

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Mecánica y Ciencias de la  
Producción**

Diseño de un robot móvil autónomo para entregas de alimentos en un  
campus universitario

**PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**Ingenieros en Mecatrónica**

Presentado por:

Piedad Jamileth Miranda Chiquito

Alex Xavier Anrrango Tibanquiza

GUAYAQUIL - ECUADOR

Año: 2022

## DEDICATORIA

Este proyecto se lo dedico a mis padres, por su confianza y apoyo incondicional. A mis abuelos, tías y a mi hermana, que siempre creyeron en mi y me han servido de inspiración a lo largo de mi carrera universitaria.

**Alex Anrrango**

El presente proyecto se lo dedico primeramente a Dios por darme la firme convicción de culminar mis estudios. A mis padres, Alejandro y Nery, por el apoyo incondicional que me han otorgado, a mis hermanos, Alejandra y Brayan, por su compañía durante todo este proceso. A mis tíos y abuelos por la motivación que me brindaron cada día.

**Piedad Miranda**

## **AGRADECIMIENTOS**

Mis totales agradecimientos a mis padres, por su esfuerzo y apoyo durante toda la carrera. Al Ph.D. Christian Silva, por su valiosa guía y constante apoyo en el desarrollo de este trabajo. A mis amigos y demás familiares que siempre han creído en mí. Se los agradezco por siempre

**Alex Anrrango**

Agradezco sinceramente a Dios por tan linda experiencia durante el pregrado, a mis padres, quienes siempre dedicaron sus esfuerzos en mi educación y me guiaron en todo momento, a mis hermanos, por su apoyo en las largas jornadas de estudio. Al Ph.D. Christian Silva, por su guía durante el desarrollo de la tesis, y a mis amigos, de quienes me llevo un grato recuerdo de esta etapa.

**Piedad Miranda**

## DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Alex Xavier Anrrango Tibanquiza* y *Piedad Jamileth Miranda Chiquito* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Alex Xavier Anrrango Tibanquiza



Piedad Jamileth Miranda Chiquito

# EVALUADORES

---

**Efraín Terán, M.Sc.**  
PROFESOR DE LA MATERIA

---

**Christian Silva, Ph.D.**  
PROFESOR TUTOR

## RESUMEN

El servicio de entregas de comida se ha convertido en una necesidad para clientes, que debido al clima soleado o poco tiempo libre, prefieren evitar el traslado hacia un local de comida. Es común encontrar estos usuarios en un campus universitario, pero no suele haber un servicio de entregas que les facilite el traslado. El presente proyecto plantea diseñar un robot móvil autónomo, utilizando técnicas de control y algoritmos de navegación, para entregas de alimentos en el campus universitario de la ESPOL.

En el desarrollo de la solución, se utilizó el software SolidWorks 2020 para el modelado de la estructura mecánica, MATLAB R2022a para el diseño de control y simulación del robot, y el lenguaje Python para el desarrollo de un algoritmo de visión por computadora, para la detección de rampas y escaleras en el entorno.

La estructura final se diseñó en el material ABS y tuvo dimensiones de 500mmx600mm x 500mm, con un peso total de 40 kg. La autonomía se la determinó con el consumo de potencia de los dispositivos electrónicos, logrando un tiempo de trabajo de al menos 4 horas. Además, se pudo simular el robot en un ambiente 3D, donde los algoritmos de navegación y control permitieron validar el movimiento en una ruta deseada.

Finalmente, se concluye que el diseño es capaz de transportar alimentos de forma autónoma, alrededor de un terreno con irregularidades, obstáculos y presencia de rampas a lo largo del camino.

**Palabras clave:** Robot móvil, MATLAB, Simulink, Visión por computadora, Simulación.

## **ABSTRACT**

*Food delivery service has become a necessity for customers who, due to sunny weather or limited free time, prefer to avoid the commute to a food store. It is common to find these users on a university campus, but there is usually no delivery service to facilitate the transfer. This project proposes to design an autonomous mobile robot, using control techniques and navigation algorithms, for food deliveries in the university campus of ESPOL.*

*In the development of the solution, SolidWorks 2020 software was used for the modeling of the mechanical structure, MATLAB R2022a for the control design and simulation of the robot, and the Python language for the development of a computer vision algorithm for the detection of ramps and stairs in the environment.*

*The final structure was designed in ABS material and had dimensions of 500mmx600mm x 500mm, with a total weight of 40 kg. The autonomy was determined with the power consumption of the electronic devices, achieving a working time of at least 4 hours. In addition, it was possible to simulate the robot in a 3D environment, where the navigation and control algorithms allowed validating the movement in a desired route.*

*Finally, it is concluded that the design is capable of transporting food autonomously, around a terrain with irregularities, obstacles and the presence of ramps along the way.*

**keywords:** *Mobile Robot, MATLAB, Simulink, Computer Vision, Simulation*

# ÍNDICE GENERAL

RESUMEN .....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS .....	VI
SIMBOLOGÍA .....	VII
ÍNDICE DE FIGURAS .....	VIII
ÍNDICE DE TABLAS .....	X
CAPÍTULO 1 .....	1
1. Introducción .....	1
1.1 Descripción del problema .....	2
1.2 Justificación del problema .....	3
1.3 Objetivos .....	4
1.3.1 Objetivo general .....	4
1.3.2 Objetivos específicos .....	4
1.4 Marco teórico .....	5
1.4.1 <i>Delivery</i> en la industria de alimentos .....	5
1.4.2 Robots móviles con ruedas .....	7
1.4.3 Sistema de navegación .....	8
1.4.4 Esquema de control .....	9
1.4.5 Estado del arte .....	11
CAPÍTULO 2 .....	14
2. Metodología .....	14
2.1 Concepto de la solución .....	14
2.1.1 Alternativas de solución .....	14
2.1.2 Requerimientos de diseño .....	16
2.1.3 Diseño conceptual .....	17

2.1.4	Metodología .....	18
2.2	Modelos matemáticos .....	19
2.2.1	Cinemática directa .....	19
2.2.2	Cinemática inversa .....	20
2.2.3	Modelo de suspensión .....	21
2.3	Diseño electrónico .....	22
2.3.1	Consumo de potencia .....	22
2.3.2	Selección de la batería .....	23
2.3.3	Cálculo de la autonomía .....	24
2.4	Diseño mecánico .....	25
2.4.1	Selección de motor .....	25
2.4.2	Diseño del muelle de suspensión .....	27
2.4.3	Diseño de eje cardán .....	28
2.4.4	Selección de materiales .....	29
2.4.5	Análisis de carga por elementos finitos .....	31
2.5	Diseño de control .....	33
2.5.1	Planificación global .....	34
2.5.2	Localización en el mapa.....	35
2.5.3	Planificación local .....	37
2.5.4	Visión por computadora .....	39
2.5.5	Algoritmo de evasión de obstáculos .....	41
2.5.6	Control del seguimiento de ruta .....	43
2.5.7	Controlador PI de las velocidades de las ruedas .....	45
CAPÍTULO 3 .....		49
3.	Resultados y análisis .....	49
3.1	Diseño de la estructura mecánica .....	49
3.2	Dispositivos electrónicos seleccionados .....	51
3.3	Detección y subida de rampas .....	53
3.4	Simulación .....	55
3.4.1	Modelo en MATLAB - Simulink .....	55
3.4.2	Descripción del escenario.....	56
3.4.3	Análisis de la ruta obtenida .....	58

3.4.4	Controlador PI de los motores .....	61
3.5	Análisis de costos .....	63
CAPÍTULO 4 .....		65
4.	Conclusiones y recomendaciones.....	65
4.1	Conclusiones.....	65
4.2	Recomendaciones.....	67
BIBLIOGRAFÍA .....		68
APÉNDICES .....		77

## ABREVIATURAS

<b>ABS</b>	Acrilonitrilo Butadieno Estireno
<b>ASA</b>	Acrilonitrilo Estireno Acrilato
<b>BMS</b>	Battery Management System
<b>BLDC</b>	Brushless Direct Current
<b>CAD</b>	Computer-Aided Design
<b>CPU</b>	Central Processing Unit
<b>DC</b>	Corriente Directa
<b>DOF</b>	Grados de Libertad
<b>EKF</b>	Extended Kalman Filter
<b>ESPOL</b>	Escuela Superior Politécnica del Litoral
<b>GPS</b>	Global Positioning System
<b>GPU</b>	Graphics Processing Unit
<b>IA</b>	Inteligencia Artificial
<b>ICR</b>	Centro Instantáneo de Rotación
<b>IMU</b>	Unidad de Medición Inercial
<b>PID</b>	Proporcional Integral Derivativo
<b>PRM</b>	Probabilistic Roadmap
<b>PWM</b>	Modulación por Ancho de Pulsos
<b>RMR</b>	Robots Móviles con Ruedas

## SIMBOLOGÍA

A Amperios

h Hora

kg Kilogramo

km Kilómetro

m Metro

mm Milímetro

MPa Megapascal

N Newton

rad Radian

rpm Revolución por minuto

s Segundo

V Voltaje

W Watts

## ÍNDICE DE FIGURAS

Figura 1.1. Riesgo de transmisión asociado al reparto de comida .....	5
Figura 1.2. Configuraciones cinemáticas de los RMR .....	7
Figura 1.3. Esquema del PID .....	10
Figura 1.4. Robots Starship en el campus de West Lafayette - Universidad Purdue	11
Figura 1.5. KiwiBot en el campus de la Universidad de Gonzaga. ....	12
Figura 1.6. Robot móvil Robby 2.....	13
Figura 2.1. Concepto de la solución .....	17
Figura 2.2. Metodología .....	18
Figura 2.3. Esquema cinemático del robot.....	19
Figura 2.4. Modelo de suspensión pasiva.....	21
Figura 2.5. Batería de Li-ion 36V 100 Ah .....	23
Figura 2.6. Llantas seleccionadas para el robot <i>delivery</i> . ....	25
Figura 2.7. Diagrama de cuerpo libre de vehículo subiendo una pendiente. ....	26
Figura 2.8. Muelle de compresión helicoidal.....	27
Figura 2.9. Muelle de compresión helicoidal diseñado.....	28
Figura 2.10. Configuración en Z, eje cardan de entrada y salida en paralelo. ....	28
Figura 2.11. Diseño del eje tipo cardán.....	29
Figura 2.12. Sección transversal de las capas del contenedor. ....	29
Figura 2.13. Porcentaje de daño en el estudio de fatiga. ....	31
Figura 2.14. Vida útil obtenida en el estudio de fatiga.....	31
Figura 2.15. Arquitectura de control .....	33
Figura 2.16. Planificación global .....	35
Figura 2.17. Comparación de la localización con EKF y la real .....	36
Figura 2.18. Sensores virtuales del modelo en Simscape Multibody .....	37
Figura 2.19. Diagrama de flujo del comportamiento del robot durante su trayectoria	38
Figura 2.20. Detección de rampa y escalera en el <i>testing</i> .....	39
Figura 2.21. Ilustración del campo de visión de profundidad .....	40
Figura 2.22. Acciones de control según las señales de los sensores de ultrasonido	42

Figura 2.23. Diagrama de flujo del control de seguimiento de ruta .....	44
Figura 2.24. Esquema de la planta y el controlador PID .....	46
Figura 2.25. Respuesta con auto-tuning .....	47
Figura 2.26. Respuesta con el controlador final .....	48
Figura 3.1. Ensamble de robot. ....	49
Figura 3.2. Esquema de sensores, actuadores y controladores .....	51
Figura 3.3. Detección de objetos con número de clase y nivel de confianza .....	53
Figura 3.4. Validación del algoritmo de detección de rampas y escaleras .....	54
Figura 3.5. Prueba del algoritmo de detección de rampas y escaleras .....	54
Figura 3.6. Bloques principales del modelo desarrollado en MATLAB-Simulink ...	55
Figura 3.7. Vista superior del ambiente 3D en MATLAB .....	56
Figura 3.8. Robot evadiendo obstáculo .....	57
Figura 3.9. Robot detectando (izquierda) y bajando (derecha) una rampa .....	57
Figura 3.10. Seguimiento de puntos y evasión de un obstáculo .....	58
Figura 3.11. Velocidad lineal y angular del robot .....	59
Figura 3.12. Acción de evasión a la izquierda .....	60
Figura 3.13. Ruta de desvío a las rampas .....	61
Figura 3.14. Comparación de las velocidades <i>setpoint</i> y las velocidades reales....	62

## ÍNDICE DE TABLAS

Tabla 1.1. Diferencias observadas entre un robot y un humano <i>delivery</i> .....	6
Tabla 2.1. Criterios de evaluación .....	15
Tabla 2.2. Ponderación de los criterios .....	15
Tabla 2.3. Matriz de decisión .....	16
Tabla 2.4. Requerimientos del sistema .....	16
Tabla 2.5. Consumo total del sistema .....	23
Tabla 2.6. Especificaciones de la batería .....	24
Tabla 2.7. Características de la llanta .....	25
Tabla 2.8. Criterios y condiciones de diseño.....	26
Tabla 2.9. Características del motor .....	27
Tabla 2.10. Parámetros de diseño del muelle .....	27
Tabla 2.11. Materiales de las capas .....	29
Tabla 2.12. Material para el robot .....	30
Tabla 2.13. Datos para el algoritmo PRM .....	34
Tabla 2.14. Acciones de Control para evador obstáculos .....	41
Tabla 2.15. Datos para el bloque <i>Pure Pursuit Controller</i> .....	43
Tabla 2.16. Parámetros del motor BLDC .....	45
Tabla 2.17. Características transientes iniciales .....	46
Tabla 2.18. Características transientes finales .....	47
Tabla 3.1. Componentes principales de la estructura del robot. ....	49
Tabla 3.2. Pesos de los dispositivos electrónicos seleccionados .....	52
Tabla 3.3. Presupuesto para la construcción del robot móvil .....	63

# CAPÍTULO 1

## 1. INTRODUCCIÓN

El servicio de entregas de comida en Ecuador se ofrece mayormente por aplicaciones como UberEats, Glovo, Rappi, PedidosYa, entre otras [1]. Muchos establecimientos de comida empezaron a adoptar esta modalidad, ya que representa una forma de aumentar las ganancias, además de la comodidad y ahorro de tiempo por parte del cliente.

En países como Estados Unidos y Alemania existen modalidades de entrega que hacen uso de robots móviles autónomos. Estos se asocian con establecimientos como Starbucks, PepsiCo, Domino's Pizza, entre otros. Su adaptación en la sociedad ha sido muy relevante, ya que se los utiliza en entornos como ciudades y universidades.

Por otro lado, de manera local no existen ese tipo de tecnologías. Además, el servicio de *delivery* se limita a realizar entregas alrededor de la ciudad y no en entornos pequeños como una universidad. Sin embargo, en estos lugares sí existe la necesidad de estos servicios, ya que estudiantes, profesores y personal administrativo cuentan con horarios restringidos que les dificulta trasladarse y esperar su orden, considerando que es común que estos establecimientos estén con una alta demanda de clientes.

Debido a esto, el presente proyecto propone el diseño de un robot móvil autónomo, capaz de navegar dentro de un campus universitario como el de la Espol, para hacer el servicio de entrega de alimentos. Para simular el proceso se presentará un modelo en MATLAB-Simulink, donde se integre la arquitectura de control con la dinámica del sistema mecánica, de tal forma que se pueda observar una entrega exitosa.

## 1.1 Descripción del problema

La industria de entregas de comida a domicilio o *delivery*, es uno de los servicios de mayor auge a consecuencia de la pandemia. Según [2], en Ecuador, los ingresos por *delivery* en 2021, fueron un 20 % superiores al 2020, con proyecciones del 53.12 % hasta el 2025. Esto se debe a que durante la pandemia, muchos negocios se reactivaron y lograron recuperar sus ventas gracias a la adopción del *delivery* [3]. Actualmente, existe una variedad de empresas que ofrecen este servicio, mediante una aplicación móvil. Es importante destacar, que el *delivery* no solo ayuda a las empresas a aumentar el alcance de sus productos, sino a buscar el confort de sus usuarios, que hacen pedidos ya sea por comodidad o por escaso tiempo libre [4].

Este servicio opera de forma zonificada, con repartidores atendiendo dentro de un cierto sector de la ciudad. Sin embargo, dentro de entornos más pequeños como un campus universitario, no es común ofrecer este servicio. Esto lleva a que en varios establecimientos, como los que atienden dentro de la Espol, se observe diariamente a una gran cantidad de clientes esperando para que los atiendan, les preparen sus pedidos y poder encontrar una mesa disponible. El problema radica en la pérdida de tiempo que esto conlleva a los usuarios, ya que los tiempos libres suelen ser muy cortos para hacer todo el proceso. Además, el factor del clima, como un sol intenso, provoca incomodidad a los usuarios y suele ser un impedimento para salir.

Por otro lado, los propietarios de estos locales prefieren evitar contratar personal para este servicio, debido a los gastos laborales y las bajas comisiones por entrega, al ser distancias más cortas. Ante esto, se necesita de un método de entregas eficiente y que no requiera de la supervisión de una persona.

## 1.2 Justificación del problema

El servicio de *delivery* en Ecuador es un proceso logístico que en la actualidad se lleva a cabo por medio de los usuarios de las plataformas de reparto, esta modalidad que hace algunos años se presentó como una alternativa para todo tipo de entregas, ahora representa una oportunidad de expansión para los establecimientos de comida. No obstante, durante la pandemia la continua actividad de este servicio puso en riesgo la salud tanto de repartidores como de aquellos clientes que recibían los pedidos. Siendo este el catalizador para la adopción de nuevas modalidades de entrega, aun después de la reducción de contagios y la llegada de las vacunas.

Cabe recalcar que esta modalidad con robots ya había sido implementada en otros países como muestra del avance tecnológico que se ha tenido en estos años [5]. Sin embargo, a raíz de la pandemia y la necesidad de precautelar el bienestar de la sociedad, las transformaciones han sido significativas y las entregas realizadas por robots se ha presentado como una solución que prevalece [6]. Dicho esto, la contribución principal de este proyecto es presentar una alternativa de reparto dentro de los establecimientos de alimentos y bebidas en los campus universitarios. Además de ser una solución que reduce la propagación de cualquier virus, adaptándose a ambientes externos y movilizandolos productos comestibles a un punto en específico.

De esta manera, la logística estaría en función de robots autónomos, lo que facilitaría el alcance de los productos dada la dificultad de movilidad ya sea por cuestiones climáticas o por falta de tiempo de los estudiantes, profesores y servidores de las universidades.

## **1.3 Objetivos**

### **1.3.1 Objetivo general**

Diseñar un robot móvil autónomo con ruedas utilizando técnicas de control y algoritmos de navegación para entregas de alimentos en el campus universitario de la ESPOL.

### **1.3.2 Objetivos específicos**

- Diseñar la base y estructura mecánica del robot, analizando el material y los esfuerzos involucrados, para que soporte las cargas generadas.
- Diseñar el sistema electrónico, mediante la selección de los componentes y fuentes de alimentación adecuados, para el funcionamiento acorde a los requerimientos.
- Diseñar el sistema de control, mediante el uso de algoritmos de navegación, para la planificación de trayectorias, seguimiento de ruta y evasión de obstáculos.
- Simular el modelo del robot en un entorno virtual, para la validación del funcionamiento, mostrando la entrega de una orden exitosa.



similar en el reparto de alimentos [12]. Inicia con la recolección del pedido, que es colocado en su interior por el trabajador que tiene acceso por medio de una aplicación, el recorrido hacia el punto de entrega, planificando la ruta y esquivando posibles obstáculos del camino, y la finalización del proceso de entrega, luego el cual, una vez concluido, el robot retorna nuevamente al punto de partida.

A continuación, en la Tabla 1.1 se presenta una evaluación comparativa de una entrega realizada por un robot y un humano, basada en factores relacionados a la alimentación y el servicio [13]. Las 3 marcas de verificación (✓✓✓) significan un buen rendimiento, las dos marcas (✓✓) indican un rendimiento medio y una marca (✓) evidencia un mal rendimiento:

**Tabla 1.1 Diferencias observadas entre un robot y un humano *delivery***

Factores	Rendimiento observado	
	Robot	Humano
Comida segura	✓✓✓	✓✓✓
Eficiencia del servicio	✓✓	✓✓✓
Sostenibilidad	✓✓✓	✓
Conveniencia	✓✓✓	✓✓
Seguridad	✓✓	✓
Disponibilidad	✓✓✓	✓

Cada criterio de la Tabla 1.1 responde a:

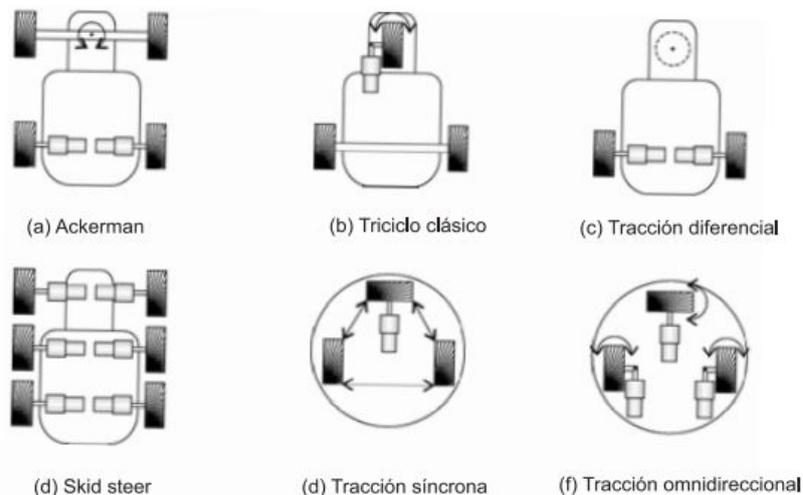
- **Comida segura:** Temperatura y envasado de los alimentos sin alteraciones por condiciones externas.
- **Eficiencia del servicio:** Tiempo que le toma realizar la entrega.
- **Sostenibilidad:** Libre de emisión de CO<sub>2</sub>.
- **Conveniencia:** Entrega realizada en el lugar acordado inicialmente y no en otro punto.
- **Seguridad:** Nivel de seguridad percibida en la entrega.
- **Disponibilidad:** Disposición de realizar la entrega completa y no declinar por muy cerca o muy poca comisión.

Por último, en entornos como universidades, estos robots se han convertido en un miembro de la sociedad y los suelen considerar como mascotas dada la interacción que pueden tener con los usuarios y peatones [14].

#### 1.4.2 Robots móviles con ruedas

Un robot móvil es un sistema mecatrónico capaz de moverse en su entorno de manera autónoma. Estos se utilizan en aplicaciones como exploraciones espaciales, misiones de rescate, reconocimiento de terreno, manufactura, transporte, entre otros. Los robots móviles se clasifican según su tipo de locomoción: por ruedas, patas y orugas [15]. La selección del tipo de locomoción del robot dependerá tanto de la aplicación, como de las restricciones del entorno.

Los robots móviles con ruedas (RMR) son los que han tenido mayor desarrollo en comparación con los otros. Esto se debe a las ventajas que presentan las ruedas en su diseño, como el ahorro energético, menor desgaste, uso de una menor cantidad de partes y facilidad de control [15]. Los RMR se clasifican de acuerdo con su configuración cinemática, en la Figura 1.2 se pueden observar diferentes tipos.



**Figura 1.2 Configuraciones cinemáticas de los RMR [15]**

Para aplicaciones de transporte en entornos urbanos se utilizan las siguientes configuraciones:

- **Ackerman:** La movilidad es similar a la utilizada en los vehículos a motor, el modelo cuenta con cuatro ruedas: las delanteras, que son orientables y se encargan del direccionamiento, mientras que las traseras, que son fijas y producen la tracción [16].
- **Skid Steer:** Esta configuración es considerada "todoterreno", gracias a la naturaleza robusta de su estructura mecánica. El modelo cuenta con tracción en cada rueda y el direccionamiento se logra mediante el control de la velocidad diferencial de las ruedas a cada lado [17].
- **Omnidireccional:** Este modelo puede moverse en cualquier dirección, sin tener que girar en el sentido de avance del robot. Esto le da mayor facilidad para navegar en entornos estrechos y dinámicos. Para esta configuración existen diferentes tipos de ruedas: castor, mecanum, ortogonales, universales y esféricas [18].

### 1.4.3 Sistema de navegación

El sistema de navegación es el que se encarga de guiar al robot, para que se traslade de forma independiente en su entorno de trabajo [19]. Este sistema debe incluir la planificación, localización, percepción y control.

La planificación de trayectorias se presenta en dos etapas: planificación global y planificación local. La primera consiste en la generación de rutas dentro de un mapa conocido, es decir, una aproximación al camino que tomará el robot hasta llegar a su objetivo; y la segunda, se resuelve en tiempo real de ejecución, y es la encargada de evadir las obstrucciones presentadas en el camino, ya sean objetos desconocidos u obstáculos dinámicos, para generar la ruta final que el robot debe seguir [20].

Para la percepción y localización se suelen utilizar diferentes sensores. A continuación se presenta una descripción de los más utilizados en robótica:

- **Sensores ultrasónicos:** Son sensores que miden la distancia hacia objetos, mediante el reflejo de pulsos ultrasónicos [21]. Los sensores ultrasónicos pueden detectar objetos a distancias de hasta 6 m; sin embargo, su ángulo de medición es de  $30^\circ$  y en implementaciones se los suele colocar en lotes para cubrir un mayor campo de visión [22].
- **Cámaras de profundidad:** Estos dispositivos además de generar imágenes normales del entorno, pueden producir imágenes con una perspectiva de profundidad, con lo que, mediante algoritmos de control, y sensores láser incorporados, se pueden obtener distancias reales a los objetos dentro del campo visual [22].
- **Encoder:** Son sensores que permiten obtener datos del desplazamiento y sentido de giro de las ruedas en robots móviles [23]. Estos datos se utilizan para estimar la velocidad real y el cambio de posición del robot con respecto a un marco de referencia.
- **GPS - IMU:** El GPS es un módulo de localización, que interactúa con satélites para obtener la lectura de la latitud y longitud en la que se encuentra [24]. Se lo suele utilizar junto con un sensor inercial IMU para estimar tanto la posición como la orientación del robot en entornos abiertos [25].

#### 1.4.4 Esquema de control

El diseño propuesto utiliza un sistema de control retroalimentado para asegurar el cumplimiento de la trayectoria trazada por el algoritmo de localización. En ese sentido, se plantea el uso de un controlador tipo proporcional, integral, derivativo (PID), que minimiza el error entre la trayectoria de referencia y la trayectoria recorrida.

La Figura 1.3 presenta la estructura de un controlador PID. El algoritmo se compone de tres componentes: proporcional (P), integral (I) y derivativa (D). El bloque proporcional realiza cambios proporcionales al error, para modificar la salida a una respuesta esperada, sin reducir el error a cero. Por otro lado, el término integral integra la variable de error, procesada en el tiempo y acumula este resultado para generar una compensación que elimina el error de estado estable a la salida. Por último, el bloque derivativo se encarga de revisar la tasa de cambio de la variable de proceso y modifica la salida al encontrar acciones inusuales. El ajuste de estos tres parámetros es decisión del diseñador, de manera que se pueda conseguir un rendimiento óptimo del sistema [26].

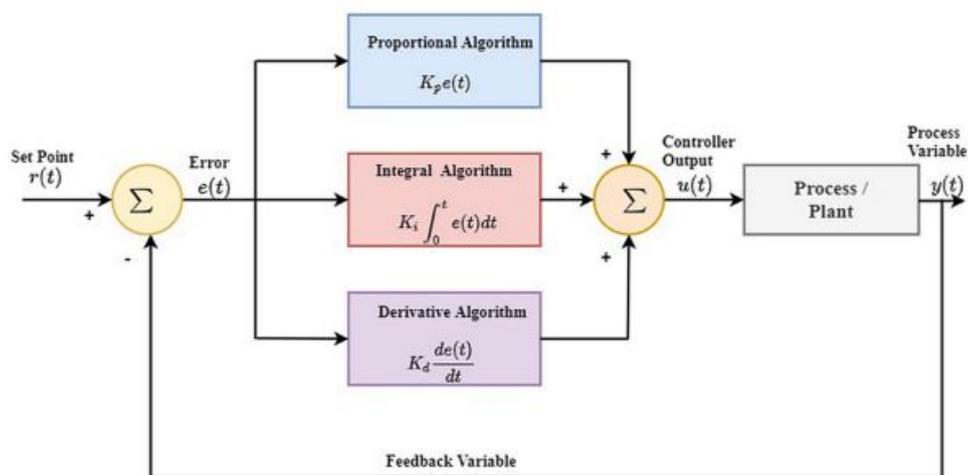


Figura 1.3 Esquema de control de un proceso con PID [26]

### 1.4.5 Estado del arte

Actualmente, en Ecuador no se encuentran robots autónomos destinados al servicio de entrega de comida, sin embargo, esta tecnología se ha ido desarrollando en otros países. Dentro de las soluciones existentes, está el robot delivery Starship, que posee una autonomía de 12 horas, realiza sus entregas en un radio de 6 km, tiene una capacidad de carga de 10 kg y una velocidad máxima de 6 km/h [27]. Se presenta como un robot tecnológico que puede ser vinculado a varios establecimientos de alimentos y bebidas para abastecer a la comunidad de estudiantes, profesores y servidores en un campus. Tal es el caso de la Universidad Purdue [5], ubicada en West Lafayette, Indiana EE. UU., que inició su operación con 30 robots en el año 2019.

El robot es monitoreado constantemente y se puede tomar control manual de emergencia si es necesario, aunque el 99% del tiempo viajan solos. Cuentan con 12 cámaras para su visión por computadora, usa inteligencia artificial (IA) para aprender más sobre el entorno, de forma continua [28]. Cabe recalcar que antes de comenzar las operaciones en una nueva ubicación, Starship escanea el entorno y crea un mapa virtual en 3D.



**Figura 1.4 Robots Starship en el campus de West Lafayette de la Universidad Purdue [5]**

En segunda instancia está el KiwiBot, un robot autónomo de cuatro ruedas desarrollado en Colombia y California, tiene hasta 6 horas de autonomía, trabaja en un radio de 10 km, posee una capacidad de carga de 8

kg y una velocidad de 3 km/h [29]. Su participación en los establecimientos de comida ha sido clave, pues ha logrado posicionarse en escuelas como la Universidad Estatal de Nuevo México, la Universidad de Gonzaga de Chicago, entre otras.

El robot utiliza una combinación de visión por computadora y GPS para conocer su ubicación, tomar decisiones en la planificación de rutas y la evasión de obstáculos [30]. Además, puede interactuar con las personas a través de parlantes y una pantalla con ojos parpadeantes. También cuentan con el apoyo remoto de supervisores humanos.



**Figura 1.5 KiwiBot en el campus de la Universidad de Gonzaga. [31]**

Dentro del mercado también está Robby 2, un robot autónomo de seis ruedas también llamado *snackbot*, que opera para *Hello Goodness*, una filial del gigante alimenticio PepsiCo, y que en la actualidad funciona en la Universidad del Pacífico en California [32], donde realiza entregas en el campus de Stockton. Dentro de sus especificaciones el robot tiene la capacidad de recorrer 32 km, su distancia de trabajo es de 16 km, su capacidad de carga está en el rango de los 8 a 12 kg y tiene una velocidad de 4 km/h. Es impulsado por un destacado sistema de visión por computadora y tecnología IA [33].



**Figura 1.6 Robot móvil Robby 2 [34]**

Estos tres robots ayudan a repartir los pedidos en varios campus universitarios y están equipados con una bandera a la altura de la vista humana, lo que les permite ser identificados a la hora de transitar junto a los peatones. Es así como estos robots son referentes para proyectos innovadores en el ámbito logístico, realizando entregas locales de alimentos, bebidas, encomiendas, medicinas, etc.

# CAPÍTULO 2

## 2. METODOLOGÍA

En este capítulo se presentan las diferentes etapas del diseño. Empezando por las consideraciones conceptuales de la solución, mediante el planteamiento de alternativas de robots móviles con ruedas, siguiendo con la identificación de los requerimientos técnicos, y posteriormente, analizando todo lo concerniente con el diseño mecánico, electrónico y de control necesarios para la obtención del modelo final.

### 2.1 Concepto de la solución

#### 2.1.1 Alternativas de solución

Con base en la problemática, se presentaron tres posibles alternativas de solución para el tipo de robot móvil con ruedas, considerando la cinemática y estructura mecánica: Robot Ackerman, Robot Omnidireccional con ruedas tipo Mecanum y Robot tipo Skid-Steer. Para la selección de la mejor alternativa, se hizo una evaluación que considera los criterios de la Tabla 2.1 y sus ponderaciones en la Tabla 2.2.

**Tabla 2.1 Criterios de evaluación**

<b>Criterio</b>	<b>Descripción</b>
<b>Movilidad</b>	Capacidad de cambiar de dirección y traslado con mayor precisión, para el seguimiento de rutas y evasión de obstáculos.
<b>Consumo</b>	Menor gasto energético en la cantidad de motores y drivers requeridos.
<b>Precio</b>	Implementación viable, a un precio moderado y disponibilidad de los componentes de manera local.
<b>Adaptabilidad</b>	Facilidad de movimiento en terrenos con irregularidades.
<b>Estabilidad</b>	Capacidad de soportar y mantener estable cierta cantidad de carga durante el traslado.
<b>Mantenimiento</b>	Facilidad de desmontaje para revisiones, instalación de repuestos y limpieza.

**Tabla 2.2 Ponderación de los criterios**

<b>CRITERIOS DE SELECCIÓN</b>			
<b>Peso</b>	<b>Criterio</b>	<b>Prioridad</b>	<b>% de decisión</b>
3	Movilidad	1	27
2.3	Mantenimiento	2	21
2.3	Consumo	2	21
1	Precio	4	9
1.2	Adaptabilidad	3	11
1.2	Estabilidad	3	11
11	<b>Total</b>		100

En la Tabla 2.3 se presenta la matriz de decisión, donde se evaluaron las diferentes opciones. Es evidente que aquella que más se ajusta a los criterios, fue la del robot Skid Steer con una puntuación de 24.4, siendo este el concepto seleccionado para utilizarse en el diseño de la solución.

**Tabla 2.3 Matriz de decisión**

Pesos	CRITERIOS						RESULTADOS		
	3	2.3	2.3	1	1.2	1.2	Puntajes sin peso	Puntaje con peso	Prioridad
Opciones	Movilidad	Mantenimiento	Consumo	Precio	Adaptabilidad	Estabilidad			
Alternativa 1	1	3	3	3	1	2	13	23.4	2
Alternativa 2	2	2	2	2	3	3	14	24.4	1
Alternativa 3	3	1	2	1	2	2	11	21.7	3

### 2.1.2 Requerimientos de diseño

Los requerimientos que debe cumplir el robot móvil para hacer entregas de comida en un campus universitario como el de la ESPOL se presentan en la Tabla 2.4:

**Tabla 2.4 Requerimientos del sistema**

<b>Autónomo</b>	Capaz de planificar trayectorias, seguir la ruta y evitar obstáculos.
<b>Entorno</b>	El robot es de exteriores y se traslada en terrenos irregulares. Debe ser capaz de identificar rampas para discapacitados y subirlas.
<b>Material</b>	Se espera un chasis resistente al sol, humedad y golpes. Debe soportar temperaturas de 18 a 33 °C.
<b>Compartimento</b>	Modular y ajustables si así lo requiere el pedido de café y postres para que se mantenga estable.
<b>Capacidad</b>	La carga máxima soportada en el compartimento debe ser 6 kg.
<b>Tamaño</b>	Adaptable a las rampas dentro del campus, con dimensiones no superiores a 500x600x500 mm.
<b>Alimentación</b>	Baterías con facilidad de carga y con alta autonomía.

### 2.1.3 Diseño conceptual

En la Figura 2.1 se observa el concepto de la solución planteada. El modelo es un robot móvil autónomo de cuatro ruedas. En la parte frontal cuenta con sensores ultrasónicos para la detección de obstáculos y una cámara de profundidad para el reconocimiento de escaleras y rampas en el camino. Además, tiene un GPS para el posicionamiento y una IMU para la orientación. La estructura mecánica se basa en el robot Skid-Steer, el cual es un robot diferencial robusto con cuatro ruedas actuadas. Además, se tiene un compartimento con materiales de aislamiento térmico, para la conservación y transporte de los alimentos.

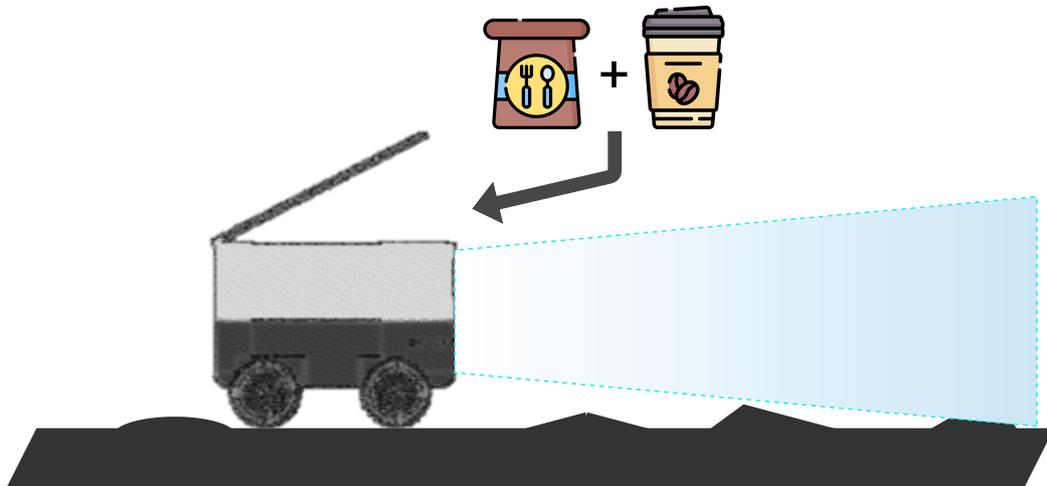


Figura 2.1 Concepto de la solución

## 2.1.4 Metodología

En la Figura 2.2 se presenta un diagrama con las diferentes etapas del diseño para el desarrollo de la solución.

