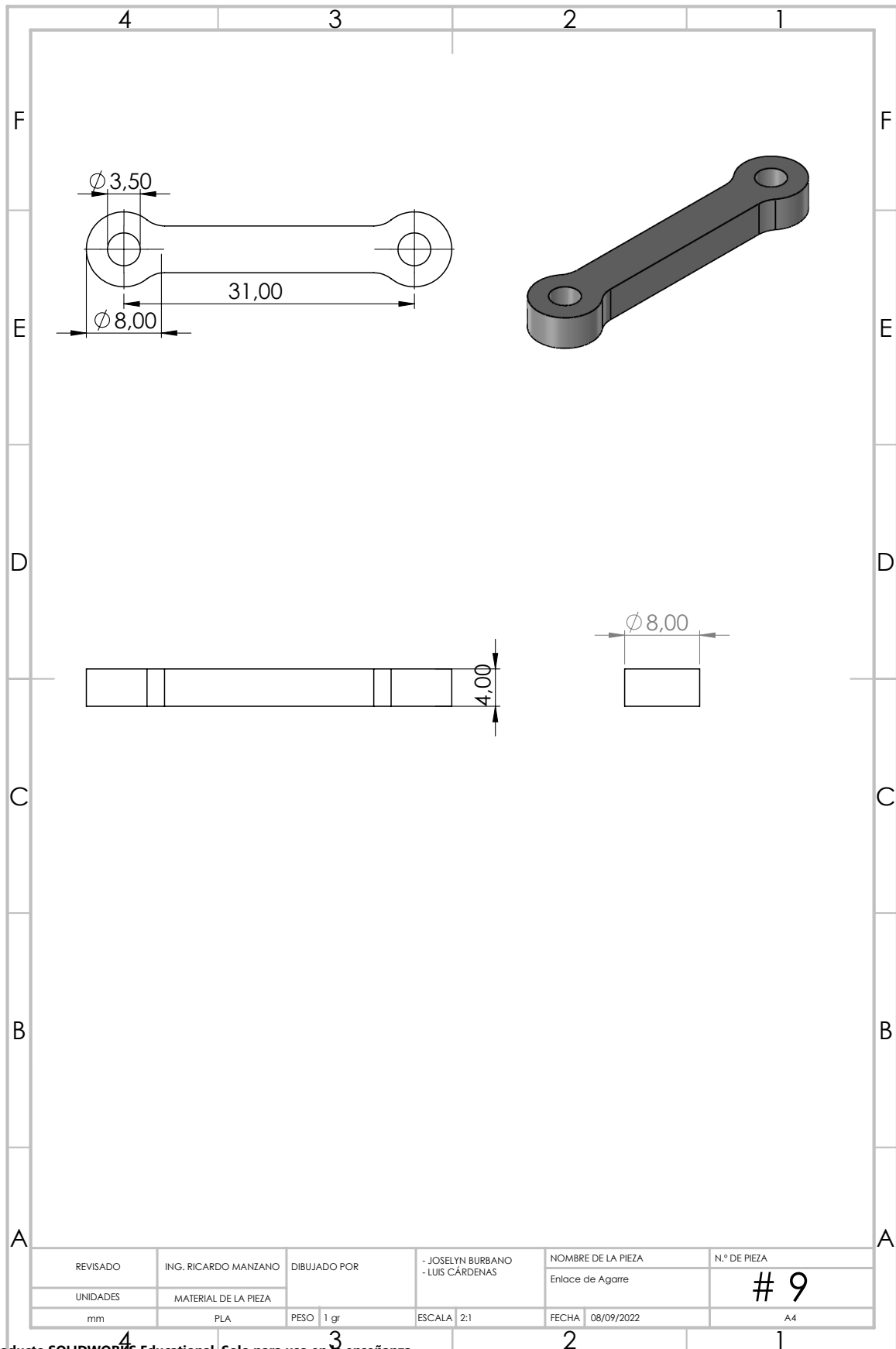
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 82. Lámina de Base de Engranaje(2/2)



