

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Mecánica y Ciencias de la Producción

Desarrollo de un robot móvil con manejo diferencial a partir de motores hub

INGE-2600

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Mecatrónica

Presentado por:

Adrián Rafael Correa Mora

Johan Andrés Agurto Guevara

Guayaquil - Ecuador

Año: 2024

Dedicatoria

El presente proyecto se lo dedico a mis padres, Pedro Correa y Shirley Mora, quienes me apoyaron incondicionalmente para alcanzar cada una de mis metas, siendo el pilar fundamental de mi formación. A mi abuelita, Laura Aguilar, que siempre me apoyó en todos mis objetivos y se llena de orgullo por todo lo que he logrado.

Finalmente, se lo dedico a mi pareja, Samantha Guamán, y mis amigos que siempre estuvieron para apoyarme e hicieron este camino una experiencia inolvidable.

Adrián Correa

El presente proyecto va dedicado a mis padres, Johan Agurto y Yudi Guevara, de quienes siempre he recibido su apoyo incondicional y quienes formaron el hombre que soy hoy. A mi hermana, Andrea Agurto, que me brindó todo su apoyo en mis últimas etapas de la carrera y gracias a ella hoy he podido culminarla. A mi hermana, Danna Agurto, de quien tomo motivación para ser mejor cada día. A mi abuelito Luis Agurto, quien me apoyó en mis primeros pasos de mi formación profesional.

Por último, dedico este trabajo a todas las personas que me acompañaron en los momentos difíciles, que siempre confiaron en mí y de quienes siempre he recibido su apoyo, los llevo conmigo a todos lados y siempre estaré agradecido.

Johan Agurto.

Agradecimientos

Agradezco a mis padres y a toda mi familia, por todo el apoyo que me brindaron en el transcurso de mi vida y formación profesional. A mi pareja, por todo el apoyo incondicional y motivación para alcanzar esta meta.

A mi amigo y compañero de tesis Johan Agurto, a pesar de todo el trabajo y dificultades supo seguir adelante con toda la motivación, y poder construir nuestro robot BOTLER.

Al M.Sc. Ing. Ronald Solis, por su confianza en estos dos años de ser su ayudante académico

Agradezco al Ph.D. Ing. Dennys Paillacho, por darnos la oportunidad de trabajar con BOTLER. Agradezco al Ph.D. Ing. Carlos Saldarriaga por guiarnos en nuestro proyecto y concluirlo con éxito.

Adrián Correa

Agradezco a Dios por guiarme y darme sabiduría a lo largo de la carrera y el desarrollo de este trabajo.

A mis padres por mostrarme su apoyo en todo momento y siempre darme esas palabras de aliento para confiar en mis capacidades.

A mi amigo y compañero de tesis Adrián Correa, por nunca rendirse y hacer que construir un robot fuera algo sencillo.

Agradezco al Ph.D. Ing. Dennys Paillacho al Ph.D. Ing. Carlos Saldarriaga por confiarnos el proyecto y apoyarnos en darle vida a BOTLER.

Por último, me gustaría agradecer a aquellas personas que me acompañaron en el proceso de elaboración de este proyecto, con grandes ideas, opiniones o a veces simplemente con su compañía.

Johan Agurto.

Declaración Expresa

Nosotros **Johan Andrés Agurto Guevara** y **Adrián Rafael Correa Mora** acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, **22 de mayo** del 2024.

**Johan Andrés
Agurto Guevara**

**Adrián Rafael
Correa Mora**

Evaluadores

Carlos Xavier Saldarriaga Mero. PhD.

Profesor de Materia

Dennys Fabian Paillacho Chiliza PhD.

Tutor de proyecto

Resumen

El sector de servicios en Ecuador, que contribuye significativamente al PIB, enfrenta desafíos para incorporar innovaciones tecnológicas. Este proyecto presenta un robot de manejo diferencial diseñado para mejorar la eficiencia operativa en la industria de servicios, con un enfoque en el sector restauranero y catering. El robot utiliza motores HUB y tecnologías de mapeo en tiempo real, empleando ROS (Robot Operating System) y un sensor LiDAR para la navegación autónoma en entornos dinámicos.

El diseño del robot incluye una estructura mecánica óptima y un sistema de control basado en un modelo cinemático diferencial, gestionado por un microcontrolador P2 Edge, y un ESP32 para la comunicación con ROS. Se llevó a cabo un análisis mecánico de los materiales y componentes para asegurar la resistencia y la capacidad de transportar cargas de hasta 30 kg, distribuidas uniformemente en su estructura. Además, se evaluaron alternativas priorizando la carga computacional, eficiencia energética y costos de producción.

El robot ha demostrado operar eficientemente en entornos cerrados, superando obstáculos y adaptándose a cambios en tiempo real durante 4 horas. Un análisis de costos sugiere que el robot es una solución económica y viable, con un precio estimado de \$2500, competitivo en comparación con productos similares en el mercado.

Palabras clave: robot de servicio, ROS, mapeo, navegación autónoma

Abstract

The service sector in Ecuador, which contributes significantly to the PIB, faces challenges in incorporating technological innovations. This project presents a differential-drive robot designed to improve operational efficiency in the service industry, with a focus on the restaurant and catering sector. The robot utilizes HUB motors and real-time mapping technologies, employing ROS (Robot Operating System) and a LiDAR sensor for autonomous navigation in dynamic environments.

The robot's design includes an optimal mechanical structure and a control system based on a differential kinematic model, managed by a P2 Edge microcontroller and an ESP32 for communication with ROS. A mechanical analysis of materials and components was conducted to ensure durability and the ability to transport loads of up to 30 kg, evenly distributed across its structure. Additionally, alternatives were evaluated with a focus on computational load, energy efficiency, and production costs.

The robot has demonstrated efficient operation in indoor environments, overcoming obstacles and adapting to real-time changes for up to 4 hours. A cost analysis suggests that the robot is an economical and viable solution, with an estimated price of \$2500, competitive compared to similar products on the market.

Keywords: service robot, ROS, mapping, autonomous navigation

Índice General

Resumen.....	I
Abstract.....	II
Índice general.....	III
Abreviaturas.....	VII
Simbología.....	VIII
Índice de figuras.....	IX
Índice de tablas.....	XII
Índice de planos.....	XIII
Capítulo 1.....	1
1.1 Introducción.....	2
1.2 Descripción del problema.....	3
1.3 Justificación del problema.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5
1.5 Marco Teórico.....	5
1.5.1 Robot móvil diferencial.....	5
1.5.2 Robot autónomo.....	6
1.5.3 Motores HUB.....	7
1.5.4 ROS.....	8

1.5.4.1	Localización y mapeado para robots móviles	9
1.5.4.2	Planificación de trayectorias y navegación para robots móviles	10
1.5.5	Sensor LiDAR.....	10
1.5.6	Estado del arte.....	10
Capítulo 2	14
2.1	Metodología	14
2.1.1	Requerimientos del producto	15
2.1.2	Soluciones propuestas.....	15
2.1.3	Análisis y selección de alternativa de solución.....	16
2.2	Proceso de diseño.....	20
2.3	Diseño mecánico.....	22
2.3.1	Elementos del diseño mecánico	22
2.4	Diseño electrónico	24
2.4.1	Modulo P2 Edge (32 MB RAM)	24
2.4.2	Controlador universal de motor	25
2.4.3	Motor Hub con encoder	27
2.4.4	Microcontrolador ESP32	28
2.4.5	RPLiDar A1	28
2.4.6	Cámara de profundidad Intel RealSense D435i.....	29
2.4.7	Jetson Nano	30
2.4.9	Diagrama – Circuito Electrónico	30

2.5 Diseño computacional.....	31
2.5.1 Estructura de comunicación para el mapeo del entorno y la localización del robot ...	32
2.5.2 Estructura de comunicación para la navegación y la localización del robot	33
2.6 Diseño de control	34
2.6.1 Modelo cinemático – Robot móvil diferencial	35
2.6.2 Sistema de control.....	38
Capítulo 3.....	39
3.1 Resultados y análisis	40
3.1.1 Materiales.....	40
3.1.2 Análisis del diseño mecánico	43
3.1.2.1 Definición de cargas y restricciones	43
3.1.2.2 Generación de malla para análisis de elementos finitos	44
3.1.2.3 Análisis de elementos finitos	45
3.1.3 Control de velocidad	49
3.1.4 Implementación real.....	51
3.1.4.1 Armado de base y plataforma	51
3.1.4.2 Conexiones eléctricas	53
3.1.4.3 Ensamblaje final del robot.....	54
3.1.5 Funcionamiento del robot	54
3.1.5.1 Entorno de trabajo y mapeo.....	54
3.1.5.2 Planificación y navegación.....	56

3.1.6 Análisis de costos.....	59
Capítulo 4.....	61
4.1 Conclusiones y recomendaciones	62
4.1.1 Conclusiones.....	62
4.1.2 Recomendaciones	63
Referencias.....	64
Apéndices.....	67

Abreviaturas

INEC	Instituto Nacional de Estadísticas y Censos
DC	Corriente Directa
ROS	Robot Operating System
API	Interfaz de programación de aplicaciones
SLAM	Simultaneous Localization and Mapping
AMCL	Adaptive Monte Carlo Localizer
LiDAR	Light Detection and Ranging
TTL	Lógica transistor a transistor
PWM	Modulación por ancho de pulso
PID	Proporcional, integral y derivativo
MDF	Fibras de densidad media
CIDIS	Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales

Simbología

km/h	Kilómetro / hora
°C	Grados centígrados
kg	kilogramo
N	Newton
m/s ²	metro / segundo cuadrado
mm	milímetro
m/s	metro / segundo

Índice de figuras

Figura 1.1 Configuración del sistema con una sola rueda loca.....	6
Figura 1.2 Robot DINERBOT T10 de la empresa Keenon	7
Figura 1.3 Motor HUB 6.5” con Encoder	8
Figura 1.4 Robot BellaBot de Pudu Robotics	12
Figura 1.5 Robot KettyBot de Pudu Robotics	12
Figura 1.6 Keenon T8 de KEENON Robotics.....	13
Figura 2.1 Mapa de metodología de diseño	21
Figura 2.2 Vista explosionada del motor HUB	23
Figura 2.3 Vista explosionada del diseño.....	23
Figura 2.4 Modelo 3D del robot	24
Figura 2.5 P2 Edge Module with 32MB RAM.....	25
Figura 2.6 Universal Motor Driver P2 Add-on Board	26
Figura 2.7 Motor Hub con Encoder	27
Figura 2.8 ESP32 DevKit V1.....	28
Figura 2.9 Sensor RPLIDAR A1	29
Figura 2.10 Cámara de profundidad Intel RealSense D435i	29
Figura 2.11 Kit de desarrollo Jetson Nano.....	30
Figura 2.12 Diagrama de conexiones del prototipo final.....	31
Figura 2.13 Estructura de comunicación entre sensores y actuadores para mapeo, localización y movimiento	33
Figura 2.14 Estructura de comunicación entre sensores y actuadores para navegación y localización	34
Figura 2.14 Esquema del robot móvil diferencial con sus variables	35
Figura 2.15 Sistema de lazo cerrado para manejo de velocidades.....	38

Figura 3.1 Propiedades físicas de material MDF	41
Figura 3.2 Propiedades físicas de material Aluminio 6061	41
Figura 3.3 Propiedades físicas de material Acero ASTM A36	42
Figura 3.4 Propiedades físicas de material Acero inoxidable AISI 304	42
Figura 3.5 Primer escenario – Estructura en aceleración constante y carga en el nivel más alto.....	43
Figura 3.6 Segundo escenario – Estructura en aceleración constante y carga en el segundo nivel.....	44
Figura 3.7 Tercer escenario – Estructura en aceleración constante y cargas en el tercer nivel.....	44
Figura 3.8 Vista de malla para análisis de elementos finitos	45
Figura 3.9 Análisis de elementos finitos: Tensión de Von Mises – Primer Caso.....	46
Figura 3.10 Análisis de elementos finitos: Desplazamiento – Primer Caso	46
Figura 3.11 Análisis de elementos finitos: Tensión de Von Mises – Segundo Caso.....	47
Figura 3.12 Análisis de elementos finitos: Desplazamiento – Segundo Caso	48
Figura 3.13 Análisis de elementos finitos: Tensión de Von Mises – Tercer Caso.....	48
Figura 3.14 Análisis de elementos finitos: Desplazamiento – Tercer Caso	49
Figura 3.15 Control de velocidad – Primera etapa	50
Figura 3.16 Controlador PID con ajuste de parámetros.....	50
Figura 3.17 Control PID con respecto a cambio de velocidades	51
Figura 3.18 Montaje de motores hub y castor wheels en la base principal.	52
Figura 3.19 Armado de la plataforma de 3 pisos.	52
Figura 3.20 Conexiones electrónicas entre los motores HUB y el microcontrolador P2 Edge	53
Figura 3.21 Segunda base del robot con el sensor RPLIDAR A1	53
Figura 3.22 Construcción final del robot.....	54

Figura 3.23 Entorno de trabajo para pruebas del robot.....	55
Figura 3.24 Mapa extraído del entorno utilizando SLAM.....	55
Figura 3.25 Localización y mapa de costos local y global utilizando DWAPlanner y AMCL	57
Figura 2.26 Robot realizando la navegación transportando un paquete	58
Figura B.1 .. Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Primera Parte	69
Figura B.2.Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Segunda Parte	70
Figura B.3 ... Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Tercera Parte	71
Figura B.3 Programación ESP32 – Lenguaje: C++ – Primera Parte	72
Figura B.4 Programación ESP32 – Lenguaje: C++ – Segunda Parte	73
Figura B.5 Programación ESP32 – Lenguaje: C++ – Tercera Parte.....	74
Figura C.1 Estructura de comunicación – Nodos y Tópicos – Mapeo	75
Figura C.2 Estructura de comunicación – Nodos y Tópicos – Navegación	76
Figura C.3-Estructura de comunicación – Árbol de Transformadas de los Marcos de Referencia	77

Índice de tablas

Tabla 2.1 Requerimientos de la solución final.....	15
Tabla 2.2 Evaluación del peso específico de cada criterio.....	17
Tabla 2.3 Evaluación de la ponderación del criterio Mantenimiento	17
Tabla 2.4 Evaluación de la ponderación del criterio Costo de Producción.....	18
Tabla 2.5 Evaluación de la ponderación del criterio Carga Computacional	18
Tabla 2.6 Evaluación de la ponderación del criterio Eficiencia Energética.....	19
Tabla 2.7 Evaluación de la ponderación del criterio Adaptable.....	19
Tabla 2.8 Matriz de decisión.....	20
Tabla 2.10 Características P2 Edge de Parallax.....	25
Tabla 2.11 Características Controlador Universal de Parallax.....	26
Tabla 2.12 Características del Motor Hub con Encoder.....	27
Tabla 3.1 Tabla de los materiales para la estructura completa del robot (No incluye motores o ruedas).....	40
Tabla 3.2 Tabla de elementos y mano de obra	59

Índice de planos

Plano 1. Cuerpo del robot – Lista de elementos

Plano 2. Cuerpo del robot – Vista explosionada

Plano 3. Base inferior del robot – ruedas

Plano 4. Base media del robot – sensor RPLIDAR A1

Plano 5. Plataforma grande del cuerpo del robot

Plano 6. Plataforma mediana y pequeña del cuerpo del robot

Plano 7. Soporte para el eje central del robot - escuadra

Capítulo 1

1.1 Introducción

Los avances tecnológicos han generado un gran cambio en todo el mundo, durante los últimos años ha existido un gran avance en nuevas tecnologías como la inteligencia artificial y la robótica, a pesar de estos beneficios, países en vías de desarrollo como el Ecuador aún se encuentran en el proceso de adoptar las nuevas tecnologías en los sectores productivos y de servicio, esto provoca que el crecimiento productivo se desarrolle a un ritmo lento por la brecha entre las empresas ecuatorianas y el potencial de las nuevas tecnologías.

Si nos enfocamos en la industria restaurantera, existen muchos desafíos en base a la eficiencia operativa, satisfacción al cliente e innovación para diferenciarse de otras empresas; los robots de servicio permitirían resolver estos desafíos, las tareas repetitivas serían realizadas por los robots, permitiendo al personal de servicio enfocarse en otras áreas como la interacción con el cliente, incorporar estos robots sería un factor clave para la diferenciación de la competencia, atrayendo a varios clientes que estén interesado en interactuar con estas nuevas tecnologías.

A pesar de los múltiples beneficios que trae consigo la implementación, existen temores sobre el reemplazo de las personas por robots que realicen sus tareas, pero de acuerdo con Alexandra Caro, propietaria de Kibou Kaiten Sushi (primer restaurante ecuatoriano en poseer un robot mesero), los robots no reemplazan a los seres humanos, son un apoyo para realizar las tareas de una manera más eficiente, y al ser algo llamativo provoca un mayor número de clientes y a su vez genera más plazas de trabajo. [1]

En el presente trabajo, se abordarán tópicos relacionados al impacto de la innovación tecnológica en el sector comercial restaurantera, se mencionará todos los conceptos que giran en torno a los robots móviles autónomos, eventualmente se analizarán las soluciones que se encuentran actualmente en el mercado para establecer un matriz de decisión que facilitará la selección de la solución. Finalmente, se diseñará e implementará un robot de servicio autónomo

orientando su uso y aplicación en restaurantes utilizando motores hub, programándolo con ROS y utilizando un LiDAR para el mapeo.

1.2 Descripción Del Problema

La industria dedicada al servicio y la atención al cliente en Ecuador afronta frecuentemente dificultades para aprovechar los avances tecnológicos y hacer uso de estos para mejorar y/o agilizar sus actividades, con nuevas tecnologías apareciendo cada día, muchas actividades comerciales relacionadas tienden a desaprovechar las nuevas oportunidades que surgen para realizar sus operaciones diarias de manera más eficiente y atrayendo más la atención del cliente.

Las actividades comerciales como restaurantes, supermercados y hoteles son los principales afectados por esta problemática, mismos cuya mano de obra representa un alto costo, del mismo modo, contratar y capacitar nuevo personal requiere de tiempo y dinero, sin mencionar las dificultades que se presentan durante el proceso.

Los factores que impiden el uso de nuevas tecnologías son la falta de inversión de empresas pequeñas y medianas para gestionar la compra y capacitación para el uso de robots de servicio; además de no tener un análisis preciso sobre los beneficios potenciales de su implementación. Dando como resultado que muchas tareas repetitivas en el sector de servicio sigan realizándose de manera manual y no exista una diferencia con la competencia.

En esencia, la falta de uso de nuevas tecnologías, el alto costo de mano de obra y la realización de tareas repetitivas son las principales problemáticas por resolver para que el sector de servicio y atención al cliente en Ecuador se presente una innovación tecnológica, mejorando la eficiencia en las tareas a realizar.

1.3 Justificación del Problema

La industria de servicio y atención al cliente en Ecuador, incluyendo restaurantes, supermercados y hoteles, enfrenta una gran problemática por no aprovechar los avances tecnológicos. Esto impide la modernización y eficiencia operativa, afectando negativamente la competitividad y la capacidad de atraer y retener clientes. De acuerdo con los datos obtenidos por el Instituto Nacional de Estadísticas y Censos “INEC”, el sector de alojamiento y servicio de comida ha generado en 2022, un total de ventas en miles de dólares de 2.564.883,79, este valor fue generado por 74.549 empresas. Estos datos resaltan la importancia del sector terciario en la economía ecuatoriana, la falta de innovación y uso de nuevas tecnologías para brindar un servicio que sea llamativa a los clientes es uno de los factores para un mayor crecimiento económico del país. [2]

Las empresas ecuatorianas que no adoptan herramientas tecnológicas enfrentan una desventaja competitiva respecto a las que sí las usan. La tecnología permite diferenciarse, atraer clientes y mejorar la eficiencia operativa y la calidad del servicio. Sistemas automatizados reducen tiempos de espera, aumentan la precisión y la satisfacción del cliente. Sin inversión tecnológica, persisten métodos manuales ineficientes y propensos a errores, resultando en servicios homogéneos que no destacan, limitando la atracción y retención de clientes, lo que impacta negativamente en la economía ecuatoriana. [3]

La razón por la cual se debe abordar este problema radica en la necesidad de modernizar el sector de servicio y atención al cliente en Ecuador mediante el uso de robots de servicio que realicen tareas repetitivas. La adopción de tecnologías avanzadas no solo aumentará la eficiencia y los costos operativos, sino que también aumentará la competitividad y la capacidad de atraer y retener clientes en un mercado cada vez más globalizado y exigente.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar un robot mesero con manejo diferencial utilizando motores HUB y un sistema de mapeo en tiempo real, para desplazarse de manera autónoma y eficiente en entornos dinámicos.

