



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

DISEÑO E IMPLEMENTACIÓN DE UN KIT DE ROBÓTICA MÓVIL PARA UN TALLER
INTRODUCTORIO DE ROBÓTICA DIRIGIDO A ESTUDIANTES DE BACHILLERATO DE
LA CIUDAD DE CUENCA DENTRO DEL PROYECTO PEQUEÑ@S CIENTÍFIC@S

Trabajo de titulación previo a la obtención del
título de Ingeniero en Electrónica

AUTORES: DIEGO PATRICIO ORTIZ YANZA
CARLOS GUSTAVO VÁSQUEZ LOJA

TUTOR: ING. LUIS ALFREDO CALLE ARÉVALO. MGT.

Cuenca – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros Diego Patricio Ortiz Yanza con documento de identificación N° 0107409930 y Carlos Gustavo Vásquez Loja con documento de identificación N° 0302691076, 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 de titulación.

Cuenca, 08 de abril del 2022

Atentamente,



Diego Patricio Ortiz Yanza
0107409930



Carlos Gustavo Vásquez Loja
0302691076

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

Nosotros Diego Patricio Ortiz Yanza con documento de identificación N° 0107409930 y Carlos Gustavo Vásquez Loja con documento de identificación N° 0302691076 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 Proyecto Técnico: “Diseño e implementación de un kit de robótica móvil para un taller introductorio de robótica dirigido a estudiantes de bachillerato de la ciudad de Cuenca dentro del proyecto Pequeñ@s Científic@s”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, 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 final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 08 de abril del 2022

Atentamente,



Diego Patricio Ortiz Yanza
0107409930



Carlos Gustavo Vásquez Loja
0302691076

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo Luis Alfredo Calle Arévalo con documento de identificación N° 0105109847, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría, fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN KIT DE ROBÓTICA MÓVIL PARA UN TALLER INTRODUCTORIO DE ROBÓTICA DIRIGIDO A ESTUDIANTES DE BACHILLERATO DE LA CIUDAD DE CUENCA DENTRO DEL PROYECTO PEQUEÑ@S CIENTÍFIC@S, realizado por Diego Patricio Ortiz Yanza con documento de identificación N° 0107409930 y Carlos Gustavo Vásquez Loja con documento de identificación N° 0302691076, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 08 de abril del 2022

Atentamente,



Ing. Luis Alfredo Calle Arévalo Mgt.

0105109847

Dedicatoria

El presente trabajo está dedicado a Dios por mantenernos con fortaleza en los momentos más duros que se atraviesan. A mis padres que con su amor y apoyo fue posible llegar a cumplir una meta más en mi vida.

Diego Patricio Ortiz Yanza

Este trabajo y todos mis ciclos de estancia en la Universidad está dedicado a Jean Marck Roussel y Diane Gauthier quienes patrocinaron mis estudios y me enseñaron el significado de la determinación y como perseverar en aquello que anhelamos.

Carlos Gustavo Vásquez Loja

Agradecimientos

Agradezco a Dios por bendecirnos con la vida y darnos la fuerza para culminar este trabajo. A mis padres por el apoyo en todos estos años de la carrera universitaria. Al proyecto “Pequeñ@s Científic@s” por acogernos en sus instalaciones y brindar lo necesario para cumplir con este trabajo. Y a nuestro tutor Ing. Luis Calle por confiar en nosotros y haber compartido sus conocimientos y experiencia para guiar este trabajo.

Diego Patricio Ortiz Yanza

Agradesco a mis padres quienes estuvieron a mi lado toda mi vida, a mi pareja y amigos porque sin ellos esto no podía ser posible. Al Ing. Luis Calle por darnos su confianza y apoyarnos en la realización de este proyecto de titulación. Y a Jean y Dian por brindarme la oportunidad de poder culminar mis estudios de pregrado.

Carlos Gustavo Vásquez Loja

Índice

Certificado de responsabilidad y auditoria	I
Certificado de cesión de derechos de autor	II
Certificación de dirección	III
Dedicatoria	IV
Agradecimientos	V
Resumen	XV
Abstract	XVI
Introducción	XVII
Antecedentes	XVIII
Justificación	XIX
Objetivos	XX
0.1. Objetivo General	XX
0.2. Objetivos Específicos	XX
1. CAPÍTULO 1: EDUCACIÓN STEM Y LA ROBOTICA	1
1.1. Educación STEM	1
1.2. Educación STEM y la robótica	2
1.3. Robot omnidireccional	3
1.3.1. Tipos de ruedas omnidireccionales	3
1.3.2. Disposición de las ruedas	5
1.4. Movimiento de robots con ruedas mecanum.	7
2. CAPÍTULO 2: DISEÑO Y FABRICACIÓN DEL PROTOTIPO DEL RO- BOT OMNIDIRECCIONAL	9
2.1. Descripción general del robot	9
2.2. Diseño de la estructura inferior	9
2.2.1. Uniones centrales	10

2.3.	Diseño de las paredes del chasis	11
2.4.	Diseño de la cubierta del chasis	13
2.5.	Diseño de módulo del efector final	13
2.6.	Diseño de módulo led	15
2.7.	Diseño y fabricación de la placa electrónica	16
2.7.1.	Reglas de enrutamiento	20
2.8.	Fabricación de las piezas del cuerpo y módulos del robot	22
3.	CAPÍTULO 3: PROGRAMACIÓN	24
3.1.	Programación del control con comunicación bluetooth low energy (BLE)	24
3.1.1.	Diseño de la interfaz de la app	24
3.1.2.	Programación del control	26
3.2.	Cinemática del robot omnidireccional	34
3.3.	Identificación de la planta	35
3.4.	Controlador PI	37
3.5.	Librería para la programación en el Arduino nano 33 BLE	39
3.6.	Programación del firmware del robot	43
3.6.1.	Programación del control de velocidad	43
3.7.	Programación de la comunicación I2C	44
4.	CAPÍTULO 4: MANUAL PRACTICO PARA ENSAMBLE PROGRAMACIÓN Y USO DE MÓDULOS DEL KIT ROBÓTICO	47
4.1.	Ensamble del cuerpo del robot	47
4.1.1.	Ensamble de la base	47
4.1.2.	Ensamble de las paredes y acople de ruedas.	53
4.1.3.	Ensamble de la cubierta del robot	64
4.1.4.	Ensamble final del cuerpo del robot	68
4.2.	Ensamble de componentes en la tarjeta electrónica	68
4.2.1.	Colocación de la batería	77
4.2.2.	Colocación de la tarjeta electrónica	78
4.2.3.	Conexión de motores y batería	79
4.3.	Programación para movimientos del robot	82
4.4.	Módulo del efector final, ensamble y programación	85
4.4.1.	Instalación del módulo del efector final	85
4.4.2.	Programación del módulo del efector final	86

4.5. Módulo led, ensamble y programación	87
4.5.1. Instalación del módulo led	88
4.5.2. Programación del módulo led	89
4.6. Programación para control por bluetooth	90
5. CAPÍTULO 5: PRUEBAS Y ANÁLISIS DE RESULTADOS	93
5.1. Armado de la tarjeta electrónica y kit robótico completo	93
5.2. Aplicativo móvil	95
5.3. Movimiento y módulo led	96
5.4. Módulo del efector final	98
6. CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES	99
6.1. Conclusiones	99
6.2. Recomendaciones	100
7. TRABAJO FUTURO	100
Referencias	102
ANEXOS	108

Lista de Tablas

1.	Propuesta de la educación STEM.	1
2.	Piezas necesarias para constituir la estructura base del robot.	10
3.	Fórmulas para generar un engranaje con Autodesk inventor.	14
4.	Valores de ruteo (mm).	20
5.	Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta.	38
6.	Parámetros obtenidos para controlador PI.	38
7.	Funciones del bluetooth.	41
8.	Funciones para el movimiento del robot.	41
9.	Tabla letras asignadas para los movimientos del robot omnidireccional.	41
10.	Funciones para el módulo del efector final.	42
11.	Letras asignadas para las acciones del módulo gripper.	42
12.	Funciones para el módulo led.	42
13.	Valores asignados para el color del módulo led.	43
14.	Pernos de que se emplearán en la sujeción de la base.	47
15.	Piezas que forman la estructura inferior del robot.	48
16.	Pernos que se emplearán en la sujeción de las paredes con la base.	53
17.	Piezas que forman la estructura lateral del robot.	55
18.	Pernos de que se emplearán en la sujeción de la cubierta.	64
19.	Piezas que forman la cubierta del robot.	65
20.	Módulos de la tarjeta electrónica parte 1.	69
21.	Módulos de la tarjeta electrónica parte 2.	70
22.	Funciones para el desplazamiento del robot.	83
23.	Letras asignadas al movimiento del robot.	83
24.	Códigos para movimientos.	84
25.	Funciones para uso del módulo gripper.	87
26.	Funciones para uso del módulo led.	90
27.	Funciones para comunicación bluetooth.	91
28.	Materiales utilizados en el proyecto.	103
29.	Lista de pernos empleados para el armado del robot.	104

Lista de Figuras

1.	Amplio abanico de herramientas digitales en el aula STEM.	2
2.	Actividades STEM aplicadas a la formación inicial de profesores de ciencias, esquema de robot utilizado y desafío.	3
3.	Rueda omni.	4
4.	Rueda mecanum	4
5.	Rueda omni ball	5
6.	Rueda omni-crawl	5
7.	3WD chasis de robot inteligente Omni Wheel(A),Chassis de Robot Triangular Omni-Direccional 3WD (B)	6
8.	3WD chasis de robot inteligente Omni Wheel(A),Chassis de Robot Triangular Omni-Direccional 3WD (B)	6
9.	Ruedas mecanum (A), Disposición de las ruedas, vista superior de chasis osoyo Modelo 2019016600 (B)	7
10.	Movimiento del robot según la dirección y velocidad angular de las ruedas, configuración 1.	7
11.	Movimiento del robot según la dirección y velocidad angular de las ruedas, configuración 2.	8
12.	Vista general de las partes del prototipo del robot omnidireccional	9
13.	Estructura inferior (soporte principal).	10
14.	Vista del espacio disponible para la batería en la estructura inferior.	11
15.	Paredes del robot.	11
16.	Pared frontal del robot.	12
17.	Paredes laterales modulares.	12
18.	Tapa de acrílico para protección de componentes y sujeción de módulo led.	13
19.	Variables descritas para diseñar un engranaje.	14
20.	Módulo efector final (gripper).	15
21.	Piezas que conforman el módulo led.	15
22.	Soporte de módulo led.	16
23.	Diagrama de bloques del funcionamiento del robot	16
24.	Vista 2D footprint tarjeta electrónica.	18
25.	Vista 3D superior tarjeta electrónica.	19
26.	Vista 3D inferior tarjeta electrónica.	20
27.	Valores minimos que puede tener una pista.	21

28.	Tarjeta impresa, módulos electrónicos, conectores molex y peinetas.	21
29.	Base de impresión Prusa Mini (A), base de impresión MK3S (B).	22
30.	Parámetros para la impresión.	22
31.	Piezas impresas con soportes.	23
32.	Herramientas disponibles para crear APP con sistema operativo Android . .	24
33.	Interfaz del control remoto del robot	25
34.	Búsqueda de dispositivos bluetooth disponibles.	26
35.	Definición de varibales globales.	27
36.	Pantalla de inicio.	27
37.	Establecimiento de conexión bluetooth.	28
38.	Cancelar búsqueda.	28
39.	Desconexión bluetooth.	29
40.	Caractéres de movimiento, estado 1.	29
41.	Caractéres de movimiento, estado 2.	30
42.	Caracteres de apertura y cierre de módulo gripper.	30
43.	Estado de conexión.	31
44.	Cambio de color de módulo led.	32
45.	Envío de caracteres por bluetooth.	33
46.	Diagrama de cuerpo de robot omnidireccional	34
47.	Respuesta ante el escalón del motor DC.	36
48.	Grafica de la respuesta al escalón, estimación con respecto a la original. . . .	37
49.	Curva de respuesta en forma S y parámetros a considerar para el Método Ziegler Nichols.	38
50.	Respuesta del sistema implementando un controlador PI (verde), sistema sin controlador PI (azul).	39
51.	Sketch de programación principal.	40
52.	Control de velocidad.	44
53.	Función I2C.	45
54.	Programación de la cinemática del robot omnidireccional.	45
55.	Programación de los movimientos del efector final.	46
56.	Llaves hexagonales tipo hallen (A), Llaves hexagonales tipo boca(B).	47
57.	Base y soporte con inserción de pernos.	49
58.	Base y soportes.	49
59.	Base y soportes con inserción de pernos.	50
60.	Bases y soportes.	50

61.	Bases y soportes con inserción de pernos.	51
62.	Bases y soportes de placa.	52
63.	Bases y soportes de placa con inserción pernos.	52
64.	Bases y soportes de placa final	53
65.	Base y paredes laterales grises.	56
66.	Base y paredes laterales grises con inserción de pernos.	56
67.	Base y paredes laterales azules.	57
68.	Base y paredes laterales azules con inserción de pernos.	57
69.	Base, paredes y motores 1 y 3.	58
70.	Base, paredes y motores 1 y 3 con inserción de pernos.	58
71.	Base, paredes y motores 2 y 4.	59
72.	Base, paredes y motores 2 y 4 con inserción de pernos.	59
73.	Base, paredes, motores y pared frontal azul.	60
74.	Base, paredes, motores y pared frontal azul con inserción de pernos, (vista superior).	60
75.	Base, paredes, motores y pared frontal azul con inserción de pernos, (vista inferior).	61
76.	Base, paredes, motores y pared frontal gris.	62
77.	Base, paredes, motores y pared frontal gris con inserción de pernos, (vista superior).	62
78.	Base, paredes, motores y pared frontal gris con inserción de pernos, (vista inferior).	63
79.	Inserción de pernos laterales para union de paredes.	63
80.	Ensamble del chasis, paredes y motores del robot.	64
81.	Ensamble de la tapa de acrílico	66
82.	Piezas para ensamble de la tapa de acrílico.	67
83.	Piezas para ensamble de la tapa de acrílico con inserción de pernos.	67
84.	Apariencia del robot al final del ensamble.	68
85.	Posición de inicio de la tarjeta electrónica.	71
86.	Colocación de Arduino nano 33 (A), orientación.	72
87.	Colocación de convertidores lógicos 8 canales (A), orientación (B).	72
88.	Colocación de convertidor lógico 4 canales (A), orientación (B).	73
89.	Colocación de módulo Xbee (A), orientación (B).	73
90.	Colocación de drives 1 y 3 (A), orientación (B).	74
91.	Colocación de drives 2 y 4 (A), orientación (B).	74

92. Colocación de mini maestro (A), orientación (B).	75
93. Colocación de regulador 7V (A), orientación (B).	75
94. Colocación de regulador 3.3V (A), orientación (B).	76
95. Colocación de regulador 5V (A), orientación (B).	76
96. Tarjeta armada (Vista superior).	77
97. Tarjeta armada (Vista inferior).	77
98. Colocación de la batería.	78
99. Posición para la colocación de la tarjeta electrónica dentro del robot.	78
100. Ubicación de los pernos para sujetar la tarjeta electrónica.	79
101. Vista de la batería y tarjeta electrónica dentro del robot.	79
102. Conexión de los cables para los cuatro motores del robot.	80
103. Orientación Motor 2 y 4 (A), Orientación Motor 1 y 3 (B).	80
104. Conexión del cable de alimentación a la batería.	81
105. Conexión del conector de alimentación a la tarjeta electrónica.	81
106. Vista de las conexiones de la batería y los motores en la tarjeta electrónica.	82
107. Diagrama de cuerpo.	82
108. Conexión del cable micro USB al microcontrolador Arduino nano 33.	84
109. Módulo del efector final con inserción de pernos.	85
110. Módulo del efector final acoplado al cuerpo del robot	86
111. Conector del módulo del efector final.	86
112. Código de programación módulo del efector final.	87
113. Alineación del módulo led, vista superior.	88
114. Alineación del módulo led, vista inferior.	88
115. Módulo led instalado.	89
116. Forma de conexión de los tres cables del módulo, verde (VCC), blanco (Din) y café (GND).	89
117. Código de programación módulo led.	90
118. Programa para comunicación bluetooth.	92
119. Armado de la tarjeta electrónica.	93
120. Kit robótico completo.	94
121. Ensamble del kit robótico.	94
122. Kit robótico armado por completo.	95
123. Aplicativo móvil.	96
124. Robot en movimiento empujando 1kg de peso.	96
125. Robot en movimiento empujando 1.8kg de peso.	97

126. Valores de los motores con el controlador vs puntos de referencia.	97
127. Módulo del efector final en funcionamiento	98
128. Pruebas de ensamble.	104
129. Impresión en material PETG	105
130. Armado del circuito eléctrico en el protoboard	105
131. Chasis del robot	106
132. Módulo del efector final	106
133. Módulo led apagado	107
134. Módulo led encendido	107
135. Cubierta de acrílico insertado un botón de encendido.	108

Resumen

En el siguiente documento se presenta el diseño y construcción de un kit robótico didáctico basado en un robot omnidireccional con cuatro ruedas mecanum que cuenta con un efector final y un módulo de anillo led neopixel. Será empleado para dictar talleres introductorios de robótica utilizando el plan de estudio STEM con el propósito es reforzar el conocimiento teórico aplicado en la práctica, como también mejorar la destreza de ensamble de piezas.

Al ser un robot dirigido al ambiente educativo se propuso un diseño modular de la estructura donde si una pieza se ve afectada las otras mantienen al robot funcional, esto fue posible gracias a la disponibilidad y prestaciones del software Autodesk inventor. Su tarjeta electrónica esta diseñada y fabricada de tal modo que los estudiantes puedan insertar los módulos electrónicos por separado.

Teniendo resuelto el apartado de hardware, la solución expuesta para el control del robot por el usuario final es la programación por medio de una librería de autoría propia para simplificar la programación y hacer mas entendible el código. Y finalmente es necesario una interfaz con la que sea posible las acciones del robot y para ello se presentara un aplicativo móvil compatible con el sistema operativo Android.

Palabras clave: robot omnidireccional, STEM, mecanum, kit robótico, software libre

Abstract

The following document presents the design and construction of a didactic robotic kit based on an omnidirectional robot with four mecanum wheels that has a final effector and a neopixel LED ring module. It will be used to teach introductory robotics workshops using the STEM curriculum with the purpose of reinforcing the theoretical knowledge applied in practice, as well as improving the skill of assembling parts.

Being a robot aimed at the educational environment, a modular design of the structure was proposed where if one piece is affected the others keep the robot functional, this was possible thanks to the availability and features of the Autodesk inventor software. Its electronic card is designed and manufactured in such a way that students can insert the electronic modules separately.

Having solved the hardware section, the solution exposed for the control of the robot by the end user is programming through a library of own authorship to simplify programming and make the code more understandable. And finally it is necessary an interface with which the actions of the robot are possible and for this a mobile application compatible with the Android operating system will be presented.

keywords: omnidirectional robot, STEM, mecanum, robotic kit, free software

INTRODUCCION

La robótica es considerada como una disciplina que requiere ciertas capacidades para su implementación. Stem ofrece soluciones prácticas y didácticas para que la enseñanza y el aprendizaje no tenga porque ser tedioso, sino muy proactivo y motivador.

Con el objetivo de ofrecer un diseño que tenga capacidad adaptativa a futuros talleres dentro del proyecto Pequeñ@s Cientific@s se tiene que partir de diferentes diseños de robots omnidireccionales que actualmente se comercializan, indagar en información sobre los distintos tipos y sus configuraciones particulares puesto que son primordial para elaborar el modelo cinemático del movimiento del robot.

Para el desarrollo de este kit robótico se emplearán hardware libre en la tarjeta electrónica que sean de fácil adquisición. Se utilizará el software libre para programación de los microcontroladores Arduino, todo esto con el fin de que el usuario final; los estudiantes beneficiarios del proyecto puedan adquirir y aprender sobre estos recursos. Con el uso del kit robótico se pretende que los estudiantes puedan implementar sus propios proyectos.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

El surgimiento de nuevas tecnologías está influyendo en la educación y la vida de las personas en todos sus aspectos. En los últimos dos años debido a una pandemia mundial se aceleró el proceso de adopción de educación 4.0 en el Ecuador dentro de un marco referencial a lo ante expuesto y ante esta situación el ministerio de educación genero un lineamiento estratégico de la agenda digital para los años 2021-2022 en donde el lineamiento estratégico 5 en su cuarto indicador propone identificar y desarrollar nuevos perfiles de cualificación profesional, a fin de impulsar nuevas carreras tecnológicas acorde a la situación actual, para facilitar el desarrollo y la inserción laboral en función a la demanda. (MINTEL, 2021). En Ecuador, para el año 2021 el porcentaje de graduados en áreas de ciencia y tecnología no supera el 12% según estadísticas de la secretaria de educación superior, ciencia, tecnología e innovación, esta situación indica que el país requiere nuevas competencias profesionales en áreas tecnológicas. (de educación, 2021)

Se plantea un presente edificador de una cuarta revolución industrial donde los robots son los protagonistas y gracias a los últimos acontecimientos mundiales tomaron mayor relevancia. Uno de los mayores desafíos para los gobiernos y las empresas es formar la fuerza laboral del futuro y, al mismo tiempo, hacer la transición a esta nueva economía ayudando a desarrollar destrezas para conseguir una sinergia y evitar el desplazamiento de mano de obra. (Schwab, 2016).

La innovación educativa que demanda la sociedad actual pasa necesariamente por la introducción paulatina de asignaturas relacionadas con las TICs en las que los alumnos dejen de ser meros espectadores de los profesores y puedan diseñar y construir sus propios proyectos. (Jorge Noboa, 2016). Un laboratorio virtual donde los estudiantes puedan practicar, simular y jugar es de gran ventaja a la hora de aprender, no obstante, se ha visto la necesidad el uso de hardware en diciplinas como la robótica, donde el estudiante debe necesariamente observar, manipular y entender el comportamiento de sus creaciones para poder hacer ajustes.

EL proyecto de vinculación con la sociedad “Pequeñ@s científic@s”, desde el 2012 hasta la fecha de realizado este documento, ha llevado a cabo 170 talleres de ciencia y tecnología, en el cual han participado 1781 estudiantes con edades de 9 a 13 años.

Justificación(importancia y Alcance)

La robótica es un método interdisciplinario en el que se trabajan las áreas de Matemáticas, Tecnología, Ciencias e Ingeniería. Su carácter transversal posibilita que los jóvenes también desarrollen el pensamiento lógico, la imaginación y la lingüística (incluidos términos en inglés).(Schwab, 2016).

Una de las tendencias más en auge es la del aprendizaje experimental o aprendizaje por la experiencia, donde los usuarios se encuentran como actores y no como receptores pasivos de la información proporcionada. El aprendizaje experimental implica un alto grado de interactividad de los usuarios, tanto con otros usuarios, así como con los objetos disponibles en el ambiente virtual y físico. (Jorge Noboa, 2016).

Las instituciones educativas acuden al proyecto de “Pequeñ@s científic@s” por la necesidad de incorporar el aprendizaje de temas sobre ingeniería y tecnología, ya que en su plan de estudio no contemplan asignaturas relacionadas con estos temas, por tal razón han realizado convenios con el proyecto para que sus estudiantes participen en sus talleres. Con esto en mente, el proyecto “Pequeñ@s científic@s” pretende incluir a estudiantes con edades de entre 14 a 18 años en nuevos talleres enfocados en su desarrollo académico de bachillerato para animar a estos estudiantes a optar por carreras de ingeniería.

La implementación de el kit propuesto permitirá a los estudiantes tener un acercamiento físico con el hardware de un robot móvil quitando o añadiendo sus componentes, de la misma forma con el software donde tendrán la posibilidad de entender los modelos matemáticos y líneas de código que hacen que el robot pueda realizar tareas específicas.

Se empleará un control basado en Arduino, siendo este un entorno de desarrollo y código abierto para introducir a los estudiantes al ambiente de la programación y la robótica. Toda la información respecto al uso y la configuración del software y hardware está disponible en el internet con mucho material de apoyo como imágenes, videos, esquemas, documentos completos que hablan sobre proyectos y como usar esa misma información en otras aplicaciones.(Beiroa, 2020).

Cabe recalcar que con el proyecto se pretende entregar una motivación extrínseca que provendrá de estímulos de satisfacción de mover un objeto, completar una secuencia, cargar un programa entre otros objetivos que se detallaran en un manual de usuario. Jean Piaget es un fiel creyente de las posibles adaptaciones del individuo a su medio y de cómo los estímulos físicos, afectivos, posibilidades fisiológicas influyen para que este puede desarrollar sus distintas destrezas.(Elena Durán, 2010).

Objetivos

0.1. Objetivo General

Desarrollar un kit modular de un robot móvil omnidireccional para un taller introductorio de robótica dirigido a estudiantes de bachillerato de la ciudad de Cuenca dentro del proyecto “Pequeñ@s científic@s”.

0.2. Objetivos Específicos

- Diseñar y construir las piezas de la estructura mecánica y la tarjeta electrónica para el robot.
- Escribir un manual de prácticas detallando información sobre la construcción y la programación del kit robótico.
- Verificar el funcionamiento del kit de robótica móvil mediante pruebas de ensamble de piezas y elementos electrónicos.

1. CAPÍTULO 1: EDUCACIÓN STEM Y LA ROBOTICA

1.1. Educación STEM

STEM son las siglas en inglés para Ciencia, Tecnología, Ingeniería y Matemáticas; este acrónimo se utiliza en situaciones donde se juntan estas cuatro disciplinas. La propuesta de educación basada en este concepto se puede entender, en el contexto de las ciencias integradas, como una aproximación para la enseñanza de las ciencias, tecnologías, ingenierías y matemáticas de forma interdisciplinaria, donde la rigurosidad de los conceptos científicos es desarrollada mediante actividades didácticas inmersivas aplicadas al mundo real, es decir se enfoca en el desarrollo del conocimiento de los estudiantes a través de una aplicación práctica.(Yonnhatan García Cartagena, 2017)

Habilidades	Beneficios	Retos
Investigación.	Desarrollo de la capacidad para resolver problemas.	Dar respuesta a desafíos económicos globales. Solucionar problemas ambientales y tecnológicos. Desarrollar conocimientos y competencias.
Pensamiento crítico.	Retención del conocimiento a largo plazo.	
Solución de problemas.	Promueve el aprendizaje proactivo.	
Creatividad.	Fomenta el trabajo en equipo y toma de decisiones de manera conjunta.	
Comunicación.		
Colaboración.		

Tabla 1: Propuesta de la educación STEM.

Conforme lo expuesto en la Tabla 1, quizá el mayor reto al que se enfrenta este y otros modelos de educación es la motivación de alumnos y docentes. Ante el desafío de la implementación del modelo de enseñanza se otorga un gran porcentaje de responsabilidad a los recursos que se tengan a disposición en las instalaciones educativas.(Rosmari García Mejia, 2020)

Recientemente se ha considerado a las artes como otra de las disciplinas que no pueden ser marginadas, dando origen al enfoque STEAM (del inglés STEM + Arts). A partir del trabajo educativo bajo estos dos enfoques, se ha evidenciado que los estudiantes logran comprender conceptos de forma práctica y más creativa.(Yonnhatan García Cartagena, 2017)

1.2. Educación STEM y la robótica

Una de las herramientas consideradas de gran potencial pedagógico por estas nuevas corrientes innovadoras son precisamente los dispositivos móviles, dado que, además de ser de gran relevancia para nuestros alumnos fuera del aula, brindan un sinnúmero de ventajas, entre las cuales se encuentran la flexibilidad, inmediatez, portabilidad, motivación, dinamismo, accesibilidad a variedad de formatos de contenido, autonomía, etc.(Cáceres, 2019)

Se ha llegado a hablar de nuevas competencias “diseño e invención”, donde el conocimiento científico, matemático y tecnológico juega un importante papel en el uso y adaptación de las tecnologías digitales para la capacidad de personalización e incluso fabricación digital de apps, video-juegos, entornos digitales, simulaciones y otras herramientas . Estas competencias han dado paso al nacimiento de laboratorios de creación digital (denominados FAB LAB) mismos que son característicos por emplear hardware y software libre; tal es el caso de las placas Arduino que permiten la creación y personalización de dispositivos electrónicos.(Víctor Lopez, 2020)



Figura 1: Amplio abanico de herramientas digitales en el aula STEM.

Fuente:(Víctor Lopez, 2020)

Un laboratorio tiene mucha implicancia al momento de implementar esta nueva técnica de aprendizaje, puesto que STEM depende mucho de las instalaciones y los recursos para su

enseñanza, en la Figura 2, se aprecia un breve resumen de su implementación. La robótica es una disciplina que combina un sin número de ciencias, mismas que con apoyo de la metodología STEM ayudara al aporte y crecimiento de esta nueva revolución.

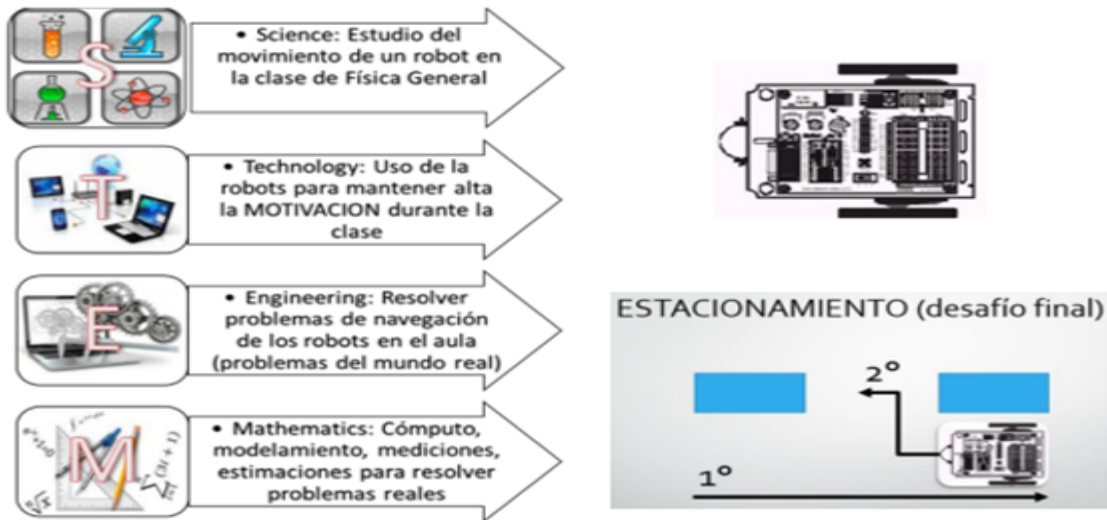


Figura 2: Actividades STEM aplicadas a la formación inicial de profesores de ciencias, esquema de robot utilizado y desafío.

Fuente:(Yonnhatan García Cartagena, 2017)

1.3. Robot omnidireccional

Se los conoce también como holonómico, pertenece a los robots móviles con ruedas. Las ruedas que emplean son su principal diferenciador del resto de robots, cada rueda proporciona al robot una fuerza normal al eje del motor y paralela a la superficie sobre la cual se desplaza y la suma de ellas permite la traslación y rotación de la estructura.

Su capacidad de moverse en todas las direcciones sin la necesidad de realizar giros como los vehículos terrestres convencionales se los logra gracias a la forma y disposición de las ruedas en el chasis del robot, su configuración mínima es de 3 ruedas y la más común es de 4 ruedas.

1.3.1. Tipos de ruedas omnidireccionales

Omni Wheel, son ruedas convencionales con rodillos en su perímetro montados perpendiculares al eje de rotación de la rueda cuyo comportamiento al rodar otorgan tracción en sentido

tangencial, mientras que en sentido transversal sus rodillos giran libremente provocando el desplazamiento.(Jhonny Ortega, 2017)



Figura 3: Rueda omni.

Mecanum wheel, comparte similitud con una rueda omni wheel con la diferencia de que sus rodillos están a $\pm 45^\circ$. Al girar producen una fuerza perpendicular al eje del rodillo lo que provoca la rotación del rodillo y una fuerza paralela al eje del rodillo ejerce fuerza sobre la rueda. (Tzafestas, 2014)



Figura 4: Rueda mecanum

Fuente:(Tzafestas, 2014)

Rueda omni ball (de bola o esférica) no cuenta con restricciones al movimiento, el eje de rotación de la rueda puede tener cualquier dirección arbitraria. Una forma de conseguir esto es mediante el uso de un anillo activo accionado por un motor y una caja de engranajes para transmitir potencia a la bola a través de rodillos y fricción, que es libre de girar en cualquier dirección instantáneamente. Debido a su difícil construcción, la rueda de bolas es muy raramente utilizado en la práctica.(Tzafestas, 2014)

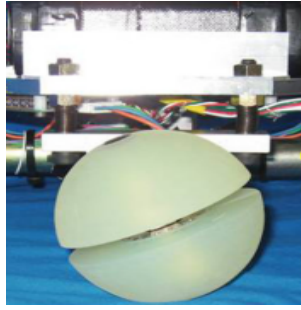


Figura 5: Rueda omni ball

Fuente:(Tzafestas, 2014)

Omni crawl, este dispositivo es único, ya que utiliza dos orugas cilíndricas que toman prestadas propiedades de las inusuales ruedas en forma de bola de dos piezas del equipo. Las Omni-Balls consisten en dos ruedas hemisféricas a juego conectadas entre sí a cada lado de un eje corto. Las mitades separadas pueden girar independientemente unas de otras, o en tándem como una esfera completa.(Kenjiro Tadakuma, 2008)

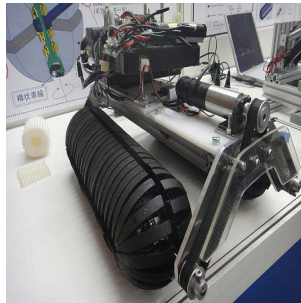


Figura 6: Rueda omni-crawl

Fuente:(Kenjiro Tadakuma, 2008)

1.3.2. Disposición de las ruedas

Si se emplean Omni Wheel se debe tener presente que existen diferentes formas de configurarlas. La configuración debe obedecer a una regla importante, la cual dice que: no es posible que las ruedas puedan ubicarse todas paralelamente. Siguiendo esta regla se puede encontrar configuraciones como de la Figura 7 (A), cuya configuración hace que el control tenga cierto grado de dificultad, esto no sucede para el caso de la Figura 7(B), misma que cuenta con una disposición axial.

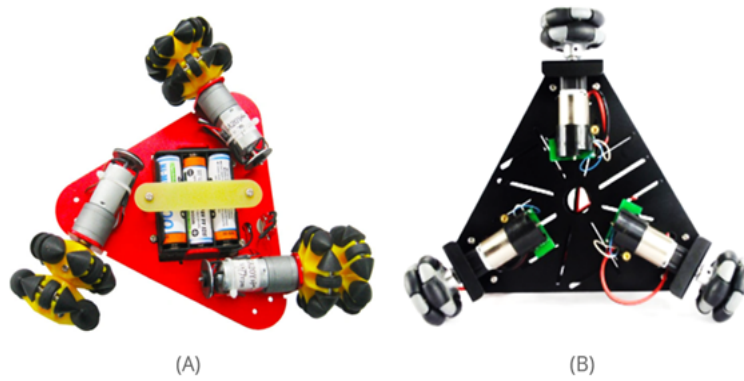


Figura 7: 3WD chasis de robot inteligente Omni Wheel(A),Chassis de Robot Triangular Omni-Direccional 3WD (B)

Con las configuraciones axiales podemos tener chasis de formas variadas, cuadro, circulo o poligonal, de tres a cuatro ruedas las más comunes. La condición que deben de cumplir es que las ruedas queden alineadas y los ejes deben cortar en un único punto como se muestra en la Figura 8.

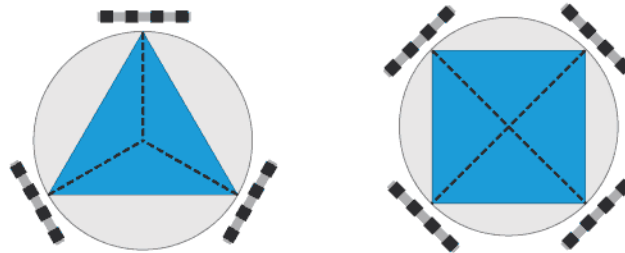


Figura 8: 3WD chasis de robot inteligente Omni Wheel(A),Chassis de Robot Triangular Omni-Direccional 3WD (B)

Fuente:(Llamas, 2018)

Los vehículos típicos basados en ruedas Mecanum tienen una configuración cuadrada o rectangular, con dos ruedas a cada lado del chasis. El uso de cuatro de estas ruedas proporciona omnidireccional. La capacidad del robot depende de que cada rueda descansa firmemente en el suelo, algunas son equipadas con sistemas de suspensión.(Ioan Doroftei y Spinu, 2007)

La posición de las ruedas es muy importante para que el robot pueda ir en todas las direcciones y para ello establece la siguiente configuración: Si tomamos como referencia una vista superior debemos tener presente que los rodillos de las ruedas queden en Angulo respecto al centro del robot y las direcciones de cada rueda deben formar una X con su rueda esquinera correspondiente, tal y como se muestra en la Figura 9,(B). Y si vemos desde la parte inferior,

las direcciones de los rodillos no están en sentido al centro del robot, se podría decir que sus extremos forman un triángulo respecto a la cara lateral donde están montados los motores.

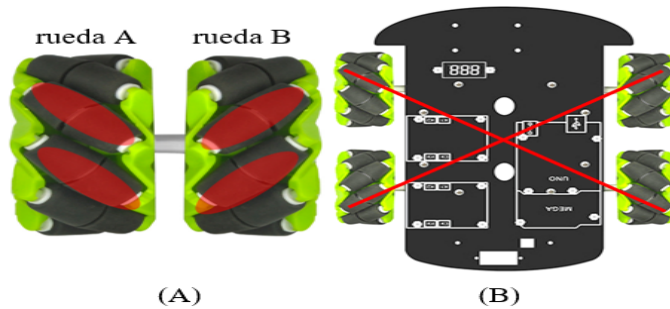


Figura 9: Ruedas mecanum (A), Disposición de las ruedas, vista superior de chasis osoyo Modelo 2019016600 (B)

Fuente:(osoyo, 2019)

1.4. Movimiento de robots con ruedas mecanum.

El procedimiento que se describe en la Figura 10, muestra movimiento y giro de las ruedas visto desde arriba del robot que para el tema de estudio resultada valido. En esta configuración el robot es capaz de traducir en cualquier dirección, hacia adelante / hacia atrás, pero también hacia los lados izquierda/derecha, y girando en el lugar, debido a que se emplean ruedas mecanum. Este tipo de robots son de gran utilidad cuando se tiene que maniobrar en un entorno controlado y estrecho, como una fábrica.

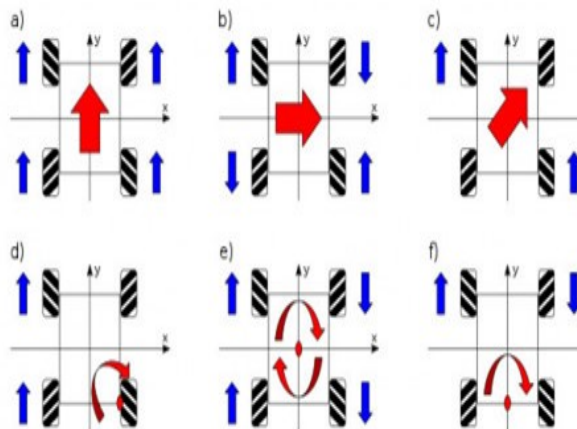


Figura 10: Movimiento del robot según la dirección y velocidad angular de las ruedas, configuración 1.

Fuente:(bricoGeek, 2020)

Se puede obtener otra configuración solo con rotar la vista, es decir: la superior pasa a ser la inferior y la inferior pasa a ser la superior como se muestra en la Figura 11, en esta disposición las ruedas cambian la dirección de su ángulo respecto al centro y para moverse en el mismo sentido que la Figura 10, el sentido de giro de las ruedas cambian. Los robots con esta configuración son más propensos a giros bruscos por lo que sus aplicaciones están dirigidas a ambientes recreativos.

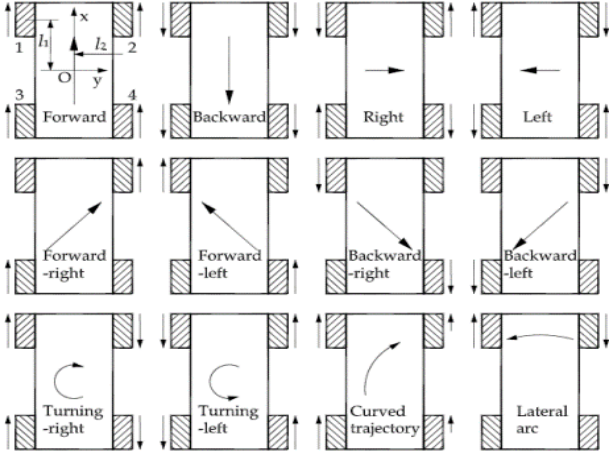


Figura 11: Movimiento del robot según la dirección y velocidad angular de las ruedas, configuración 2.

Fuente:(Ioan Doroftei y Spinu, 2007)

Entonces tanto la configuración de la Figura 10 y la Figura 11, son válidas para la solución real de la orientación de sus rodillos, en ambos casos el proceso de control es diferente.

2. CAPÍTULO 2: DISEÑO Y FABRICACIÓN DEL PROTOTIPO DEL ROBOT OMNIDIRECCIONAL

2.1. Descripción general del robot

El prototipo del robot omnidireccional se divide en tres partes principales base inferior, paredes laterales y tapa superior. Los módulos adicionales como el módulo del efector final y módulo led se consideran como partes independientes a la estructura principal. Las dimensiones generales del prototipo del robot sin módulos son 29cm de largo, 33 cm de ancho y 13 cm de alto aproximadamente.

En la fabricación de las piezas del prototipo se utilizaron las impresoras 3d Prusa Mini y Prusa MK3S con filamentos PLA de color azul, negro y gris. Para el prototipo también se diseñó una tarjeta electrónica que contendrá todos los módulos para el funcionamiento, en la Figura 12 se presenta una vista general del prototipo del robot ensamblado junto con los dos módulos.

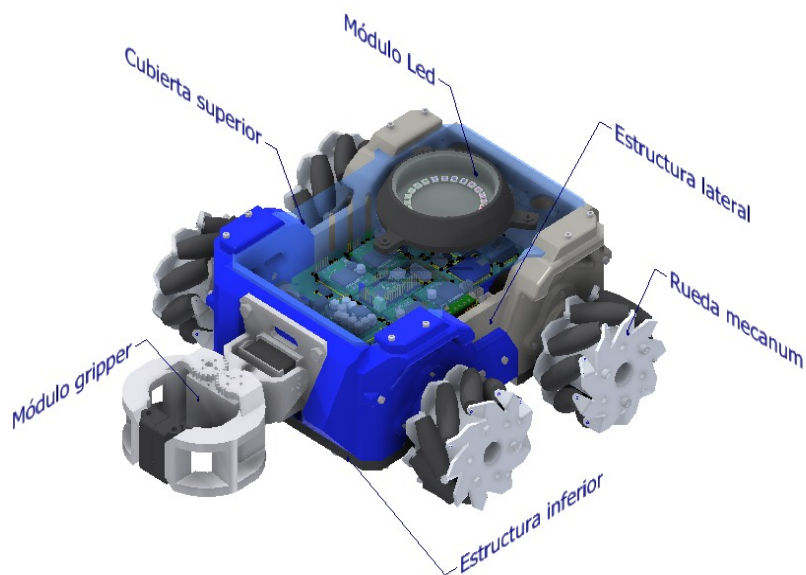


Figura 12: Vista general de las partes del prototipo del robot omnidireccional

2.2. Diseño de la estructura inferior

La estructura inferior Figura 13, se diseñó con el propósito de dar una base sólida para las paredes laterales, la tarjeta electrónica y batería del robot. La fijación de las piezas se

utilizan pernos M6 y M4. Las partes que componen la estructura se detallan en la Tabla 2.

Cantidad	Nombre de pieza	Descripción
2	base de paredes	sujeción de paredes laterales
1	unión central	unión de las bases inferiores
2	unión auxiliar	unión de las bases inferiores y espacio para la batería
1	soporte A	soporte para tarjeta electrónica
1	soporte B	soporte para tarjeta electrónica

Tabla 2: Piezas necesarias para constituir la estructura base del robot.