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

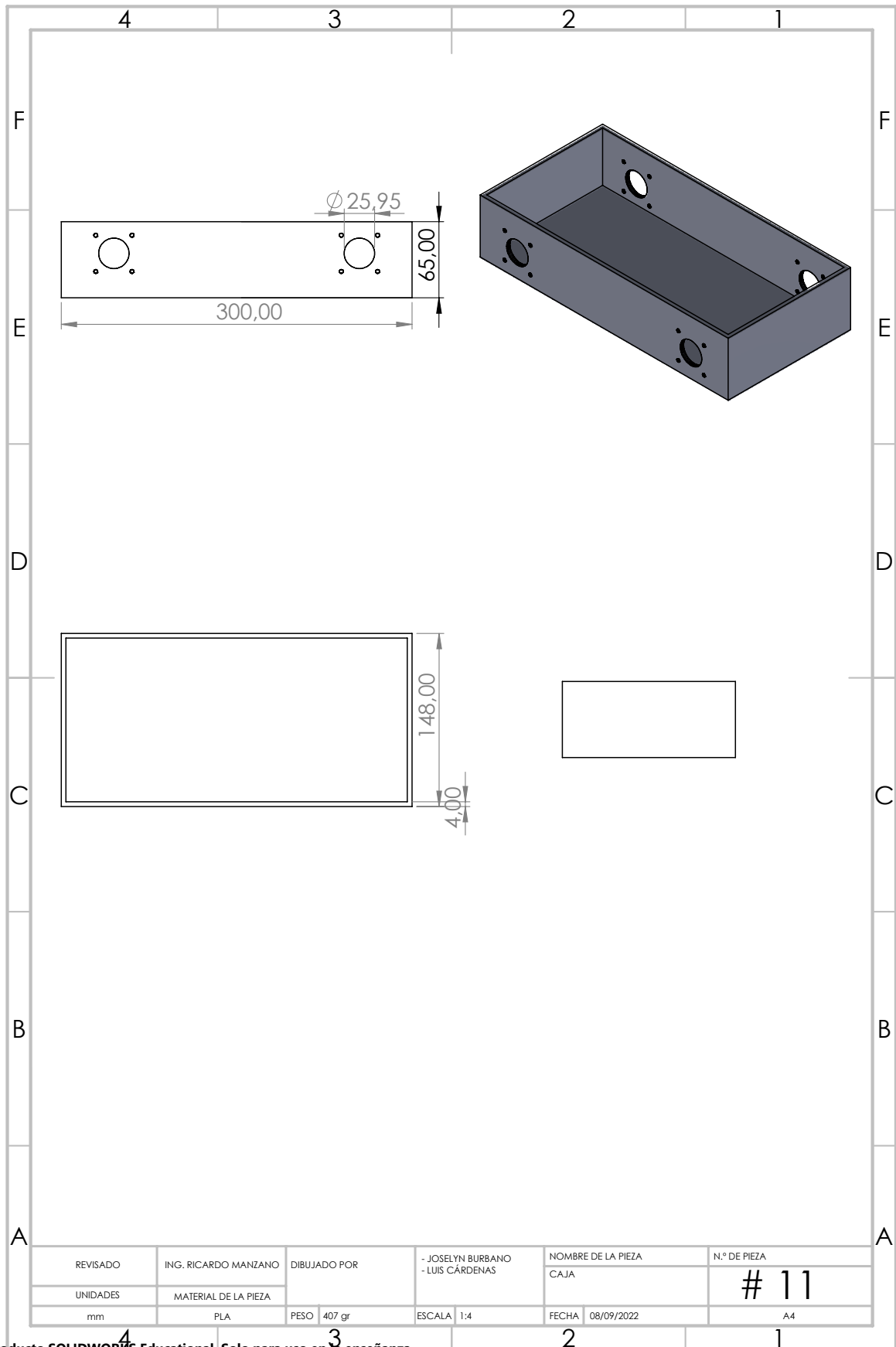
Figura 83. Lámina de Base de Enlace de Agarre



REVISADO	ING. RICARDO MANZANO	DIBUJADO POR	- JOSELYN BURBANO - LUIS CÁRDENAS	NOMBRE DE LA PIEZA	N.º DE PIEZA
UNIDADES	MATERIAL DE LA PIEZA			Enlace de Agarre	# 9
mm	PLA	PESO 1 gr	ESCALA 2:1	FECHA 08/09/2022	A4