1.4.2 Objetivos específicos

1. Diseñar e implementar la estructura interna y externa para la integración de dispositivos, sensores y actuadores del robot mesero.
2. Implementar un sistema de navegación autónoma utilizando algoritmos de control y mapeo del entorno para el manejo diferencial.
3. Implementar un sistema de comunicación inalámbrica entre el robot mesero y el usuario para controlar el movimiento del robot a tiempo real.

1.5 Marco Teórico

A lo largo de esta tesis se manejarán conceptos y terminologías muy relacionadas a los robots móviles, es por ello que el presente marco teórico se estructura con información necesaria para comprender de mejor manera el contenido de este trabajo. Se iniciará mencionando las características de un robot móvil diferencial y un robot móvil autónomo y eventualmente se mencionará la herramienta de software que se utilizará para su programación, así como una descripción del sistema de mapeo que se utilizará.

1.5.1 Robot Móvil Diferencial

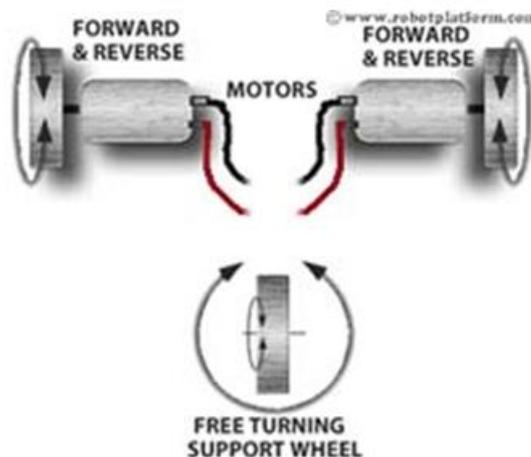
Un robot móvil diferencial tiene un sistema de transmisión de tracción dos ruedas controladas cada uno por su propio motor, el movimiento se robot es a base de la configuración de las velocidades de cada rueda en un mismo eje, la base del robot se puede tanto una como

dos ruedas locas o castor wheel para obtener una mayor estabilidad en el movimiento del robot.

[4]

Figura 1.1

Configuración del sistema con una sola rueda loca.



Nota. Gráfica de la configuración obtenida de Robot Platform [4]

Las 4 configuraciones de velocidad y dirección para un robot diferencial son:

- Misma velocidad y dirección: avance en línea recta (adelante o atrás).
- Misma dirección y diferentes velocidades: giro a la izquierda o derecha dependiendo de cuál motor tenga mayor velocidad.
- Misma velocidad y diferente dirección: gira sobre su propio eje (horario o antihorario).

1.5.2 Robot Autónomo

Un robot autónomo es aquél capaz de ejecutar actividades y tareas sin la intervención del ser humano, estos robots tienen la capacidad de detectar obstáculos o colisiones para tomar decisiones operativas para evitar riesgos, tener rutas establecidas para la realización de sus actividades y tomar ejecutar tareas durante tiempos prolongados [5]. Una categoría de los robots autónomos son los robots dedicados al servicio y atención al cliente, su objetivo es generar una

mayor eficiencia al momento de realizar tareas, gracias al análisis de su entorno y toma de decisiones en tiempo real.

Las características principales de un robot autónomo son:

- Capacidad de navegar en un entorno de manera autónoma y mediante la recopilación de información tomar decisiones para la evasión de obstáculos.
- Utilización de sensores para analizar el entorno y generar un mapeo de la zona.
- Planificación de rutas seguras mediante el control de los motores para generar movimientos fluidos y coordinados.

Figura 1.2

Robot DINERBOT T10 de la empresa Keenon

Nota. Gráfica obtenida de Keenon [6]

1.5.3 Motores HUB

Un motor HUB, conocido como motor en rueda, es un tipo de motor eléctrico de imán permanente integrado directamente en la rueda del vehículo, eliminando la necesidad de un sistema para transmitir la potencia del motor hacia la rueda [7]. El motor HUB se alimenta con corriente directa (DC) y se caracteriza por tener tres cables de alimentación y, según el modelo, líneas para los sensores hall que retroalimentan la posición del rotor.

Los tres cables de alimentación llevan la corriente hacia el motor permitiéndole que gire, la naturaleza de esta alimentación consiste en conmutar las señales con los tres cables para generar una secuencia de conmutación que gira el rotor, esta secuencia utiliza los sensores hall o la fuerza electromotriz de retorno para ajustar la corriente y alcanzar la velocidad deseada [7].

La principal ventaja de los motores HUB es la mejora en la eficiencia al momento de no necesitar un sistema de transmisión de potencia, del mismo modo, es más silencioso y el

mantenimiento es reducido. Por otro lado, entre sus desventajas están la sensibilidad a altas temperaturas, la dificultad para disipar el calor y el alto costo [8].

Figura 1.3

Motor HUB 6.5'' con Encoder



Nota. Imagen extraída de Parallax.com [9]

1.5.4 ROS

ROS (Robot Operating System) es un framework para el desarrollo de software robótico, compuesto por herramientas y bibliotecas que simplifican la creación y programación de comportamientos complejos en robots. En su núcleo, ROS actúa como middleware, proporcionando la infraestructura necesaria para la comunicación eficiente entre sensores, actuadores y aplicaciones de software [10].

Las aplicaciones en ROS consisten en múltiples procesos independientes que se conectan entre sí como nodos en un grafo. Estos nodos se comunican a través de canales estrictamente tipificados, evitando la infiltración de datos incorrectos. Esta estructura no solo simplifica el desarrollo de aplicaciones mediante el uso de una API en cada nodo, sino que también aumenta la robustez del sistema al aislar fallos potenciales. La gestión del grafo, incluyendo la negociación de conexiones y el registro y búsqueda de recursos, es manejada por el programa ROS Master. Además, un Parameter Server se encarga de los parámetros persistentes accesibles por todos los nodos [10].

ROS ofrece varios mecanismos de comunicación:

- **Paso de mensajes:** Utiliza un sistema anónimo de Publicador/Subscriber a través de buses llamados ROS topics, permitiendo comunicación asíncrona entre el nodo emisor y los nodos receptores.
- **Llamada remota de procedimientos:** Emplea un modelo Cliente/Servidor que facilita la comunicación síncrona entre el nodo cliente que realiza la llamada y el nodo servidor que responde.
- **Acciones:** Permite una comunicación Cliente/Servidor con retroalimentación, posibilitando la monitorización del progreso y la opción de interrumpir la llamada.

Además, ROS cuenta con herramientas avanzadas para el desarrollo de aplicaciones robóticas complejas, incluyendo la localización y mapeo de robots móviles, la planificación de trayectorias y navegación de robots móviles, y la planificación de movimientos y trayectorias de robots manipuladores [10].

1.5.4.1 Localización y mapeado para robots móviles

La localización corresponde al cálculo de la ubicación del robot utilizando principalmente sensores, esta información indica la posición y orientación del robot tomando como referencia un punto de origen.

El mapeo corresponde a la generación del mapa del entorno como una representación en 2D o 3D utilizando diferentes sensores. Los recursos utilizados para el mapeo aportan en la calidad y resolución de este [10].

ROS utiliza algoritmos estadísticos para la localización y mapeo de robots móviles, basados en datos de sensores. Estos algoritmos pueden tener errores por datos defectuosos, causando distorsiones en el mapa. El algoritmo SLAM corrige estos problemas, permitiendo al robot ajustar su posición y las distorsiones del mapa en entornos desconocidos [10].

1.5.4.2 Planificación de trayectorias y navegación para robots móviles

La Planificación de Trayectorias es el proceso mediante el cual el robot determina el camino óptimo para desplazarse del punto de inicio al punto objetivo evitando obstáculos, respetando las restricciones del entorno y considerando las propiedades físicas del robot [10].

La planificación cuenta con los siguientes componentes:

- Mapas de costes: Utilizando el mapa generado por el módulo SLAM, ROS lo convierte en un mapa de coste, que sería un mapa con toda la información acerca de los lugares por los que el robot puede desplazarse sin colisionar.
- Planificador global: Genera un camino para la navegación global sobre el espacio disponible sin obstáculos tomando en cuenta restricciones cinemáticas del robot. ROS implementa algoritmos como A*, Dijkstra, Voronoi o SBPL.
- Planificador local: Este planificador responde en caso de la presencia de obstáculos que no fueron detectados previamente, el resultado converge con la navegación principal lo que permite al robot evadir obstáculos en entornos dinámicos. Esta planificación se realiza a tiempo real utilizando sensores adecuados, por ejemplos, un sensor LiDAR.

1.5.5 Sensor LiDAR

Un sensor LiDAR (Light Detection and Ranging) es un sistema de medición y detección de obstáculos utilizando rayos láser infrarrojos. El mecanismo para obtener las distancias a las cuales se encuentran los objetos se realiza a partir de una medición del tiempo que tarda el pulso láser en regresar [10].

1.5.6 Estado Del Arte

Pudu Robotics es una empresa china líder en el mercado global de robots de servicios comerciales. Utiliza robots autónomos para mejorar la eficiencia en negocios enfocados en

servicio y atención al cliente. Sus productos cubren áreas como Smart Catering, Smart Retail y Fabricación Inteligente.

La empresa Pudu Robotics ofrece soluciones robóticas tanto para el sector de la restauración (Smart Catering) como para el comercio minorista (Smart Retail), mejorando la eficiencia operativa y reduciendo costos en ambos sectores. En Smart Catering, se mejora la eficiencia de 2 a 5 veces y reduciendo costos en más del 50%, esto permite una respuesta rápida a las necesidades del cliente, distribución flexible de mano de obra, experiencia de usuario más estable y gestión digital de pedidos y entregas [11]. En Smart Retail, se mejora la competitividad y eficiencia al ofrecer una cobertura multiescenarios y multifuncional, actuando como spots publicitarios interactivos y para guías o entregas.

Los robots KettyBot y BellaBot son ideales para servicios de catering y retail. KettyBot mide 435 x 450 x 1120 mm, tiene una velocidad máxima de 1.2 m/s y una capacidad de carga de 30 kg, con un display de 18.5". BellaBot mide 565 x 537 x 1290 mm, con una carga máxima de 40 kg, una batería de 12-24H y velocidad ajustable de 0.5 m/s a 1.2 m/s [12].

Figura 1.4

Robot BellaBot de Pudu Robotics



Nota. Imagen extraída del sitio oficial de Pudu Robotics [1]

Figura 1.5

Robot KettyBot de Pudu Robotics



Nota. Imagen extraída del sitio oficial de Pudu Robotics [11]

El robot Keenon T8, diseñado por KEENON Robotics, mide 384 x 462 x 1096 mm y pesa 38 kg. Es un robot de entrega autónoma ideal para restaurantes, hoteles y centros comerciales. Tiene 3 repisas con capacidad de carga: 5 kg en las superiores y 10 kg en la inferior. [13]

Figura 1.6

Keenon T8 de KEENON Robotics



Nota. Gráfica obtenida del manual de usuario de KEENON Robotics [13]

El robot tiene 5 modos de escenario, configurables mediante su pantalla táctil. Utiliza un sensor LiDAR de 204° para mapear el entorno, enviando la información por Wi-Fi al controlador para definir las posiciones deseadas. Esto le permite planificar la mejor ruta para entregas rápidas. Además, cuenta con 2 sensores de visión binocular de 120° para detección dinámica de obstáculos en tiempo real, un encoder y un sensor de colisión, permitiéndole esquivar obstáculos y pasar por espacios de al menos 55 cm.

Capítulo 2

2.1 Metodología

2.1.1 Requerimientos del producto

Para establecer las alternativas de solución para nuestra problemática es necesario establecer los requerimientos de nuestro producto final, por lo tanto, se planteó los siguientes requerimientos generales:

Tabla 2.1

Requerimientos de la solución final

Requerimientos	
1.	El robot se debe configurar una sola vez para realizar el mapeo del entorno de trabajo.
2.	El robot debe ser capaz de trabajar de manera autónoma alrededor de una jornada de 4 horas.
3.	El robot debe detectar y evadir obstáculos en tiempo real.
4.	Los motores del robot deben soportar el peso de la estructura y de los productos que traslada.
5.	El robot debe ser fácil de mantener y reparar en el caso de un fallo.
6.	El robot se debe desplazar a una velocidad óptima para transportar los alimentos sin ningún inconveniente en el trayecto.

2.1.2 Soluciones propuestas

Mediante el análisis de los requerimientos se obtuvieron 3 ideas de diseño para las alternativas de solución de nuestra problemática:

- **Solución A:** Diseño de un robot móvil de servicio con manejo diferencial capaz de realizar planeación y mapeo utilizando algoritmo SLAM en conjunto con el

controlador NVIDIA, y localización basada en odometría láser para ejecutar un sistema de navegación autónoma y evasión de obstáculos en tiempo real.

- **Solución B:** Diseño de un robot móvil de servicio con manejo diferencial capaz de realizar planeación y mapeo basado en características en conjunto con el controlador NVIDIA, y localización basada en odometría visual para ejecutar un sistema de navegación autónoma y evasión de obstáculos en tiempo real.
- **Solución C:** Diseño de un robot móvil de servicio con manejo diferencial capaz de realizar planeación y mapeo utilizando algoritmo SLAM en conjunto con el controlador Raspberry Pi, y localización basada en odometría láser para ejecutar un sistema de navegación autónoma y evasión de obstáculos en tiempo real.

2.1.3 Análisis y selección de alternativa de solución

En base a las soluciones propuestas, se estableció un conjunto de criterios para realizar un análisis y seleccionar la mejor opción. Los criterios son los siguientes:

- **Mantenimiento:** el producto debe ser fácil de realizar un mantenimiento periódico para evitar fallos.
- **Costo de producción:** la solución debe costar menos que los productos actuales en el mercado de robots de servicio.
- **Carga computacional:** capacidad del controlador para ejecutar el conjunto de tareas del robot.
- **Eficiencia energética:** permitir el mayor rendimiento y funcionalidad del robot con el menor consumo de energía posible.
- **Adaptable:** se debe poder configurar el robot para adaptarlo a las necesidades del usuario.

Se definió la importancia de cada criterio mediante el análisis de la siguiente tabla:

Tabla 2.2*Evaluación del peso específico de cada criterio*

Criterio	Mantenimiento	Costo de producción	Carga computacional	Eficiencia energética	Adaptable	$\Sigma+1$	Ponderación
Mantenimiento		0.5	0	0	0.5	2	0.13
Costo de producción	0.5		0	0.5	1	3	0.20
Carga computacional	1	1		0.5	1	4.5	0.28
Eficiencia energética	1	0.5	0.5		1	4	0.27
Adaptable	0.5	0	0	0		1.5	0.1
					Suma	15	1

Gracias a la ponderación se puede establecer el orden jerárquico de los criterios, siendo este:

Carga Computacional > Eficiencia Energética > Costo de Producción > Mantenimiento > Adaptable

Para la selección de la alternativa de solución, se evaluó cada criterio específico con respecto a las 3 alternativas de solución propuesta.

Tabla 2.3*Evaluación de la ponderación del criterio Mantenimiento*

Mantenimiento	<i>Solución 1</i>	<i>Solución 2</i>	<i>Solución 3</i>	$\Sigma+1$	Ponderación
<i>Solución 1</i>		1	1	3	0.5
<i>Solución 2</i>	0		0.5	1.5	0.25

<i>Solución 3</i>	0	0.5		1.5	0.25
			<i>Suma</i>	6	1

Dando como resultado: Solución 1 > Solución 2 = Solución 3

Tabla 2.4

Evaluación de la ponderación del criterio Costo de Producción

Costo de Producción	Solución 1	Solución 2	Solución 3	$\Sigma+1$	Ponderación
Solución 1		1	0.5	2.5	0.42
Solución 2	0		0	1	0.16
Solución 3	0.5	1		2.5	0.42
			<i>Suma</i>	6	1

Dando como resultado: Solución 1 = Solución 3 > Solución 3

Tabla 2.5

Evaluación de la ponderación del criterio Carga Computacional

Carga Computacional	Solución 1	Solución 2	Solución 3	$\Sigma+1$	Ponderación
Solución 1		0.5	1	2.5	0.42
Solución 2	0.5		1	2.5	0.42
Solución 3	0	0		1	0.16
			<i>Suma</i>	6	1

Dando como resultado: Solución 1 = Solución 2 > Solución 3

Tabla 2.6*Evaluación de la ponderación del criterio Eficiencia Energética*

Eficiencia Energética	Solución 1	Solución 2	Solución 3	$\Sigma+1$	Ponderación
Solución 1		1	1	3	0.5
Solución 2	0		0	1	0.17
Solución 3	0	1		2	0.33
			Suma	6	1

Dando como resultado: Solución 1 > Solución 3 > Solución 2

Tabla 2.7*Evaluación de la ponderación del criterio Adaptable*

Mantenimiento	Solución 1	Solución 2	Solución 3	$\Sigma+1$	Ponderación
Solución 1		0	1	2	0.33
Solución 2	1		1	3	0.5
Solución 3	0	0		1	0.17
			Suma	6	1

Dando como resultado: Solución 2 > Solución 1 > Solución 3

Con la información de manera individual de cada criterio con respecto a las alternativas de solución, se realizó una matriz de decisión, tomando en consideración la ponderación de los criterios individuales para seleccionar la mejor alternativa

Tabla 2.8*Matriz de decisión*

	Mantenimiento	Costo de producción	Carga computacional	Eficiencia energética	Adaptable	Σ	Ponderación
Alternativa 1	0.065	0.084	0.118	0.135	0.033	0.435	1
Alternativa 2	0.033	0.032	0.118	0.046	0.050	0.280	2
Alternativa 3	0.033	0.084	0.045	0.090	0.017	0.270	3

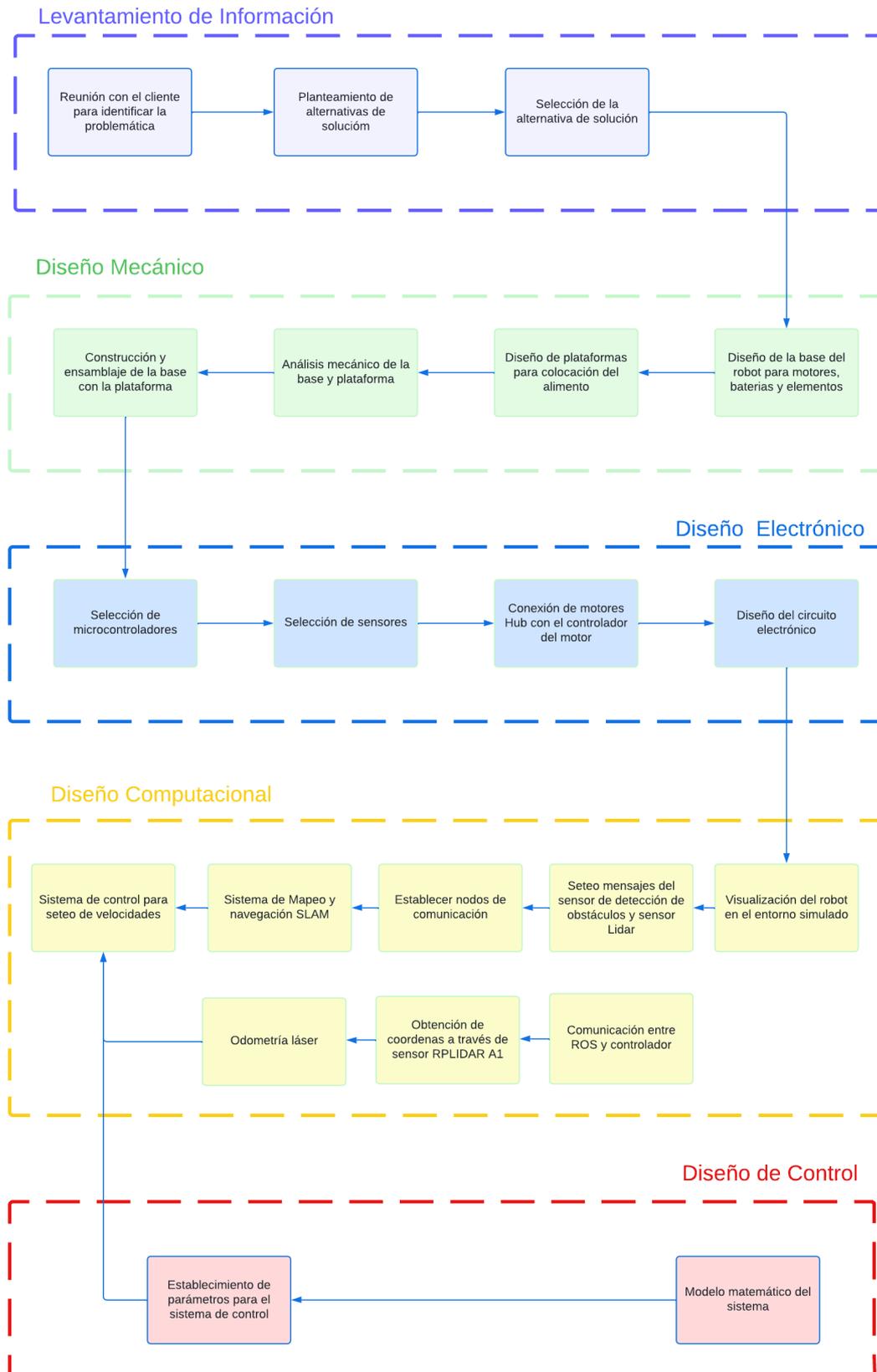
Mediante la matriz de decisión se logró obtener la alternativa con mejor ponderación, siendo esta la alternativa 1 (mapeo SLAM con odometría láser, utilizando un controlador NVIDIA).