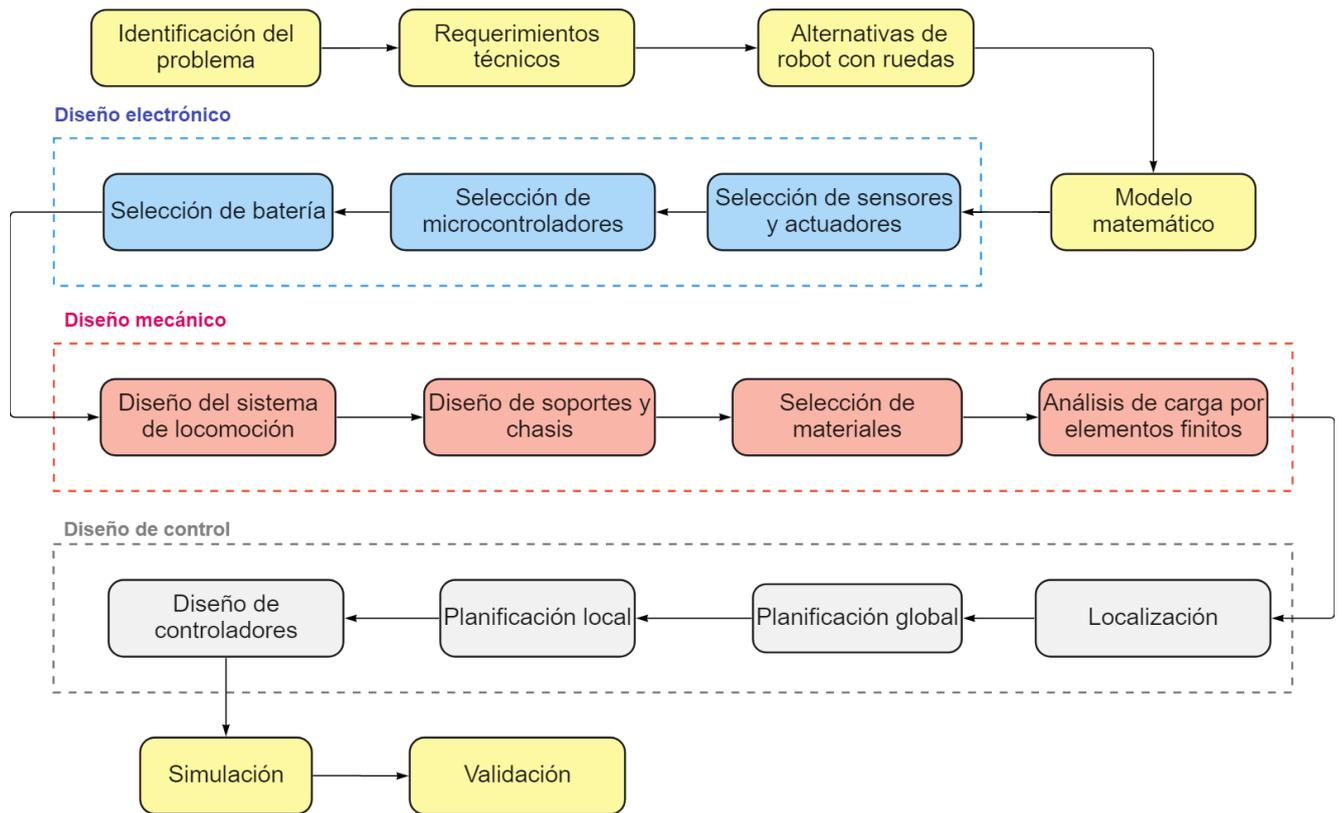


Figura 2.2 Metodología del diseño

## 2.2 Modelos matemáticos

### 2.2.1 Cinemática directa

La cinemática del robot Skid Steer es del tipo diferencial, la cual toma las velocidades angulares de las ruedas como entradas de control para obtener un movimiento deseado con respecto a un marco inercial.

En la Figura 2.3 se observa el diagrama cinemático del sistema. Dado un marco de coordenadas inercial  $\{O\}$ , la posición del robot se representa por el marco local  $\{B\}$ , donde la velocidad lineal en el eje  $x$  tiene un valor de  $v$ , mientras que en el eje  $y$  se aproxima a cero, debido a la condición de no deslizamiento. La pose del robot se representa por las coordenadas  $(x, y, \theta)$  en el marco  $\{B\}$ .

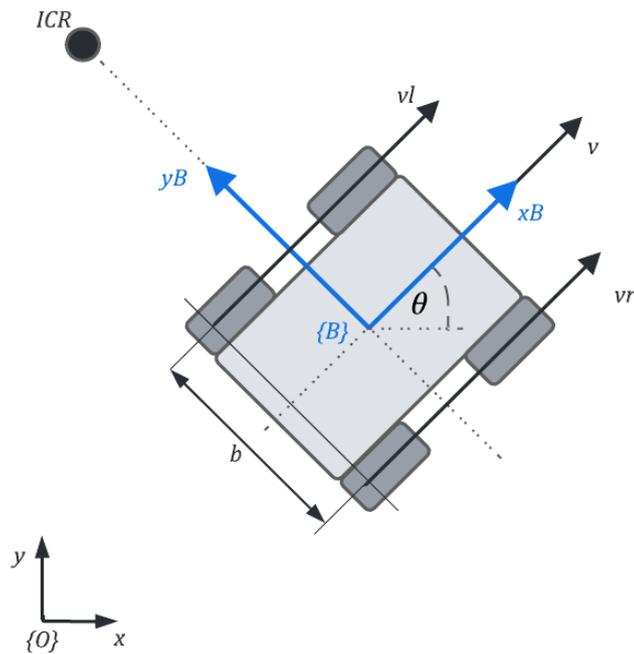


Figura 2.3 Esquema cinemático del robot

Las ruedas de cada lado giran a la misma velocidad y tienen velocidades lineales  $v_l$  y  $v_r$ . Estas se relacionan con las entradas de control  $w_l$  y  $w_r$  mediante el radio de las ruedas  $r$ . En las Ecs. 2.1 y 2.2, se presenta el modelo de cinemática directa según [35].

$$v = \frac{v_r + v_l}{2} = \frac{r(w_r + w_l)}{2}, \quad (2.1)$$

$$w = \frac{v_r - v_l}{b} = \frac{r(w_r - w_l)}{b}, \quad (2.2)$$

### 2.2.2 Cinemática inversa

Para los algoritmos de planificación, es necesario poder calcular las entradas de control ( $w_l$  y  $w_r$ ) a partir de una velocidad lineal y una velocidad angular deseada. Con las Ecs. 2.1 y 2.2 se obtiene el modelo de cinemática inversa en las Ecs. 2.3 y 2.4

$$w_l = \frac{v - (b/2)w}{r} \quad (2.3)$$

$$w_r = \frac{v + (b/2)w}{r} \quad (2.4)$$

### 2.2.3 Modelo de suspensión

Existen sistemas de suspensión que varían de acuerdo al tipo de control que poseen [36], activa, pasiva y semi-activa. El sistema de suspensión pasiva es el que mantiene sus parámetros predeterminados sin regulación automática, siendo el sistema que se ajusta al robot.

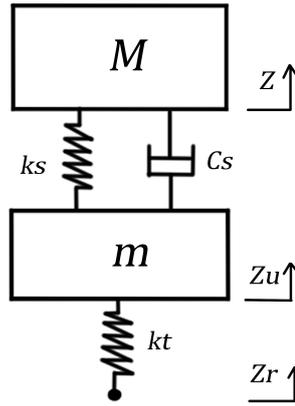


Figura 2.4 Modelo de suspensión pasiva.

La Figura. 2.4 muestra el modelo de suspensión pasiva donde  $M$  es la masa suspendida (chasis),  $m$  es la masa no suspendida (neumático),  $k_s$  es la rigidez de la suspensión,  $k_t$  la rigidez del neumático y  $C_s$  es el coeficiente de amortiguamiento de la suspensión. Otras variables que intervienen en el movimiento son: el desplazamiento debido a las irregularidades del camino  $Z_r$ , el desplazamiento vertical del neumático  $Z_u$  y el desplazamiento vertical del chasis  $Z$ .

Para el análisis teórico, un modelo de un cuarto de vehículo es generalmente usado [37,38]. La representación dinámica del sistema de suspensión de espacio de estados es expresada con las ecs. 2.5 y 2.6.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(k_s + k_t)}{m} & \frac{-C_s}{m} & \frac{k_s}{m} & \frac{C_s}{m} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{M} & \frac{C_s}{M} & \frac{-k_s}{M} & \frac{-C_s}{M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_t}{m} \\ 0 \\ 0 \end{bmatrix} Z_r \quad (2.5)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} Z_r \quad (2.6)$$

## 2.3 Diseño electrónico

El sistema electrónico del robot se compuso de placas de control, sensores, motores y elementos de potencia. En esta sección se presenta una estimación del consumo de potencia de los componentes seleccionados en el Apéndice A, además de la selección de la batería que más se ajusta al sistema.

### 2.3.1 Consumo de potencia

El sistema requiere tres niveles de voltaje: 36V para los motores, 9V para el Arduino y 5V para la computadora a bordo, servomotor y sensores. Para lograr esto, se planteó el uso de reguladores DC-DC, los cuales permiten la conversión de voltaje a una corriente de salida inferior a 3A, con un 92% de eficiencia.

En la Tabla 2.5 se presenta una estimación del consumo de potencia del sistema. En el caso de los motores, se tiene el consumo que tendrían en el escenario donde el robot sube una rampa de 30° durante todo su recorrido. Por otro lado, para el consumo de los reguladores, se consideraron pérdidas del 8% de la corriente transmitida en cada uno de los canales, siendo 0.02A en la reducción 36V - 9V y 0.23A en la reducción 9V - 5V. Para el resto de componentes se utilizaron los valores de sus respectivas tablas presentadas en el Apéndice A.

**Tabla 2.5 Consumo total del sistema**

Componente	Cantidad	Voltaje [V]	Corriente [A]	Potencia [W]
Jetson Nano	1	5	2	10
Arduino Mega	1	9	0.25	2
Sensor ultrasónico	9	5	0.015	0.7
IMU	1	5	0.0035	0.02
Controlador BLDC	4	36	0.1	14
Motor BLDC	4	36	6	864
Servomotor	1	5	0.7	4
Convertidor DC-DC	1	36	0.02	0.7
Convertidor DC-DC	1	9	0.23	2
<b>Total</b>				897

### 2.3.2 Selección de la batería

Entre las alternativas de baterías recargables en el mercado, hay modelos de iones de litio, níquel-cadmio, níquel-metal hidruro, ácido de plomo y polímero de litio. Para este proyecto se consideró una batería de iones de litio, al ser la alternativa con mayor densidad energética, larga vida útil y una alta eficiencia energética [39].

Para el proyecto se estableció una batería de 36V con una capacidad de 100 Ah. El modelo elegido, Figura 2.5, cuenta con un sistema BMS, un circuito de protección ante la sobrecarga, descarga inferior al voltaje de corte, cortocircuitos y sobrecalentamiento. En la Tabla 2.6 se encuentran sus especificaciones técnicas.



**Figura 2.5 Batería de Li-ion 36V 100 Ah [40]**

**Tabla 2.6 Especificaciones de la batería [40]**

Especificaciones	Datos
Voltaje de funcionamiento	36 V
Capacidad	100 Ah
Tipo	10S Lithium
Voltaje pico	42 V
Voltaje de corte	28 V
Temperatura de operación	-20 a 60 °C
Dimensiones	230x220x140 mm
Peso	14 Kg

### 2.3.3 Cálculo de la autonomía

En la ec. 2.8 se presenta el cálculo del tiempo de descarga  $h_b$  de la batería, donde  $P_b$  es la potencia por hora de la batería y  $P_s$  la potencia del sistema obtenida de la Tabla 2.5, considerando un escenario donde el robot sube rampas en todo su recorrido.

$$P_b = V_b \cdot I_{hb}, \quad (2.7)$$

$$P_b = (36)(100) = 3600 \text{ Wh}$$

$$h_b = \frac{P_b}{P_s}, \quad (2.8)$$

$$h_b = \frac{3600}{897} = 4.01 \text{ h}$$

A pesar de tener un modelo con un circuito de protección ante la sobredescarga, se consideró un factor de seguridad  $f_{sb}$  en la autonomía, para mantener la batería a niveles superiores al voltaje de corte. Esta acción evita daños en las celdas al descargar totalmente la batería, aumentando su vida útil. En la ec. 2.9 se calcula el tiempo de funcionamiento recomendado  $h_{rec}$  con un  $f_{sb}$  de 1.25, para un consumo de hasta el 80 % de su capacidad.

$$h_{rec} = \frac{h_b}{f_{sb}}, \quad (2.9)$$

$$h_{rec} = \frac{4.01}{1.25} = 3.21 \text{ h}$$

## 2.4 Diseño mecánico

En esta sección se realizó el cálculo y selección de componentes mecánicos, con el fin de desarrollar una solución que se ajuste a las condiciones físicas y cumpla con los requerimientos previamente definidos en la Tabla 2.4.

### 2.4.1 Selección de motor

Previo al dimensionamiento del motor se seleccionó la llanta, donde se consideró la condición del terreno y también su capacidad de carga. Para el caso de un terreno irregular, se eligió una llanta todo terreno sin espuma de característica modular.



Figura 2.6 Llantas seleccionadas para el robot *delivery* [41]

Tabla 2.7 Características de la llanta

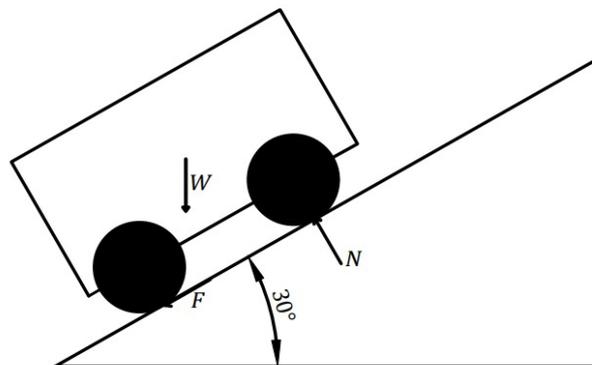
Especificaciones	Datos
Diámetro exterior	0.2 m
Diámetro del buje	0.024 m
Peso	0.65 kg
Carga admisible	16 kg

Luego de la selección de las llantas, se estableció la Tabla 2.8 que muestra los criterios y condiciones para el dimensionamiento del motor [42, 43]:

**Tabla 2.8 Criterios y condiciones de diseño**

Parámetro	Valor
Masa del robot	40 kg
Peso del robot	392.4 N
Peso en cada rueda motriz (carga normal)	98.1 N
Radio de la rueda	0.1 m
Velocidad máxima deseada	5 km/h
Tiempo de aceleración deseado	5 s
Densidad del aire a 33°	1.153 kg/m <sup>3</sup>
Coefficiente de arrastre de forma (frontal)	1.05
Factor por pérdidas de fricción	1.12
Coefficiente de fricción (adoquín)	0.70
Coefficiente de resistencia a la rodadura (adoquín)	0.017

Se calcularon las fuerzas opuestas comunes que se espera que actúen contra el vehículo (en condiciones de funcionamiento máximas) y se definen a partir de la Figura 2.7.



**Figura 2.7 Diagrama de cuerpo libre de vehículo subiendo una pendiente.**

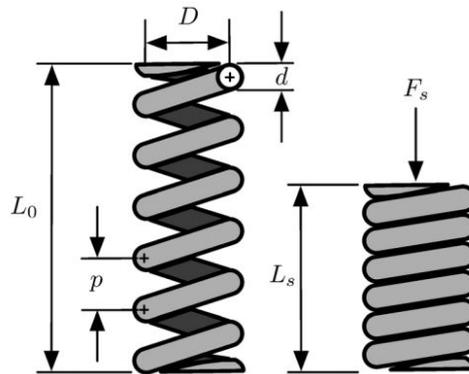
El proceso del análisis se detalla en el Apéndice B y los resultados se muestran en la Tabla 2.9, el cual presenta las características para la selección del motor.

**Tabla 2.9 Características del motor**

Especificaciones	Datos
Torque	6.56 Nm
Velocidad	152 RPM
Potencia	104.2 W

### 2.4.2 Diseño del muelle de suspensión

El sistema de suspensión es el encargado de absorber las irregularidades de la trayectoria por las que circulará el robot en el campus. Por ende, en esta sección se muestra el diseño el resorte.



**Figura 2.8 Muelle de compresión helicoidal. [44]**

Según la metodología del capítulo 10 de diseño mecánico de resortes en [45] y en [44], se desarrolló el proceso de diseño en el Apéndice C para un muelle de suspensión. Los parámetros de diseño se muestran en la Tabla 2.10:

**Tabla 2.10 Parámetros de diseño del muelle**

Parámetro	Datos
Paso	6 mm
Diámetro de hélice	30 mm
Diámetro de alambre	4 mm
Número de bobinas activas	14

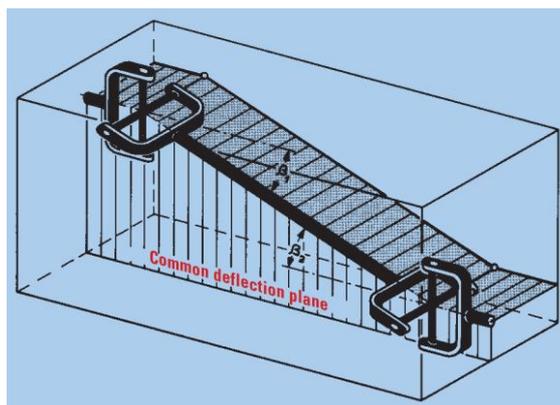
Basado en ese proceso, la Figura 2.9 muestra el resultado final con una constante elástica de 223 N/m.



**Figura 2.9 Muelle de compresión helicoidal diseñado.**

### **2.4.3 Diseño de eje cardán**

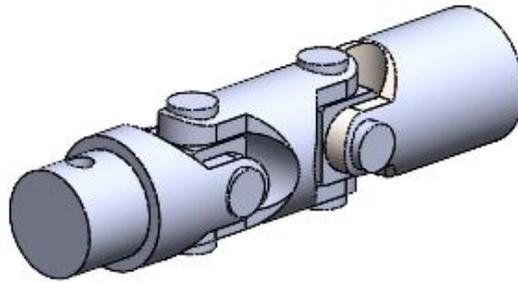
La transmisión de la potencia desde el motor hasta la rueda se realiza, en la mayoría de los casos, mediante ejes. El robot al ser de naturaleza dinámica presenta cambios en la alineación de la entrada y salida del eje, es decir, no se encuentran en el mismo plano por los cambios que genera la suspensión, como se demuestra en la Figura 2.11.



**Figura 2.10 Configuración en Z: eje cardan de entrada y salida en paralelo. [46]**

El eje cardán proporciona esa dinámica por medio de su dos juntas universales que permiten la transferencia de potencia en diferentes ángulos. La ventaja de utilizar las juntas en U es que pueden soportar cierto nivel de deslizamiento y tienen menos necesidad de lubricación [47], además de

ser un mecanismo de transferencia de velocidad constante. El detalle del diseño se encuentra en el Apéndice D.

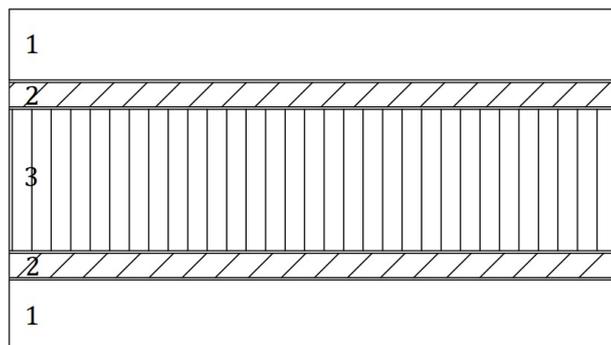


**Figura 2.11** Diseño del eje tipo cardán

#### 2.4.4 Selección de materiales

- **Contenedor de aislamiento térmico**

Dado que la bolsa térmica está destinada a colocarse dentro del robot de entrega, los materiales serán flexibles para que se adapte con mayor facilidad [48]. La Figura 2.12 ilustra esquemáticamente las capas de la bolsa aislada, de acuerdo con un aspecto diseño.



**Figura 2.12** Sección transversal esquemática de las capas del contenedor

**Tabla 2.11** Materiales de las capas de la Figura 2.12

N°	Capa	Material
1	Interna	Alfombrilla aislante de Aerogel
2	Intermedia	Polímero recubierto de aluminio
3	Externa	Nylon antibacteriano

La transferencia de calor puede ocurrir a través de la conducción, la convección y la radiación [49]. Las tres capas que se muestran en la Figura 2.12 están optimizadas para minimizar la transferencia de calor a través de esos tres canales. Los materiales con menor conductividad térmica transmiten menos calor a través de ellos y son mejores aislantes. Por lo tanto, para determinar el material con mayor aislamiento se calcula la conductividad donde se va a determinar la tasa de flujo de calor a través de las capas, se puede cuantificar mediante la siguiente relación:

$$q = \frac{kA(T_2 - T_1)}{d} \quad (2.10)$$

donde  $q$  es la tasa de flujo de calor (calor conducido por unidad de tiempo, medido en vatios),  $k$  es la conductividad térmica del material en cuestión,  $A$  es el área,  $T_2$  y  $T_1$  son las temperaturas respectivas del exterior y del interior del material, y  $d$  es su espesor. Por ende, los materiales con menor conductividad térmica dejan pasar menos calor a través de ellos, y son mejores aislantes.

#### ■ Cubierta del robot

Para la selección del material en el diseño mecánico se consideraron las propiedades de los posibles materiales, ya que el ABS y ASA son adecuadas para aplicaciones en exteriores [50, 51]. Las propiedades a considerar con mayor relevancia son su disponibilidad en el país, su densidad y su resistencia al calor.

**Tabla 2.12 Material para el robot**

Propiedad	ASA	ABS
Densidad	1070 kg/m <sup>3</sup>	1030 kg/m <sup>3</sup>
Temperatura máx.	83°	75-80°
Conductividad térmica	0.16-0.21 W/(m·K)	0.2256 W/(m·K)
Calor específico	1300 J/(kg·K)	1386 J/(kg·K)
Disponibilidad	Poca	Mucha

Aunque el ABS y el ASA tienen propiedades muy parecidas, la disponibilidad es un factor a considerar por lo que se eligió el material ABS por esta apreciable diferencia. Además, al ser un robot de tamaño considerable, en

la medida que aumentan las dimensiones se vuelve más pesado por lo que el ABS resultó una opción muy competitiva.

### 2.4.5 Análisis de carga por elementos finitos

Con el análisis estático del Apéndice E se obtuvo un factor de seguridad de 21, sin embargo, para un análisis de resistencia de material precisa de un estudio de fatiga. El resultado obtenido en la Figura 2.13 con respecto a los daños es de máximo 10%, dando un buen indicio de resistencia.

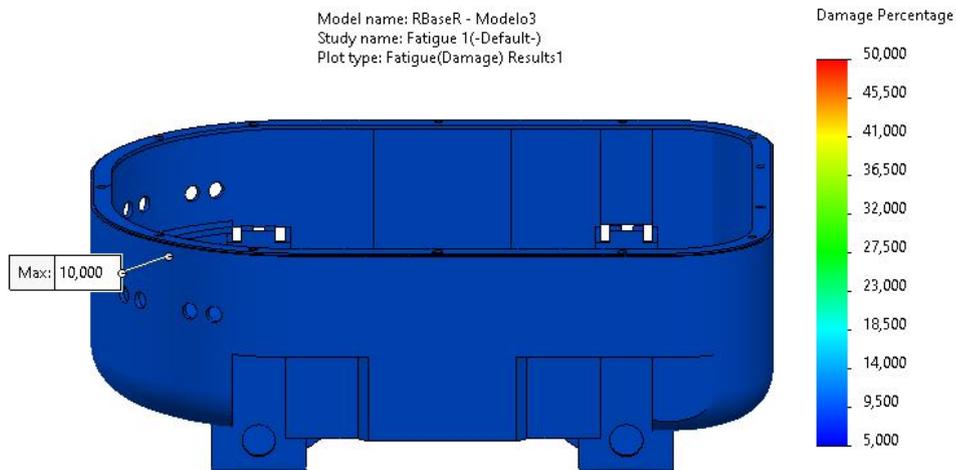


Figura 2.13 Porcentaje de daño en el estudio de fatiga.

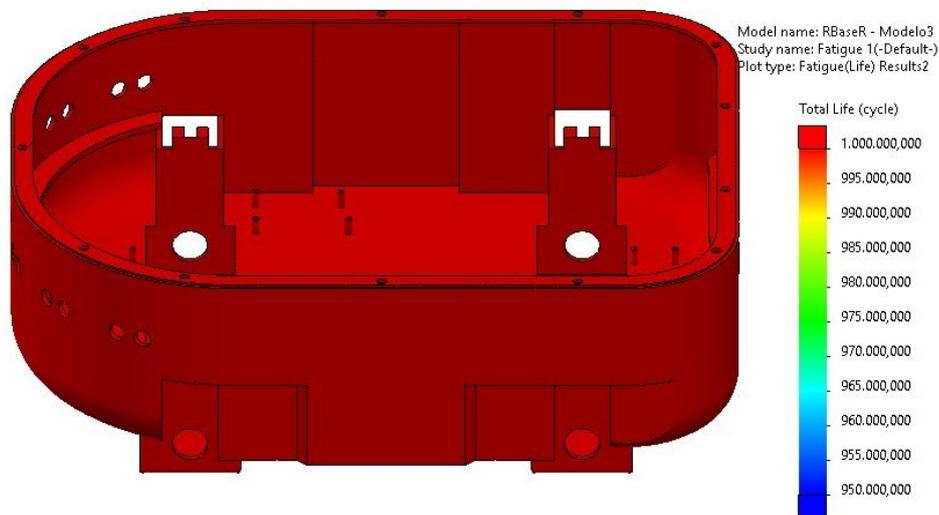
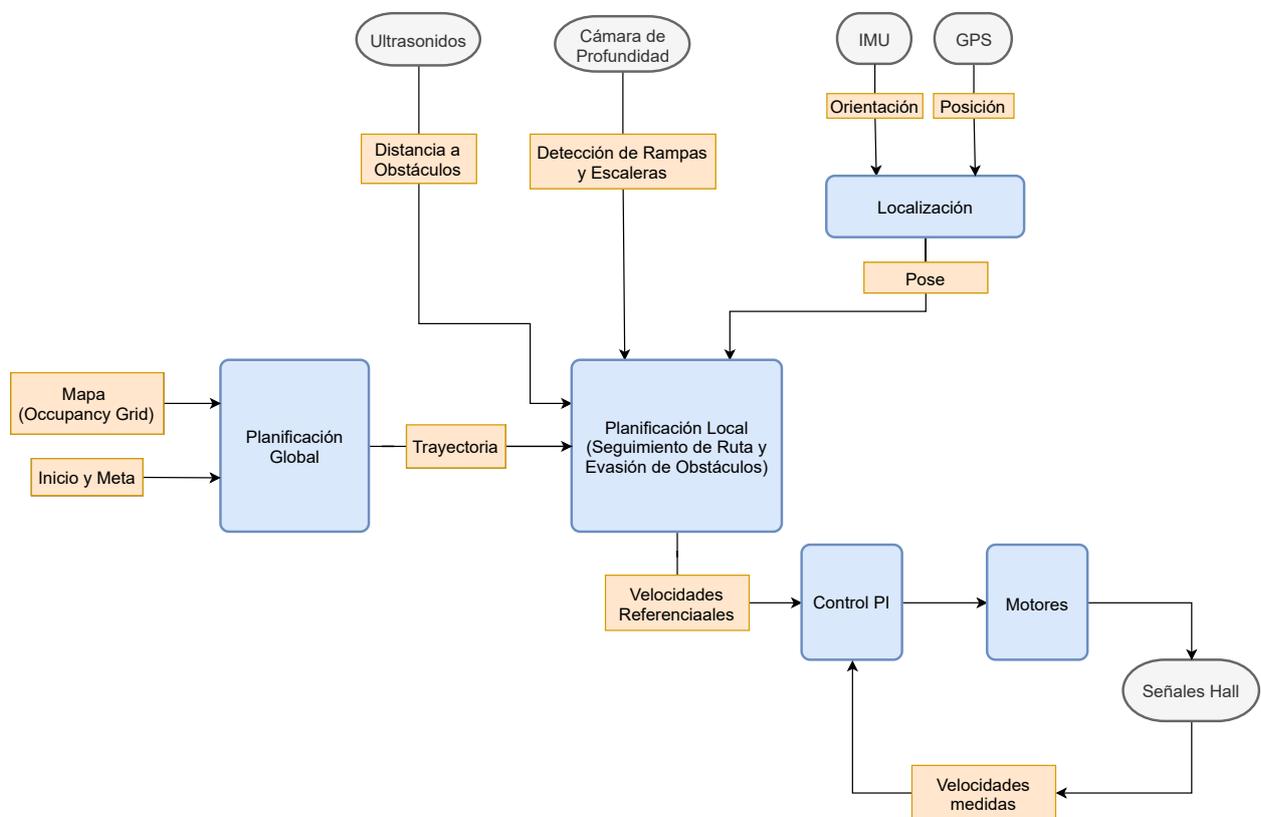


Figura 2.14 Vida útil obtenida en el estudio de fatiga.

Con respecto a la vida útil de la pieza, el resultado que muestra la Figura 2.14 presenta un fallo a partir del ciclo número 955,000.00 de un total de 1,000,000.00 ciclos. Siendo una vida total muy alta, por lo que la pieza puede durar un tiempo considerable.

## 2.5 Diseño de control

Para cumplir una entrega exitosa, el robot debe poder planificar y seguir una ruta, entre dos puntos. Para esto, se requiere conocer la pose del robot durante todo el traslado, además de la información del entorno proveniente de los sensores. En la Figura 2.15 se presenta la arquitectura de control del robot, donde se establecen las diferentes etapas con los algoritmos y controladores a desarrollar.



**Figura 2.15 Arquitectura de control**

Para la simulación el diseño se eligió utilizar MATLAB y Simulink, debido a que presentan diferentes herramientas de robótica que facilitan la implementación de diferentes algoritmos y el diseño de controladores.

### 2.5.1 Planificación global

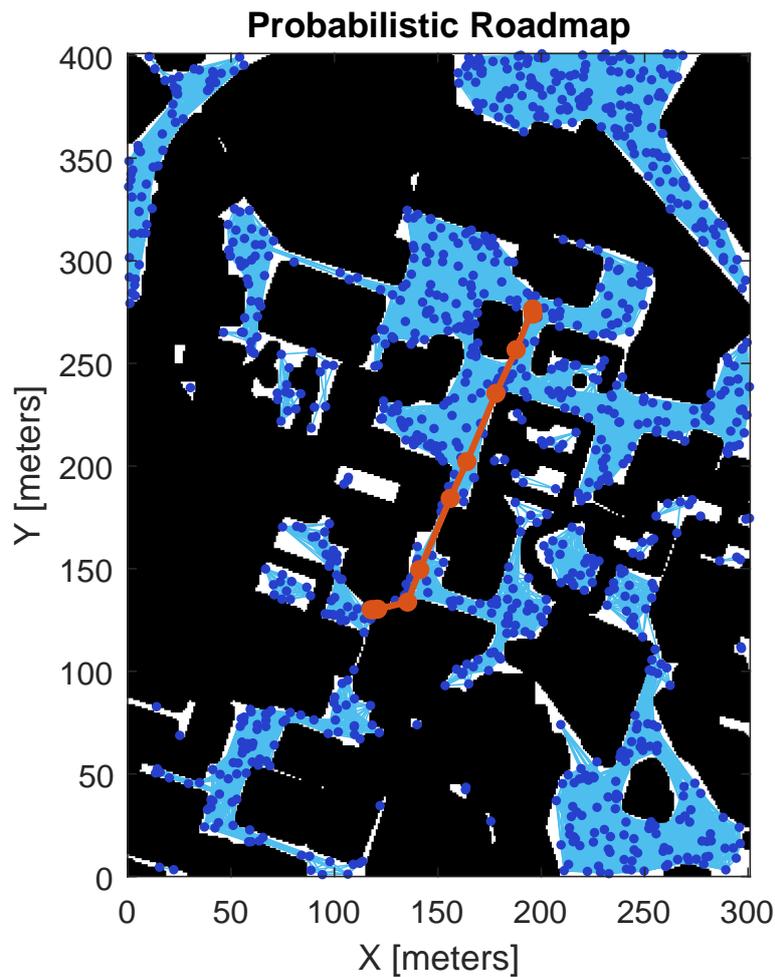
La ruta global se obtuvo utilizando el algoritmo PRM (*Probabilistic Roadmap*), un método de planificación de trayectorias que ubica una cantidad de nodos  $N$  de manera aleatoria en el espacio libre de un mapa discreto. Estos se conectan a una distancia  $d$  y los grafos resultantes se utilizan para identificar la mejor trayectoria entre un punto inicial  $P_{init}$  y una meta  $P_{goal}$  [52]. Los obstáculos del mapa son las celdas con valor 1 y se representan con color negro. Una medida de seguridad en este diseño, es aumentar el espacio ocupado por los obstáculos con un factor  $i_{nf}$ .

Para la implementación se utilizó la librería de Robotic System Toolbox de MATLAB [53], con los datos de la Tabla 2.13, además del mapa de la Figura F.3, cuyo proceso de obtención se detalla en el Apéndice E.

**Tabla 2.13 Datos para el algoritmo PRM**

Propiedad	Dato
$N$	1000
$d$	50
$i_{nf}$	2.2
$P_{init}$	(119, 130)
$P_{goal}$	(196, 274)

Con los datos definidos, se obtuvo la ruta de una posible entrega, mostrada en la Figura 2.16, siendo esta la utilizada para la simulación en el ambiente 3D con el modelo CAD diseñado.

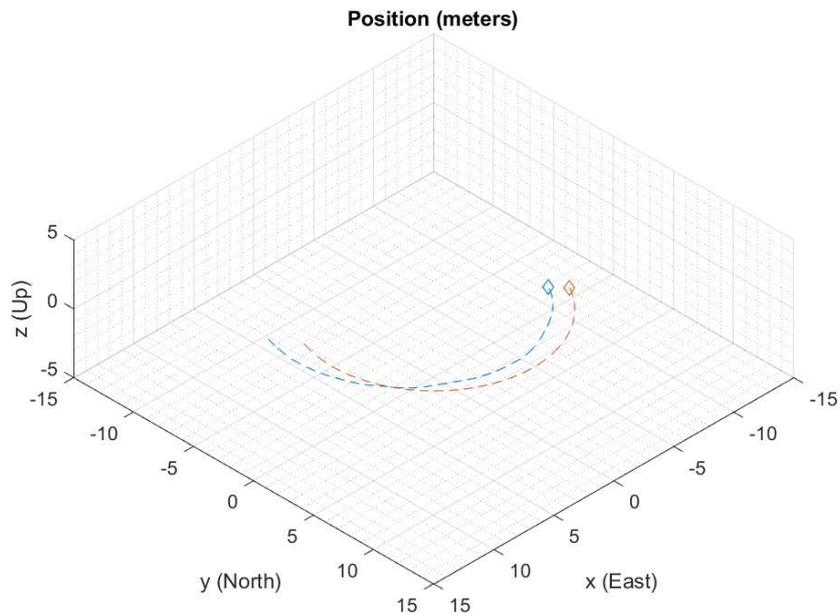


**Figura 2.16 Planificación global con PRM**

### 2.5.2 Localización en el mapa

La localización del robot se planteó con el uso de una GPS y una unidad de medición inercial (IMU) de 6 grados de libertad. En base a [35], al implementar estos sensores, se debe utilizar un algoritmo de fusión sensorial con un filtro extendido de Kalman [54]. Este consiste en filtrar los errores de precisión que presentan ambos sensores al trabajar de manera individual. Los GPS pueden dar posiciones con un error de hasta 10 m, mientras que la IMU tiene el problema del sesgo, el cual aumenta con el tiempo de forma lineal.

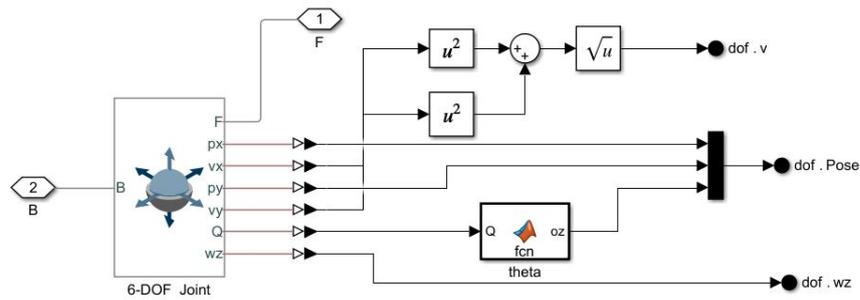
Para verificar esto, se utilizó el toolbox *Sensor Fusion and Tracking* de MATLAB [55], ingresando la frecuencia, datos de tolerancia y rango de los sensores propuestos, para simular la estimación de la pose durante una trayectoria circular. En la Figura 2.17 se presenta el resultado, donde la línea azul representa la trayectoria estimada con los sensores, y la roja la trayectoria real que sigue el robot.