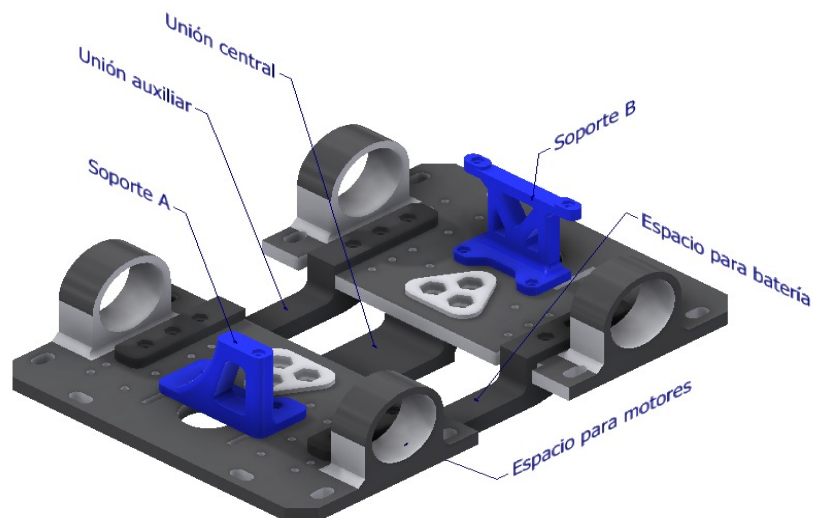


Figura 13: Estructura inferior (soporte principal).

Los soportes A y B fueron diseñados siguiendo el modelo final de la tarjeta electrónica fabricada. El diseño del soporte A permite conectar un cable USB al Arduino mega sin la necesidad de desmontar los soportes de la estructura inferior.

2.2.1. Uniones centrales

Se diseñaron las uniones centrales para aumentar el largo general del robot, debido al límite del área de impresión en la impresora 3D utilizada. Además, en estas piezas tienen un espacio dedicado para una batería de dimensiones máximas de 140mm x48mm.

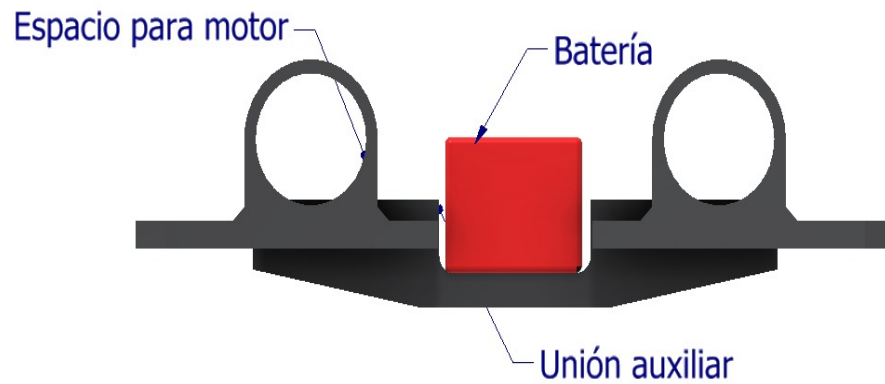


Figura 14: Vista del espacio disponible para la batería en la estructura inferior.

2.3. Diseño de las paredes del chasis

El prototipo tiene dos tipos de paredes las cuales son paredes frontales y paredes de motor. El frente del robot se diferencia por el color de las piezas y lo podemos observar en la Figura 15, siendo el color azul para el frente y el color gris para la parte posterior.

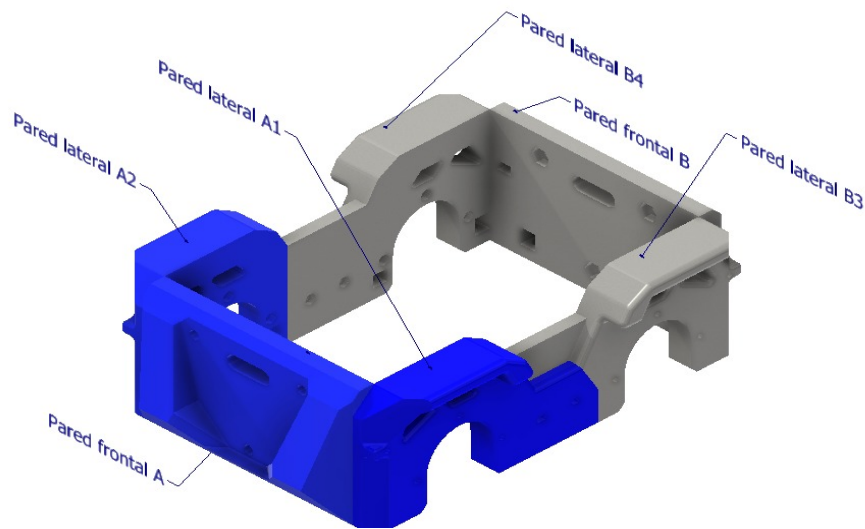


Figura 15: Paredes del robot.

Las paredes frontales tienen en el centro una ranura para incorporar módulos al robot utilizando tres pernos M6; además se tiene un espacio en la parte superior de la ranura para pasar los cables de los módulos hacia la tarjeta electrónica. En los laterales se realizó un cuerpo sólido a modo de guardachoques para proteger a las ruedas de posibles impactos frontales,

cabe recalcar que su firmeza depende también de dos uniones con las paredes laterales por medio de pernos M3.

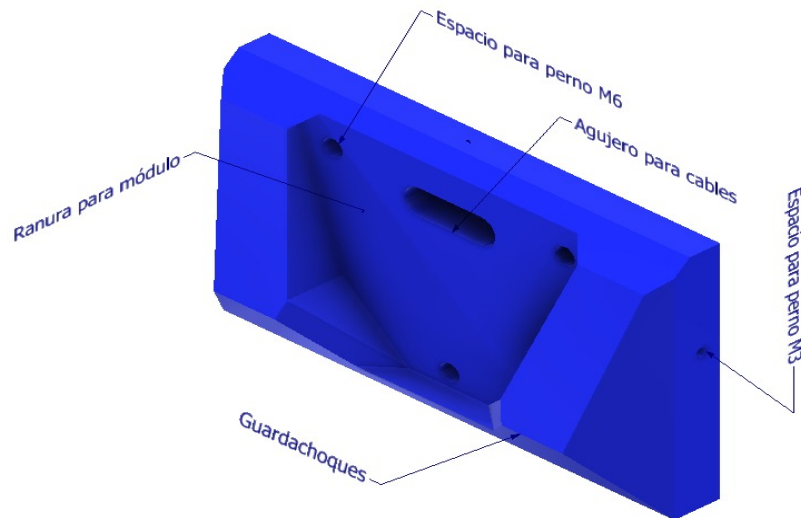


Figura 16: Pared frontal del robot.

En las paredes laterales se sujetan los motores a través de 3 pernos M3. La pared lateral está compuesta por la pared lateral A que pertenece al frente del robot, la pared lateral B que pertenece a la parte posterior del robot y los sujetadores de los motores. Las dos paredes se unen en la mitad por medio de dos pernos M3, este diseño permite diferenciar el frente del robot. Figura 17.

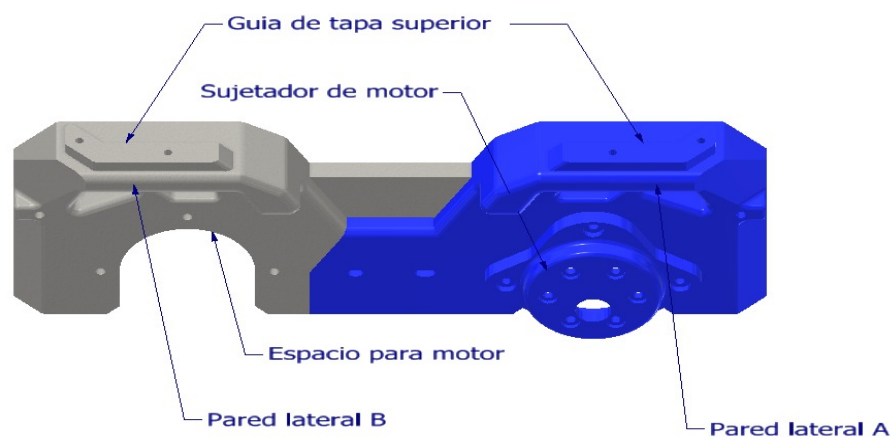


Figura 17: Paredes laterales modulares.

El sujetador del motor está diseñado para motores de 37 mm de diámetro con eje no

céntrico. Esta pieza se fija a la pared a través de tres pernos M3.

2.4. Diseño de la cubierta del chasis

La tapa superior está compuesta en por una plancha de acrílico con agujeros para los cables del módulo led y para colocar un pulsante de encendido y apagado. Este acrílico se fija sobre las paredes por medio de cuatro sujetadores con dos pernos M3 cada uno. En la parte superior se tiene guías cilíndricas para colocar el módulo led.

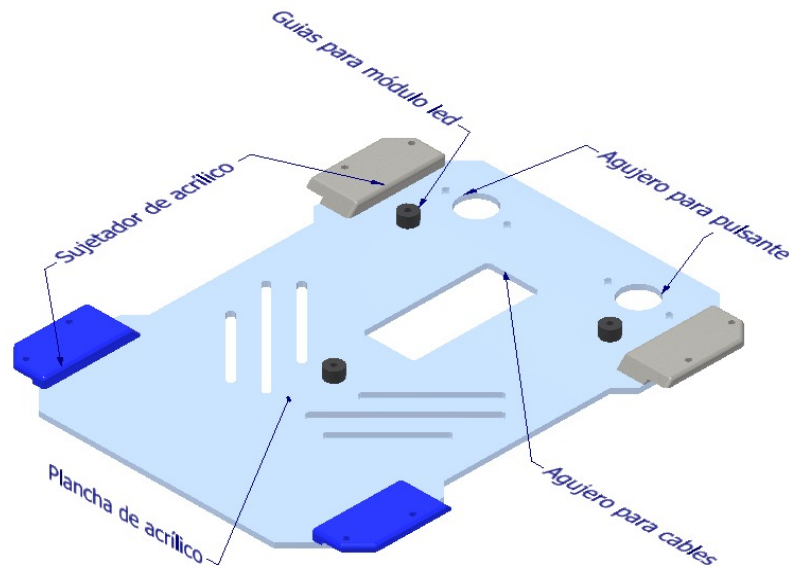


Figura 18: Tapa de acrílico para protección de componentes y sujeción de módulo led.

2.5. Diseño de módulo del efector final

Teniendo presente las medidas finales del cuerpo del robot se propuso emplear servos de doble eje mismo que por el tamaño podían ir internos en la estructura. Las posibles soluciones para cierre de garra fueron: 1) Empleando la fuerza de dos servos para cerrar la garra. 2) Emplear la fuerza de un servo y cerrar la garra por medio de una palanca conectada a un eje giratorio. 3) Emplear una cremallera que transforme el movimiento rotacional en lineal y de esa forma provocar el cierre de la garra. 4) Emplear el movimiento rotacional y transmitir la fuerza por engranes conectados a la garra.

Se vio factible la cuarta propuesta y finalmente se optó por emplear un solo servo y transmitir la fuerza rotacional a través de engranes para provocar la apertura y cierre de la

garra, para ello se emplearon las fórmulas de la Tabla 3, y sus variables se describen en la Figura 19.

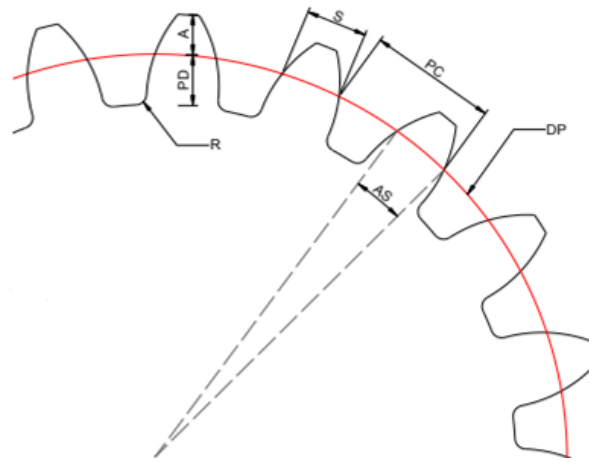


Figura 19: Variables descritas para diseñar un engranaje.

Fuente:(CADistas, 2020)

Descripción	Formula	Resultados obtenidos
Diámetro primitivo (DP)	$DP=ZM$	27,6 mm
Paso circular (PC)	$PC=M$	3,76 mm
Cabeza del diente (A)	$A=M$	1,2 mm
Espesor del diente (S)	$S=PC(19/40)$	1,78 mm
Pie del diente (PD)	$PD=1.25 M$	1,5 mm
Radio de entalle (R)	$R=PC/12$	0,31 mm
Ángulo de espesor (AS)	$AS=(360 S)/(2(DP/2))$	7,39 mm
Módulo (M)		1,2 mm
Número de dientes (Z)		23

Tabla 3: Fórmulas para generar un engranaje con Autodesk inventor.

Fuente:(CADistas, 2020)

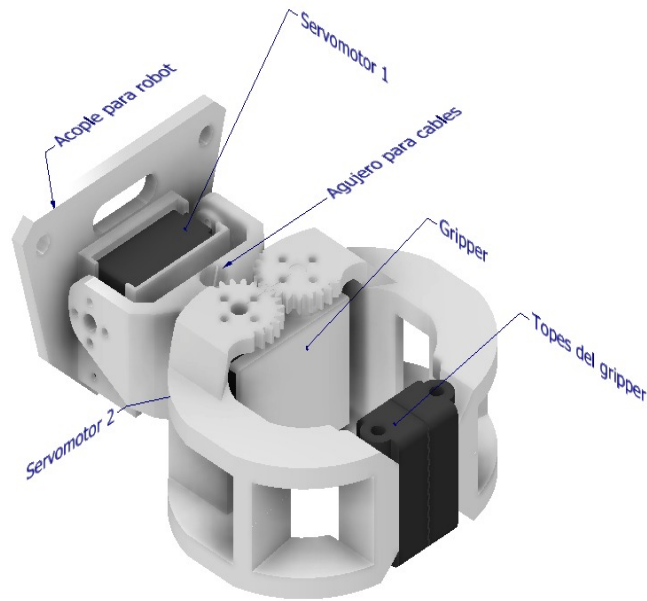


Figura 20: Módulo efector final (gripper).

2.6. Diseño de módulo led

El módulo led este compuesto por un anillo de 24 leds de tipo neopixel y una cubierta con forma de cono truncado, que la parte superior tiene una tapa de circular de acrílico blanco para reflejar de mejor manera los colores RGB de los leds. en la parte inferior del módulo se tiene guías para pasar los cables de los leds y los espacios para los pernos.

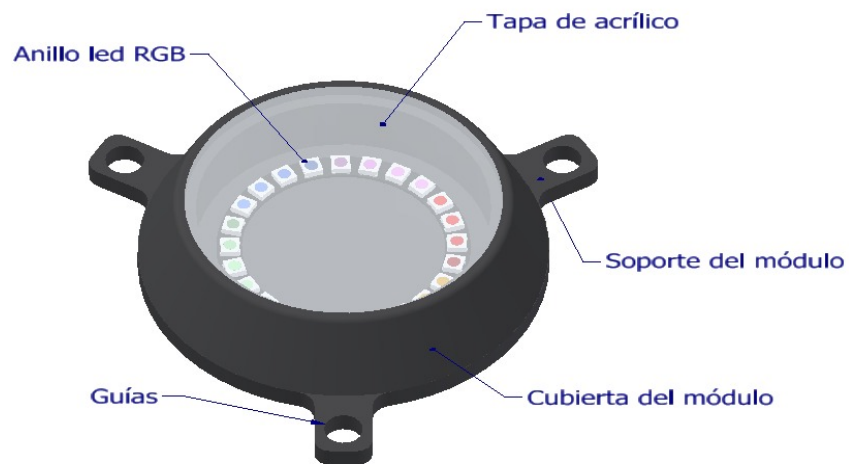


Figura 21: Piezas que conforman el módulo led.

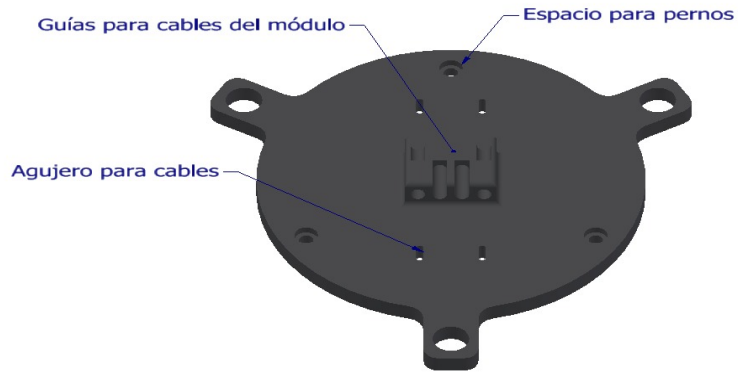


Figura 22: Soporte de módulo led.

2.7. Diseño y fabricación de la placa electrónica

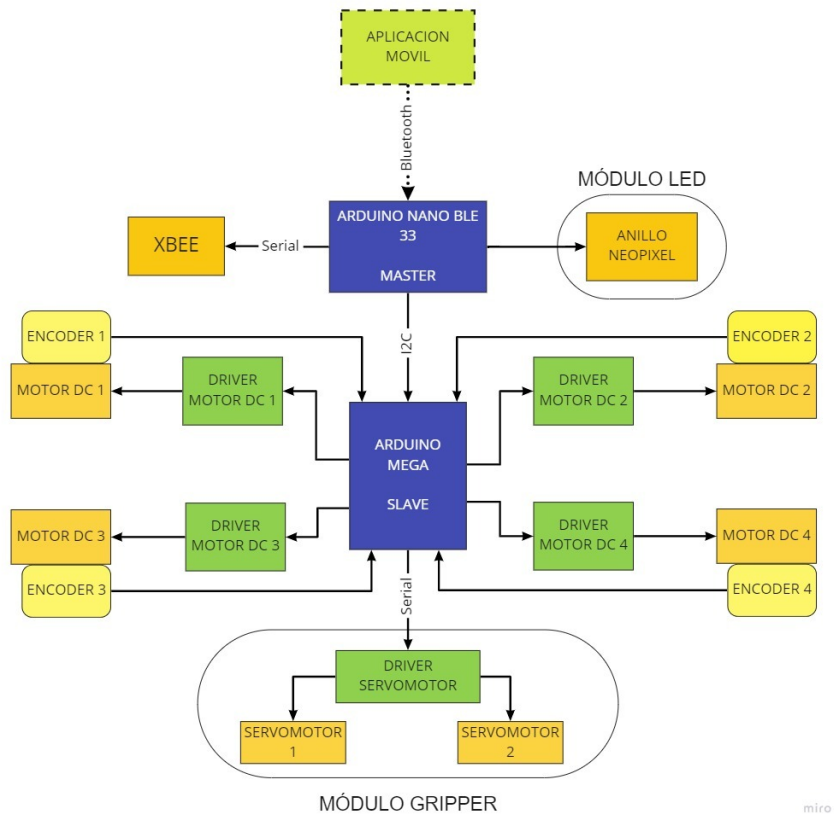


Figura 23: Diagrama de bloques del funcionamiento del robot

El diseño de la placa se basó en la tecnología "plug and play" puesto que debía complementar el objetivo del robot. La operación de los movimientos y el uso de los módulos del robot se los realiza a distancia por medio de una aplicación móvil conectada por bluetooth, por ello se emplea un Arduino nano BLE el cual cuenta con este aditamento.

La desventaja presente con esta tarjeta es su número de pines disponibles y ante ello se optó por utilizar un Arduino MEGA 2560 el cual cuenta con 54 pines distribuidos entre entradas y salidas, esto frente a 14 disponibles con los que cuenta el Arduino nano. Otra ventaja a parte de tener la comunicación bluetooth, el Arduino nano nos da la opción que todos sus pines pueden ser usados para interrupciones, entonces; se tenían dos controladores con distintas prestaciones que se podían aprovechar al máximo.

La comunicación entre los dos Arduinos se realiza por comunicación serial i2C o interfaz de dos hilos donde se emplea un bus de datos de dos conexiones; (SDA) serial data y (SCL) serial clock. Este protocolo nos permite comunicar a velocidades estándar de 100Khz, 400Khz y 1 Mhz de dos modos diferentes:

- Maestro y Esclavo donde el maestro transmite y el esclavo recibe, este modo se usa cuando se desea configurar un registro del esclavo I2C.
- Esclavo y maestro donde el maestro recibe y el esclavo transmite, este modo se usa cuando queremos leer información de sensores I2C.

Con los controladores comunicados se designaron las entradas y salidas que se emplearían para el movimiento. El ATMEGA conecta a los drivers de los motores y al mini maestro para servomotores, mientras que el Arduino nano conecta al módulo led. Los pines sobrantes quedan a disposición de un uso futuro en mejoraras o adición de módulos del robot al igual que el módulo de comunicación Xbee.

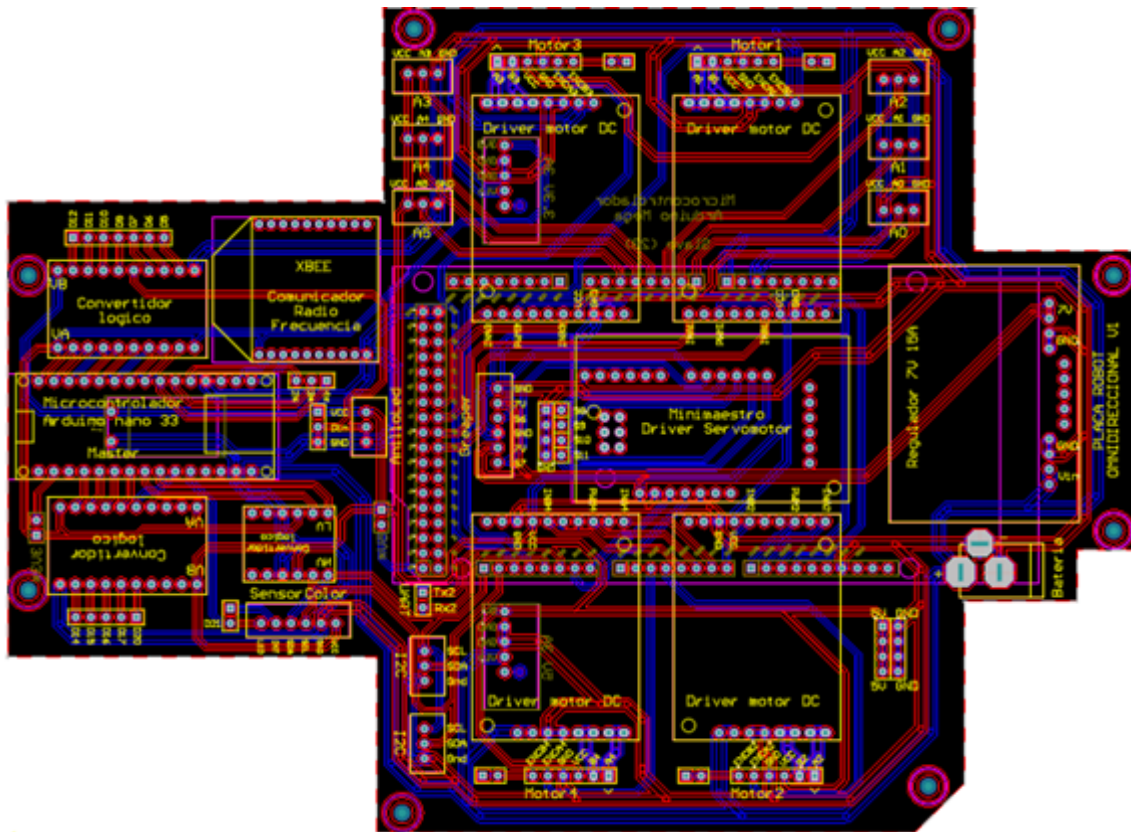


Figura 24: Vista 2D footprint tarjeta electrónica.

Adicional se emplearon peinetas para enchufar los distintos módulos y conectores molex para los motores, servos motores y módulo led, Figura 28. La alimentación se realiza con una batería de 11.1V 6200mAh conectada por medio de un Jack DC hembra 5.5 x 2.1 alimentación de Arduino. Las pistas se rutearon de forma manual cuidando que el flujo de corriente sea fluido y evitando pistas bruscas que puedan provocar cortos circuitos a futuro. La placa final se muestra en la Figura 25 y las pistas de la capa superior e inferior se muestran en la Figura 24, cuenta con dos caras de impresión y respectiva señalización de la ubicación de los componentes, así como también los agujeros para sujetar la placa al chasis del robot.

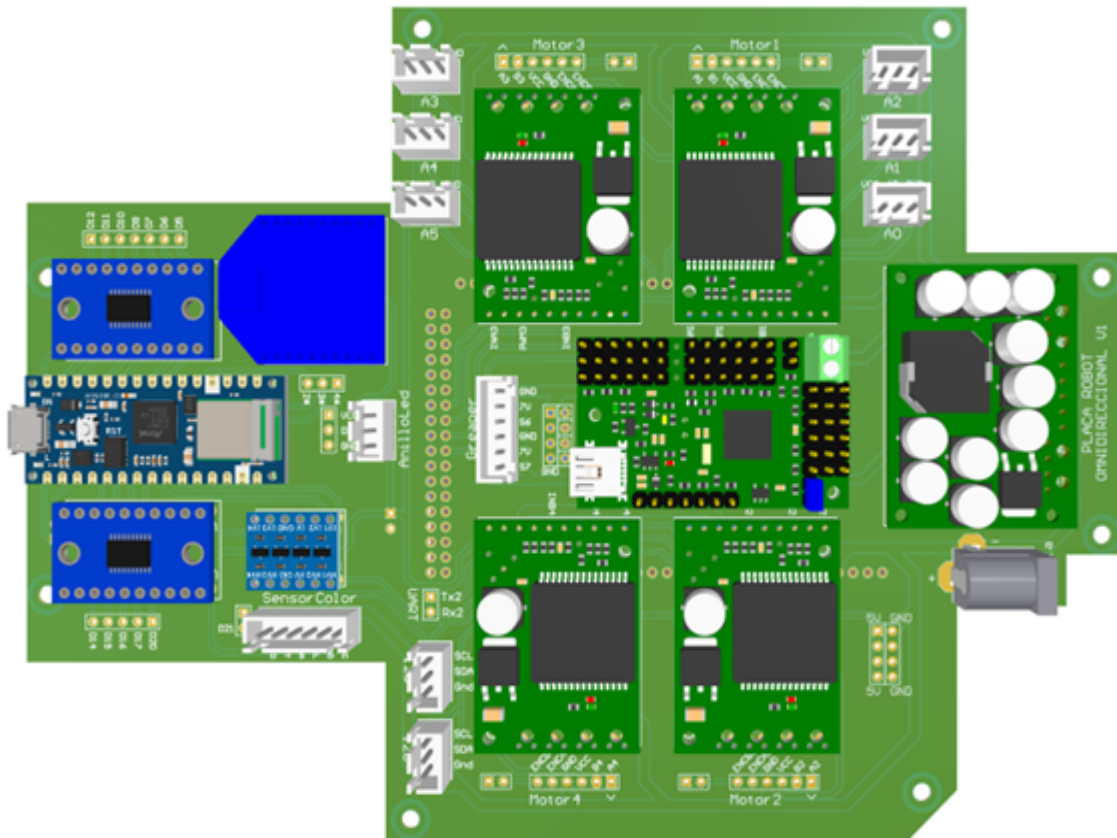


Figura 25: Vista 3D superior tarjeta electrónica.

Los módulos empleados se describen en la Tabla 28 (Anexos), de entre ellos un regulador de 7V 15A para alimentar al mini maestro, un regulador de 5V 3A y un regulador de 3.3V 3A para uso futuro, dos convertidores lógicos de 3V a 5V para las señales de entrada y salida de los pines del microcontrolador BLE, un conversor lógico bidireccional de 3V a 5V para comunicación i2C y cuatro drivers que operan de 5V a 24V con corrientes de entre 12A a 30A máximo. La disposición final de los componentes en la placa se puede observar en la Figura 25 y Figura 26, vistas superior e inferior respectivamente.

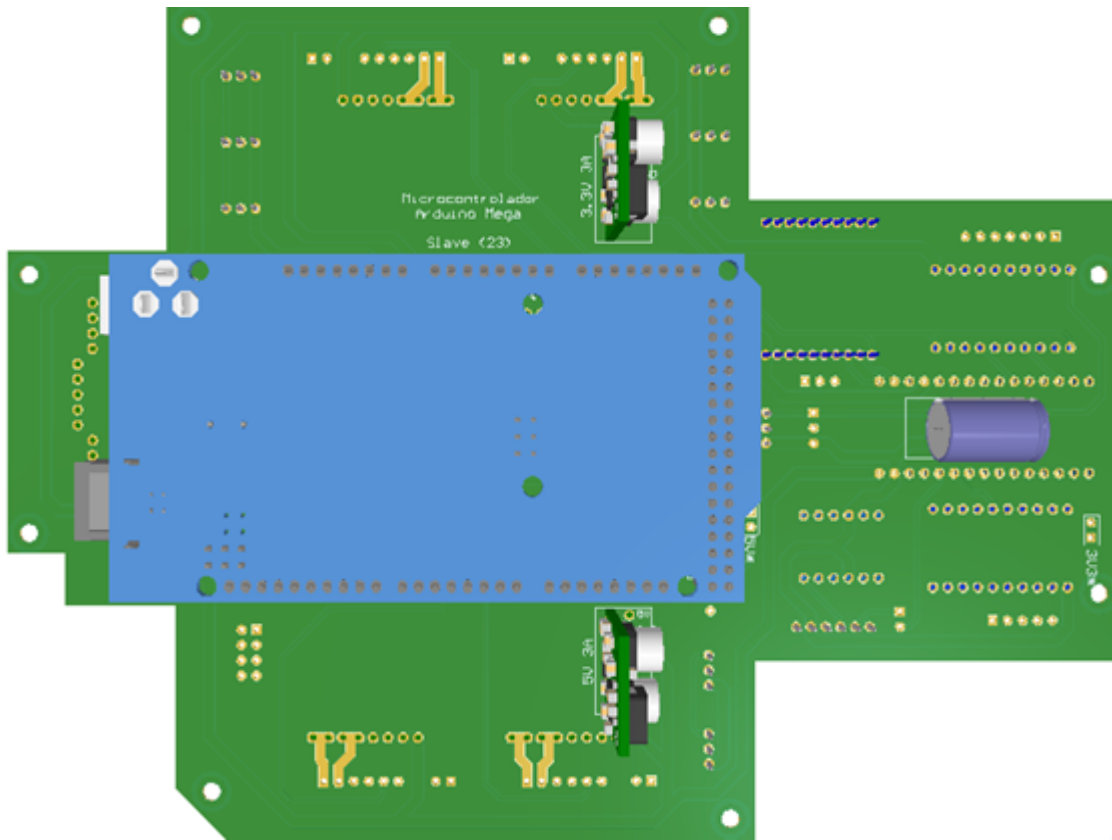


Figura 26: Vista 3D inferior tarjeta electrónica.

2.7.1. Reglas de enrutamiento

Por las pistas de cobre circulan diferentes voltajes y corrientes por lo que es necesario delimitar las dimensiones que tendrán según su función. Las reglas aplicadas se describen en la Tabla 4.

Pista	Ancho mínimo	Ancho preferido	Ancho Máximo
7V 15A	1	1.2	1.5
5V	0.35	0.6	1
Batería	1	1.5	3
Gnd	1	1.5	3
Otros	0.35	0.35	1.6

Tabla 4: Valores de ruteo (mm).

También es necesario designar el diámetro máximo y mínimo que tendrán los agujeros revestidos de cobre para los pines de los distintos módulos, entando entre 1.3mm de diámetro externo y 0.7mm de diámetro interno. Las reglas se aplican respetando el valor mínimo que puede tener los agujeros y las pistas, véase en la Figura 27.

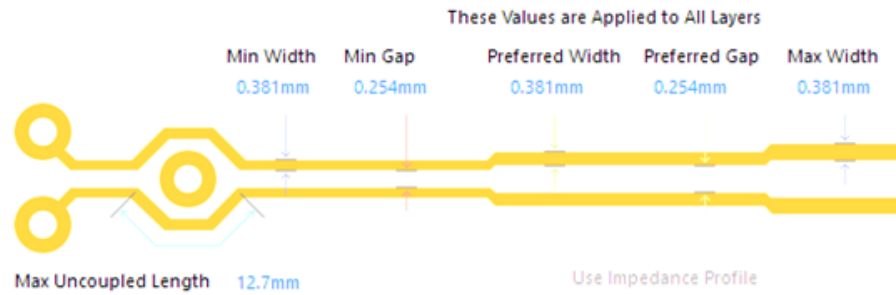


Figura 27: Valores minimos que puede tener una pista.

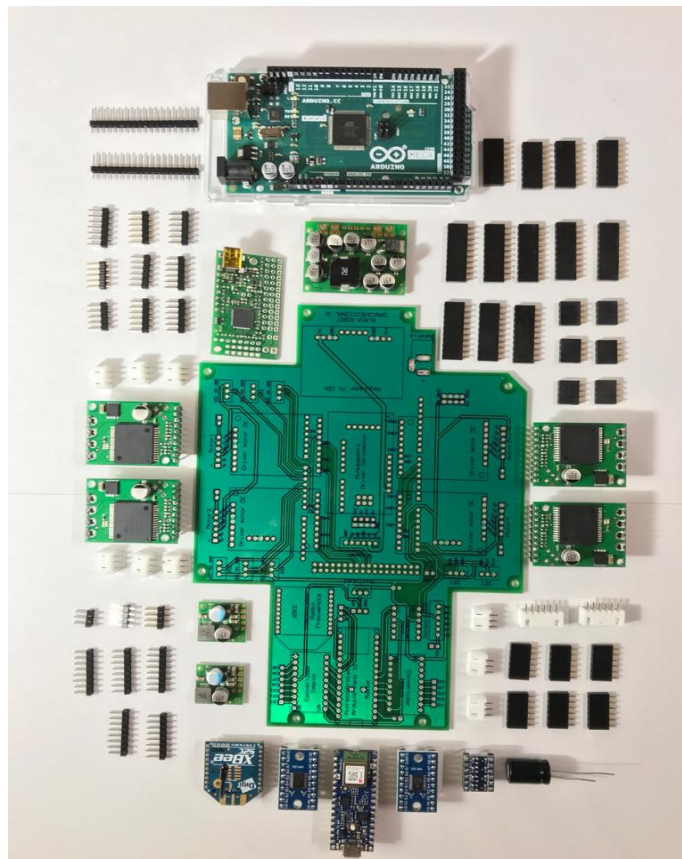


Figura 28: Tarjeta impresa, módulos electrónicos, conectores molex y peinetas.

2.8. Fabricación de las piezas del cuerpo y módulos del robot

Al término de la etapa de diseño se inspecciono todas las piezas en busca de algún defecto para su modificación. Previamente en este capítulo se menciona las impresoras utilizadas, una impresora grande con un volumen de impresión de 25x21x21 cm Figura 29(B) y una impresora pequeña con un volumen de impresión de 18x18x18 cm Figura 29(A).

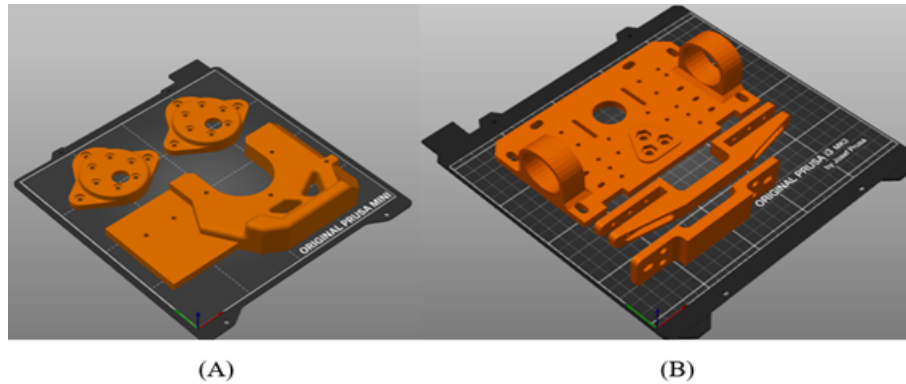


Figura 29: Base de impresión Prusa Mini (A), base de impresión MK3S (B).

La configuración con las que se trabajó para el prototipo se describen en la Figura 30, y el material empleado para la fabricación del cuerpo y módulos del robot están dados por; un 87 % es impresión en filamento PLA, un 3 % es impresión en filamento PETG y un 10 % en cortes y grabado de acrílico blanco y transparente en 3mm de espesor. Si sumamos el tiempo de impresión de las dos impresoras tenemos un total de 105 horas y 25 minutos de trabajo de impresión, cabe recalcar que se utilizaron ambas impresoras al mismo tiempo para imprimir piezas de manera simultanea.

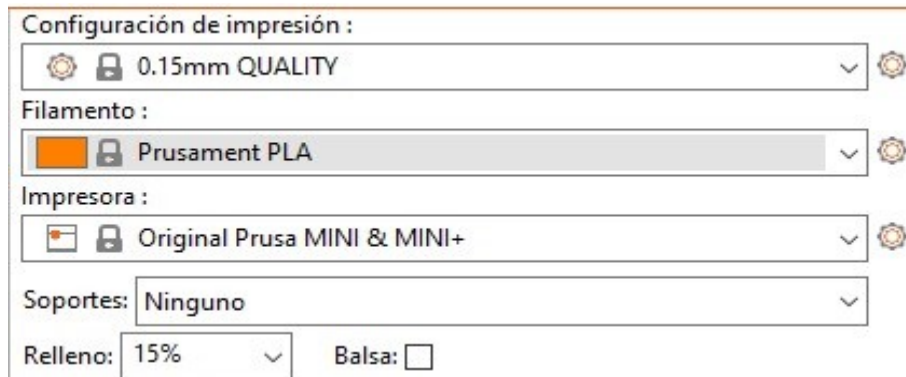


Figura 30: Parámetros para la impresión.

En algunos casos fueron necesario configurar para imprimir con soportes, debido a la forma

con las que fueron diseñadas las piezas, tal fue el caso de las piezas que iban a conformar el módulo del efector final ya que las garras fueron hechas a mitades para su posterior impresión poderlas unir con pegamento industrial. El resultado de imprimir con soportes se muestra en la Figura 31.

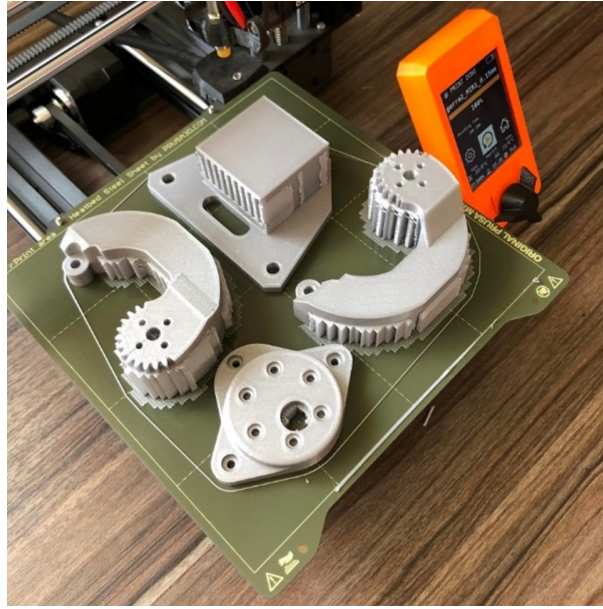


Figura 31: Piezas impresas con soportes.

3. CAPÍTULO 3: PROGRAMACIÓN

3.1. Programación del control con comunicación bluetooth low energy (BLE)

En el control remoto del robot omnidireccional se utilizó el Bluetooth 5.0 incorporado en el Arduino BLE (Bluetooth Low Energy). Este protocolo de comunicación de bluetooth se diferencia del bluetooth clásico porque transmite varios datos de información de un tamaño reducido de bits utilizando un bajo consumo de energía. El Bluetooth LE organiza los datos en servicios y características, el servicio contiene una o varias características y una característica contiene el valor del dato. En la transmisión se utiliza un dispositivo central, para consultar servicios y modificar características; y un dispositivo periférico que obtiene los valores para las características y también recibe las características modificadas.

El control se realizó con una aplicación móvil para Android utilizando el software libre de MIT AppInventor. Esta plataforma permite realizar aplicaciones con una interfaz sencilla y con programación por bloques. Además, se puede agregar extensiones para soportar protocolos de comunicación como serial de Arduino, Bluetooth y Bluetooth LE.

3.1.1. Diseño de la interfaz de la app

La interfaz de la aplicación se diseñó utilizando los elementos de la interfaz de usuario de App Inventor como botones, imágenes, listas, etiquetas y organizadores de espacios. Las propiedades de estos elementos pueden personalizarse como el tamaño, color, texto, imagen.



Figura 32: Herramientas disponibles para crear APP con sistema operativo Android

También se incluyen elementos adicionales como la ventana de notificación y la extensión para el soporte bluetooth LE. Este último es una extensión que no está incluido por defecto en la plataforma por lo que es necesario descargar e importar la extensión al proyecto, en la Figura 32 se muestra la característica agregada.

En la Figura 33, se presenta la interfaz del control remoto del robot instalada en un dispositivo Android. La interfaz tiene la siguiente distribución de botones:

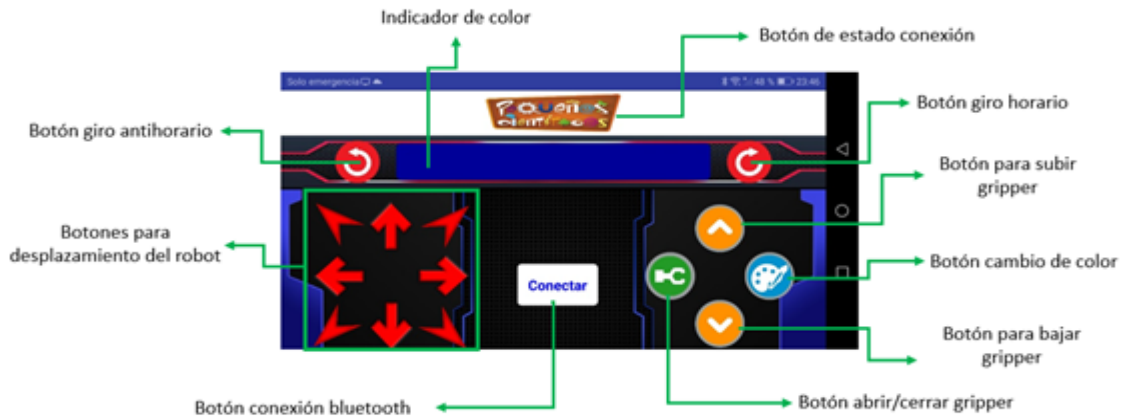


Figura 33: Interfaz del control remoto del robot

- Botón de estado de conexión: para mostrar/ocultar el estado de la conexión como “desconectado”, “conectado al dispositivo”
- Indicador de color: aquí se visualiza el color para mostrar en el módulo led.
- Botón de giro horario: mantener presionado gira el robot en sentido horario.
- Botón de giro antihorario: mantener presionado gira el robot en sentido antihorario.
- Botón para subir gripper: sube el gripper con un determinado ángulo al presionar.
- Botón para bajar gripper: baja el gripper con un determinado ángulo al presionar.
- Botón abrir/cerrar gripper: al presionar el gripper cambia de estado abierto a cerrado y viceversa.
- Botón cambio de color: cambia la selección del color para el módulo led.
- Botón de conexión: inicia la búsqueda de dispositivos bluetooth disponibles.

- Botones de desplazamiento: mantener presionado desplaza al robot en la dirección seleccionada.

Al presionar el botón de conexión la pantalla cambia a la Figura 34.



Figura 34: Búsqueda de dispositivos bluetooth disponibles.

En esta pantalla se muestra una lista de los dispositivos bluetooth detectados. Para conectarse al dispositivo se debe seleccionar directamente en la lista. Si no se encuentra el dispositivo se puede pulsar el “botón cancelar búsqueda” para cancelar la búsqueda y volver a la pantalla del control.