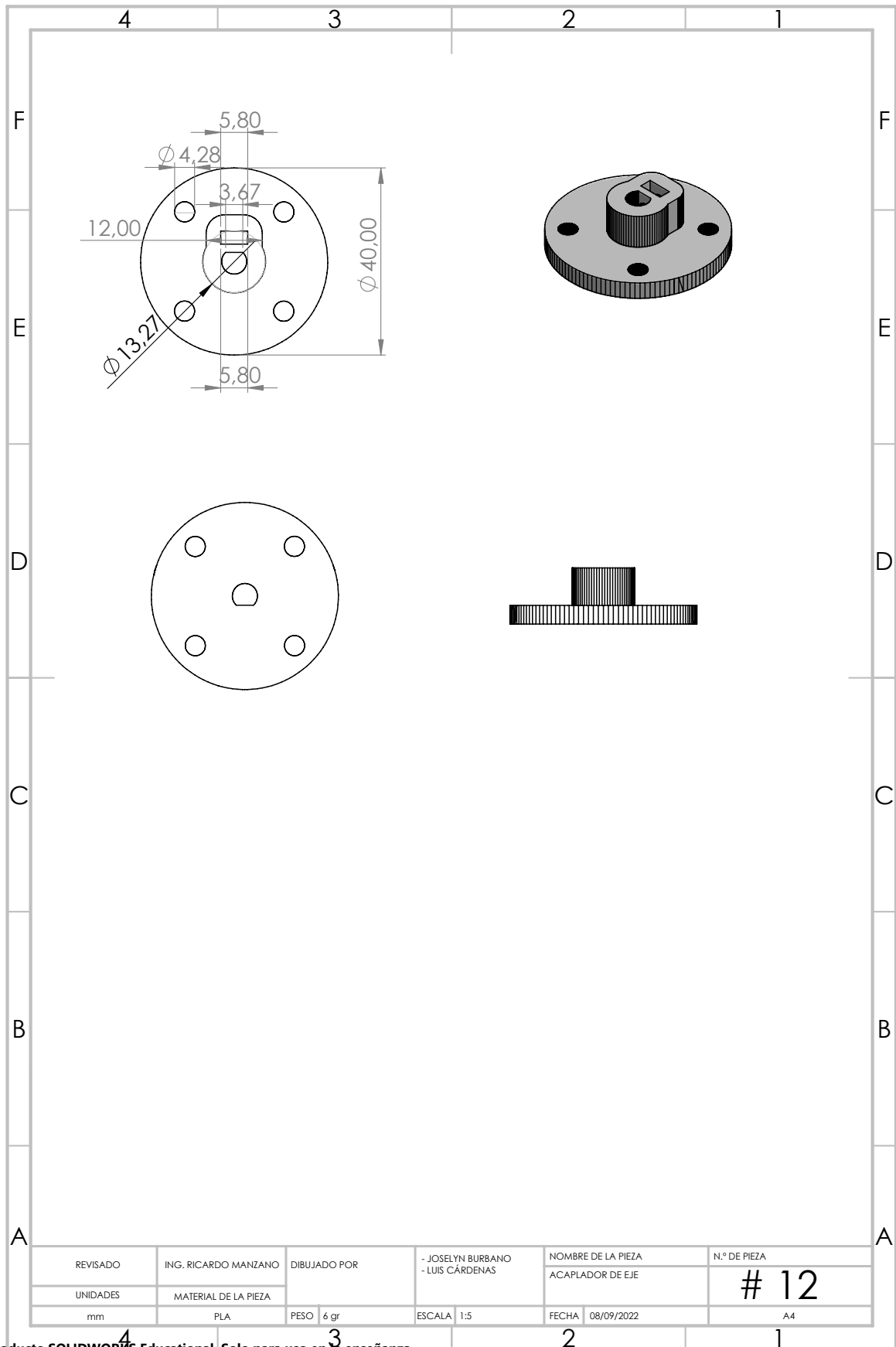
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 84. Lámina de Caja