2.2 Proceso de Diseño

Se realizó un mapa de metodología con respecto a todos los procesos de diseño individuales para lograr el diseño funcional de nuestra alternativa de solución.

Figura 2.1

Mapa de metodología de diseño



2.3 Diseño mecánico

Para el diseño mecánico se utilizó la herramienta Autodesk Inventor para modelar la estructura del robot con elementos, en su mayoría, normalizados. Utilizando esta misma herramienta, se hizo uso de un análisis de elementos finitos para garantizar que la selección de materiales y el diseño del robot sean capaces de soportar las cargas a las que se someterían a lo largo de su ciclo de vida útil.

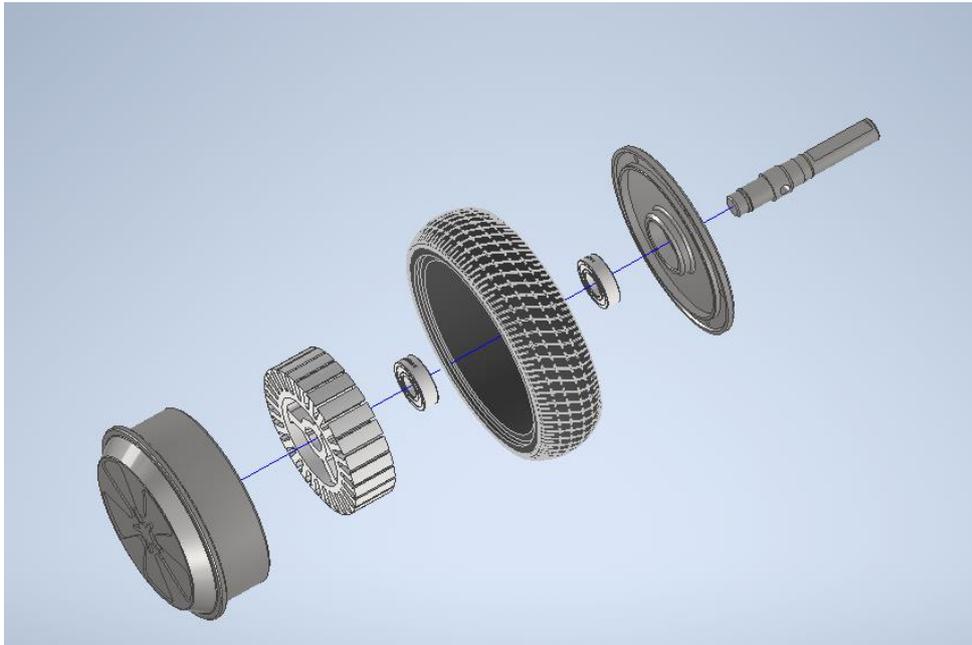
2.3.1 Elementos del diseño mecánico

El diseño mecánico cuenta con dos motores HUB como actuadores principales encargados del movimiento y tracción, dos ruedas giratorias utilizadas para facilitar el balance del cuerpo, tres bases circulares de 40 cm de diámetro para soportar el cuerpo del robot y los componentes electrónicos y tres platos circulares de 40, 30 y 20 cm de diámetro destinados a utilizarse como transporte. Cuenta con un perfil cuadrado de aluminio de 80 cm x 2 cm x 2 cm.

Para fijar los elementos se utilizaron tornillos de cabeza hexagonal AS 1110 M4 y M5 de 8 mm y 10 mm de longitud, tuercas cuadradas y tuercas hexagonales. Las dimensiones del diseño mecánico son aproximadamente 113 cm x 43 cm y se espera que pueda transportar sin problemas 30 kg adicional al propio peso de la estructura. El diseño es desmontable y los elementos se fijan utilizando pernos y tuercas, de modo que, si se presentase algún fallo, la reparación de la estructura no sería complicada y dependería únicamente de reemplazar la pieza deteriorada.

Figura 2.2

Vista explosionada del motor HUB



Nota: Vista explosionada elaborada a partir del modelo 3D elaborado por Corentin Polette, 2018. GrabCad Comunity (<https://grabcad.com/library/6-5-hoverboard-hubmotor-wheel-1>)

Figura 2.3

Vista explosionada del diseño

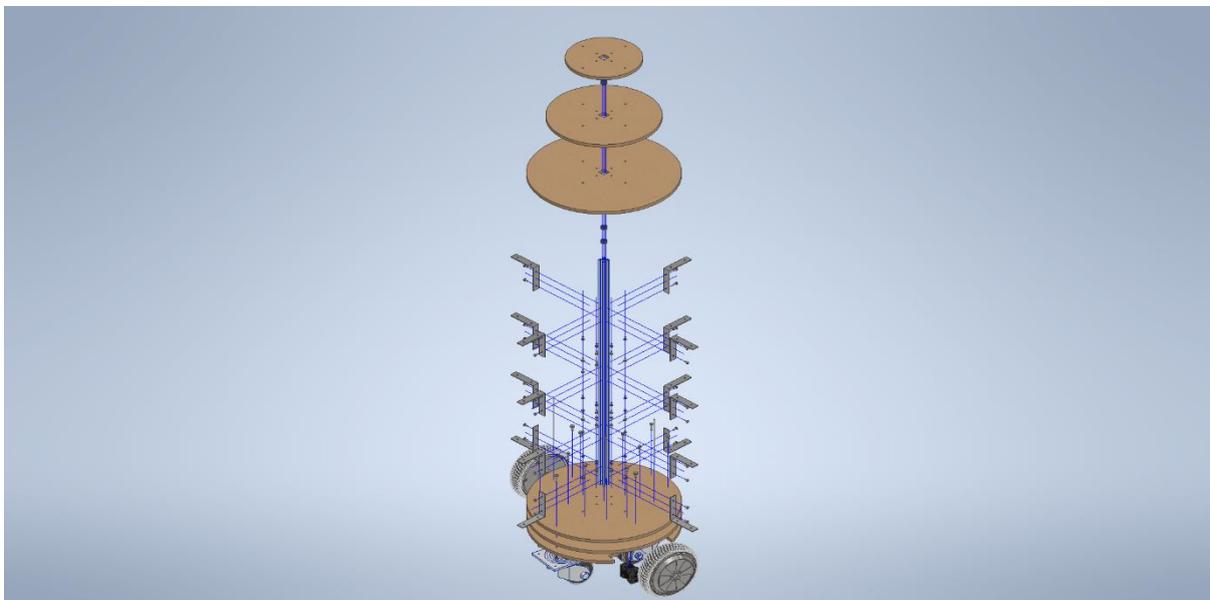
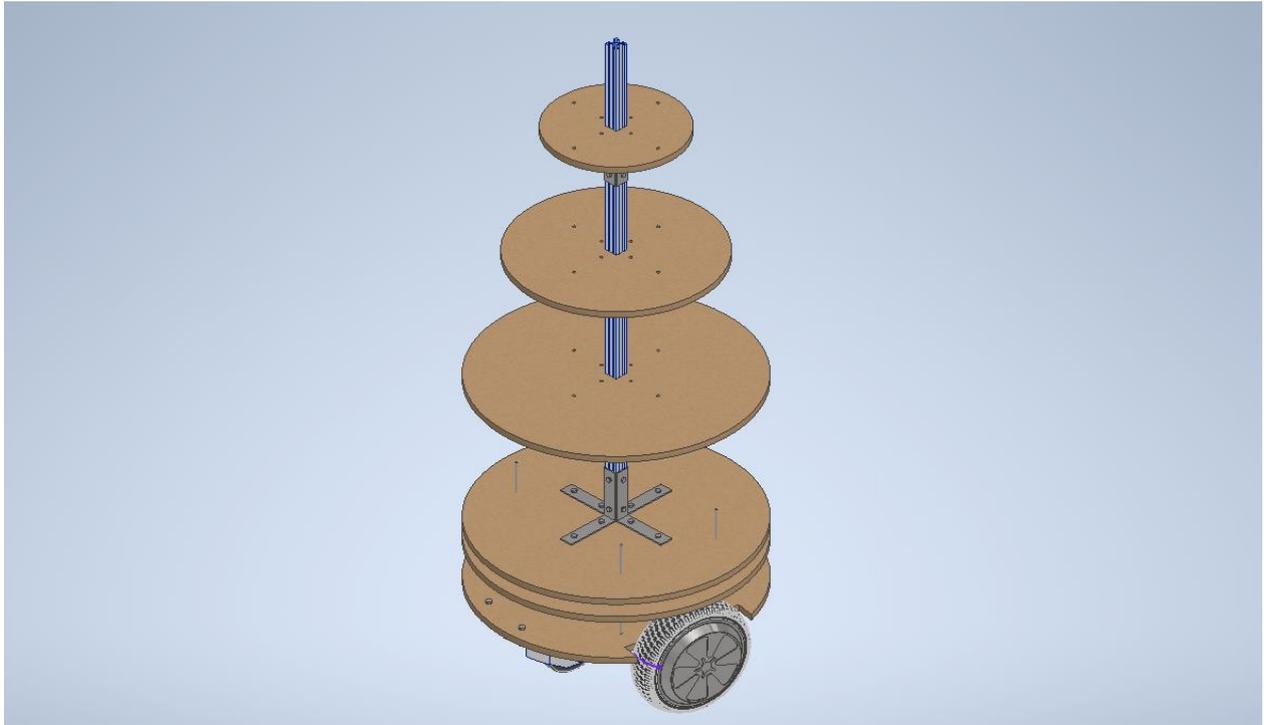


Figura 2.4

Modelo 3D del robot

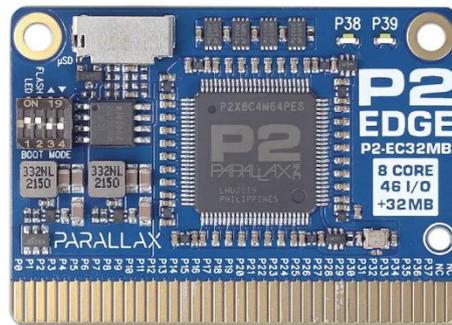


2.4 Diseño Electrónico

Para el funcionamiento del robot, se realizó el análisis de diferentes componentes necesarios para cumplir con el objetivo planteado en la alternativa de solución.

2.4.1 Modulo P2 Edge (32 MB RAM)

El controlador P2 Edge de la empresa Parallax, permite el control de los motores hub y la lectura de los encoders a través del uso de librerías, se utilizó este controlador para el uso de librerías de los motores, que se encuentran en formato “.spin2”, este tipo de lenguaje fue desarrollado por la empresa Parallax Inc.

Figura 2.5*P2 Edge Module with 32MB RAM*

Nota: Imagen obtenida de la página oficial de Parallax. [14]

Las características más importantes del microcontrolador son las siguientes:

Tabla 2.10*Características P2 Edge de Parallax.*

Especificaciones	
Voltaje de Entrada	5-16 VDC
Frecuencia	20MHz
Pines I/O	46
Temperatura de trabajo	-40 °C a 85 °C
Flash (Memoria no volátil)	16 MB

Nota: Información obtenida de la página oficial de Parallax. [14]

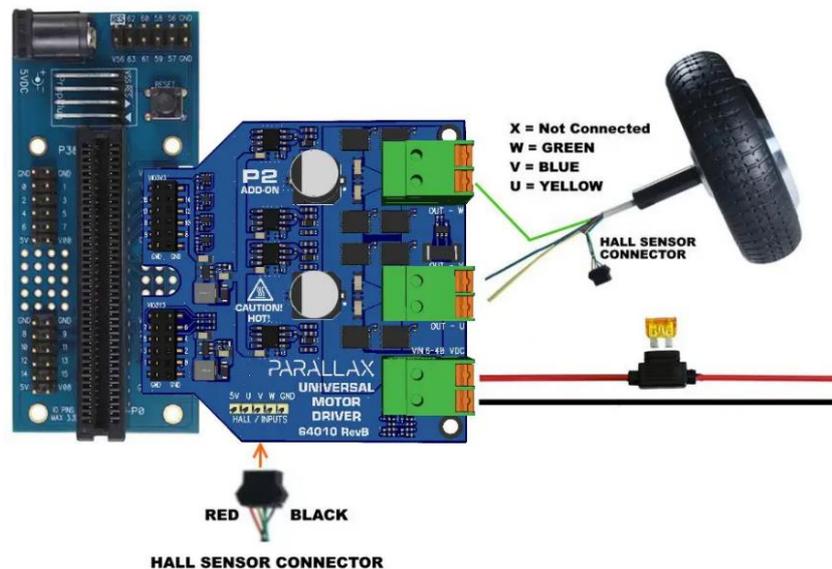
2.4.2 Controlador Universal de Motor

Se utilizaron 2 controladores universales de la empresa Parallax, estos permiten el control de motores hub trifásicos, además de contar con conector del sensor hall para retroalimentación de control del motor. [15]

Tabla 2.11*Características Controlador Universal de Parallax.*

Especificaciones	
Voltaje de Entrada	6-30 VDC
Comunicación	TTL, PWM
Controladores	MOSFET UCC27211D
Temperatura de trabajo	-40 °C a 85 °C

Nota: Información obtenida de la página oficial de Parallax. [15]

Figura 2.6*Universal Motor Driver P2 Add-on Board*

Nota. Imagen obtenida de la página oficial de Parallax, muestra las conexiones del motor Hub y el sensor hall al controlador universal. [15]

2.4.3 Motor Hub con Encoder

Los motores hub seleccionados son de la empresa Parallax, con un diámetro de 6.5 pulgadas, estos motores cumplen con la capacidad de carga elevada y torque necesario para el diseño del robot móvil. [9]

Figura 2.7

Motor Hub con Encoder



Nota. Imagen obtenida de la página oficial de Parallax [9]

Tabla 2.12

Características del Motor Hub con Encoder.

Especificaciones	
Voltaje de funcionamiento	12-36 VDC
Capacidad de carga	240 libras por rueda
Velocidad máxima	15 km/h
Corriente	2 – 15 Amperios dependiendo de la carga
RPM sin carga	600 a 36 VDC
RPM con máxima carga	60 a 36 VDC
Temperatura de trabajo	0 °C a 49 °C

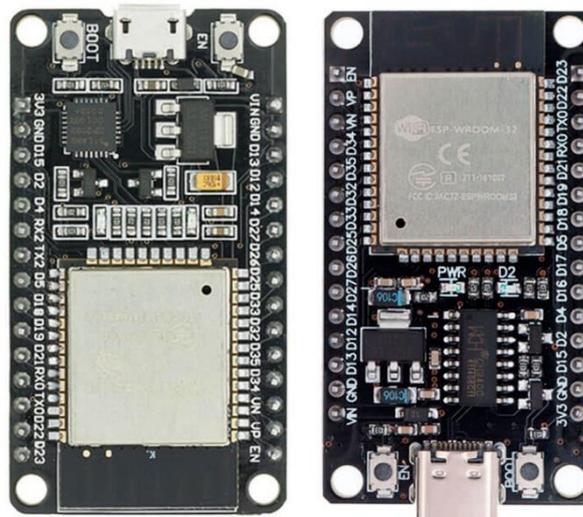
Nota: Información obtenida de la página oficial de Parallax. [9]

2.4.4 Microcontrolador ESP32

El microcontrolador seleccionado para realizar la comunicación entre ROS y el seteo de velocidades de los motores fue la ESP32, esto se debe, a que el microcontrolador P2 Edge no posee las librerías necesarias para establecer una comunicación serial con ROS, por lo tanto, los mensajes enviados por ROS, serán recibidos por la ESP32, y mediante comunicación serial entre la ESP32 y el P2 Edge, se realizará el intercambio de información. [16]

Figura 2.8

ESP32 DevKit V1



Nota. Imagen obtenida de la página UNIT ELECTRONICS. [16]

2.4.5 RPLIDAR A1

El sensor seleccionado para el mapeo del entorno cerrado fue el RPLIDAR A1 de la empresa SLAMTEC, el sistema presenta gira en el sentido de las manecillas del reloj, permitiendo un escaneo 360⁰, perfecto para la aplicación del algoritmo SLAM y generación del mapa área. [17]

Figura 2.9

Sensor RPLIDAR A1



Nota. Imagen obtenida de la página oficial de SLAMTEC. [17]

2.4.6 Cámara de profundidad Intel RealSense D435i

Para el sistema de evasión de obstáculos se utilizó la cámara Intel RealSense D435i, mediante el uso de paqueterías permite el intercambio de información con ROS, permitiendo la detección de personas en un entorno dinámico y la distancia que se encuentra mediante los sensores de profundidad. [18]

Figura 2.10

Cámara de profundidad Intel RealSense D435i



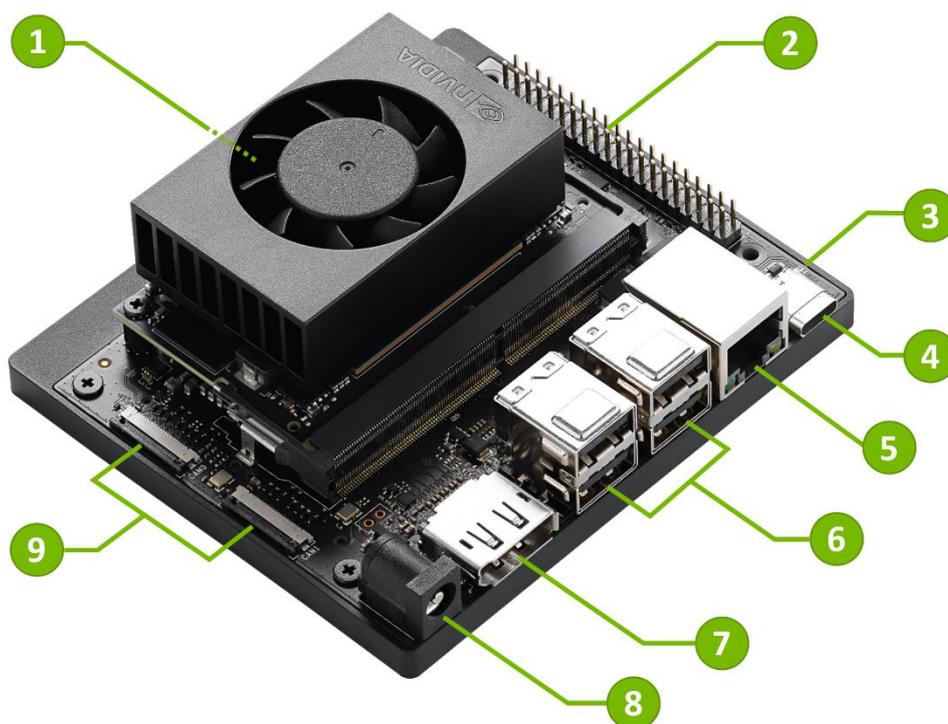
Nota. Imagen obtenida de la página oficial de Intel RealSense. [18]

2.4.7 Jetson Nano

El computador seleccionado para realizar la solución de nuestro producto fue la placa de desarrollo Jetson Nano, gracias al GPU de NVIDIA permite el desarrollo de aplicaciones de inteligencia artificial y procesamientos de datos para ejecutar tareas de navegación autónoma en el área de robots móviles.