Cuando se establezca la conexión aparece un símbolo de bluetooth en la parte central y el botón texto del botón de conexión cambia de “conectar” a “desconectar”.

3.1.2. Programación del control

Definición de variables globales y valores de inicio.

La programación comienza con la definición de variables globales, en la Figura 35, se presentan las variables con el identificador UUID para el servicio y las características del bluetooth. Para el control del robot se define un servicio y cuatro características referentes al movimiento del robot, el módulo gripper y el módulo led. Este identificador se explicará en la sección de la programación del Arduino nano.



Figura 35: Definición de variables globales.

Cuando la pantalla de la aplicación se abre por primera vez se ejecutarán las funciones de **resetValores** y **mostrarPaneles** para mostrar la pantalla del control tal como se presenta en la Figura 36.

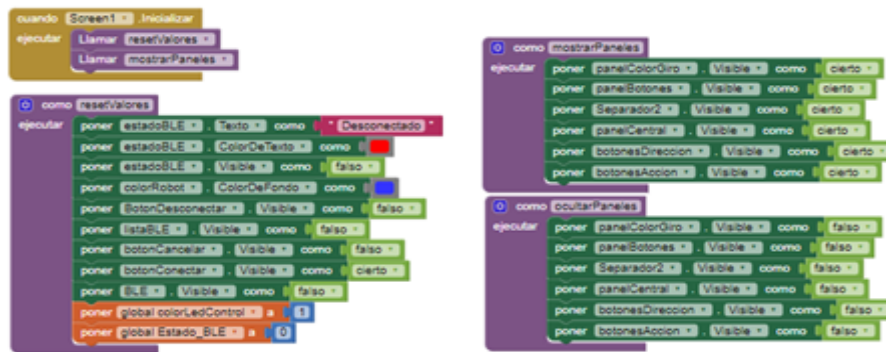


Figura 36: Pantalla de inicio.

La función **resetValores** reinicia la apariencia del control y de los valores globales **Estado_ble** y **colorLedControl**. La función de **mostrarPaneles** se encarga de hacer visible todos los botones del control y la función **ocultarPaneles** oculta todos los botones del control.

Botón de conexión y establecimiento de conexión bluetooth.

La conexión bluetooth se inicia al presionar el botón conectar, el cual inicia la búsqueda de dispositivos bluetooth y cada dispositivo encontrado es colocado en la lista de dispositivos. Cuando se selecciona un dispositivo se llama a la función de conectar, misma que establecerá conexión con el dispositivo seleccionado de la lista y detendrá la búsqueda de dispositivos. Cuando se establece conexión se recupera el nombre del dispositivo conectado y vuelve a mostrar la pantalla del control.



Figura 37: Establecimiento de conexión bluetooth.

Botón cancelar búsqueda.

Cuando se presiona el botón se detiene la búsqueda de dispositivos y se procede a limpiar la lista de dispositivos encontrados. El estado de conexión se muestra como “selección cancelada” y se regresa a la pantalla principal del control.



Figura 38: Cancelar búsqueda.

Botón desconectar y desconexión bluetooth.

Cuando el control está conectado al robot el botón de desconectar es visible y puede presionarse para finalizar la comunicación bluetooth. Cuando se finaliza la conexión el estado de conexión cambia a “desconectado”, se llama a la función **resetValores** y se muestra una notificación en la pantalla indicando que el dispositivo fue desconectado.

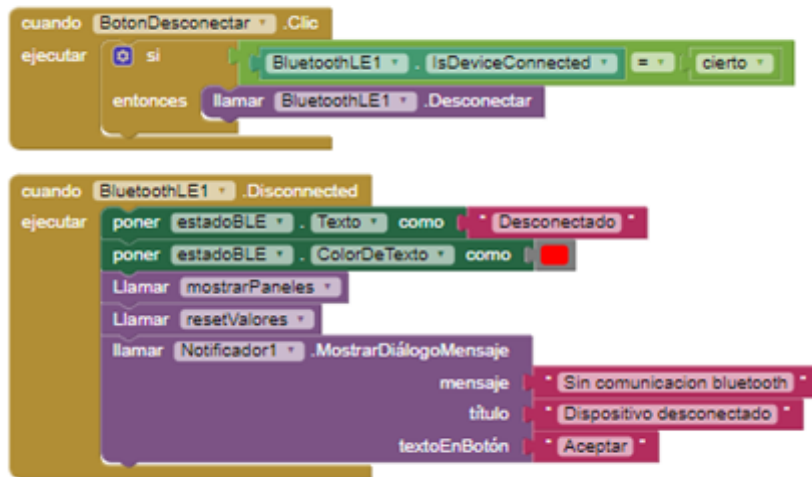


Figura 39: Desconexión bluetooth.

Botones de movimiento.

Los botones de movimiento tienen asignado un carácter definido en Figura 40, para cada uno de los diez movimientos del robot, al presionar uno de los botones el carácter se envía a la función **comandoControl**.



Figura 40: Carácter de movimiento, estado 1.

Cuando se deja de presionar cualquiera de los botones el carácter que se envía a la función es “x”.

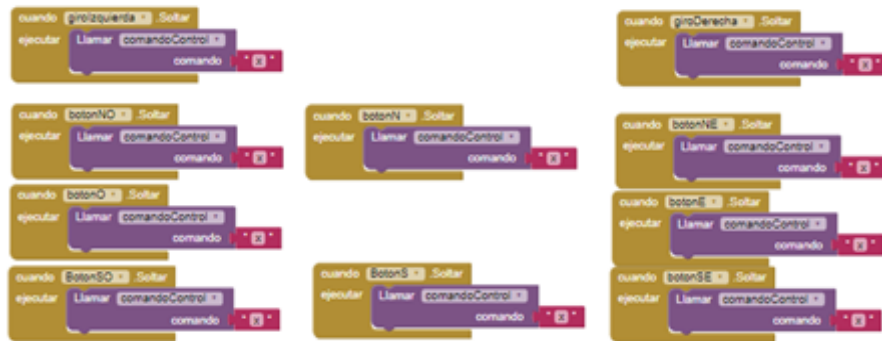


Figura 41: Caracteres de movimiento, estado 2.

Botones del módulo de efector final.

De forma similar a los botones de movimiento los botones para subir, bajar y abrir/cerrar gripper tiene asignado un carácter único. Cuando son presionados se envía el carácter a la función `comandoGripper` y cuando se sueltan se envía el carácter “m”.

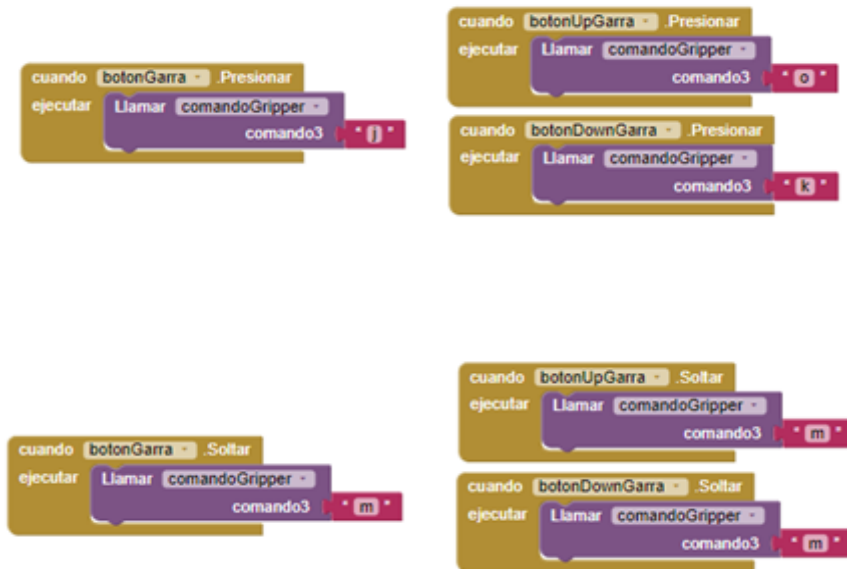


Figura 42: Caracteres de apertura y cierre de módulo gripper.

Botón de estado de conexión.

El estado de la conexión se muestra en pantalla cuando se hace clic sobre el logo de “Pequeñ@s Cientific@s”, si se vuelve a hacer clic el texto se oculta de la pantalla.



Figura 43: Estado de conexión.

Botón de cambio de color para módulo led.

Cuando se presiona el botón de cambio de color se aumenta la variable **colorLedColor** y se llama a la función **tablaColor** que en base al valor de esta variable asigna el color al indicador del control y envía el valor como comando en la función **comandoLedpixel**. Los colores disponibles son azul, rojo y verde.

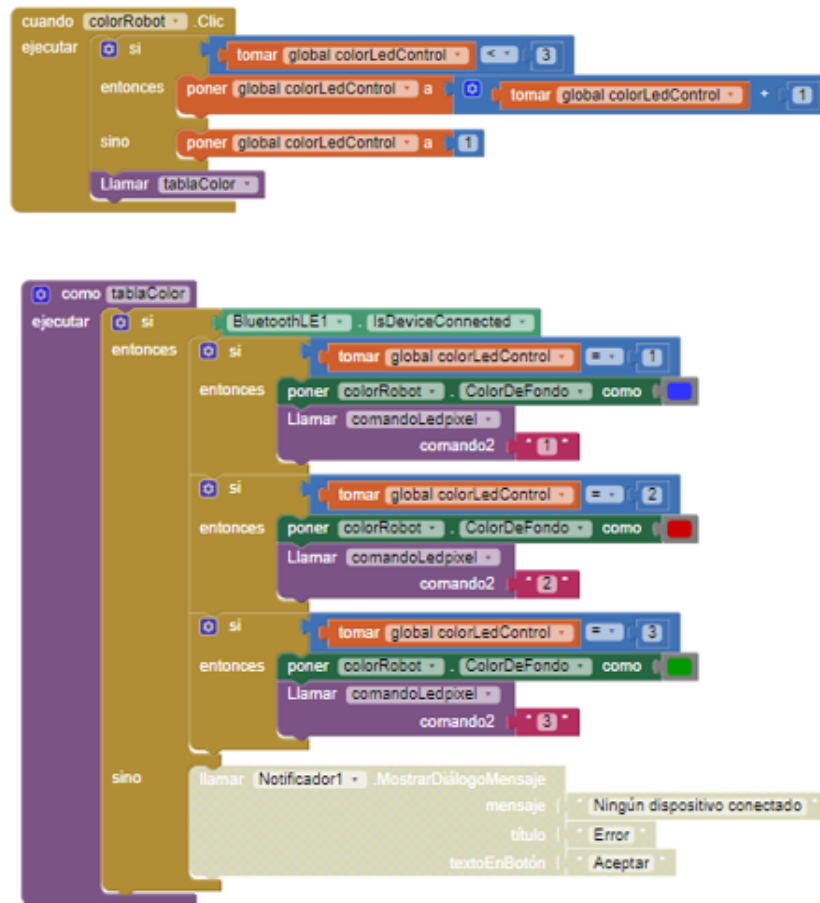


Figura 44: Cambio de color de módulo led.

Funciones para envío de caracteres por bluetooth.

Los comandos se envían utilizando las tres características definidas para el servicio del robot. Estas tres características envían los caracteres para el movimiento del robot, módulo del efector final y módulo led por medio de las funciones de **comandoControl**, **comandoGripper** y **comandoLedpixel**, respectivamente. En el envío de los comandos se utiliza la función de **WriteBytesWithResponse** que utiliza el identificador UUID del servicio y característica correspondiente para enviar el valor por bluetooth.

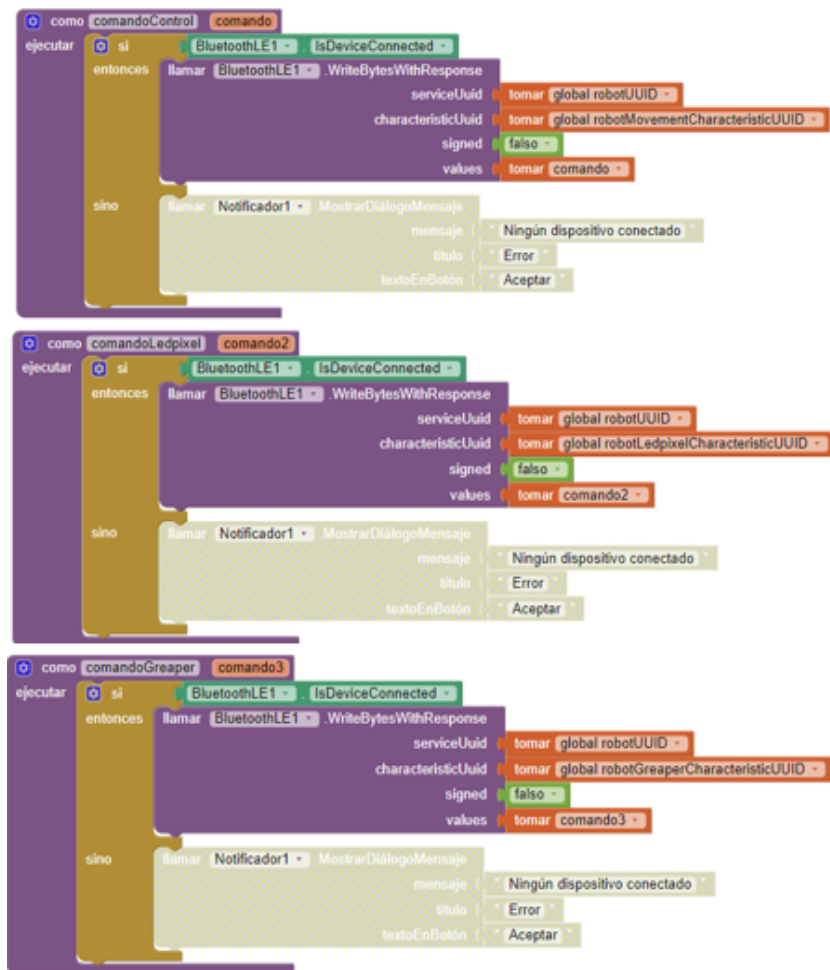


Figura 45: Envío de caracteres por bluetooth.

3.2. Cinemática del robot omnidireccional

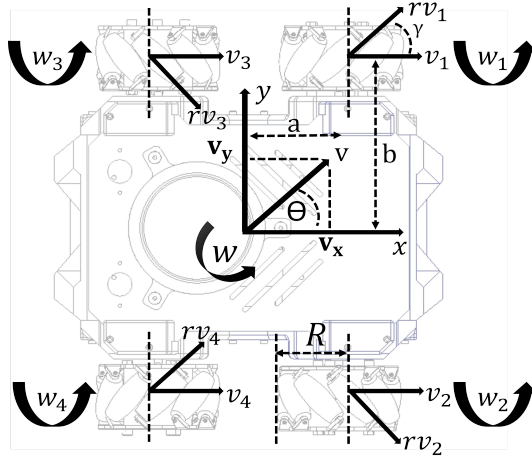


Figura 46: Diagrama de cuerpo de robot omnidireccional

w : Velocidad angular global

vx : Velocidad lineal en x

vy : Velocidad lineal en y

r : Radio de los rodillos

a : Distancia del centro del robot al eje de la rueda en dirección x

b : Distancia del centro del robot al eje de la rueda en dirección y

γ : Angulo gamma, cuyo valor es constante

$$vx = v \cos \Theta \quad (1)$$

$$vy = v \sin \Theta \quad (2)$$

$$ai = \{a, a, -a, -a\}$$

$$bi = \{b, b, -b, -b\}$$

$$\gamma i = \left\{ \frac{\pi}{4}, -\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4} \right\}$$

$$vi + rvi \cos(\gamma i) = vx - bi w \quad (3)$$

$$rvi \sin(\gamma i) = vy + ai w \quad (4)$$

Debemos despejar las velocidades lineales en x de la ecuación 3 y las componentes de

velocidades en y de la ecuación 4

$$v_i = v_x - b_i w - r v_i \cos(\gamma_i)$$

$$r v_i = \frac{v_y + a_i w}{\sin(\gamma_i)}$$

Remplazando la ecuación 4 en la ecuación 3 obtenemos la ecuación 5

$$v_i = v_x - b_i w - \frac{v_y + a_i w}{\tan(\gamma_i)} \quad (5)$$

Remplazando valores de $\tan(\gamma_i)$ con $\gamma_i = \left\{ \frac{\pi}{4}, -\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4} \right\}$ y también "i":

$$v_1 = v_x - v_y - (a + b) w$$

$$v_2 = v_x + v_y + (a + b) w$$

$$v_3 = v_x + v_y - (a + b) w$$

$$v_4 = v_x - v_y - (a + b) w$$

Expresado en forma matricial:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix} \quad (6)$$

Por último, expresado las velocidades en termino de velocidades angulares $v_i = w_i R$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix} \quad (7)$$

3.3. Identificación de la planta

Se emplean cuatros motores 37Dx68L de la marca Pololu con Motorreductor cepillado metálico 30:1, y un codificador CPR 64 (piñón helicoidal) de cuadratura integrado que

proporciona una resolución de 64 recuentos por revolución del eje del motor, lo que corresponde a 1920 recuentos por revolución del eje de salida de la caja de engranajes. Funciona a 12V y puede llegar a proveer un torque de 14 kgcm (190 ozin) con un consumo de 5.5A. La obtención de datos del comportamiento del motor ante una entrada escalón se la realizo con el serial de Arduino, con un total de 70 datos en unidades RPM con un tiempo de muestreo de 0.047s.

Para poder observar el comportamiento del motor ante la entra escalón y obtener su modelo matemático para posterior implementar un PID, fue necesario normalizar los datos y graficar en el software Matlab. Figura 47.

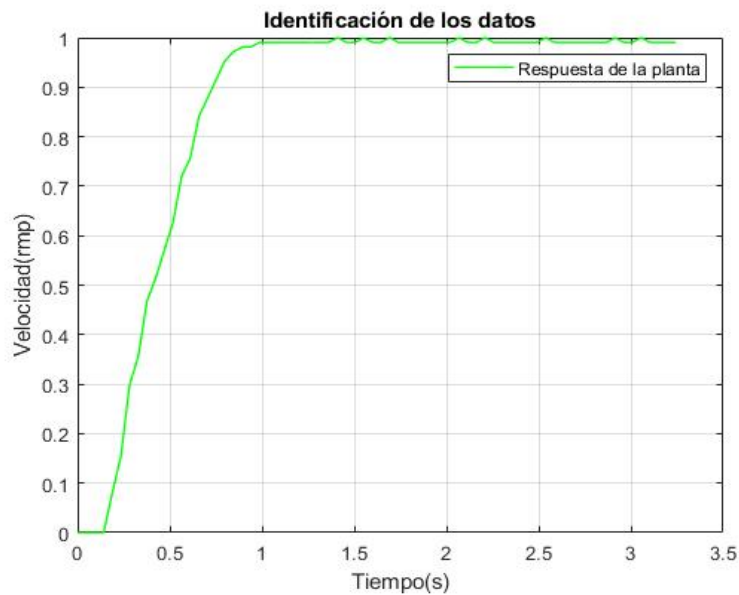


Figura 47: Respuesta ante el escalón del motor DC.

Según la Figura 47, podemos inferir que el sistema puede ser interpretado como uno de primer orden con retraso, por ello se puede emplear la ecuación 8 como función de transferencia.

$$G(s) = \frac{K}{\alpha s + 1} e^{-\Gamma s} \quad (8)$$

Donde K es una ganancia expresada como el valor final al que tiende la variable de salida. La constante tiempo α , es el tiempo que le toma al sistema en llegar al 63.2% del estado estable.

Mientras que Γ es el tiempo que le toma al sistema en reaccionar.

Se empleo un método alternativo con el cual se obtiene el mismo resultado, con ayuda de la herramienta PID tuner de Matlab se pueden leer los datos obtenidos del mapeo de la señal

y obtener una aproximación de la función de transferencia del motor. En la Figura 48, se puede observar que las curvas tienen una similitud del 91.36 % según lo entregado por Matlab y la ecuación 9 es la función de transferencia obtenida.

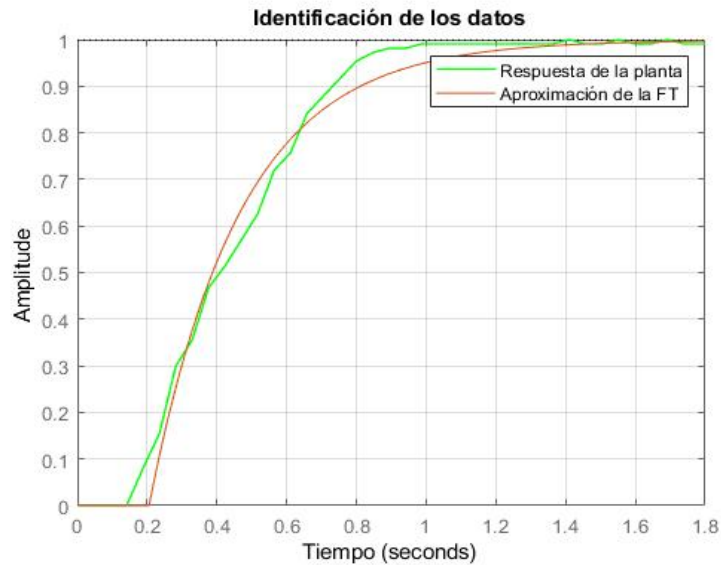


Figura 48: Grafica de la respuesta al escalón, estimación con respecto a la original.

$$G(s) = \frac{0,99888}{0,26207s + 1} e^{-0,20544s} \quad (9)$$

3.4. Controlador PI

Con un controlador proporcional, integral y derivativo conseguimos modular la señal del control del sistema en función del error que existe entre el valor medido y el valor deseado. Se puede emplear el método de Ziegler Nichols, el cual ayuda a regular las ganancias a partir de la respuesta del sistema en lazo abierto de sistemas estables considerando su retardo. (katsuhiko ogata, 1970)

- La parte proporcional aumenta la velocidad de respuesta del sistema, disminuye el error de régimen permanente y aumenta la inestabilidad del sistema.
- La parte integrativa disminuye el error del sistema en régimen permanente y aumenta la inestabilidad y la velocidad del sistema.

- La parte derivativa por otro lado aumenta la estabilidad del sistema controlado, disminuye la velocidad del sistema y el error de régimen permanente no cambia.

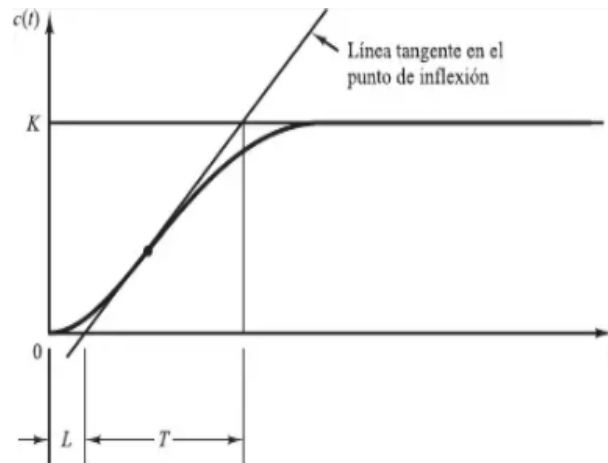


Figura 49: Curva de respuesta en forma S y parámetros a considerar para el Método Ziegler Nichols.

Fuente:(katsuhiko ogata, 1970)

Las variables necesarias para este método se describen en la Figura 49, siendo L un tiempo de retardo y T una constante de tiempo, los valores encontrados con el software Matlab son; $L= 0.47067$ y $T=0.1725$. Con los valores obtenidos se procede a remplazar en las ecuaciones de la Tabla 5 obteniendo como resultado los valores descritos en la Tabla 6.

Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	infinito	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0.5L$

Tabla 5: Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta.

Fuente:(katsuhiko ogata, 1970)

Controlador	K_p	T_i	T_d
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
	0.32985	1.5689	0

Tabla 6: Parámetros obtenidos para controlador PI.

Para probar el funcionamiento se puede utilizar la herramienta de simulink de Matlab para tener una idea teórica del comportamiento de la planta controlada ante una entrada escalón. En la Figura 50, tenemos las curvas obtenidas del sistema en lazo cerrado, notando que el controlador ayuda a que la velocidad llegue al régimen permanente establecido por la planta en lazo abierto, mientras que la planta sin controlar tiende a inestabilizarse y no alcanza el valor máximo de régimen permanente.

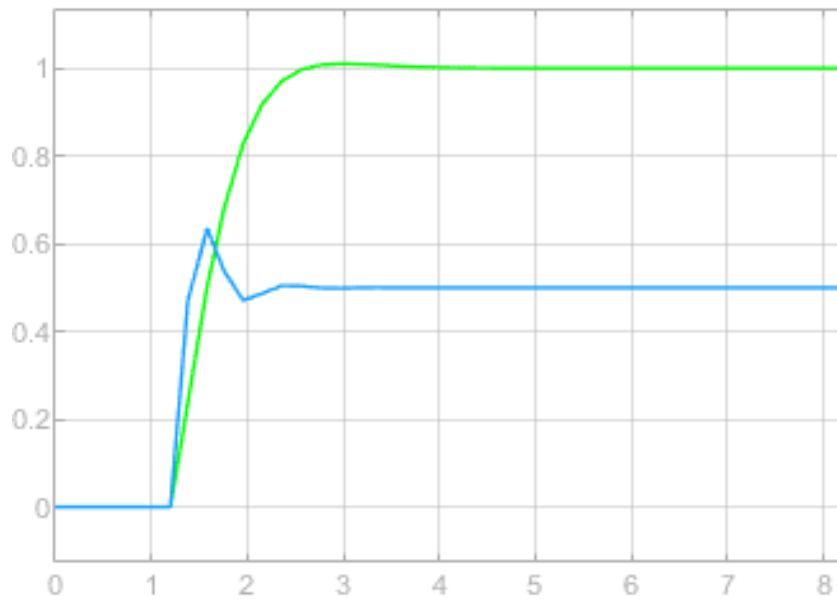


Figura 50: Respuesta del sistema implementando un controlador PI (verde), sistema sin controlador PI (azul).

3.5. Librería para la programación en el Arduino nano 33 BLE

El robot omnidireccional tiene dos microcontroladores el Arduino nano 33 BLE que se utiliza para la comunicación con un control remoto y para la programación de las practicas con el robot, y el Arduino mega que se utiliza como controlador de los motores DC y de los servomotores, en este microcontrolador se encuentran las ecuaciones de la cinemática del robot omnidireccional junto con un control PI para el control de velocidad de los cuatro motores. Del Arduino mega también se utilizan los puertos serial e I2C para la comunicación con los drivers y el Arduino nano 33 BLE.

Con el propósito de simplificar la programación para las practicas con el robot omnidireccional se escribieron funciones para el sketch del Arduino nano 33 BLE. El objetivo fue recopilar todas las líneas de código, que estarían en la programación normal del Arduino,

en un conjunto de funciones con nombres sencillos y pocos parámetros de entrada. Las funciones resultantes se colocaron en el archivo “LibreriaRobot.h” para que en el sketch de programación principal se escriba una sola línea de código, misma que contiene todas las líneas de programación del robot omnidireccional.

En la Figura 51, se presenta el sketch de programación principal con el código para una de las prácticas en la cual se prueban todas las acciones implementadas en el robot omnidireccional.

```
RobotOmnidireccional_ble $ LibreriaRobot.h moduloAnilloLed.h
#include "LibreriaRobot.h"

void setup() {
  Serial.begin(9600);
  setUpI2C();
  setUpBluetooth();
  setUpAnilloLed();
}

void loop() {
  BLEDevice central = BLE.central();
  if (central.connected()){
    BleDesconectado();
    while (central.connected()){
      char comandoMovimiento = moveChar.value();
      moverRobot(comandoMovimiento);
      char comandoGripper = gripperChar.value();
      gripperRobot(comandoGripper);
      arcoirisAnilloLed(30);
      //testCharacteristicsSerial();
    }
  }
  else{BleDesconectado();}
}
```

Figura 51: Sketch de programación principal.

Dentro de la librería “LibreriaRobot.h” se incluyen otras librerías propias de Arduino para la comunicación bluetooth y la comunicación I2C. A continuación, se describirán las funciones de la librería “LibreriaRobot.h”

Funciones del bluetooth	
Función	Descripción
setUpBluetooth()	Contiene las líneas de código para la inicialización del bluetooth y configuración de servicios y características
BleConectado()	Si hay conexión con un dispositivo enciende el led integrado del Arduino, también puede encender el módulo led si está configurado.
BleDesconectado()	Si no hay conexión con un dispositivo apaga el led integrado del Arduino, también puede pagar el módulo led si está configurado.
testCharacteristicsSerial()	Sirve para visualizar los valores de las características enviadas por bluetooth en el monitor serial del Arduino.

Tabla 7: Funciones del bluetooth.

Funciones para el movimiento del robot	
Función	Descripción
moverRobot(char comando)	Esta función envía un valor tipo char por el bus I2C hacia el Arduino mega para establecer la dirección de movimiento del robot. cada dirección tiene asignado una letra en minúscula como se muestra en la Tabla 9.

Tabla 8: Funciones para el movimiento del robot.

Dirección	Letra	Dirección	Letra
Adelante	w	Adelante izquierda	q
Atrás	s	Adelante derecha	e
Izquierda	a	Atrás izquierda	z
Derecha	d	Atrás derecha	c
Giro horario	t	Detenerse	x
Giro antihorario	r		

Tabla 9: Tabla letras asignadas para los movimientos del robot omnidireccional.

Funciones para el módulo del efector final	
Función	Descripción
greaperRobot (char comando)	Esta función envía un valor tipo char por el bus I2C hacia el Arduino mega para subir, bajar y abrir/cerrar el gripper. Cada acción tiene asignado una letra en minúscula como se muestra en la Tabla 11.
greaperAbrir()	Envía la orden de abrir el gripper.
greaperCerrar()	Envía la orden de cerrar el gripper.

Tabla 10: Funciones para el módulo del efector final.

Acción	Letra
Abrir gripper	i
Cerrar gripper	n
Subir	o
Bajar	k

Tabla 11: Letras asignadas para las acciones del módulo gripper.

Funciones para el módulo led	
Función	Descripción
setUpAnilloLed()	Esta función tiene la configuración de los leds neopixel para funcionar con el Arduino.
apagarAnilloLed()	Apaga todos los 24 leds del anillo
setColorAnilloLed (uint8_t R,uint8_t G,uint8_t B)	Esta función permite controlar el color del anillo led con los valores RGB.
colorAnilloLed(int color)	Esta función envía un color determinado en base al valor de la variable color, como se presenta en Tabla 13. .
arcoirisAnilloLed(int wait)	Esta función realiza la animación de arcoíris giratorio en los leds. La variable wait configura el tiempo de actualización de la animación en milisegundos. Por defecto debe tener un valor de 30.

Tabla 12: Funciones para el módulo led.

Color	Valor
Azul	1
Rojo	2
Verde	3

Tabla 13: Valores asignados para el color del módulo led.

3.6. Programación del firmware del robot

El Arduino mega se utiliza como firmware del robot omnidireccional para recibir comandos y controlar la velocidad de los motores DC y la posición de los servomotores. Este Arduino tendrá el código cargado previamente y no será programado por los estudiantes en las prácticas del taller. Debido a esto, la programación está escrita dentro de un solo sketch y se utilizan funciones tal cual están definidas las librerías de Arduino que son:

- Wire: para la comunicación I2C.
- pololuMaestro: para el control de los servomotores por el puerto serial.
- PID_V1: para establecer el control de la velocidad de los motores DC con un controlador PI.
- Util/atomic: para devolver los valores de los encoders después de la interrupción.

3.6.1. Programación del control de velocidad

El control de velocidad de los motores DC se ejecuta en el Loop del sketch. Dentro del Loop se calcula el valor de control para las entradas PWM en el driver de cada motor DC, después se mide la velocidad de los motores en RPM con el valor de los encoders. La velocidad se calcula con la siguiente fórmula:

$$velocidad = \frac{60 \cdot (valorActual - valorPrevio)}{resolucionEncoder \cdot (tiempoActual - tiempoPrevio)} [RPM] \quad (10)$$


```

void loop() {
  int encoderMotor1 = 0;
  int encoderMotor2 = 0;
  int encoderMotor3 = 0;
  int encoderMotor4 = 0;
  checkSetpoint();
  myPID1.Compute();
  myPID2.Compute();
  myPID3.Compute();
  myPID4.Compute();
  analogWrite(PWM1, Output1);
  analogWrite(PWM2, Output2);
  analogWrite(PWM3, Output3);
  analogWrite(PWM4, Output4);
  ATOMIC_BLOCK(ATOMIC_RESTORESTATE) {
    encoderMotor1 = pos_i1;
    encoderMotor2 = pos_i2;
    encoderMotor3 = pos_i3;
    encoderMotor4 = pos_i4;
  }
  getVelocidadMotor(encoderMotor1, encoderMotor2, encoderMotor3, encoderMotor4);
  printVelocidades();
}

```

Figura 52: Control de velocidad.

Se utilizaron cuatro pines de interrupción para la lectura de los encoders en los cuatro motores DC. Los pines de interrupción se configuraron para reaccionar al flanco de subida y además se configuro la resistencia de pull-up en la entrada de estos.

Los valores de las constantes $K_p = 0.36$ y $K_i = 1.52$ son las implementadas después de la sintonización por Ziegler-Nichols y una afinación experimental en base a la respuesta percibida del robot omnidireccional.

3.7. Programación de la comunicación I2C

Cuando se recibe un dato por el bus I2C se ejecuta una función para clasificar el carácter como comando para movimiento del robot o comando para el módulo del efector final.

```

//funcion i2c
void comandoBle(int a){
  char accion = Wire.read();
  if (accion == 'w' || accion == 'a' || accion == 's' || accion == 'd' || accion == 'q' || accion == 'e' ||
      accion == 'z' || accion == 'c' || accion == 'r' || accion == 't' || accion == 'x')
  {
    movimientoRobot = accion;
    if (accion == 'x'){testSetpoint =0;}
    velocidadGlobalRobot();
  }
  if ((accion == 'j') || (accion == 'o') || (accion == 'k') || (accion == 'm') || (accion == 'i') || (accion == 'n')){
    greaperRobot = accion;
    greaperSubirBajar();
    greaperAbrirCerrar();
  }
  //Serial.println(accion);// Serial.println(" ");
}

```

Figura 53: Función I2C.

Si se recibe un comando de movimiento se asigna un valor para las velocidades V_x , V_y , W dependiendo del carácter recibido. Con estas velocidades se calcula la velocidad angular en RPM para cada una de las cuatro ruedas utilizando la cinemática del robot visto en la sección 3.2 del documento. Con las velocidades resultantes se ajusta el punto de consigna para el control PI y se determina la dirección de giro de las ruedas.

```

void setPointMotores(){
  double velocidad1 = ((Vx-Vy-(A+B)*W)*60.0)/(2.0*PI*R);
  double velocidad2 = ((Vx+Vy+(A+B)*W)*60.0)/(2.0*PI*R);
  double velocidad3 = ((Vx+Vy-(A+B)*W)*60.0)/(2.0*PI*R);
  double velocidad4 = ((Vx-Vy+(A+B)*W)*60.0)/(2.0*PI*R);
  if (velocidad1 > 0.0){digitalWrite(INA1, false);digitalWrite(INB1, true);}
  if (velocidad1 < 0.0){digitalWrite(INA1, true);digitalWrite(INB1, false);}
  if (velocidad2 < 0.0){digitalWrite(INA2, false);digitalWrite(INB2, true);}
  if (velocidad2 > 0.0){digitalWrite(INA2, true);digitalWrite(INB2, false);}
  if (velocidad3 > 0.0){digitalWrite(INA3, false);digitalWrite(INB3, true);}
  if (velocidad3 < 0.0){digitalWrite(INA3, true);digitalWrite(INB3, false);}
  if (velocidad4 < 0.0){digitalWrite(INA4, false);digitalWrite(INB4, true);}
  if (velocidad4 > 0.0){digitalWrite(INA4, true);digitalWrite(INB4, false);}
  Setpoint1 = abs(velocidad1);
  Setpoint2 = abs(velocidad2);
  Setpoint3 = abs(velocidad3);
  Setpoint4 = abs(velocidad4);
}

```

Figura 54: Programación de la cinemática del robot omnidireccional.

Si se recibe un comando para el módulo gripper este se envía a dos funciones correspondientes a las acciones de subir/bajar el gripper o abrir/cerrar el gripper. En estas funciones se utiliza la librería del driver mini maestro de Pololu para controlar la posición del servomotor en base al tiempo de pulso en milisegundos.

```

void greaperAbrirCerrar(){
  switch(greaperRobot){
    case 'j':
      if ((greaperAbierto == true) && (estadoGreaper == 0)){
        driverServo18.setTarget(servo2, servo2Min*4); greaperAbierto = false; estadoGreaper = 1;
      }
      if ((greaperAbierto == false)&& (estadoGreaper == 0)){
        driverServo18.setTarget(servo2, servo2Max*4); greaperAbierto = true; estadoGreaper = 1;
      }
      break;
    case 'm':
      estadoGreaper = 0;
      break;
    case 'i':
      driverServo18.setTarget(servo2, servo2Min*4); greaperAbierto = false;
      break;
    case 'n':
      driverServo18.setTarget(servo2, servo2Max*4); greaperAbierto = true;
      break;
    default:
      //maestro.setTarget(servo2, 2500*4); greaperAbierto = true;
      break;
  }
}

```

Figura 55: Programación de los movimientos del efector final.

4. CAPÍTULO 4: MANUAL PRACTICO PARA ENSAMBLE PROGRAMACIÓN Y USO DE MÓDULOS DEL KIT ROBÓTICO

4.1. Ensamble del cuerpo del robot

El objetivo de esta práctica es ensamblar todas las piezas del robot omnidireccional. El robot omnidireccional consta de tres partes:

- 1 Estructura inferior
- 2 Estructura de soporte de ruedas
- 3 Cubierta superior

Para el ensamble del robot se necesitarán las siguientes herramientas:



Figura 56: Llaves hexagonales tipo hallen (A), Llaves hexagonales tipo boca(B).

4.1.1. Ensamble de la base

Para el ensamble de la base inferior se utilizarán los siguientes pernos:

Diametro[mm]	Longitud[mm]	Cantidad
M6	20	6
M4	20	4
M3	16	4

Tabla 14: Pernos de que se emplearán en la sujeción de la base.


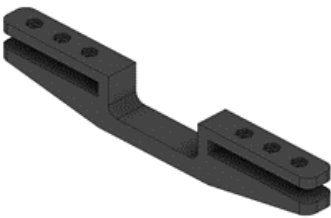
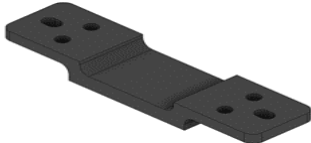
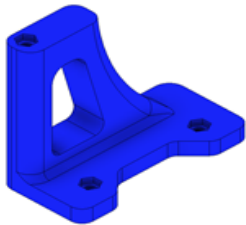
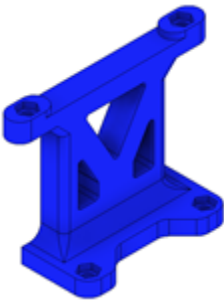
Nombre	Cantidad	Imagen
base	2	
unión Auxiliar	2	
unión central	1	
soporte de tarjeta electrónica A	1	
soporte de tarjeta electrónica B	1	

Tabla 15: Piezas que forman la estructura inferior del robot.

Paso 1

Piezas	Cantidad
Base	1

Perno	Longitud	Cantidad
M6	20mm	3

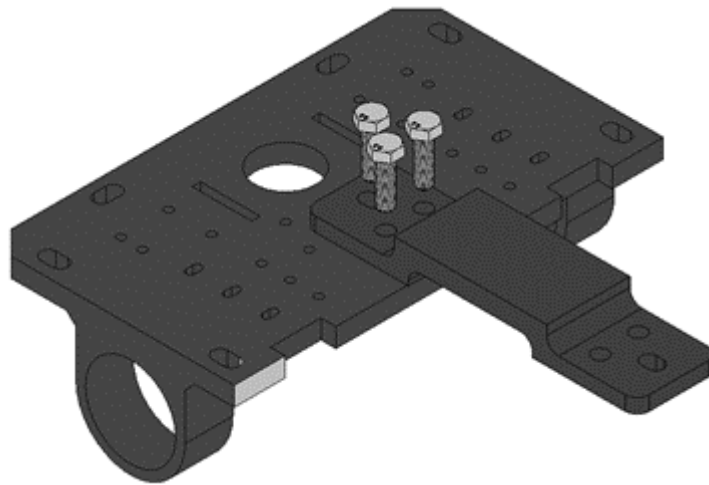


Figura 57: Base y soporte con inserción de pernos.

Paso 2

Piezas	Cantidad
Unión auxiliar	2

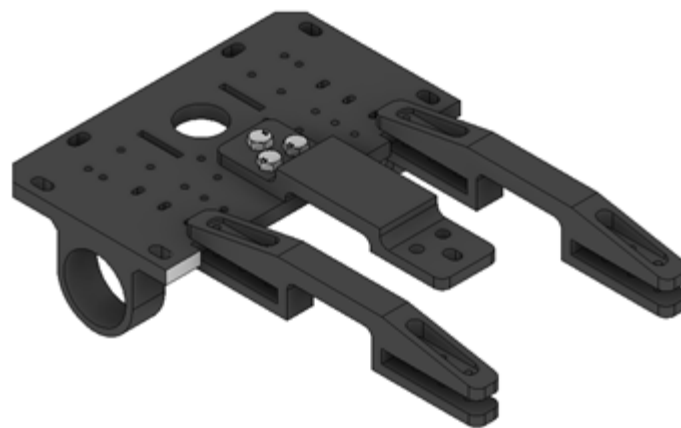


Figura 58: Base y soportes.

Paso 3

Perno	Longitud	Cantidad
M4	20mm	2

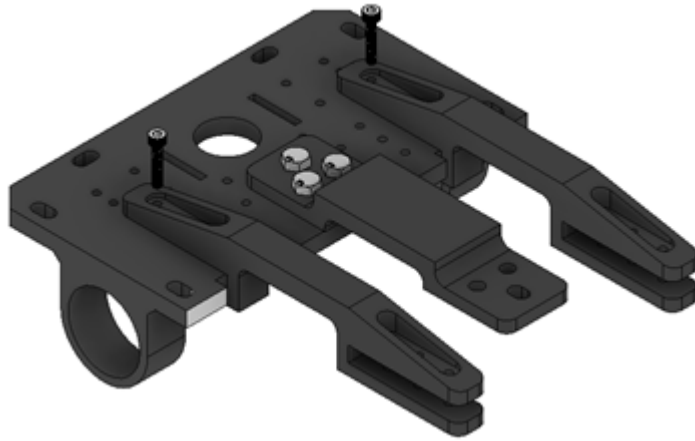


Figura 59: Base y soportes con inserción de pernos.

Paso 4

Piezas	Cantidad
Base	1



Figura 60: Bases y soportes.

Paso 5

Perno	Longitud	Cantidad
M6	20mm	3
M4	20mm	2

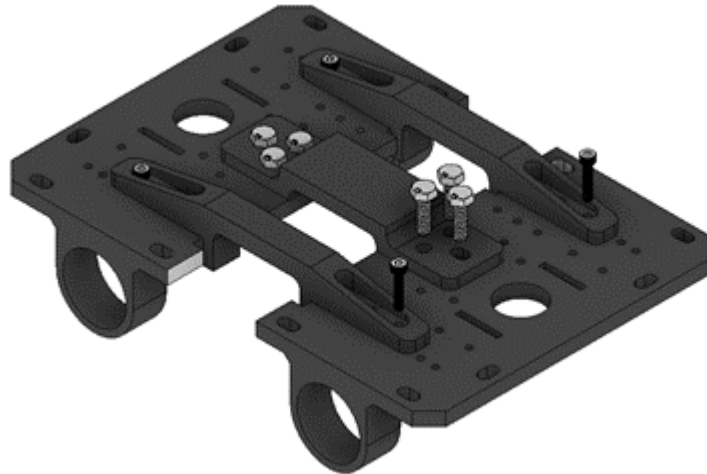


Figura 61: Bases y soportes con inserción de pernos.