**Figura 2.17 Comparación de la localización con EKF y la real**

En la gráfica se observa que la pose estimada no es exacta; sin embargo, se puede notar una reducción en el error del GPS, ya que el modelo a utilizar tiene un error de 2.5m en cada eje y, en el resultado, la diferencia es de máximo 1m. Por otro lado la orientación tiene mayor precisión, al lograr la forma deseada.

Este método es necesario en una implementación física, sin embargo, para la simulación se utilizaron los sensores virtuales provenientes del bloque 6DOF Joint de Simscape. Este es utilizado para generar el movimiento del robot en el marco global. En la figura 2.18 se presentan las conexiones del bloque.



**Figura 2.18 Sensores virtuales del modelo en Simulink**

### 2.5.3 Planificación local

Esta etapa se ejecuta en movimiento y consiste en el procesamiento de los datos de los sensores, para el seguimiento de ruta y la evasión de obstáculos ocultos en el mapa. En la Figura 2.19 se presenta el diagrama de flujo general del comportamiento del robot. En las siguientes secciones se describen los algoritmos utilizados en esta etapa.

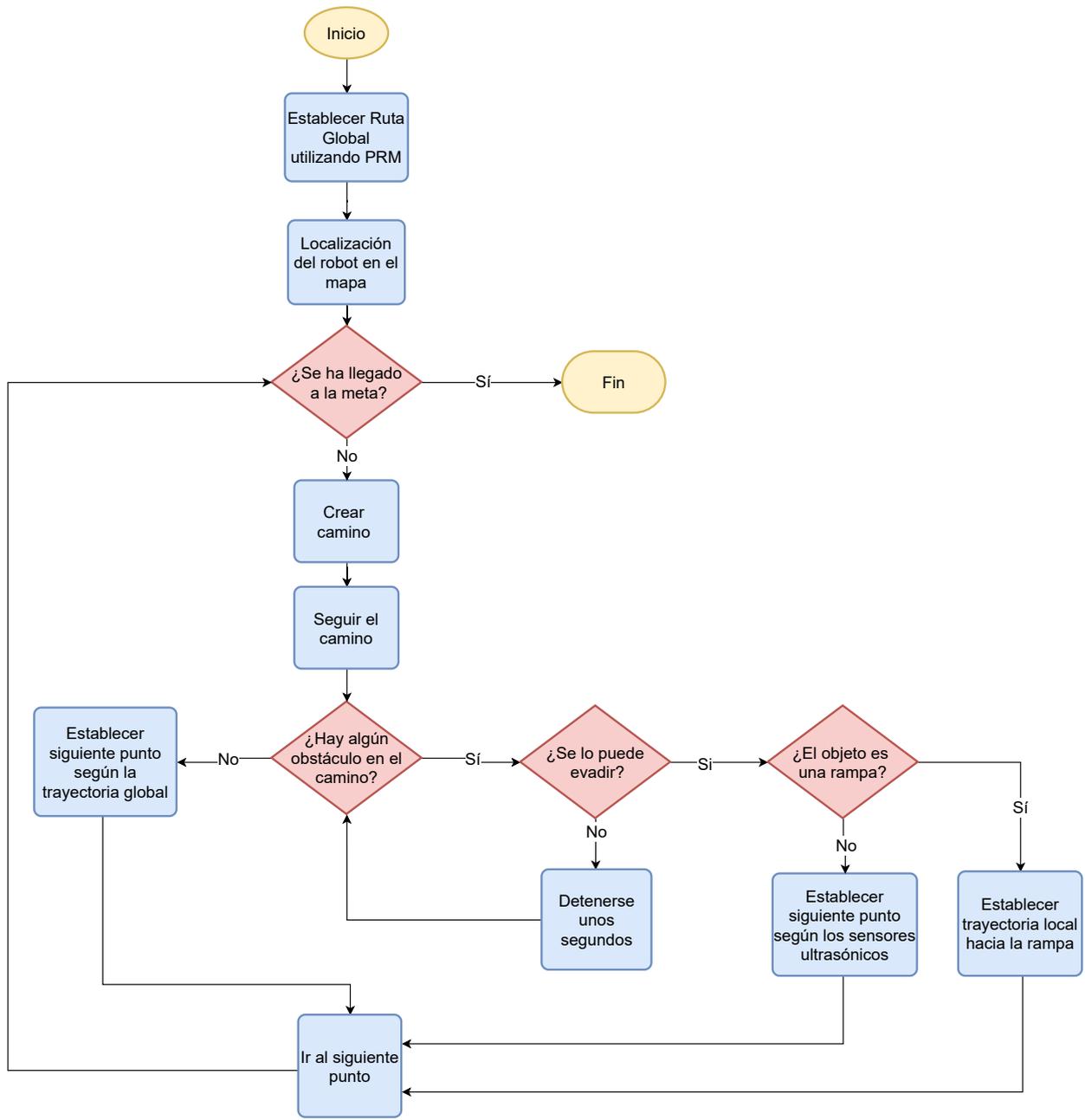


Figura 2.19 Diagrama de flujo del comportamiento del robot durante su trayectoria

## 2.5.4 Visión por computadora

En el campo de visión artificial, existen varios algoritmos de detección, sin embargo, la aplicación en tiempo real es de mucha relevancia, por lo que se implementó YOLOv5 para la detección de rampas y escaleras.

Con el desarrollo presentado en el Apéndice G, ahora se puede empezar el entrenamiento en un entorno PyTorch realizado por Ultralytics en su repositorio [56]. El proceso es iterativo y una vez que se obtenga un buen rendimiento en el entrenamiento, es posible usar los pesos para una posterior inferencia. Se puede verificar la detección de objetos con vídeos de la misma categoría de detección y obtener resultados como se muestra en la Figura 2.20.



**Figura 2.20** Detección de rampa y escalera en el *testing*

Además de detectar el objeto, con la cámara también es posible obtener la profundidad, de manera que el robot podrá replanificar su trayectoria hacia la rampa. El campo de visión de profundidad (*DFOV*) de la cámara a cualquier distancia ( $Z$ ) puede calcularse mediante la ecuación:

$$DFOV = \frac{FOHV}{2} + \tan^{-1} \left[ \tan \left( \frac{FOHV}{2} - \frac{B}{Z} \right) \right] \quad (2.11)$$

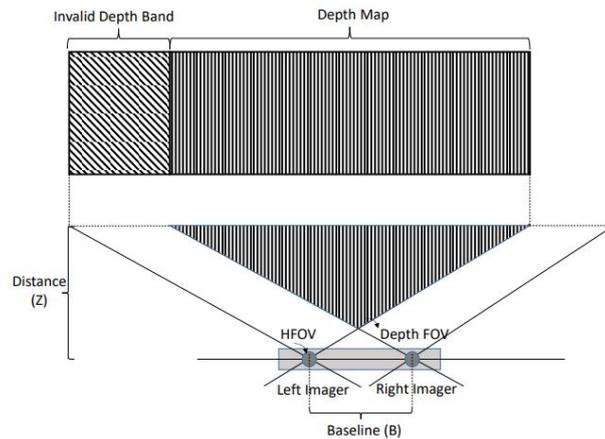
donde:

$DFOV$  = Campo de visión de profundidad

$FOHV$  = Campo de visión horizontal del generador de imágenes izquierdo en el módulo de profundidad

$B$  = Línea de base

$Z$  = Distancia de la escena al módulo de profundidad



**Figura 2.21 Ilustración del campo de visión de profundidad [57]**

No obstante, con los datos de la posición, la cámara puede obtener la distancia  $Z$  del objeto. Ya que cuenta con documentación y librerías desarrolladas en [58] que procesan la información.

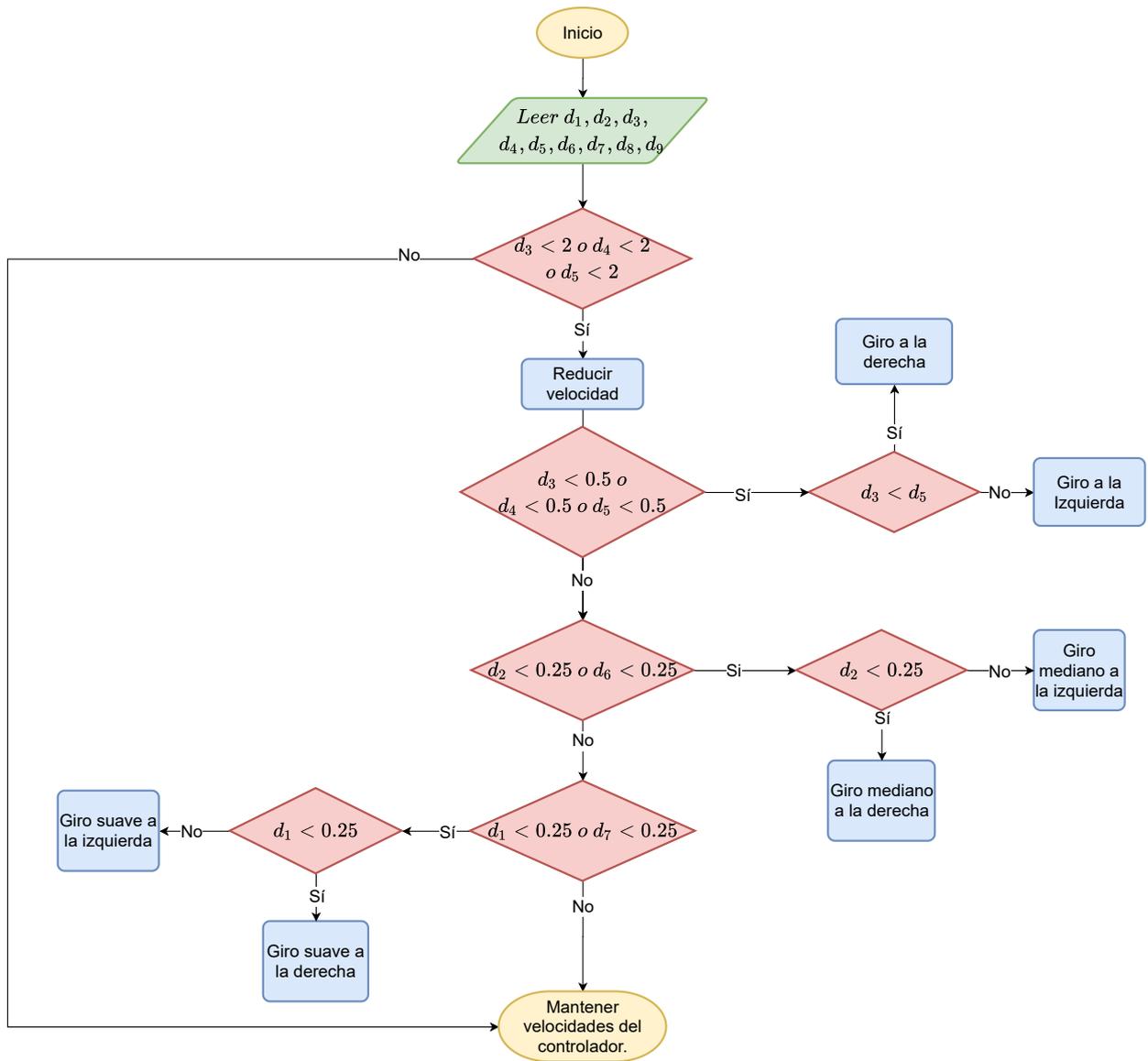
### 2.5.5 Algoritmo de evasión de obstáculos

Para la evasión de obstáculos se utilizaron las señales de los sensores de ultrasonido como entrada y las velocidades angulares de las ruedas como salida. En el Apéndice H se presenta la descripción del modelado de las señales de ultrasonido, para el diseño del algoritmo y la simulación.

El algoritmo para evadir obstáculos se modificó de [59] y el esquema de acciones se presenta en la Figura 2.22 y en la Tabla 2.14 su significado. Cada  $d_i$  representa la distancia medida por cada sensor de ultrasonido, donde  $d_1, d_2$  son las distancias de la izquierda,  $d_3, d_4, d_5$  en el frente,  $d_6, d_7$  en la derecha, y  $d_8, d_9$  en la parte trasera.

**Tabla 2.14 Acciones de Control para evadir obstáculos**

<b>Acción</b>	$w_r$ [rad/s]	$w_l$ [rad/s]
Giro a la derecha	-10	10
Giro a la izquierda	10	-10
Giro mediano a la derecha	-5	10
Giro mediano a la izquierda	10	-5
Giro suave a la derecha	5	10
Giro suave a la izquierda	10	5



**Figura 2.22 Acciones de control según las señales de los sensores de ultrasonido**

## 2.5.6 Control del seguimiento de ruta

Para el control del seguimiento de ruta, se utilizó el algoritmo de *Pure-Pursuit Controller*, un método que consiste en calcular la velocidad lineal y angular para llegar a un punto objetivo a una distancia *lookahead* [60]. Para la implementación se utilizó el bloque *Pure Pursuit Controller* de la librería *Robotic System Toolbox* [53] y se lo configuró con los parámetros presentados en la Tabla 2.15.

**Tabla 2.15 Datos para el bloque Pure Pursuit Controller**

Parámetro	Valor
Velocidad lineal [m/s]	1.38
Velocidad angular Máxima [rad/s]	4.5
Distancia <i>lookahead</i> [m]	0.1

Este controlador por sí solo no es suficiente, ya que al llegar a la meta en lugar de detenerse, continua su movimiento alrededor del punto en forma de oscilaciones. Para solucionar esto, junto con una reducción de la velocidad lineal en las curvas, se implementó el algoritmo de la Figura 2.23, donde se modifican las salidas del controlador para las diferentes situaciones.

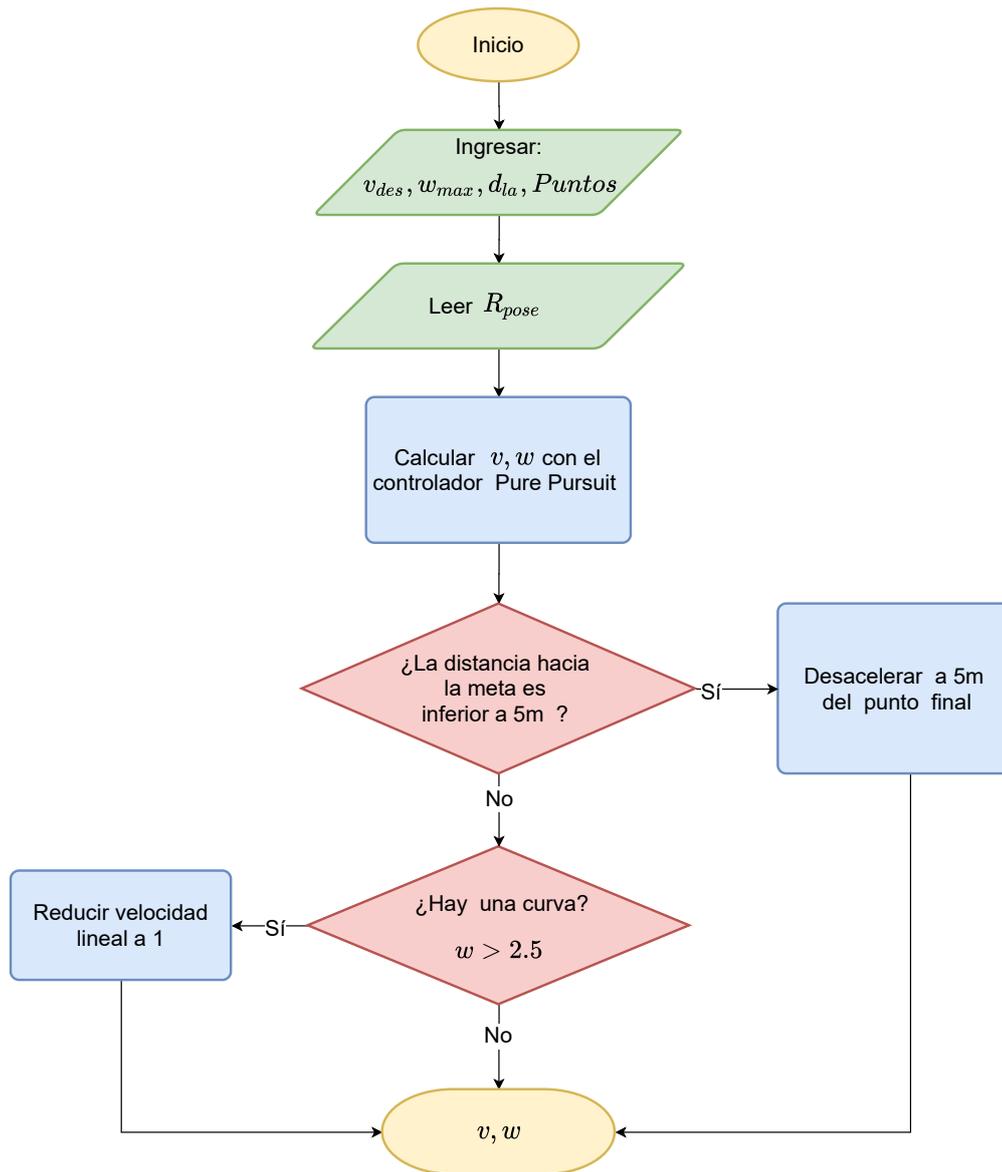


Figura 2.23 Diagrama de flujo del control de seguimiento de ruta

### 2.5.7 Controlador PI de las velocidades de las ruedas

En base a [61], la función de transferencia de un motor de corriente continua sin escobillas (BLDC) se presenta como en la ecuación 2.12, donde  $t_m$  es la constante de tiempo mecánica (ecuación 2.15),  $t_e$  es la constante de tiempo eléctrica (Ecuación 2.16) y  $K_b$  la constante de voltaje inducido.

$$\frac{W(s)}{V(s)} = \frac{1}{K_b} \frac{1}{t_m t_e s^2 + t_m s + 1} \quad (2.12)$$

$$t_m = \sum \frac{J_{eq} R}{K_b K_i} = \frac{J_{eq} \sum R}{K_b K_i} = \frac{3 J_{eq} R}{K_b K_i} \quad (2.13)$$

$$t_e = \sum \frac{L}{R} = \frac{L}{\sum R} = \frac{L}{3R} \quad (2.14)$$

En la Tabla 2.16 se presentan los valores de las diferentes constantes presentadas en las ecuaciones 2.12, 2.13 y 2.14, obtenidas de la hoja de datos del motor en [62] y [63].

**Tabla 2.16 Parámetros del motor BLDC**

Parámetro	Símbolo	Valor
Constante Back - EMF	$K_b$	0.04679 V rad/s
Constante de torque	$K_i$	0.09 Nm/A
Resistencia	R	0.39 $\Omega$
Inductancia	L	0.001H $\Omega$
Inercia total	$J_{eq}$	23 $\mu\text{kgm}^2$

Reemplazando los valores, se obtiene la constante de tiempo mecánica en la ecuación 2.15, la constante de tiempo eléctrica en 2.16 y la función de transferencia del motor en 2.17.

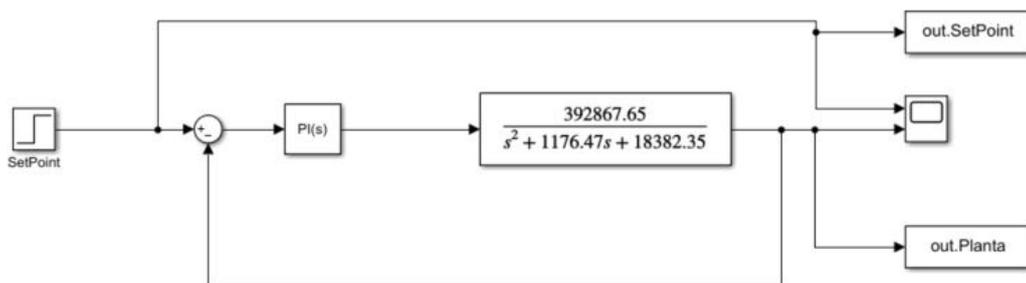
$$t_m = \frac{3 J_{eq} R}{K_b K_i} = \frac{3 \cdot 23 \times 10^{-6} \cdot 0.39}{0.04679 \cdot 0.09} = 0.064 \quad (2.15)$$

$$t_e = \frac{L}{3R} = \frac{0.001}{3 \cdot 0.39} = 0.00085 \quad (2.16)$$

$$G(s) = \frac{W(s)}{V(s)} = \frac{1}{0.064 \cdot 0.00085s^2 + 0.064s + 1} \quad (2.17)$$

$$G(s) = \frac{21.372}{54.4\mu s^2 + 0.064s + 1} = \frac{392867.65}{s^2 + 1176.47s + 18382.35}$$

Para la obtención de los parámetros del controlador se utilizó el método heurístico, con la planta mostrada en la Figura 2.24.

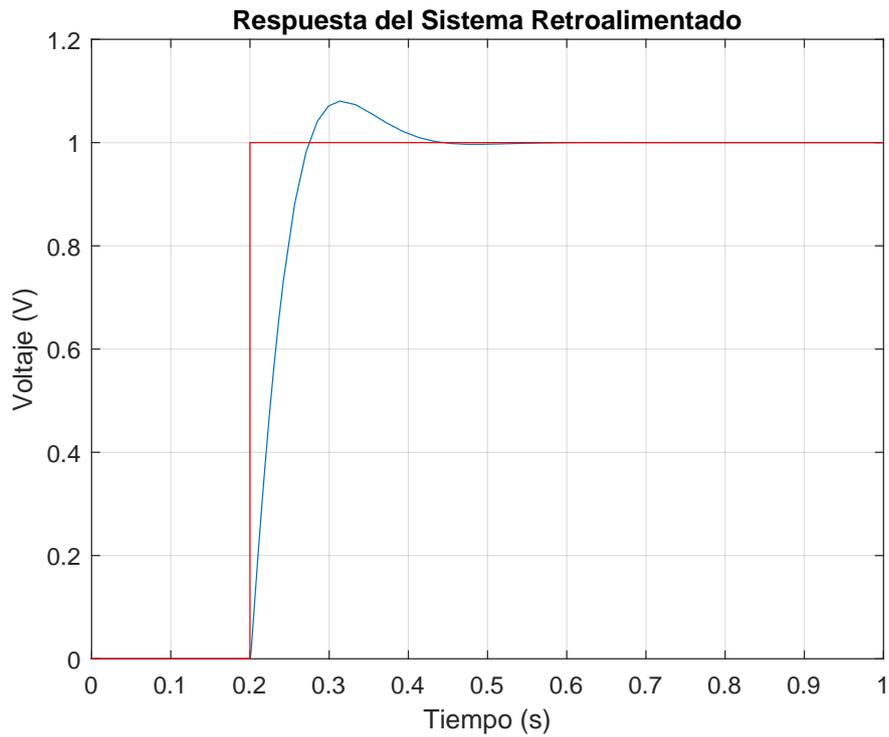


**Figura 2.24** Esquema de la planta y el controlador PID

En primer lugar, con la función de *auto tuning* de MATLAB, se obtuvo un controlador PI con ganancias  $K_p = 0.0646$  y  $K_i = 2.07$ . Con estos valores se obtuvo la respuesta al escalón unitario de la Figura 2.25, cuyas características transientes se muestran en la Tabla 2.17.

**Tabla 2.17** Características transientes iniciales

Característica	Valor
<i>Overshoot</i>	8.08 %
Tiempo de respuesta	0.0535 s
Tiempo de asentamiento	0.1948 s

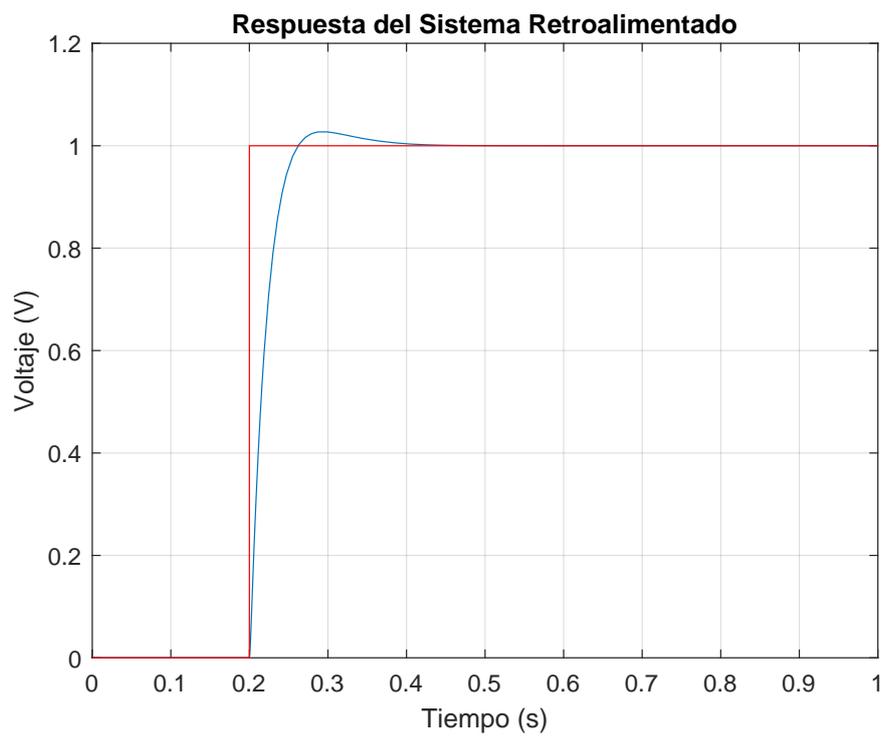


**Figura 2.25 Respuesta con auto - tuning**

A partir de los valores anteriores, se utilizó la herramienta *PID tuner* para lograr un mejor ajuste. Las ganancias resultantes del controlador fueron  $K_p = 0.1353$  y  $K_i = 2.8882$ . Con estos valores, se obtuvo la respuesta al escalón unitario de la Figura 2.26 con las características transientes de la Tabla 2.18. Se observó una gran reducción en el % de *overshoot*, así como en los tiempos de respuesta y asentamiento, por lo que se decidió optar por este controlador para el diseño.

**Tabla 2.18 Características transientes finales**

Característica	Valor
<i>Overshoot</i>	2.78 %
Tiempo de respuesta	0.0376s
Tiempo de asentamiento	0.126 s



**Figura 2.26 Respuesta con el controlador final**

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

### 3.1 Diseño de la estructura mecánica

El diseño final del robot móvil *delivery* se puede visualizar en la Figura 3.1 y adjunto en la Tabla 3.1 se muestran las piezas principales con las características de mayor relevancia.

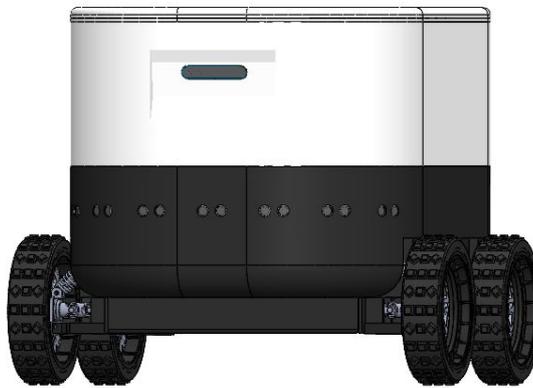
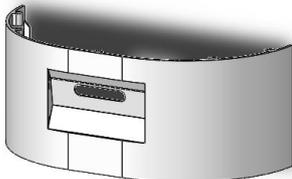
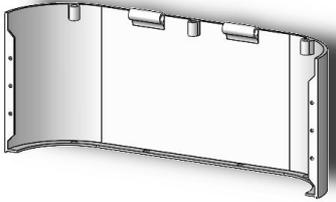
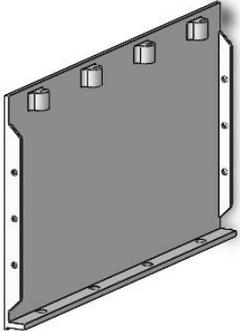
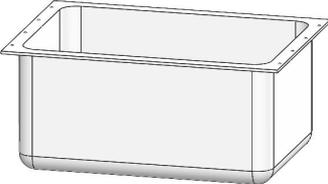
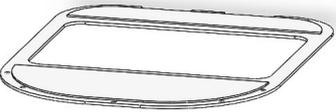
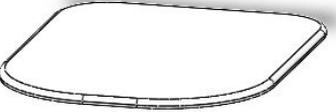


Figura 3.1 Ensamble de robot.

Tabla 3.1 Componentes principales de la estructura del robot.

Componente	Imagen	Características
Chasis		<b>Dimensiones:</b> 650x500x222mm <b>Cantidad:</b> 1 u <b>Peso:</b> 4.2 kg
Pieza frontal		<b>Dimensiones:</b> 200x500x202mm <b>Cantidad:</b> 1 u <b>Peso:</b> 0.76 kg

Pieza posterior		<p><b>Dimensiones:</b> 100x500x202mm</p> <p><b>Cantidad:</b> 1 u</p> <p><b>Peso:</b> 0.65 kg</p>
Pieza lateral: izquierda y derecha		<p><b>Dimensiones:</b> 350x5x202mm</p> <p><b>Cantidad:</b> 2 u</p> <p><b>Peso:</b> 0.39 kg</p>
Contenedor		<p><b>Dimensiones:</b> 350x490x202mm</p> <p><b>Cantidad:</b> 1 u</p> <p><b>Peso:</b> 2.51 kg</p>
Base de la tapa		<p><b>Dimensiones:</b> 650x500x10mm</p> <p><b>Cantidad:</b> 1 u</p> <p><b>Peso:</b> 1.02 kg</p>
Tapa		<p><b>Dimensiones:</b> 650x500x15mm</p> <p><b>Cantidad:</b> 1 u</p> <p><b>Peso:</b> 1.02 kg</p>
Rueda		<p><b>Dimensiones:</b> R100x64mm</p> <p><b>Cantidad:</b> 4 u</p> <p><b>Peso:</b> 0.65 kg</p>

### 3.2 Dispositivos electrónicos seleccionados

En la Figura 3.2 se observa el esquema general de conexiones de los diferentes elementos de hardware, que componen la electrónica del robot. Se resaltan los protocolos de comunicación que utilizan los dispositivos de control y los sensores de localización, además del tipo de señal que se envía a los actuadores y que se recibe de los sensores. En el Apéndice I se presenta el plano detallado de las conexiones por cada pin. Por otro lado, en la Tabla 3.2 se presenta la lista del peso de cada componente, teniendo una carga total de 20.271 kg.

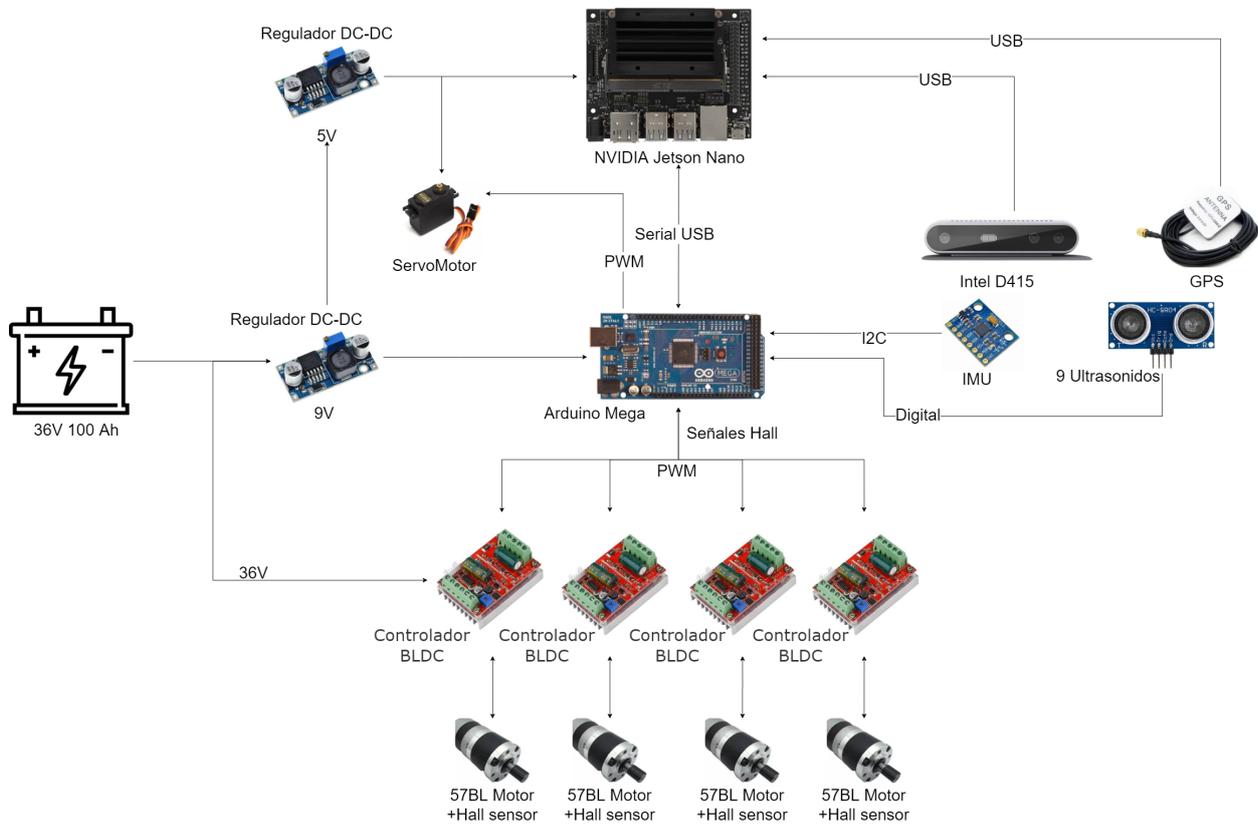


Figura 3.2 Esquema de sensores, actuadores y controladores

**Tabla 3.2 Pesos de los dispositivos electrónicos seleccionados**

<b>Dispositivo</b>	<b>Cantidad</b>	<b>Masa por unidad [g]</b>	<b>Masa total [g]</b>
NVIDIA Jetson Nano	1	340	340
Arduino Mega	1	37	37
Módulo 4G + WiFi + GPS	1	204	204
Intel D415 Depth Camera	1	241	241
Sensor HC-SRO4	9	10	90
IMU GY-521	1	3	3
Controlador BLDC	4	73	292
Motor BLDC 57BL04A	4	1,250	5,000
Servomotor TowerPro MG995	1	55	55
Convertidor DC-DC Buck LM2596	2	11	9
Batería de Li-ion 36V, 100Ah	1	14,000	14,000
<b>Total</b>			<b>20,271</b>

### 3.3 Detección y subida de rampas

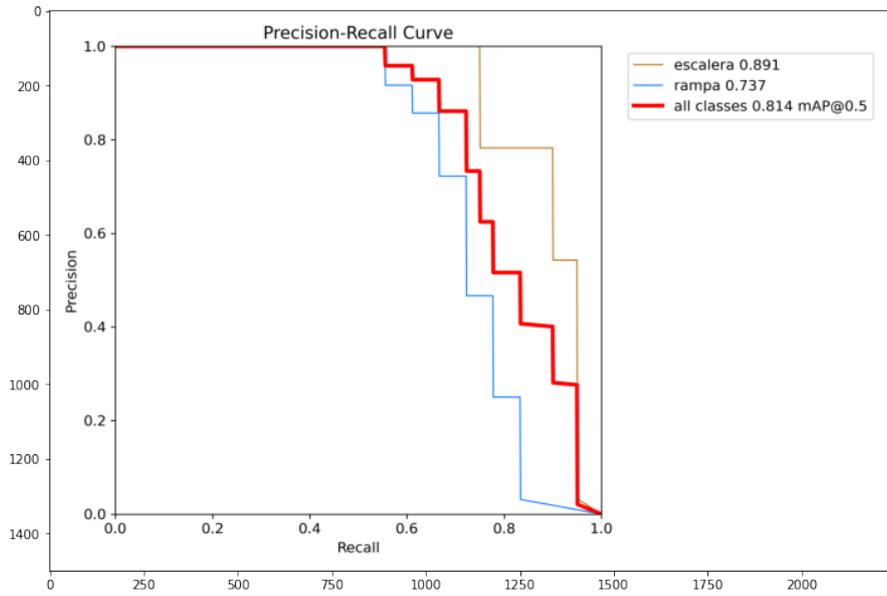
La detección de rampas y escaleras se realizó con la cámara, debido a que se estaba entrenando con YOLOv5, inicialmente los pesos que se usaron en el entrenamiento fueron unos predeterminados. Una vez que se terminó la primera prueba, se obtuvieron los nuevos pesos en un archivo para el siguiente entrenamiento.

En la Figura 3.3 se observa la clase y el nivel de confianza de cada detección. Cabe recalcar que el algoritmo fue entrenado con imágenes de la ESPOL, por lo que el primer entrenamiento fue una buena base para mejorarlo.