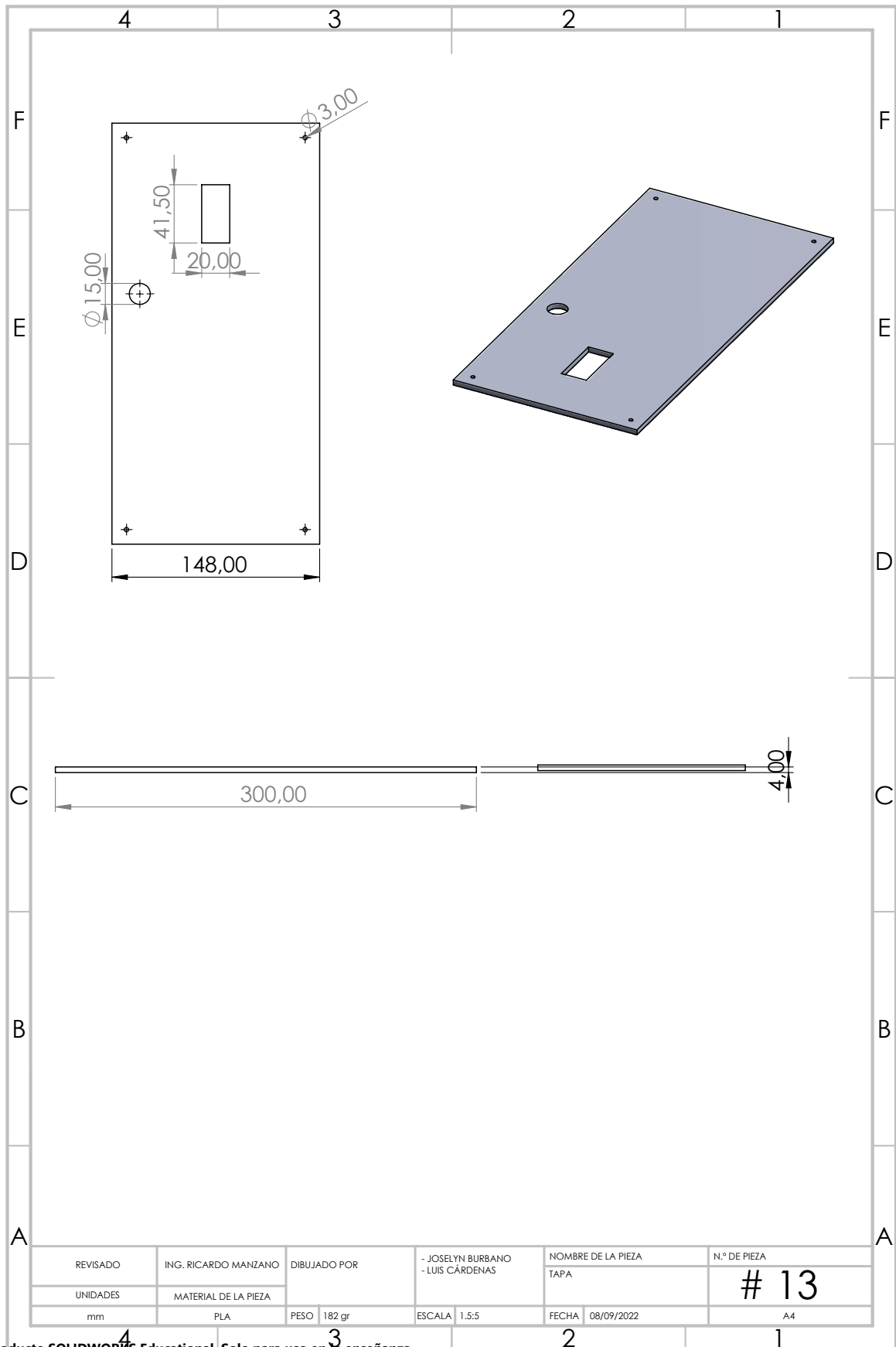
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 85. Lámina de Acoplador de Eje



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 86. Lámina de la Tapa del Robot



REVISADO	ING. RICARDO MANZANO	DIBUJADO POR	- JOSELYN BURBANO - LUIS CÁRDENAS	NOMBRE DE LA PIEZA	N.º DE PIEZA
UNIDADES	MATERIAL DE LA PIEZA			TAPA	# 13
mm	PLA	PESO 182 gr	ESCALA 1:5	FECHA 08/09/2022	A4