Paso 6

Piezas	Cantidad
Soporte tarjeta A	1
Soporte tarjeta B	1

Perno	Longitud	Cantidad
M3	16mm	4

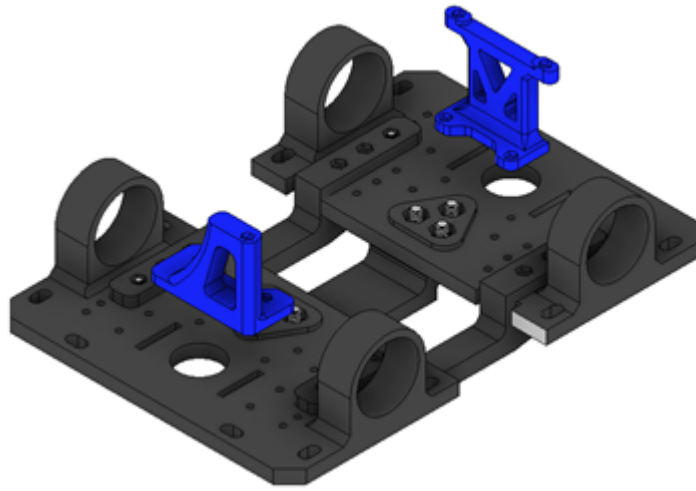


Figura 62: Bases y soportes de placa.

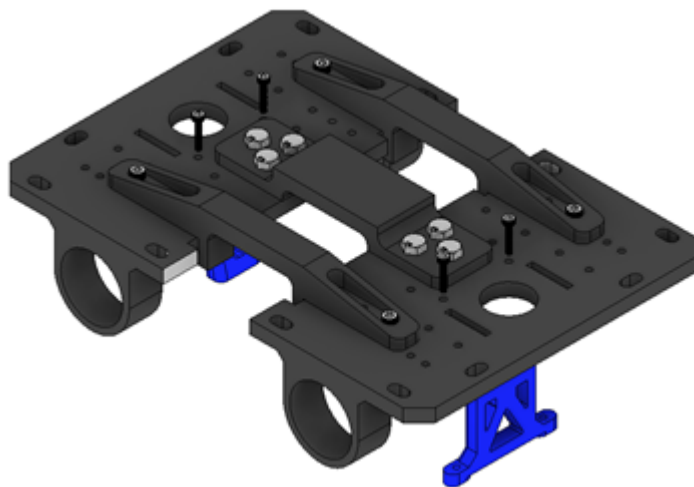


Figura 63: Bases y soportes de placa con inserción pernos.

La estructura inferior terminada debe tener la siguiente vista:

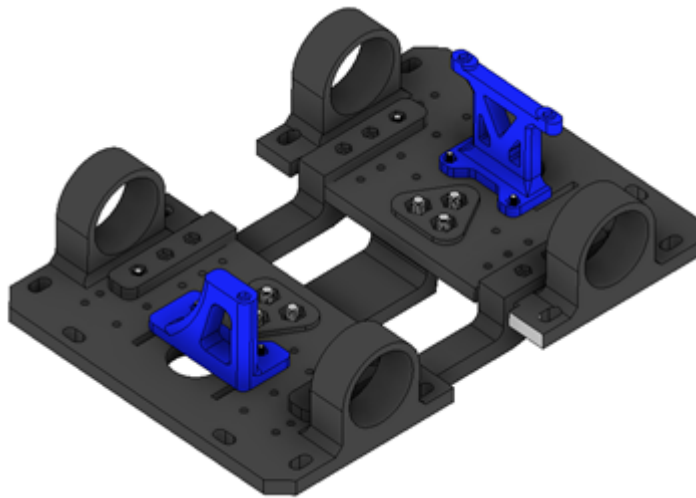


Figura 64: Bases y soportes de placa final

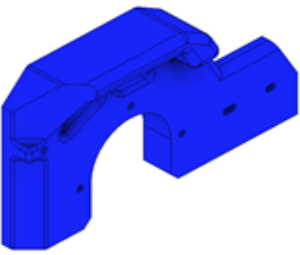
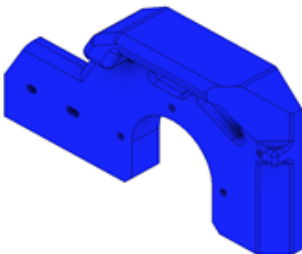
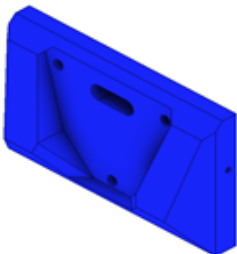
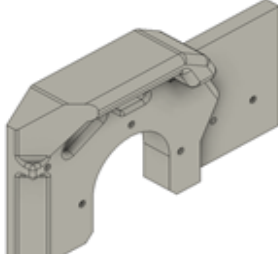
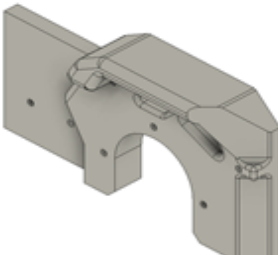
4.1.2. Ensamble de las paredes y acople de ruedas.

Para el ensamble de la base inferior se utilizarán los siguientes pernos:

Diametro[mm]	Longitud[mm]	Cantidad
M6	25	8
M6	20	4
M3	30	4
M3	20	12
M3	18	4

Tabla 16: Pernos que se emplearán en la sujeción de las paredes con la base.

Las piezas de la estructura lateral son las siguientes:

Nombre	Cantidad	Imagen
Pared lateral A1	1	
Pared lateral A2	1	
Pared frontal azul	1	
Pared lateral B4	1	
Pared lateral B3	1	

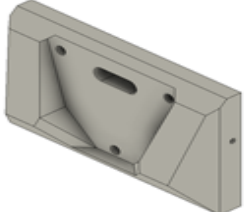
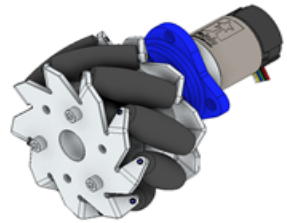
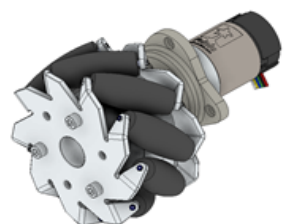
Pared frontal gris	1	
Motor y rueda con soporte azul	2	
Motor y rueda con soporte gris	2	

Tabla 17: Piezas que forman la estructura lateral del robot.

Paso 1

Piezas		Cantidad
Pared lateral B3		1
Pared lateral B4		1

Perno	Longitud	Cantidad
M6	25mm	2
M6	20mm	2

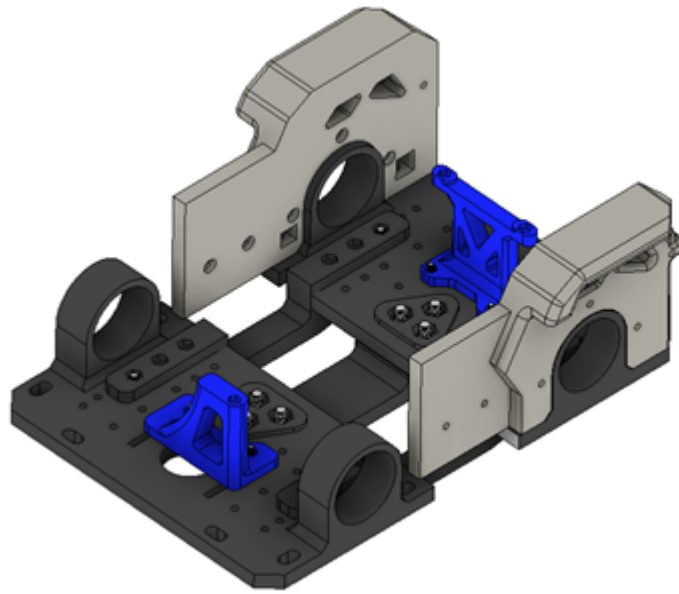


Figura 65: Base y paredes laterales grises.

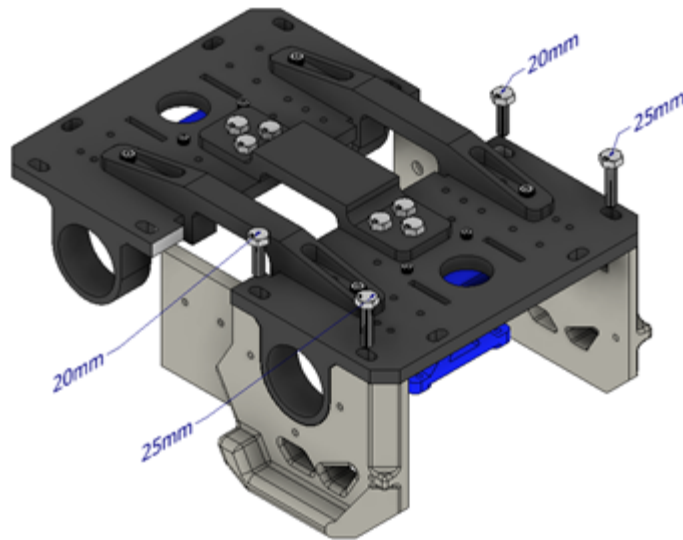


Figura 66: Base y paredes laterales grises con inserción de pernos.

Paso 2

Piezas	Cantidad
Pared lateral A1	1
Pared lateral A2	1

Perno	Longitud	Cantidad
M6	25mm	2
M6	20mm	2

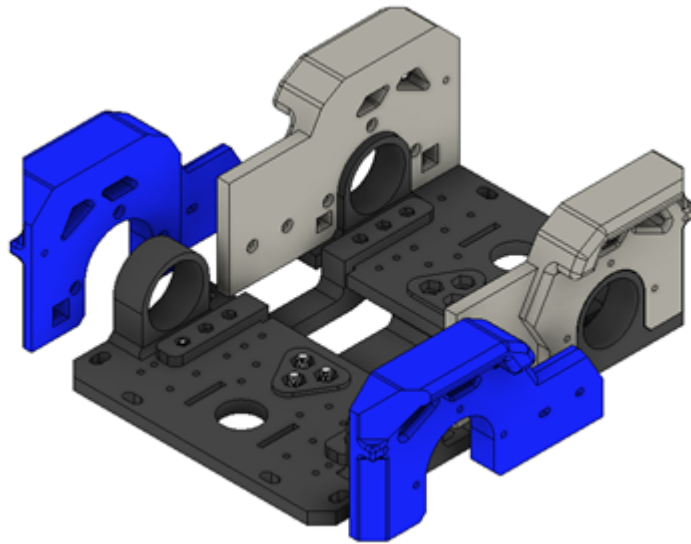


Figura 67: Base y paredes laterales azules.

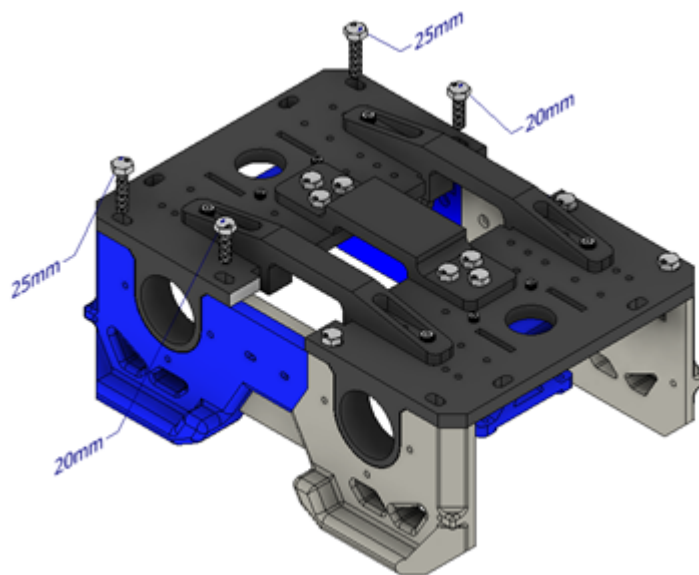


Figura 68: Base y paredes laterales azules con inserción de pernos.

Paso 3

Piezas		Cantidad
Motor 1 con sujetador azul		1
Motor 3 con sujetador gris		1

Perno	Longitud	Cantidad
M3	20mm	6

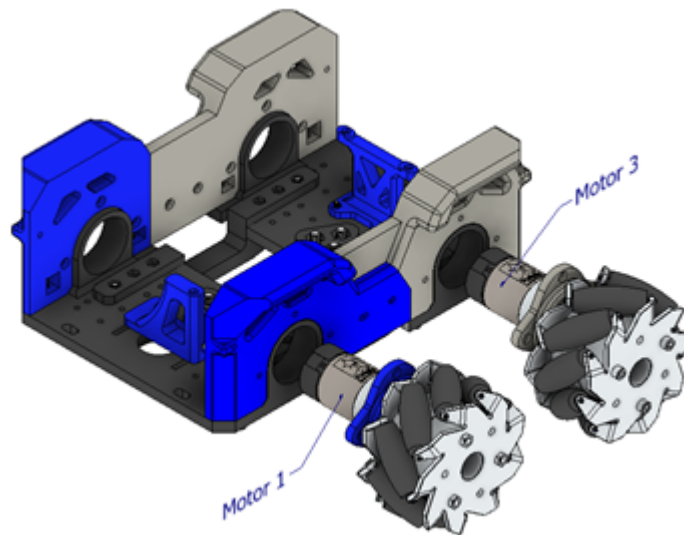


Figura 69: Base, paredes y motores 1 y 3.

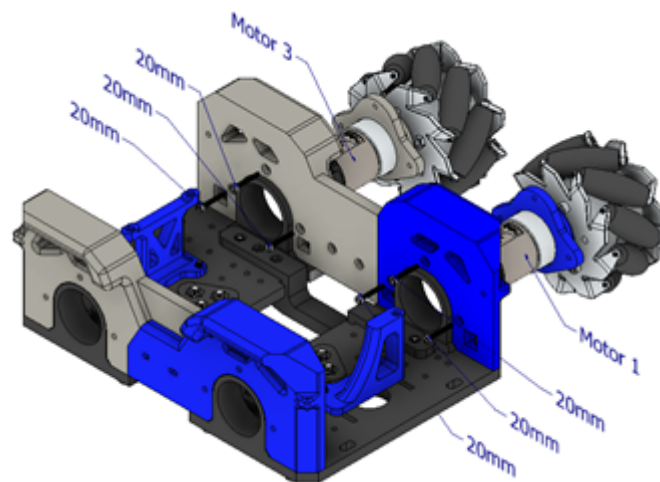


Figura 70: Base, paredes y motores 1 y 3 con inserción de pernos.

Paso 4

Piezas		Cantidad
Motor 2 con sujetador azul		1
Motor 4 con sujetador gris		1

Perno	Longitud	Cantidad
M3	20mm	6

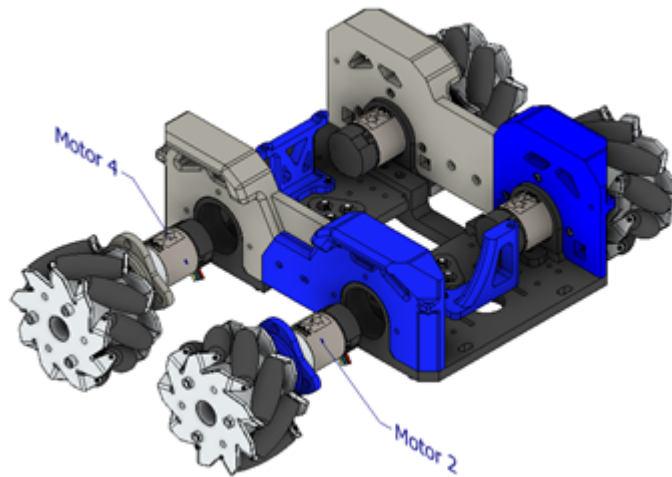


Figura 71: Base, paredes y motores 2 y 4.

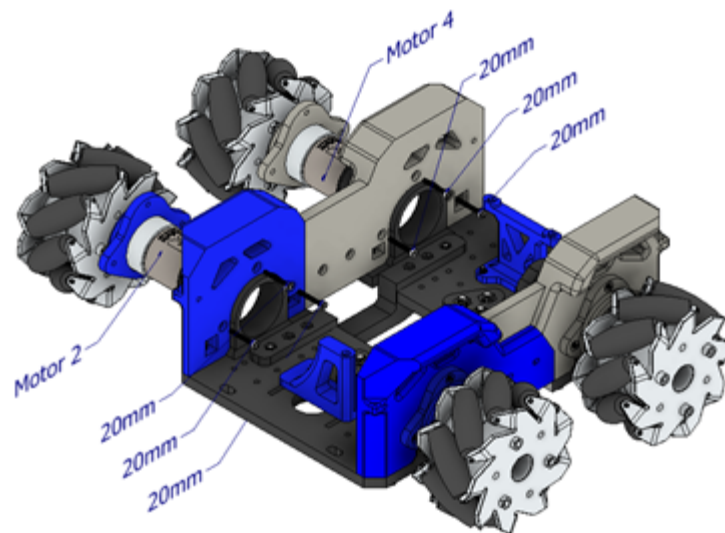


Figura 72: Base, paredes y motores 2 y 4 con inserción de pernos.

Paso 5

Piezas		Cantidad
Pared frontal azul		1
Perno	Longitud	Cantidad
M6	25mm	2
M3	30mm	2

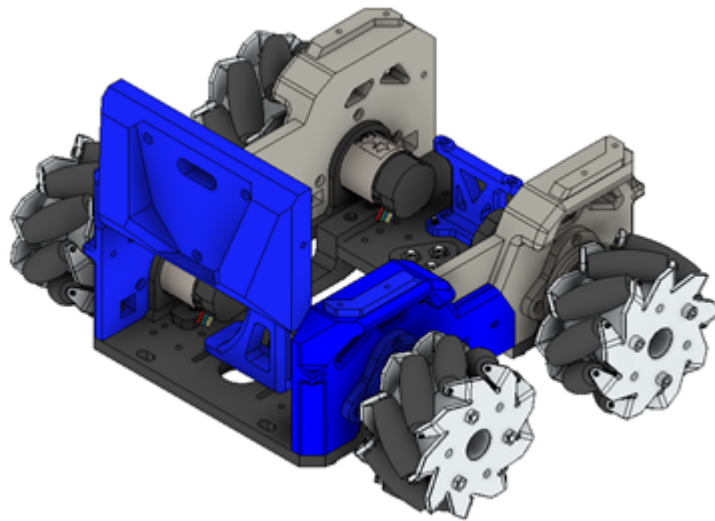


Figura 73: Base, paredes, motores y pared frontal azul.

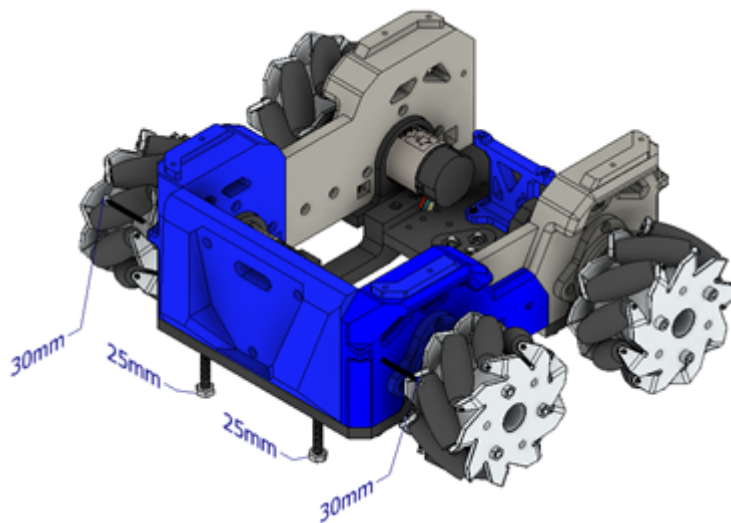


Figura 74: Base, paredes, motores y pared frontal azul con inserción de pernos, (vista superior).

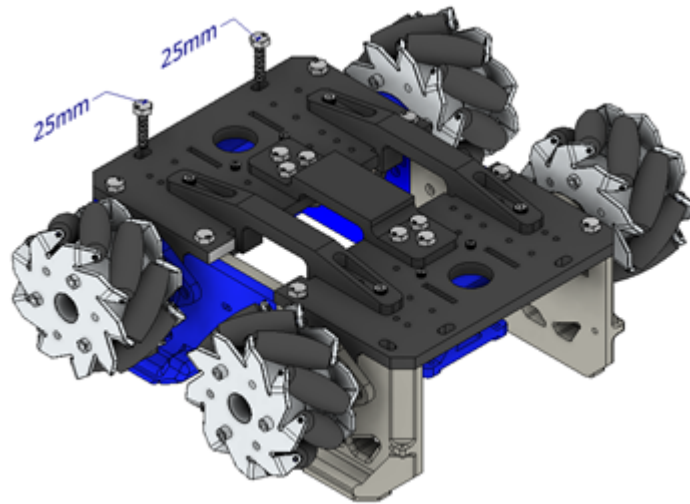


Figura 75: Base, paredes, motores y pared frontal azul con inserción de pernos, (vista inferior).

Paso 6

Piezas		Cantidad
Pared frontal gris		1
Perno	Longitud	Cantidad
M6	25mm	2
M3	30mm	2

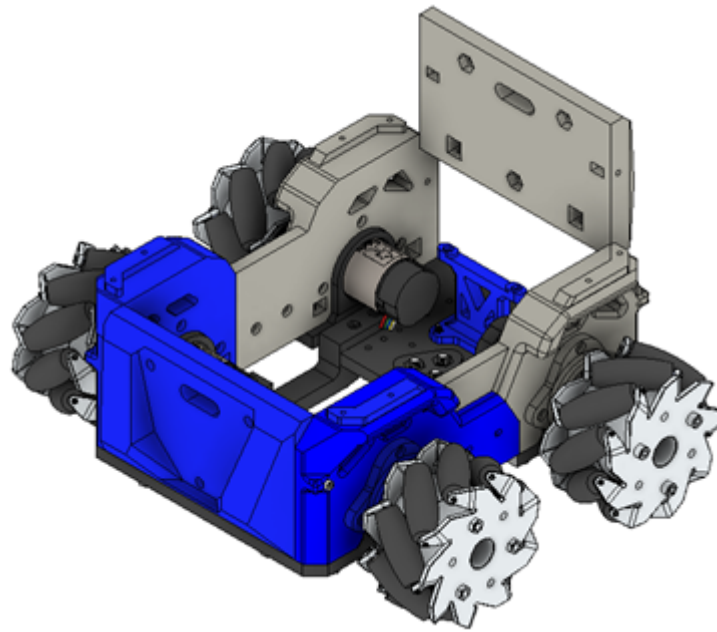


Figura 76: Base, paredes, motores y pared frontal gris.

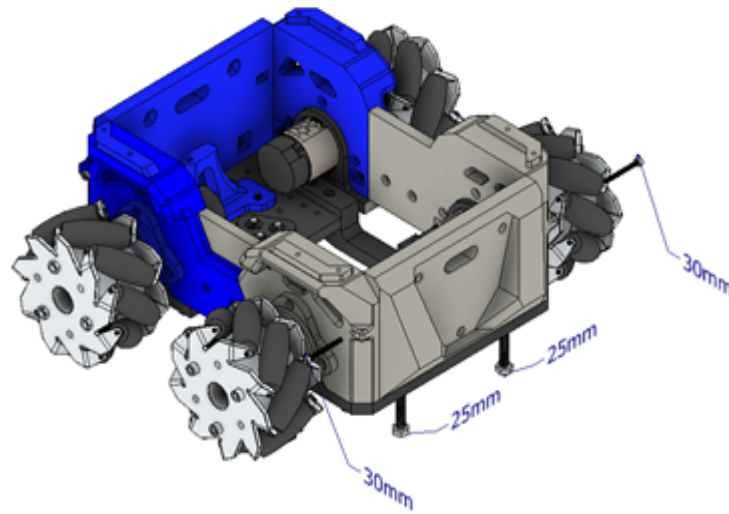


Figura 77: Base, paredes, motores y pared frontal gris con inserción de pernos, (vista superior).

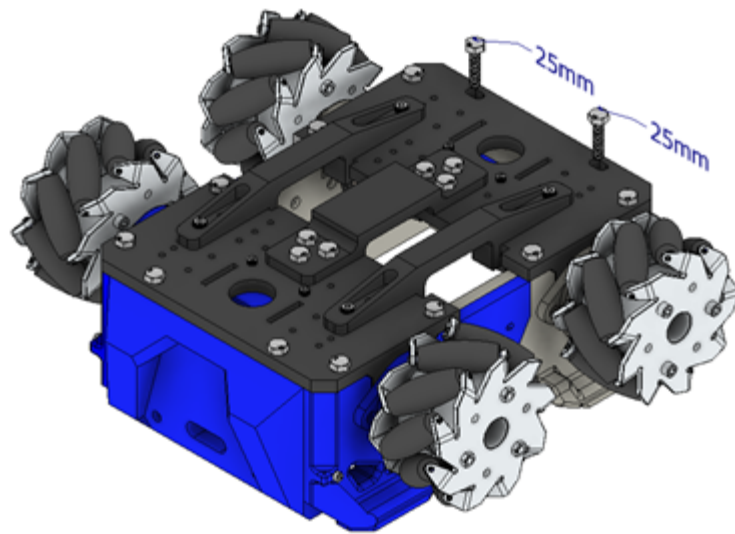


Figura 78: Base, paredes, motores y pared frontal gris con inserción de pernos, (vista inferior).

Paso 7

Perno	Longitud	Cantidad
M3	18mm	4

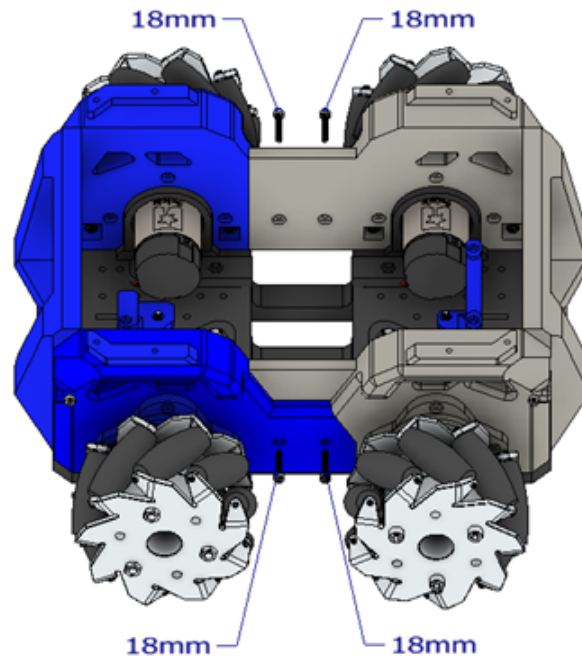


Figura 79: Inserción de pernos laterales para union de paredes.

El ensamble final debe tener la siguiente vista:

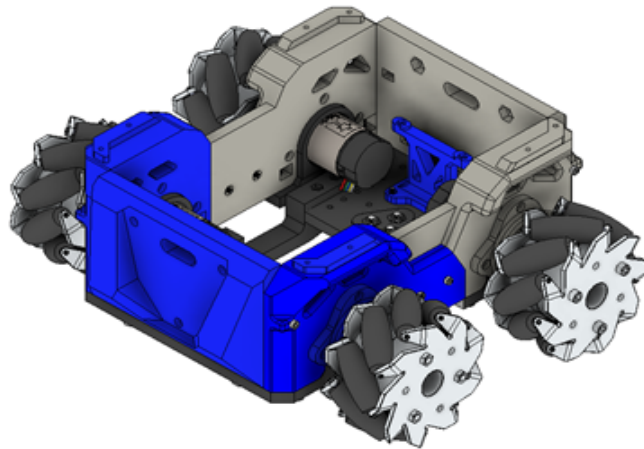


Figura 80: Ensamble del chasis, paredes y motores del robot.

4.1.3. Ensamble de la cubierta del robot

Para el ensamble de la cubierta se utilizarán los siguientes pernos:

Diametro[mm]	Longitud[mm]	Cantidad
M3	8	8

Tabla 18: Pernos de que se emplearán en la sujeción de la cubierta.

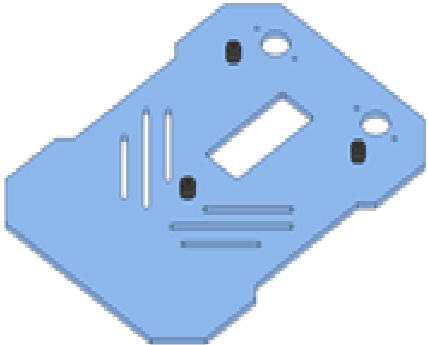
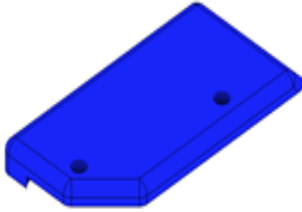
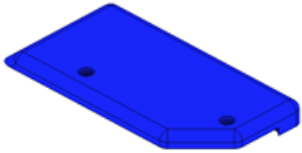
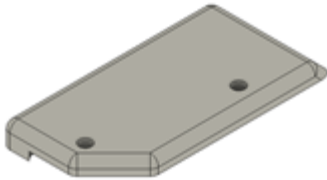
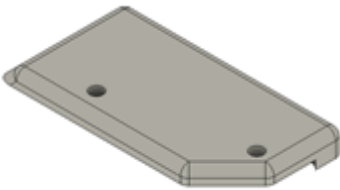
Nombre	Cantidad	Imagen
Tapa de acrílico	1	
Sujetador de acrílico azul izquierdo	1	
Sujetador de acrílico azul derecho	1	
Sujetador de acrílico gris derecho	1	
Sujetador de acrílico gris izquierdo	1	

Tabla 19: Piezas que forman la cubierta del robot.

Paso 1

Piezas	Cantidad
Tapa de acrílico	1

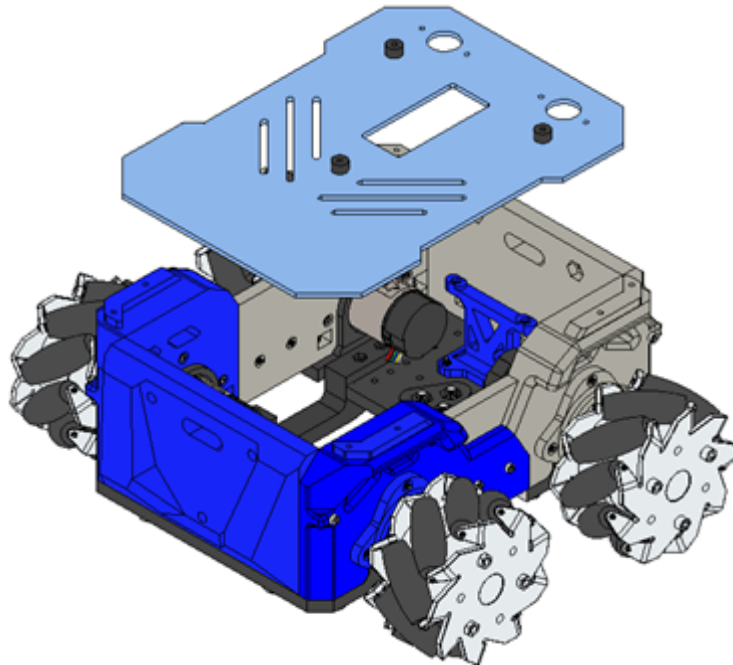


Figura 81: Ensamble de la tapa de acrílico

Paso 2

Piezas	Cantidad
Sujetador acrílico azul izquierdo	1
Sujetador acrílico azul derecho	1
Sujetador acrílico gris izquierdo	1
Sujetador acrílico gris derecho	1

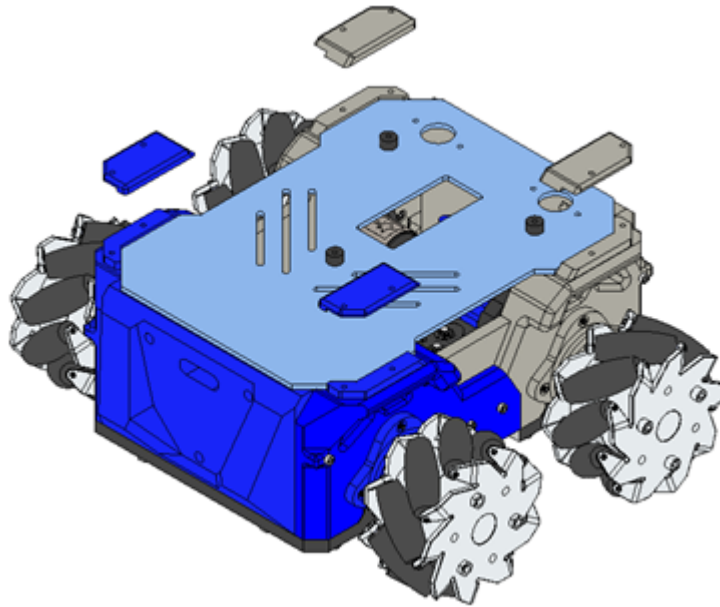


Figura 82: Piezas para ensamble de la tapa de acrílico.

Paso 3

Perno	Longitud	Cantidad
M3	8mm	8

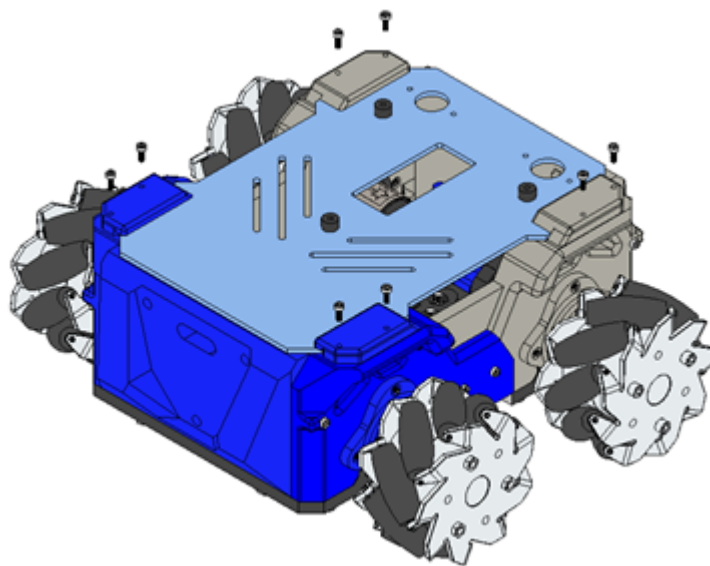


Figura 83: Piezas para ensamble de la tapa de acrílico con inserción de pernos.

4.1.4. Ensamble final del cuerpo del robot

Al terminar de ensamblar las tres partes el robot omnidireccional debe tener la siguiente apariencia.

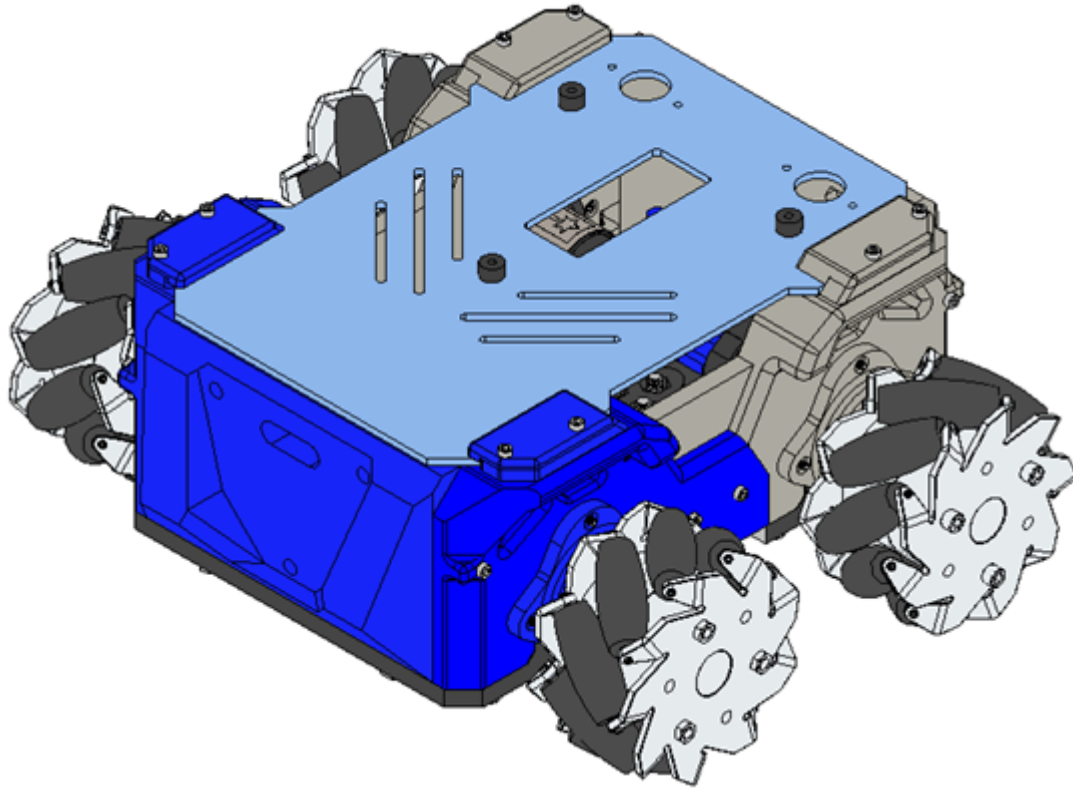


Figura 84: Apariencia del robot al final del ensamble.

4.2. Ensamble de componentes en la tarjeta electrónica

El objetivo de esta práctica es colocar los componentes electrónicos en la tarjeta electrónica del robot omnidireccional.

En la siguiente tabla se muestran los componentes electrónicos que se colocaran:

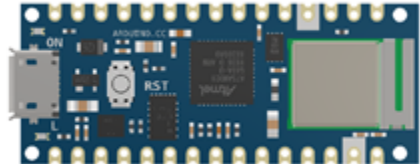
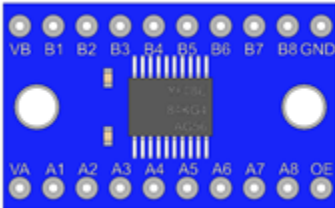
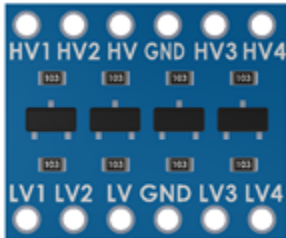
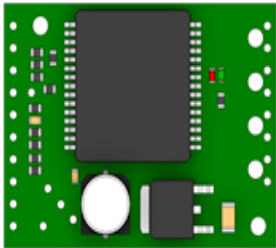
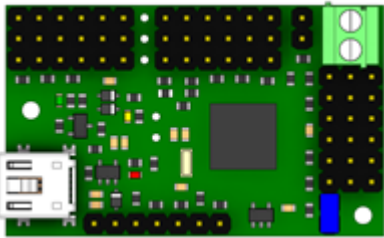
Nombre	Cantidad	Imagen
Arduino nano 33	1	
Convertidor lógico 8 canales	2	
Convertidor lógico 4 canales	1	
Driver motor DC	4	
Driver servo motores	1	

Tabla 20: Módulos de la tarjeta electrónica parte 1.

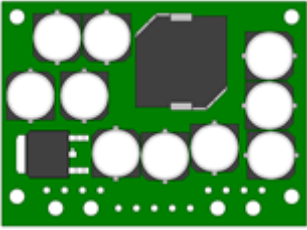
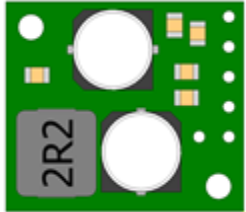
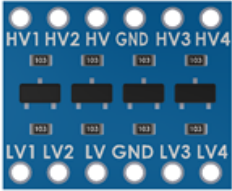
Nombre	Cantidad	Imagen
Regulador de voltaje de 7V	1	
Regulador de voltaje 2R2	1	
Regulador de voltaje 3R3	1	

Tabla 21: Módulos de la tarjeta electrónica parte 2.

Para comenzar, se colocará la tarjeta electrónica en la siguiente posición:

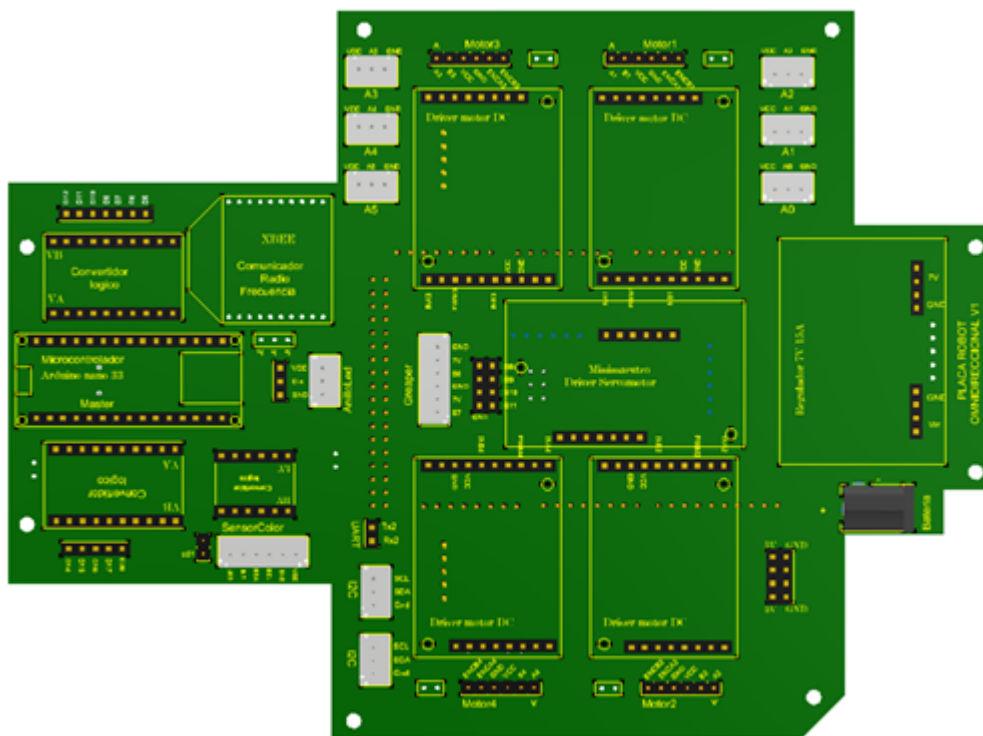


Figura 85: Posición de inicio de la tarjeta electrónica.

Paso 1: se colocarán los siguientes componentes:

No.	Nombre	Cantidad
1	Microcontrolador Arduino nano 33	1
2	Convertidor lógico de 8 canales	2
3	Convertidor lógico de 4 canales	1
4	Comunicador de radio frecuencia Xbee	1

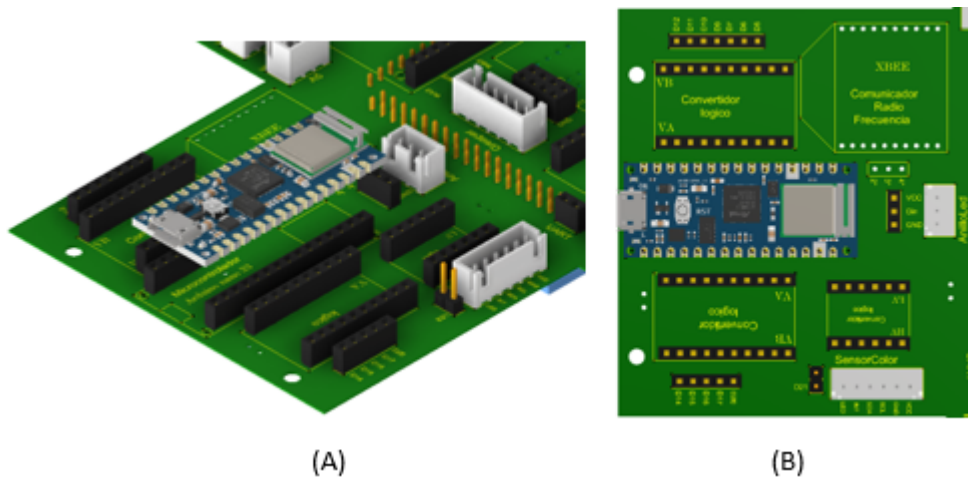


Figura 86: Colocación de Arduino nano 33 (A), orientación.