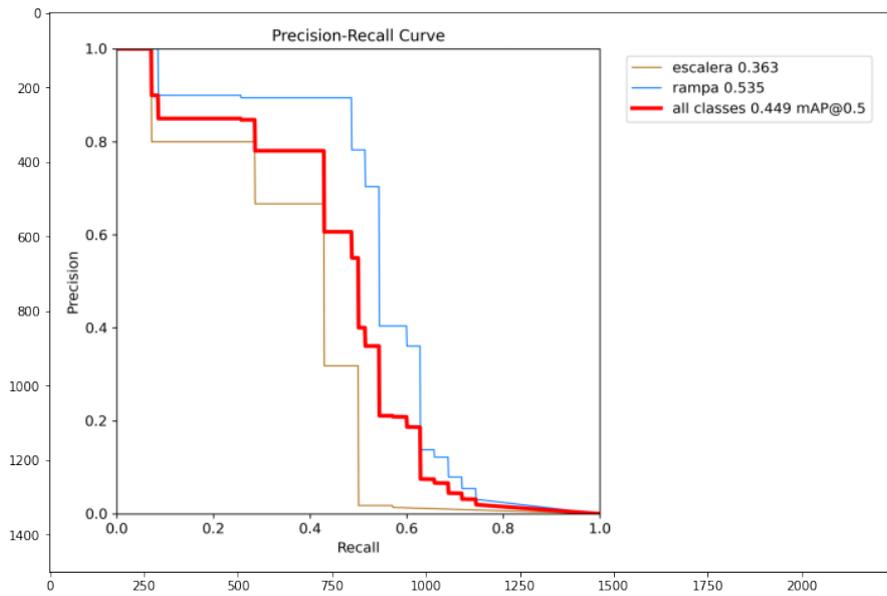


**Figura 3.3 Detección de objetos con número de clase y nivel de confianza**

Con los entrenamientos realizados se obtuvo la validación mostrada en la Figura 3.4, que muestra una precisión media, en validación, de 0.8 para todas las clases. Pero en la prueba se obtuvo una precisión baja, de 0.5 aproximadamente, como se puede visualizar en la Figura 3.5.



**Figura 3.4 Validación del algoritmo de detección de rampas y escaleras**

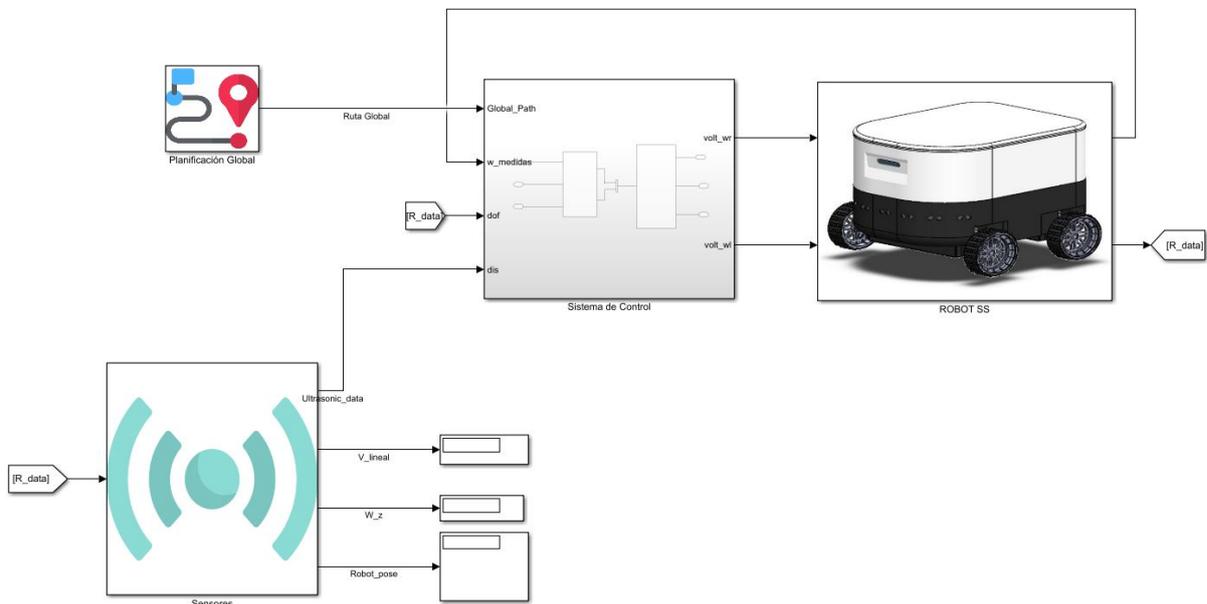


**Figura 3.5 Prueba del algoritmo de detección de rampas y escaleras**

## 3.4 Simulación

### 3.4.1 Modelo en MATLAB - Simulink

En la Figura 3.6 se presenta la vista principal del modelo desarrollado. En el bloque de planificación global se cargan los puntos de la ruta global, obtenidos desde el archivo ejecutable Data.m del directorio del proyecto. El bloque Sistemas de Control contiene todos los algoritmos planteados para la planificación local, siendo estos el control del seguimiento de ruta, evasión de obstáculos, planificación hacia las rampas, y el control PI de los motores. En el bloque ROBOT SS se encuentra el modelo CAD exportado desde Solidworks, además del ambiente virtual. El bloque sensores contiene el modelo de los sensores ultrasónicos, además de los sensores virtuales del bloque 6DOF para monitorear la pose, velocidad lineal y angular.



**Figura 3.6 Bloques principales del modelo desarrollado en MATLAB-Simulink**

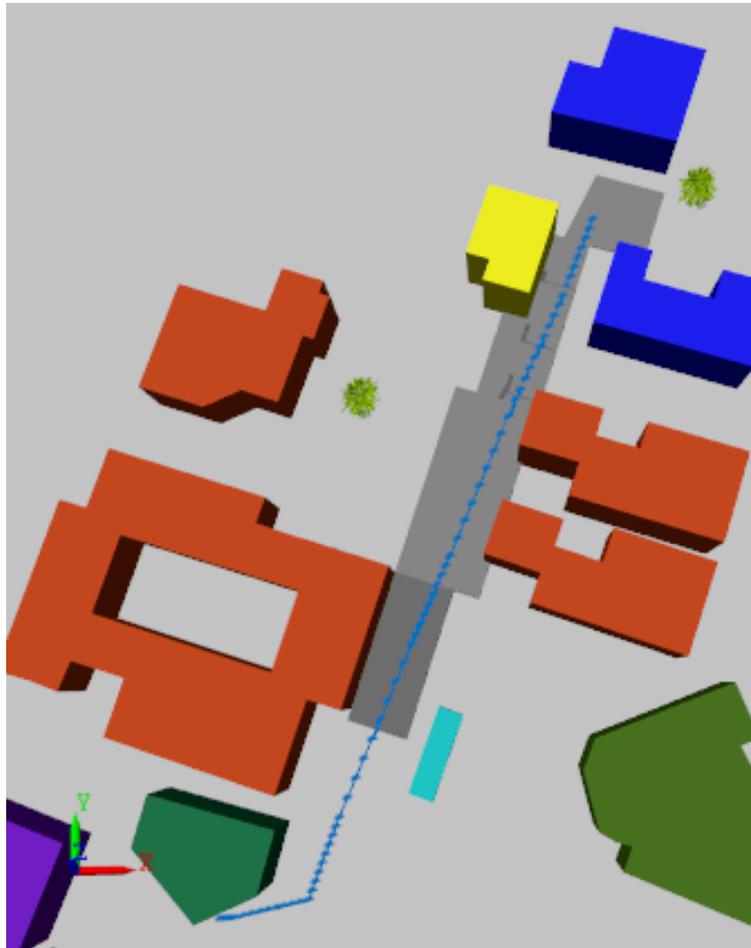
En el Apéndice H se presenta más en detalle ciertos bloques y metodologías utilizadas para la obtención del modelo 3D. Además, los archivos del proyecto pueden obtenerse de un repositorio en GitHub. <sup>1</sup> creado para

<sup>1</sup>[https://github.com/alexanrr/Modelo\\_Robot\\_Delivery.git](https://github.com/alexanrr/Modelo_Robot_Delivery.git)

este proyecto.

### 3.4.2 Descripción del escenario

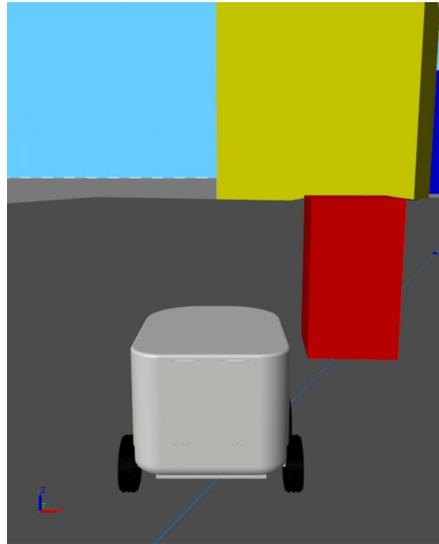
Para la simulación se definió la ruta obtenida de la Figura 2.16, la cual representa el escenario de una posible entrega, en el campus Gustavo Galindo de la ESPOL, desde un punto cercano al restaurante Sweet & Coffee, hasta la explanada de la FIMCP. En la Figura 3.7 se observa la ruta global resaltada en el ambiente virtual.



**Figura 3.7 Vista superior del ambiente 3D en MATLAB**

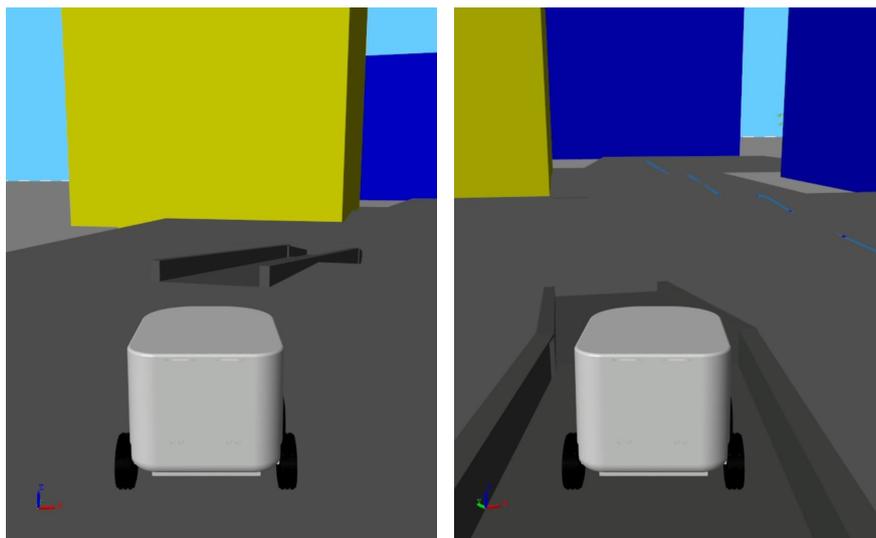
Durante el recorrido aparece un obstáculo en una posición aleatoria. El robot lo detecta a cierta distancia, reduce su velocidad y lo evade según el espacio disponible. En este caso al tener ambos lados disponibles, se definió que por defecto la evasión por la izquierda, como se observa en la

Figura 3.8.



**Figura 3.8 Robot evadiendo obstáculo**

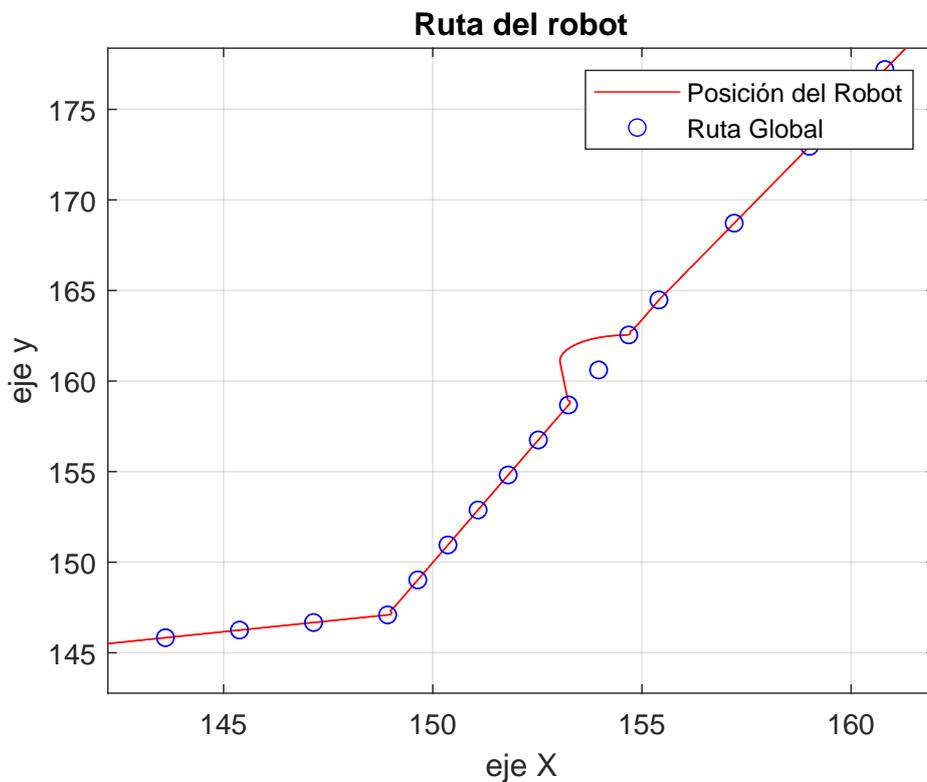
Para llegar al punto objetivo, el robot debe pasar por algunas rampas. El algoritmo de visión por computadora, detecta la rampa a una distancia y orientación aproximada. En el modelo se simularon estos datos, introduciendo una trayectoria alterna a la que el robot deba desviarse, cuando la rampa se encuentre en el rango focal de la cámara. En la Figura 3.9 se puede apreciar la acción descrita.



**Figura 3.9 Robot detectando (izquierda) y bajando (derecha) una rampa**

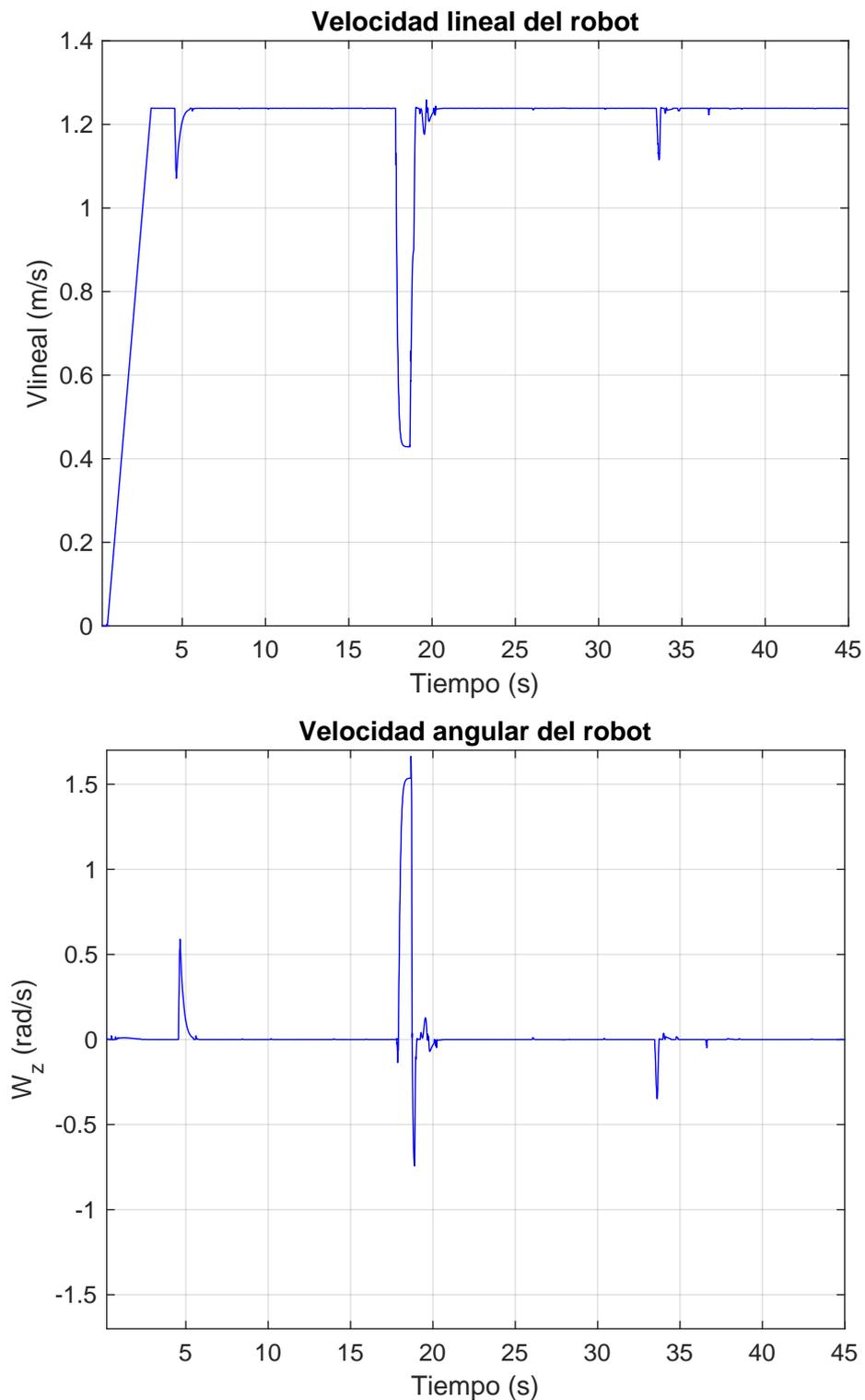
### 3.4.3 Análisis de la ruta obtenida

En la Figura 3.10 se presenta una porción de la ruta, donde ocurren dos escenarios: El primero, es el cambio brusco de dirección en el punto (149, 147). En las pruebas iniciales el robot se salía demasiado de la ruta y la acción de girar provocaba inestabilidad al momento de regresar a la ruta global. Para esto se programó una reducción de la velocidad lineal cuando la velocidad propuesta por el controlador sea superior a 2.5 rad/s.



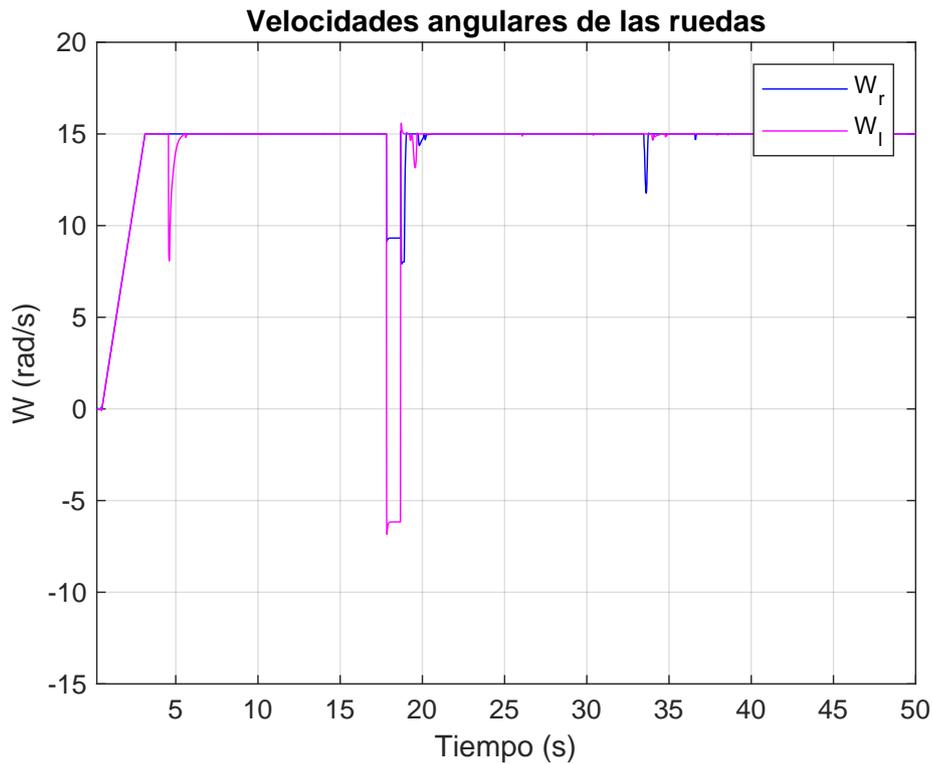
**Figura 3.10 Seguimiento de puntos y evasión de un obstáculo**

En la Figura 3.11 se observa el cambio de la velocidad lineal durante un aumento de la velocidad angular en el camino. El segundo escenario es la presencia de un obstáculo, el robot reduce su velocidad cuando uno de los sensores frontales detecta distancias inferiores a 1.5m, para luego girar dependiendo del espacio disponible. En este caso, la acción de control fue la de moverse a la izquierda, actuando las ruedas de la izquierda en sentido contrario a las de la derecha, tal como se ve en la Figura 3.12.



**Figura 3.11 Velocidad lineal y angular del robot**

Posterior a esto, el robot trata de regresar a la ruta global, sin embargo, al bordear el obstáculo, los sensores de los lados mandan un comando de control al detectar 20 cm. La acción de control fue de girar ligeramente a la izquierda, hasta no detectar esta distancia (20 cm). En la Figura 3.13



**Figura 3.12 Acción de evasión a la izquierda**

se presenta otra porción de la ruta. Aquí se observa el instante que el robot deja la ruta global y sigue la ruta local hacia las rampas. En esta ruta también se utiliza un *pure pursuit controller* pero a una velocidad inferior de 3.5 km/h, por seguridad al momento de bajar las rampas. La duración de la simulación fue de 180 segundos. Sin embargo, es un escenario ideal, donde solo hay un obstáculo y el resto del camino esta libre. Tomando en cuenta estos aspectos, se puede aproximar que el robot se demoraría de 3 a 5 minutos en hacer el recorrido de ida y un total de 8 a 12 minutos en realizar el proceso completo (ida, entrega y retorno). Con estos valores y considerando que la batería tiene una autonomía de 3.21 horas en su máximo funcionamiento; se estima un máximo de 19 entregas a lo largo de esta ruta, antes de tener que recargar la batería.

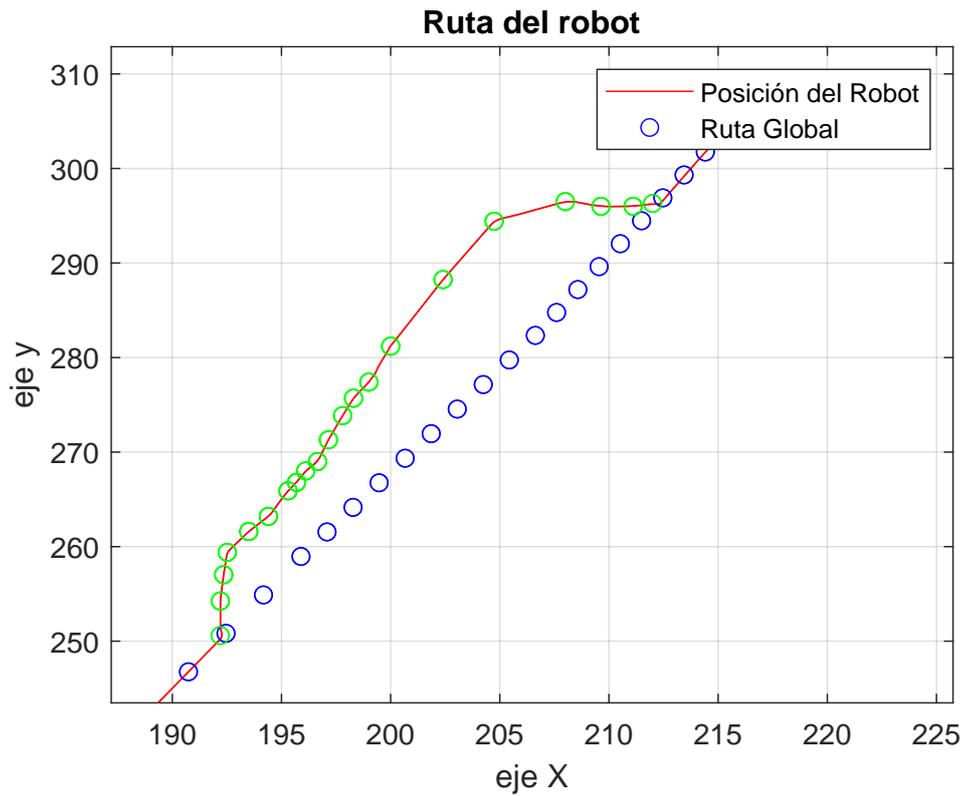
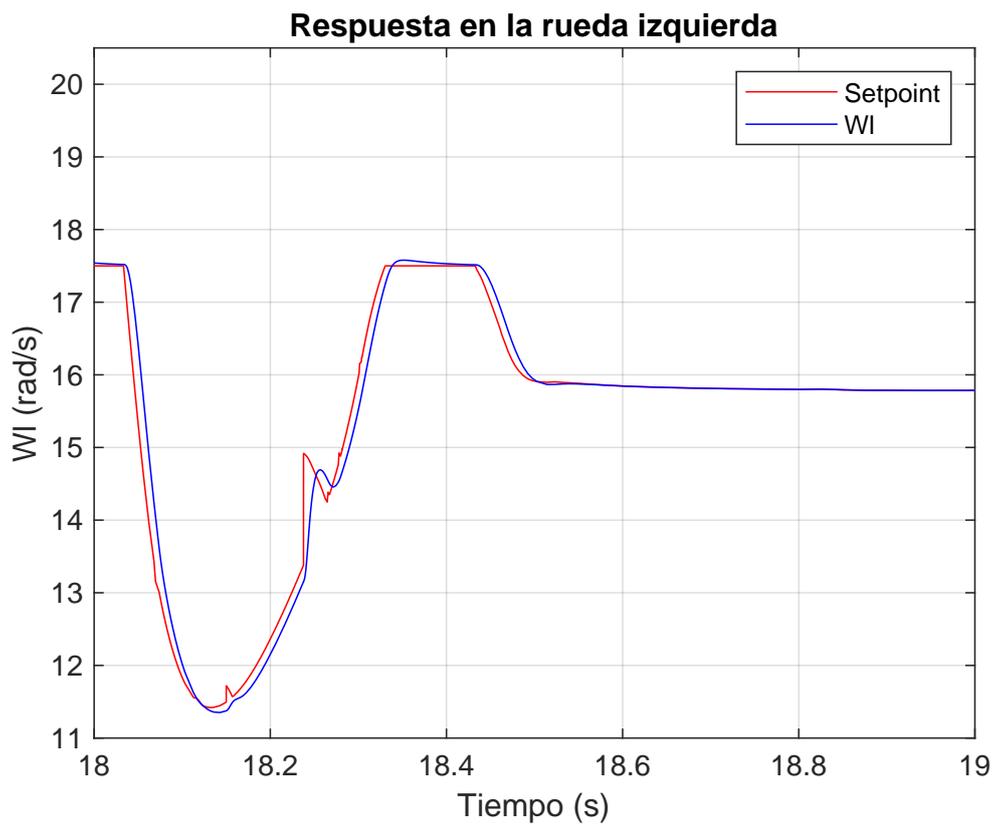


Figura 3.13 Ruta de desvío a las rampas

### 3.4.4 Controlador PI de los motores

En la Figura 3.14 se presentan las velocidades *setpoint* y las obtenidas como salida de la función de transferencia del motor. En ambos casos, el controlador compensa la señal de error al lograr una respuesta muy cercana a la entrada. Sin embargo, en puntos donde el cambio de velocidad es muy alto se observa el *overshoot*, pero este no representa un problema, al ser muy bajo.



**Figura 3.14** Comparación de las velocidades angulares *setpoint* y las reales

### 3.5 Análisis de costos

En la Tabla 3.3 se presenta el detalle de los costos a considerar en la implementación del proyecto.

**Tabla 3.3 Presupuesto para la construcción del robot móvil**

Descripción	Cantidad	Costo unitario	Costo total
<b>Componentes electrónicos</b>			
NVIDIA Jetson Nano	1	\$ 325	\$ 325
Arduino Mega	1	\$ 25	\$ 25
Intel D415 Depth Camera	1	\$ 300	\$ 300
Motores BLDC 57BL	4	\$70	\$ 200
Controlador BLDC	4	\$ 10	\$ 40
Sensor ultrasonido HC-SR04	9	\$ 2	\$ 18
IMU GY-521	1	\$5	\$5
Módulo 4G + GPS + Wi-Fi	1	\$ 100	\$ 100
Servomotor MG995	1	\$ 20	\$ 20
Regulador DC-DC LM2596	2	\$6	\$ 12
Batería de Li-Ion 36V 100Ah	1	\$ 600	\$ 600
Cargador de batería 5A	1	\$ 200	\$ 200
Varios	-	-	\$ 30
<b>Partes mecánicas</b>			
Ruedas el par	2	\$ 229	\$ 458
Amortiguadores	4	\$ 18	\$ 72
Plástico ABS	1	\$ 50	\$ 50
Barra de acero AISI 416	1	\$ 20	\$ 20
Barra de acero AISI 303	1	\$ 17	\$ 17
Alfombrilla aislante de Aerogel	4	\$ 20	\$ 80
Polímero recubierto de aluminio	8	\$ 2	\$ 16
Nylon antibacteriano	8	\$ 35	\$ 280
Varios	-	-	\$30
<b>Puesta en marcha</b>			
Mano de obra	2	\$ 640	\$ 1280
Construcción	-	-	\$1000
<b>Total</b>			\$ 5000

Los componentes del sistema electrónico son los que abarcan la mayor parte de costo del proyecto. Esto se debe a que en ciertos componentes se incluyeron costos de envío, al no estar disponibles en tiendas nacionales. Más aún, se incluyó un margen de costos varios de \$30, considerando el costo de cables, pines de alimentación, adaptadores, entre otros.

Asimismo, para los componentes mecánicos se incluyeron los precios de las ruedas y amortiguadores seleccionados, además del costo por kilogramo de los materiales considerados en el diseño de la estructura y

el eje cardan. En costos varios, se incluyo una estimación del precio total de los diferentes elementos de sujeción necesarios para la estructura. Además, se incluyeron los costos relacionados a la puesta en marcha del robot. Para la mano de obra, se considero un costo por hora de \$8, en base a un sueldo referencial de un Ingeniero, entre \$1,000 y \$1,500. Para la implementación, se consideró un tiempo máximo de 2 semanas, con dos personas trabajando. Por otra parte, se considero un aproximado de los costos relacionados a la construcción de los elementos diseñados, siendo este un valor que puede cambiar dependiendo del lugar donde se lo haga.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Se diseñó la estructura y piezas principales del robot, siendo el ABS el material seleccionado para la carrocería. Además, se realizó un análisis estático en SolidWorks, donde se obtuvo un factor de seguridad de 21 para el chasis, lo que indica que la pieza base podrá soportar la carga definida de 40 kg y un poco más. También se realizó un estudio por fatiga, donde se verificó la resistencia del material y se obtuvo un porcentaje de daño de 10 % con un fallo en la pieza a partir del ciclo número 955,000.00.
- El sistema electrónico fue dividido en placas de control, sensores, sistema de locomoción y batería. Primero se hizo un proceso de selección para los dispositivos de control, teniendo una NVIDIA Jetson nano como computadora principal y una tarjeta Arduino Mega como placa secundaria. Las características de estos dispositivos permitieron la selección de tecnologías para la localización y percepción del entorno. La parte de locomoción compone a los motores BLDC, los cuales fueron dimensionados en base al torque requerido para mover una carga de 40 kg, en una cuesta con una inclinación de 30°. Para suplir el consumo de potencia de los componentes, se seleccionó una batería de iones de Litio con una capacidad de 100 Ah a 36V, la cual le da al robot una máxima autonomía de 4 horas, en total funcionamiento.

- El sistema de control se diseñó considerando una arquitectura compuesta por varias etapas. Primero se utilizó el algoritmo PRM, para obtener la ruta global de navegación en un mapa definido. Luego, se implementaron diferentes algoritmos para seguir la ruta de manera estable, y que a la vez se puedan evadir obstáculos en el camino, sin perder el rumbo hacia la meta. En uno de estos fue un algoritmo de visión por computadora para la detección de rampas y escaleras, entrenado con imágenes reales del campus. La integración de este sistema permitió al robot trabajar de manera autónoma, siendo esta su principal característica.
- Se desarrolló un modelo en MATLAB - Simulink, en el que se pudo demostrar el funcionamiento del robot con los algoritmos de control propuestos. El tiempo de simulación sirvió para estimar el número de entregas, con la autonomía de las baterías, teniendo un máximo de 19 por carga. Sin embargo, este valor considera que las entregas son en la misma ruta y en un plano inclinado a 30°; por lo que puede variar al tener entregas en rutas más cortas o en terreno plano.

## 4.2 Recomendaciones

- El diseño de las piezas esta sujeto a modificaciones, ya que el material es el ABS, sin embargo la segunda propuesta, el ASA, también es apta para soportar las condiciones climáticas en Guayaquil, además de tener mayor resistencia a los golpes que el material seleccionado. Debido a esta última característica se podría reducir el ancho de las paredes y obtener mayor capacidad de carga para el robot.
- Se deben tomar ciertos cuidados para aumentar la vida útil de la batería. Esta no debe descargarse totalmente, por lo que se recomienda operar a lo mucho durante 3.21 horas. Es importante mantener un constante control de los ciclos de carga, ya que al estar cerca de los 1000, la batería debe ser cambiada.
- Aunque el algoritmo de detección de rampas y escaleras fue entrenado con imágenes de la ESPOL, las estadísticas muestran una precisión media (*mean Average Precision*) de aproximadamente 0.8, por lo que se recomienda ampliar la base de datos con 500 imágenes por clase, es decir, 1000 imágenes en total para que la precisión sea mayor y se logre reducir la detección de falsos positivos.
- Para la implementación de este proyecto, se recomienda el desarrollo de una aplicación móvil, tanto para hacer los pedidos, como para monitorear los puntos de entrega y la ubica
- Una posible mejora en el proyecto, es el aumento del número de ruedas a seis, junto con el diseño de un sistema mecánico que permita al robot subir al menos un escalón. Esto aumentaría el alcance de los puntos de entrega, junto con una mejor tracción.

# BIBLIOGRAFÍA

- [1] E. Universo, “Las apps de entrega de comida a domicilio que podemos usar en Ecuador,” Abr. 2, 2021. [En línea]. Disponible:<https://www.eluniverso.com/noticias/ecuador/las-apps-de-entrega-de-comida-a-domicilio-que-podemos-usar-en-ecuador-nota/> [Accedido: Sept. 01, 2022].
- [2] J. P. D. A. Ponce, “Aplicaciones de delivery en ecuador en 2022,” Abril 1, 2021. [En línea]. Disponible: <https://www.ekosnegocios.com/articulo/aplicaciones-de-delivery-en-ecuador-en-2022> [Accedido: Sep. 01, 2022].
- [3] D. S. López and M. M. Villacis, “Factores que inciden en el uso de apps de delivery para la compra de comida rápida en la ciudad de guayaquil,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 6, no. 2, pp. 1180–1197, 2022.
- [4] J. Schmidt Urdanivia, H. A. Young Gonzales, and S. Podestá Cuadros, “El servicio de delivery como estrategia competitiva,” pp. 81–86, 2015.
- [5] The Exponent, “Food-delivery robots struggle to navigate campus,” Sep. 13 2019. [En línea]. Disponible: [https://www.purdueexponent.org/campus/article\\_8840a70a-d65a-11e9-827f-7b1653a57e1d.html](https://www.purdueexponent.org/campus/article_8840a70a-d65a-11e9-827f-7b1653a57e1d.html) [Accedido: Jun. 5, 2022].
- [6] M. M. Abrar, R. Islam, and M. A. H. Shanto, “An autonomous delivery robot to prevent the spread of coronavirus in product delivery system,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2020, pp. 0461–0466.
- [7] C. Mvula, “Impacts of employee turnover on public services delivery, a case of public service commission,” Ph.D. dissertation, Mzumbe University, 2020.
- [8] D. Sinha and D. Pandit, “A simulation-based study to determine the negative externalities of hyper-local food delivery,” *Transportation Research Part D: Transport and Environment*, vol. 100, p. 103071, 2021.
- [9] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, “Environmental and health impacts of air pollution: a review,” *Frontiers in public health*, p. 14, 2020.

- [10] E. Ortiz-Prado, A. R. Henriquez-Trujillo, I. A. Rivera-Olivero, T. Lozada, M. A. Garcia-Bereguaiain *et al.*, “High prevalence of sars-cov-2 infection among food delivery riders. a case study from quito, ecuador,” *Science of the Total Environment*, vol. 770, p. 145225, 2021.
- [11] A. Telenti, A. Arvin, L. Corey, D. Corti, M. S. Diamond, A. García-Sastre, R. F. Garry, E. C. Holmes, P. S. Pang, and H. W. Virgin, “After the pandemic: perspectives on the future trajectory of covid-19,” *Nature*, vol. 596, no. 7873, pp. 495–504, 2021.
- [12] T. Hoffmann and G. Prause, “On the regulatory framework for last-mile delivery robots,” *Machines*, vol. 6, no. 3, p. 33, 2018.
- [13] K. Byrd, A. Fan, E. Her, Y. Liu, B. Almanza, and S. Leitch, “Robot vs human: expectations, performances and gaps in off-premise restaurant service modes,” *International Journal of Contemporary Hospitality Management*, 2021.
- [14] A. Lee and A. L. Toombs, “Robots on campus: understanding public perception of robots using social media,” in *Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing*, 2020, pp. 305–309.
- [15] V. R. B. Sotelo, J. R. G. Sánchez, and R. S. Ortigoza, “Robots móviles: Evolución y estado del arte,” *Polibits*, no. 35, pp. 12–17, 2007.
- [16] L. Marín, Á. Soriano, V. Mayans, M. Vallés, Á. Valera, and P. Albertos, “Localización asistida por gps para robots moviles en configuracion ackermann de recursos limitados.”
- [17] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, “Experimental kinematics for wheeled skid-steer mobile robots,” in *2007 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2007, pp. 1222–1227.
- [18] H. Taheri and C. X. Zhao, “Omnidirectional mobile robots, mechanisms and navigation approaches,” *Mechanism and Machine Theory*, vol. 153, p. 103958, 2020.
- [19] G. Bermudez *et al.*, “Robots móviles. teoría, aplicaciones y experiencias,” *Tecnura*, vol. 5, no. 10, pp. 6–17, 2002.