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 87. Robot Móvil Ensamblado

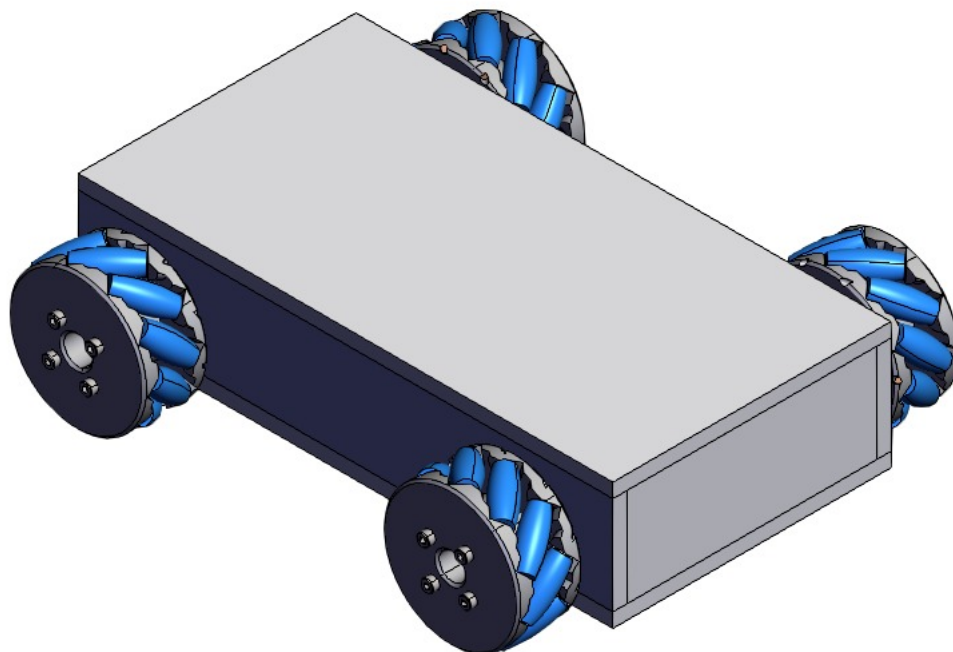


Figura 88. Robot Móvil Explosionado

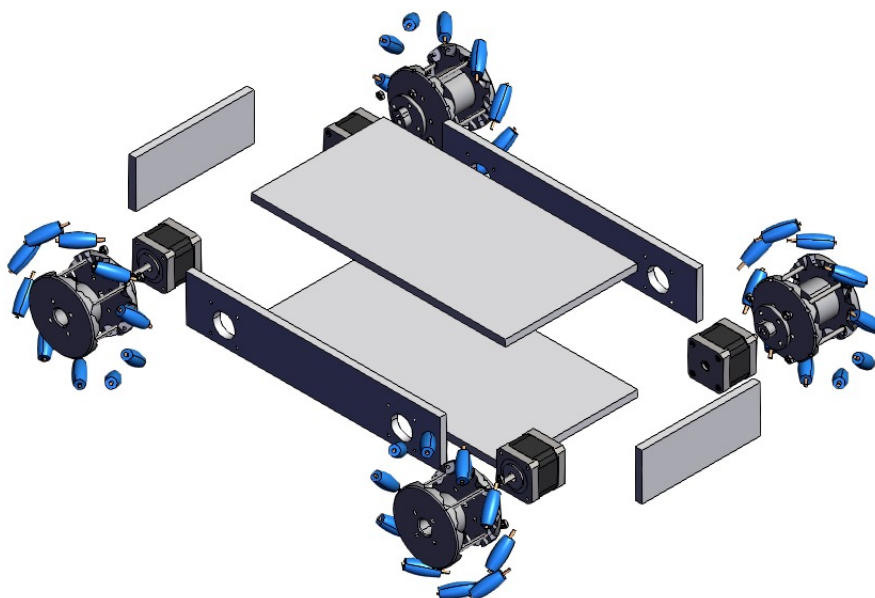


Figura 89. Brazo Robótico Ensamblado