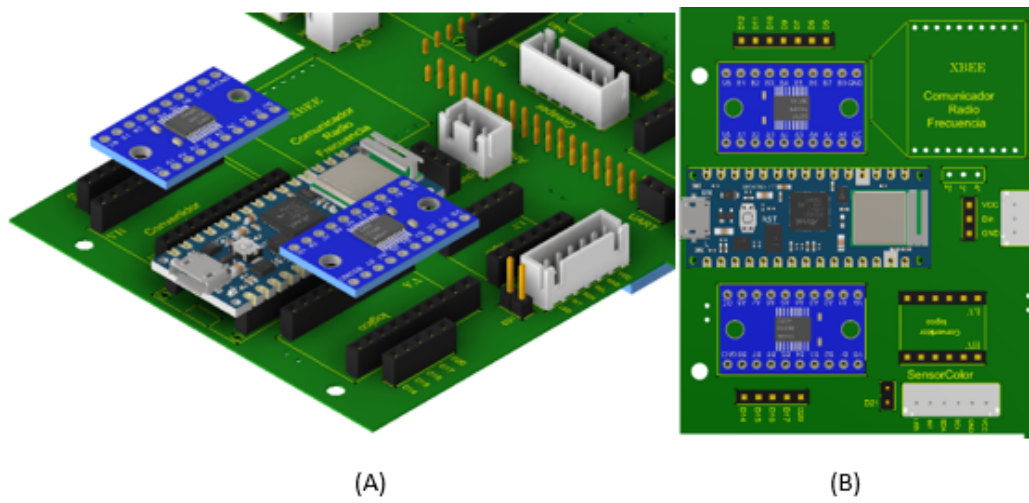


Figura 87: Colocación de convertidores lógicos 8 canales (A), orientación (B).

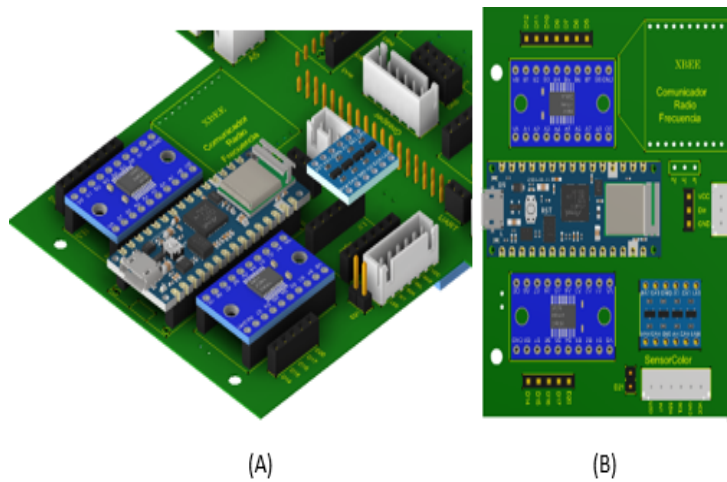


Figura 88: Colocación de convertidor lógico 4 canales (A), orientación (B).

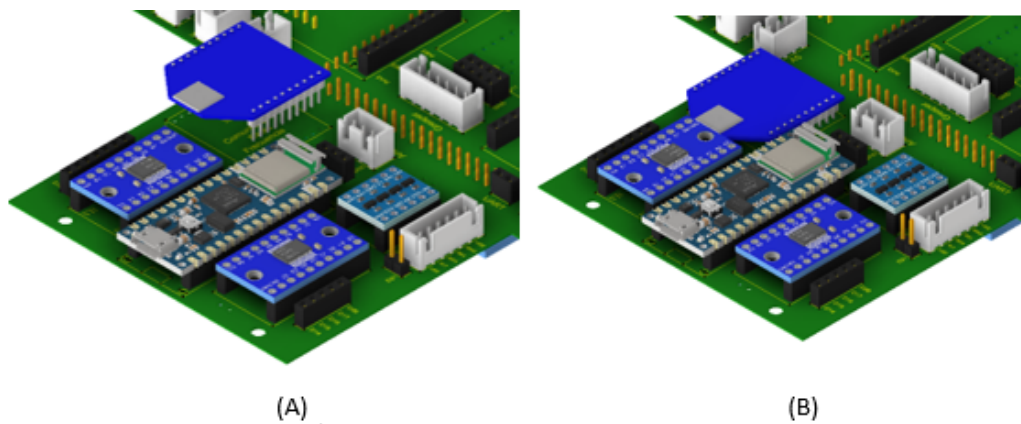


Figura 89: Colocación de módulo Xbee (A), orientación (B).

Paso 2: se colocarán los siguientes componentes:

No.	Nombre	Cantidad
1	Driver de motor DC	4
2	Driver de servomotor	1
3	Regulador de voltaje 7V	1

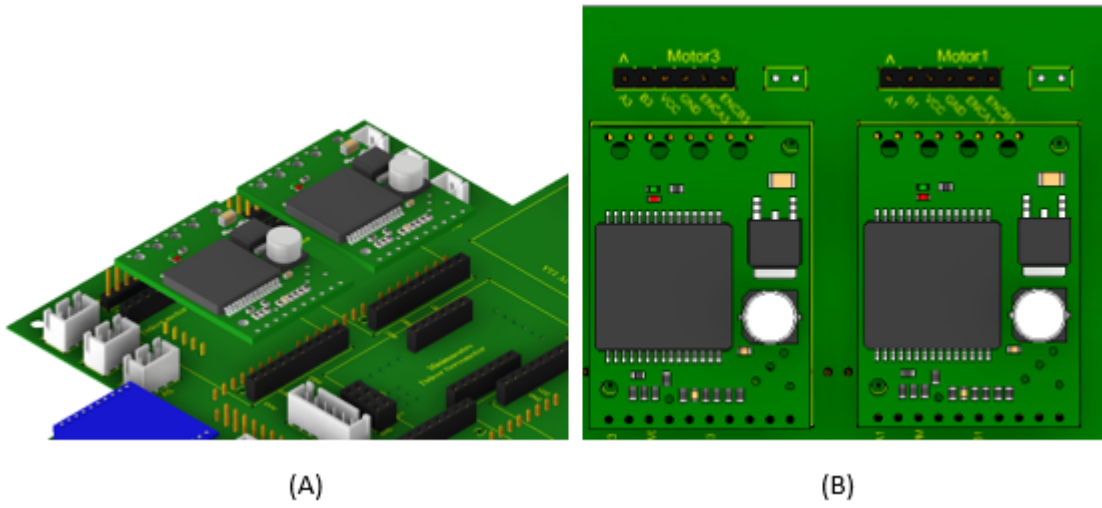


Figura 90: Colocación de drives 1 y 3 (A), orientación (B).

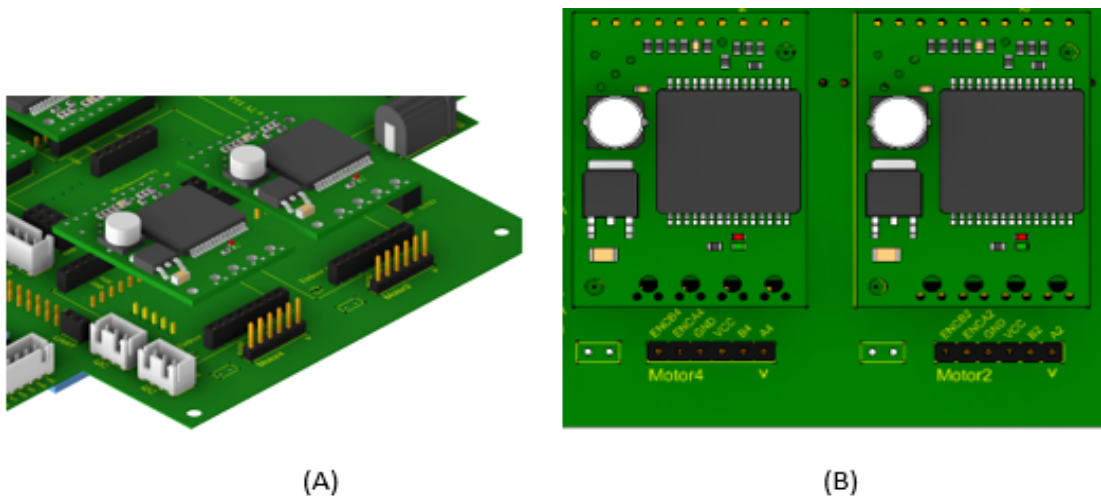


Figura 91: Colocación de drives 2 y 4 (A), orientación (B).

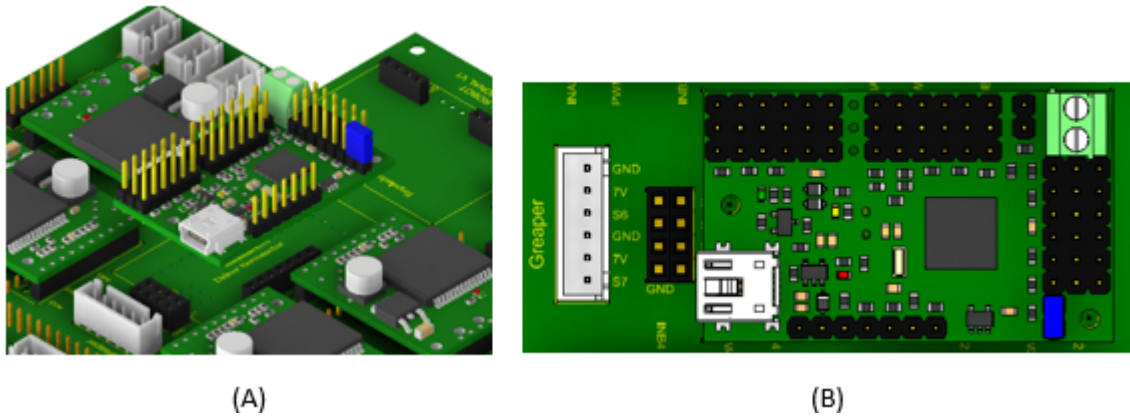


Figura 92: Colocación de mini maestro (A), orientación (B).

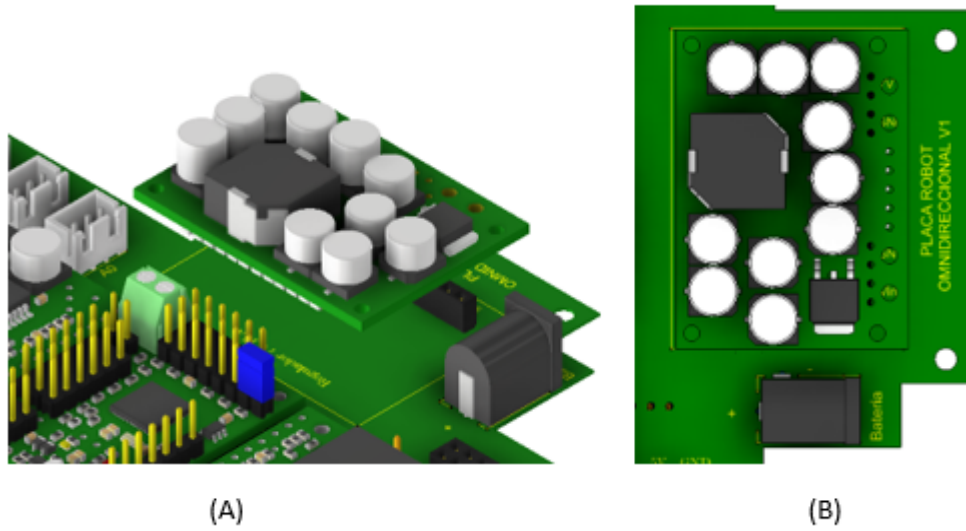


Figura 93: Colocación de regulador 7V (A), orientación (B).

Paso 3: con la tarjeta volteada procedemos a colocar los siguientes componentes:

No.	Nombre	Cantidad
1	Regulador voltaje 3.3V (2R2)	1
2	Regulador voltaje 5V (3R3)	1

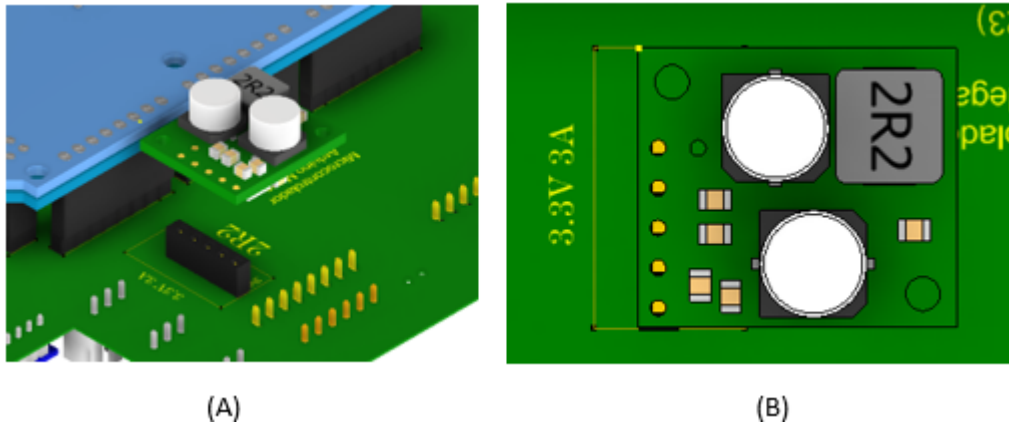


Figura 94: Colocación de regulador 3.3V (A), orientación (B).

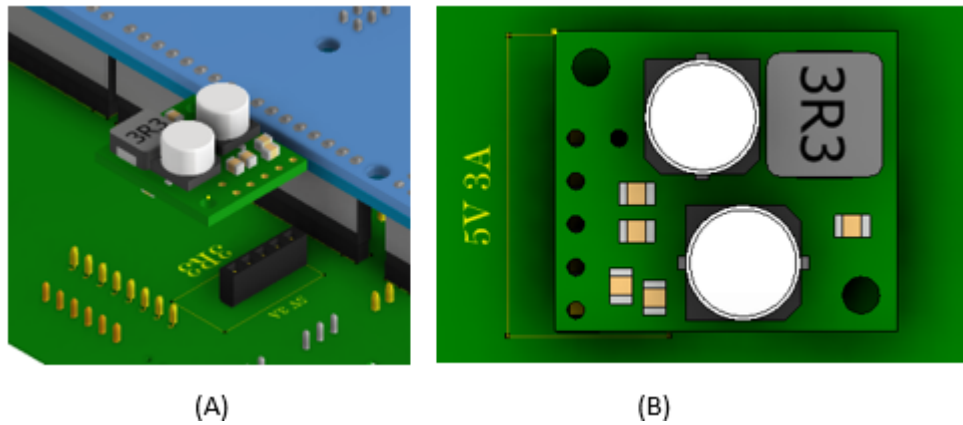


Figura 95: Colocación de regulador 5V (A), orientación (B).

Al terminar la tarjeta electrónica debe tener la siguiente apariencia.

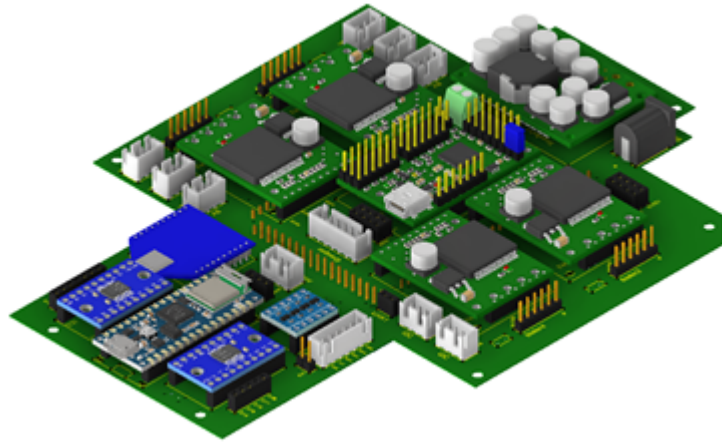


Figura 96: Tarjeta armada (Vista superior).

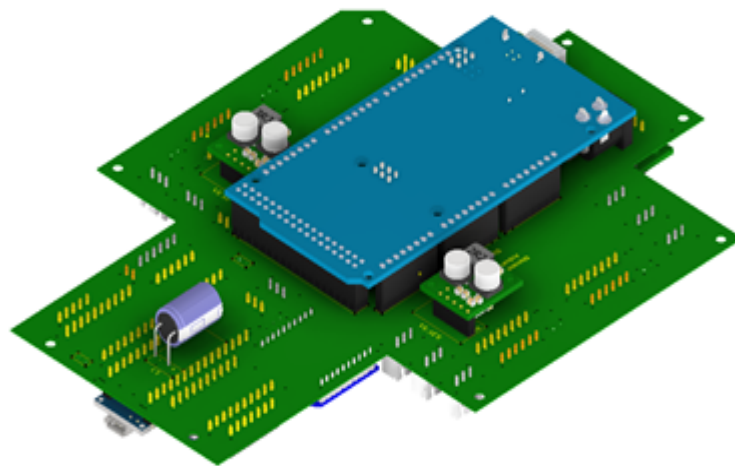


Figura 97: Tarjeta armada (Vista inferior).

4.2.1. Colocación de la batería

La batería se colocará quitando la cubierta superior del robot. la batería debe colocarse de tal forma que los cables rojo y negro con el conector amarillo queden en la parte gris del robot.

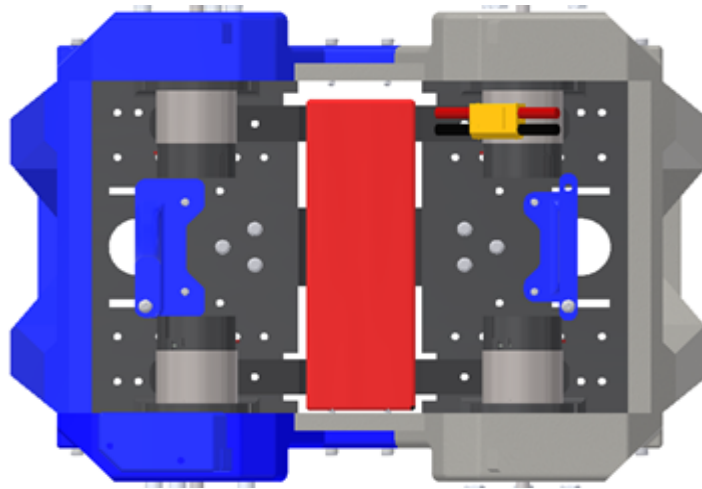


Figura 98: Colocación de la batería.

4.2.2. Colocación de la tarjeta electrónica

Después de colocar la batería se ubicará la tarjeta electrónica en la parte superior con la siguiente posición.

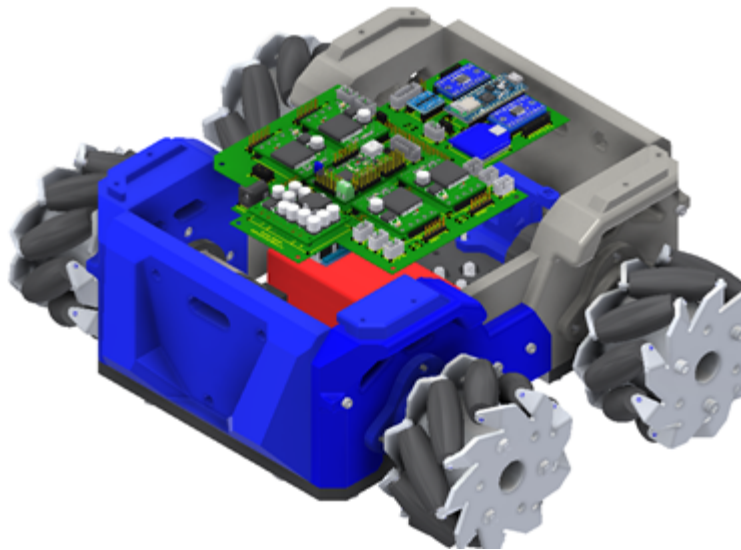


Figura 99: Posición para la colocación de la tarjeta electrónica dentro del robot.

Para sujetar la tarjeta electrónica se utilizan 3 pernos M3 de 6mm de longitud en las siguientes posiciones.

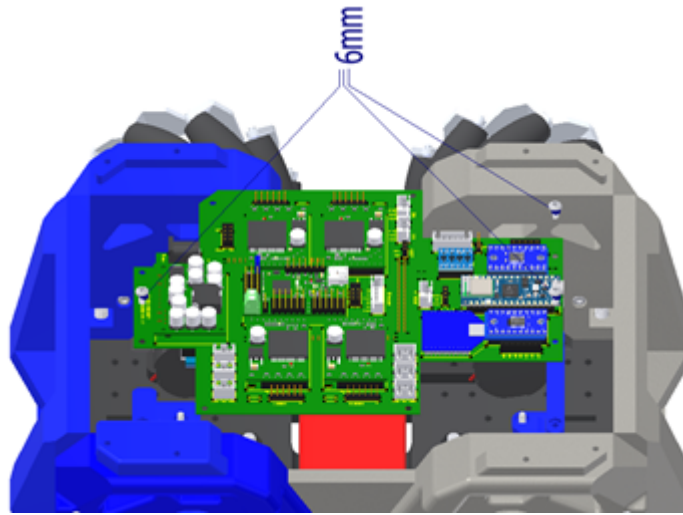


Figura 100: Ubicación de los pernos para sujetar la tarjeta electrónica.

El robot debe quedar de la siguiente manera, con la batería y tarjeta electrónica ya colocadas.

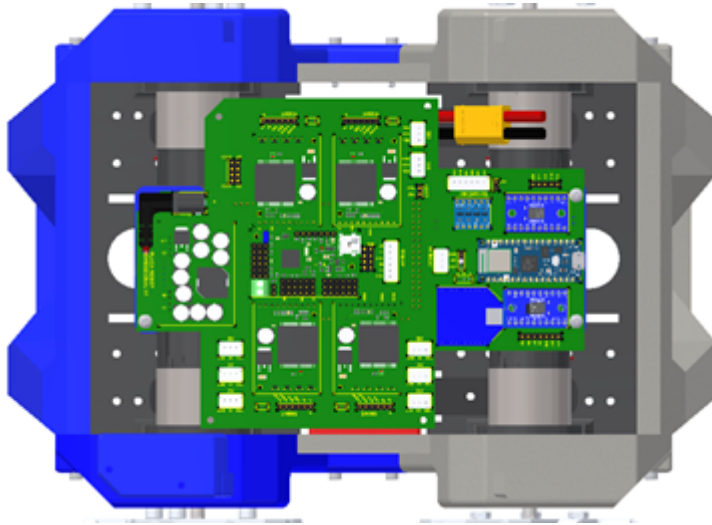


Figura 101: Vista de la batería y tarjeta electrónica dentro del robot.

4.2.3. Conexión de motores y batería

Se conectan los cables de los motores en las siguientes ubicaciones de la placa

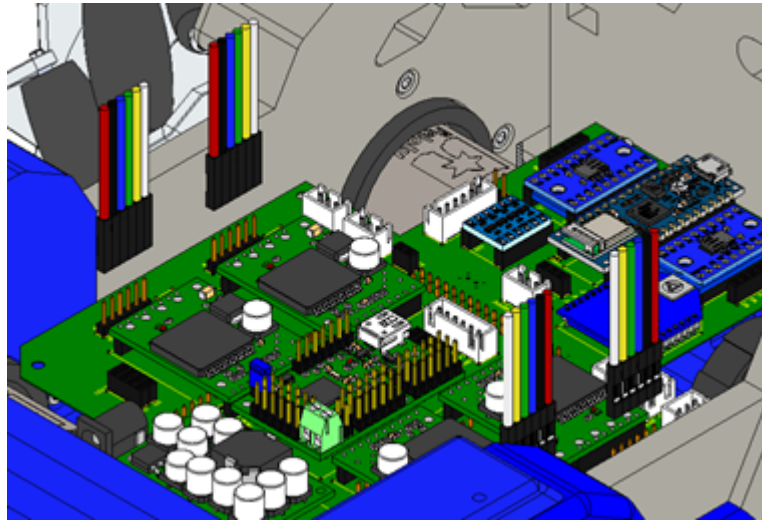


Figura 102: Conexión de los cables para los cuatro motores del robot.

Los cables deben conectarse de tal forma que el cable rojo este a la misma altura que la viñeta.

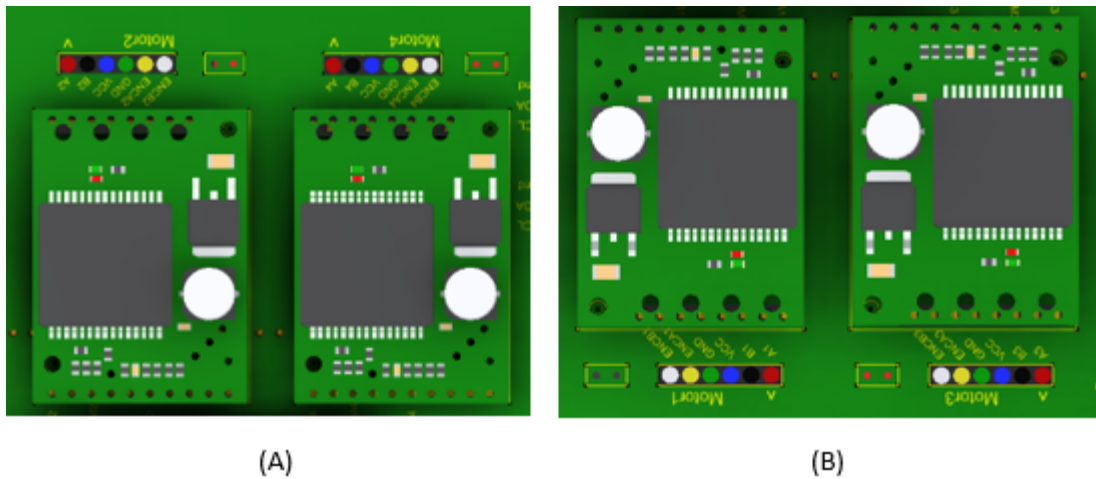


Figura 103: Orientación Motor 2 y 4 (A), Orientación Motor 1 y 3 (B).

La batería se conecta con el conector amarillo ubicado en la tapa de acrílico de la siguiente manera:

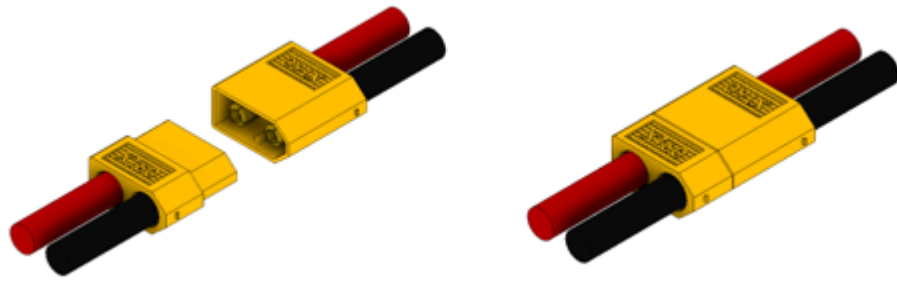


Figura 104: Conexión del cable de alimentación a la batería.

Después se conecta en la tarjeta electrónica el conector de alimentación ubicado en la tapa de acrílico, de la siguiente manera:

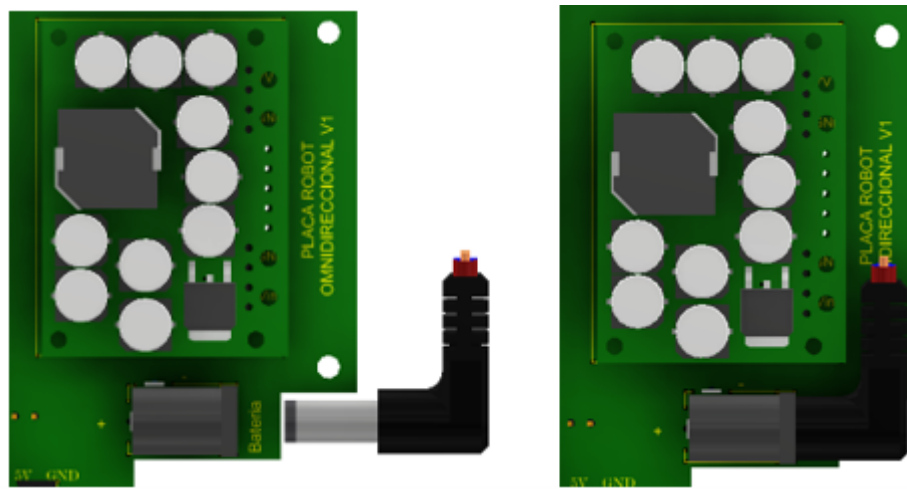


Figura 105: Conexión del conector de alimentación a la tarjeta electrónica.

Las conexiones internas del robot deben quedar de la siguiente forma:

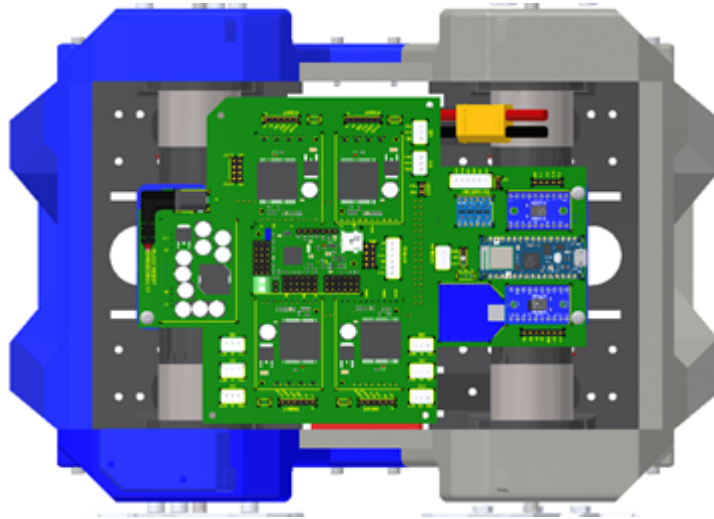


Figura 106: Vista de las conexiones de la batería y los motores en la tarjeta electrónica.

4.3. Programación para movimientos del robot

Con el robot omnidireccional ensamblado junto con la tarjeta electrónica se realizará la prueba de movimientos del robot. Los movimientos del robot son dados por la ecuación de cinemática del robot. estas ecuaciones tienen como parámetros: V_x (la velocidad en X), V_y (la velocidad en Y), y W (la velocidad angular).

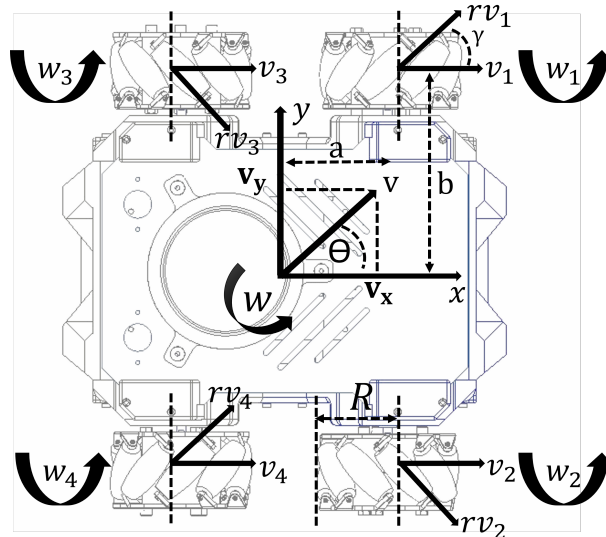


Figura 107: Diagrama de cuerpo.

Cada una de las cuatro ecuaciones dan como resultado la velocidad para la rueda corres-

pendiente. Dependiendo de las velocidades V_x , V_y y W el robot omnidireccional puede llevar a cabo 10 movimientos.

En esta práctica se utilizan las siguientes funciones de la librería “LibreriaRobot.h”

Función	Parámetros de entrada	Descripción
setUpl2C()	Ninguno	Configura la comunicación con el controlador de los motores.
moverRobot(char movimiento)	Movimiento: variable de tipo carácter (letra)	Para mover al robot de acuerdo la letra.

Tabla 22: Funciones para el desplazamiento del robot.

En la siguiente tabla se muestra la letra asignada para cada movimiento del robot:

Grupo A1		Grupo A2		Grupo A3	
Movimiento	Letra	Movimiento	Letra	Movimiento	Letra
Detenerse	x	Adelante izquierda	q	Giro horario	t
Adelante	w	Adelante derecha	e	Giro antihorario	r
Atrás	s	Atrás izquierda	z		
Izquierda	a	Atrás derecha	c		
Derecha	d				

Tabla 23: Letras asignadas al movimiento del robot.

Antes de iniciar la programación se conecta el cable micro USB al microcontrolador Arduino nano 33.

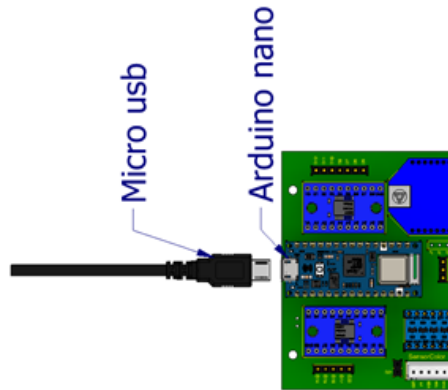


Figura 108: Conexión del cable micro USB al microcontrolador Arduino nano 33.

Primero se probarán los movimientos en el siguiente orden: grupo 1, grupo 2, grupo 3. Con los siguientes códigos cargados por separado.

Primer código	Segundo código	Tercer código
<pre> #include "LibreriaRobot.h" void setup() { Serial.begin(9600); setUpI2C() } void loop() { char adelante = 'w'; char atras = 's'; char izquierda = 'a'; char derecha = 'd'; char detenerse = 'x'; moverRobot(adelante); delay(1000); moverRobot(atras); delay(1000); moverRobot(izquierda); delay(1000); moverRobot(derecha); delay(1000); moverRobot(detenerse); delay(1000); } </pre>	<pre> #include "LibreriaRobot.h" void setup() { Serial.begin(9600); setUpI2C() } void loop() { char adelante_derecha = 'e'; char adelante_izquierda = 'q'; char atras_izquierda = 'z'; char atras_derecha = 'c'; char detenerse = 'x'; moverRobot(adelante_derecha); delay(1000); moverRobot(adelante_izquierda); delay(1000); moverRobot(atras_izquierda); delay(1000); moverRobot(atras_derecha); delay(1000); moverRobot(detenerse); delay(1000); } </pre>	<pre> #include "LibreriaRobot.h" void setup() { Serial.begin(9600); setUpI2C() } void loop() { char giro_horario = 't'; char giro_antohorario = 'r'; char detenerse = 'x'; moverRobot(giro_horario); delay(1000); moverRobot(detenerse); delay(1000); moverRobot(giro_antohorario); delay(1000); moverRobot(detenerse); delay(1000); } </pre>

Tabla 24: Códigos para movimientos.

4.4. Módulo del efector final, ensamble y programación

El objetivo de esta práctica es probar los movimientos del gripper para abrir/cerrar y, subir y bajar el efector final. Primero se instalará el módulo en la parte frontal del robot (frente azul) siguiendo estos pasos:

4.4.1. Instalación del módulo del efector final

Primero se instalará el módulo en la parte frontal del robot (frente azul). Como en las siguientes Figuras.

Piezas		Cantidad
Módulo del efector final		1

Perno	Longitud	Cantidad
M6	20mm	2

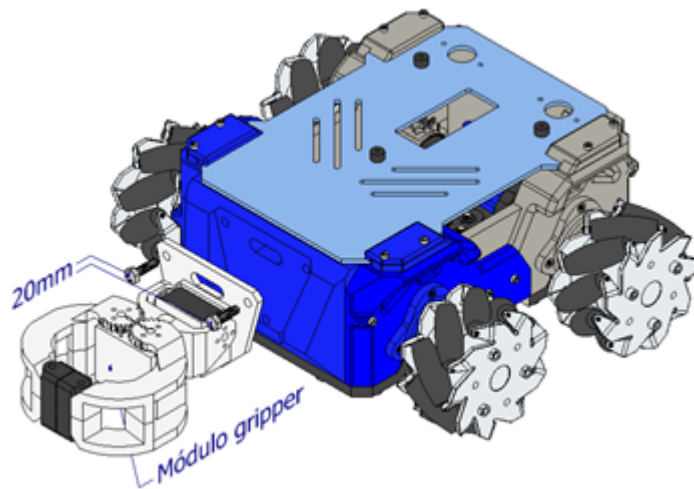


Figura 109: Módulo del efector final con inserción de pernos.

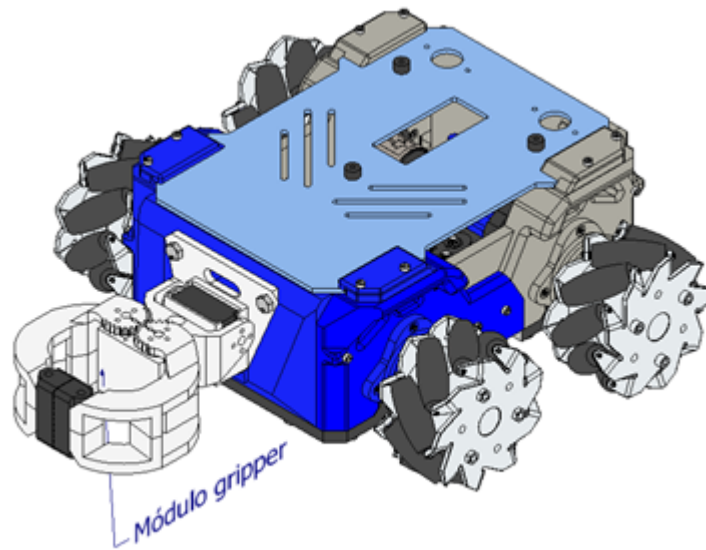


Figura 110: Módulo del efector final acoplado al cuerpo del robot

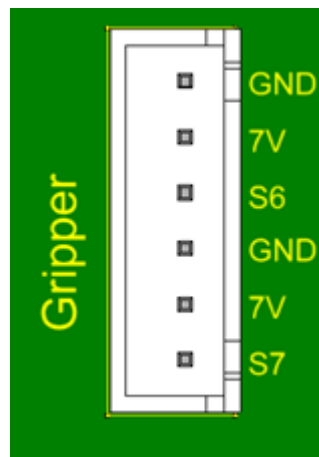


Figura 111: Conector del módulo del efector final.

4.4.2. Programación del módulo del efector final

Funciones para utilizar el módulo del efector final.

Función	Parámetros de entrada	Descripción
gripperAbrir()	Ninguno	Función para abrir el gripper
gripperCerrar()	Ninguno	Función para cerrar el gripper
gripperSubir(int nivel)	Nivel: variable tipo int (entero) para definir cuantos niveles sube el gripper	Función para subir el gripper
grippeBajar(int nivel)	Nivel: variable tipo int (entero) para definir cuantos niveles baja el gripper	Función para bajar el gripper

Tabla 25: Funciones para uso del módulo gripper.

El código para el Arduino nano 33 es el siguiente:

```

#include "LibreriaRobot.h"

void setup() {
  Serial.begin(9600);
  setUpI2C()
}

void loop() {
  greaperAbrir();
  delay(2000);
  greaperSubir(2);
  delay(1000);
  greaperCerrar();
  delay(2000);
  greaperBajar(2);
  delay(1000);
}

```

Figura 112: Código de programación módulo del efector final.

4.5. Módulo led, ensamble y programación

El objetivo de esta práctica es probar el módulo led utilizando los valores RGB para cambiar los colores del módulo. Primero se instalará el módulo en la cubierta de acrílico superior del robot siguiendo estos pasos:

4.5.1. Instalación del módulo led

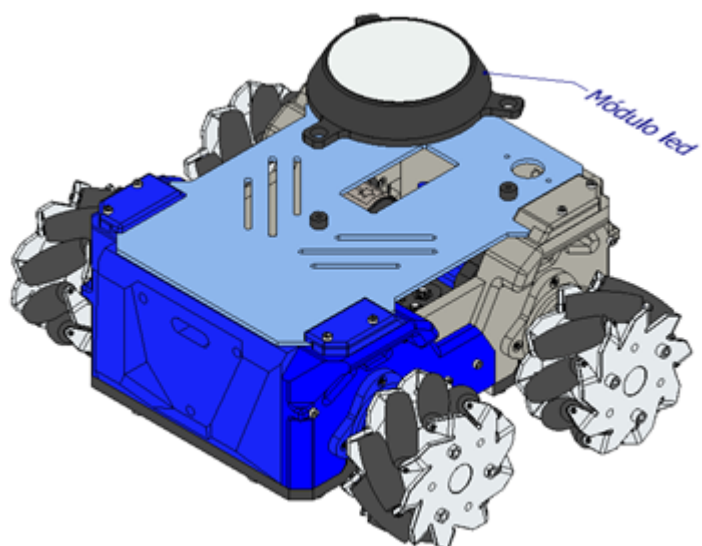


Figura 113: Alineación del módulo led, vista superior.

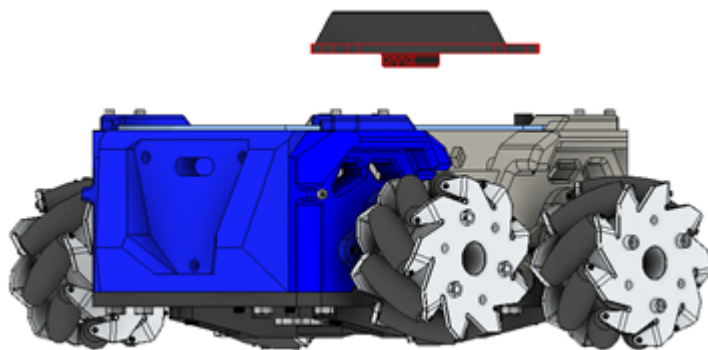


Figura 114: Alineación del módulo led, vista inferior.

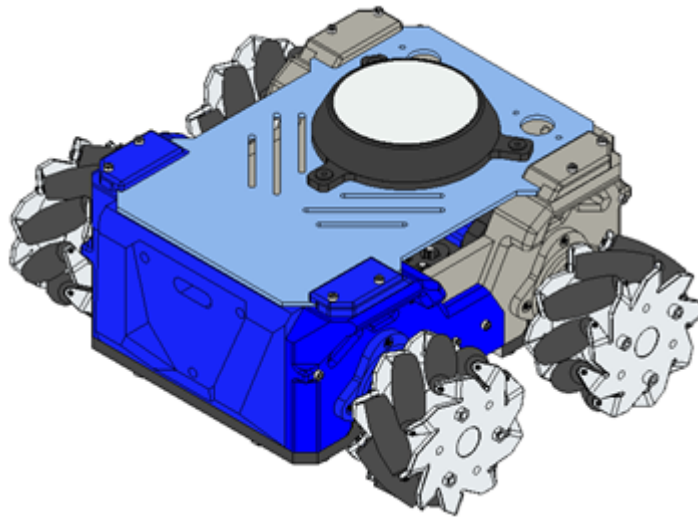


Figura 115: Módulo led instalado.

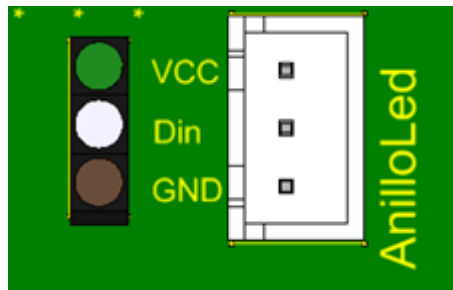


Figura 116: Forma de conexión de los tres cables del módulo, verde (VCC), blanco (Din) y café (GND).

4.5.2. Programación del módulo led

En esta práctica se utilizan las siguientes funciones para utilizar el módulo led.

Función	Parámetros de entrada	Descripción
setUpAnilloLed()	Ninguno	Configura el anillo led para su uso.
setColorAnilloLed(int R, int G, int B)	R: variable tipo int(entero) de 0 a 255 para el color rojo del led. G: variable tipo int(entero) de 0 a 255 para el color verde del led. B: variable tipo int(entero) de 0 a 255 para el color azul del led	Función para definir el color del módulo led en base a los valores RGB.