Figura 2.11

Kit de desarrollo Jetson Nano



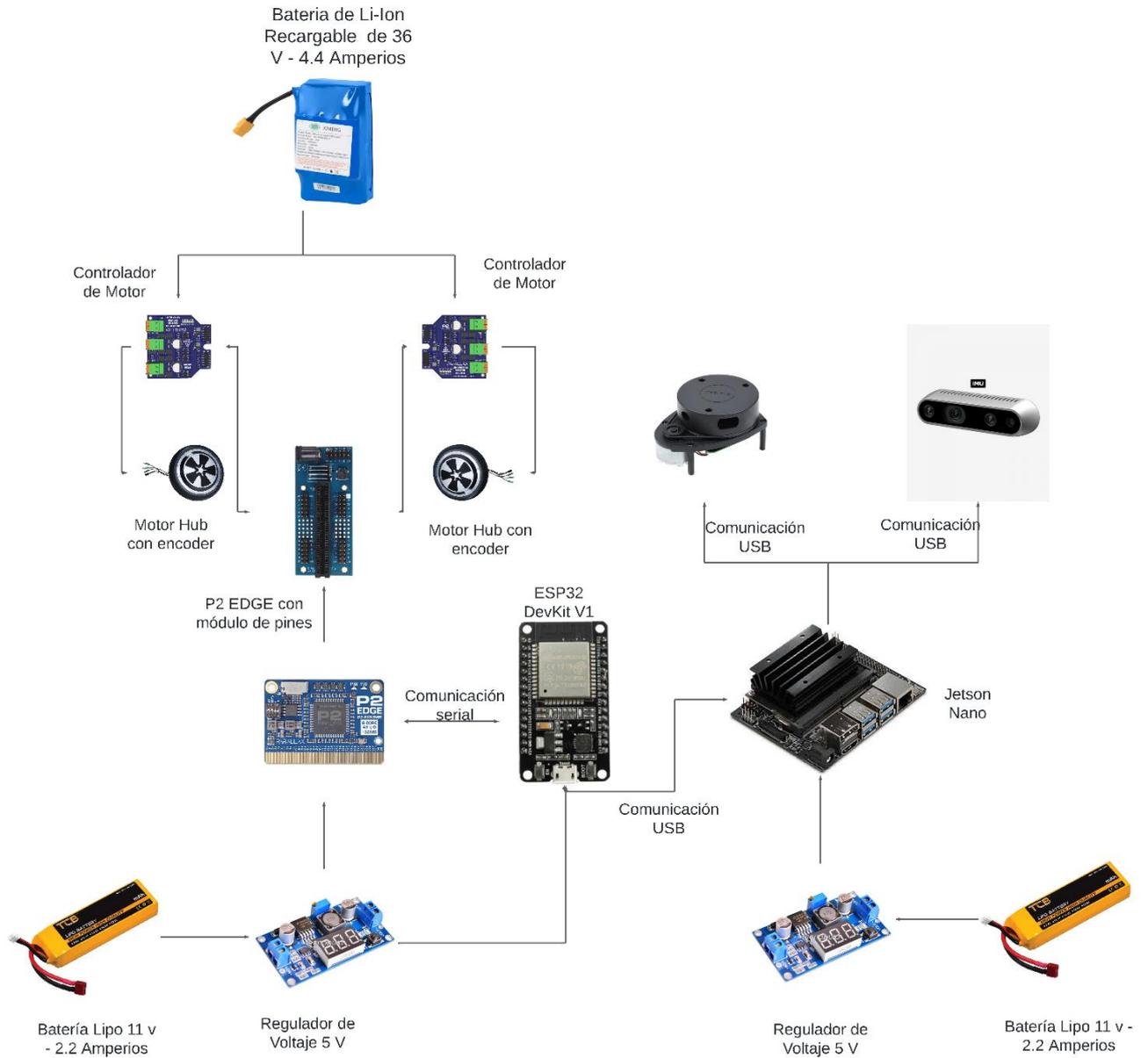
Nota. Imagen obtenida de la página NVIDIA Developer. [19]

2.4.9 Diagrama – Circuito Electrónico

Luego de la selección de componentes para nuestro prototipo final, se realizó el diagrama de conexiones de todos los componentes.

Figura 2.12

Diagrama de conexiones del prototipo final.



2.5 Diseño Computacional

La funcionalidad del robot consiste en poder desplazarse alrededor del perímetro de manera autónoma en entornos dinámicos desconocidos, para ello inicialmente se requiere un sistema encargado del mapeo, planificación y localización, que le permita identificar el

perímetro para formar el mapa de costes con la información del área que el robot puede recorrer considerando los obstáculos estáticos y las dimensiones del robot. A partir de ahí, el siguiente sistema se encargaría de la navegación y localización.

Para hacer esto posible, se utilizará Robot Operating System (ROS) como middleware para la comunicación entre los componentes del sistema, el mapeo, navegación y localización, para ello se seguirán las siguientes dos estructuras de comunicación.

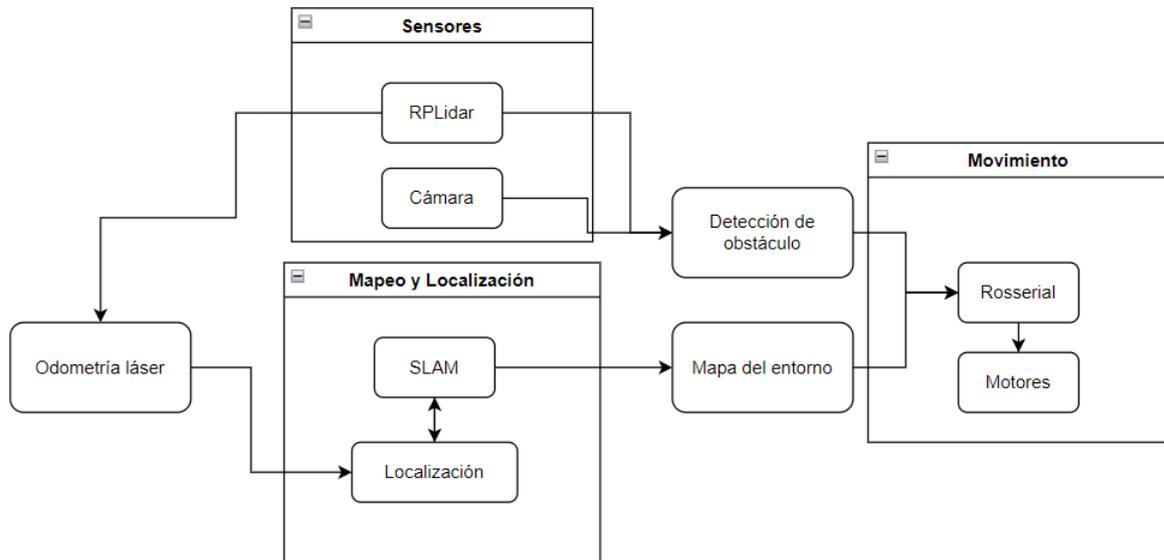
2.5.1 Estructura de comunicación para el mapeo del entorno y la localización del robot

La estructura se maneja de la siguiente manera: el sensor de distancia RPLidar se comunica con ROS para utilizar el algoritmo de mapeo y localización simultáneo (SLAM), este algoritmo utiliza la misma información del sensor de distancia para localizar el robot, de modo que se puede clasificar como odometría láser. La cámara utilizaría visión por computador para detectar obstáculos.

La información resultante del SLAM permitiría obtener un mapa del entorno, que en conjunto con la detección de obstáculos y la velocidad de los motores se podría establecer la planificación global, con ello, a través de Rosserial se enviaría la velocidad de los motores hacia el controlador para ajustar su comportamiento.

Figura 2.13

Estructura de comunicación entre sensores y actuadores para mapeo, localización y movimiento



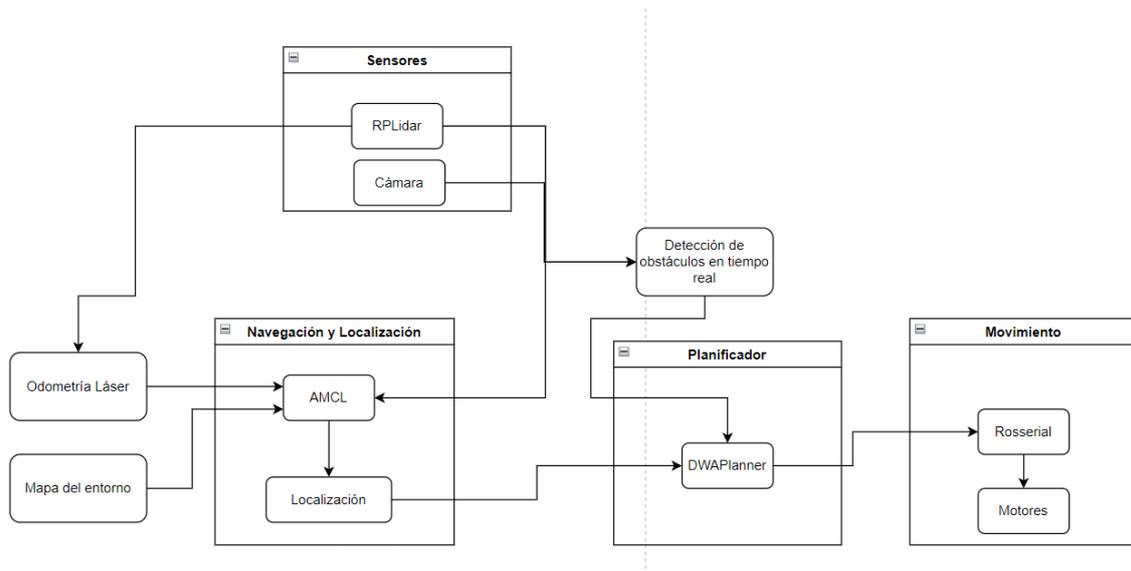
2.5.2 Estructura de comunicación para la navegación y la localización del robot

Para el caso de la navegación, la información del sensor RPLidar al igual que el mapa del entorno generado por el algoritmo SLAM y la odometría láser se utilizarían para estimar la posición del robot, contribuyendo a la localización de este utilizando el algoritmo de Monte Carlo (AMCL) y el mapa extraído del mapeo. Esta información iría al planificador de rutas que traduce esa información en velocidades para los motores, esta información la envía hacia el Rosserial encargado de comunicarse con el controlador hacia los motores.

Adicionalmente, la Cámara continuamente se encuentra analizando la presencia de obstáculos. En caso de detectarse algún obstáculo, se envía un mensaje hacia el planificador para que se comunique con el controlador a través del Rosserial para alterar inmediatamente las velocidades de los motores de modo que se esquite el obstáculo estableciendo una nueva ruta para continuar con el recorrido.

Figura 2.14

Estructura de comunicación entre sensores y actuadores para navegación y localización



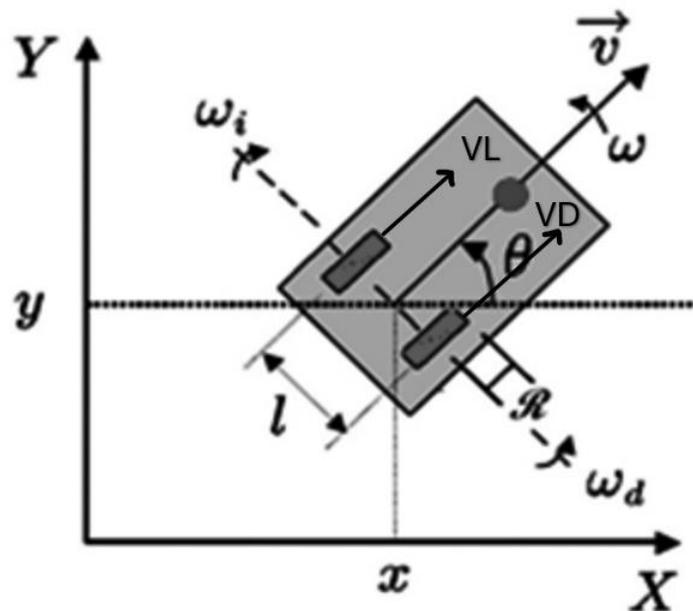
2.6 Diseño de control

2.6.1 Modelo Cinemático – Robot Móvil Diferencial

Para describir el movimiento en función de las velocidades lineales y angulares de nuestro robot, se desarrolló un modelo cinemático, tomando en consideración que el robot se desplaza en una superficie plana y las ruedas giran sin deslizamiento.

Figura 2.14

Esquema del robot móvil diferencial con sus variables



Nota. Figura obtenida del artículo “Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA”. Citar

Las variables y parámetros utilizados para el desarrollo de las ecuaciones cinemáticas para nuestro modelo fueron las siguientes:

- R: radio de las ruedas.
- L: distancia entre las ruedas.
- V_L : velocidad lineal de la rueda izquierda.

- V_D : velocidad lineal de la rueda derecha.
- V : velocidad lineal del robot.
- x : posición del robot en el eje x.
- y : posición del robot en el eje y.
- θ : ángulo del robot con respecto al marco de referencia.
- w : velocidad angular del robot.
- w_i : velocidad angular de la rueda izquierda.
- w_d : velocidad angular de la rueda derecha.

En el caso que las velocidades líneas de la rueda izquierda y derecha sean iguales, el robot se desplaza en una línea recta, por lo tanto, la velocidad lineal del robot se puede definir como el promedio de las velocidades líneas de ambas ruedas por el radio de las ruedas.

$$V = R \frac{V_L + V_D}{2} \quad (1)$$

La velocidad angular del robot se puede definir como la diferencia de la velocidad lineal de cada rueda sobre la distancia entre ambas ruedas, también se debe considerar que es proporcional al radio de las ruedas.

$$w = R \frac{V_L - V_D}{L} \quad (2)$$

Se puede establecer una relación de la velocidad de cada rueda con respecto a la velocidad lineal y angular del robot:

$$V_L = V - \frac{L}{2}w \quad (3)$$

$$V_D = V + \frac{L}{2}w \quad (4)$$

Para obtener el modelo cinemático de nuestro sistema, se toma en consideración el cambio de las velocidades lineales en dirección de cada eje y la velocidad angular con respecto al tiempo.

$$\dot{x} = v * \cos\theta \quad (5)$$

$$\dot{y} = v * \sen\theta \quad (6)$$

$$\dot{\theta} = w \quad (7)$$

Se puede representar las ecuaciones del modelo cinemático en su forma matricial, dando como resultado:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sen\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (8)$$

Se reemplaza las variables v y w con las ecuaciones (1) y (2), obteniendo la siguiente ecuación:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sen\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R \frac{V_L + V_D}{2} \\ R \frac{V_L - V_D}{L} \end{bmatrix} \quad (9)$$

Finalmente, se multiplican las matrices de rotación y de velocidad para obtener como resultado nuestro sistema matricial de velocidades de nuestro robot con respecto al marco de referencial global.

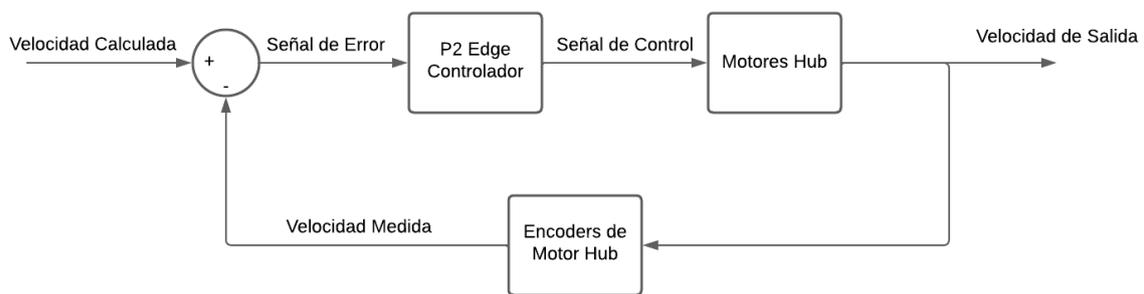
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\cos\theta}{2} \\ R \frac{\sen\theta}{2} & R \frac{\sen\theta}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} w_d \\ w_i \end{bmatrix} \quad (10)$$

2.6.2 Sistema de control

Para el diseño de nuestro sistema de control de velocidades de los motores Hub utilizados en nuestro proyecto, se decidió utilizar las ecuaciones (3) y (4) de nuestro modelo cinemático para establecer las velocidades de cada rueda, el microcontrolador P2 Edge es el encargado de recibir las velocidades seteadas mediante una comunicación serial con el microcontrolador ESP3, y gracias a la librería que ofrece Parallax se utilizó los encoders de los motores para medir la velocidad real de nuestros motores Hub.

Figura 2.15

Sistema de lazo cerrado para manejo de velocidades



Para el uso del control PID para la regulación de la velocidad de cada motor HUB, se utilizó el controlador de driver universal de la empresa Parallax, las 2 variables requeridas para el uso del controlador PID integrado en nuestro controlador de motor fueron: velocidad de referencia (velocidad enviada mediante comunicación serial a través del ESP32) y velocidad medida, esta velocidad se obtiene mediante el uso de los sensores hall integrados en los motores Hub, gracias a estos valores se logró controlar el movimiento del robot de una manera óptima.

CAPÍTULO 3

3.1 Resultados y análisis

3.1.1 Materiales

Tabla 3.1

Tabla de los materiales para la estructura completa del robot (No incluye motores o ruedas)

Pieza	Material
Primer nivel	MDF
Segundo nivel	MDF
Tercer nivel	MDF
Base principal	MDF
Base Lidar	MDF
Base de las ruedas	MDF
Eje principal	Aluminio 6061
Ejes de las bases	Aluminio 6061
Perno hexagonal M5x20mm	Acero ASTM A36
Pernos cilíndricos M5x10mm	Acero ASTM A36
Tuercas T M5	Acero ASTM A36
Soportes escuadra	Acero inoxidable AISI 304

Las propiedades físicas de cada material fueron extraídas del software Autodesk Inventor y se detallan a continuación:

Figura 3.1

Propiedades físicas de material MDF

Editor de materiales: MDF Tablero de fibras de densidad media

Identidad Apariencia **Físico**

Comportamiento isotrópico

▼ Térmico básico

Conductividad térmica 1,200E-01 Con (m · k)

Calor específico 1,300 J/ (G · ° C)

Coefficiente de dilatación térmica 0,100 $\mu\text{m}/(\text{m}\cdot^{\circ}\text{C})$

▼ Mecánico

Módulo de Young 2,400 GPa

Coefficiente de Poisson 0,25

Módulo cortante 0,000 MPa

Densidad 0,800 g/cm^3

Coefficiente de amortiguamiento 0,00

▼ Resistencia

Límite de elasticidad 7,720 MPa

Resistencia máxima a tracción 7,720 MPa

Aceptar Cancelar Aplicar

Figura 3.2

Propiedades físicas de material Aluminio 6061

Editor de materiales: Aluminio 6061

Identidad Apariencia **Físico**

▼ Térmico básico

Conductividad térmica 1,670E+02 Con (m · k)

Calor específico 0,897 J/ (G · ° C)

Coefficiente de dilatación térmica 23,600 $\mu\text{m}/(\text{m}\cdot^{\circ}\text{C})$

▼ Mecánico

Módulo de Young 68,900 GPa

Coefficiente de Poisson 0,33

Módulo cortante 25864,000 MPa

Densidad 2,700 g/cm^3

Coefficiente de amortiguamiento 0,00

▼ Resistencia

Límite de elasticidad 275,000 MPa

Resistencia máxima a tracción 310,000 MPa

Tratado térmicamente

Aceptar Cancelar Aplicar

Figura 3.3

Propiedades físicas de material Acero ASTM A36

Editor de materiales: Acero ASTM A36

Identidad Apariencia **Físico**

▼ Térmico básico

Conductividad térmica 4,500E+01 Con (m · k)

Calor específico 0,480 J/ (G · ° C)

Coefficiente de dilatación térmica 11,700 $\mu\text{m}/(\text{m}\cdot^{\circ}\text{C})$

▼ Mecánico

Módulo de Young 199,959 GPa

Coefficiente de Poisson 0,30

Módulo cortante 76908,000 MPa

Densidad 7,850 g/cm^3

Coefficiente de amortiguamiento 0,00

▼ Resistencia

Límite de elasticidad 248,225 MPa

Resistencia máxima a tracción 399,900 MPa

Tratado térmicamente

Aceptar Cancelar Aplicar

Figura 3.4

Propiedades físicas de material Acero inoxidable AISI 304

Editor de materiales: Acero inoxidable

Identidad Apariencia **Físico**

▼ Térmico básico

Conductividad térmica 1,620E+01 Con (m · k)

Calor específico 0,477 J/ (G · ° C)

Coefficiente de dilatación térmica 10,400 $\mu\text{m}/(\text{m}\cdot^{\circ}\text{C})$