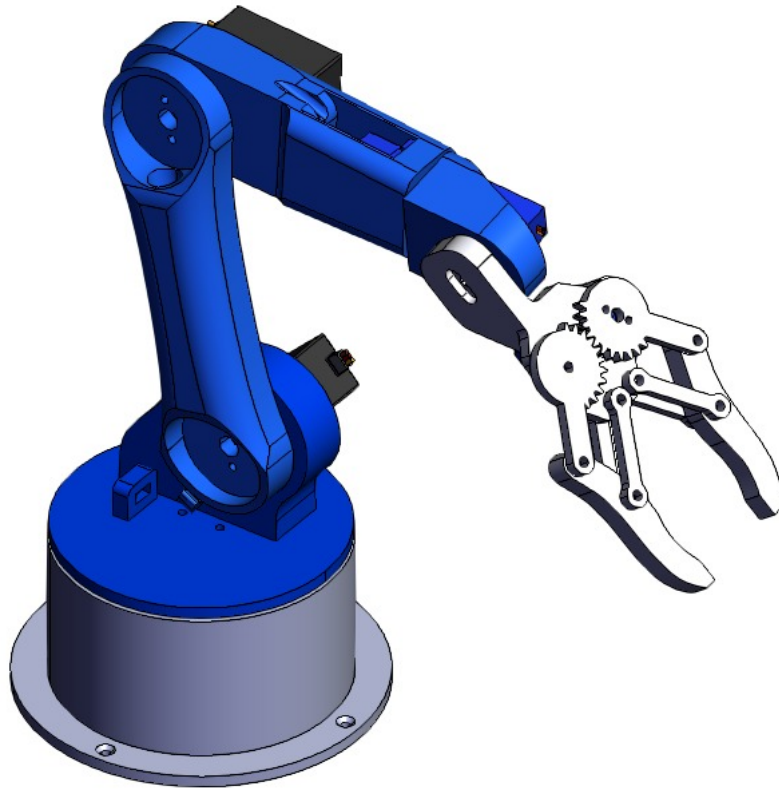


Figura 90. Brazo Robótico Explosionado

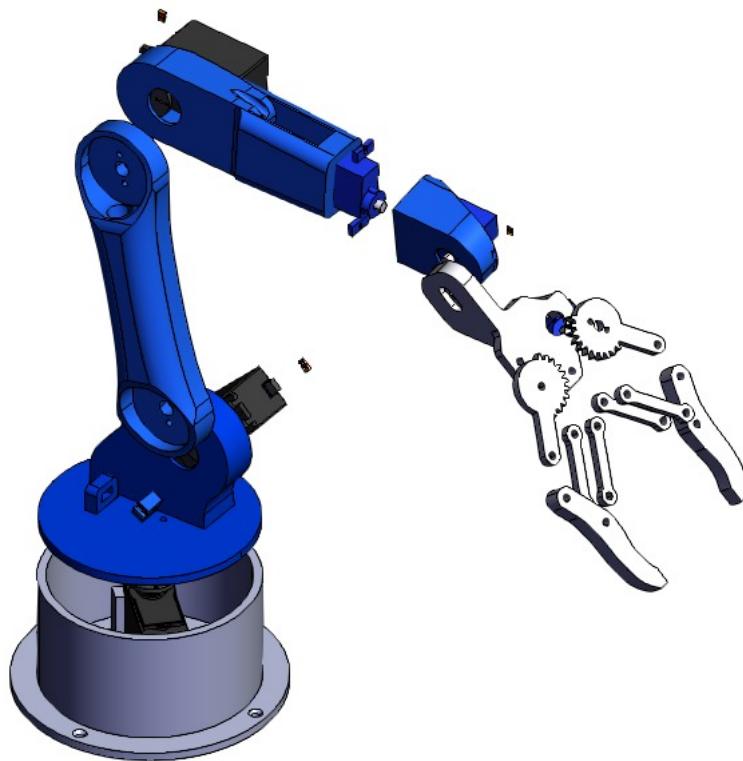


Figura 91. Modelado Ensamblado

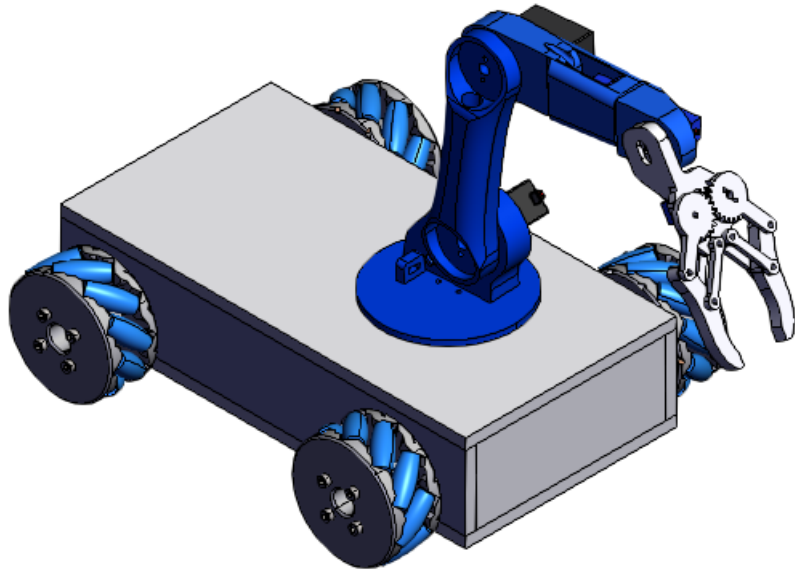


Figura 92. Modelado Ensamblado Explosionado