Tabla 26: Funciones para uso del módulo led.

El código para el Arduino nano 33 es el siguiente:

```
#include "LibreriaRobot.h"

void setup() {
  Serial.begin(9600);
  setUpAnilloLed();
}

void loop() {
  setColorAnilloLed(255,0,0);
  delay(3000);
  setColorAnilloLed(0,255,0);
  delay(3000);
  setColorAnilloLed(0,0,255);
  delay(3000);
}
```

Figura 117: Código de programación módulo led.

4.6. Programación para control por bluetooth

El objetivo de esta práctica es controlar el movimiento y módulo del efector final del robot omnidireccional por medio de una aplicación en un dispositivo móvil. Para esto el módulo del

efector final y el módulo led deben ser instalados en el robot.

En esta práctica se utilizan las siguientes funciones de la librería “LibreriaRobot.h”

Función	Parámetros de entrada	Descripción
setUpBluetooth()	Ninguno	Configura el bluetooth del Arduino nano 33 para la comunicación.
bleConectado()	Ninguno	Enciende el led integrado del Arduino si hay conexión bluetooth.
bleDesconectado()	Ninguno	Apaga el led integrado del Arduino si no hay conexión bluetooth.
gripperRobot(char comando)	Comando: variable tipo carácter (letra) para controlar el módulo gripper.	Función para controlar el módulo gripper por comandos
arcoirisAnilloLed (int tiempo)	Tiempo: variable tipo entero para controlar la velocidad	Función de efecto arcoíris para el módulo led.

Tabla 27: Funciones para comunicación bluetooth.

El código para el microcontrolador Arduino nano 33 es el siguiente:


```
RobotOmnidireccional_ble $ LibreriaRobot.h moduloAnilloLed.h
#include "LibreriaRobot.h"

void setup() {
  Serial.begin(9600);
  setUpI2C();
  setUpBluetooth();
  setUpAnilloLed();
}

void loop() {
  BLEDevice central = BLE.central();
  if (central.connected()) {BleConectado();
    while (central.connected()) {
      char comandoMovimiento = moveChar.value();
      moverRobot(comandoMovimiento);
      char comandoGripper = gripperChar.value();
      gripperRobot(comandoGripper);
      arcoirisAnilloLed(30);
    }
  }
  else{BleDesconectado();}
}
```

Figura 118: Programa para comunicación bluetooth.

5. CAPÍTULO 5: PRUEBAS Y ANÁLISIS DE RESULTADOS

5.1. Armado de la tarjeta electrónica y kit robótico completo

Con la placa impresa y los respectivos conectores designados a cada módulo, se procedió a medir el tiempo que lleva armar la placa de control del robot. Se procedió a simular con los componentes a disposición y el área adecuada tal y como se espera que se lleven a cabo las practicas. Como autores del proyecto el tiempo de ensamble no supera los 2 minutos y 25 segundos, para los usuarios finales se puede trazar un tiempo estimado de 45 minutos, teniendo como factores el manejo de terminología en componentes electrónicos y materia referente a la misma, como también previamente haber leído la sección 4.2 del manual práctico realizado en este proyecto de titulación. Cabe recalcar que los usuarios a los que va dirigida son estudiantes de colegios con poco o nulo conocimiento de la materia y rondan las edades de entre 16 a 18 años.



Figura 119: Armado de la tarjeta electrónica.

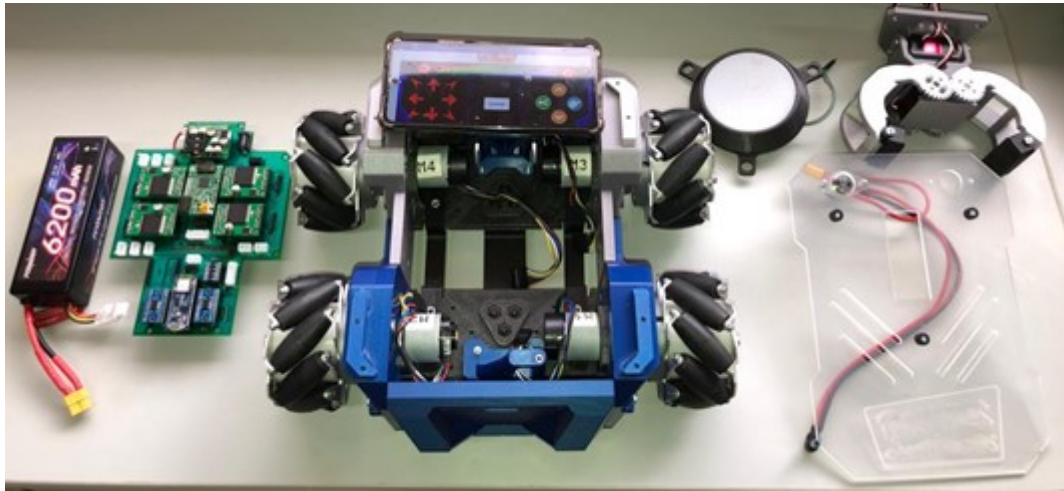


Figura 120: Kit robótico completo.

Respecto al tiempo de armado de chasis y módulos del robot rondan la hora de trabajo previamente leída la sección 4.1 del manual práctico. Teniendo todos los componentes disponibles como en la Figura 120, el tiempo que lleva ensamblar por completo el robot ronda los 5 a 10 minutos.

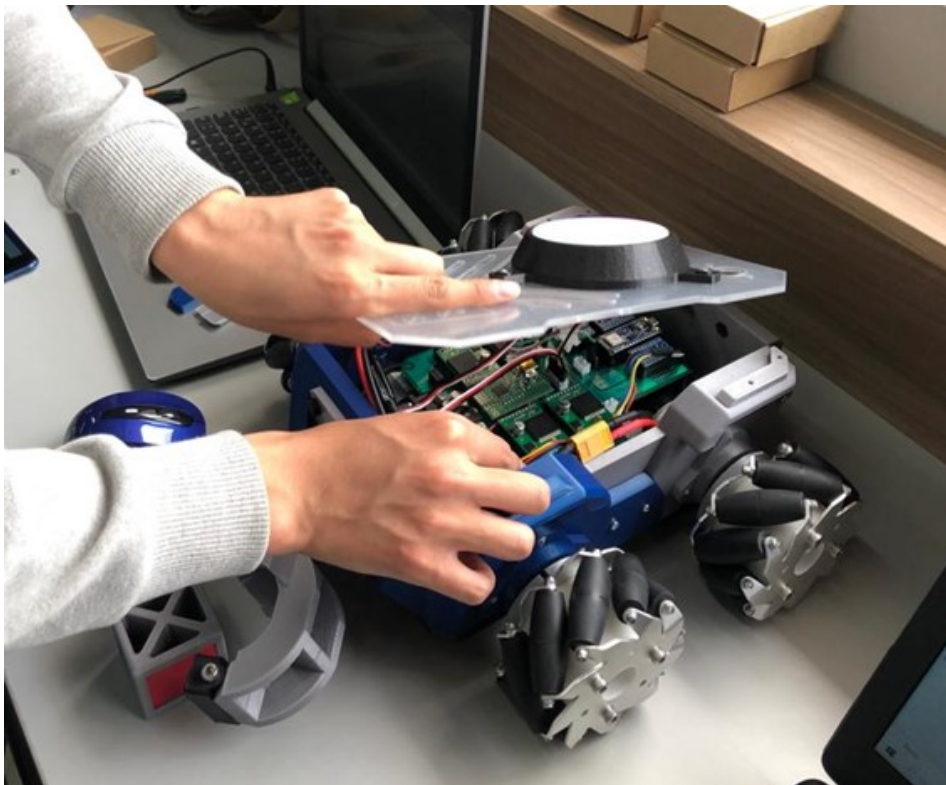


Figura 121: Ensamble del kit robótico.

El kit completamente armado se observa en la Figura 122.



Figura 122: Kit robótico armado por completo.

5.2. Aplicativo móvil

La aplicación móvil fue creada en línea en la plataforma MIT APP inventor para sistemas operativos Android, al ser un servicio gratuito no se puede subir a plataformas de descargas como playstore o app store. Su descarga debe realizarse de la misma plataforma.

Previamente en la Figura 33, se detalla los caracteres a los cuales están asociados los distintos botones con los que cuenta el aplicativo. En la Figura 123, se observa la app del control en un dispositivo móvil y en la Figura 120, se preseta el funcionamiento del robot desde la app del control.

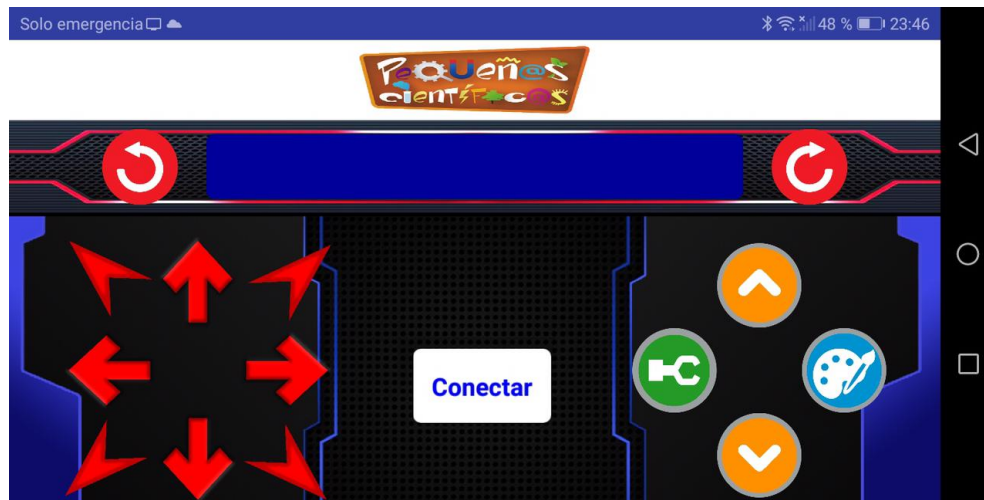


Figura 123: Aplicativo móvil.

5.3. Movimiento y módulo led

Una de las características llamativas del robot es su atractiva iluminación superior misma que puede ser controlado por la aplicación móvil y cambiar sus colores a decisión del usuario.

Además, puede ser utiliza al mismo tiempo que el robot se desplaza en su entorno, tal y como se puede ver en la Figura 125, donde el robot se encuentra moviendo en línea recta y empujando una caja con un rollo de filamento PLA de 1kg de peso.



Figura 124: Robot en movimiento empujando 1kg de peso.

A termino de estudió se decidió aumentar el peso siendo el siguiente obstáculo una maceta

de madera de 1.8 kg de peso, en donde cuyo resultado fue el previsto, el robot gracias a su controlador PI programado consiguió aumentar la fuerza de las ruedas progresivamente de tal forma que consiguió mover el peso.



Figura 125: Robot en movimiento empujando 1.8kg de peso.

Se observa que todos los motores se establecen en diferentes puntos de consigna, dando como puntos de referencias 115, 105, 95,85 rpms. La simulación se la realizó con el plotter de Arduino Ide.

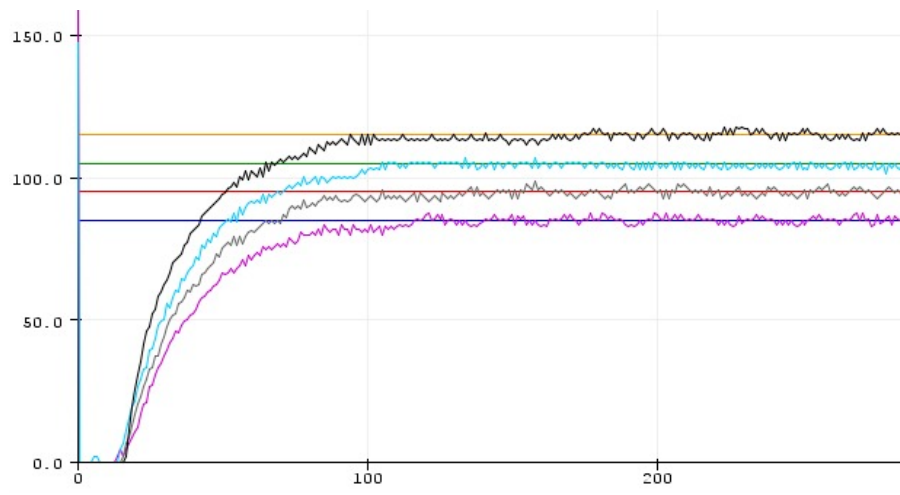


Figura 126: Valores de los motores con el controlador vs puntos de referencia.

5.4. Módulo del efector final

Este módulo cuenta con un grado de libertad por lo que tiene que valerse del movimiento del cuerpo completo para poder posicionarse para levantar el objeto y llevarlo a una altura máxima de 12 cm del suelo y poder transportarlo. Su movimiento de cierre y apertura es directo mientras que el movimiento de arriba abajo puede ser regulado conforme se presionen los botones designados en el control de la aplicación.



Figura 127: Módulo del efector final en funcionamiento

6. CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

La metodología STEM se caracteriza por el empleo de herramientas didácticas en la enseñanza, para ello es necesario pensar en las necesidades del usuario final y cómo será el comportamiento del dispositivo tecnológico empleado para el aprendizaje. Stem se puede entender como un método que socializa ciencias en busca de impartir y ganar conocimiento. El diseño del kit robótico presentado está basado en la temática de las instalaciones del proyecto Pequeñ@s Científic@s, su estilo asemeja a los trazos y líneas marcadas que son características de un computador gamer.

En el diseño se debía tener presente el área de impresión que tenían las impresoras puesto que no podíamos pasar las medidas presentadas en la sección 2.1, por ello se decidió estructurar el chasis del robot de forma modular y cuyo efecto fue de beneficio pues se consiguió cumplir con parte de los objetivos planteados. Otra ventaja del diseño y la fabricación del robot es que sus piezas pueden ser rediseñadas por separado sin la necesidad de un rediseño completo del kit robótico.

Se consiguió realizar la escritura de la librería “Libreríarobot.h” en donde se concentró la mayor parte de líneas de programación para realizar las comunicaciones bluetooth, I2C y el control de los leds neopixel. El mayor reto fue escribir la programación para que el código se fluido y no colapsen ninguna de las comunicaciones. Para el firmware también se presentó la misma situación debido a que este también se utilizan los puertos de comunicación serial, I2C y también se manejan los pines de interrupciones externas.

El manual de prácticas se diseñó para un taller piloto, donde se empleó gran cantidad de graficas a fin de guiar en todo momento al usuario final y se redujo la teoría para mantener la atención en las cosas prácticas, como; el ensamble del robot y la interacción con la programación del robot.

Durante la fase de pruebas se comprobó la firmeza de la estructura del robot mediante el ensamble de las piezas del robot y de la tarjeta electrónica, así como el montaje de todos los componentes del kit robótico.

6.2. Recomendaciones

Para el diseño de las piezas se recomienda realizar empalmes en las esquinas exteriores, para evitar lesiones al momento de manipular las piezas. De igual forma previo al diseño y la impresión se recomienda realizar impresiones de prueba con medidas que se estén utilizando en el diseño con el fin de conocer las tolerancias de la impresora.

Se recomienda que cada nuevo programa sea probado fuera de la librería y en ningún caso se utilicen funciones de Arduino para retener el avance del código como la función “delay” ya que dependiendo de las funciones que se utilicen de la “Libreriarobot.h” se pueden obstaculizar las comunicaciones.

Para imprimir las piezas se recomienda que el ambiente de impresión no este propenso a cambios bruscos de clima puesto que esto puede afectar a las medidas de la pieza final. También se recomienda utilizar fijador de cabello extrafuerte marca Biolance el cual fue utilizado en este proyecto. También se debe tener en cuenta que la mayoría de las piezas diseñadas en este proyecto superaron las 6 horas de impresión por lo que se recomienda distribuir las horas de impresión y utilizar de la mejor manera posible el área disponible de impresión. Se recomienda revisar los datasheet de los componentes electrónicos en busca de la distribución de los pines la potencia de consumo para diseñar la tarjeta electrónica y evitar daños futuros.

7. TRABAJO FUTURO

- Se propone implementar más módulos utilizando los pines digitales, analógicos y pwm disponibles en la tarjeta del robot omnidireccional. Los módulos pueden incorporar sensores para aumentar la percepción del entorno del robot.
- Se debe implementar el manual práctico en un taller piloto para rediseñar las prácticas de acuerdo con el nivel de aceptación y comprensión de la materia por parte de los estudiantes.
- Se propone incorporar el sistema operativo ROS en el robot omnidireccional, esto puede hacerse cambiando el Arduino nano 33Ble por un Arduino nano 33Iot y modificando la programación para acoplar con la librería de ROS y Arduino.

Referencias

- Beiroa, R. (2020). Las posibilidades de arduino en el mundo profesional. *Disponible en:* <https://www.vermislabs.com/las-posibilidades-de-arduino-en-el-mundo-profesional/>.
- bricoGeek (2020). Rueda mecanum 65mm. *Disponible en:* <https://tienda.bricogeek.com/ruedas-robotica/1571-rueda-mecanum-65mm-izquierda.html>.
- CADistas (2020). Formulas para generar un engranaje con autodesk inventor. *Disponible en:* <https://drive.google.com/file/d/1ucXBuMVXQoHC6kOsubveBgsxb7SkfRO/view>.
- Cáceres, N. M. (2019). *Educación STEM/STEAM. Apuesta hacia la formación, impacto y proyección de seres críticos*. Servando Garces.
- de educación, M. (2021). Estadísticas educativas-datos abiertos. *Disponible en:* <https://educacion.gob.ec/datos-abiertos/>.
- Elena Durán, D. G. (2010). Diseño y construcción de un kit didáctico de experimentación científica-tecnológica para motivar a niños y niñas entre diez y once años por la ciencia y tecnología. Master tesis, Universidad de Politecnica Salesiana, Cuenca Ecuador.
- Ioan Doroftei, V. G. y Spinu, V. (2007). Omnidirectional mobile robot – design and implementation. *Disponible en:* https://cdn.intechopen.com/pdfs/465/InTech-Omnidirectional_mobile_robot_design_and_implementation.pdf.
- Jhonny Ortega, L. Y. (2017). Construcción de un robot móvil híbrido omnidireccional. *Disponible en:* [https://dspace.ups.edu.ec/bitstream/123456789/14048/1/UPS %20-%20ST003073.pdf](https://dspace.ups.edu.ec/bitstream/123456789/14048/1/UPS%20-%20ST003073.pdf).
- Jorge Noboa, C. B. (2016). Laboratorios virtuales: una alternativa para mejorar el rendimiento de los estudiantes y la optimización de recursos económicos. *Disponible en:* <https://revistas.uide.edu.ec/index.php/innova/article/view/79/105>.
- katsuhiko ogata (1970). *Ingeniería de Control Moderna*. Pearson.
- Kenjiro Tadakuma, Riichiro Tadakuma, K. N. K. Y. K. I. (2008). Crawler mechanism with circular section to realize a sideling motion. *Disponible en:* [http://www.robotpark.com/academy/VP/11076-Crawler %20Mechanism %20with %20Circular %20Section %20IROS08_1351MS.pdf](http://www.robotpark.com/academy/VP/11076-Crawler%20Mechanism%20with%20Circular%20Section%20IROS08_1351MS.pdf).

- Llamas, L. (2018). Robot con omni wheel controlado por arduino. *Disponible en:* <https://www.luisllamas.es/robot-con-omni-wheel-controlado-por-arduino/>.
- MINTEL (2021). Agenda digital 2021-2022. *Disponible en:* <https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/05/Agenda-Digital-del-Ecuador-2021-2022-222-comprimido.pdf>.
- osoyo (2019). Mecanum wheel robotic(arduino mega2560)-lesson1 assembling the car (model 2019016600). *Disponible en:* <https://osoyoo.com/2019/11/13/mecanum-omni-wheel-robotic-kit-v1-for-arduino-mega2560-lesson-14/>.
- Rosmari García Mejía, C. G. V. (2020). Metodología steam y su uso en matemáticas para estudiantes de bachillerato en tiempos de pandemia covid-19. *Disponible en:* <https://dominiodelasciencias.com/ojs/index.php/es/article/view/1212/1973>.
- Schwab, K. (2016). *La cuarta revolución industrial*. Debate.
- Tzafestas, S. G. (2014). *Introduction to Mobile Robot Control*. Elsevier.
- Víctor Lopez, Digna Couso, C. S. (2020). Educaciónstem en y para un mundo digital: el papel de lasherramientas digitales en el desempeño de prácticas científicas, ingenieriles y matemáticas. *Disponible en:* <https://revistas.um.es/red/article/view/410011>.
- Yonnhatan García Cartagena, David Reyes González, F. B. O. (2017). Actividades stem en la formación inicial de profesores: nuevos enfoques didácticos para los desafíos del siglo xxi. *Disponible en:* <https://dialnet.unirioja.es/servlet/articulo?codigo=6212470>.

ANEXOS

Apéndice A: Lista de materiales usados.

Cantidad	Descripción	Precio unitario	Total
1	Arduino nano 33	\$ 33,00	\$ 33,00
1	Arduino Mega 2560	\$ 38,00	\$ 38,00
2	Módulo de conversión lógica 8 canales TXS0108E	\$ 8,49	\$ 16,98
2	Módulo de conversión lógica 4 canales	\$ 8,00	\$ 16,00
1	Anillo de 24 leds RGB neopixel	\$ 17,35	\$ 17,35
1	XBEE serie 2	\$ 28,95	\$ 28,95
2	Servos de doble eje LD-220MG	\$ 21,99	\$ 43,98
1	Sensor de color RGB tcs34725	\$ 10,09	\$ 10,09
1	Batería 11,1v 6200mAh	\$ 75,99	\$ 75,99
4	Motores DC con encoders	\$ 34,95	\$ 138,80
4	Driver motor VNH5019	\$ 54,95	\$ 219,80
1	Regulador voltaje 5v 3,2A	\$ 19,95	\$ 19,95
1	Regulador voltaje 3v 3,6A	\$ 19,95	\$ 19,95
1	Mini maestro 18 canales	\$ 37,95	\$ 37,95
1	Otros	-	\$ 250,00
Total			\$ 966,79

Tabla 28: Materiales utilizados en el proyecto.

Perno	Longitud	Cantidad
M6	20mm	12
M6	25mm	8
M4	20mm	4
M3	20mm	12
M3	18mm	4
M3	8mm	8
M3	30mm	4
M3	16mm	4
M3	6mm	3

Tabla 29: Lista de pernos empleados para el armado del robot.

Apéndice B: Armado del robot.

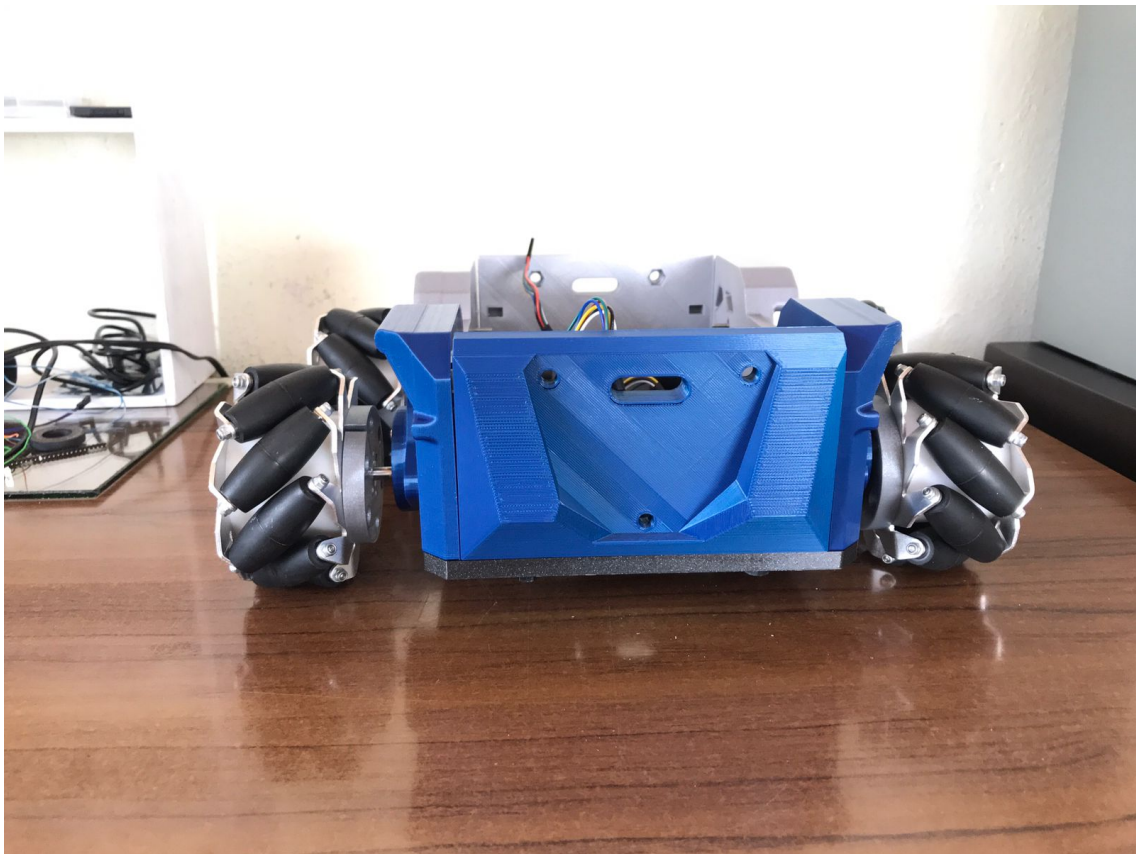


Figura 128: Pruebas de ensamble.

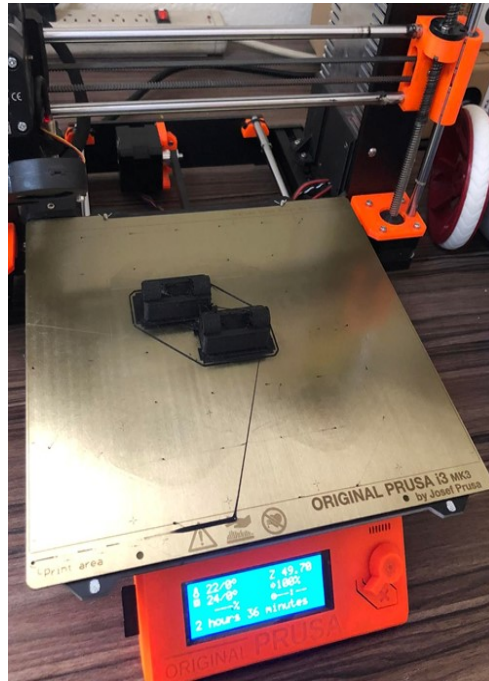


Figura 129: Impresión en material PETG

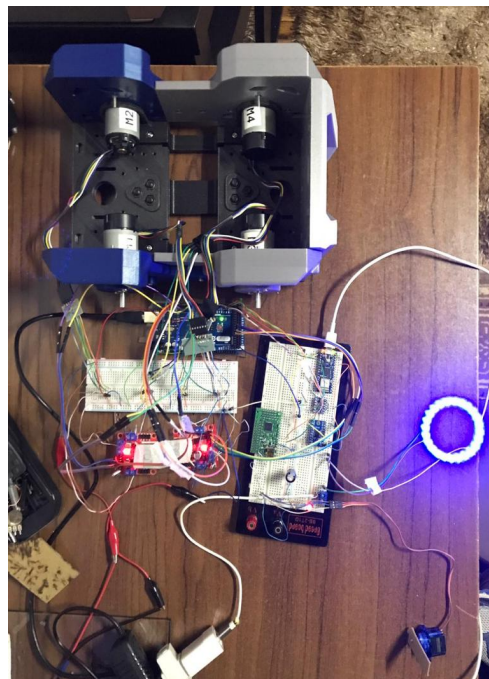


Figura 130: Armado del circuito eléctrico en el protoboard

Apéndice C: Módulos y cuerpo del robot.

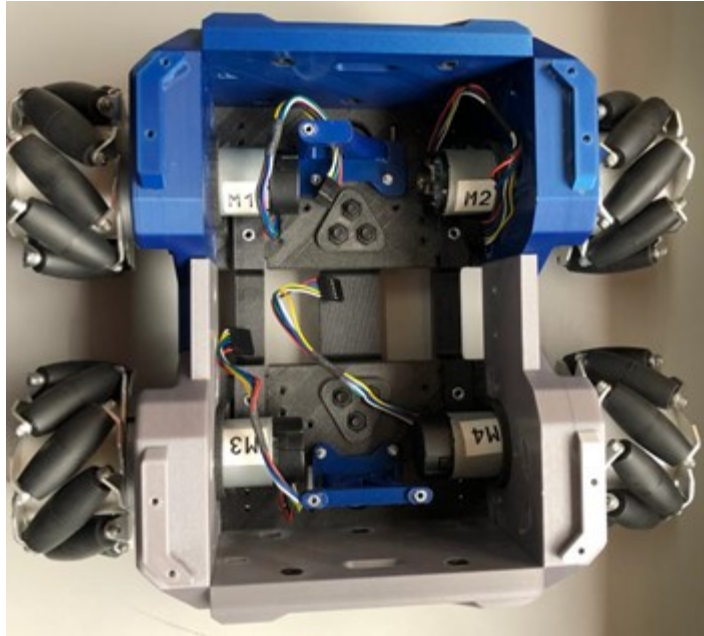


Figura 131: Chasis del robot

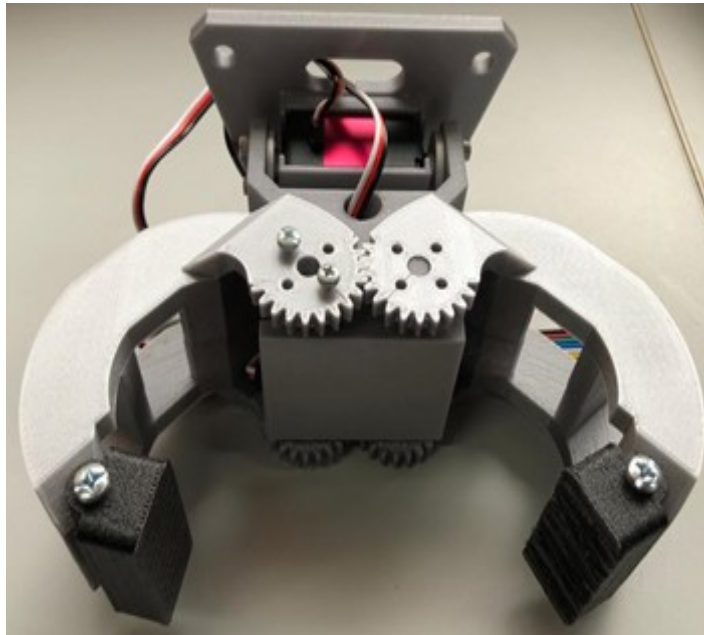


Figura 132: Módulo del efector final



Figura 133: Módulo led apagado



Figura 134: Módulo led encendido

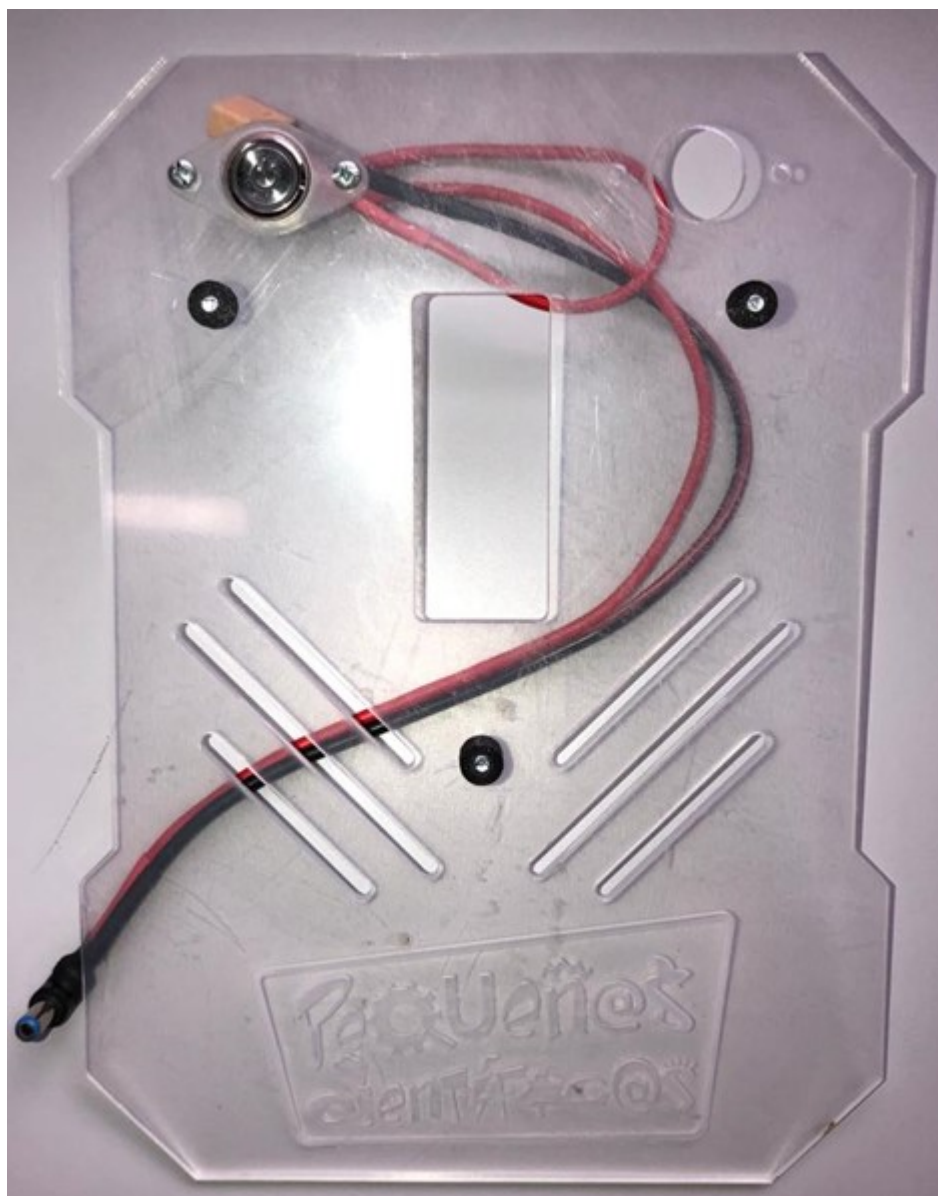
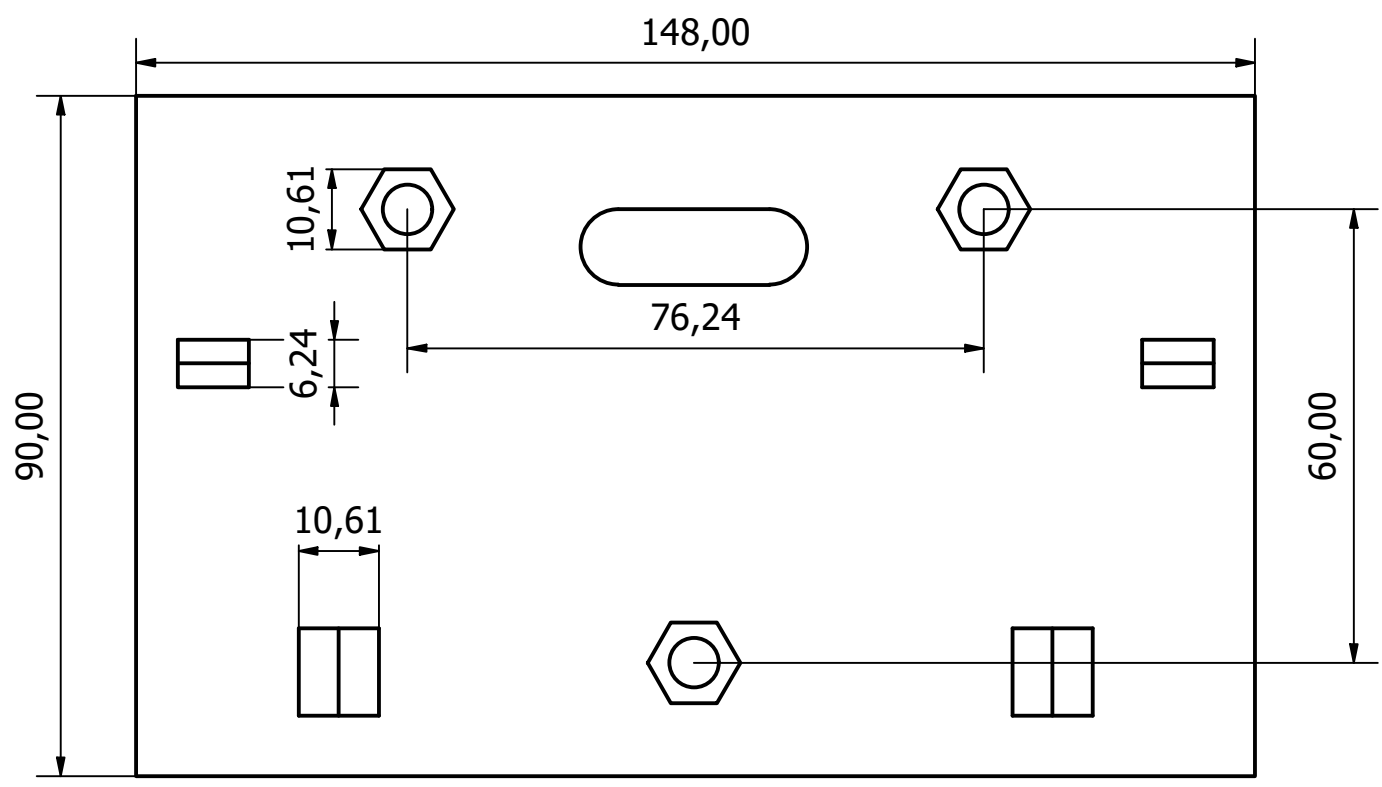
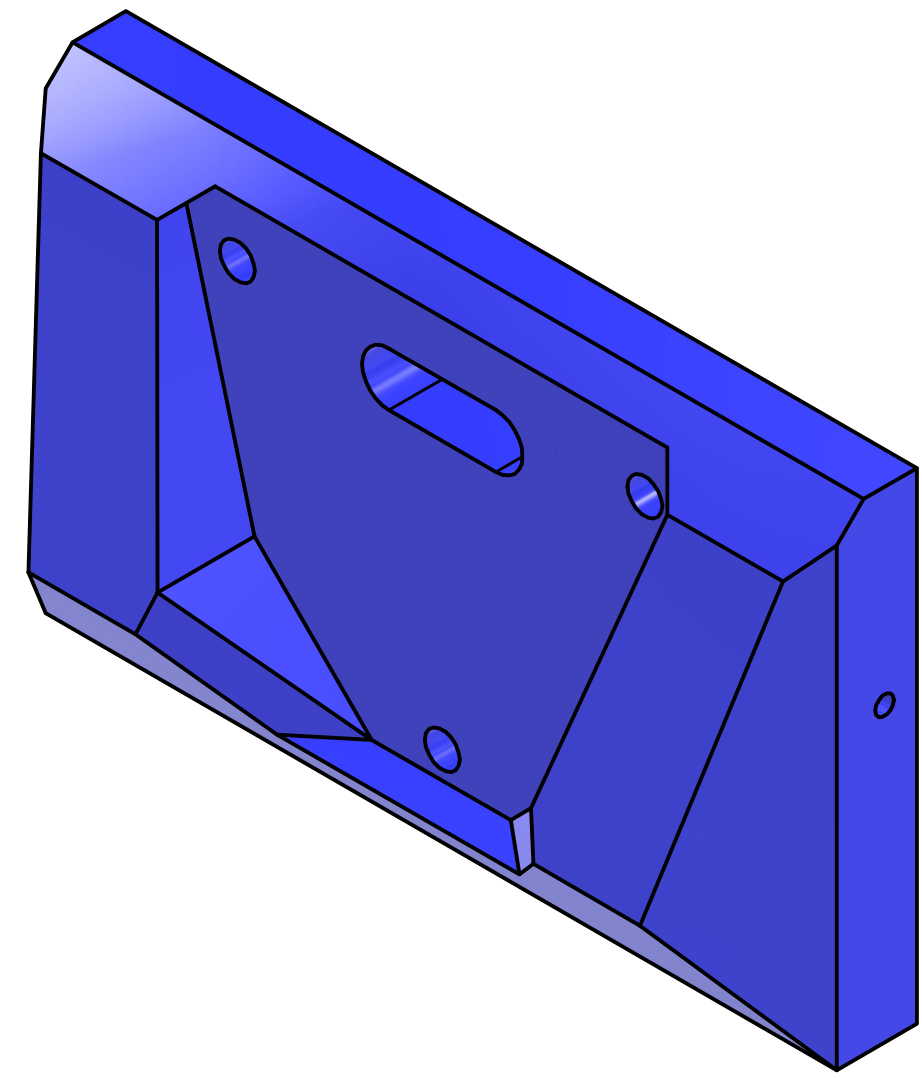
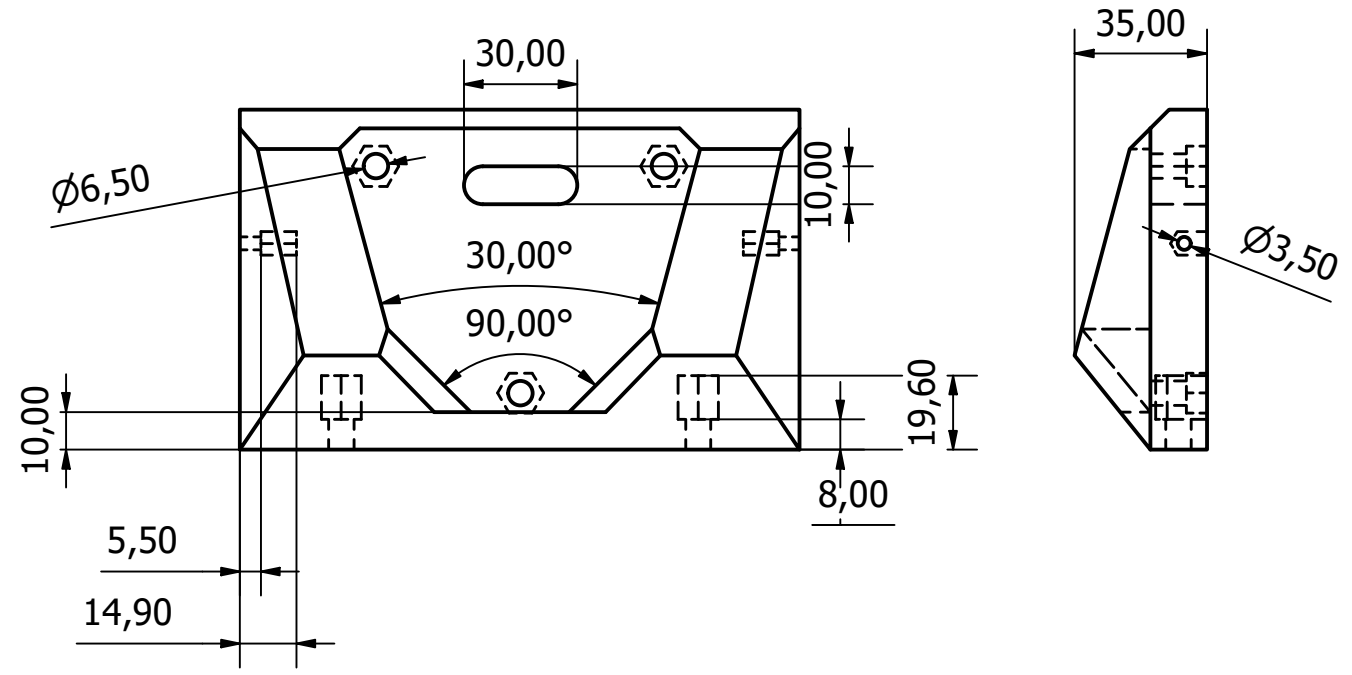
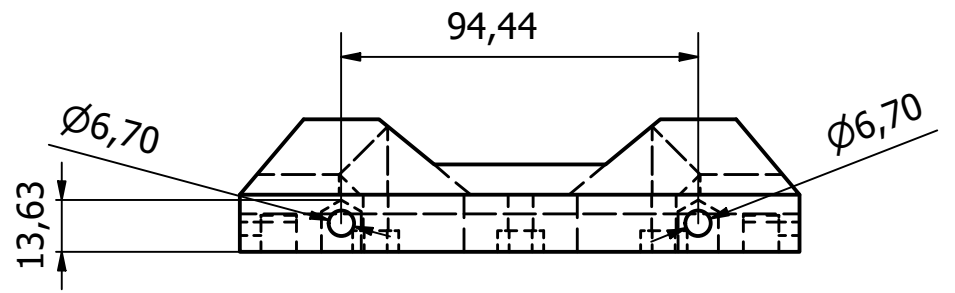
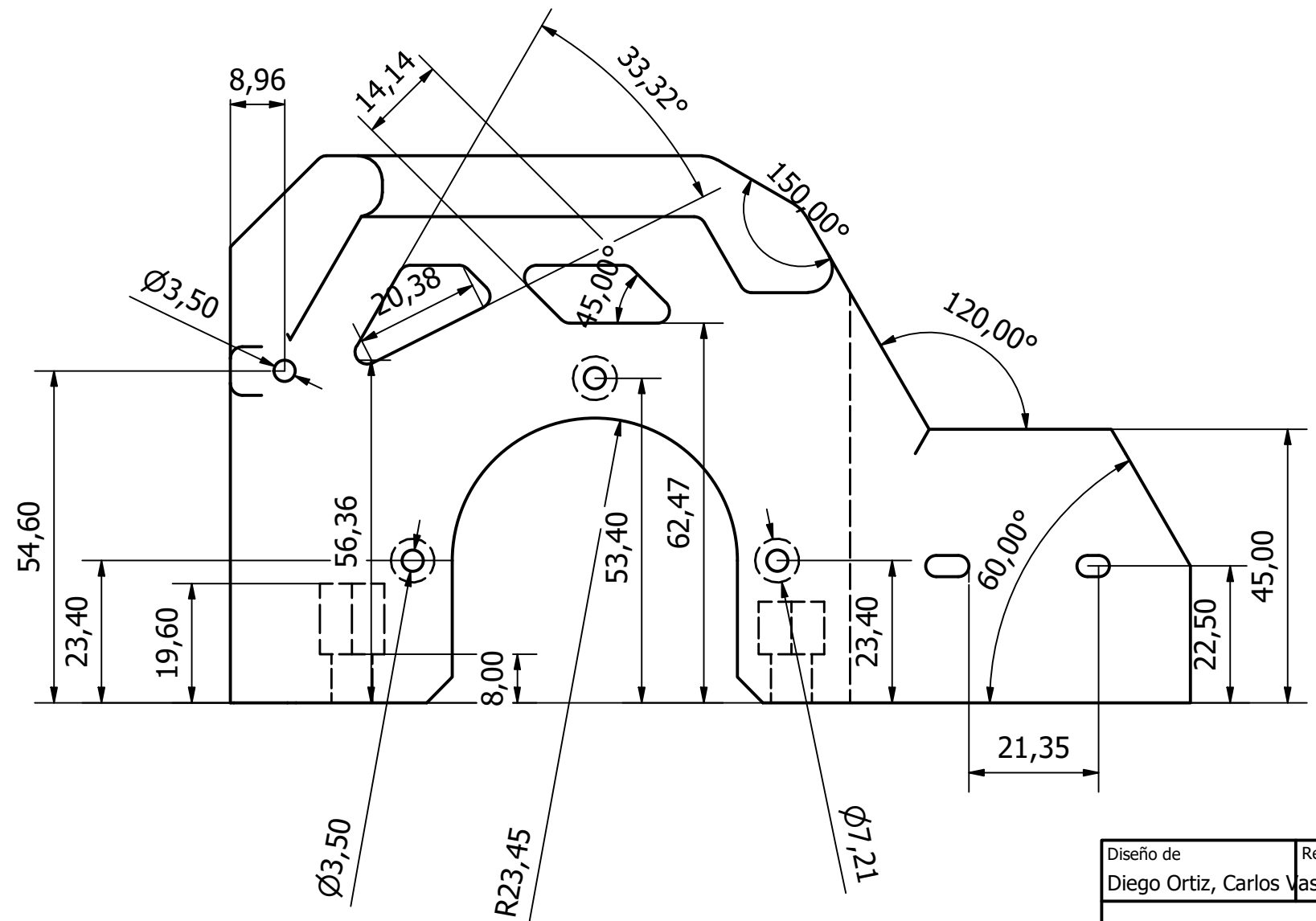
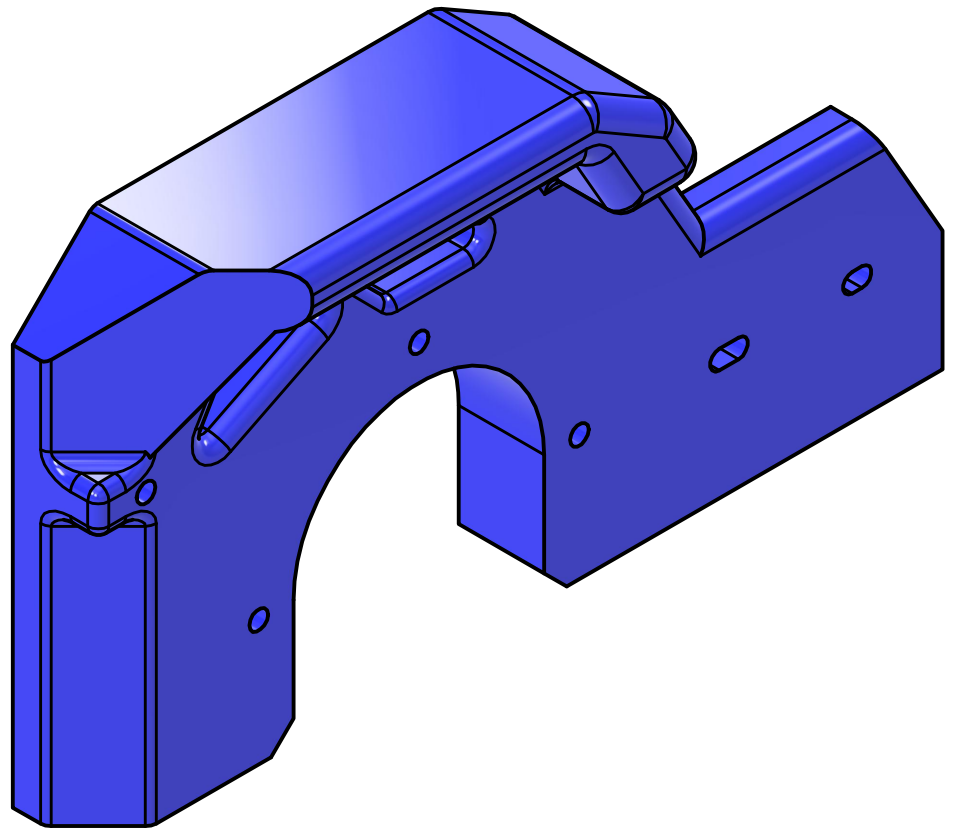
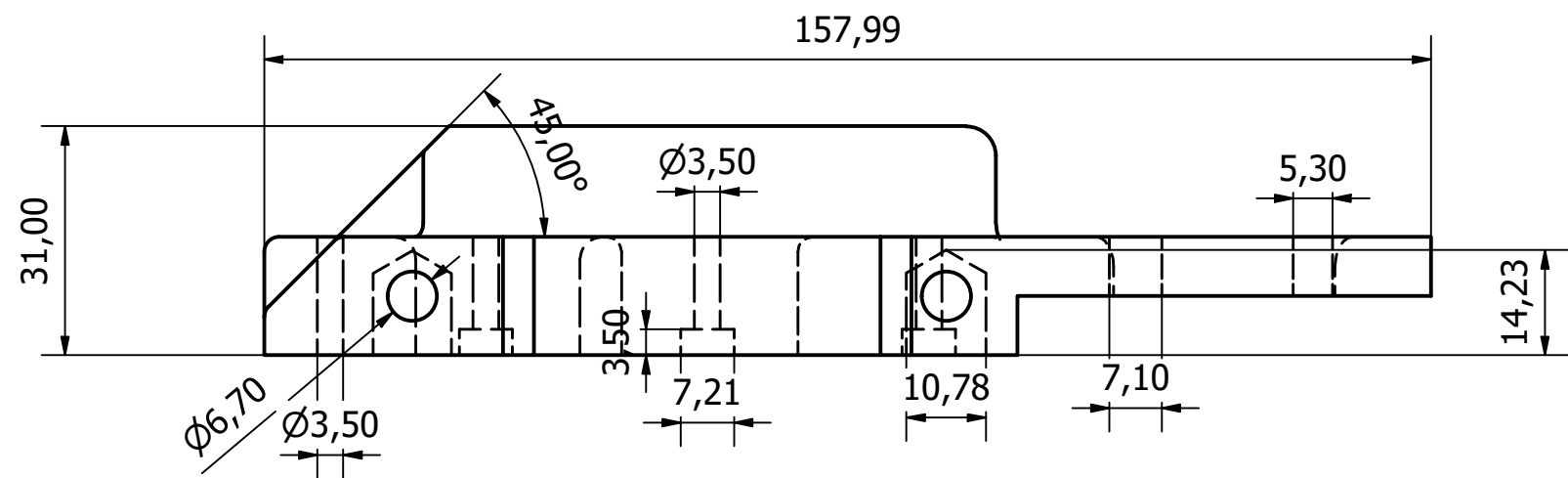