- [20] G. Alvarez and O. Flor, “Desempeño en métodos de navegación autónoma para robots móviles,” *Minerva*, vol. 1, no. 2, pp. 19–29, 2020.
- [21] S. G. M. Gutiérrez, A. C. Pazarán, R. G. Mejía, L. N. O. Moreno *et al.*, “Desarrollo de plataforma para implementación de robots colaborativos,” *Visión electrónica*, vol. 12, no. 1, pp. 22–31, 2018.
- [22] M. Długosz, P. Wegrzyn, and M. Roman, “Autonomous delivery robot aquilo,” in *Advanced, Contemporary Control*. Springer, 2020, pp. 1213–1224.
- [23] A.-D. Suárez-Gómez and W.-J. Pérez-Holguín, “Sistema de odometría basado en codiseño h/s para un robot móvil diferencial.”
- [24] M. M. Abrar, R. Islam, and M. A. H. Shanto, “An autonomous delivery robot to prevent the spread of coronavirus in product delivery system,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2020, pp. 0461–0466.
- [25] G.-S. Cai, H.-Y. Lin, and S.-F. Kao, “Mobile robot localization using gps, imu and visual odometry,” in *2019 International Automatic Control Conference (CACCS)*. IEEE, 2019, pp. 1–6.
- [26] R. P. Borase, D. Maghade, S. Sondkar, and S. Pawar, “A review of pid control, tuning methods and applications,” *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, 2021.
- [27] Starship Technologies, “Autonomous robots for industry 4.0,” 2020. [En línea]. Disponible: <https://starshipdeliveries.com/industry/> [Accedido: Jun. 1, 2022].
- [28] B. Nemire, “Autonomous robots to deliver your shopping in 2016,” Nov. 10, 2016. [En línea]. Disponible: <https://developer.nvidia.com/blog/self-driving-delivery-robots-coming-in-2016/> [Accedido: Jul. 7, 2022].
- [29] D. A. Chávarro, A. Londoño Restrepo, and J. Hurtado Morales, “Avances en la industria web 4.0 en colombia y su aplicabilidad,” 2020.
- [30] U. Howard, “Meet Kiwibot - SODEXO,” 2021. [En línea]. Disponible: [https://howard.sodexomyway.com/kiwibot/kiwibot\\_FAQ](https://howard.sodexomyway.com/kiwibot/kiwibot_FAQ) [Accedido: Jul. 10, 2022].

- [31] C. Elliott, "Food Delivery Robots Are Going Back To School This Fall," Aug. 10, 2021. [En Línea]. Disponible: <https://www.forbes.com/sites/christopherelliott/2021/08/10/food-delivery-robots-are-going-back-to-school-this-fall/?sh=2e15efdd567a> [Accedido: Jun. 5, 2022].
- [32] K. Berezina, O. Ciftci, and C. Cobanoglu, "Robots, artificial intelligence, and service automation in restaurants," in *Robots, artificial intelligence, and service automation in travel, tourism and hospitality*. Emerald Publishing Limited, 2019.
- [33] T. Robby, "Deliver an autonomous future," 2019. [En línea]. Disponible: <https://robby.io/> [Accedido: Jul. 1, 2022].
- [34] R. Kraus, "Pepsico's new food delivery robots are every college stoner's dream," Jan. 4, 2019. [En línea]. Disponible: <https://mashable.com/article/pepsico-snackbot-food-delivery-robots#o.poRnmZvSq2> [Accedido: Jun. 15, 2022].
- [35] P. I. Corke and O. Khatib, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.
- [36] J. H. Ezeta, A. Mandow, and A. G. Cerezo, "Los sistemas de suspensión activa y semiactiva: una revisión," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 10, no. 2, pp. 121–132, 2013.
- [37] D. Hrovat, "Survey of advanced suspension developments and related optimal control applications," *Automatica*, vol. 33, no. 10, pp. 1781–1817, 1997.
- [38] A. de Souza Mendes, "Quarter car model: Simulation and animation of a quarter car model." Jan. 30, 2021. [En línea]. Disponible: [https://la.mathworks.com/matlabcentral/fileexchange/105835-quarter-car-model?s\\_tid=FX\\_rc1\\_behav](https://la.mathworks.com/matlabcentral/fileexchange/105835-quarter-car-model?s_tid=FX_rc1_behav) [Accedido: Jul. 30, 2022].
- [39] V. Quintero, "Lithium ion batteries: features and applications," 2021.
- [40] EVPOWER, "Paquete de batería para bicicleta eléctrica, 36V, 48V, 60V, 20ah, 30ah, 100ah, Scooter, motocicleta," 2022. [En línea]. Disponible: <https://es.aliexpress.com/item/1005003780381390.html> [Accedido: Ago. 20, 2022].

- [41] T. M, "Pair of cross-cutter 2"with lightweight centres for brp hubs," Jan. 4, 2021. [En línea]. Disponible: <https://www.detroitperformancerc.com/product-page/pair-of-cross-cutter-2-with-lightweight-centres-for-brp-hubs> [Accedido: Jul. 20, 2022].
- [42] R. K. Noon, *Engineering analysis of vehicular accidents*. CRC Press, 1994.
- [43] S. Chauhan, "Motor torque calculations for electric vehicle," *International journal of scientific & technology research*, vol. 4, no. 8, pp. 126–127, 2015.
- [44] J. T. Allison, T. Guo, and Z. Han, "Co-design of an active suspension using simultaneous dynamic optimization," *Journal of Mechanical Design*, vol. 136, no. 8, 2014.
- [45] R. G. Budynas, J. K. Nisbett *et al.*, *Shigley's mechanical engineering design*. McGraw-hill New York, 2011, vol. 9.
- [46] e. group, "Application guidelines: Calculation data." Oct. 2020. [En línea]. Disponible: [https://www.elbe-group.de/files-elbe-group/EN\\_KAT\\_ElsoElbe\\_2020.pdf](https://www.elbe-group.de/files-elbe-group/EN_KAT_ElsoElbe_2020.pdf) [Accedido: Ago. 20, 2022].
- [47] A. Cardoso, J. Pardal, R. Chales, C. Martins, M. Silva, S. Tavares, B. Pedroza, and C. Barbosa, "Fatigue resistance performance of universal cardan joint for automotive application," *Engineering Failure Analysis*, p. 106128, 2022.
- [48] M. A. Hobson, "Light-weight multi-layer insulating enclosure," Oct. 3 1989, uS Patent 4,871,597.
- [49] L. Väin, T. Biene, S.-J. Annus, and P. Uustal, "Device and system for insulating items during delivery by a mobile robot," Apr. 21 2020, uS Patent 10,625,926.
- [50] Munich Matmatch, "Acrylonitrile styrene acrylate (asa)." Oct. 2020. [En línea]. Disponible: <https://matmatch.com/es/materials/mbas017-acrylonitrile-styrene-acrylate-asa-> [Accedido: Ago. 20, 2022].
- [51] Munich Matmatch, "Acrylonitrilo butadieno estireno (abs)." Oct. 2020. [En línea]. Disponible: <https://matmatch.com/es/materials/tizi0002-abs-3d> [Accedido: Ago. 20, 2022].

- [52] N. Kumar, Z. Vámosy, and Z. M. Szabó-Resch, "Robot path pursuit using probabilistic roadmap," in *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 2016, pp. 000 139–000 144.
- [53] T. MathWorks, "Robotics system toolbox," 2021. [En Línea]. Disponible: [https://la.mathworks.com/help/robotics/index.html?s\\_tid=CRUX\\_lftnav](https://la.mathworks.com/help/robotics/index.html?s_tid=CRUX_lftnav) [Accedido: Sept. 7, 2022].
- [54] R. E. Kalman, "A new approach to linear and prediction problems," 1960.
- [55] T. MathWorks, "IMU and GPS Fusion for Inertial Navigation," 2022. [En Línea]. Disponible: <https://la.mathworks.com/help/fusion/ug/imu-and-gps-fusion-for-inertial-navigation.html> [Accedido: Sept. 8, 2022].
- [56] G. Jocher, "Yolov5." May. 18, 2020. [En línea]. Disponible: <https://github.com/ultralytics> [Accedido: Ago. 12, 2022].
- [57] R. TECHNOLOGY, "Datasheet: Intel® realsensetm d400 series product family." Jan. 2019. [En línea]. Disponible: <https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf> [Accedido: Ago. 1, 2022].
- [58] RealSense TECHNOLOGY, "IntelRealSense." Jan. 2019. [En línea]. Disponible: <https://github.com/IntelRealSense/librealsense> [Accedido: Ago. 2, 2022].
- [59] Swarooph Seshadri, "Mobile robot simulation for collision avoidance with simulink," Mar. 21 2014 [En línea]. Disponible: [https://la.mathworks.com/videos/mobile-robot-simulation-for-collision-avoidance-with-simulink-90193.html?s\\_tid=srchtitle\\_robot%20mobil\\_11](https://la.mathworks.com/videos/mobile-robot-simulation-for-collision-avoidance-with-simulink-90193.html?s_tid=srchtitle_robot%20mobil_11) [Accedido: Sept. 2, 2022].
- [60] M. Samuel, M. Hussein, and M. B. Mohamad, "A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle," *International Journal of Computer Applications*, vol. 135, no. 1, pp. 35–38, 2016.
- [61] V. K. S. Patel and A. Pandey, "Modeling and performance analysis of pid controlled bldc motor and different schemes of pwm controlled bldc motor," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, pp. 1–14, 2013.

- [62] S. Automation, "57BL04 brushless dc motor rated 36v 4000rpm 180w," 2019. [En línea]. Disponible: [https://www.alibaba.com/product-detail/57BL04-brushless-dc-motor-rated-36v\\_60788318095.html](https://www.alibaba.com/product-detail/57BL04-brushless-dc-motor-rated-36v_60788318095.html) [Accedido: Sept 1, 2022].
- [63] JKONGMOTOR, "57BLS Motor Dc sin escobillas/36V BLDC Motor/ 4000rpm Motor BLDC," 2019. [En línea]. Disponible: <https://spanish.alibaba.com/product-detail/57BLS-Dc-Brushless-Motor-36V-BLDC-60144069196.html> [Accedido: Sept 1, 2022].
- [64] NVIDIA CORPORATION, "NVIDIA JETSON NANO DEVELOPER KIT," 2019. [En línea]. Disponible: <https://siliconhighway.com/wp-content/gallery/jetson-nano-devkit-datasheet-936542-US-hr.pdf> [Accedido: Jul. 20, 2022].
- [65] Arduino, "2560 Datasheet," 2019. [En línea]. Disponible: <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf> [Accedido: Jul. 25, 2022].
- [66] E. Morgan, "HC-SR04 Ultrasonic Sensor," 2014. [En línea]. Disponible: <https://pdf1.alldatasheet.es/datasheet-pdf/view/1132203/ETC2/HC-SR04.html> [Accedido: Ago 1, 2022].
- [67] U. Electronics, "IMU MPU6050 6 Grados de libertad," 2016. [En línea]. Disponible: <https://uelectronics.com/producto/imu-mpu6050-6-grados-de-libertad/> [Accedido: Jul. 5, 2022].
- [68] I. Store, "4G-3G GPS GNSS MODULE FOR JETSON NANO LTE CAT4 GLOBAL VERSION (SIM7600G-H)," 2022. [En línea]. Disponible: <https://www.iot-store.com.au/products/4g-gps-gnss-jetson-nano-lte-sim7600g-h> [Accedido: Ago 15, 2022].
- [69] Intel, "Intel® RealSense™ Depth Camera D415," 2017. [En línea]. Disponible: [https://www.mouser.ec/pdfdocs/Intel\\_D415\\_ProductBrief.pdf](https://www.mouser.ec/pdfdocs/Intel_D415_ProductBrief.pdf) [Accedido: Jul 20, 2022].
- [70] N. GROUP, "Motor DC planetario sin escobillas," 2019. [En línea]. Disponible: <https://es.sogears.com/resumen-de-productos/Motor-de-corriente-continua-sin-escobillas-73> [Accedido: Jul 25, 2022].

- [71] M. Live, "Controlador de Motor sin escobillas, placa controladora de 12V, 24V, 48V, CC de 6V-60V, 400W, BLDC trifásico, PWM," 2022. [En línea]. Disponible: <https://es.aliexpress.com/item/1005001726928239.html> [Accedido: Ago. 20, 2022].
- [72] HETPRO, "Servomotor TowerPro MG996," 2022. [En línea]. Disponible: <https://hetpro-store.com/servomotor-towerpro-mg996/> [Accedido: Jul 20, 2022].
- [73] T. Instruments, "LM2596 Regulador Step Down 25W 3A," 2022. [En línea]. Disponible: <https://uelectronics.com/producto/modulo-regulador-ajustable-lm2596-dc-dc-step-down-3a-1-25-30v/> [Accedido: Ago. 10, 2022].
- [74] J. Larminie and J. Lowry, *Electric vehicle technology explained*. John Wiley & Sons, 2012.
- [75] B. Matmatch, "Aisi 416." Oct. 2020. [En línea]. Disponible: <https://matmatch.com/materials/mitf937-aisi-416-hardened> [Accedido: Ago. 20, 2022].
- [76] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.
- [77] D. Mery and C. Pieringer, "Deep learning in x-ray testing," in *Computer Vision for X-ray Testing*. Springer, 2021, pp. 275–336.
- [78] A. Gaikwad, "Staircase dataset." Oct. 20, 2021. [En línea]. Disponible: <https://data.mendeley.com/datasets/7m97gp4yz9/1> [Accedido: Ago. 25, 2022].
- [79] MathWorks, "Spatial contact force: Model contact between two geometries." Jul. 30, 2019. [En línea]. Disponible: <https://la.mathworks.com/help/physmod/sm/ref/spatialcontactforce.html> [Accedido: Jul. 14, 2022].
- [80] T. MathWorks, "Mechanics explorer: Visualize and explore multibody models." Jan. 2012. [En línea]. Disponible: <https://la.mathworks.com/help/physmod/sm/ref/mechanicsexplorer-app.html> [Accedido: Jul. 16, 2022].
- [81] T. MathWorks, Jan. 2012. [En línea]. Disponible: <https://la.mathworks.com/help/physmod/sm/bodies.html> [Accedido: Jun. 19, 2022].

[82] T. MathWorks, "Utilities," Jan. 2018. [En línea]. Disponible: <https://la.mathworks.com/help/physmod/simscape/modeling-environment-and-interface.html> [Accedido: Jul. 10, 2022].

# APÉNDICES

# APÉNDICE A

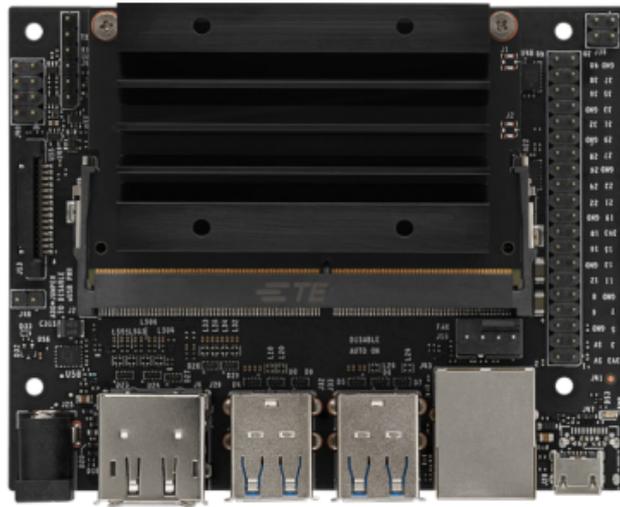
## Dispositivos Electrónicos del Robot

- **Computadora a bordo:** Es el cerebro del robot, aquí se ejecutan los diferentes algoritmos del sistema, como la localización, planificación de trayectorias, y visión por computadora. De los elementos disponibles en el mercado, se tomaron como alternativas a la Raspberry Pi 4B, Jetson nano y ASUS Tinker Board. En la Tabla A.1 se presenta una tabla comparativa de estas con sus principales características.

**Tabla A.1 Comparación de las alternativas de computadora a bordo**

Dispositivo	CPU	GPU	Peso [g]
Raspberry Pi4B	ARM A72@1.5GHz.	VideoCore VI	46
Jetson Nano	ARM A57@1.43GHz	NvidiaMaxwell	136
ASUS Tinker Board	RK3288 @1.8GHz	Mali-T764	55

Las alternativas presentan características similares en cuanto a CPU y memoria, sin embargo, en GPU la Jetson nano es superior y esto se debe a que fue diseñada específicamente para aplicaciones de inteligencia artificial y robots autónomos. Por otro lado, con respecto al peso, tanto la Raspberry como la Asus TinkerBoard son muy ligeras en comparación con la Jetson nano. No obstante, a pesar de ser la alternativa con mayor peso y dimensiones, la Jetson nano es elegida gracias a que su GPU permite una gran capacidad de procesamiento en los algoritmos de control y visión por computadora. En la Figura A.2 se puede visualizar la placa y en la Tabla A.2 sus especificaciones técnicas.



**Figura A.1 NVIDIA Jetson nano 4GB [64]**

**Tabla A.2 Especificaciones técnicas de la tarjeta NVIDIA Jetson nano [64]**

<b>Especificaciones</b>	<b>Datos</b>
CPU	Quad-Core ARM A57 @1.43 GHz
GPU	128-core Maxwell
Memoria RAM	4 GB
Connectividad	Gigabit Ethernet, M.2 Key E para Wi-Fi
GPIO	40 pines
Puertos USB	4x USB 3.0
Display	HDMI 2.0 y eDP 1.4
Almacenamiento	Micro SD
Cámara	MIPI CSI-2 DPHY
Alimentación	5V / 2A DC
Dimensiones	100 x 80 x 29 mm

- **Placa secundaria:** Es una tarjeta de desarrollo basada en un microcontrolador. Se conecta a la computadora por el puerto USB y mediante serial, envía los datos de los sensores y recibe los comandos de velocidad para los motores. Para elegir el modelo adecuado, se tomaron en cuenta las placas de desarrollo de Arduino, debido a su facilidad de uso, alta disponibilidad, software de código abierto y multiplataforma. El modelo elegido fue el Arduino Mega, Figura A.2, debido a que se requiere una gran cantidad de pines para el manejo de los sensores, conexión con la computadora y los controladores de los motores. Sus especificaciones técnicas se presentan en la Tabla A.3.

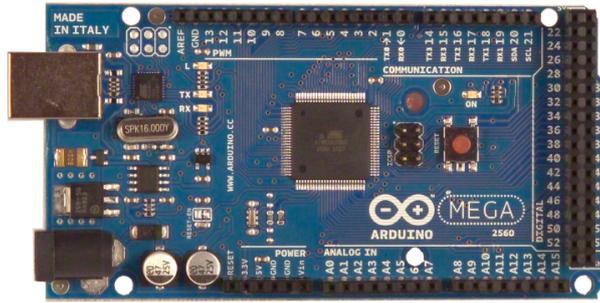


Figura A.2 Arduino Mega [65]

Tabla A.3 Especificaciones técnicas del Arduino Mega [65]

Especificaciones	Datos
Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada recomendado	7 - 12V
Pines digitales I/O	54
Pines PWM	14
Pines analógicos	16
Corriente DC(pines I/O)	40 mA
Tamaño	102 x 54 mm
Peso	37 g

- Sensor de ultrasonido:** El modelo a utilizar es el HC-SR04. Estos van conectados al Arduino y son utilizados para la detección de obstáculos. Se plantea utilizar siete en la parte frontal y dos en la parte posterior. En la Figura A.3 se observa el modelo y en la Tabla A.4 sus especificaciones técnicas.



Figura A.3 Sensor HC-SR04 [66]

**Tabla A.4 Especificaciones del sensor HC-SR04 [66]**

<b>Especificaciones</b>	<b>Datos</b>
Voltaje de operación	5V DC
Corriente de operación	15 mA
Rango de detección	0.2 - 4m
Resolución	0.3 cm
Ángulo de medición	30°
Dimensiones	45x 20x 15mm
Peso	10 g

- **IMU:** En la Figura A.4 se muestra el módulo GY-521, una IMU con giroscopio y acelerómetro de 6 ejes en total. Este módulo se comunica al Arduino mediante I2C. En la Tabla A.5 se encuentran sus especificaciones.



**Figura A.4 Módulo GY-521 [67]**

**Tabla A.5 Especificaciones del módulo GY-521 [67]**

<b>Especificaciones</b>	<b>Datos</b>
Modelo	MPU-6050
Grados de libertad	6
Comunicación	I2C
Sensores	giroscopio y acelerómetro
Voltaje de operación	5V
Corriente de operación	3.9 mA
Tamaño	15 x 20 mm
Peso	3 g

- **Módulo GPS+4G+Wifi:** En la Figura A.5 se observa un modulo extra para la Jetson nano. Este se conecta mediante USB y es montado sobre los pines de la tarjeta principal. Este módulo le permite tener conexión Wi-fi, 4G y GNSS. En la Tabla A.6 se presentan las especificaciones del GPS.



**Figura A.5 Módulo GPS+4G+Wifi [68]**

**Tabla A.6 Especificaciones de la conexión GNSS [68]**

<b>Especificaciones</b>	<b>Datos</b>
Receptor	16 canales
Sensibilidad de seguimiento	-159 dBm
Aranques en frío	-148 dBm
Precisión	inferior a 2.5m CEP
Temperatura de funcionamiento	-30°C a 80°C
Velocidad de transmisión	9600 bps - 115200 bps

- Cámara de profundidad:** En la Figura A.6 se observa el modelo de la cámara seleccionado. Esta cuenta con un rango de medición ideal de 0.5m a 3m y un campo de visión de 65°. En la Tabla A.7 se observan las demás especificaciones. Esta cámara es ideal para aplicaciones tanto de interiores como exteriores y en este proyecto se la utilizará para la detección de obstáculos específicos como escaleras y rampas.



**Figura A.6** Cámara de profundidad Intel RealSense D415 [69]

**Tabla A.7** Especificaciones de la cámara [69]

Especificaciones	Datos
Tecnología de profundidad	Estereoscópico
Resolución del obturador	1280 x 720
Fotogramas por segundo	90 fps
Rango ideal	0.5 - 3 m
Campo de visión	65°
Resolución RGB	1080p @ 30fps
Conexión	USB 3.0
Dimensiones	99 x 20 x 23 mm
Peso	241 g

- Motor BLDC:** Con los datos obtenidos en la tabla 2.9, se seleccionó un motor BLDC, de la serie 57BL con una caja reductora para aumentar su torque y reducir la velocidad. En la Figura A.7 se observa el modelo y en la Tabla A.8 sus especificaciones.

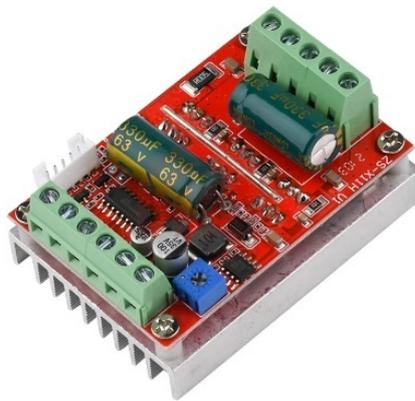


**Figura A.7** Motor BLDC con caja reductora [70]

**Tabla A.8 Especificaciones del motor [70]**

Especificaciones	Datos
Modelo del motor	57BL041
Modelo de la caja reductora	56JXE300k
Voltaje nominal	36 V
Potencia	177 W
Corriente nominal	6A
Reducción	1/23
Velocidad nominal	174 rpm
Par nominal	7.82 Nm
Peso	1.25 kg

- Controlador de Motores BLDC:** Es un módulo para el control de motores BLDC con sensores de efecto hall. Este recibe la señal PWM del Arduino y la envía en forma de voltaje al motor para alcanzar una velocidad deseada. Esta placa es alimentada con los 36 V que necesita el motor para su operación. Por recomendaciones del fabricante, se requiere el uso conjunto de un disipador, tal como se observa en la Figura A.8. Las especificaciones técnicas se muestran en la Tabla A.9.



**Figura A.8 Controlador de motor BLDC [71]**

**Tabla A.9 Especificaciones del controlador BLDC [71]**

Especificaciones	Datos
Voltaje de operación	9 - 60V
Potencia máxima	400 W
Corriente máxima	16A
Control de velocidad	PWM
Dimensiones	63 x 45 x 31 mm
Peso	73 g

- **Servomotor:** Es utilizado para accionar el seguro del compartimento. Este es controlado directamente por el Arduino con una señal PWM, sin embargo, los pines de alimentación van conectados a buses externos. En la Figura A.9 se muestra el modelo del servomotor seleccionado y en la Tabla A.10 sus especificaciones técnicas.



**Figura A.9 Servomotor TowerPro MG995 [72]**

**Tabla A.10 Especificaciones del servomotor TowerPro MG996 [72]**

<b>Especificaciones</b>	<b>Datos</b>
Voltaje de operación	4 a 7.2 V DC
Corriente de operación	900 mA
Torque a 4.8V	0.98 Nm
Engranajes	Metálicos
Dimensiones	40.6 x 19.8 x 42.9mm
Peso	55 g

- **Convertidor DC-DC Buck:** Son dispositivos utilizados para entregar diferentes niveles de voltaje desde la batería hacia el circuito. Se plantea tener un canal de 9V para alimentar el Arduino y uno de 5V para la placa principal, el servomotor y demás componentes. El modelo a utilizar es el LM2596, Figura A.10, un regulador reductor, con un 92 % de eficiencia y que requiere el uso de un disipador al trabajar con corrientes mayores a 2.5 A. En la Tabla A.11 se muestran sus especificaciones técnicas.



**Figura A.10 Regulador LM2596 [73]**

**Tabla A.11 Especificaciones del regulador LM2596 [73]**

<b>Especificaciones</b>	<b>Datos</b>
Voltaje de entrada	4.5V a 40V DC
Voltaje de salida	1.23V a 37V DC
Corriente de salida	3A
Potencia de salida	25 W
Eficiencia de conversión	92 %
Dimensiones	43x21x13mm
Temperatura	-45 °C a +85 °C
Peso	11 g

# APÉNDICE B

## Dimensionamiento del motor

El dimensionamiento del motor se desarrolla en función de las condiciones externas a las que estará expuesto el robot, considerando un pendiente de 30°, siendo la máxima inclinación a la que podrá ascender sin problemas. Las fuerzas a considerar son:

- La fuerza producto de la resistencia a la rodadura se define en la ec. B.1, donde  $m$  es la masa,  $g$ , la aceleración de la gravedad, y  $\mu_{rr}$ , el coeficiente de fricción dinámica:

$$F_{rr} = \mu_{rr}mg, \quad (\text{B.1})$$

- la fuerza producida por el peso total, es la resistencia a la pendiente y se define como:

$$F_{hc} = mg, \quad (\text{B.2})$$

- la fuerza aerodinámica, donde  $A$  es el área frontal,  $C_d$ , el coeficiente de arrastre,  $v$ , la velocidad máxima deseada, y  $\rho$ , la densidad del aire a 33°:

$$F_{ad} = \frac{1}{2}\rho AC_d v^2, \quad (\text{B.3})$$

- La fuerza de aceleración, necesaria para proveer la aceleración lineal del vehículo, donde  $a$  es la aceleración lineal.

$$F_{la} = ma, \quad (\text{B.4})$$

Ahora, la fuerza total de tracción total es la suma de todas estas fuerzas previamente definidas [74]:

$$F_{te} = F_{rr} + F_{ad} + F_{hc} + F_{la}, \quad (\text{B.5})$$

$$F_{te} = 6.671 + 0.263 + 196.2 + 11.11 = 214.24 \text{ N.}$$

Y para cada rueda, la fuerza total se divide para 4:

$$F_{te/4} = 53.56 \text{ N.}$$

El torque requerido en todas las ruedas motrices combinadas para mover el robot móvil se define como:

$$T_w = F_{te} R_w R_f, \quad (\text{B.6})$$

donde  $F_{te}$  es el esfuerzo de tracción total,  $R_w$  el radio de la rueda y  $R_f$  el factor de fricción por perdidas. El torque requerido calculado es:

$$T_w = 21 \text{ Nm},$$

y el torque para cada rueda será:

$$T_{w/4} = 5.25 \text{ Nm.}$$

Considerando un factor de seguridad (FoS) de 1.25:

$$T_{w/4} = 5.25 * FoS = 6.56 \text{ Nm.}$$

Comprobando el torque máximo de tracción  $T_{MT}$  que pueden alcanzar las cuatro ruedas antes de que se produzca el deslizamiento, suponiendo una distribución de pesos igual, donde  $W_w$  es el peso de cada rueda,  $\mu_s$  el coeficiente de fricción y  $R_w$  el radio de la rueda:

$$T_{MT} = 4W_w \mu_s R_w, \quad (\text{B.7})$$

$$T_{MT} = 24.03 \text{ Nm},$$

siendo  $T_{MT} > T_w$ , lo que indica que no existe deslizamiento, ya que el torque del robot no supera su máximo valor. Por otro lado, se obtiene la velocidad angular mediante la ec.

(B.8):

$$v = \omega R, \quad (\text{B.8})$$

$$\omega = 15.90 \text{ rad/s} = 151.6 \text{ RPM},$$

y la potencia se obtiene mediante la ec. (B.9):

$$P = T_{w/4}\omega, \tag{B.9}$$

$$P = 104.2 \text{ W}.$$

# APÉNDICE C

## Diseño de resorte

Las variables independientes en el diseño del muelle se visualizan en la Figura 2.8 y para empezar es necesario definir el diámetro de la hélice  $D$ , el diámetro del alambre  $d$ , el paso del muelle  $p$  y el número de bobinas activas  $N_a$ . Con estas variables se calculan las dimensiones del resorte, la Tabla C.1 detalla las fórmulas de acuerdo al tipo de extremo de la pieza.

**Tabla C.1 Fórmulas para calcular las dimensiones de resortes de compresión [45]**

Término	Tipo de extremos de resortes			
	Plano	Plano y esmerilado	A escuadra y cerrado	A escuadra y esmerilado
Espiras de extremo, $N_e$	0	1	2	2
Espiras totales, $N_t$	$N_a$	$N_a + 1$	$N_a + 2$	$N_a + 2$
Longitud libre, $l_0$	$pN_a + d$	$p(N_a + 1)$	$pN_a + 3d$	$pN_a + 2d$
Longitud sólida, $l_s$	$d(N_t + 1)$	$dN_t$	$d(N_t + 1)$	$dN_t$
Paso, $p$	$(l_0 - d)/N_a$	$l_0/(N_a + 1)$	$(l_0 - 3d)/N_a$	$(l_0 - 2d)/N_a$

En este caso se seleccionó de extremo cuadrado y rectificado debido a su nivel de tolerancia en las cargas y reducción de traslapo. Para empezar el diseño, se define:

- **Diámetro de la hélice:** 30 mm
- **Diámetro del alambre:** 4 mm
- **Paso del muelle:** 6 mm
- **Número de bobinas activas:** 14

Entonces, se tendría que el número total de espiras:

$$N_t = N_a + 2 \tag{C.1}$$

$$N_t = 16 \text{ espiras}$$

la longitud libre sería:

$$L_0 = pN_a + 2d \quad (\text{C.2})$$

$$L_0 = 92 \text{ mm}$$

y la longitud máxima de compresión:

$$L_s = dN_t \quad (\text{C.3})$$

$$L_s = 64 \text{ mm}$$

El material seleccionado es el alambre de piano o musical, se usa con mucha frecuencia por los fabricantes y tiene las especificaciones señaladas en el Tabla C.2.

**Tabla C.2 Propiedades mecánicas de algunos alambres para resorte [45]**

Material	Límite elástico, porcentaje de $S_{utr}$		Diámetro $d$ , pulg	$E$		$G$	
	tensión, torsión	torsión		Mpsi	GPa	Mpsi	GPa
Alambre de piano A228	65-75	45-60	<0.032	29.5	203.4	12.0	82.7
			0.033-0.063	29.0	200	11.85	81.7
			0.064-0.125	28.5	196.5	11.75	81.0
			>0.125	28.0	193	11.6	80.0
Resorte estirado duro A227	60-70	45-55	<0.032	28.8	198.6	11.7	80.7
			0.033-0.063	28.7	197.9	11.6	80.0
			0.064-0.125	28.6	197.2	11.5	79.3
			>0.125	28.5	196.5	11.4	78.6
Templado en aceite A239	85-90	45-50		28.5	196.5	11.2	77.2
Resorte de válvula A230	85-90	50-60		29.5	203.4	11.2	77.2
Cromo vanadio A231	88-93	65-75		29.5	203.4	11.2	77.2
A232	88-93			29.5	203.4	11.2	77.2
Cromo silicio A401	85-93	65-75		29.5	203.4	11.2	77.2

Por otro lado, hay que tener en cuenta que los muelles con  $C < 4$  son difíciles de formar, y los muelles con  $C > 12$  pueden enredarse. Estos requisitos proporcionan la primera restricción con la ec. C.5 y se verifica con la ec. C.4.

$$C = \frac{D}{d} \quad (\text{C.4})$$

$$C = \frac{30}{4} = 7.5$$

Lo cual indica que está dentro de rango permisible

$$4 < C < 12 \quad (\text{C.5})$$

la siguiente restricción evita el pandeo:

$$L_0 - 5.26D \leq 0 \quad (\text{C.6})$$

$$-20.2 \leq 0$$

Luego se necesita estimar la constante del resorte con la ec. 4.2, donde  $C$  es el índice del muelle y  $G$  es el módulo de corte del material (80 MPa).

$$k_s = \frac{d^4 G}{8D^2 N_a \left(1 + \frac{1}{2C^2}\right)} \quad (\text{C.7})$$

$$k_s = \frac{(4 \cdot 10^{-3})^4 \cdot (80 \cdot 10^9)}{8 \cdot 7.5^2 \cdot 14 \left(1 + \frac{1}{2 \cdot 7.5^2}\right)}$$

$$k_s = 201 \text{ N/m}$$

El esfuerzo cortante de resorte  $\tau$  en la deflexión máxima no debe superar la resultante de  $S_{sy}$ . El siguiente modelo de esfuerzo cortante incorpora el factor de aumento de Bergsträsser, que corrige el efecto de curvatura y el cortante directo. Donde la fuerza axial en la altura del muelle y se calcula mediante la ec. C.8:

$$F_s = k_s(L_0 - L_s) \quad (\text{C.8})$$

$$F_s = 201(92 - 64) \cdot 10^{-3}$$

$$F_s = 5.6 \text{ N}$$

entonces:

$$\tau = \left(\frac{4C + 2}{4C - 3}\right) \frac{8F_s D}{\pi d^3} \quad (\text{C.9})$$

$$\tau = \left(\frac{4 \cdot 7.5 + 2}{4 \cdot 7.5 - 3}\right) \frac{8 \cdot 5.6 \cdot 30 \cdot 10^{-3}}{\pi(4 \cdot 10^{-3})^3}$$

$$\tau = 7.9 \text{ MPa}$$

Ahora se definen las fuerzas de amplitud y media, y para eso se considera el peso del robot. Donde:

- $F_{mín}$ : 433 N
- $F_{máx}$ : 509 N

Se calcula la fuerza de amplitud con:

$$F_a = \frac{F_{máx} - F_{mín}}{2} \quad (C.10)$$

$$F_a = \frac{509 - 433}{2}$$

$$F_a = 38.2 \text{ N}$$

y luego la fuerza de amplitud con:

$$F_m = \frac{F_{máx} + F_{mín}}{2} \quad (C.11)$$

$$F_m = \frac{509 + 433}{2}$$

$$F_m = 471 \text{ N}$$

**Tabla C.3 Constantes para estimar la resistencia de tensión mínima de alambres [45]**

Material	ASTM núm.	Exponente <i>m</i>	Diámetro, pulg	<i>A</i> , kpsi · pulg <sup><i>m</i></sup>	Diámetro, mm	<i>A</i> , MPa · mm <sup><i>m</i></sup>	Costo relativo del alambre
Alambre de piano*	A228	0.145	0.004-0.256	201	0.10-6.5	2 211	2.6
Alambre T y R en aceite <sup>†</sup>	A229	0.187	0.020-0.500	147	0.5-12.7	1 855	1.3
Alambre estirado duro <sup>‡</sup>	A227	0.190	0.028-0.500	140	0.7-12.7	1 783	1.0
Alambre al cromo vanadio <sup>§</sup>	A232	0.168	0.032-0.437	169	0.8-11.1	2 005	3.1
Alambre al cromo silicio <sup>  </sup>	A401	0.108	0.063-0.375	202	1.6-9.5	1 974	4.0
Alambre inoxidable 302 <sup>#</sup>	A313	0.146	0.013-0.10	169	0.3-2.5	1 867	7.6-11
		0.263	0.10-0.20	128	2.5-5	2 065	
		0.478	0.20-0.40	90	5-10	2 911	
Alambre de bronce fosforado**	B159	0	0.004-0.022	145	0.1-0.6	1 000	8.0
		0.028	0.022-0.075	121	0.6-2	913	
		0.064	0.075-0.30	110	2-7.5	932	

De la Tabla C.3 y con la ec. C.12 se obtiene:

$$S_{ut} = \frac{A}{d^m} \quad (\text{C.12})$$

$$S_{ut} = \frac{2211}{4^{0.145}}$$

$$S_{ut} = 1936 \text{ MPa}$$

**Tabla C.4 Esfuerzos de torsión máximos permisibles [45]**

Material	Porcentaje máximo de la resistencia a la tensión	
	Antes de la remoción de la deformación (incluye $K_W$ o $K_B$ )	Después de la remoción de la deformación (incluye $K_s$ )
Alambre de piano y acero al carbono estirado en frío	45	60-70
Acero al carbono templado y revenido y acero de baja aleación	50	65-75
Aceros inoxidables austeníticos	35	55-65
Aleaciones no ferrosas	35	55-65

A partir de la Tabla C.4 se obtiene el factor a considerar  $S_{sy}$ :

$$S_{sy} = 0.45S_{ut} \quad (\text{C.13})$$

$$S_{sy} = 871 \text{ MPa}$$

Dando como resultado que  $\tau < S_{sy}$ . Por otro lado, también se obtiene la amplitud del esfuerzo cortante con la ec. C.14:

$$\tau_a = \left( \frac{4C + 2}{4C - 3} \right) \frac{8F_a D}{\pi d^3} \quad (\text{C.14})$$

$$\tau_a = \left( \frac{4 \cdot 7.5 + 2}{4 \cdot 7.5 - 3} \right) \frac{8 \cdot 38 \cdot 30 \cdot 10^{-3}}{\pi(4 \cdot 10^{-3})^3}$$

$$\tau_a = 54 \text{ MPa}$$

y el esfuerzo cortante medio con C.15:

$$\tau_m = \left( \frac{4C + 2}{4C - 3} \right) \frac{8F_m D}{\pi d^3} \quad (\text{C.15})$$

$$\tau_m = \left( \frac{4 \cdot 7.5 + 2}{4 \cdot 7.5 - 3} \right) \frac{8 \cdot 471 \cdot 30 \cdot 10^{-3}}{\pi (4 \cdot 10^{-3})^3}$$

$$\tau_m = 667 \text{ MPa}$$

Por último, con la ec. C.17 se aplica el criterio de fatiga de Soderberg con datos de Zimmerli y se define el factor de seguridad  $n_d$  de 1.5, donde  $S_{e2}$  se calcula con:

$$S_{e2} = \frac{0.24 S_{ut}}{n_d} \quad (\text{C.16})$$

$$S_{e2} = \frac{0.24 \cdot 1936}{1.5}$$

$$S_{e2} = 3097 \text{ MPa}$$

con la condición de:

$$\frac{\tau_a}{S_{e2}} + \frac{\tau_m}{S_{sy}} - 1 \leq 0 \quad (\text{C.17})$$

$$\frac{7.9}{3097} + \frac{667}{871} - 1 \leq 0$$

$$-0.22 \leq 0$$

Se verifica que cumpla el criterio:

$$\frac{\tau_a n_d - (241)10^6}{(241)10^6} \leq 0 \quad (\text{C.18})$$

$$\frac{54 \cdot 1.5 - (241)10^6}{(241)10^6} \leq 0$$

$$-0.66 \leq 0$$

Dado que el diseño cumple las desigualdades y las restricciones previamente mostradas, el resorte es apto a trabajar en las condiciones propuestas.