▼ Mecánico

Módulo de Young 193,000 GPa

Coefficiente de Poisson 0,30

Módulo cortante 86000,000 MPa

Densidad 8,000 g/cm^3

Coefficiente de amortiguamiento 0,00

▼ Resistencia

Límite de elasticidad 250,000 MPa

Resistencia máxima a tracción 540,000 MPa

Tratado térmicamente

Aceptar Cancelar Aplicar

3.1.2 Análisis del Diseño Mecánico

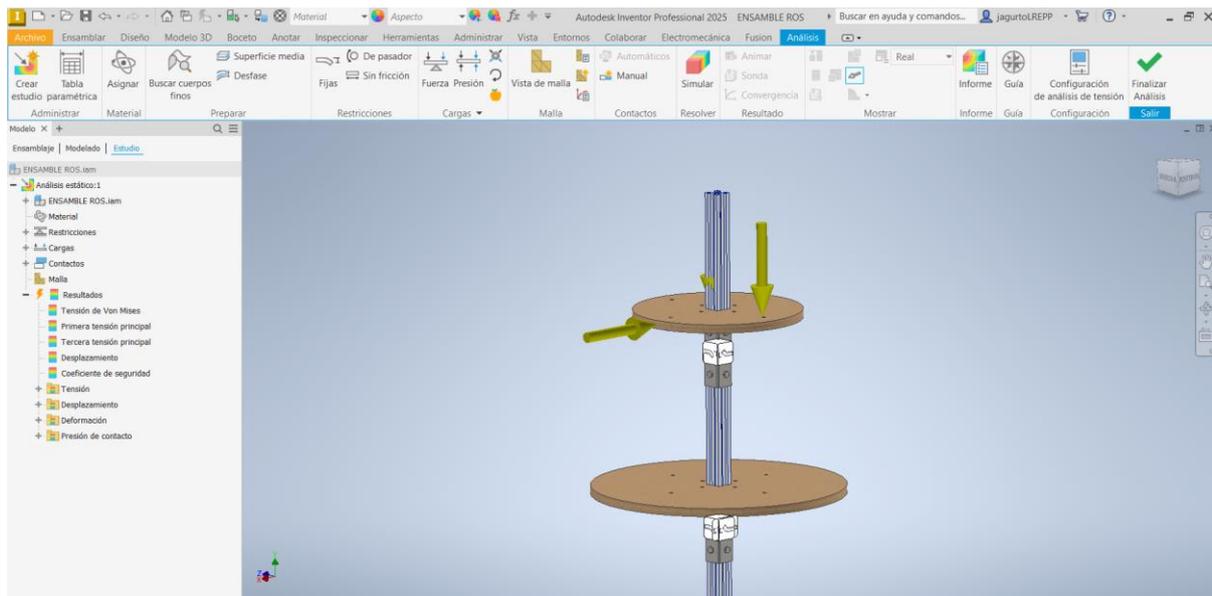
3.1.2.1 Definición de cargas y restricciones

Para realizar el análisis mecánico del diseño elaborado se utilizó el software CAD Inventor, en él, se realizará un análisis con cargas estáticas en diferentes escenarios evaluando que el diseño sea capaz de soportar cargas de hasta 30 kg. En todos los escenarios se tomará en cuenta la gravedad.

Para los tres escenarios, se asumirá que la estructura avanza con una aceleración lineal constante de 0.1 m/s^2 y se colocará la carga de 30 kg (300 N) en el extremo de cada nivel para evaluar el peor escenario, cuando la carga se coloca en un extremo del nivel (generando un torque mayor).

Figura 3.5

Primer escenario – Estructura en aceleración constante y carga en el nivel más alto.

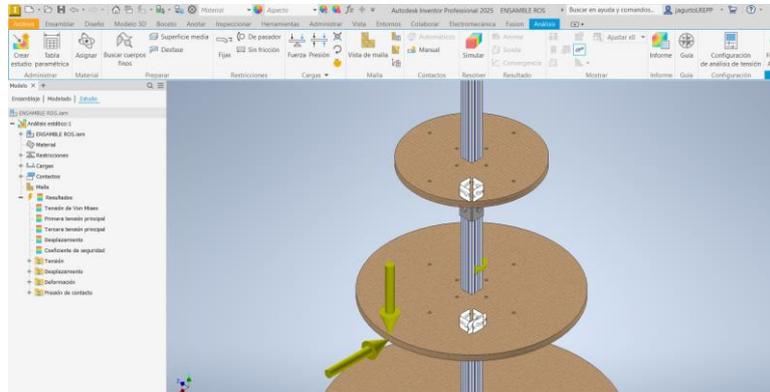


Nota: Se definió la carga de 300 N perpendicular al nivel más alto y una carga de 1 N horizontal.

La carga horizontal se estableció aproximando la masa total de la estructura siguiendo la ecuación $F = m \times a$ aproximando la masa de la estructura a 10 kg.

Figura 3.6

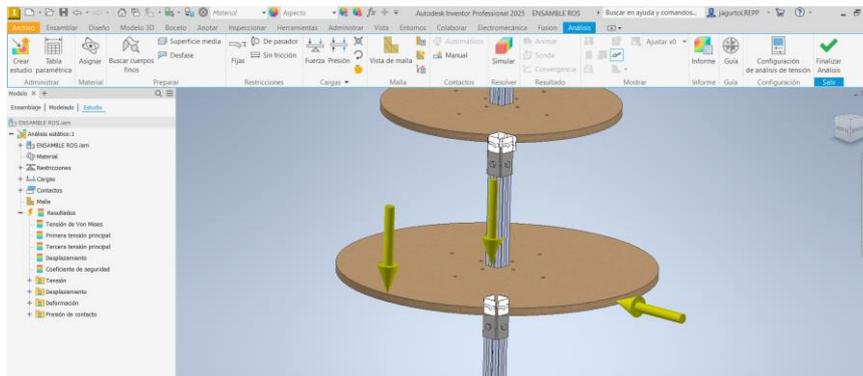
Segundo escenario – Estructura en aceleración constante y carga en el segundo nivel



Para este segundo escenario las cargas son iguales al primer escenario asumiendo el peor de los casos: que la carga se encuentre en el extremo del nivel.

Figura 3.7

Tercer escenario – Estructura en aceleración constante y cargas en el tercer nivel



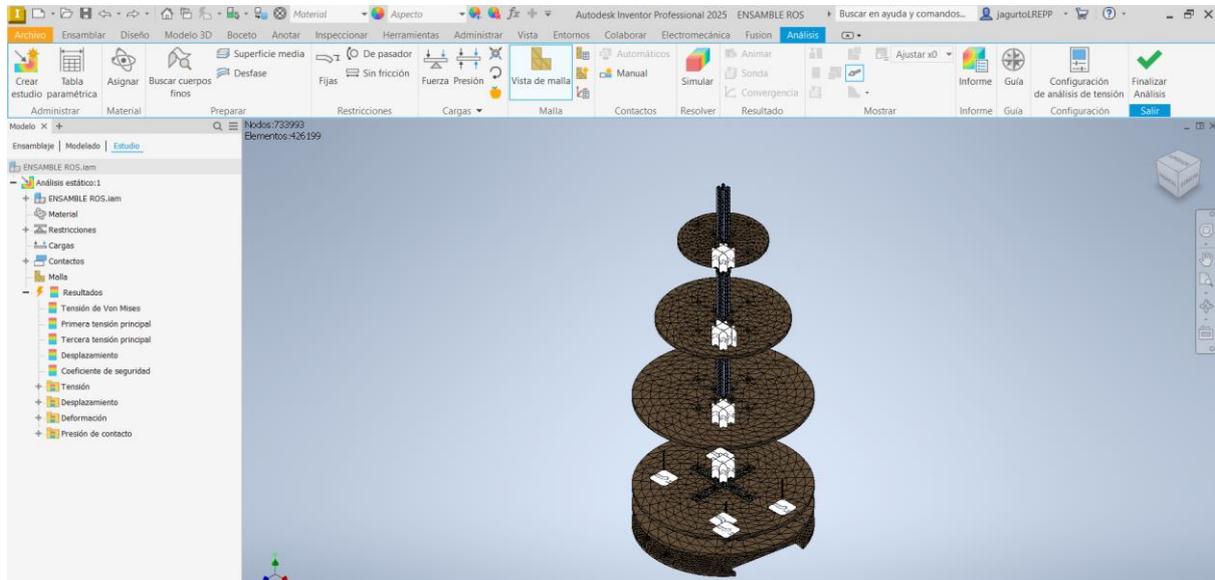
Con estos tres escenarios es suficiente para determinar el rendimiento de la estructura bajo cargas, debido a que si son capaces de soportar de manera independiente los 30 kg en el extremo más alejado del eje eso asegura que la estructura soporte los 30 kg cuando se encuentren distribuidos entre los tres niveles.

3.1.2.2 Generación de malla para análisis de elementos finitos

Para ejecutar el análisis de elementos finitos se generó la vista de malla correspondiente

Figura 3.8

Vista de malla para análisis de elementos finitos

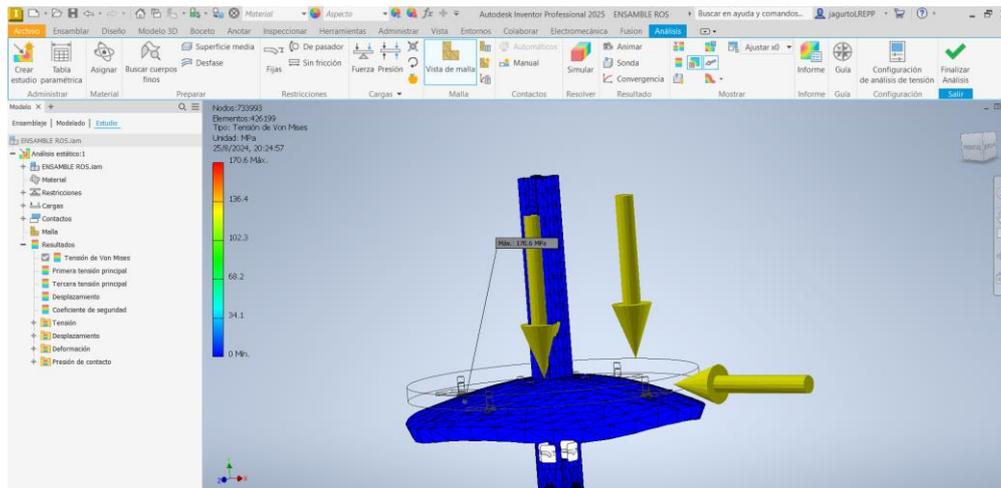


3.1.2.3 Análisis de elementos finitos

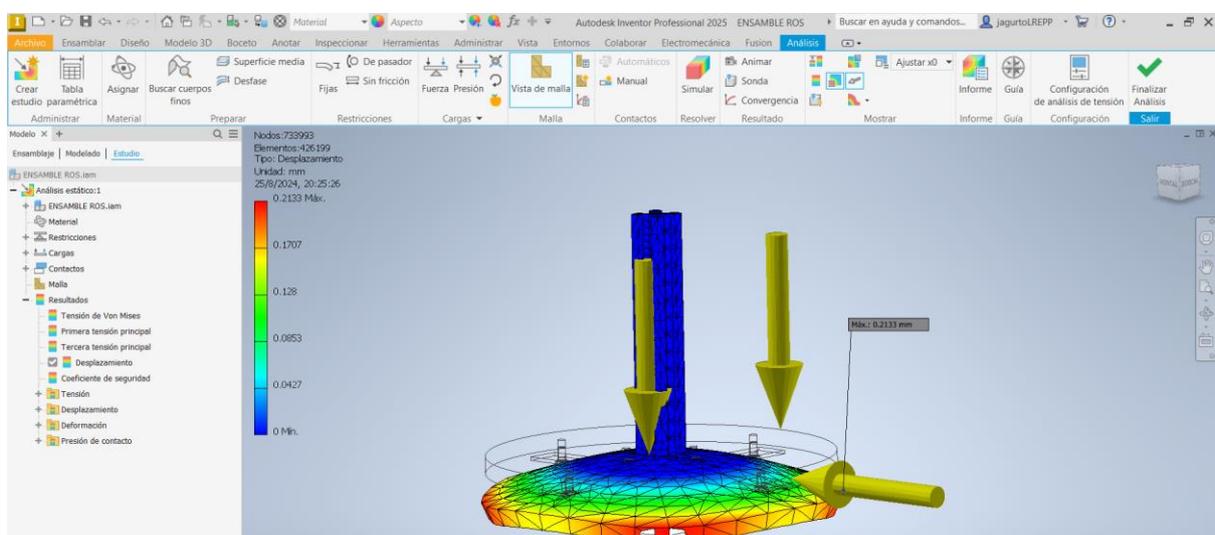
Para el análisis de elementos finitos se tomó en cuenta los resultados de Tensión de Von Mises y el Desplazamiento en cada caso.

3.1.2.3.1 Primer caso: aceleración constante y cargas en el primer nivel

Para este primer caso, se asumen todas las cargas en el primer nivel

Figura 3.9*Análisis de elementos finitos: Tensión de Von Mises – Primer Caso*

El estudio estático del primer caso arrojó como resultado un esfuerzo de Von Mises Máximo de 170.6 MPa ubicado en uno de los pernos colocados en el extremo, el material utilizado para los pernos es el Acero ASTM A36 con un límite de fluencia de 248.25 MPa, arrojándonos un coeficiente de seguridad de 1.45, de manera que el primer nivel soporta de manera óptima esta carga.

Figura 3.10*Análisis de elementos finitos: Desplazamiento – Primer Caso*

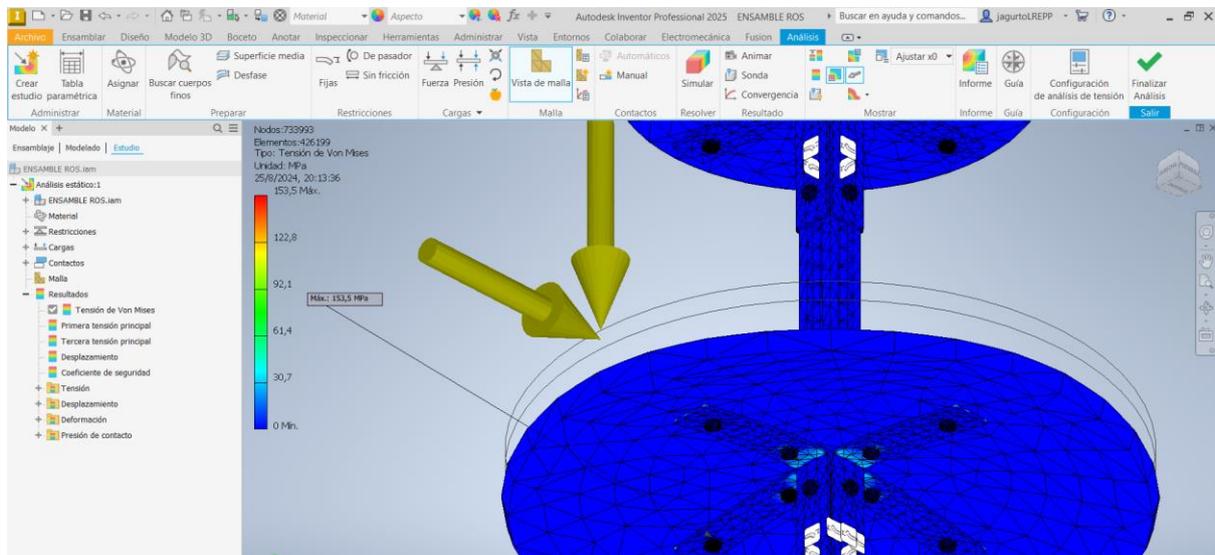
Con respecto al desplazamiento, se observa un desplazamiento de 0.21 mm en los extremos, mismos que no generan una inclinación suficiente para desestabilizar lo que está transportando y no representa un riesgo a lo largo del movimiento.

3.1.2.3.2 Segundo caso: aceleración constante y cargas en el segundo nivel

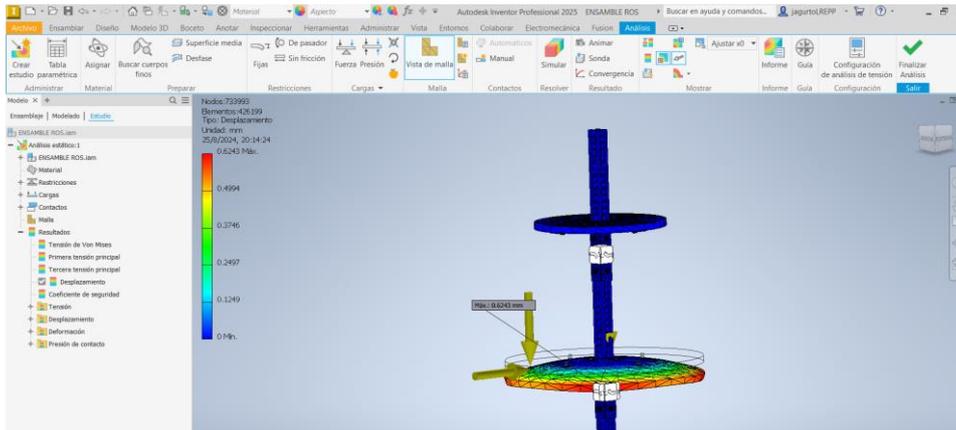
Para este segundo caso, se utilizan cargas en el extremo del segundo nivel y sometido a una aceleración constante.

Figura 3.11

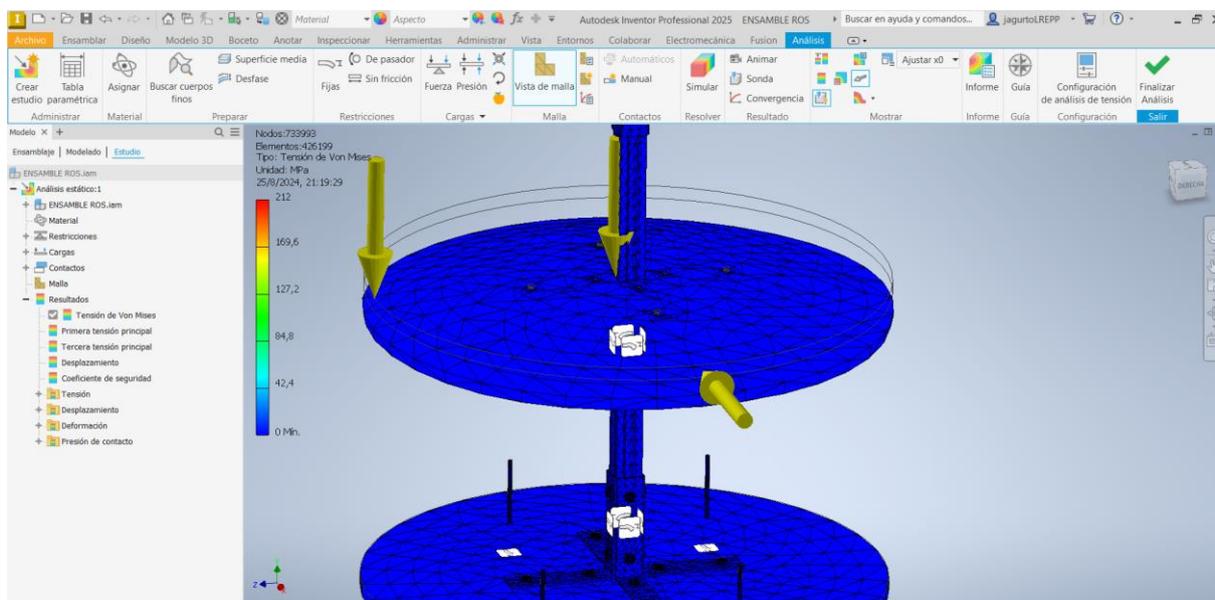
Análisis de elementos finitos: Tensión de Von Mises – Segundo Caso



Al igual que en el caso anterior, el esfuerzo máximo de Von Mises encontrado se muestra en un perno colocado en uno de los extremos de la estructura con un valor de 153.5 MPa. Comparado nuevamente con el límite de fluencia de 248.25 MPa se obtiene un coeficiente de seguridad de 1.62, lo que nuevamente garantiza que el segundo nivel soportaría sin problema una carga de 30 kg

Figura 3.12*Análisis de elementos finitos: Desplazamiento – Segundo Caso*

Para este segundo caso se observa un desplazamiento mayor en los extremos que en el caso anterior, sin embargo, al ser menor a 1 mm sigue sin representar un riesgo mayor en la estabilidad de lo que se esté transportando.