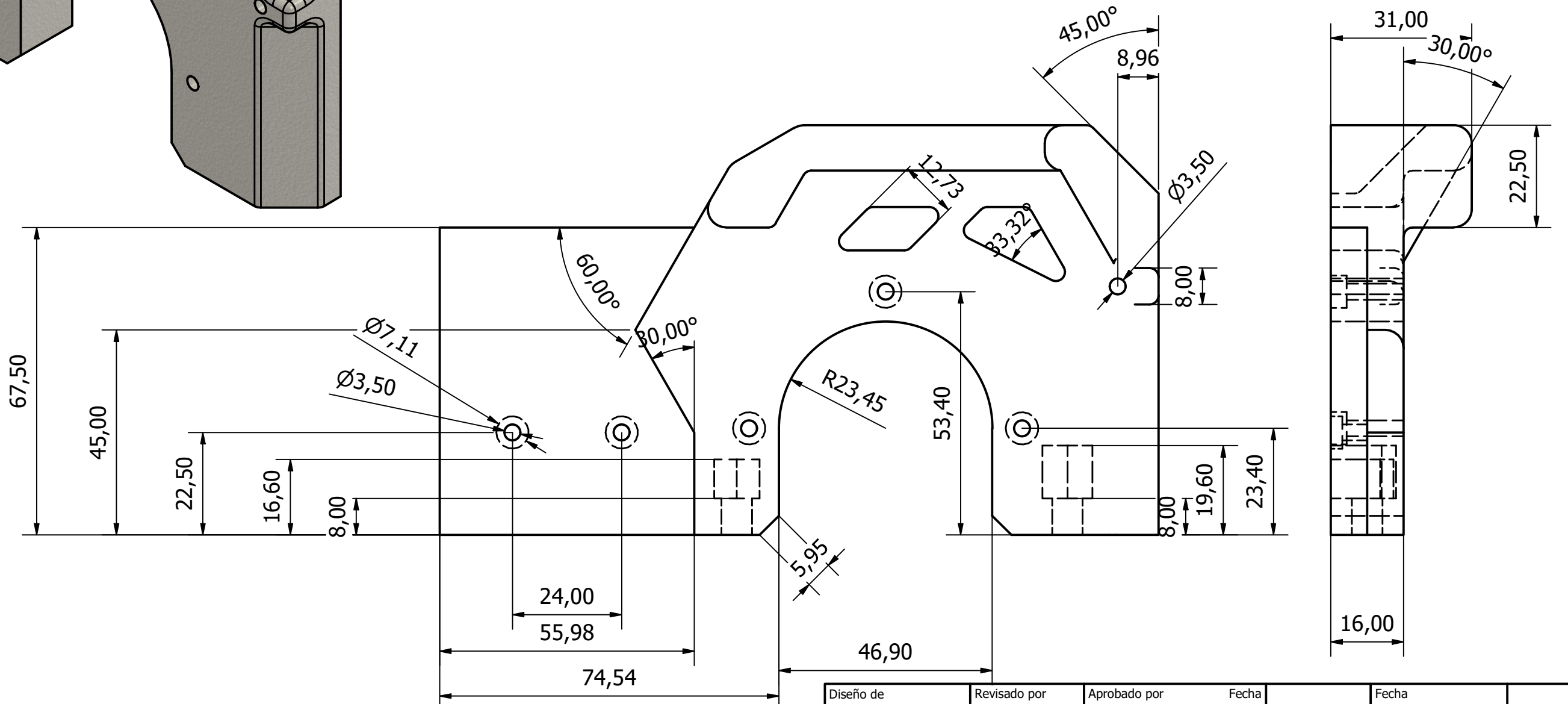
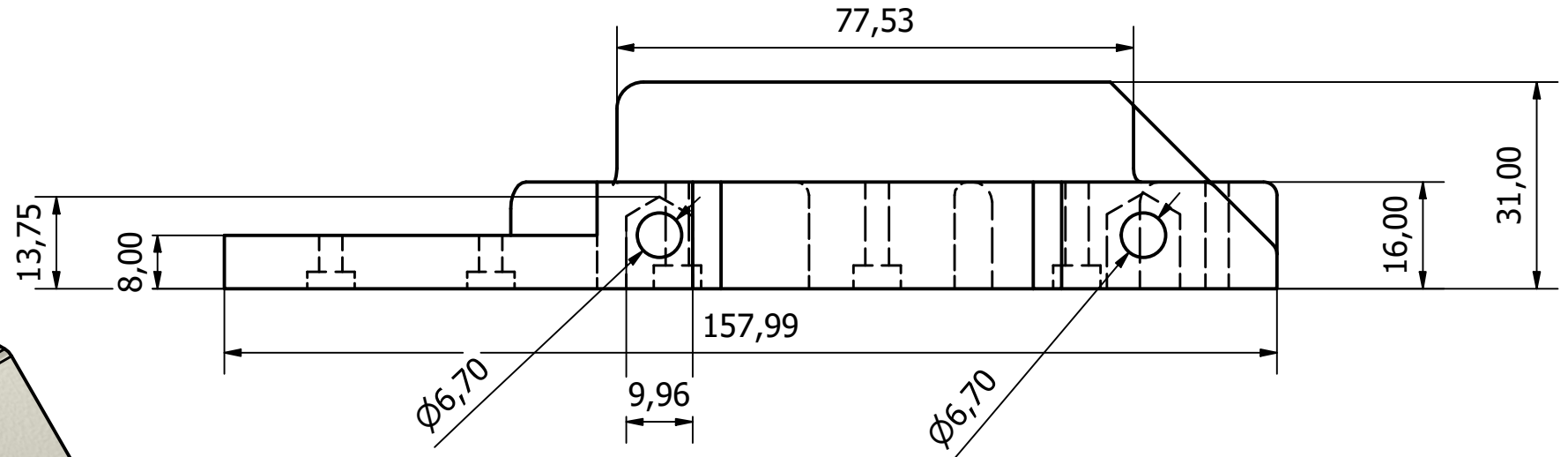
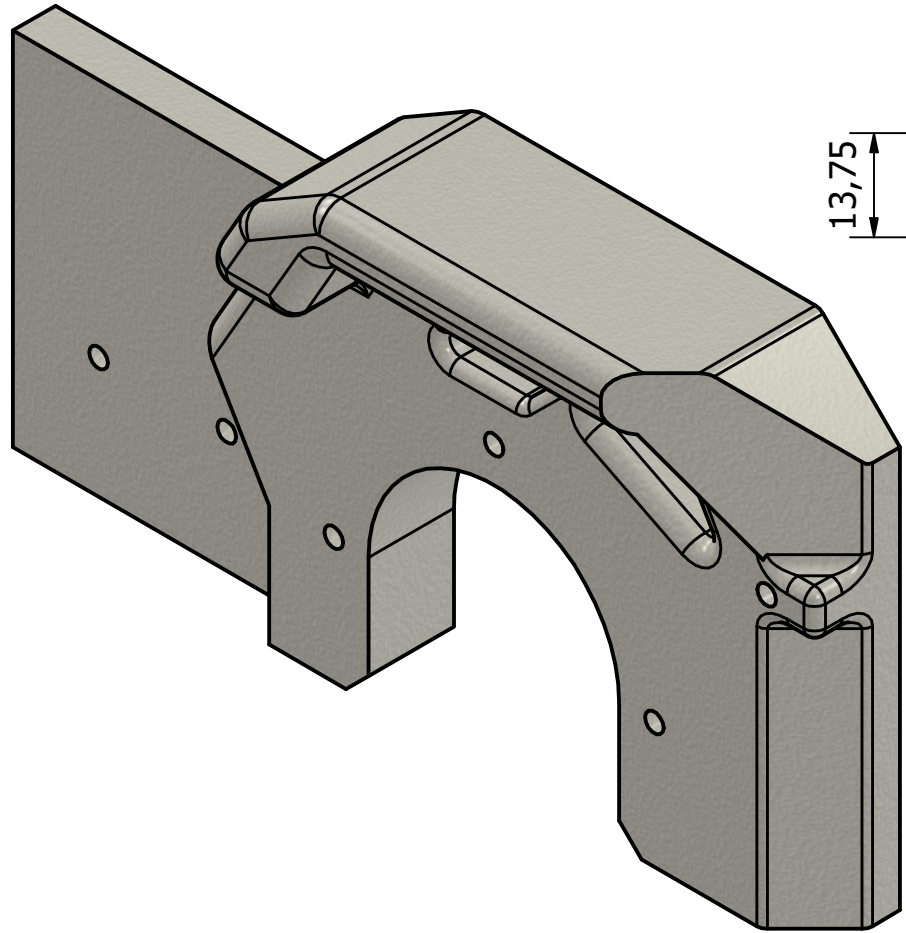
Figura 135: Cubierta de acrílico insertado un botón de encendido.



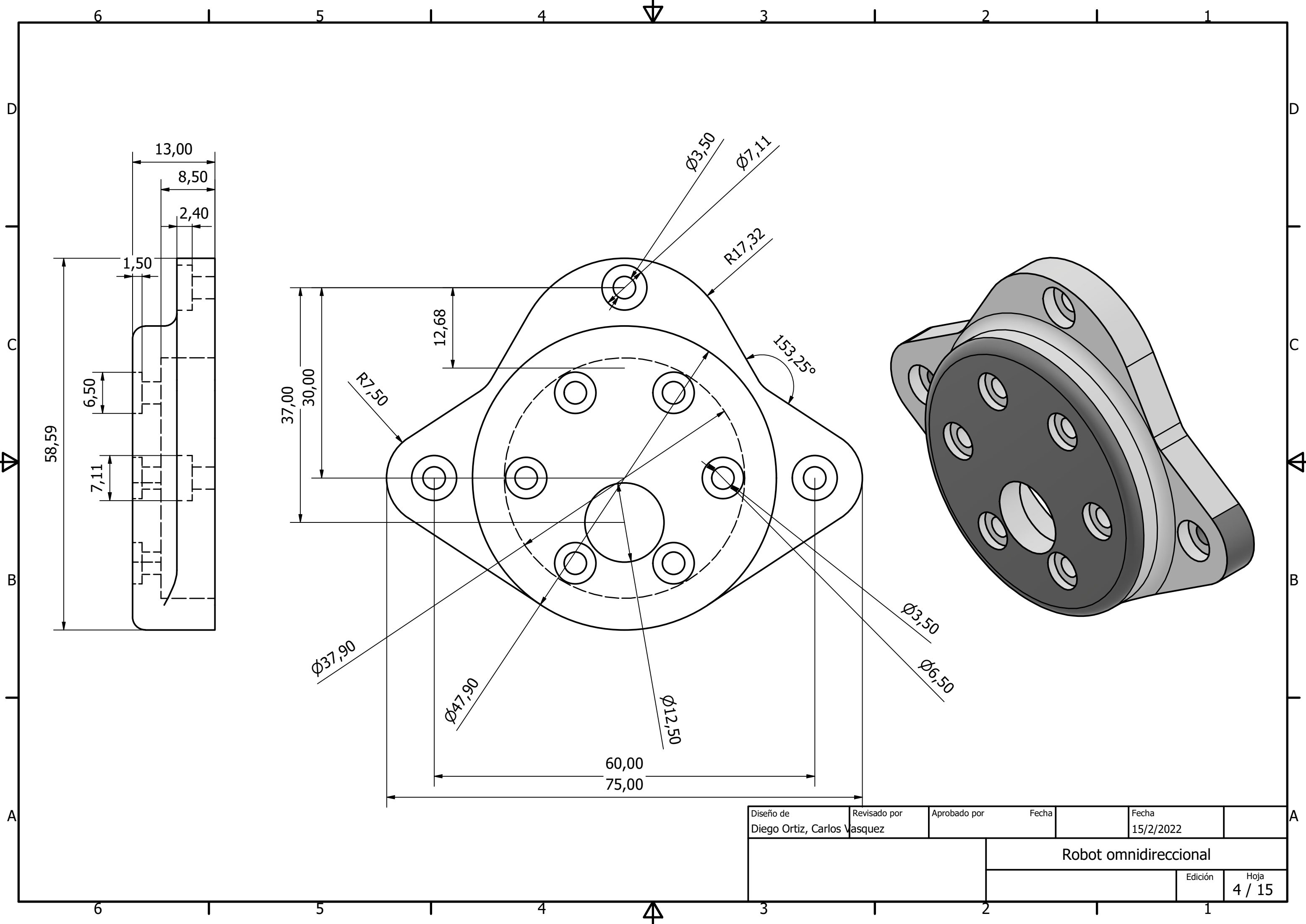
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 1 / 15	



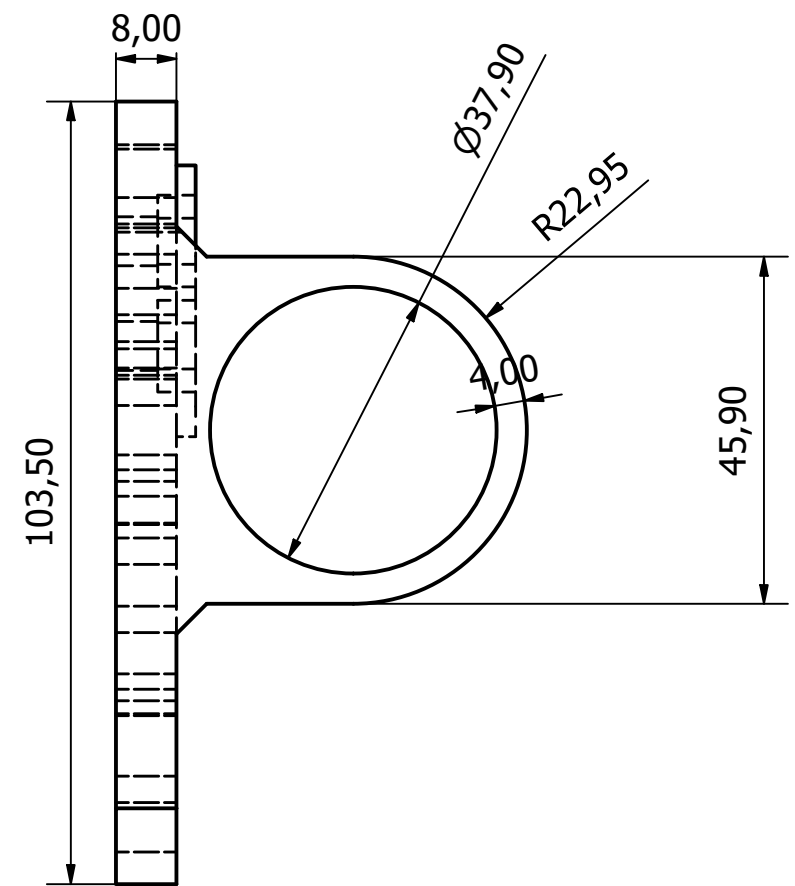
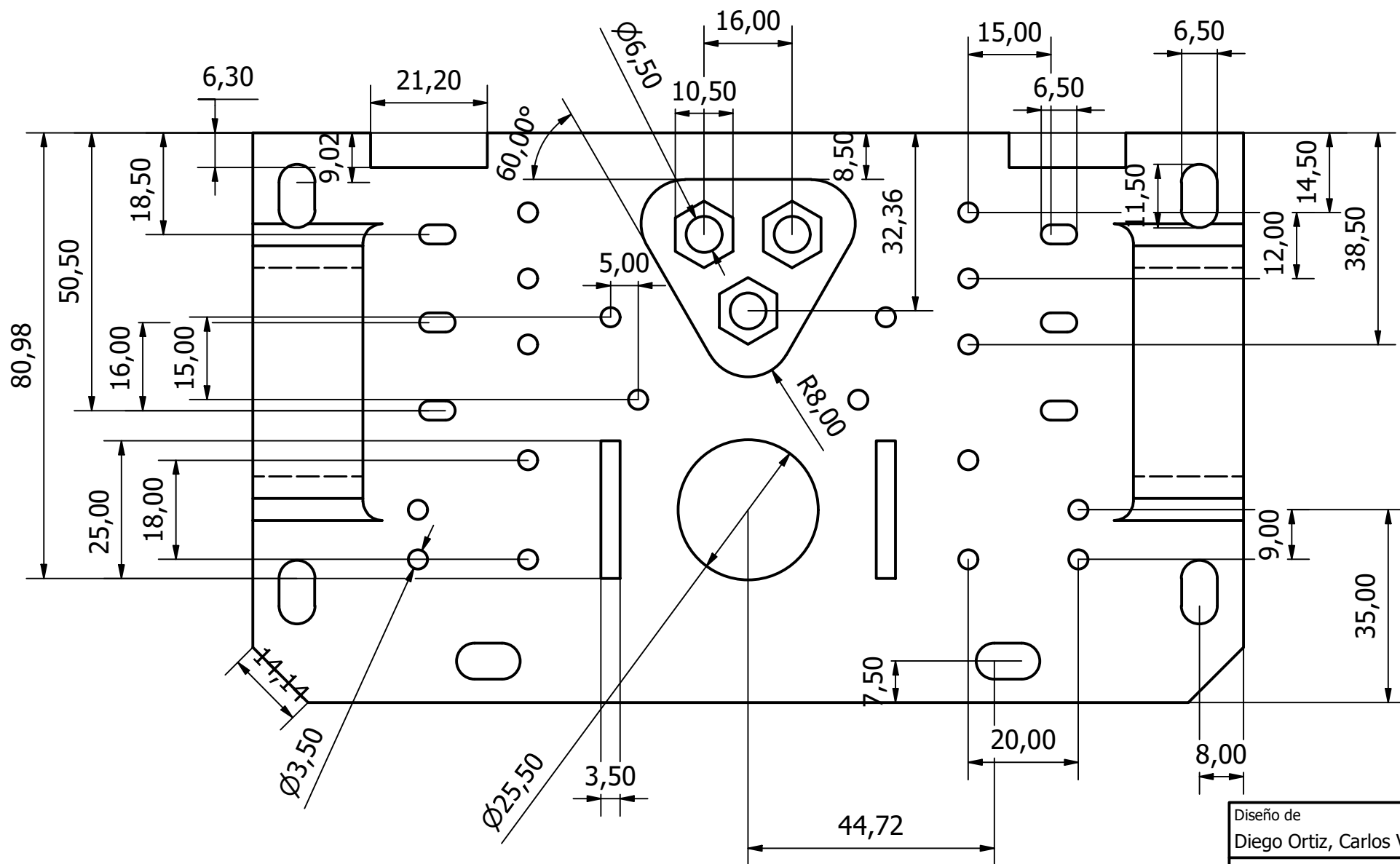
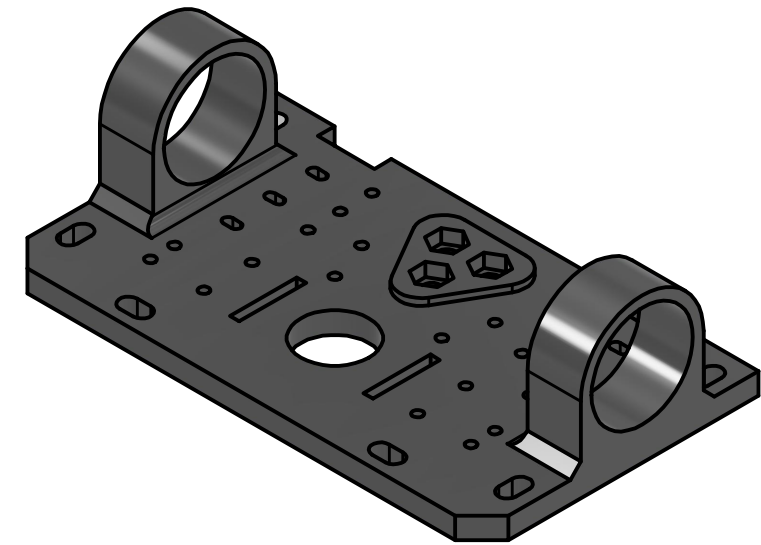
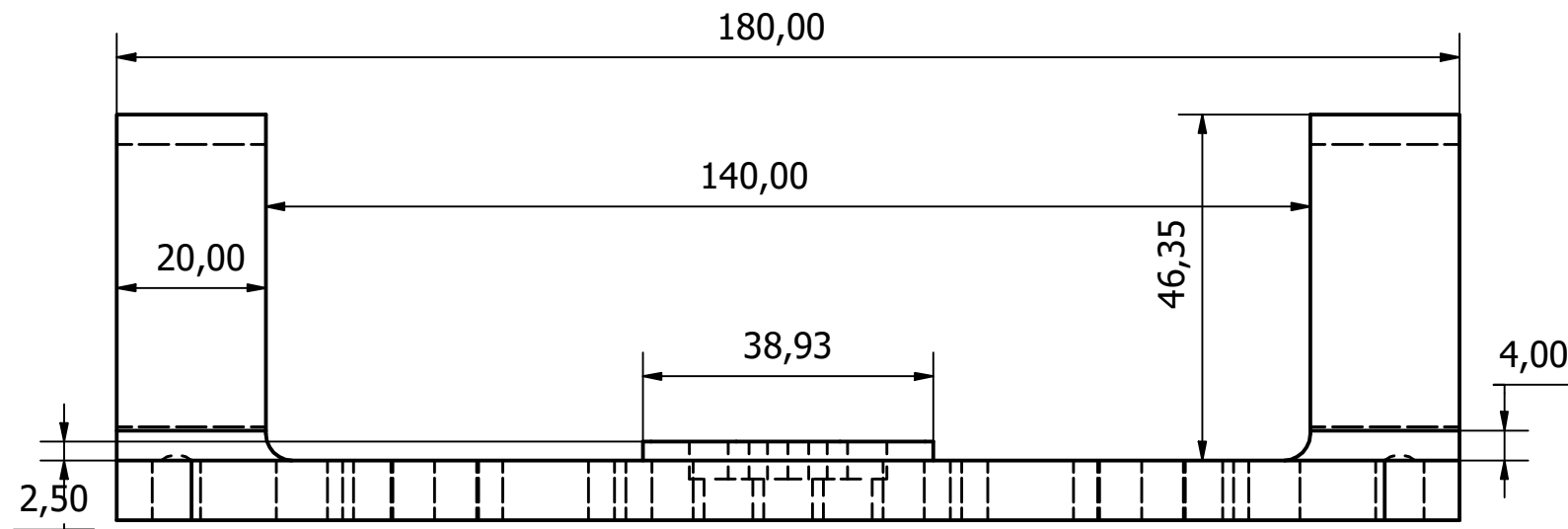
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 2 / 15	



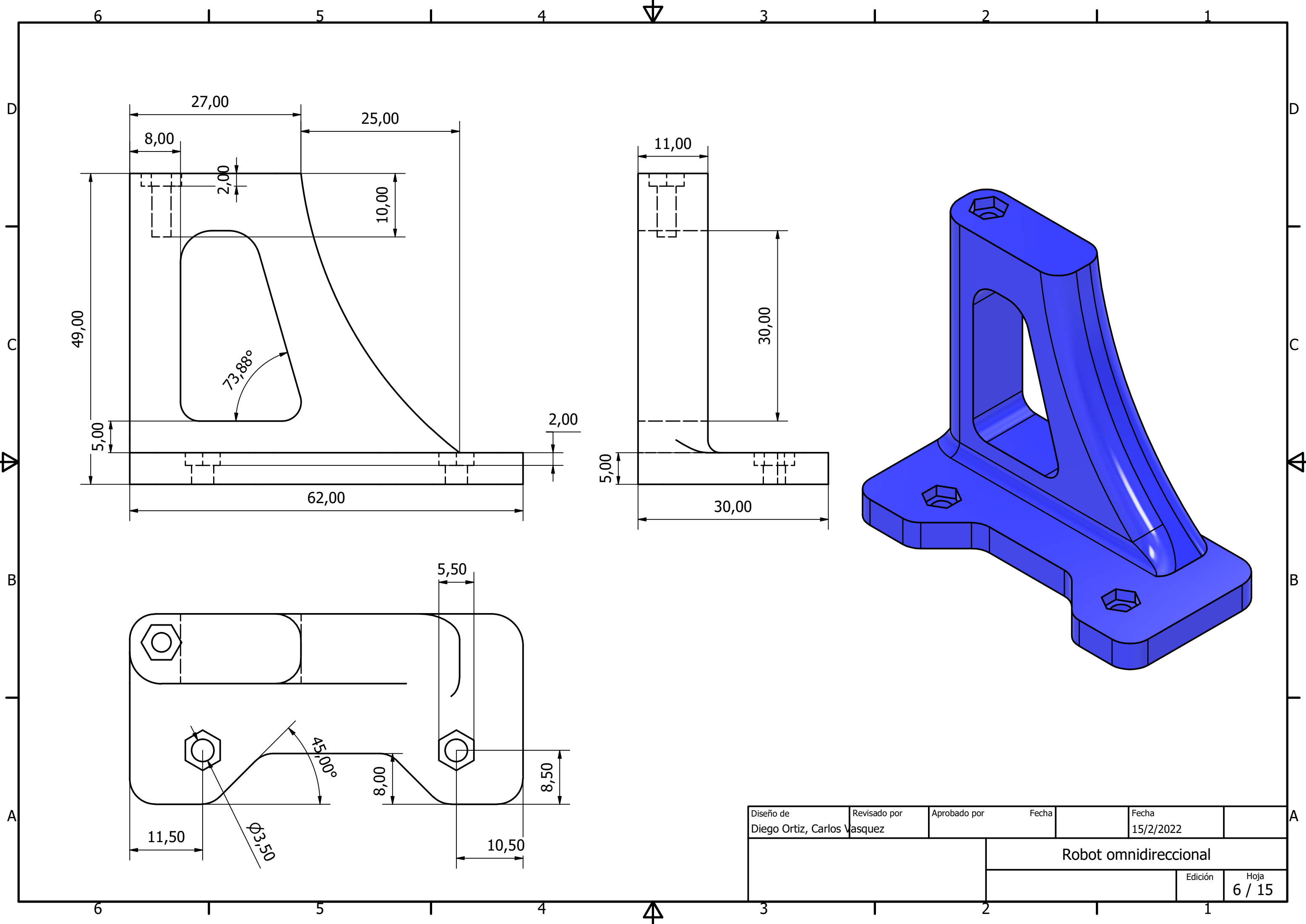
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 3 / 15



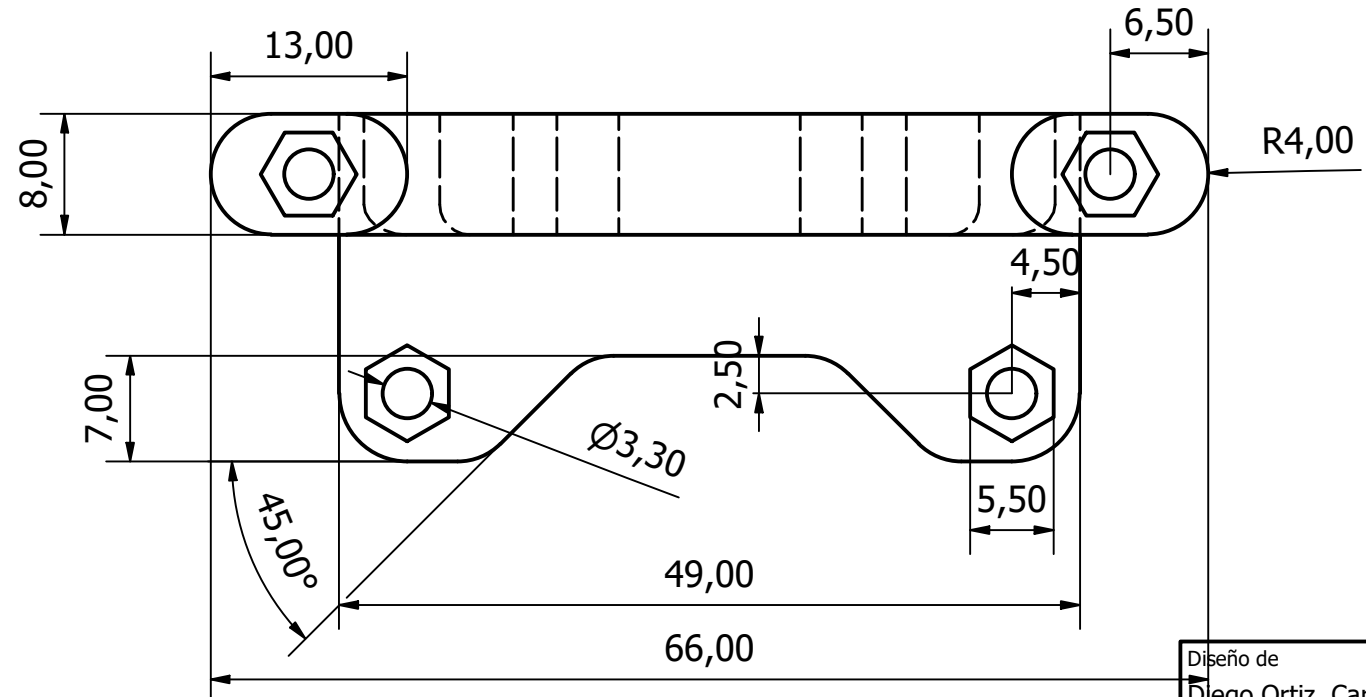
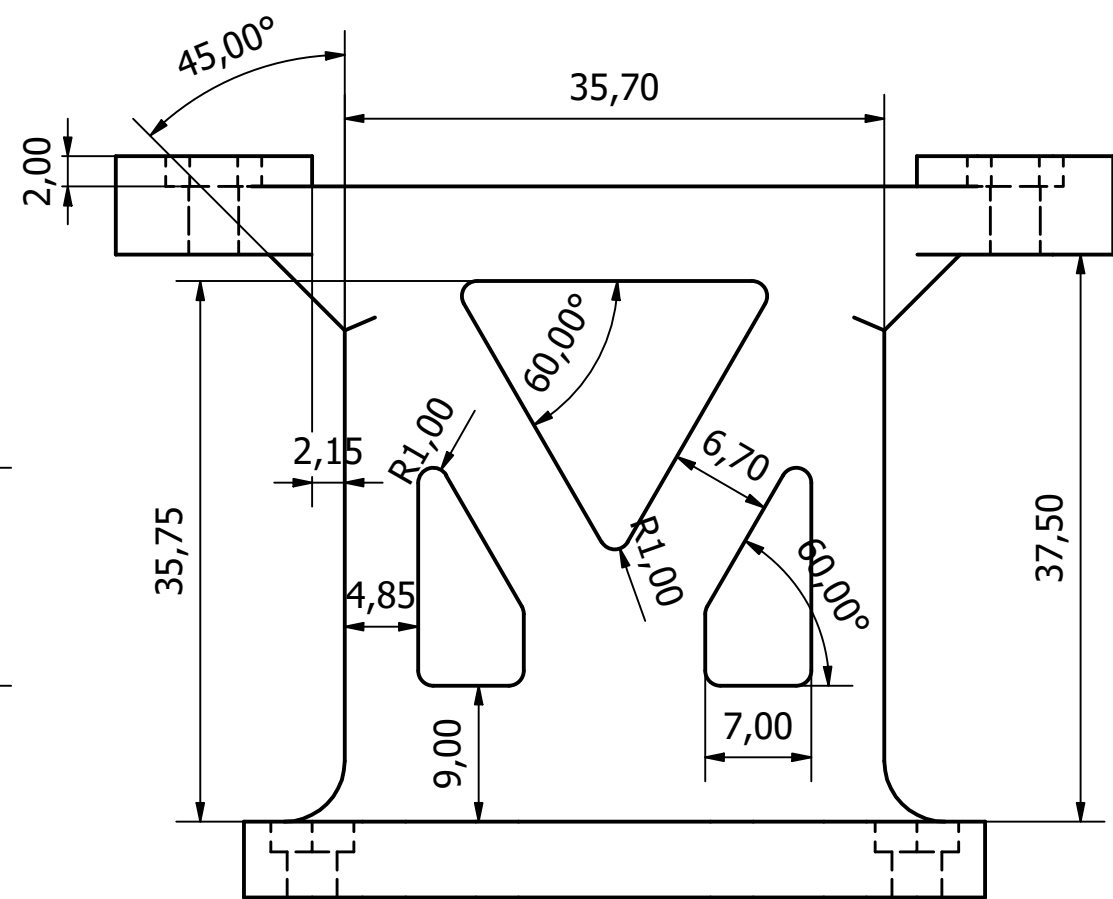
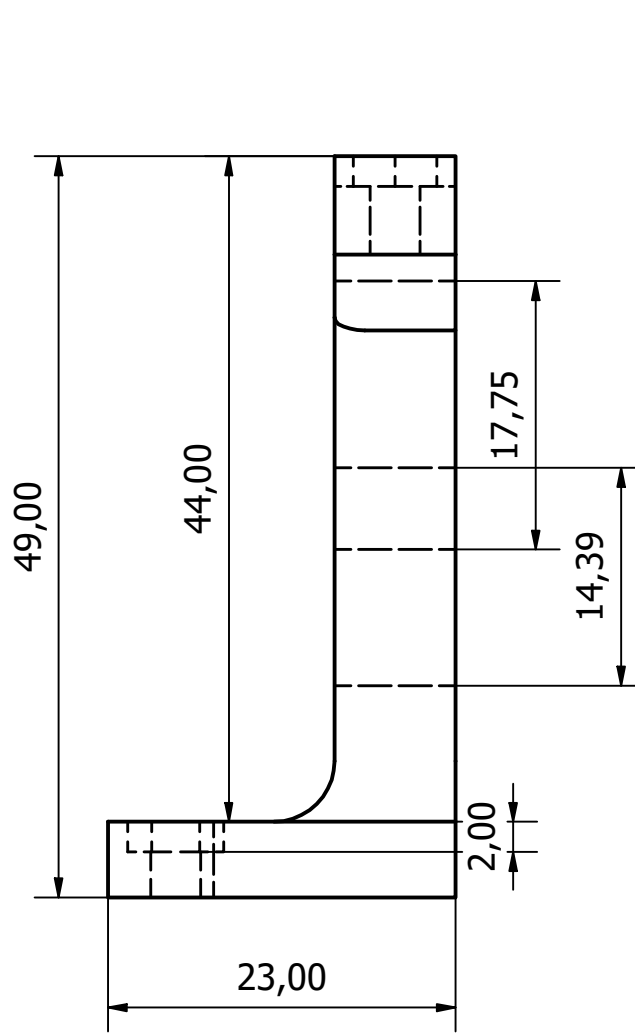
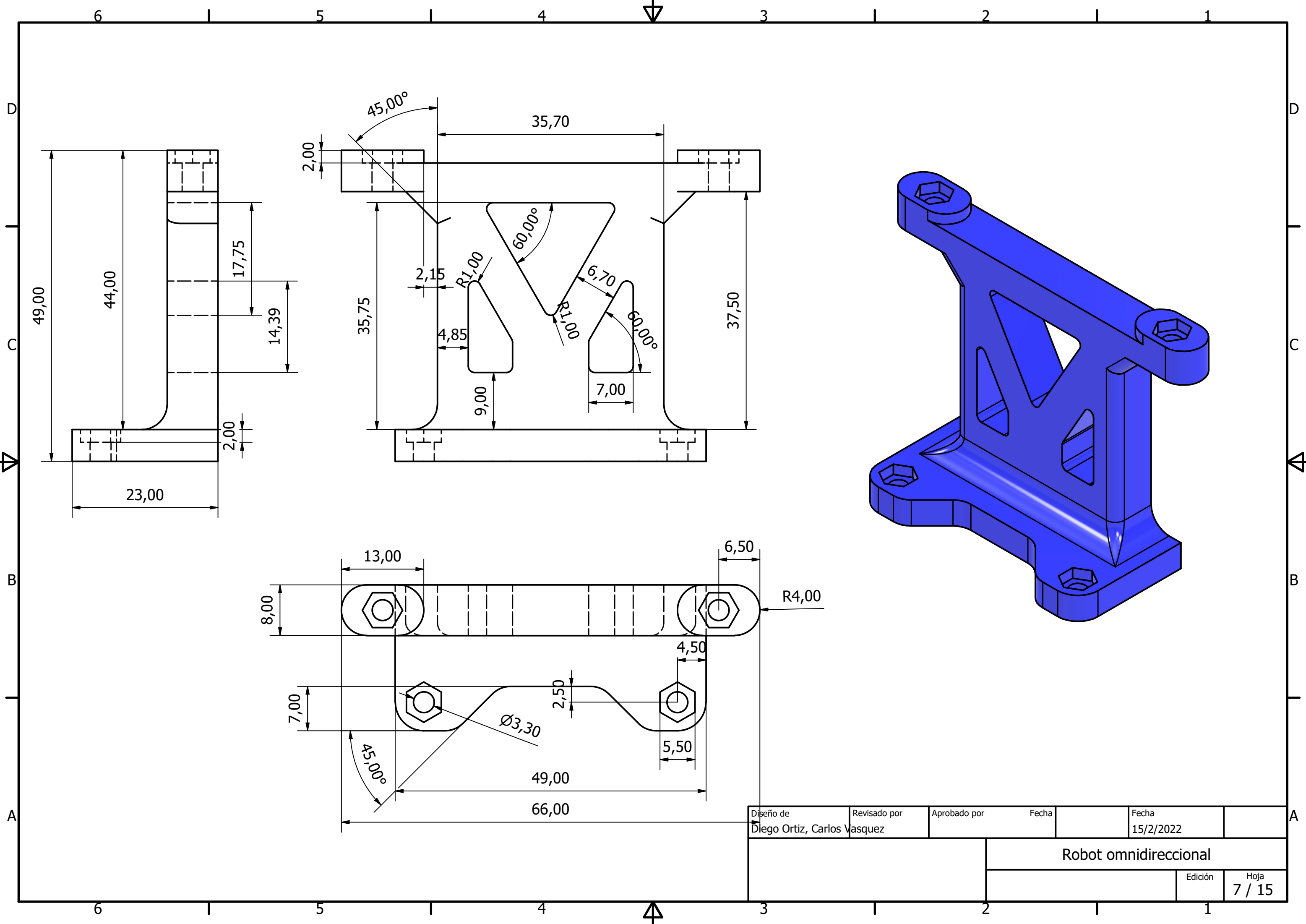
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 4 / 15