# APÉNDICE D

## Diseño de eje cardan

La Figura D.1 muestra el comportamiento de dos ejes de diferentes configuraciones. En el caso del robot, el primer diseño es el que refleja su dinámica. En los ejes de entrada y salida también se visualiza esfuerzos que se someten periódicamente a cargas de flexión alternas.

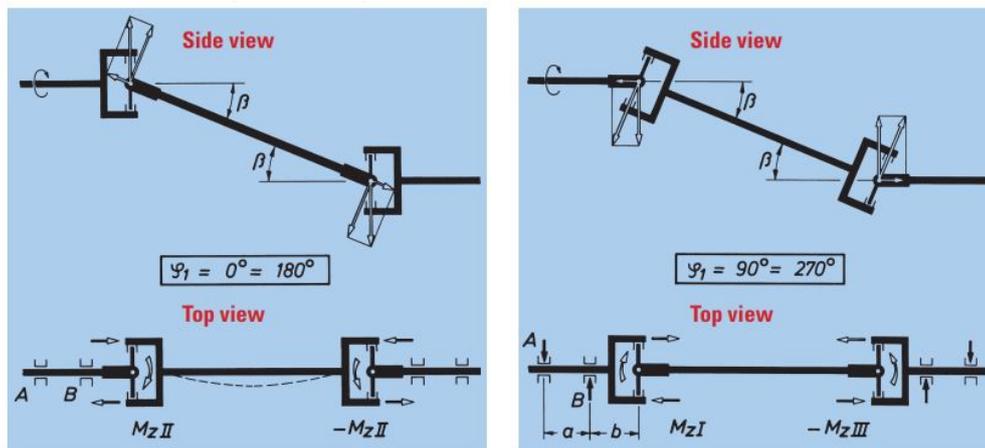


Figura D.1 Fuerzas en el eje de entrada y salida con disposición en Z [46]

Dependiendo del tipo de accionamiento o de la aplicación correspondiente, un árbol cardán puede estar sometido a cargas de choque que son considerablemente por encima del par nominal. Para tenerlos en cuenta Hay que utilizar los factores de choque o seguridad que se muestran en la Tabla D.1:

Tabla D.1 Factor de seguridad sugerido para el diseño del eje cardán [46]

Prime mover	with flexible coupling	without flexible coupling
Turbine or electric motor	1	1 to 1,5
Gasoline engine, 4 and more cylinders	1,25	1,75
Gasoline engine, 1 to 3 cylinders	1,5	2
Diesel engine, 4 and more cylinders	1,5	2
Diesel engine, 1 to 3 cylinders	2	2,5

Se consideraron los datos obtenidos de la Tabla 2.9 y las dimensiones del eje diseñado, a fin de confirmar que la pieza soporte las cargas esperadas:

- **Torque**= 6.56 Nm
- **Velocidad**: 152 RPM
- **Potencia**= 104 W
- **Ángulo de deflexión**= 15°
- **Diámetro interior del eje** = 20 mm
- **Diámetro exterior del eje** = 12 mm
- **Diámetro de la junta** = 7.2 mm

Se calcula el segundo momento polar de área, de la sección transversal del eje con la ec. D.1, donde  $x_2$  es el diámetro exterior del eje y  $x_1$  el diámetro interior:

$$J = \frac{\pi(x_2^4 - x_1^4)}{2} \quad (D.1)$$
$$J = \frac{\pi((10 \cdot 10^{-3})^4 - (6 \cdot 10^{-3})^4)}{2}$$
$$J = 1.37 \cdot 10^{-8}$$

Ahora se calcula el esfuerzo cortante con D.2, donde  $T$  es el torque y  $r$  el radio exterior:

$$\tau = \frac{Tr}{J} \quad (D.2)$$
$$\tau = \frac{6.56 \cdot 10 \cdot 10^{-3}}{1.37 \cdot 10^{-8}}$$
$$\tau = 4.78 \text{ MPa}$$

Considerando que se han seleccionaron dos materiales, para la horquilla y la junta [75]. El esfuerzo máximo soportado se calcula con la ec. D.3 donde  $G$  es el modulo de rigidez (77 GPa) y  $\tau_y$  el esfuerzo admisible del acero (135 MPa):

$$Y_{m\acute{a}x} = \frac{\tau_y}{G} \quad (D.3)$$

$$Y_{m\acute{a}x} = \frac{135}{77}$$

$$Y_{m\acute{a}x} = \frac{135}{77 \cdot 10^3}$$

$$Y_{m\acute{a}x} = 1.75 \cdot 10^{-3}$$

El factor de seguridad para el eje es:

$$FoS = \frac{\tau_y}{\tau} \quad (D.4)$$

$$FoS = \frac{135}{4.78}$$

$$FoS = 28.2$$

Se calculan los esfuerzos maximos tanto para la junta como para la horquilla, considerando sus dimensiones, primero se obtiene el rea seccionada de las dos piezas:

- rea de la horquilla, donde  $r_e$  es el radio de la secci3n donde la horquilla cruza con la junta.

$$A_h = \pi r_e^2 \quad (D.5)$$

$$A_h = \pi(3.6 \cdot 10^{-3})^2$$

$$A_h = 4.1 \cdot 10^{-5} \text{ m}^2$$

- rea de la junta, donde  $d_1$  es el dimetro exterior del eje.

$$A_j = 2r_e d_1 \quad (D.6)$$

$$A_j = 2 \cdot 3.6 \cdot 10^{-3} \cdot 6 \cdot 10^{-3}$$

$$A_j = 4.3 \cdot 10^{-4} \text{ m}^2$$

Tambi3n se calcula la carga basado en el torque con la ec. D.7:

$$P = \frac{T}{r_2} \quad (D.7)$$

$$P = \frac{6.56}{10 \cdot 10^{-3}}$$

$$P = 656 \text{ N}$$

Luego se calculan los esfuerzos máximos, sabiendo que  $\sigma_h$  es el esfuerzo calculado de la horquilla y  $\sigma_j$  es el esfuerzo calculado de la junta:

- Esfuerzo en la junta

$$\sigma_h = \frac{P}{A_h} \quad (\text{D.8})$$

$$\sigma_h = \frac{656}{4.1 \cdot 10^{-5}}$$

$$\sigma_h = 16 \text{ MPa}$$

- Esfuerzo en la junta

$$\sigma_j = \frac{P}{A_j} \quad (\text{D.9})$$

$$\sigma_j = \frac{656}{4.3 \cdot 10^{-5}}$$

$$\sigma_j = 15 \text{ MPa}$$

Finalmente, se obtienen los factores de seguridad, sabiendo que  $\sigma_{yh}$  es el esfuerzo del material de la horquilla (225 MPa) y  $\sigma_{yj}$  es el esfuerzo del material de la junta (250 MPa):

- para la horquilla

$$FoS_h = \frac{\sigma_{yh}}{\sigma_h} \quad (\text{D.10})$$

$$FoS_h = \frac{225}{16}$$

$$FoS_h = 14$$

- para la junta

$$FoS_j = \frac{\sigma_{yj}}{\sigma_j} \quad (D.11)$$

$$FoS_j = \frac{250}{15}$$

$$FoS_j = 17$$

Por lo anterior, se deduce que el eje puede soportar una potencia superior al que se ha calculado en el dimensionamiento del motor de robot.

# APÉNDICE E

## Análisis estático

Se realizó un análisis por elementos finitos de la pieza base. En el diseño mecánico del robot esa pieza corresponde al chasis, siendo la parte que lleva toda la carga de la batería, componentes electrónicos, y productos para la entrega. En este proceso de validación se consideró el ABS como material de la pieza.

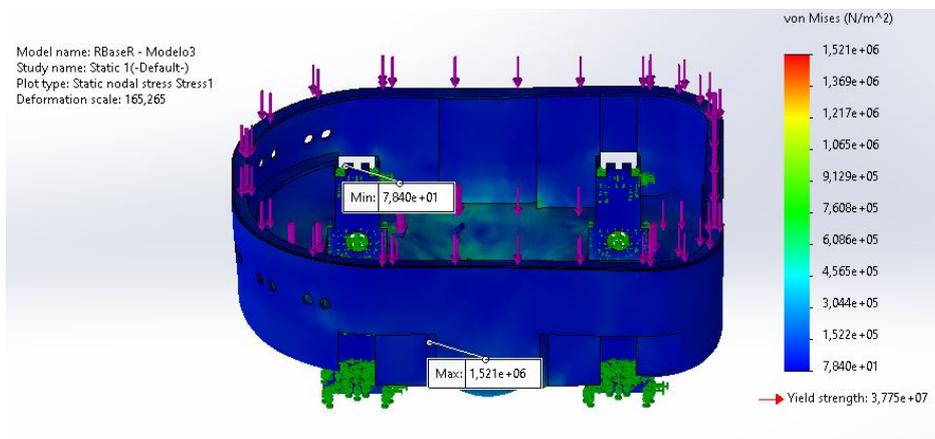


Figura E.1 Análisis estático por esfuerzos de von Mises.

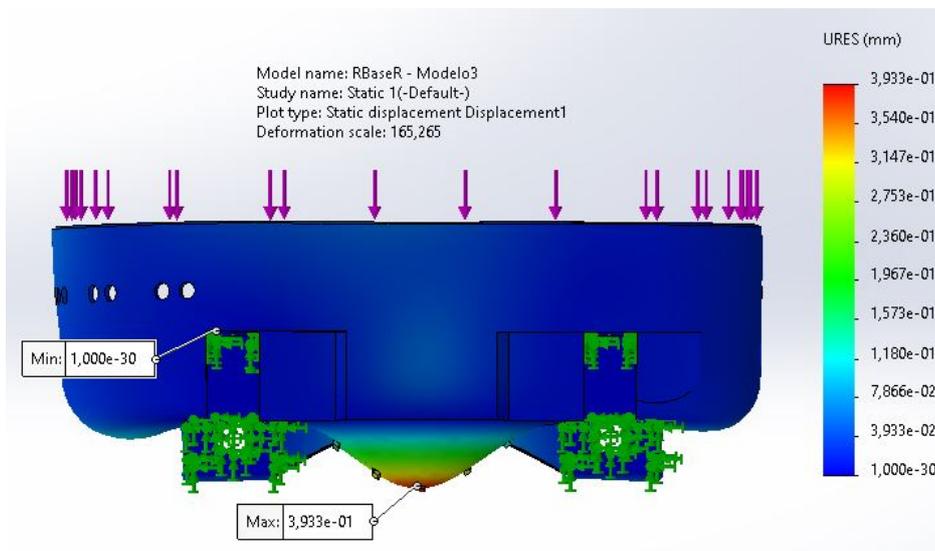
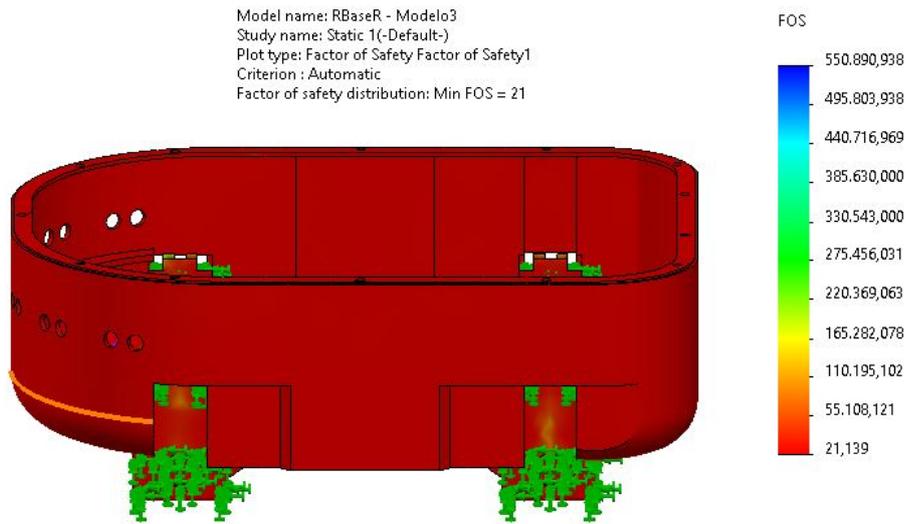


Figura E.2 Deformación lineal debido a las cargas.

En la Figura E.1 se visualiza el análisis por fatiga, donde la tensión máxima que sufre el chasis es menor al límite elástico del material, lo que indica que las cargas sobre la pieza son aceptables.

También en la Figura E.2 están las deformaciones que sufre la base al soportar las cargas, las cuales no son despreciables ya que son aproximadamente 5 cm. Por otro lado, en la Figura E.3 está el factor de seguridad estimado el cual no es menor a 1.



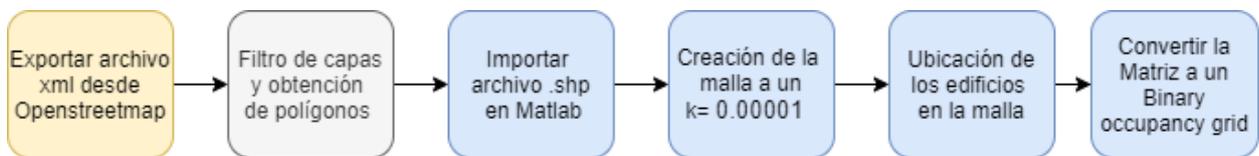
**Figura E.3 Factor de seguridad.**

Lo que se puede inferir respecto a este resultado, es que el chasis puede aguantar las cargas sin problema. En lo que respecta al diseño, de manera general, el análisis es aceptable.

# APÉNDICE F

## Obtención del *binary occupancy grid*

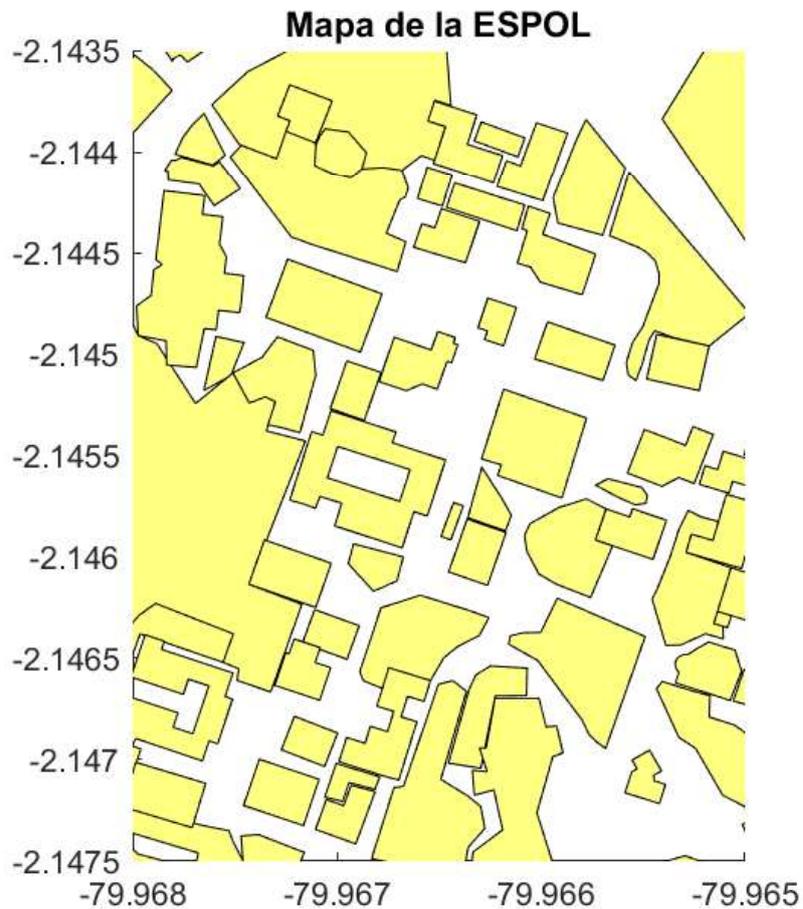
En la Figura F.1 se observan las diferentes etapas para la obtención del mapa binario, con el que se localiza el robot para hacer las entregas.



**Figura F.1 Metodología para la obtención del mapa**

En primer lugar, se delimita el entorno en el que va a trabajar el robot, siendo estos, puntos de entrega cercanos al Sweet & Coffee, ubicado en la Facultad de Ciencias Naturales y Matemáticas, y que además no tengan escaleras que impidan su traslado.

Teniendo un entorno definido, se procedió a exportar desde Openstreetmap un archivo .xml que contiene información de la ubicación de edificios, carreteras, escaleras, entre otros. Este archivo se lo procesa con una herramienta de filtrado de capas, hasta eliminar la información innecesaria del mapa y tener en un archivo únicamente los edificios. Este archivo contiene datos georeferenciados de los edificios del entorno seleccionado. En la Figura F.2 se observan los datos de este archivo, ya procesados en MATLAB.



**Figura F.2 Mapa con coordenadas georreferenciadas**

La última etapa consiste en la creación de una malla en el mapa anterior. Esta es una matriz de unos y ceros que dependiendo del paso, dimensión de cada celda de la malla, se obtiene una resolución en el mapa deseado. Los unos representan los lugares donde hay un obstáculos, mientras que los ceros el espacio disponible. Utilizando la herramienta Robotics Toolbox de MATLAB se utiliza la función `binaryoccupancygrid()` para finalmente crear el mapa presentado en la Figura F.3.



**Figura F.3 Binary occupancy grid**

En adición, se creó una función para convertir coordenadas georreferenciadas (Latitud, Longitud) a puntos X e Y dentro del mapa, pudiendo así ubicar los puntos deseados y crear las rutas globales.

# APÉNDICE G

## Pasos para iniciar un entrenamiento con YOLOv5

YOLO, acrónimo de 'You only look once', es un algoritmo de detección de objetos. Cuenta con diferentes versiones, de las cuales la versión 5 tiene mejoras con respecto al rendimiento. YOLO v5 utiliza el marco Pytorch que es más fácil de entender, además el entrenamiento del modelo es rápido, y el razonamiento por lotes produce resultados en tiempo real [76]. Para la implementación de YOLOv5 desde cero, se necesita [77] :

- **Paso 1: Base de datos (imágenes)**

En esta base de datos hay 150 imágenes y el formato del nombre es imax.jpg donde x es número de la foto [78].

- **Paso 2: Anotaciones (*bounding boxes* + clases)**

Con una base de datos previamente generada, ahora se delimitan los objetos a detectar con un cuadro y también definiendo la clase a la que pertenece con un color diferenciador.

El proceso de definir en un cuadro cada clases es para generar un archivo excel. El formato se define con las esquinas superior izquierda  $(x_1, y_1)$  e inferior derecha  $(x_2, y_2)$  como en la figura G.1. Además hay una columna para a la clase (es 0 para la clase escalera, y 1 para la clase rampa).

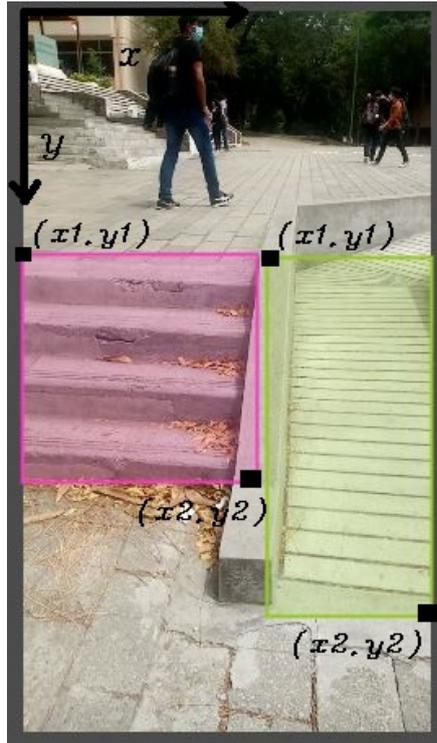


Figura G.1 Formato de anotaciones en las imágenes de entrenamiento

- Paso 3: Formatos adecuados para la versión** Con los datos del paso anterior, ahora se genera un archivo de texto por cada imagen, si la imagen se llama <nombreimagen>.jpg y tiene objetos a detectar, se debe crear un archivo <nombreimagen>.txt. Este proceso es muy tardío, pero se lo puede realizar con código, lo que se debe colocarse en ese archivo son los resultados de las siguientes ecuaciones:

$$x_n = \frac{1}{M} \frac{x_1 + x_2}{2} \quad (\text{G.1})$$

$$y_n = \frac{1}{N} \frac{y_1 + y_2}{2} \quad (\text{G.2})$$

$$w_n = \frac{x_2 - x_1}{M} \quad (\text{G.3})$$

$$h_n = \frac{y_2 - y_1}{N} \quad (\text{G.4})$$

Donde los números obtenidos corresponden a las variables:

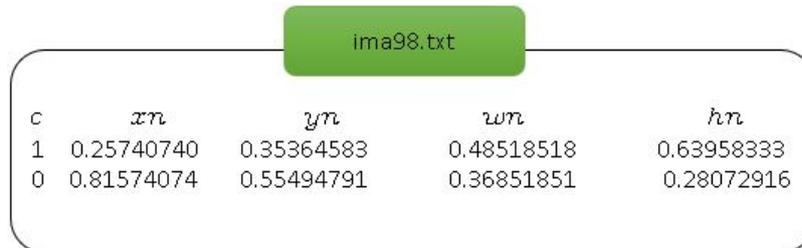
$x_n$ : centro del *bounding box* normalizado en la dirección  $x$ , donde  $M$  es el ancho en pixeles de la imagen.

$y_n$ : centro del *bounding box* normalizado en la dirección  $y$ , donde  $N$  es el alto en pixeles de la imagen.

$w_n$ : ancho normalizado del *bounding box*.

$h_n$ : alto normalizado del *bounding box*.

Dado que son 4 ecuaciones, la quinta variable y primera columna del archivo de texto es la clase y se debe considerar, ya que es parte del formato que recibe la versión 5 de YOLO tal y como se puede visualizar en la Figura G.2.



$c$	$x_n$	$y_n$	$w_n$	$h_n$
1	0.25740740	0.35364583	0.48518518	0.63958333
0	0.81574074	0.55494791	0.36851851	0.28072916

**Figura G.2 Formato para los datos en archivo de texto.**

#### ■ Paso 4: Definición de Train/Val/Test

Se dividen las imágenes de la base de datos en las carpetas de *training*, *validation* y *testing*. En este paso se copian con sus respectivos archivos de texto en cada directorios.

#### ■ Paso 5: Definición del archivo YAML

Finalmente se define el archivo `data.yaml` que contiene la información de los directorios de *training*, *validation* y *testing*, el nombre de las clases y el número de clases.

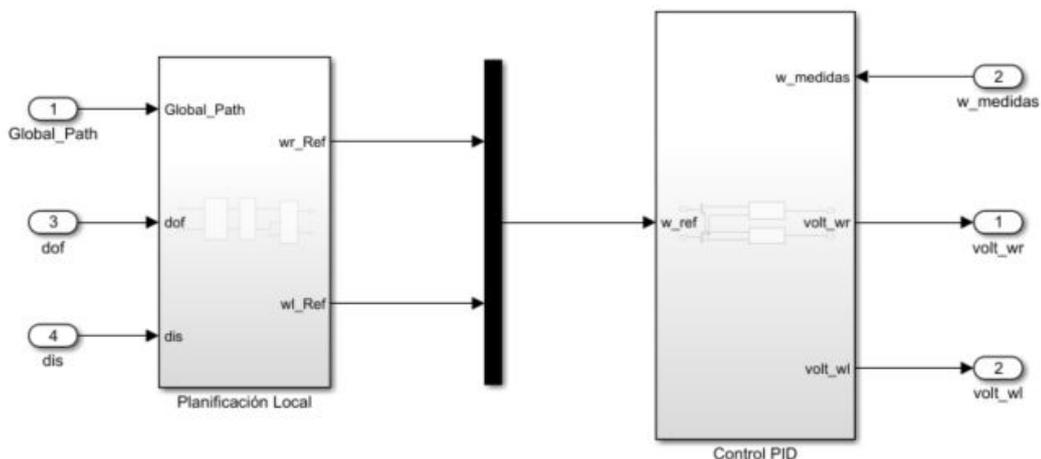
# APÉNDICE H

## Construcción del Modelo en Simulink

En esta sección se describen las metodologías y bloques principales del modelo desarrollado en MATLAB de la Figura 3.6.

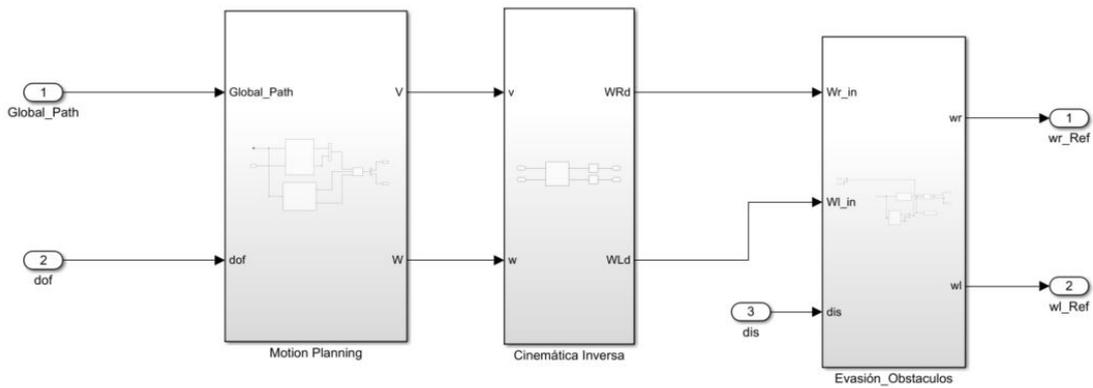
### ■ Bloques del sistema de control

En este bloque se implementan los diferentes algoritmos y métodos de control utilizados para el movimiento del robot. En la Figura H.1 Se presentan las dos etapas principales, teniendo la planificación local y el control PID. Este último recibe las velocidades angulares de las ruedas calculadas del planificador y las resta a las velocidades medidas del sistema dinámico, para obtener la señal de error y hacer la compensación.



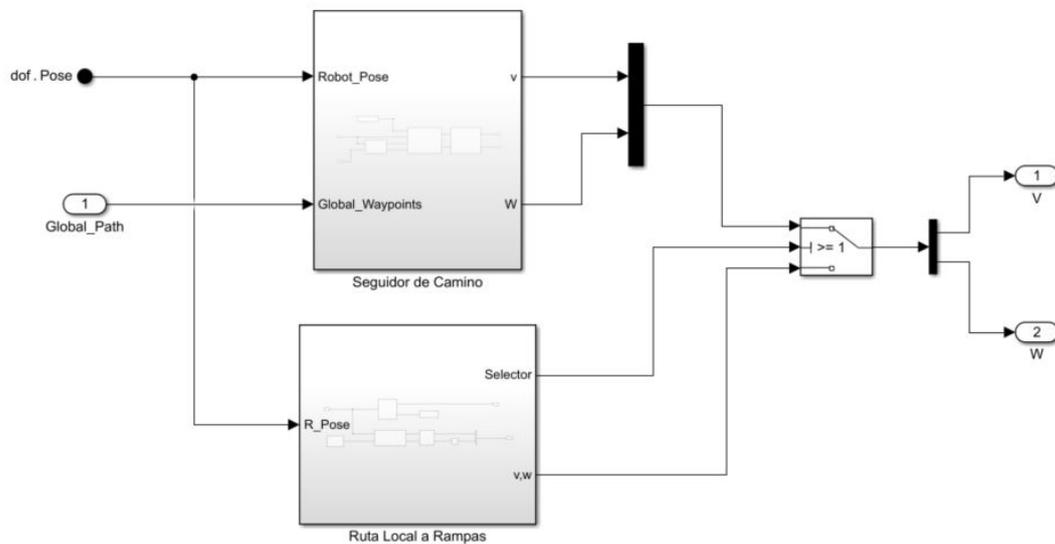
**Figura H.1 Bloque Sistema de Control**

Por otro lado, en la Figura H.2 se tienen las etapas que incluyen la planificación local. El bloque *Evasión Obstáculos* se encarga de modificar las velocidades angulares obtenidas del modelo de cinemática inversa, en base al algoritmo de evasión de obstáculos de la Figura 2.22. Las señales *dis* representan las distancias de los sensores de ultrasonido.



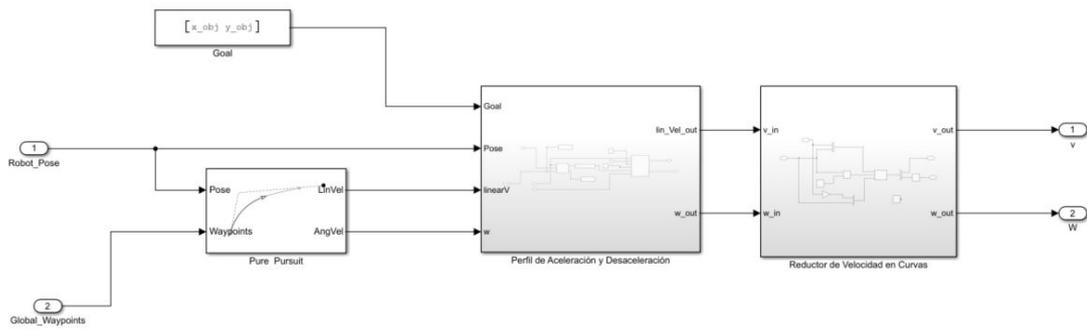
**Figura H.2 Bloque Planificación Local**

En la Figura H.3 se presentan dos bloques que utilizan el controlador de seguimiento de camino. El primero es para la ruta global, mientras que el otro tiene la ruta local de desviación cuando hay rampas. El bloque selector toma a la ruta global como salida por defecto, pero al estar detectar una rampa cercana, la señal de control del selector cambia y el robot debe planificar en la ruta que lo dirige hacia las rampas.



**Figura H.3 Bloque Motion Planning**

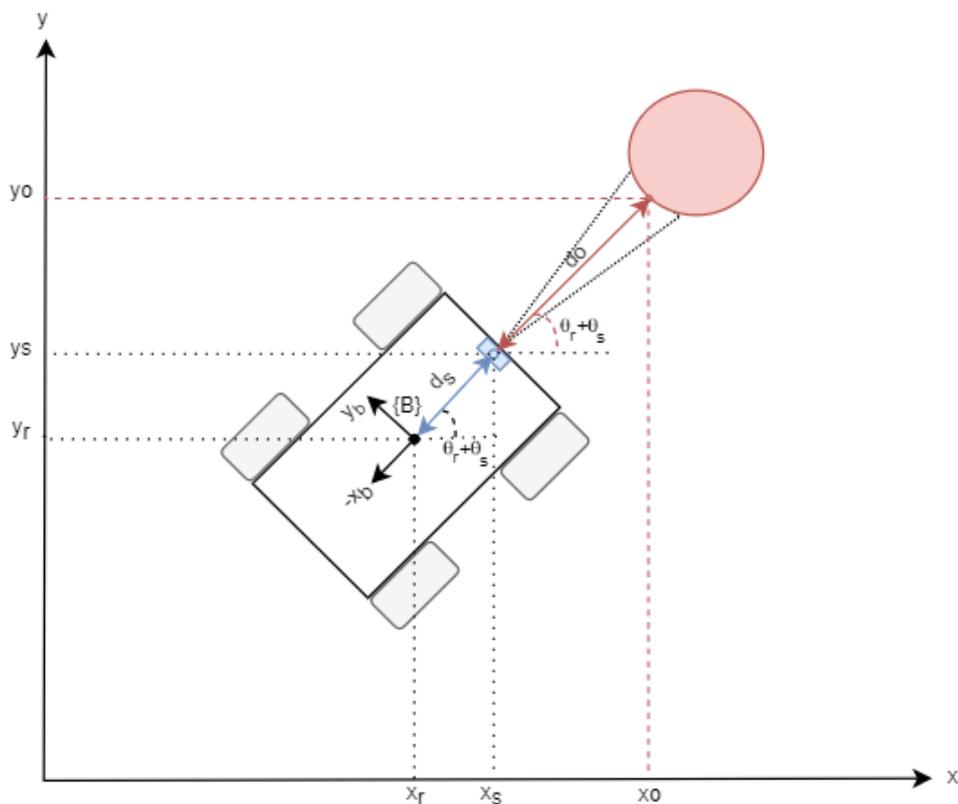
Finalmente, en la Figura H.4 se presentan las etapas del algoritmo de la Figura 2.23, encargado del seguimiento de camino y la reducción de la velocidad en caso de curvas o al estar cerca de la meta.



**Figura H.4 Bloque Seguidor de Camino**

■ **Modelado del sensor ultrasónico**

El sensor de ultrasonido siempre retorna la la distancia entre el sensor y el punto más cercano a él, dentro de su rango. En la Figura H.5 se presenta una gráfica con las variables involucradas en la detección de un objeto con un sensor de ultrasonido.



**Figura H.5 Posición del obstáculo con respecto al robot**

Dada la pose del robot  $(x_r, y_r, \theta_r)$ , la posición del sensor de ultrasonido se obtiene con la ecuación 4.2, donde  $x_s$  y  $y_s$  representan las coordenadas del sensor,  $d_s$  la distancia a la que se ubica el sensor con respecto al centro y  $\theta_s$  el desfase con

respecto al eje  $x_b$ . En el caso planteado en la Figura, el desfase es cero, ya que el sensor esta ubicado exactamente en el eje  $x_b$ .

$$x_s = x_r + d_s \cdot \cos(\theta_r + \theta_s) \quad (\text{H.1})$$

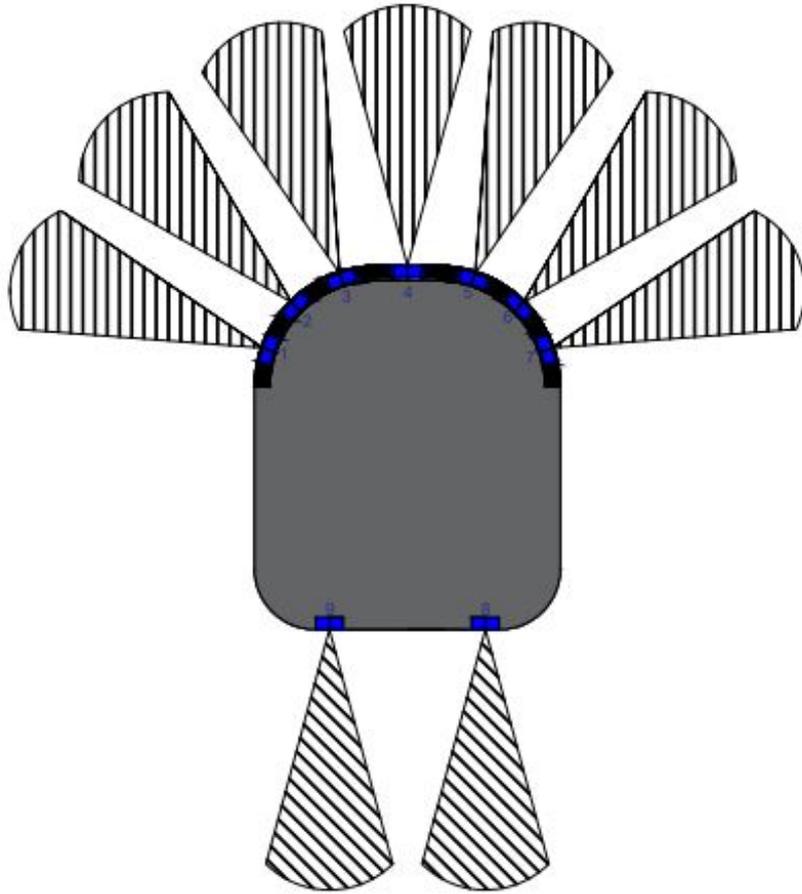
$$y_s = y_r + d_s \cdot \sin(\theta_r + \theta_s)$$

Por otro lado, la posición del obstáculo más cercano con respecto al sensor se obtiene con la ecuación H.2, donde  $x_o$  y  $y_o$  representan las coordenadas del punto más cercano del obstáculo y  $d_o$  la señal de distancia proveniente del sensor.

$$x_o = x_s + d_o \cdot \cos(\theta_r + \theta_s) \quad (\text{H.2})$$

$$y_o = y_s + d_o \cdot \sin(\theta_r + \theta_s)$$

Con las ecuaciones H.1 y H.2 se puede identificar la posición de un obstáculo conociendo la pose del robot, la ubicación de los sensores y las señales de distancia provenientes de estos. En la Figura H.6 se presenta un esquema de la ubicación de los sensores de ultrasonido y sus rangos de detección, que van desde 2 cm a 4 m con un ángulo de  $15^\circ$ .