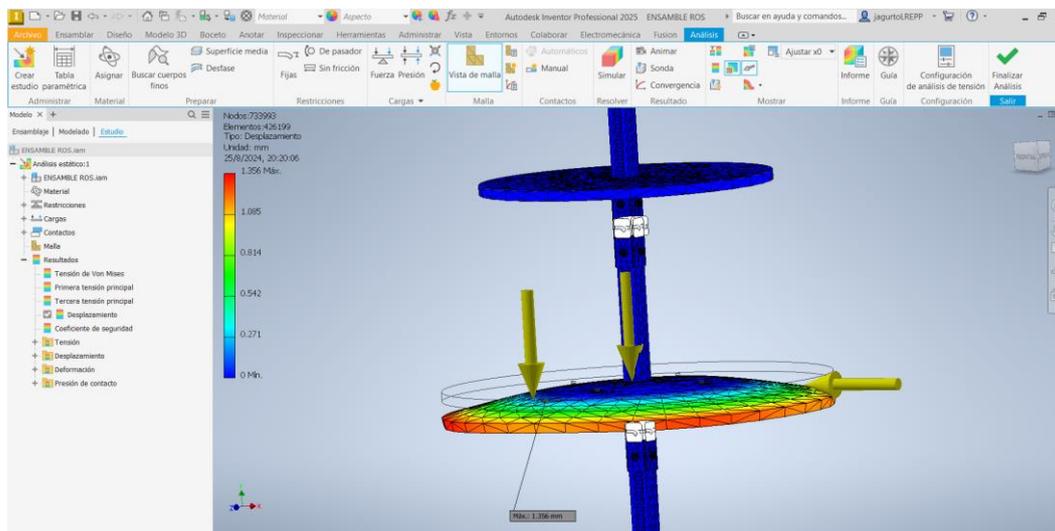
3.1.2.3.3 Tercer caso: aceleración constante y cargas en el tercer nivel**Figura 3.13***Análisis de elementos finitos: Tensión de Von Mises – Tercer Caso*

Al igual que los casos anteriores, el esfuerzo máximo de Von Mises encontrado se muestra en uno de los pernos colocado en uno de los extremos del tercer nivel con un valor de

212.33 MPa. Comparado nuevamente con el límite de fluencia de 248.25 MPa se obtiene un coeficiente de seguridad de 1.17, que, a pesar de aun ser superior a 1, es posible que con el desgaste del material este coeficiente de seguridad sea menor que 1 por lo que no se recomendaría someter al tercer nivel a cargas tan altas ni tan alejadas del eje central

Figura 3.14

Análisis de elementos finitos: Desplazamiento – Tercer Caso



El desplazamiento en este tercer caso es de 1.2 mm, a pesar de ser visualmente notable aun no representa un riesgo de estabilidad para lo que se esté transportando.

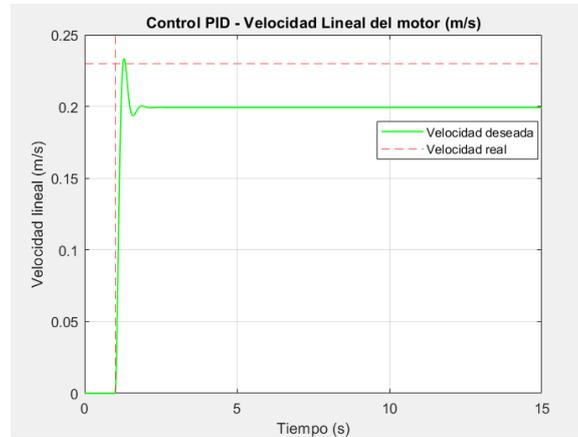
3.1.3 Control de Velocidad

Para el diseño del controlador de ambos motores se optó por utilizar las librerías proporcionadas por la empresa Parallax para el uso de un control PID en el control de velocidades de ambos motores, como se muestra en la **Figura 2.15**, el setpoint del controlador es la velocidad calculada mediante los valores de odometría de nuestro robot, la retroalimentación se realiza mediante el uso de encoders de cada motor, con dichos valores se establece la velocidad medida; al tener ambos datos se procedió a utilizar un código dentro de nuestro microcontrolador que realice un controlador PID para ajustar la velocidad enviada a los motores HUB.

Para el establecimiento de los valores para los parámetros K_p , K_i y K_d se realizó diferentes configuraciones para el control de velocidades del robot, hasta obtener un controlador que se adapte a los requerimiento y cambios del robot.

Figura 3.15

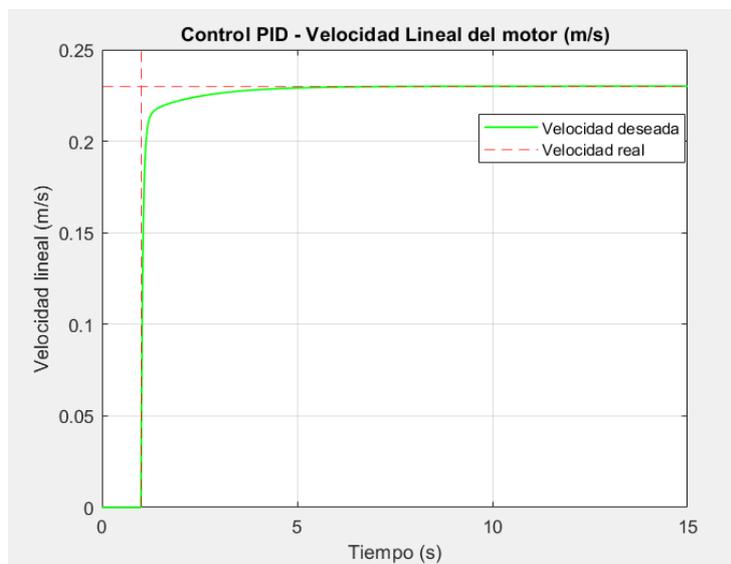
Control de velocidad – Primera etapa



En la figura se puede observar un setpoint de 0.23 m/s, utilizando un K_p de 65 se obtuvo una respuesta rápida pero el valor donde se estabiliza es 0.20 m/s.

Figura 3.16

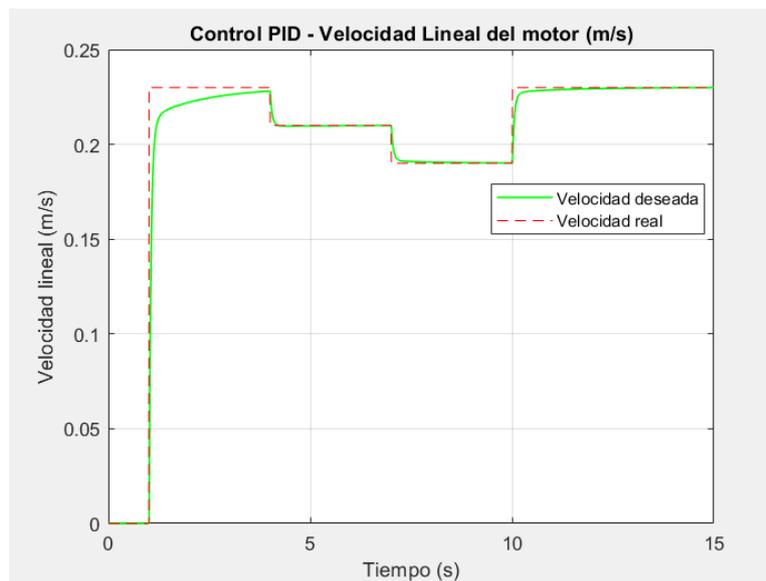
Controlador PID con ajuste de parámetros.



En la figura se observa una mejor respuesta con respecto a la **Figura 3.1**, los valores para nuestros parámetros fueron: $K_p = 100$, $K_i = 70$ y $K_d = 10$, con esta configuración al sistema se logra estabilizar en un instante, generando que no existan tiempos de retraso para el mandado de velocidades a los motores.

Figura 3.17

Control PID con respecto a cambio de velocidades



Se puede observar que el controlador PID con los parámetros establecidos, permite adaptarse al cambio de velocidades en intervalos de tiempos cercanos, gracias a esto se eliminaron los tiempos de retraso que se generaban al momento de estabilizar las velocidades de los motores.

3.1.4 Implementación Real

3.1.4.1 Armado de base y plataforma

Se imprimió la plataforma y base del robot utilizando MDF, el robot cuenta con 2 secciones, la primera sección corresponde a los componentes electrónicos necesarios para el funcionamiento (base), y la segunda sección corresponde a las plataformas encargadas de contener los alimentos durante el movimiento del robot.

Figura 3.18

Montaje de motores hub y castor wheels en la base principal.

**Figura 3.19**

Armado de la plataforma de 3 pisos.

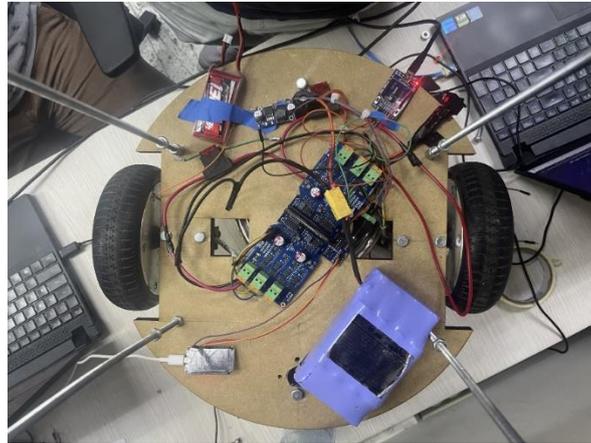


3.1.4.2 Conexiones eléctricas

De acuerdo con las conexiones realizadas en la **Figura 2.12** se realizaron las conexiones electrónicas entre los componentes seleccionados.

Figura 3.20

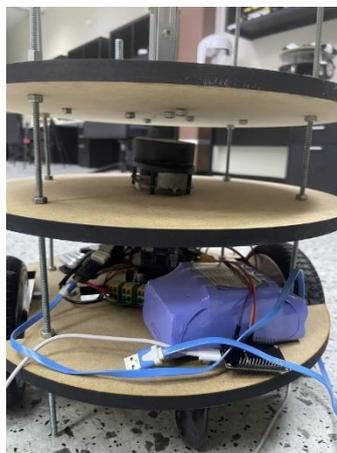
Conexiones electrónicas entre los motores HUB y el microcontrolador P2 Edge



En la siguiente base se colocó solamente el sensor RPLIDAR A1 para que pueda realizar la tarea de mapeado sin problemas, puesto que, si detecta objetos alrededor generaría conflictos al momento de generar el mapa de nuestro entorno.

Figura 3.21

Segunda base del robot con el sensor RPLIDAR A1



3.1.4.3 Ensamblaje final del robot

Al tener las conexiones electrónicas en la base de nuestro robot y el armado de la plataforma, se unieron los componentes para obtener el diseño final de nuestro robot móvil con 3 plataformas diseñadas para el transporte de alimentos en un entorno cerrado.

Figura 3.22

Construcción final del robot



3.1.5 Funcionamiento del Robot

3.1.5.1 Entorno de trabajo y Mapeo

Para realizar el mapeo del entorno inicialmente se estableció un espacio de trabajo con el cual se realizarían las pruebas pertinentes para evaluar el desempeño del robot. El espacio de trabajo que se utilizó fueron las instalaciones del CIDIS.

Figura 3.23

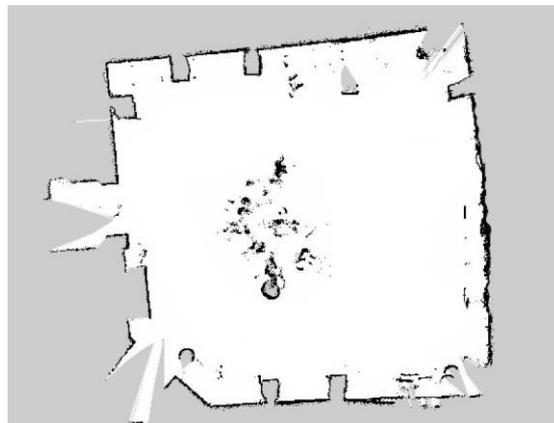
Entorno de trabajo para pruebas del robot



Nota: Para realizar el mapeo no se alteró el entorno de ninguna manera, permitiéndonos evaluar el desempeño del robot ante la presencia de diferentes tipos de obstáculos en el camino.

Figura 3.24

Mapa extraído del entorno utilizando SLAM



Para realizar el mapeo del entorno de trabajo del robot se utilizó el algoritmo de localización y mapeo simultáneo (SLAM) hector mapping acompañado de un sensor LiDar. El control del robot se realizó utilizando teleop para desplazarlo alrededor de toda el área de trabajo. Las secciones oscuras del mapa visibles en la **Figura 3.24** representan los obstáculos,

que pueden ser objetos presentes en el entorno o las paredes de este, espacios que el robot no puede atravesar, pero posiblemente sí rodear. El mapa corresponde a una representación 2D nivelada a la altura en la que se colocó el sensor LiDar del robot (aproximadamente 25 cm con respecto al suelo), lo que significa que con el mapeo no se puede distinguir la presencia de obstáculos más pequeños o que se encuentran elevados del suelo.

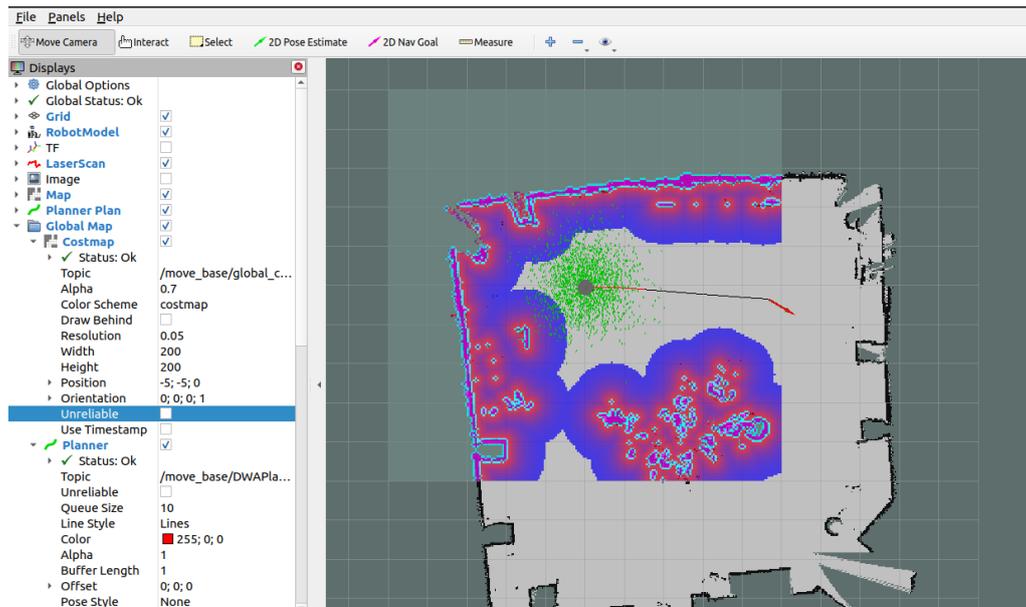
El mapa generado por el algoritmo SLAM es muy importante porque servirá como base para la navegación autónoma.

3.1.5.2 Planificación y navegación

Para realizar la navegación y planificación se utilizó el algoritmo de localización de Monte Carlo (AMCL), el cual coloca diferentes partículas en el mapa para aproximar la localización del robot con base en su odometría. A partir de ahí, se utilizó el algoritmo de planeación DWAPlaner para realizar la planificación global y local utilizando como referencia el mapa obtenido en la sección anterior, esta planificación generaría el mapa de costos con la información necesaria para que el robot conozca qué secciones del mapa puede recorrer.

Figura 3.25

Localización y mapa de costos local y global utilizando DWAPlanner y AMCL



Los colores azules y rojos en la **Figura 3.25** representan el mapa de costos del planificador local, con la información relevante sobre los lugares por el cual puede circular el robot. La flecha roja representa la meta del robot, mientras que la línea oscura que se traza entre el robot y la flecha roja representa la ruta que el planificador ha trazado como el camino más corto hacia la meta y en caso de presentarse algún otro obstáculo, el planificador local traza una nueva ruta.

Figura 2.26

Robot realizando la navegación transportando un paquete



En las pruebas realizadas el robot inicialmente tiene dificultades para encontrar la orientación correcta para la ruta planificada, sin embargo, logra alcanzar la meta final sin ningún otro inconveniente.

3.1.6 Análisis de Costos

Tabla 3.2

Tabla de elementos y mano de obra

Análisis de Costos				
Categoría	Descripción	Costo Unitario	Unidades	Costo Total
Componentes	Modulo P2 Edge	\$ 89.00	1	\$ 89.00
Electrónicos	Tabla de expansión	\$ 24.95	1	\$ 24.95
	– P2 Edge			
	Controlador	\$ 79.95	2	\$ 159.90
	Universal de Motor			
	Motor Hub	\$ 79.00	2	\$ 158.00
	Bloque de Montaje de Motor Hub	\$ 39.00	2	\$ 78.00
	Castor Wheel	\$ 39.00	2	\$ 78.00
	Rplidar A1	\$ 100.00	1	\$ 100.00
	ESP32	\$ 15.00	1	\$ 15.00
Equipo mecánico y mecanizado	Perfil de Aluminio V 20x20	\$ 6.27	1	\$ 6.27
	Varilla roscada ¼	\$ 6.00	1	\$ 6.00
	Impresión MDF	\$ 39.00	1	\$ 39.00
	Soportes en L	\$ 13.00	1	\$ 13.00
	Pernos y Tuercas	\$ 12.00	1	\$ 12.00
Mano de obra	Diseño mecánico del robot	\$ 50.00	1	\$ 75.00
	Armado del robot	\$ 25.00	1	\$ 50.00
	Programación de microcontroladores	\$ 100.00	1	\$ 200.00
	Programación y simulación en ROS	\$ 150.00	1	\$ 150.00
Total				\$ 1248.39

De acuerdo con el análisis de los materiales, componentes electrónicos y mano de obra utilizada para el diseño y construcción del robot mesero, da un total de \$1248.38, el robot puede salir al mercado con un valor de alrededor de \$2500, siendo esto una alternativa económica en comparación a productos similares presentes en el mercado, ya que sus valores oscilan entre los \$6000 y \$10000. Además, se puede plantear la opción de ofrecer el producto como un servicio, en el caso que se requiera utilizar el robot para un evento se lo puede rentar por alrededor de \$150, este valor cubriría el transporte, la configuración del robot dentro del entorno cerrado y el respectivo mantenimiento o supervisión en el caso de ocurrir un inconveniente.

CAPÍTULO 4

4.1 Conclusiones y Recomendaciones

4.1.1 Conclusiones

- Se diseñó e implementó un robot móvil diferencial capaz de realizar recorridos de manera autónoma y eficiente en entornos dinámicos cerrados, permitiéndole el transporte eficiente y seguro de alimentos y productos, manteniendo como velocidad máxima 0.20 m/s, velocidad adecuada para desplazarse en entornos rodeados de personas, con la capacidad de trabajar de manera autónoma durante una jornada de 4 horas y contando con un diseño estructural con muy pocas dificultades para dar mantenimiento y realizar reparaciones.
- Se realizó un análisis mecánico del diseño implementado para la estructura externa del robot, arrojando resultados favorables demostrando la alta confiabilidad de la selección de materiales para transportar cargas de hasta 30 kg siempre y cuando se encuentren distribuidos de manera adecuada a lo largo del cuerpo del robot móvil diferencial.
- Se implementó un sistema de navegación autónoma en ROS Noetic utilizando el algoritmo de Monte Carlo para localización basada en odometría láser, DWAPLanner para planificación de trayectorias y navegación, del mismo modo se utilizó SLAM para mapeo del entorno en tiempo real, permitiéndole al robot desenvolverse en un entorno dinámico desconocido y desplazarse evadiendo obstáculos con tiempo de respuesta entre 200 y 400 ms.
- Se realizó un análisis de costos del cual se puede rescatar que el costo total para la construcción del robot fue de \$1248.39, un valor adecuado permitiendo establecer el precio de venta de \$2500.00, precio bastante competitivo comparado con productos similares presentes en el mercado actual.

4.1.2 Recomendaciones

- Se recomienda que, para futuros trabajos, se verifique los tiempos de comunicación de los componentes del sistema por separado, para así buscar optimizarlos de manera independiente evitando altos tiempos de respuesta comprometiendo el funcionamiento.
- Para obtener una localización y una odometría más precisa del robot en tiempo real se recomienda utilizar un sensor IMU que informa parámetros como aceleración, velocidad y orientación.
- Al realizar e implementar el diseño mecánico, se recomienda verificar la posición del centro de masa y la alineación de las ruedas giratorias, ya que ambos factores contribuyen a la estabilidad del diseño durante el movimiento.