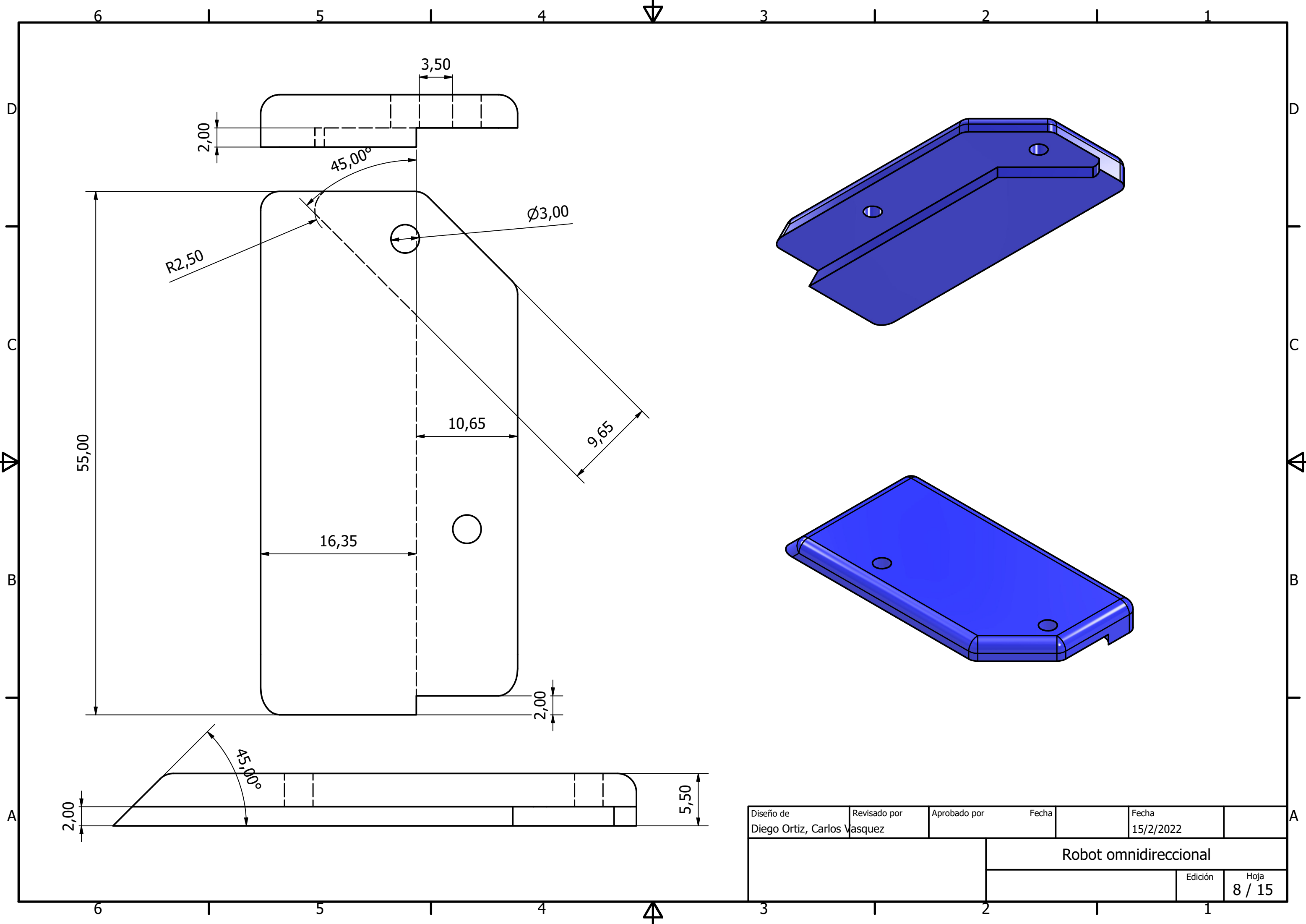
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 5 / 15	



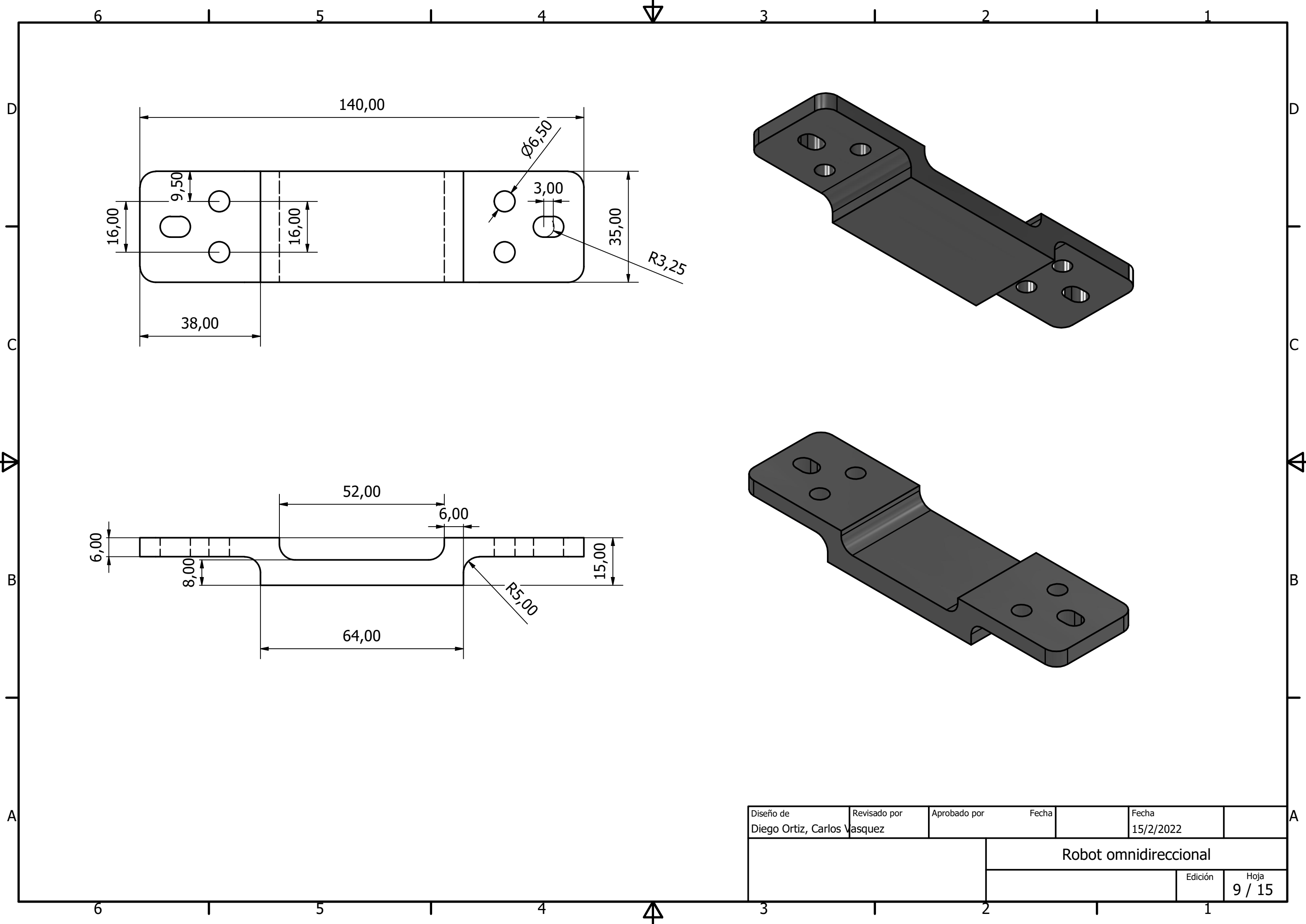
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 6 / 15



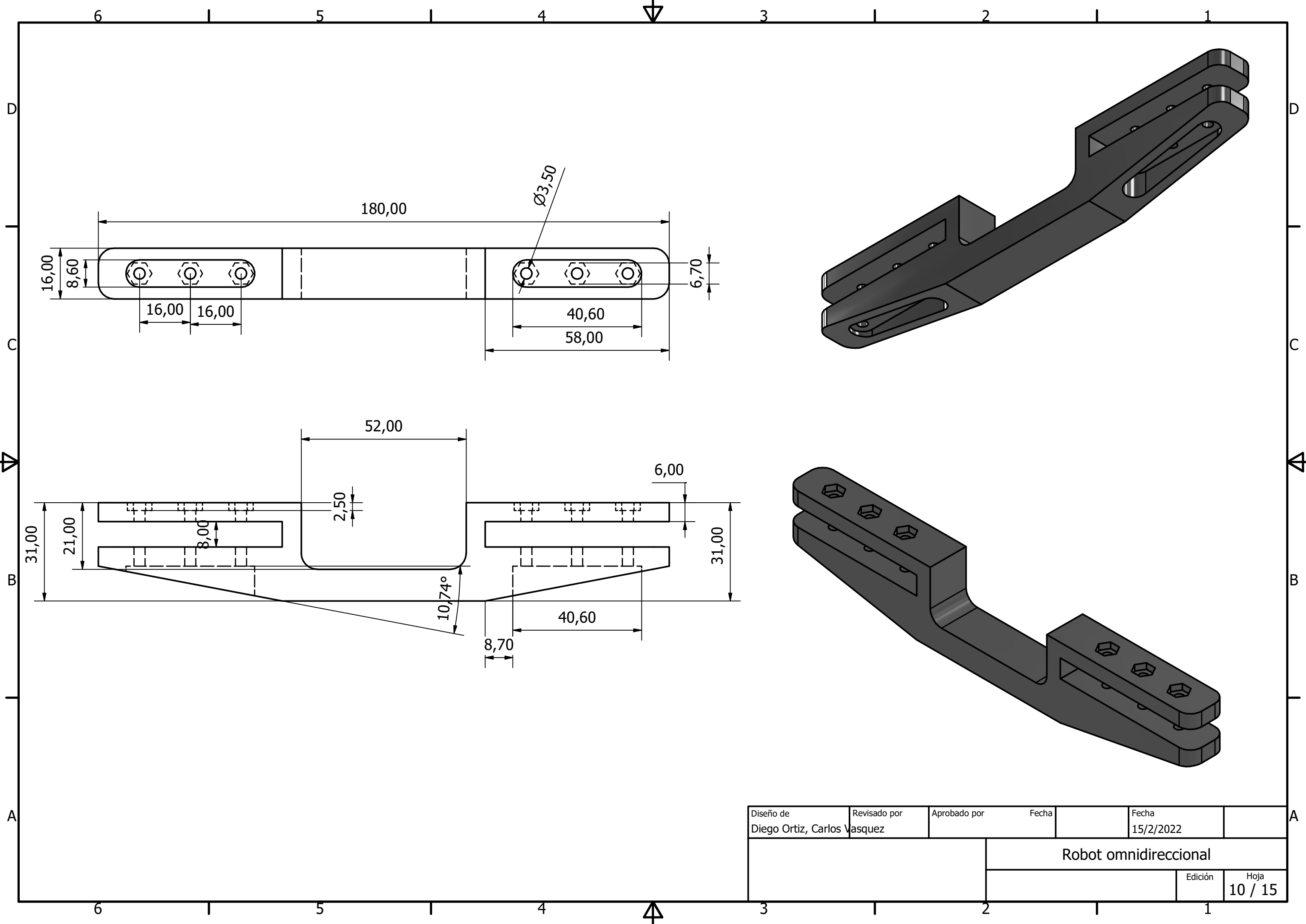
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 7 / 15



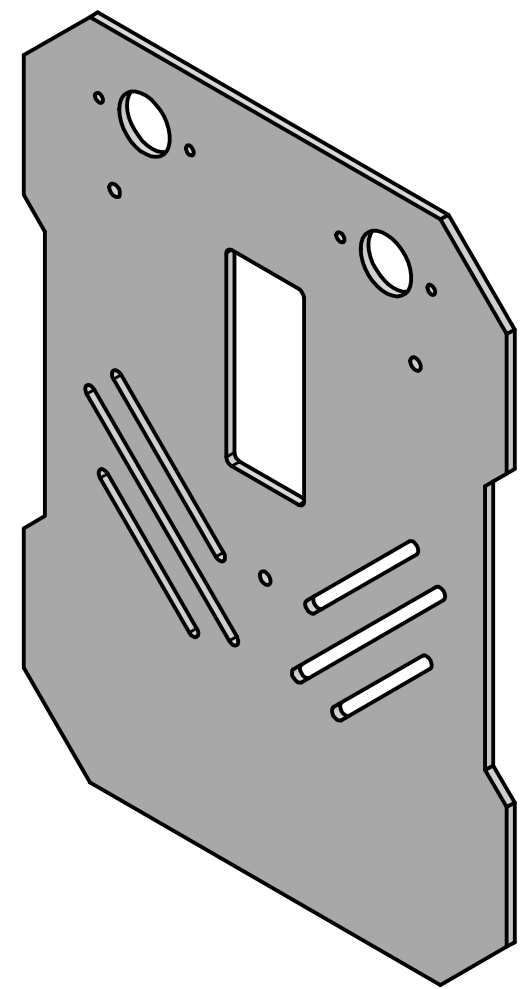
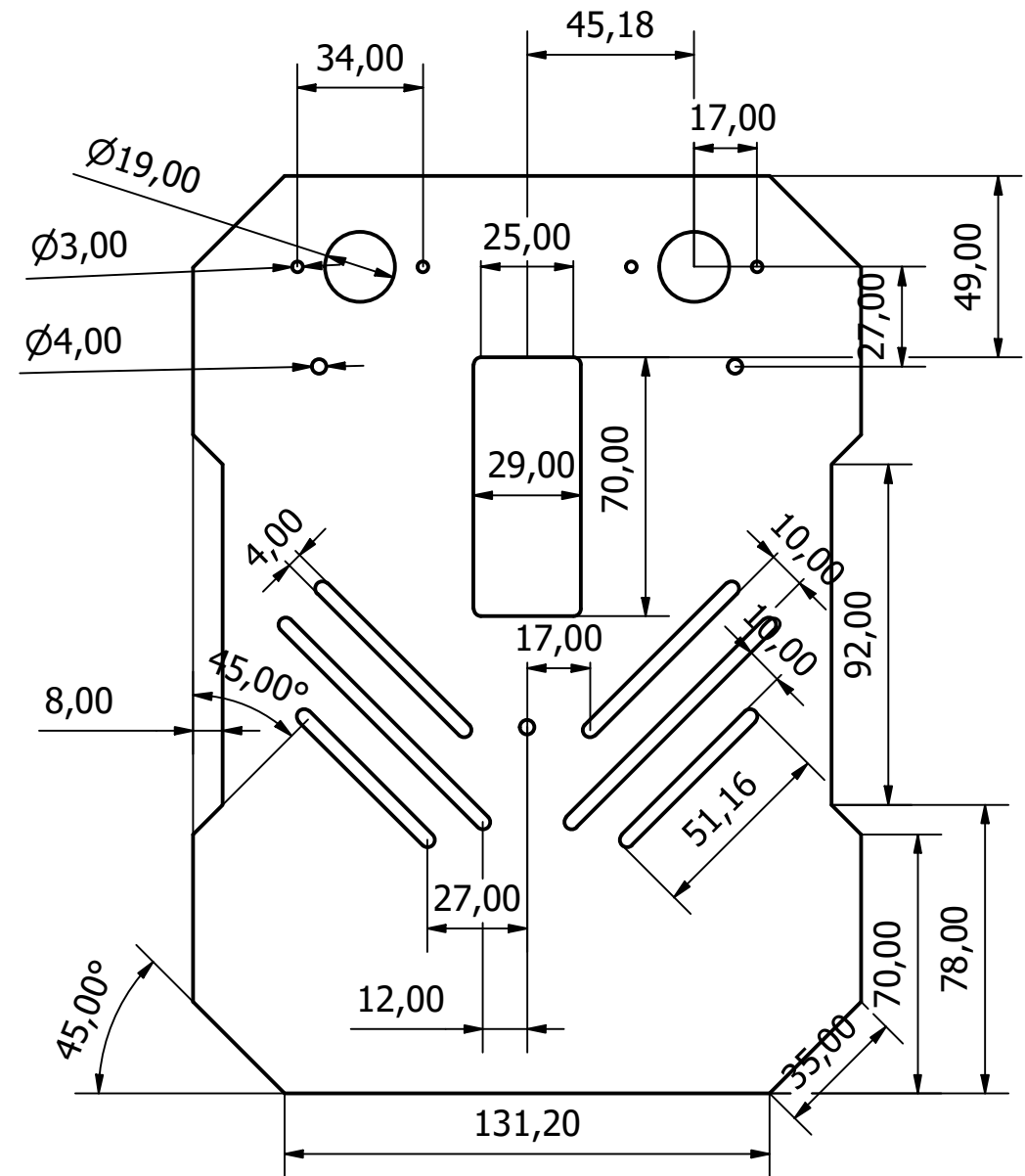
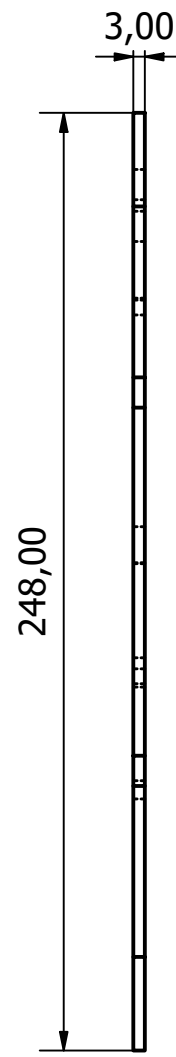
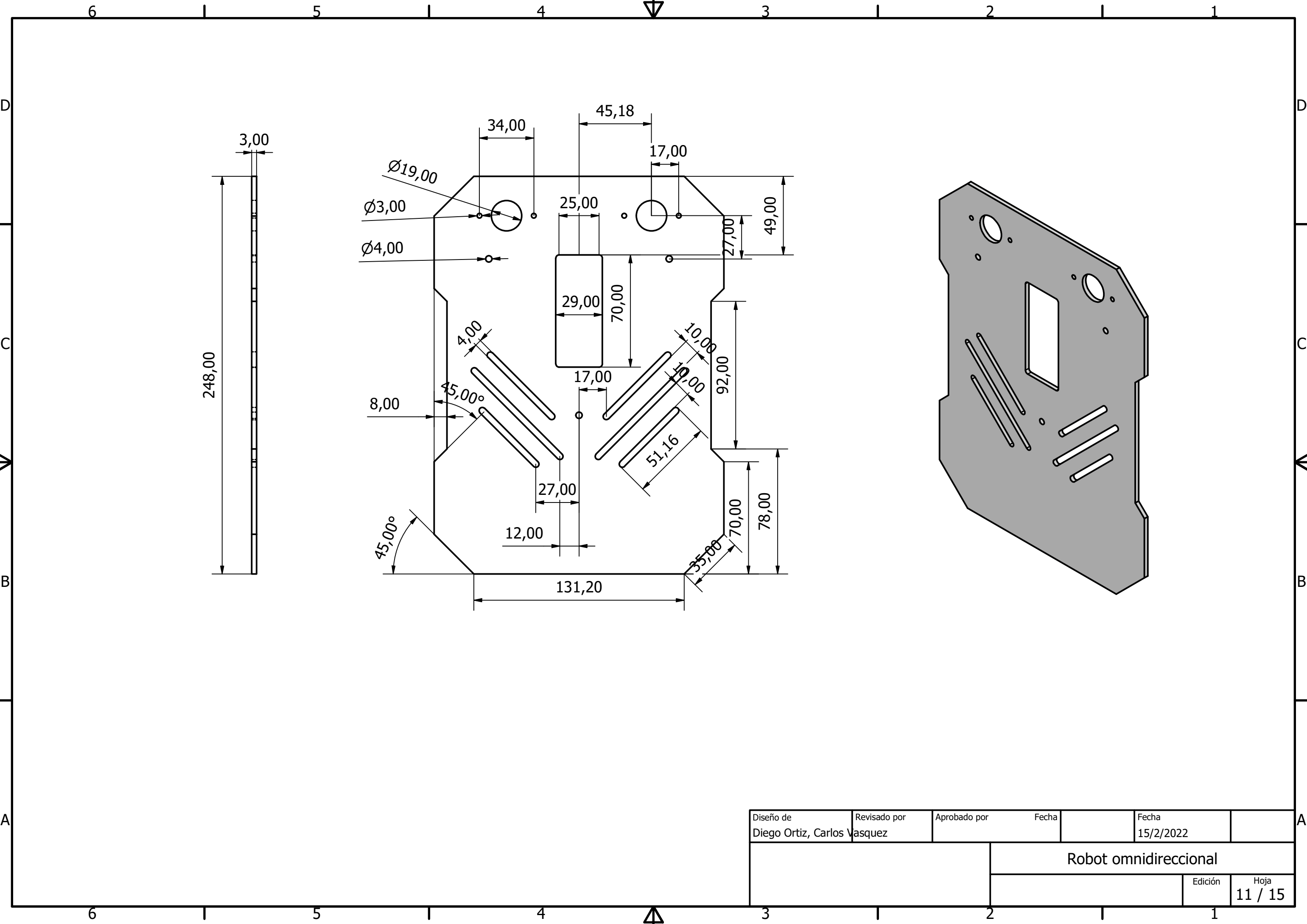
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 8 / 15	



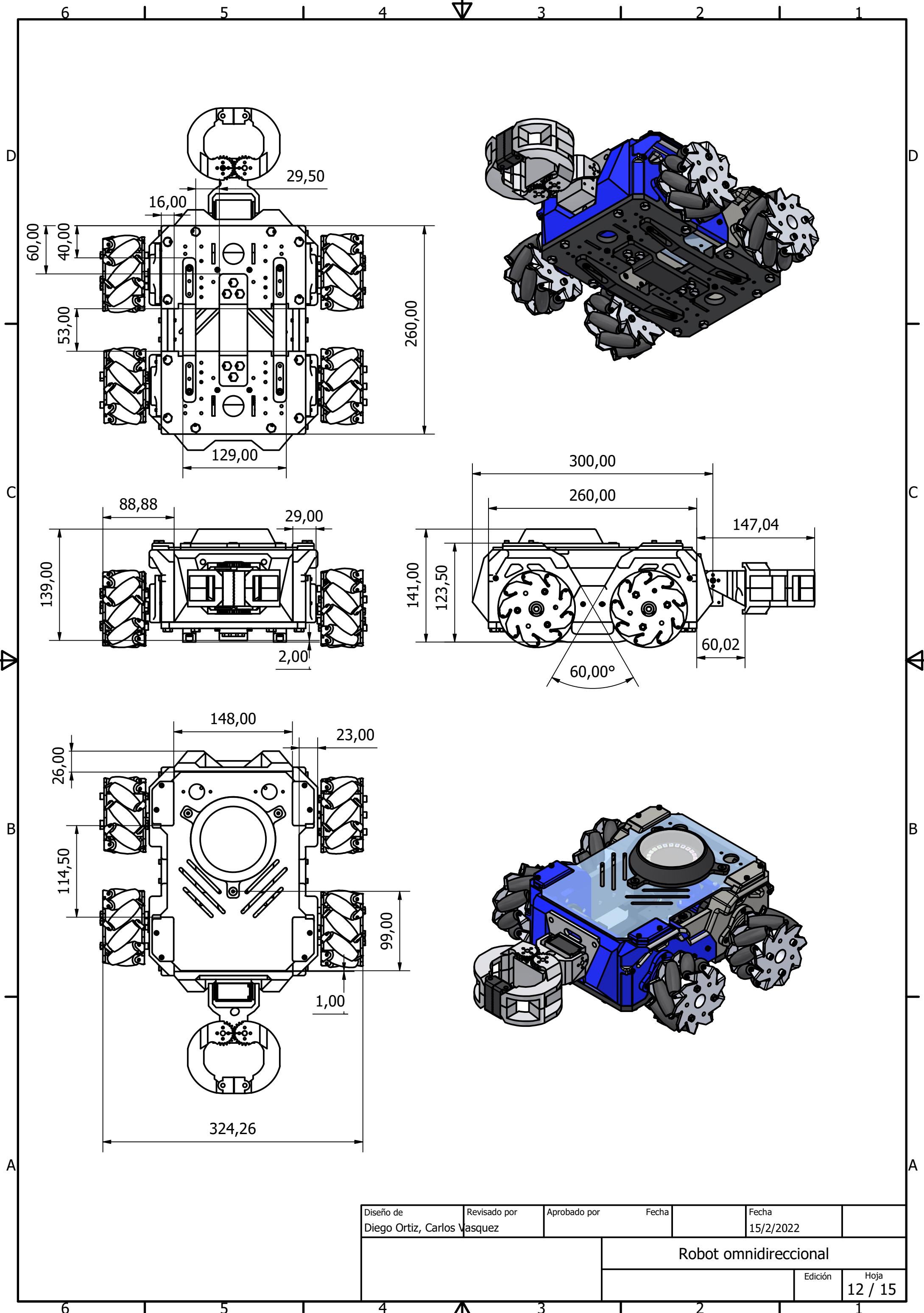
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 9 / 15



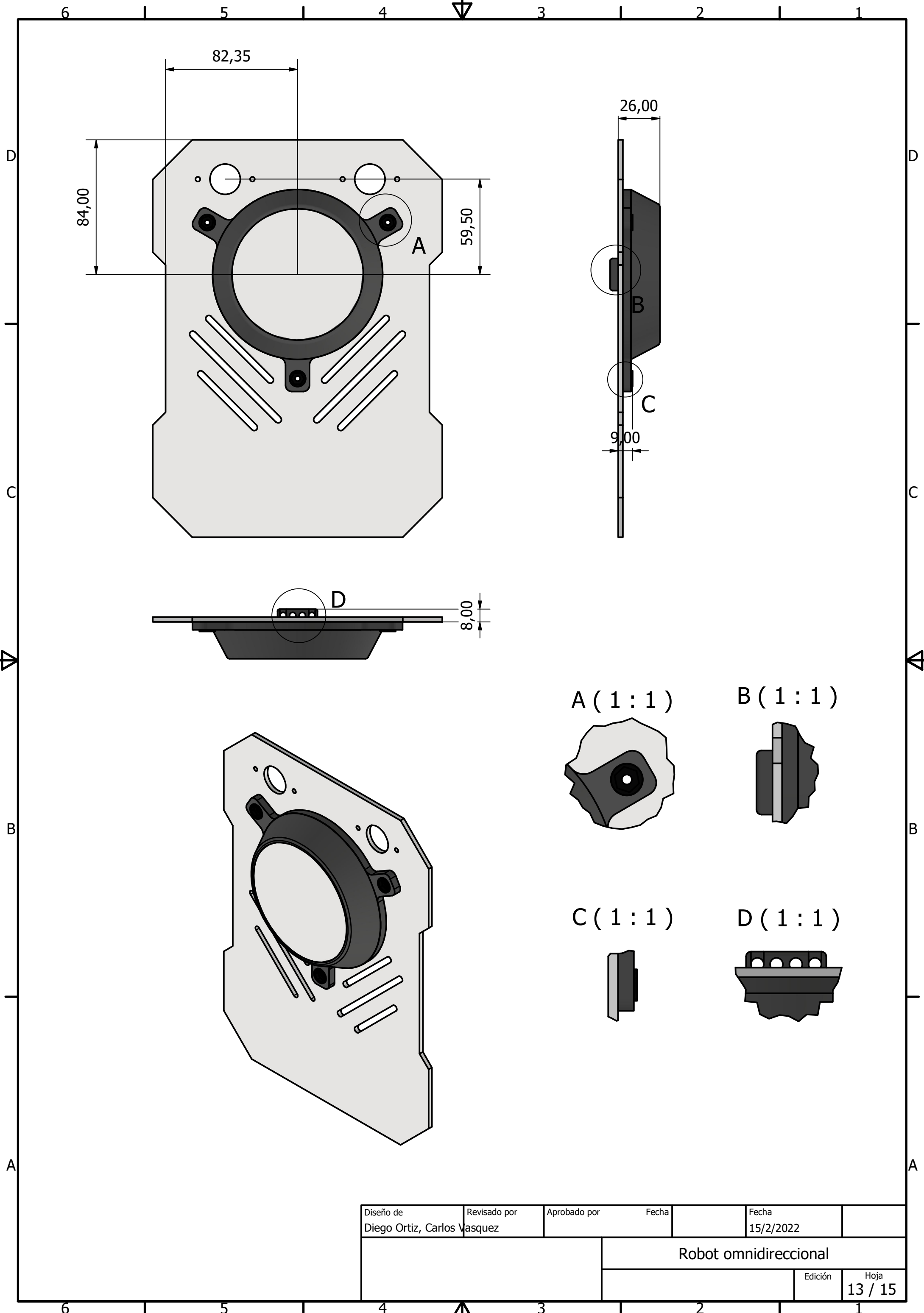
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022
			Robot omnidireccional	
			Edición	Hoja 10 / 15



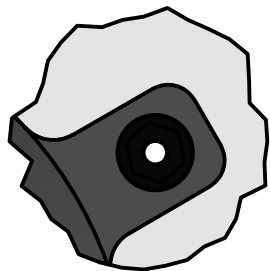
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 11 / 15	



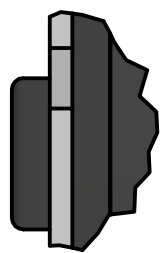
Diseño de Diego Ortiz, Carlos Vasquez	Revisado por Vasquez	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
				Edición	Hoja 12 / 15



A (1 : 1)



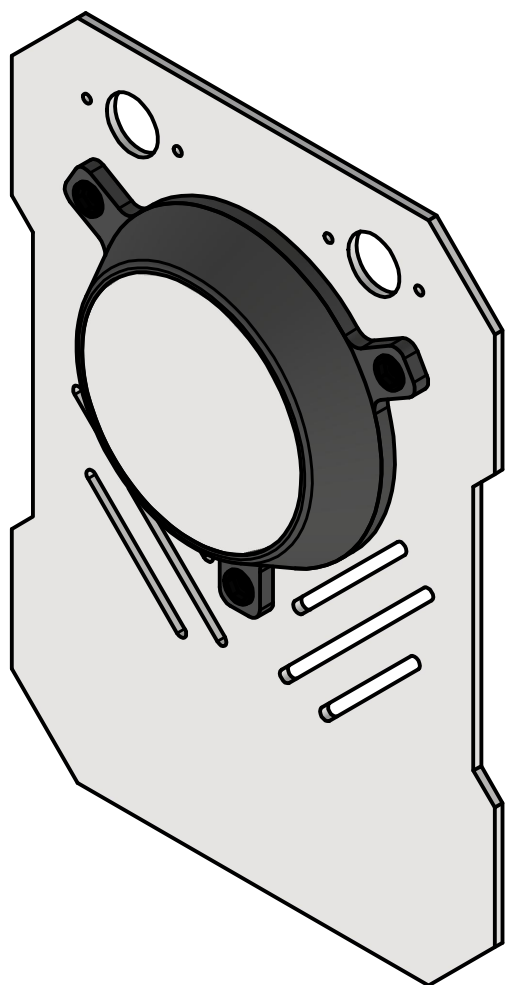
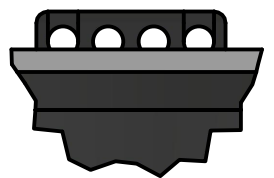
B (1 : 1)



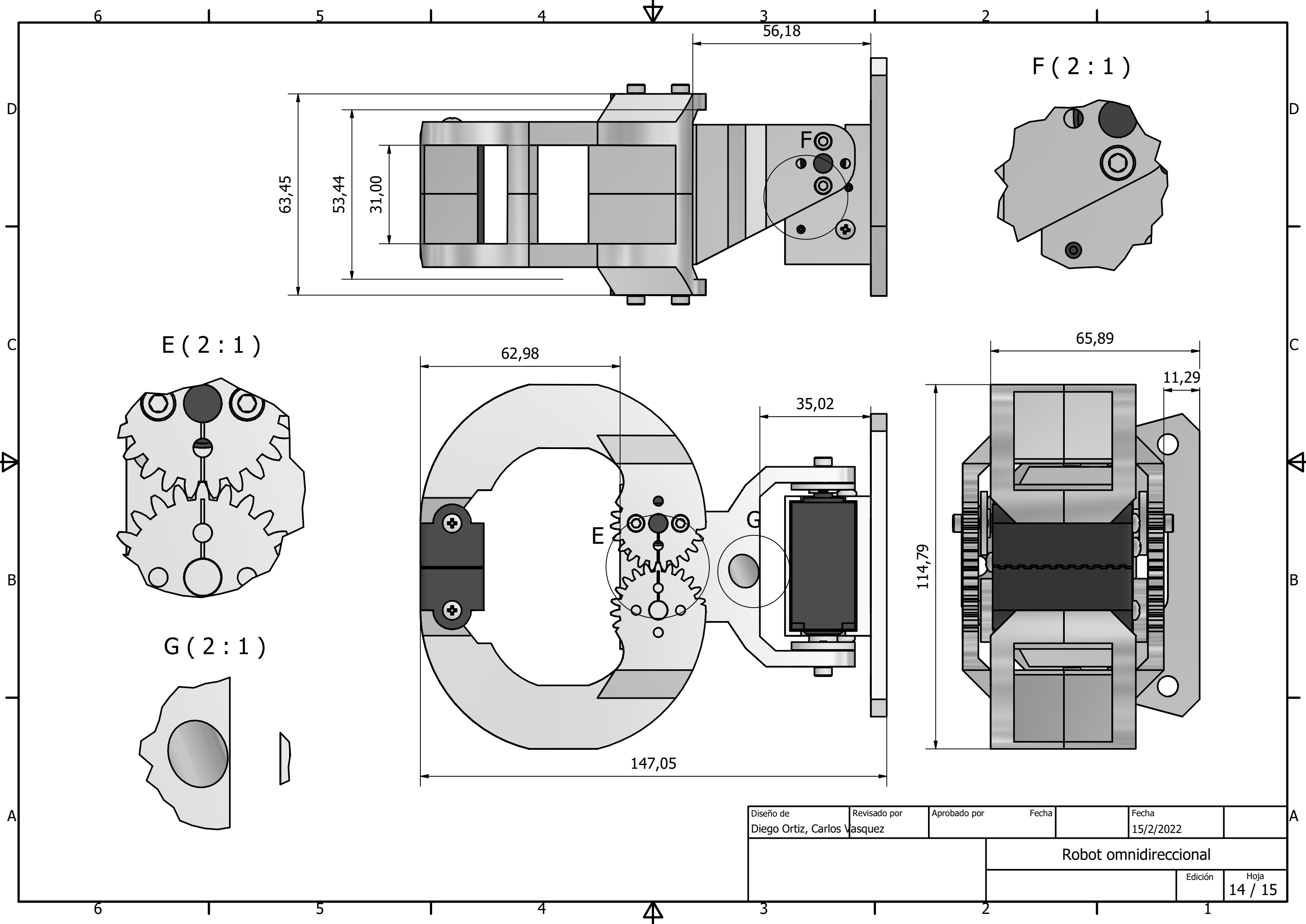
C (1 : 1)



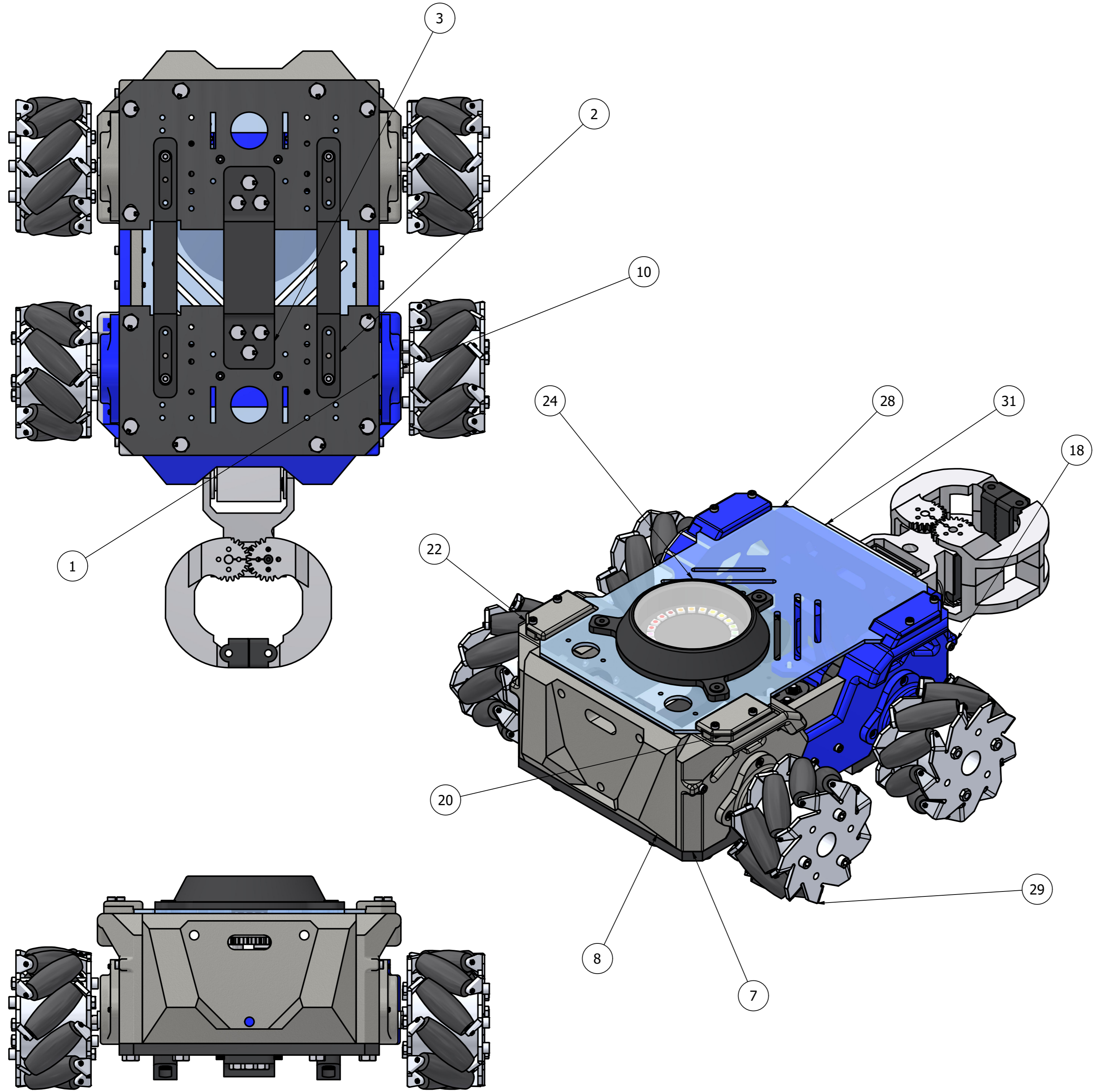
D (1 : 1)



Diseño de Diego Ortiz, Carlos Vasquez	Revisado por Vasquez	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
				Edición	Hoja 13 / 15



Diseño de Diego Ortiz, Carlos Vasquez	Revisado por	Aprobado por	Fecha	Fecha 15/2/2022	
			Robot omnidireccional		
			Edición	Hoja 14 / 15	



LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
1	2	Robot_V2_base_motor	
2	2	Robot_V3_soporte_bateria	
3	1	Robot_V3_union_central	
4	4	37d-gearmotor-19-30-encoder	STEP AP214
5	4	Rueda_test	
6	1	Motor_rueda_V3_Simplificar_1	
7	1	Robot_pared_motor_2_MIR	
8	2	Robot_V2_frente	
9	1	Robot_pared_motor_MIR_MIR	
10	4	Robot_V2_sujetador_motor	
11	1	GOLDBAT_6200mAh_2S_50C_Lipo_Battery	
12	1	caja	
13	1	Robot_pared_motor_MIR_MIR1	
14	1	Robot_pared_motor_2_MIR_MIR	
15	1	Soporte_placa	
16	1	Placa_final_Simplificar_1	
17	1	Soporte_placa_2	
18	1	Robot_pared_motor_MIR	
19	2	soporte_acrilico_inferior	
20	2	soporte_acrilico_superior	
21	2	soporte_acrilico_inferior_MIR	
22	2	soporte_acrilico_superior_MIR	
23	1	Modulo_led_inferior	
24	1	Modulo_led_superior	
25	1	acrilico_modulo_led	
26	1	neopixel_ws2812b	
27	3	Modulo_led_sujetador	
28	1	acrilico_tapa_robot	
29	2	Mecanum_Shrinkwrap_1_MIR1	
30	2	Mecanum_Shrinkwrap_1_MIR1_MIR	
31	1	Gripper_ensamble_Simplificar_2	
32	8	M6_25mm	STEP AP203
33	12	M6_20mm	STEP AP203
34	4	M4_20mm	STEP AP203
35	12	M3_20mm	STEP AP203
36	4	M3_18mm	STEP AP203
37	4	M3_30mm	STEP AP203
38	8	M3_8mm	STEP AP203
39	4	M3_16mm	STEP AP203
40	3	M3_6mm	STEP AP203
41	1	Placa_final	Open CASCADE Model

Diseño de: Diego Ortiz, Carlos Vasquez Revisado por: Aprobado por: Fecha: 15/2/2022

Robot omnidireccional