**Figura H.6 Ubicación sensores ultrasónicos**

Con las ecuaciones planteadas, se puede modelar la acción del sensor en el entorno de Simulink. La posición de los obstáculos es conocida y la señal de distancia se lo calcula con la ecuación H.3. Además, con la ecuación H.4 se obtiene el ángulo entre el sensor y el punto del obstáculo, para determinar si ese punto está dentro del rango del sensor.

$$d_{so} = \sqrt{(x_s - x_o)^2 + (y_s - y_o)^2} \quad (\text{H.3})$$

$$\phi_s = \arctan\left(\frac{y_s - y_o}{x_s - x_o}\right) \quad (\text{H.4})$$

En la Figura H.7 se presenta el diagrama de flujo para obtener la respuesta de cada sensor. De haber mas de un punto en un rango de detección, se programa para elegir como salida la distancia de menor valor.

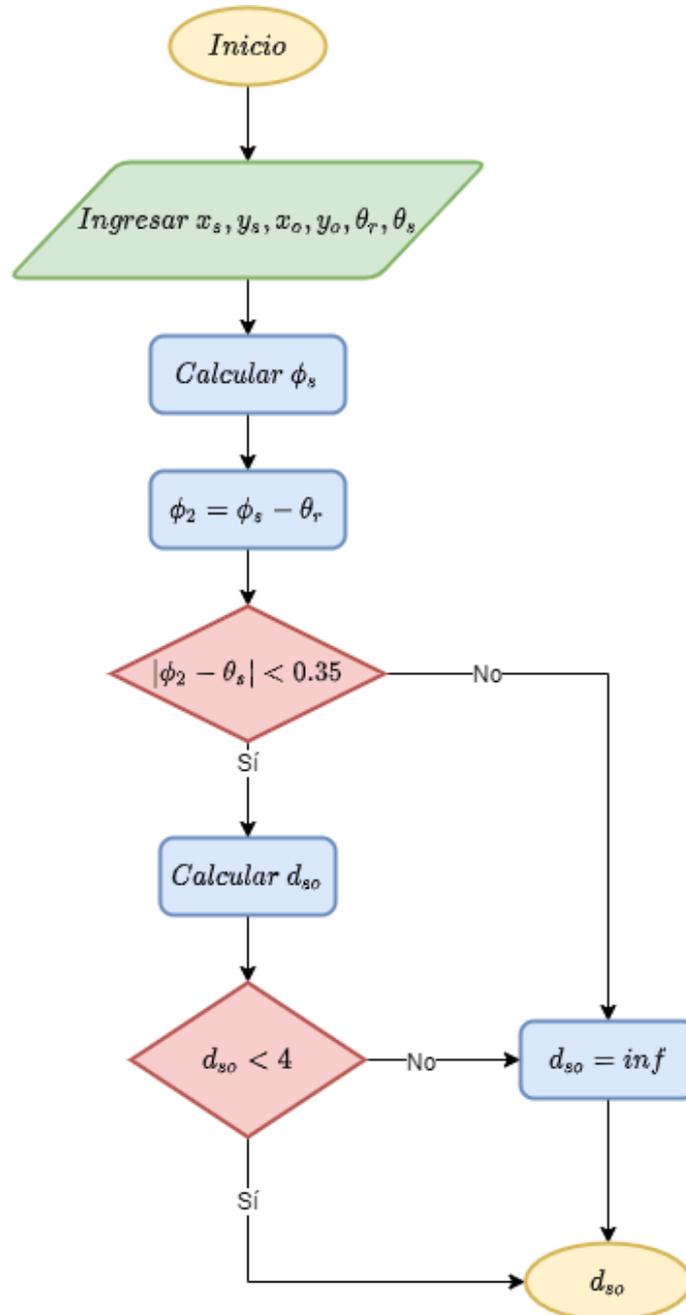


Figura H.7 Diagrama de flujo para modelar la señal del sensor ultrasónico

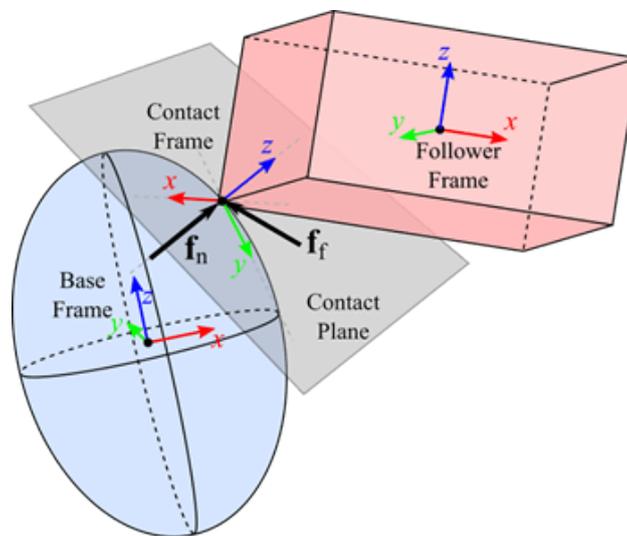
Para procesar la información se tomaron medidas de la ubicación de cada sensor  $d_s$  junto con su ángulo de desfase  $\phi_s$  en base a la Figura H.6. Los datos se presentan en la Tabla H.1.

**Tabla H.1 Ubicación geométrica de los sensores de ultrasonido**

Sensores									
Dato	1	2	3	4	5	6	7	8	9
$\theta_s$	60°	40°	20°	0°	-20°	-40°	-60°	-170°	-190°
$d_s$	295	290	295	300	295	290	295	305	305

### ■ Ambiente 3D

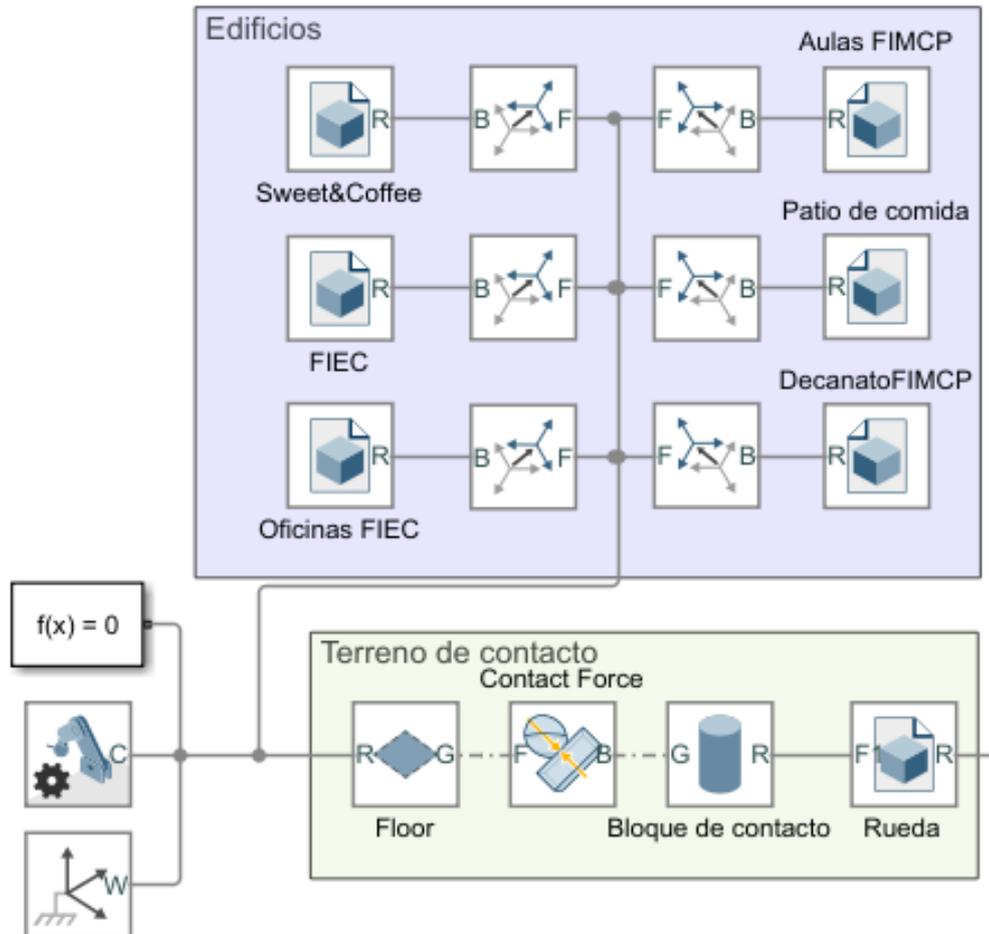
En el diseño del ambiente, Simulink cuenta con bloques que permiten crear conexiones menos ideales, es decir, generar una interacción de contacto y también gravedad con datos aproximados de fricción, rigidez, entre otros. El bloque *Spatial Contact Force* [79] tiene parámetros predeterminados, los cuales se pueden personalizar.



**Figura H.8 Modelo de contacto de fuerza entre dos geometrías [79]**

Los edificios 3D se obtuvieron con Blender, donde se importó una sección del mapa y se extrajo en tres dimensiones. Esos bloques fueron guardados como archivos con la extensión .stl, un formato compatible con Simulink, donde luego se importó con el bloque *Solid File* tal y como se detalla su función en la Tabla H.2. La visualización del modelo se lo realizó con una herramienta de Simscape [80], que permite explorar todo el modelo o una pieza, ya que muestra las propiedades de cada una.

La figura H.9 ejemplifica la conexión que debe existir con los bloques de los edificios, además de la relación de contacto que se debe establecer entre la rueda y el terreno.



**Figura H.9 Subsistemas de edificios y terreno con contacto de fuerza**

El contacto se genera con las superficies donde pasará el robot, es decir, si el robot sube pendientes entonces habrá que añadir otros bloques de contacto y relacionarlos con la geometría correspondiente. El subsistema Edificios tiene bloques donde se importan los archivos .stl, además de una transformación rígida para ser ubicados de acuerdo con el mapa.

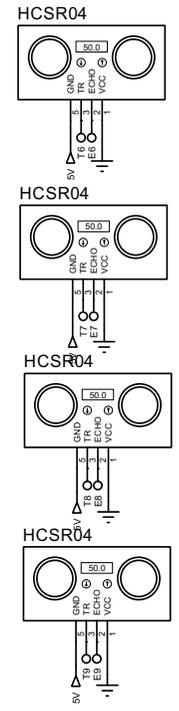
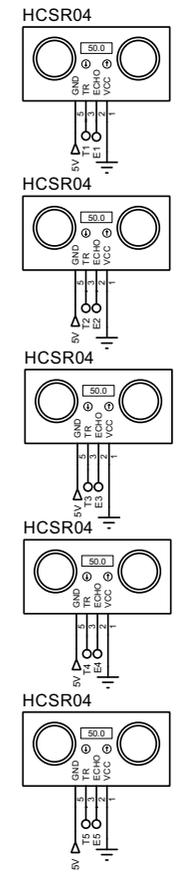
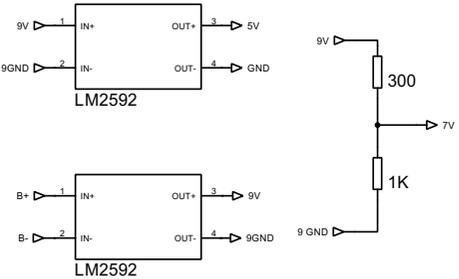
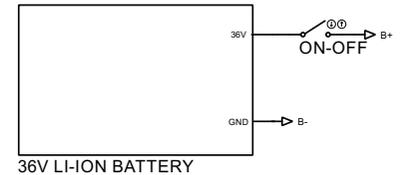
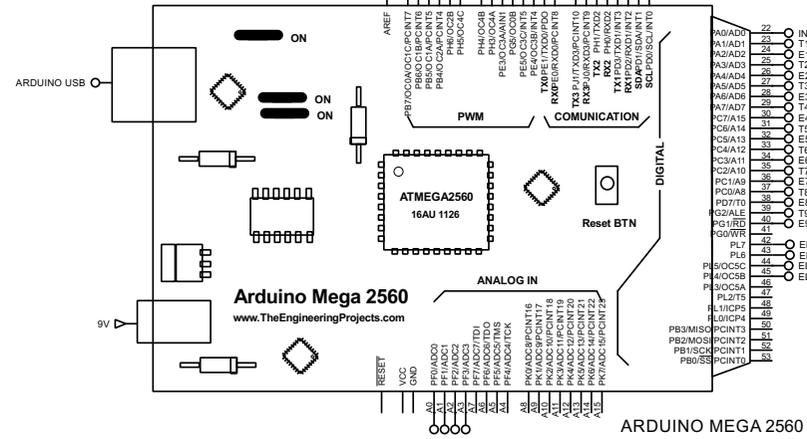
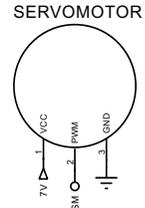
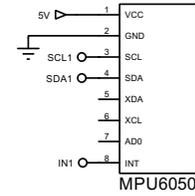
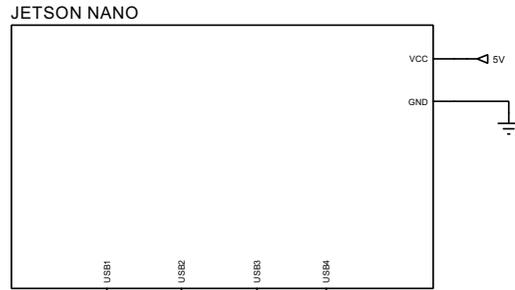
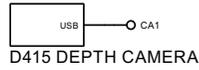
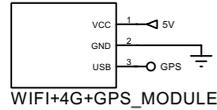
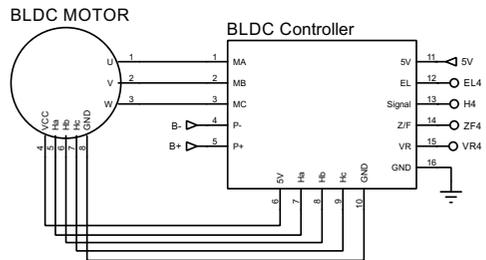
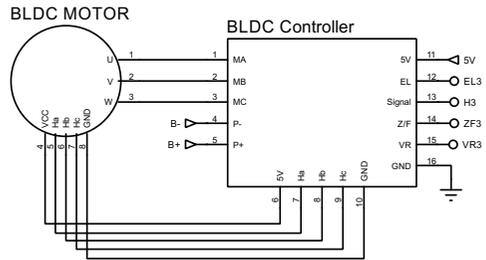
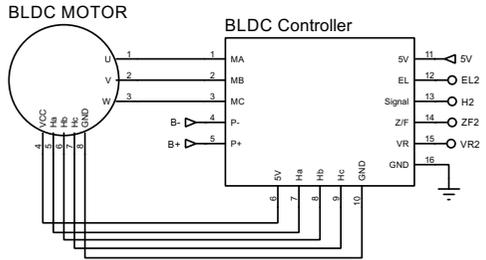
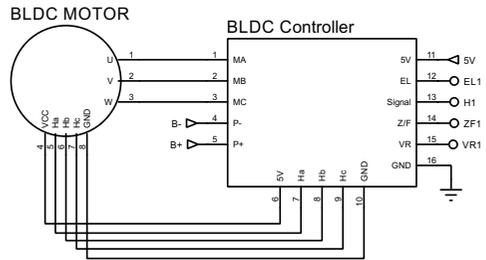
**Tabla H.2 Bloques principales del modelado de ambiente en Simulink**

	<p><i>Solver configuration</i>: especifica opciones de simulación para su sistema físico, cada diagrama de bloque Simscape lo requiere.</p>
	<p><i>Mechanism Configuration</i>: proporciona parámetros mecánicos y de simulación a un mecanismo, como la gravedad.</p>
	<p><i>World frame</i>: representa el marco de referencia global en un modelo. Este marco es inercial y en reposo absoluto.</p>
	<p><i>Infinite plane</i>: exporta un plano infinito para modelar problemas de contacto.</p>
	<p><i>File Solid</i>: modela un elemento sólido con geometría, inercia, color y marco de referencia derivado de un archivo externo.</p>
	<p><i>Spatial Contact Force</i>: modela el contacto entre geometrías asociadas con un par de cuerpos</p>
	<p><i>Rigid Transform</i>: rota y traslada el marco del puerto seguidor (F) con respecto al marco del puerto base (B)</p>

La Tabla H.2 muestra los principales bloques que intervienen en la creación del ambiente y también en la visualización del la simulación [81, 82].

# **APÉNDICE I**

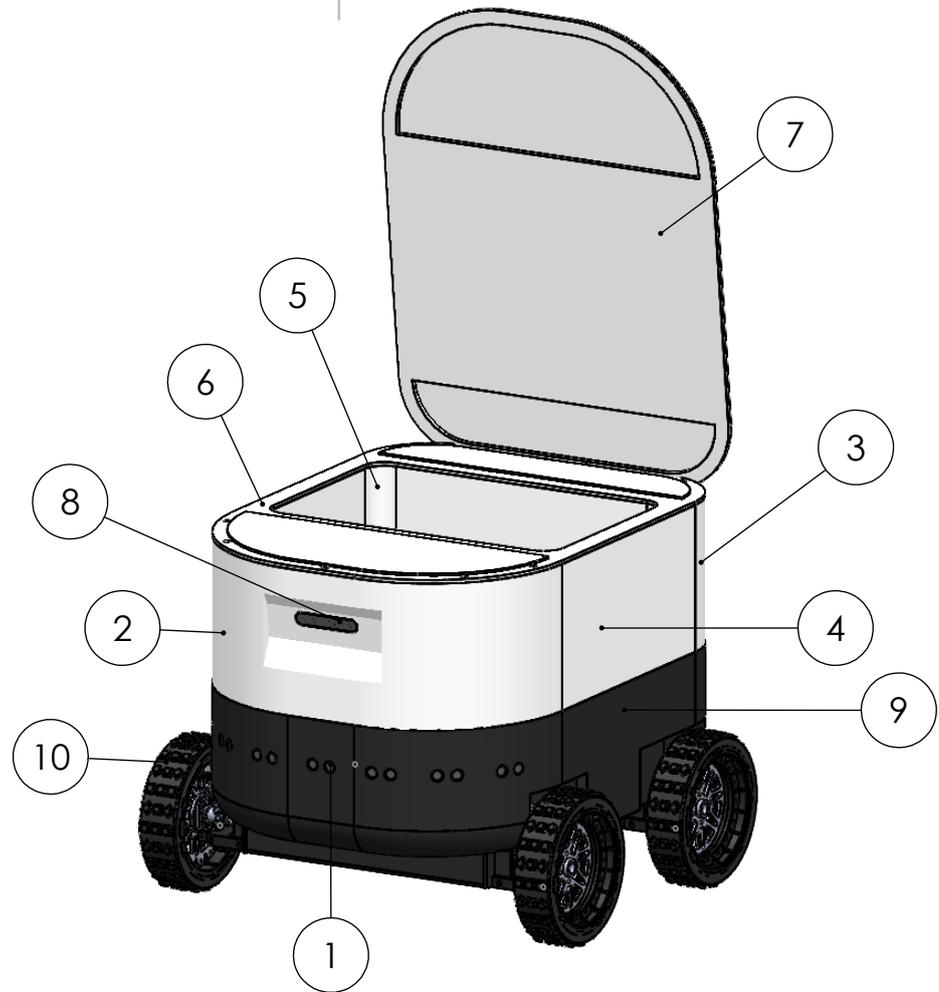
**Conexiones electrónicas del robot**



**Título:** Diseño electrónico del robot  
**Autores:** Alex Anrrango - Piedad Miranda  
**Fecha:** 11 sep. 2022  
**Página:** 1 - 1

# **APÉNDICE J**

**Planos del diseño mecánico**



ITEM NO.	COMPONENTE
1	Sensor ultrasónico
2	Pieza frontal
3	Pieza posterior
4	Pieza lateral
5	Compartimento
6	Base de la tapa
7	Tapa
8	Cámara RealSende
9	Chasis
10	Llanta

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

DENOMINACIÓN:  
**Partes principales del robot**

**espol**

Varios

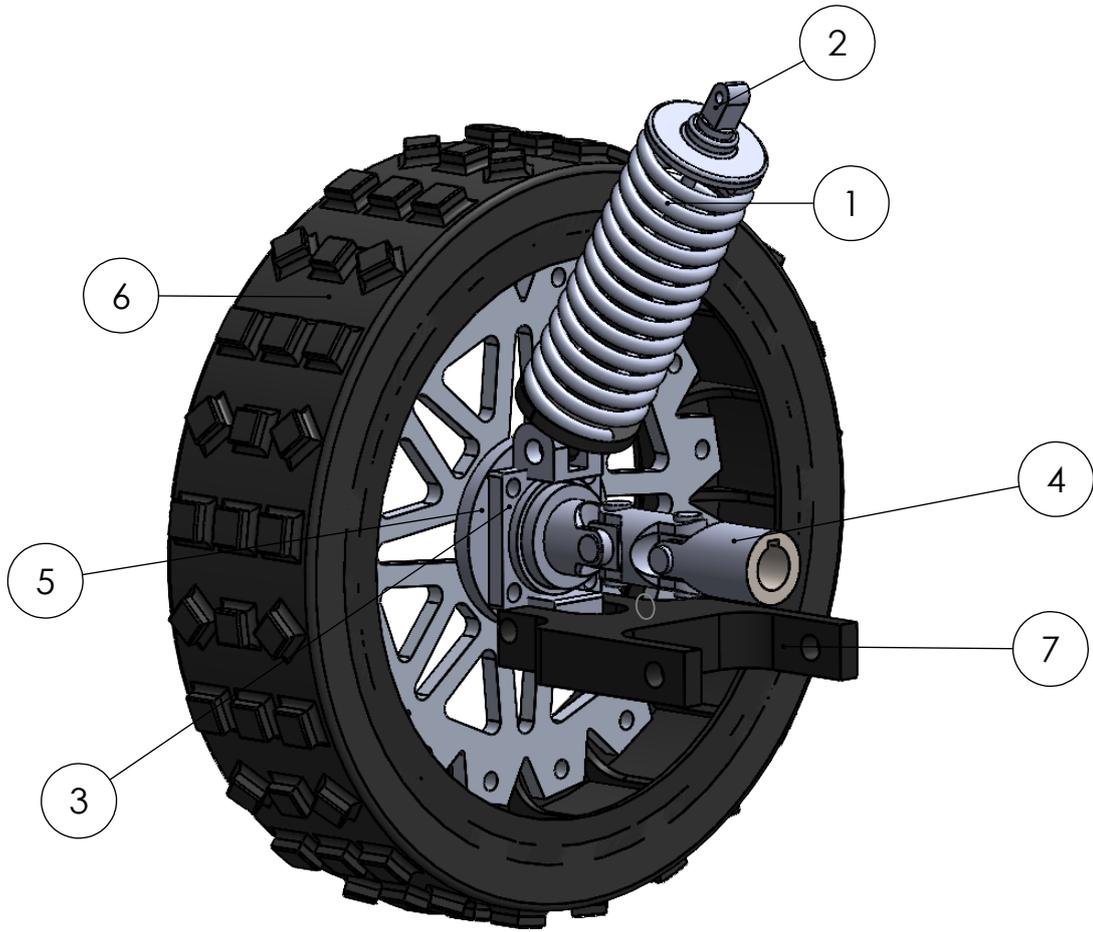
NÚMERO DE DIBUJO:  
**1**

A4

PESO:

SCALE:1:20

HOJA 1 DE 1



ITEM NO.	COMPONENTES
1	Resorte
2	Amortiguador
3	Soporte de buje
4	Eje cardán
7	Horquilla
6	Buje
7	Horquilla

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Varios

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

DENOMINACIÓN:

Partes del sistema de suspensión

NÚMERO DE DIBUJO:

2

A4

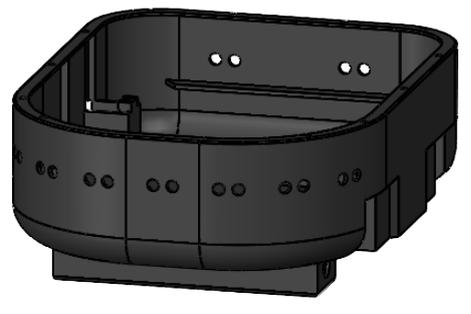
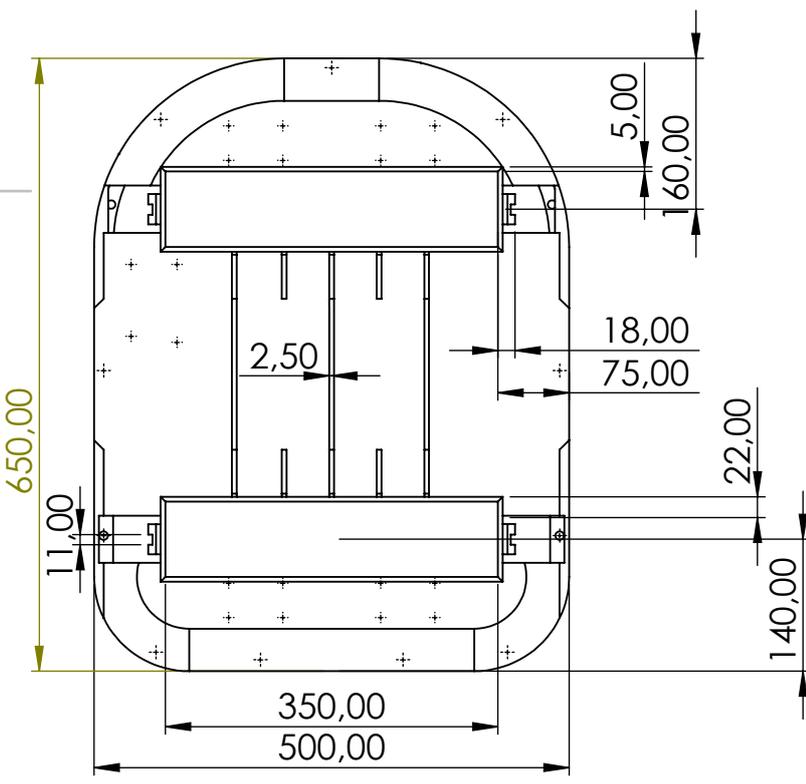
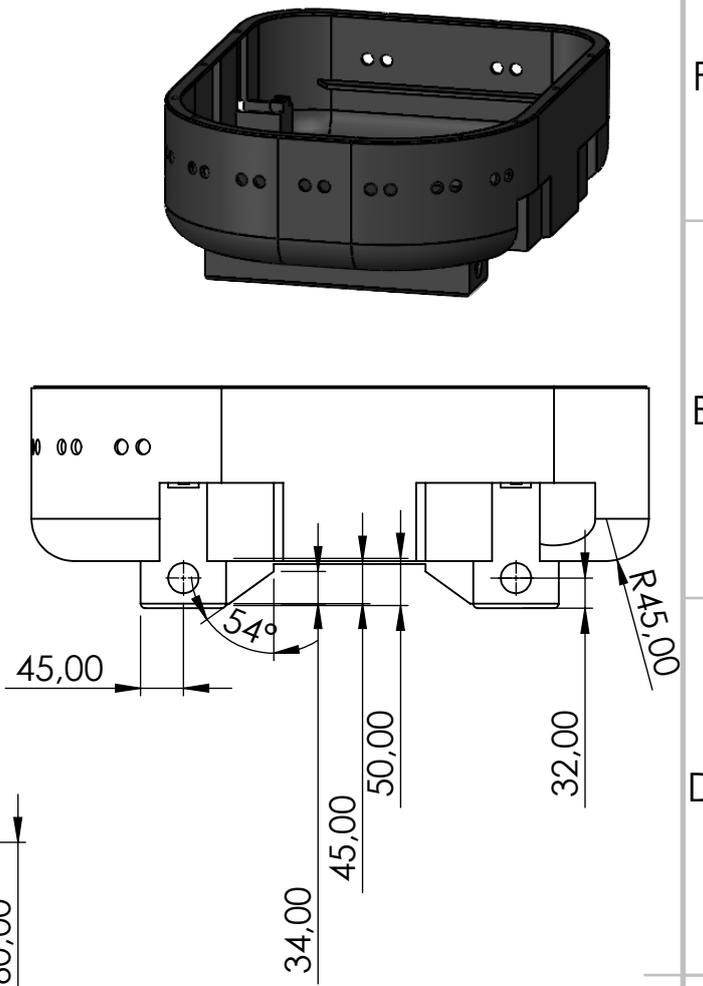
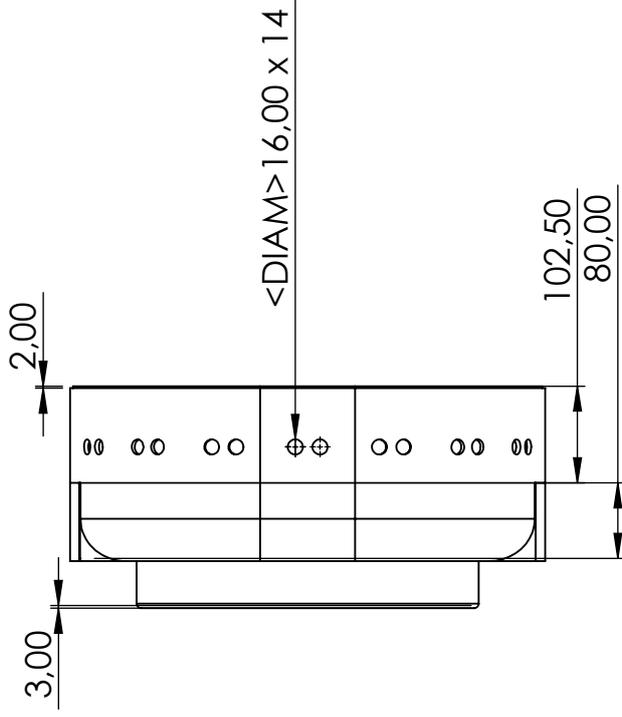
espol

PESO:

SCALE:1:5

HOJA 1 DE 1





A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

ABS

DENOMINACIÓN:

Chasis

NÚMERO DE DIBUJO:

3

A4

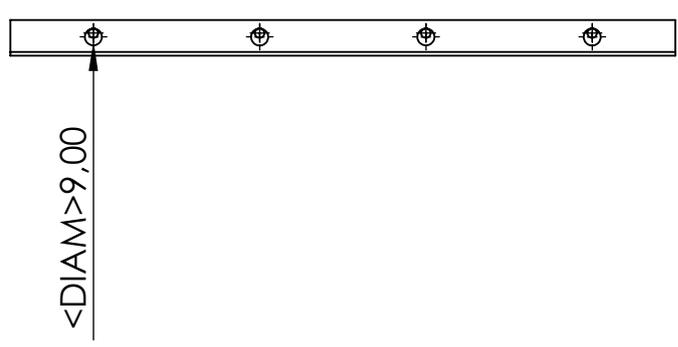
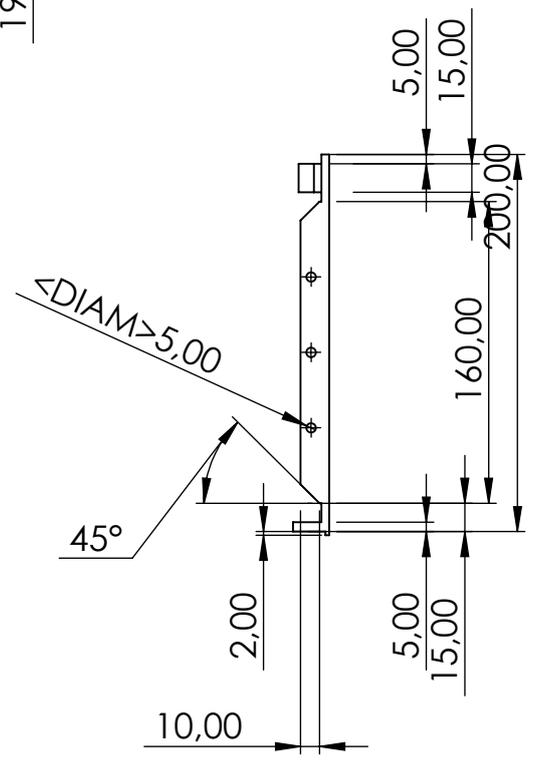
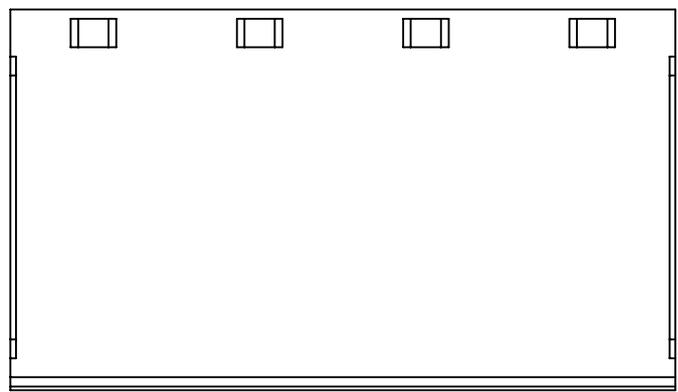
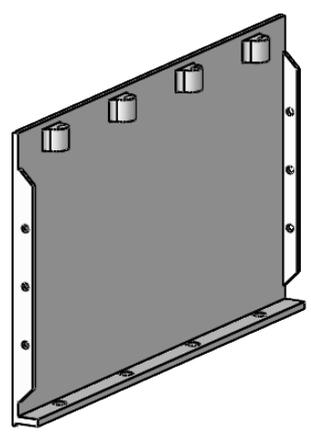
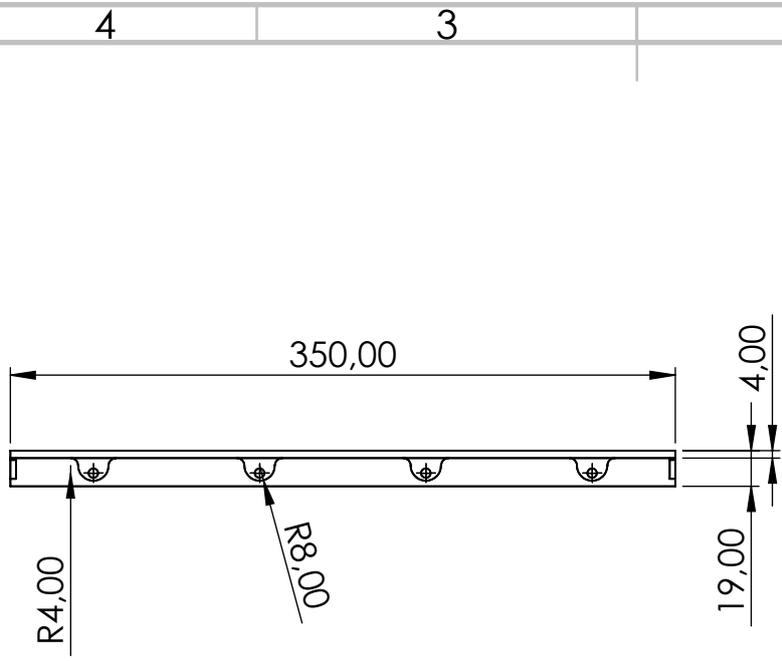


PESO:

SCALE:1:10

HOJA 1 DE 1





A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

ABS

DENOMINACIÓN:

Pieza lateral

NÚMERO DE DIBUJO:

4

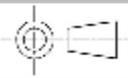
A4



PESO:

SCALE:1:5

HOJA 1 DE 1



4 3 2 1

F

F

E

E

D

D

C

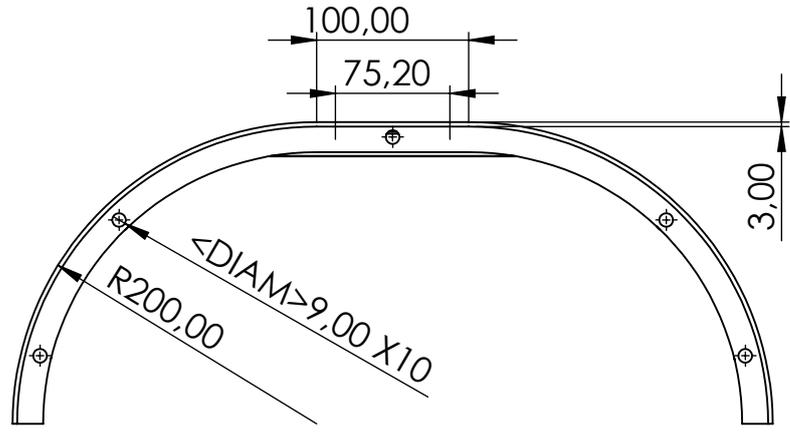
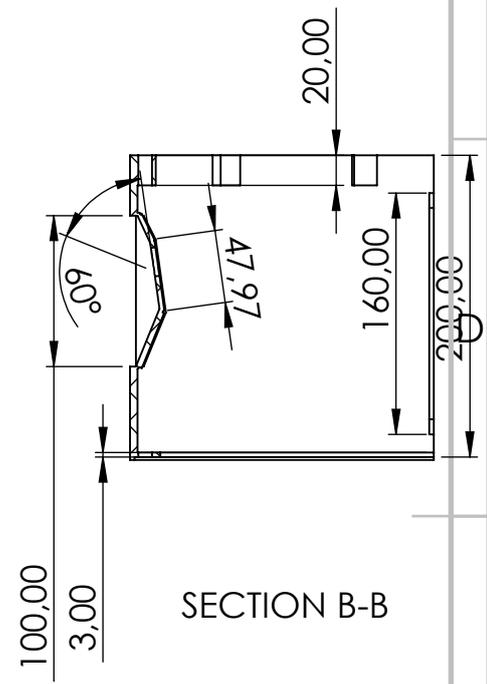
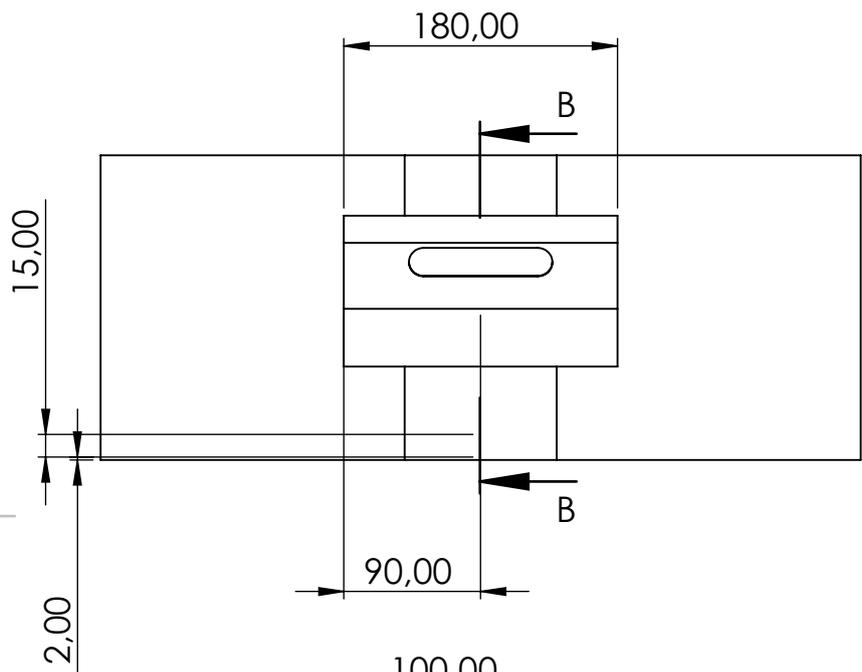
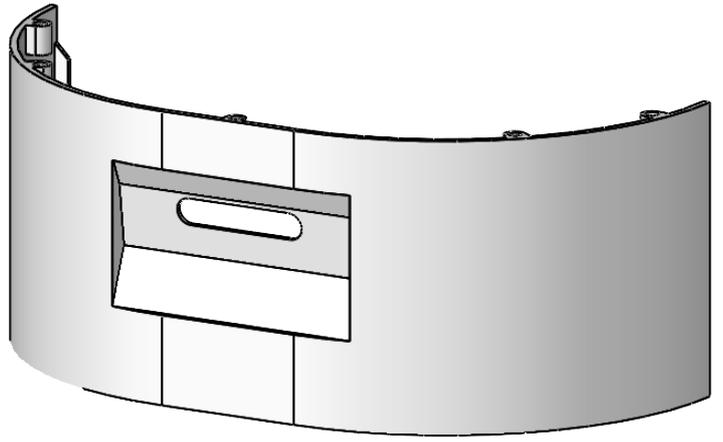
C

B

B

A

A



A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

ABS

DENOMINACIÓN:

Pieza frontal

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

NÚMERO DE DIBUJO:

5

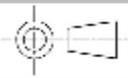
A4



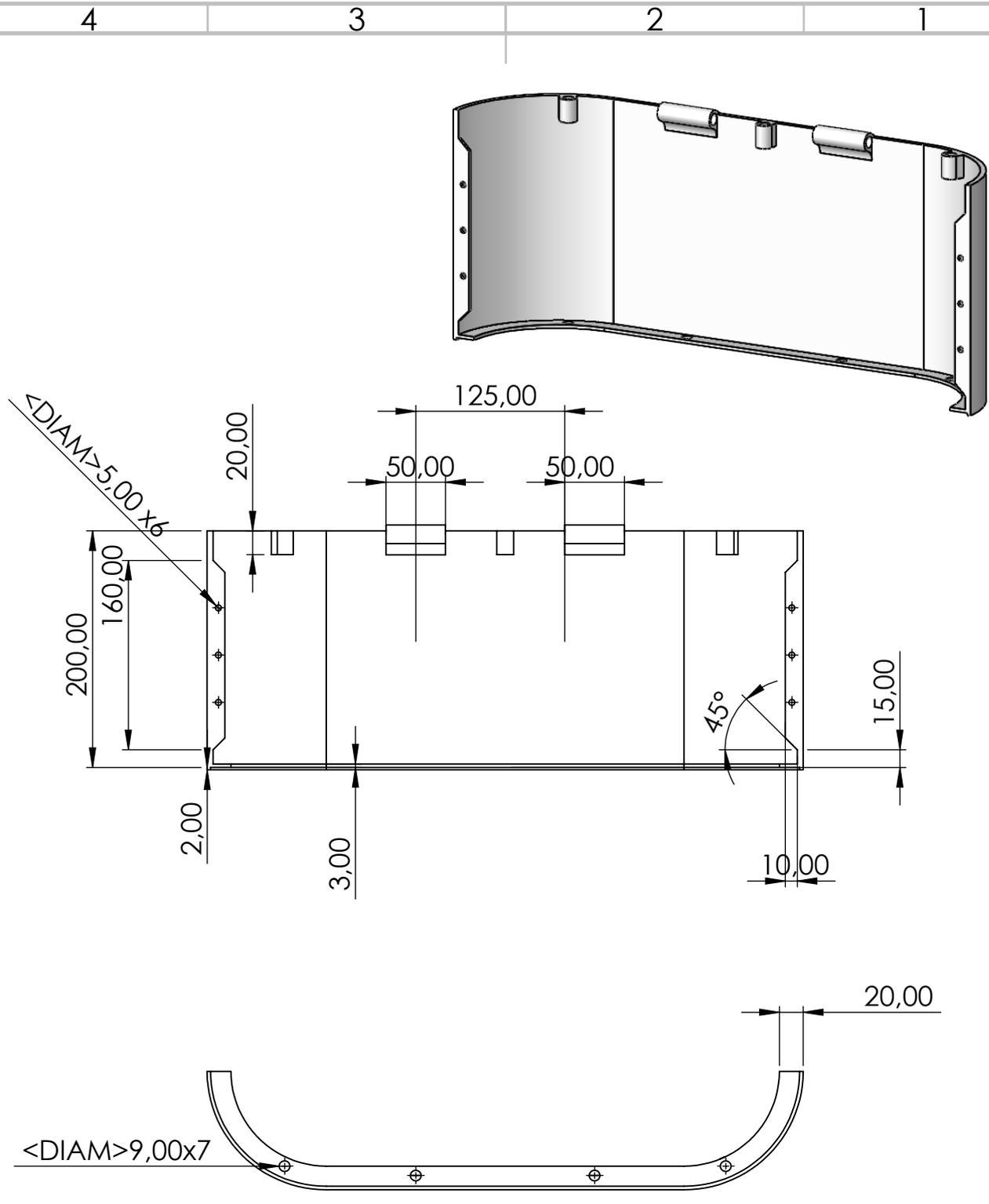
PESO:

SCALE:1:5

HOJA 1 DE 1



4 3 2 1



A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
 ACABADO DE LA SUPERFICIE:  
 TOLERANCIAS:  
 LINEALES:

REVISIÓN			

MATERIALES:  
**ABS**

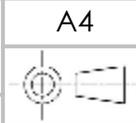
NOMBRE	MATERIA	FECHA
DIB. P. M. & A. A.	MI	12/09/22
REV.		
APROB.		

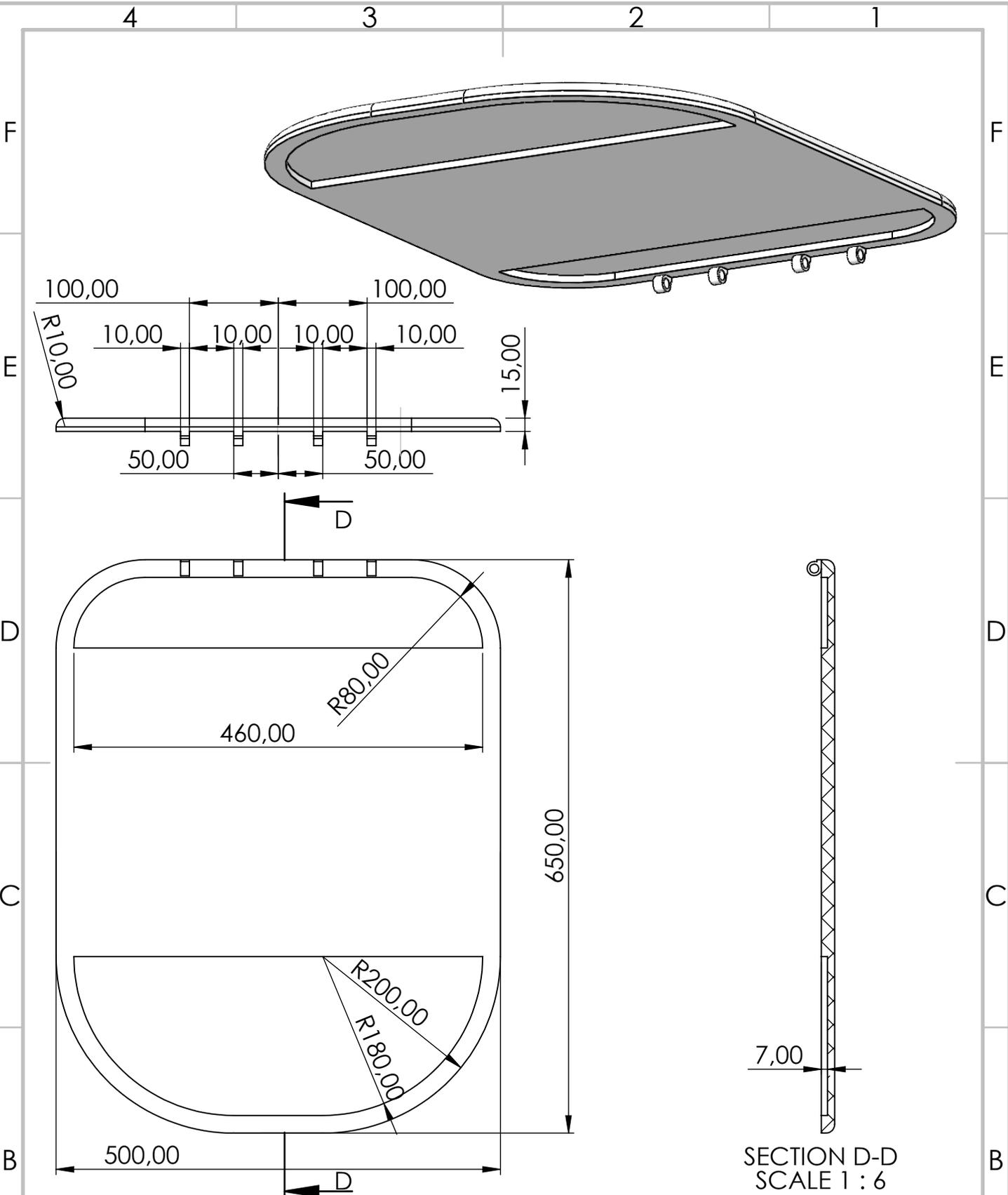
DENOMINACIÓN:  
**Pieza posterior**

**espol**

NÚMERO DE DIBUJO:  
**6**

PESO: SCALE:1:10 HOJA 1 DE 1





A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
 ACABADO DE LA SUPERFICIE:  
 TOLERANCIAS:  
 LINEALES:

REVISIÓN			
NOMBRE	MATERIA	FECHA	
DIB. P. M. & A. A.	MI	12/09/22	
REV.			
APROB.			

MATERIALES:  
**ABS**

DENOMINACIÓN:  
**Tapa**

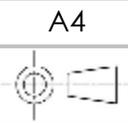
NÚMERO DE DIBUJO:  
**7**



PESO:

SCALE:1:5

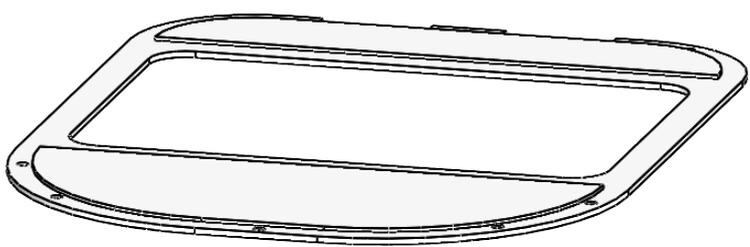
HOJA 1 DE 1



4 3 2 1

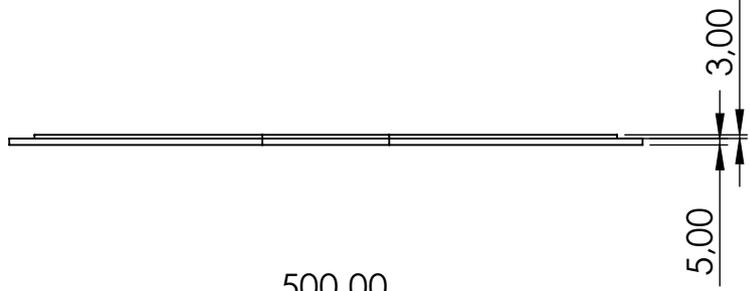
F

F



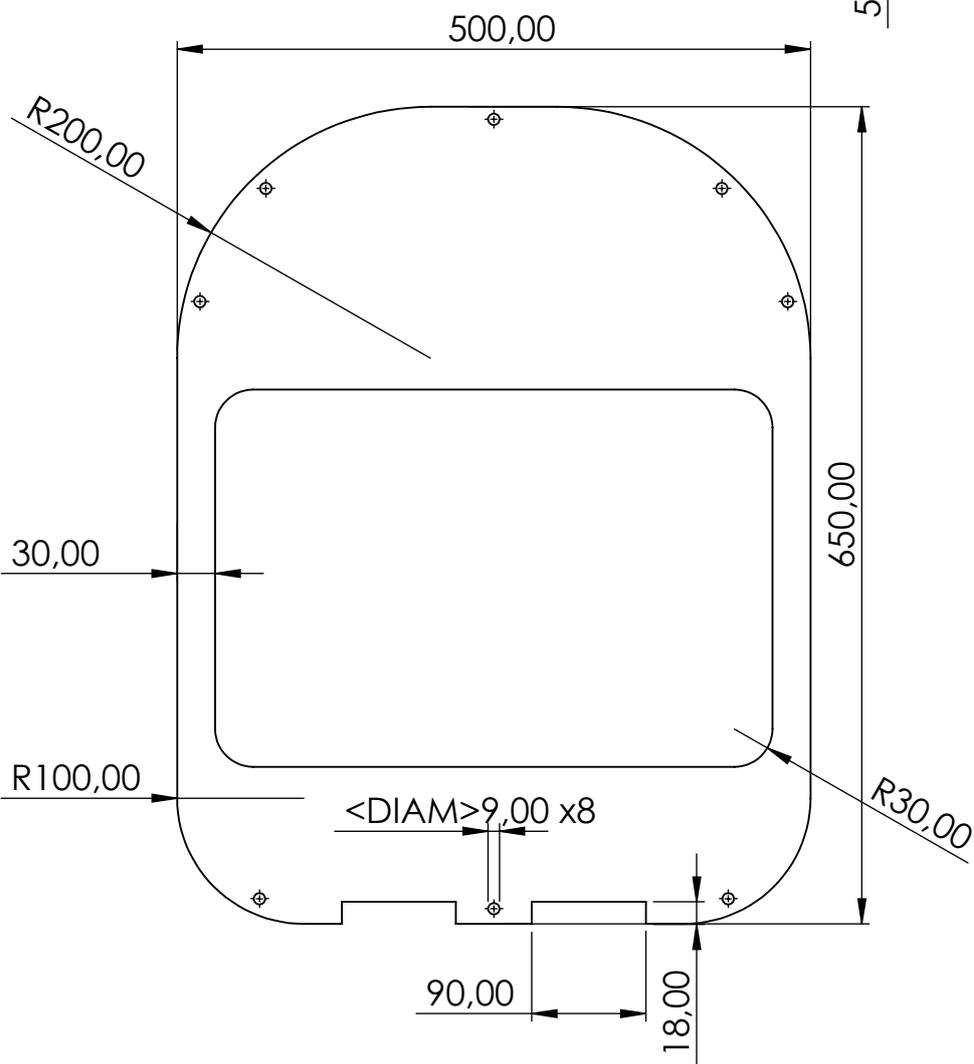
E

E



D

D



C

C

B

B

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

ABS

DENOMINACIÓN:

Base de la tapa

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

NÚMERO DE DIBUJO:

8

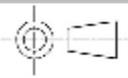
A4



PESO:

SCALE:1:5

HOJA 1 DE 1



4 3 2 1

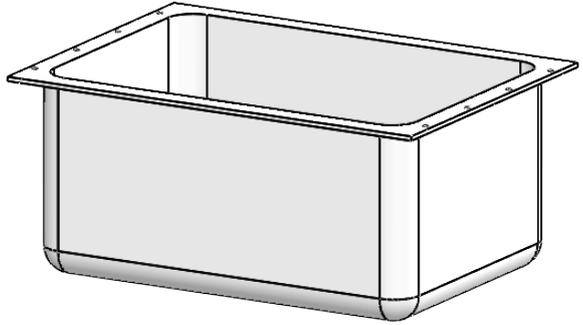
A

A

4 3 2 1

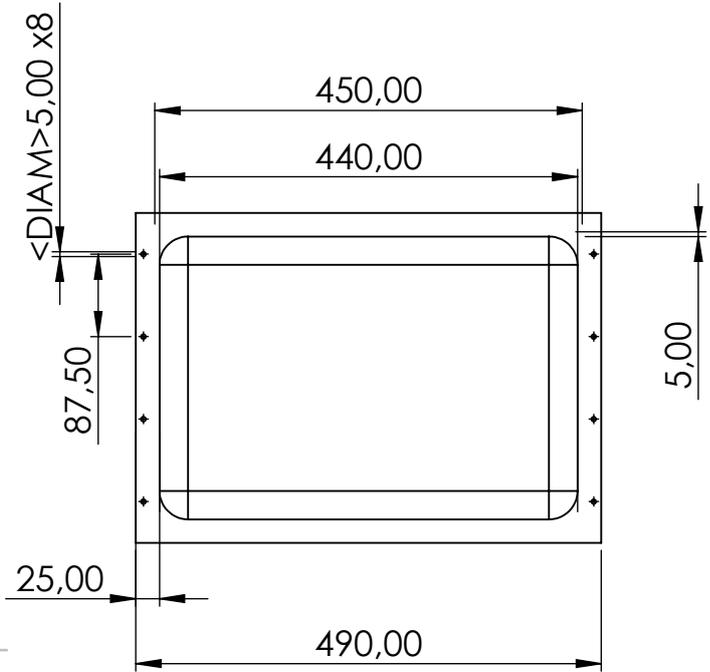
F

F



E

E

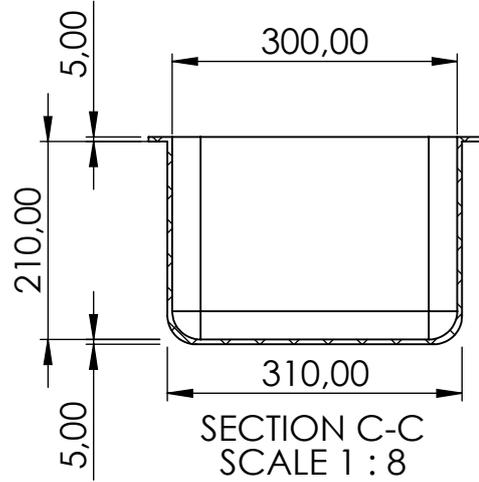
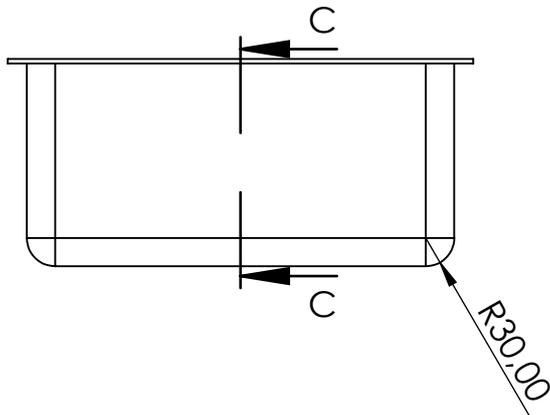


D

D

C

C



B

B

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

ABS

DENOMINACIÓN:

Compartimento

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

NÚMERO DE DIBUJO:

9

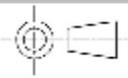
A4



PESO:

SCALE:1:10

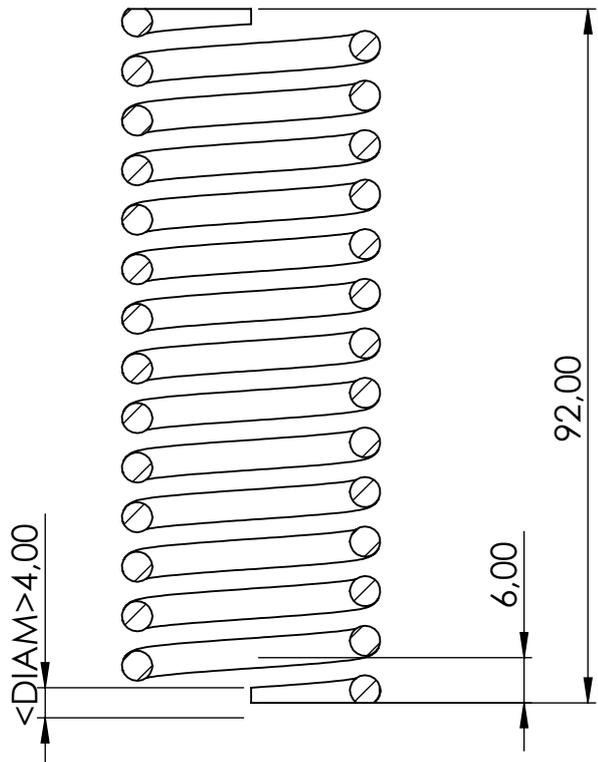
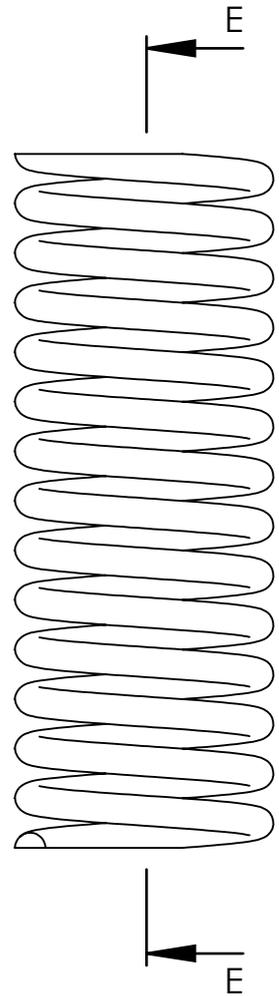
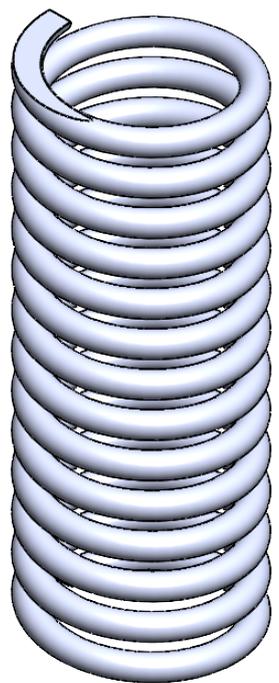
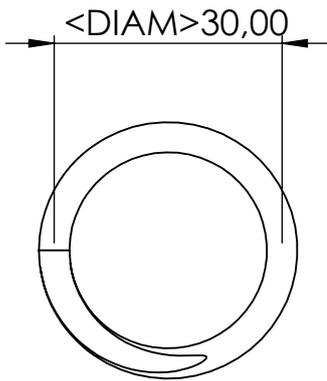
HOJA 1 DE 1



4 3 2 1

A

A



SECTION E-E

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Alambre de piano

DENOMINACIÓN:

Resorte

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

NÚMERO DE DIBUJO:

10

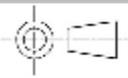
A4

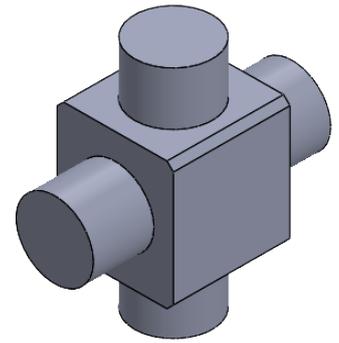
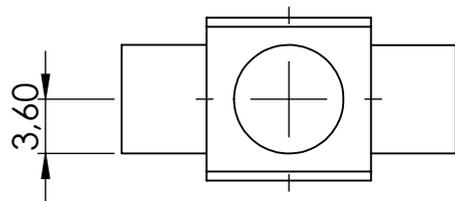
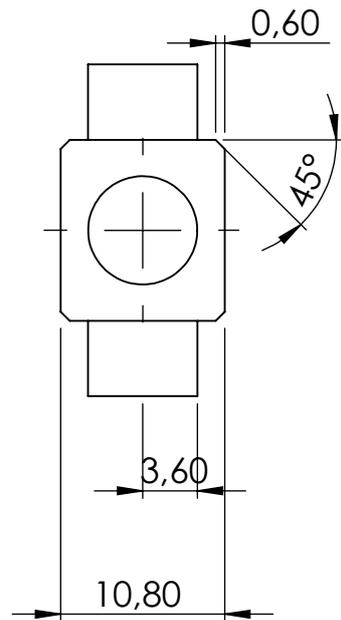
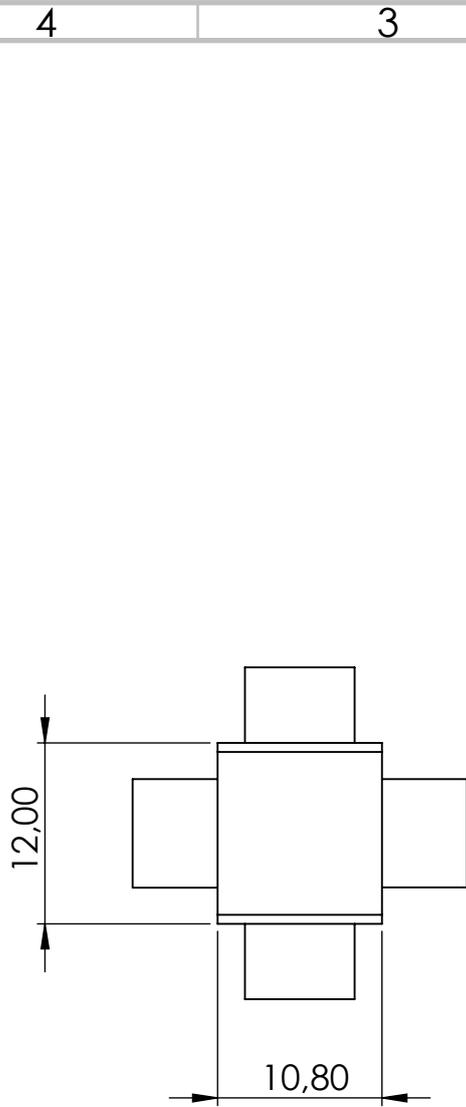


PESO:

SCALE:1:1

HOJA 1 DE 1





A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Acero AISI 303

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

DENOMINACIÓN:

Junta universal

NÚMERO DE DIBUJO:

11

A4

espol

PESO:

SCALE:2:1

HOJA 1 DE 1



4 3 2 1

F

F

E

E

D

D

C

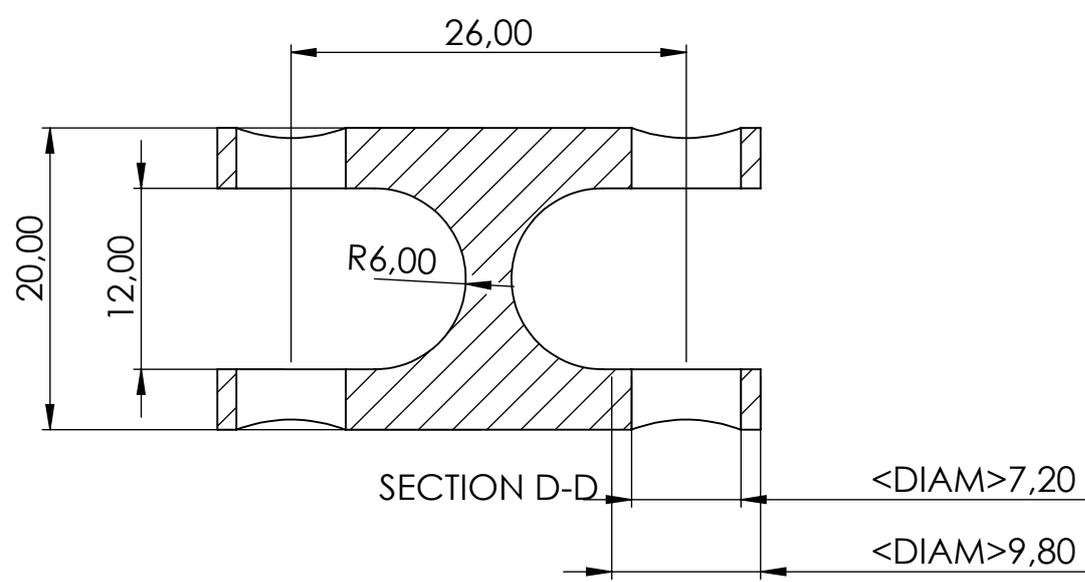
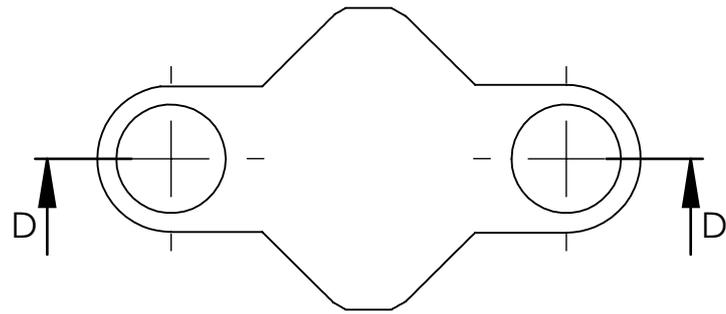
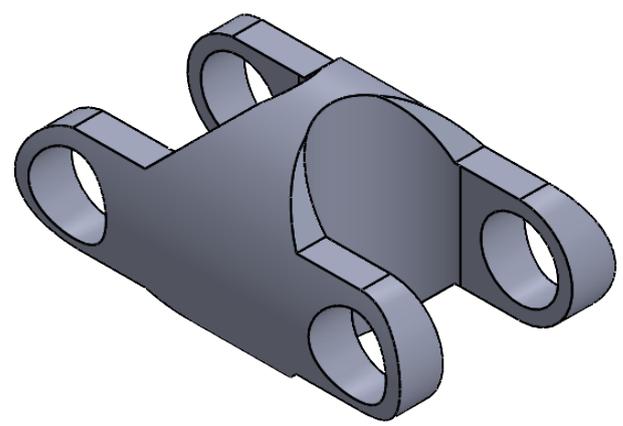
C

B

B

A

A



A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Acero AISI 416

DENOMINACIÓN:

Eje de transmisión

NÚMERO DE DIBUJO:

13

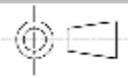
A4



PESO:

SCALE:2:1

HOJA 1 DE 1



4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

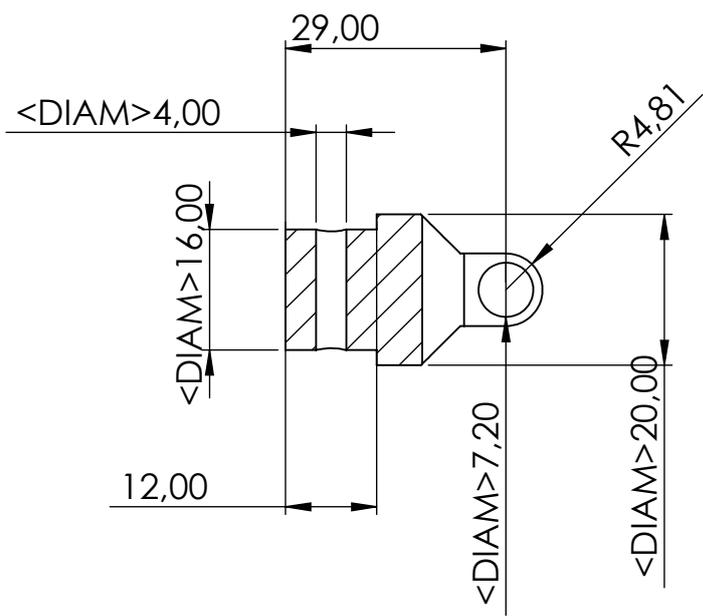
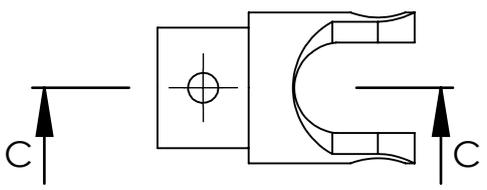
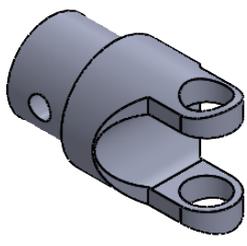
C

B

B

A

A



A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Acero AISI 416

DENOMINACIÓN:

Horquilla rueda

NÚMERO DE DIBUJO:

14

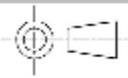
A4



PESO:

SCALE:1:2

HOJA 1 DE 1

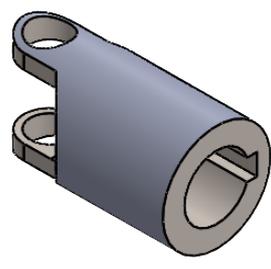


4 3 2 1

4 3 2 1

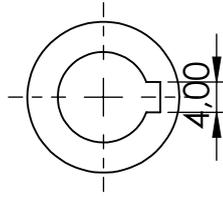
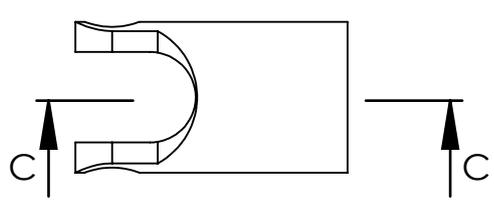
F

F



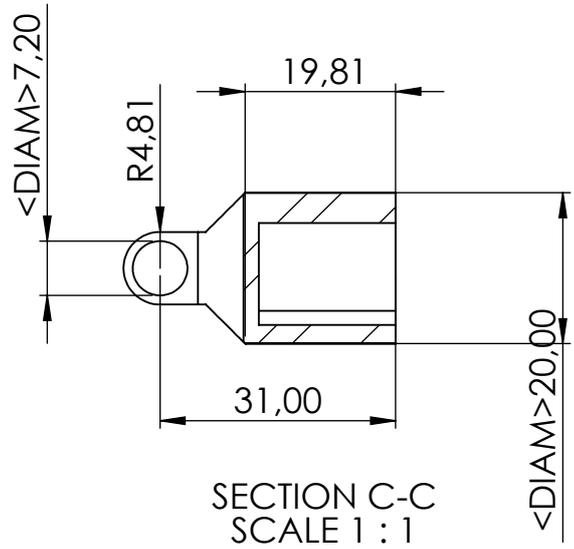
E

E



D

D



C

C

B

B

A MENOS QUE SE ESPECIFIQUE LO CONTRARIO: LAS DIMENSIONES ESTÁN EN MILÍMETROS  
ACABADO DE LA SUPERFICIE:  
TOLERANCIAS:  
LINEALES:

REVISIÓN

MATERIALES:

Acero AISI 416

	NOMBRE	MATERIA	FECHA
DIB.	P. M. & A. A.	MI	12/09/22
REV.			
APROB.			

DENOMINACIÓN:

Horquilla motor

A

A

NÚMERO DE DIBUJO:

15

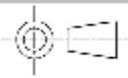
A4



PESO:

SCALE:1:2

HOJA 1 DE 1



4 3 2 1