Referencias

- [1] L. Zambrano, «Expreso,» 24 Febrero 2024. [En línea]. Available: <https://www.expreso.ec/actualidad/economia/ia-robots-alivian-carga-laboral-190908.html>. [Último acceso: 7 Junio 2024].
- [2] INEC, «Visualizador del Registro Estadístico de Empresas,» INEC, Quito, 2023.
- [3] C. W. Matos Ortiz, *RELACIÓN ENTRE INNOVACIÓN TECNOLÓGICA Y DESEMPEÑO ORGANIZACIONAL DE LAS MYPES DE FABRICACIÓN DE PRODUCTOS PLÁSTICOS*, Lima: Universidad San Ignacio De Loyola, 2020.
- [4] R. Platform, «Robot Platform,» [En línea]. Available: https://www.robotplatform.com/knowledge/Classification_of_Robots/Types_of_wheeled_robots.html. [Último acceso: 12 Junio 2024].
- [5] E. Robotics, «EDS Robotics,» 9 Noviembre 2021. [En línea]. Available: <https://www.edsrobotics.com/blog/robots-autonomos-que-son/>. [Último acceso: 11 Junio 2024].
- [6] K. Robotics, «Keenon,» [En línea]. Available: <https://www.keenon.com/en/product/T10/index.html>. [Último acceso: 11 Junio 2024].
- [7] I. Hernandez Robles, R. A. Arroyo Pérez, J. M. Lozano, J. Martínez Patiño y H. J. Estrada Garcia, «Análisis y Simulación de Motores de Imanes Permanentes para Aplicaciones a Vehículos Eléctricos,» *Identidad Energética*, vol. II, nº 1, pp. 44-48, 2019.

- [8] S. A. K. H. & M. A. C. Bharatiraja, «Investigation on Simultaneous Operation of Two Hub BLDC Motors for Electric Vehicles,» *淡江理工學刊*, vol. XXIV, n° 2, pp. 233-238, 2021.
- [9] Parallax, *6.5" Hub Motor with Encoder*.
- [10] R. Suárez, J. Rosell, M. Vinagre, F. Cortes, A. Ansuategui, I. Maurtua, D. Martín, A. Guash, J. Azpiazu, D. Serranog y N. García, *Robot Operating System (ROS)*, Asociación Española de Robótica y Automatización, 2022.
- [11] Pudu Robotics, «Pudu Robotics Solution,» 2023. [En línea]. Available: <https://www.pudurobotics.com/es/solutions/Smart%20Catering#section-2>.
- [12] Pudu Robotics, «BellaBot: Un robot de entrega innovador,» 2023. [En línea]. Available: <https://www.pudurobotics.com/es/products/bellabot>. [Último acceso: 06 Junio 2024].
- [13] L. Keenon Robotics Co., «Keenon Robotics,» 10 Mayo 2022. [En línea]. Available: <https://ftp3.syscom.mx/usuarios/amesta/keenon/usermanualT8.pdf>. [Último acceso: 2 Junio 2024].
- [14] Parallax, «Parallax,» [En línea]. Available: <https://www.parallax.com/product/p2-edge-module-with-32mb-ram/>. [Último acceso: 15 Julio 2024].
- [15] Parallax, «Universal Motor Driver P2 Add-on Board,» [En línea]. Available: <https://www.parallax.com/product/universal-motor-driver-p2-add-on-board/>. [Último acceso: 15 Julio 2024].

- [16] U. Electronics, «ESP32 DEVKIT V1 30 Pines USB-C/MicroUSB,» [En línea]. Available: <https://uelectronics.com/producto/esp32-devkit-v1-30-pines-usb-c-microusb/>. [Último acceso: 16 Julio 2024].
- [17] SLAMTEC, «RPLIDAR-A1 360°Laser Range Scanner,» [En línea]. Available: <https://www.slamtec.ai/product/slamtec-rplidar-a1/>. [Último acceso: 17 Julio 2024].
- [18] I. RealSense™, «Intel® RealSense™ Depth Camera D435i,» [En línea]. Available: <https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435i.html>. [Último acceso: 17 Julio 2024].
- [19] NVIDIA, «Kit para Desarrollador Jetson Nano,» [En línea]. Available: <https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/product-development/>. [Último acceso: 18 Julio 2024].
- [20] E. R. Sásig, «Roboticos,» 10 Enero 2021. [En línea]. Available: <https://roboticoss.com/modelo-cinematico-y-simulacion-con-python-robot-movil-diferencial/#:~:text=Un%20robot%20con%20tracción%20diferencial,instaladas%20en%20un%20único%20eje..> [Último acceso: 10 Junio 2024].

APÉNDICES

APÉNDICE A

Funcionamiento del robot – Navegación

<https://youtu.be/OMGgzL66DQI>

<https://youtu.be/vDIkFUOib80>

APÉNDICE B

Programación de microcontroladores

Figura B.1

Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Primera Parte

```

CON { timing }

CLK_FREQ = 270_000_000           ' system freq as a constant
_clkfreq = CLK_FREQ             ' set system clock

CON { fixed io pins }
RX1      = 63
TX1      = 62
{ 0 }
SF_CS    = 61
SF_CK    = 60
SF_SDO   = 59 { 0 }
SF_SDI   = 58 { I }
BAUD_RATE = 57600

#0, DDU_Unknown, DDU_MM, DDU_CM, DDU_IN, DDU_FT, DDU_M, DDU_KM, DDU_MI

#0, DRU_Unknown, DRU_HALL_TICKS, DRU_DEGREES, DRU_ROTATIONS

#0, DTU_Unknown, DTU_MILLISEC, DTU_SEC

VAR

long rx_buffer
long rx2_buffer
long rx3_buffer
long rx4_buffer
long rx5_buffer
long rx6_buffer
long vel_izq
long vel_der

OBJ { our Drive Subsystem }

user      : "isp_bldc_motor_userconfig"   ' driver configuration
wheels    : "isp_steering_2wheel"        ' the dual-drive BLDC motors
serial    : "jm_fullduplexserial"

```

Figura B.2

Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Segunda Parte

```

PUB main() | basePinLt, basePinRt, voltage, motor, detectModeLt, detectModeRt
serial.start(60, 61, 0, BAUD_RATE) 'Inicializar UART con ESP32
contador:= 0
basePinLt := wheels.validBasePinForChoice(user.LEFT_MOTOR_BASE)
basePinRt := wheels.validBasePinForChoice(user.RIGHT_MOTOR_BASE)
detectModeLt := wheels.validDetectModeForChoice(user.LEFT_BOARD_TYPE)
detectModeRt := wheels.validDetectModeForChoice(user.RIGHT_BOARD_TYPE)
voltage := wheels.validVoltageForChoice(user.DRIVE_VOLTAGE)
motor := wheels.validMotorForChoice(user.MOTOR_TYPE)

if basePinLt <> wheels.INVALID_PIN_BASE and basePinRt <> wheels.INVALID_PIN_BASE and voltage <> wheels.INVALID_VOLTAGE and
wheels.start(basePinLt, basePinRt, voltage, detectModeLt, detectModeRt) 'Inicialización de motores Hub y sensores Hall
wheels.holdAtStop(false)
wheels.setMaxSpeed(14) 'Seteo velocidad máxima 14% (0.3 m/s)
waitUntilMotorReady()
vel_izq := 0
vel_der := 0
serial.rxfllush()
repeat
'Manejo de cadenas de caracteres para extraer valores de velocidades discriminando positivos de negativos
rx_buffer := serial.rxcheck()
rx2_buffer := serial.rxcheck()
rx3_buffer := serial.rxcheck()
rx4_buffer := serial.rxcheck()
rx5_buffer := serial.rxcheck()
rx6_buffer := serial.rxcheck()
if rx_buffer <> "+" and rx_buffer <> "-"
serial.rxfllush()
if rx_buffer == "-"
if rx2_buffer <> -1
if rx3_buffer <> -1
vel_izq := -(rx2_buffer-48)*10 -(rx3_buffer-48)
else
vel_izq := -(rx2_buffer-48)
if rx_buffer == "+"
if rx2_buffer <> -1

```

Figura B.3*Programación Parallax Propeller 2 (P2) Edge – Lenguaje: Spin 2 – Tercera Parte*

```
    if rx4_buffer == "-"
    if rx5_buffer <> -1
    if rx6_buffer <> -1
        vel_der := -(rx5_buffer-48)*10 -(rx6_buffer-48)
    else
        vel_der := -(rx5_buffer-48)
    if rx4_buffer == "+"
    if rx2_buffer <> -1
    if rx3_buffer <> -1
        vel_der := (rx5_buffer-48)*10 +(rx6_buffer-48)
    else
        vel_izq := (rx5_buffer-48)
    'Envio de velocidades hacia los motores (En porcentajes)
    wheels.driveAtPower(vel_izq,vel_der)
    waitms(200)
else
    debug("* ERROR user configuration NOT valid!")

PRI waitUntilMotorReady()
if wheels.isReady() == false
    debug("* wait motors ready...")
    repeat
        if wheels.isReady()
            quit
        else
            waitms(2)
    debug("* Motors ready, let's drive!")
```

Figura B.3

Programación ESP32 – Lenguaje: C++ – Primera Parte

```
#include <Arduino.h>
#include <ros.h>
#include <std_msgs/String.h>
#include <std_msgs/Int16.h>
#include <std_msgs/Float32.h>
#include <geometry_msgs/Twist.h>

float vel_lineal=0;
float vel_angular=0;
float vel_izq=0;
float vel_der=0;
int velIzq=0;
int velDer=0;
/* Parámetros del Robot*/
float wheel_r=8.255/100; //8.255 cm de Radio
float wheel_sep=0.3; // 30 cm de Separación de ruedas

String wheel_right_speed;
String wheel_left_speed;
float velocidad = 0;

/* Inicialización de Nodo ROS */
ros::NodeHandle nh;
```

Figura B.4

Programación ESP32 – Lenguaje: C++ – Segunda Parte

```
/* Callback del mensaje del suscriptor*/  
void messageCb( const geometry_msgs::Twist& msg){  
    vel_angular = msg.angular.z;  
    vel_lineal = msg.linear.x;  
    velocidad = vel_angular * wheel_sep;  
    vel_der = vel_lineal + velocidad/2;  
    vel_izq = vel_lineal - velocidad/2;  
    velIzq = int(vel_izq*(99)/0.20);  
    velDer = int(vel_der*(99)/0.20);  
}  
  
/*Suscriptor hacia topic cmd_vel*/  
ros::Subscriber<geometry_msgs::Twist> cmd_vel_sub("cmd_vel", &messageCb );  
  
void setup()  
{  
    Serial.begin(57600);  
    Serial2.begin(57600);  
    nh.initNode();  
    nh.subscribe(cmd_vel_sub);  
}
```

Figura B.5

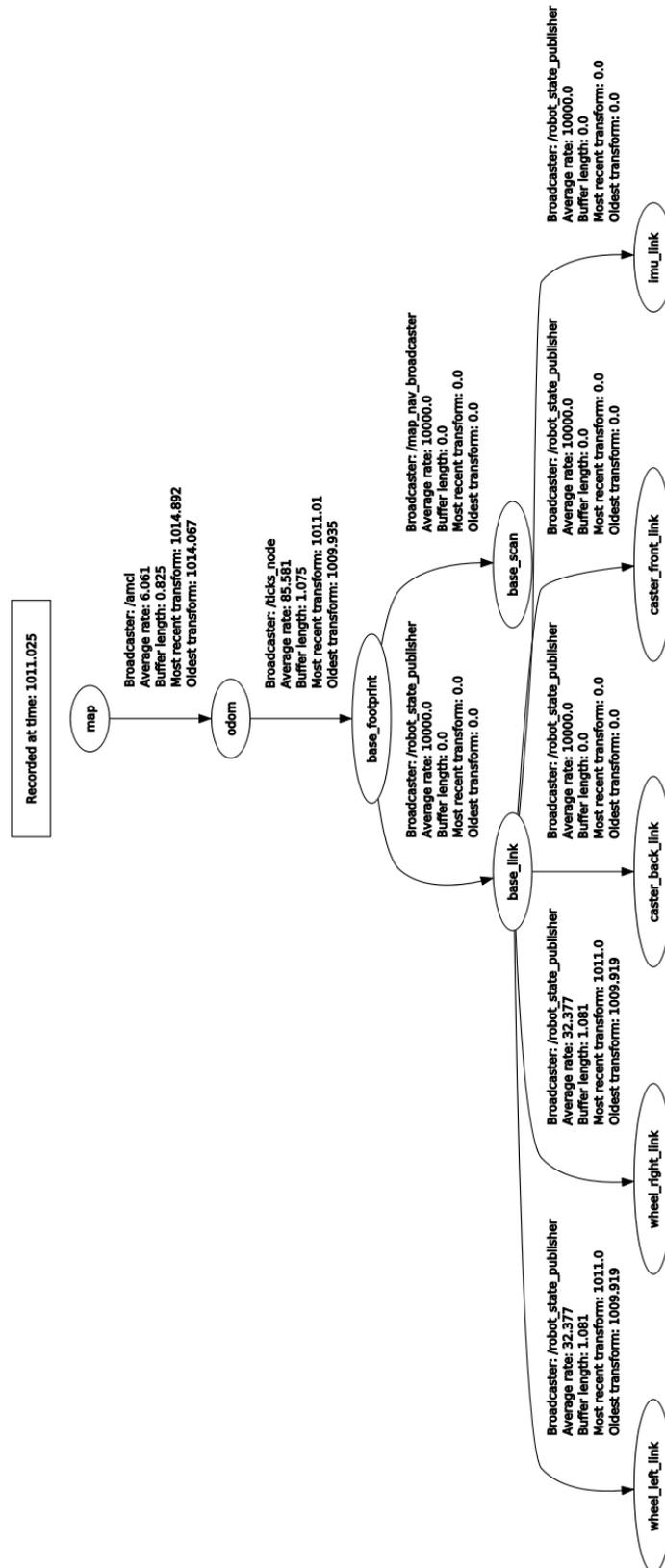
Programación ESP32 – Lenguaje: C++ – Tercera Parte

```
void loop()
{
  /* Estructuración de cadena de caracteres hacia Parallax P2 Edge*/
  if(velDer>=0 && velDer<10){
    wheel_right_speed="+0"+String(velDer);
  }
  else if(velDer<0 && velDer>-10){
    int vel = velDer * -1;
    wheel_right_speed="-0"+String(vel);
  }
  else if(velDer>=10){
    wheel_right_speed="+"+String(velDer);
  }
  else if(velDer<=-10){
    wheel_right_speed=String(velDer);
  }

  if(velIzq>=0 && velIzq<10){
    wheel_left_speed="+0"+String(velIzq);
  }
  else if(velIzq<0 && velIzq>-10){
    int vel2 = velIzq * -1;
    wheel_left_speed="-0"+String(vel2);
  }
  else if(velIzq>=10){
    wheel_left_speed="+"+String(velIzq);
  }
  else if(velIzq<=-10){
    wheel_left_speed=String(velIzq);
  }
  // Mensaje hacia Parallax P2 Edge con formato ±XX±XX
  Serial2.println(wheel_left_speed+wheel_right_speed);
  nh.spinOnce();
  delay(200);
}
```


Figura C.3

Estructura de comunicación – Árbol de Transformadas de los Marcos de Referencia



APÉNDICE D

Cálculo de velocidades para motores HUB – Parallax

Para el control de las velocidades de los motores HUB inicialmente se requería establecer un límite de potencia máximo que llamaremos Límite Primario. El límite primario tomaba valores del 0 al 100% de potencia máximo.

Tras varios análisis del comportamiento de los motores a máxima potencia con y sin carga, se llegó a la conclusión que 100% de potencia establecía un rango de velocidades entre 0 m/s y 1.42 m/s. Por lo que, según los requerimientos, se limitó esta velocidad a 0.20 m/s como velocidad máxima.

$$\text{Limite Primario} = \frac{0.20 \frac{m}{s}}{1.42 \frac{m}{s}} = 0.1408 \approx 0.14 = 14\% \quad (\text{D.1})$$

Ese 14% representaba que el rango de velocidades de los motores HUB podría tomar valores entre 0 y 0.30 m/s, mismo que se asignaba a través de un nuevo porcentaje al que se denominará como Velocidad Actual, la Velocidad Actual puede tomar valores entre 0 y 100% de la velocidad máxima establecida con el Limite Primario. Por ejemplo, asignar una velocidad actual de 100% correspondía a 0.20 m/s, una velocidad actual de 50% correspondía a 0.10 m/s y así de manera lineal.

Las velocidades enviadas desde ROS utilizaban como intermediario al ESP32 para enviar las velocidades hacia el Parallax P2 Edge que realizaría el movimiento de los motores. Para ello, el ESP32 tenía la tarea de convertir estas velocidades en porcentajes que Parallax P2 Edge pudiera comprender.

La operación que el ESP32 realizaba era la siguiente:

$$\textit{Velocidad Rueda} = \textit{Velocidad Sistema} \times \frac{99}{0.20} \quad (\text{D.2})$$

Donde Velocidad Sistema corresponde a la velocidad enviada a través de ROS siguiendo la Ecuación

$$v_R = v + \frac{\omega L}{2} \quad (\text{D.3})$$

Donde v es la velocidad lineal del robot, ω es la velocidad angular del robot y L la separación entre las ruedas.

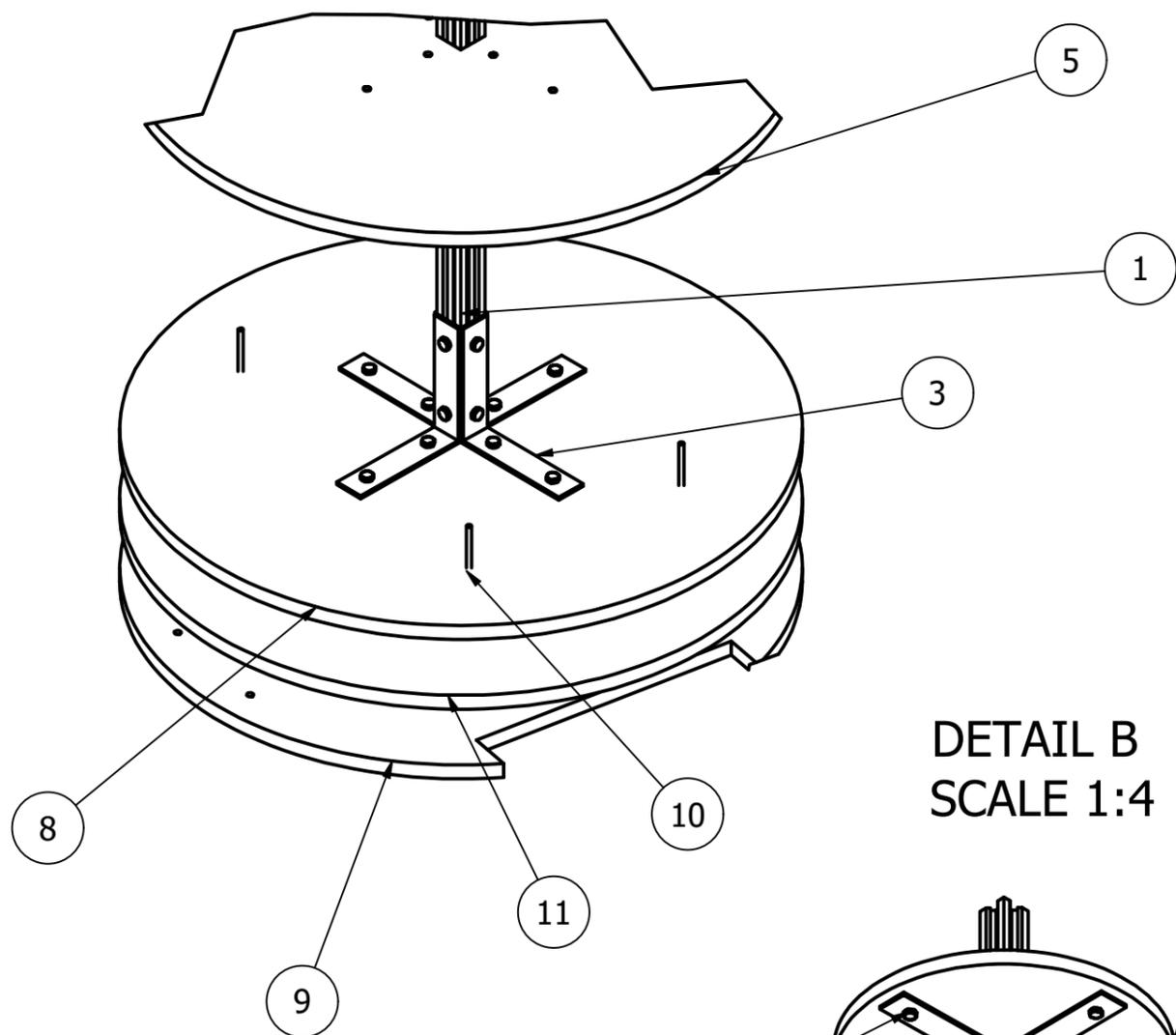
APÉNDICE E

Planos de las piezas del robot

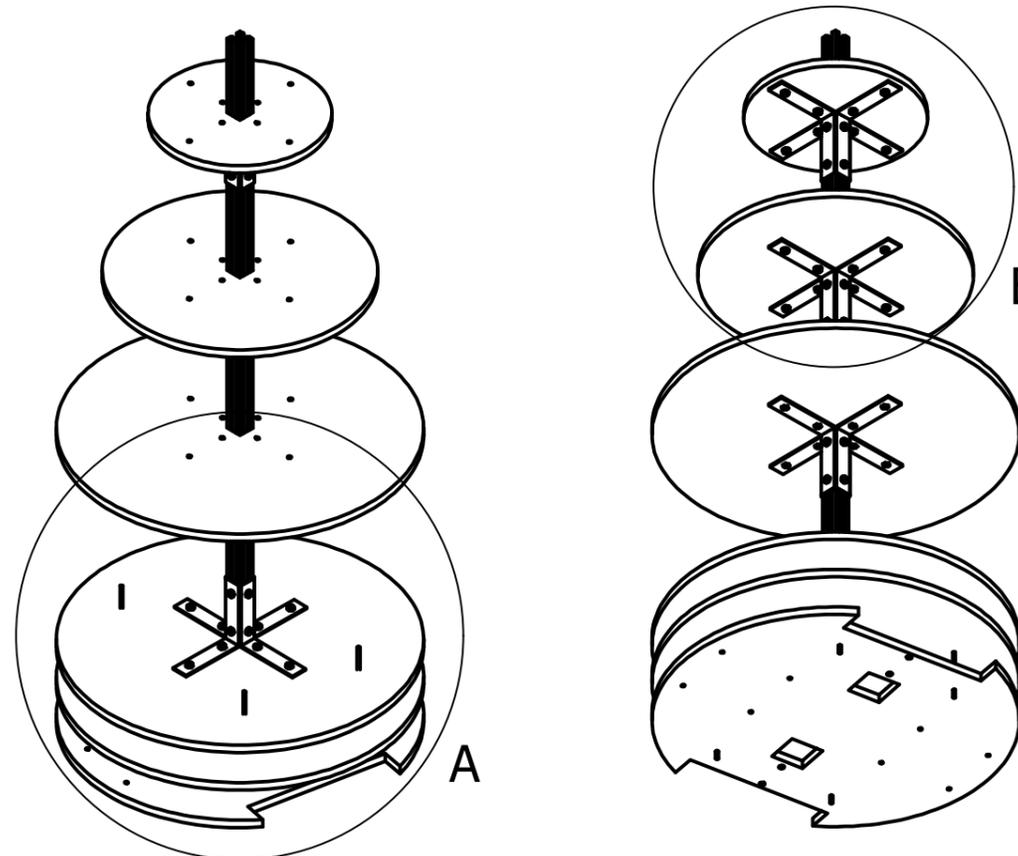
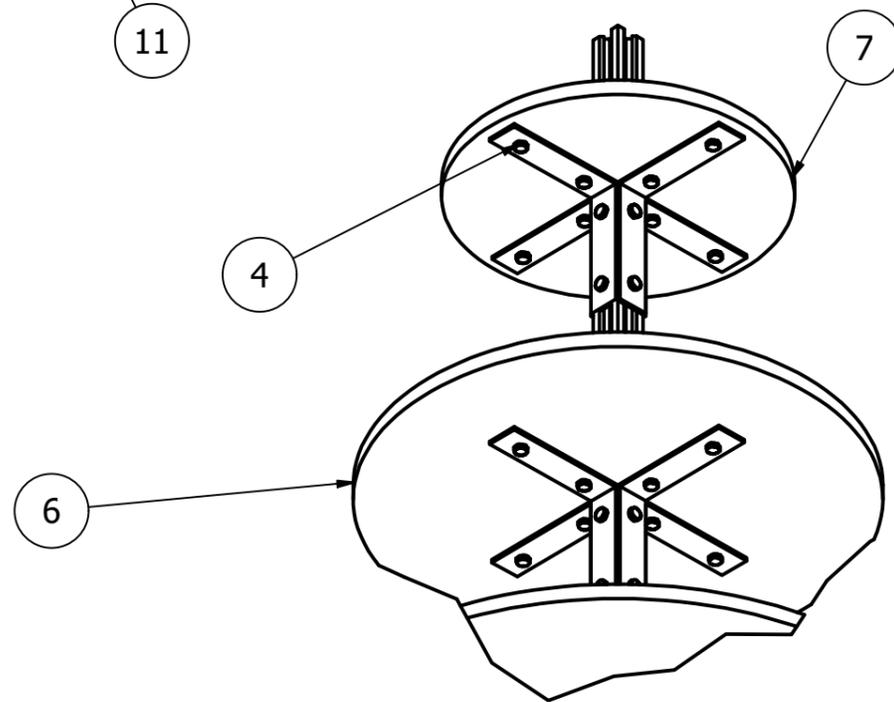
Los planos de la estructura del robot se presentan siguiendo la siguiente lista:

- Plano 1. Cuerpo del robot – Lista de elementos
- Plano 2. Cuerpo del robot – Vista explosionada
- Plano 3. Base inferior del robot – ruedas
- Plano 4. Base media del robot – sensor RPLIDAR A1
- Plano 5. Plataforma grande del cuerpo del robot
- Plano 6. Plataforma mediana y pequeña del cuerpo del robot
- Plano 7. Soporte para el eje central del robot - escuadra

DETAIL A
SCALE 1:4

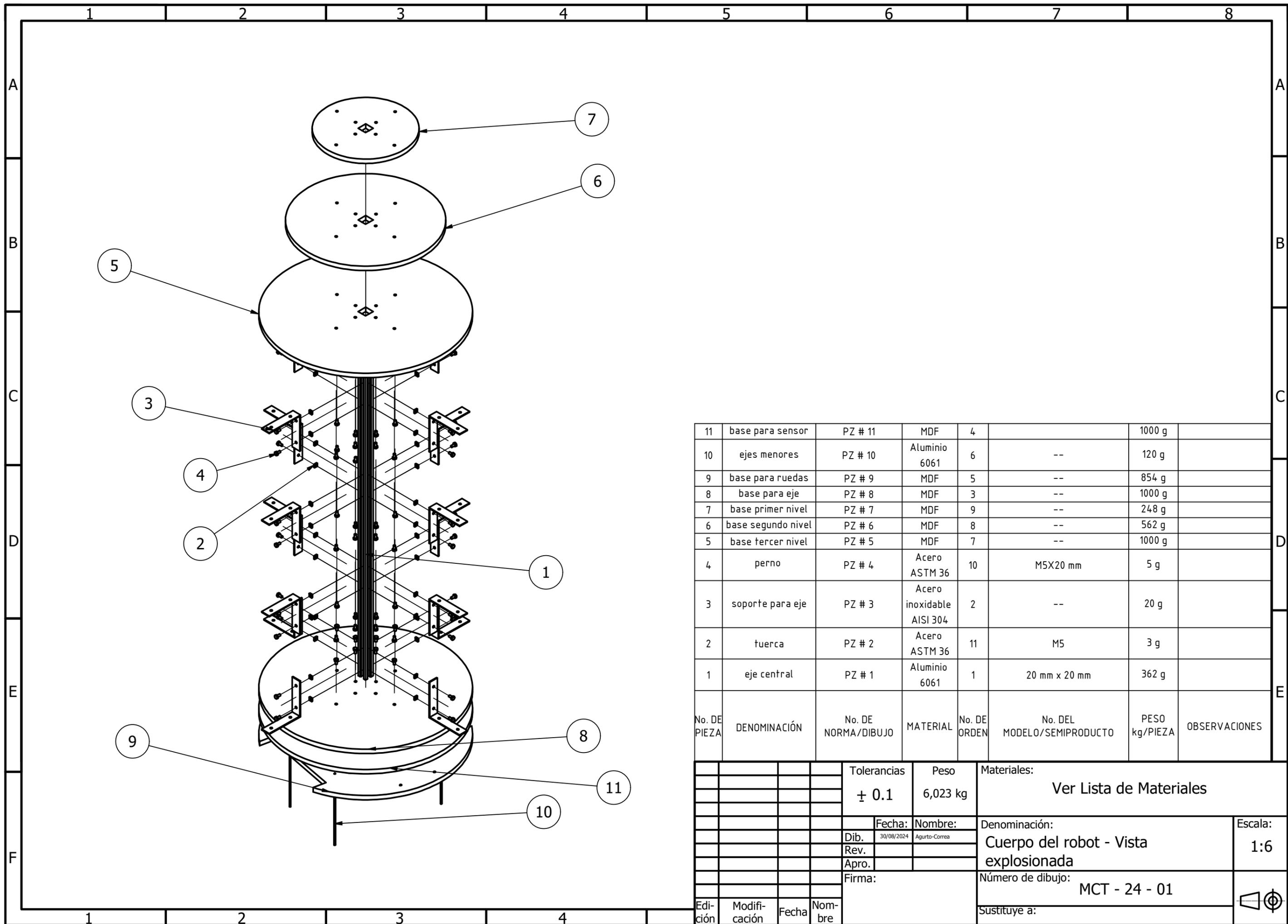


DETAIL B
SCALE 1:4



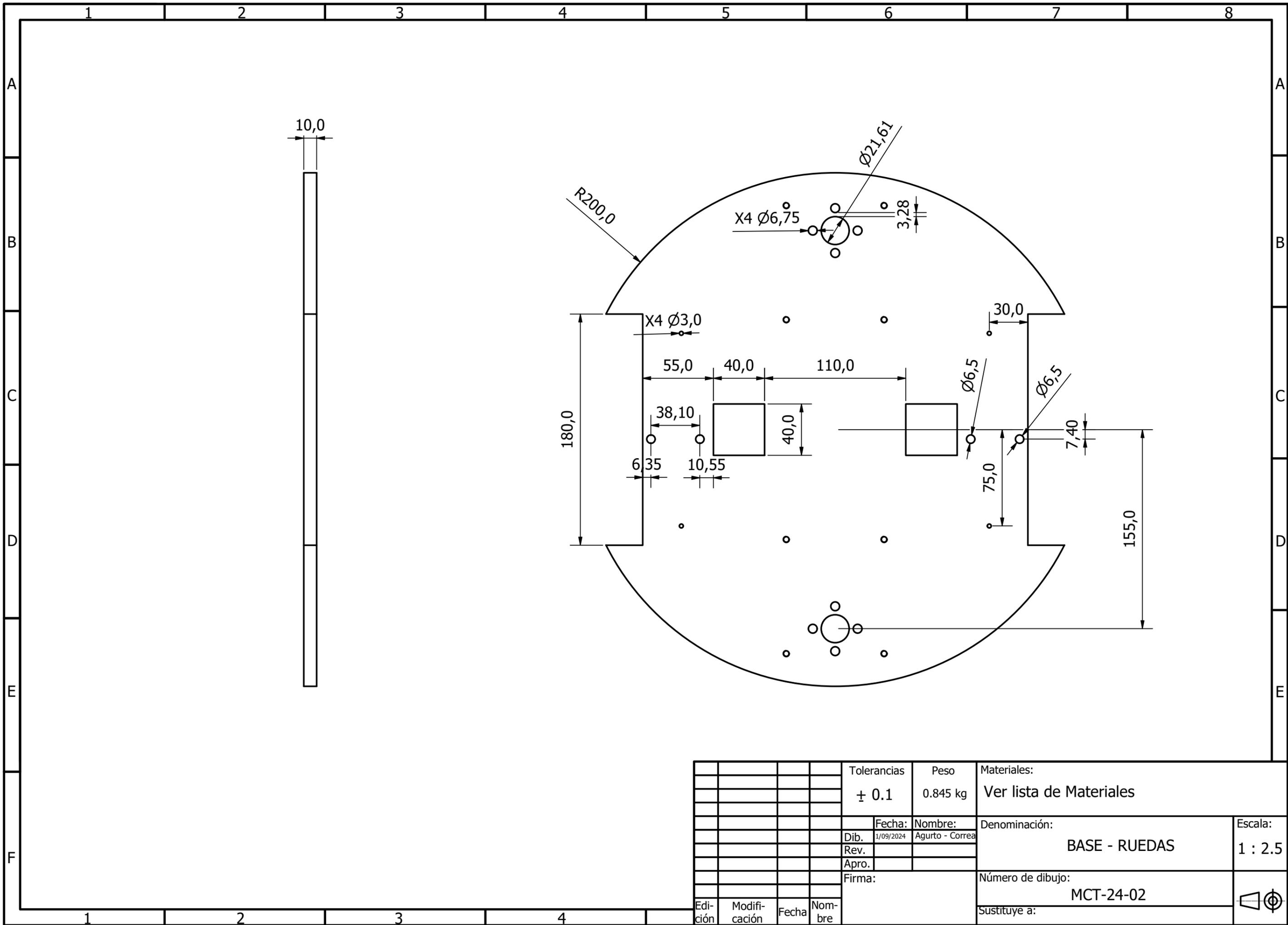
No. DE PIEZA	DENOMINACIÓN	No. DE NORMA/DIBUJO	MATERIAL	No. DE ORDEN	No. DEL MODELO/SEMIPRODUCTO	PESO kg/PIEZA	OBSERVACIONES
11	base para sensor	PZ # 11	MDF	4		1000 g	
10	ejes menores	PZ # 10	Aluminio 6061	6	--	120 g	
9	base para ruedas	PZ # 9	MDF	5	--	854 g	
8	base para eje	PZ # 8	MDF	3	--	1000 g	
7	base primer nivel	PZ # 7	MDF	9	--	248 g	
6	base segundo nivel	PZ # 6	MDF	8	--	562 g	
5	base tercer nivel	PZ # 5	MDF	7	--	1000 g	
4	perno	PZ # 4	Acero ASTM 36	10	M5X20 mm	5 g	
3	soporte para eje	PZ # 3	Acero inoxidable AISI 304	2	--	20 g	
2	tuerca	PZ # 2	Acero ASTM 36	11	M5	3 g	
1	eje central	PZ # 1	Aluminio 6061	1	20 mm x 20 mm	362 g	

Tolerancias		Peso		Materiales:			
± 0.1		6,023 kg		Ver Lista de Materiales			
Fecha:		Nombre:		Denominación:		Escala:	
30/08/2024		Agurto-Correa		Cuerpo del robot - Lista de Elementos		1:4	
Dib.		Apro.		Número de dibujo:		Firma:	
				MCT - 24 - 00			
Edición		Modificación		Sustituye a:		Firma:	



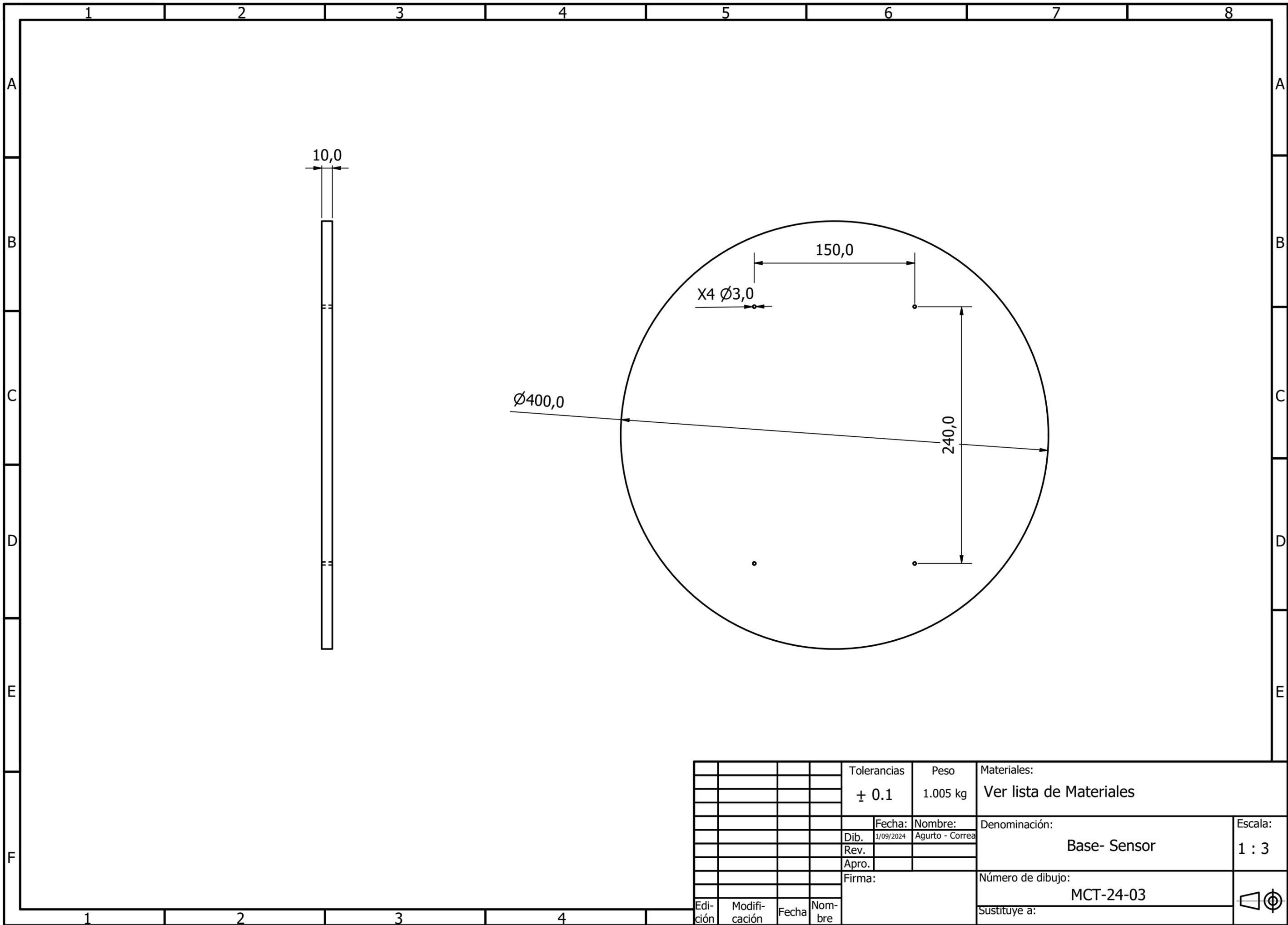
11	base para sensor	PZ # 11	MDF	4		1000 g	
10	ejes menores	PZ # 10	Aluminio 6061	6	--	120 g	
9	base para ruedas	PZ # 9	MDF	5	--	854 g	
8	base para eje	PZ # 8	MDF	3	--	1000 g	
7	base primer nivel	PZ # 7	MDF	9	--	248 g	
6	base segundo nivel	PZ # 6	MDF	8	--	562 g	
5	base tercer nivel	PZ # 5	MDF	7	--	1000 g	
4	perno	PZ # 4	Acero ASTM 36	10	M5X20 mm	5 g	
3	soporte para eje	PZ # 3	Acero inoxidable AISI 304	2	--	20 g	
2	tuerca	PZ # 2	Acero ASTM 36	11	M5	3 g	
1	eje central	PZ # 1	Aluminio 6061	1	20 mm x 20 mm	362 g	
No. DE PIEZA	DENOMINACIÓN	No. DE NORMA/DIBUJO	MATERIAL	No. DE ORDEN	No. DEL MODELO/SEMIPRODUCTO	PESO kg/PIEZA	OBSERVACIONES

Tolerancias		Peso		Materiales:			
± 0.1		6,023 kg		Ver Lista de Materiales			
Fecha:		Nombre:		Denominación:			Escala:
30/08/2024		Agurto-Correa		Cuerpo del robot - Vista explosionada			1:6
Rev.				Número de dibujo:			
Apro.				MCT - 24 - 01			
Firma:				Sustituye a:			
Edición	Modificación	Fecha	Nombre				

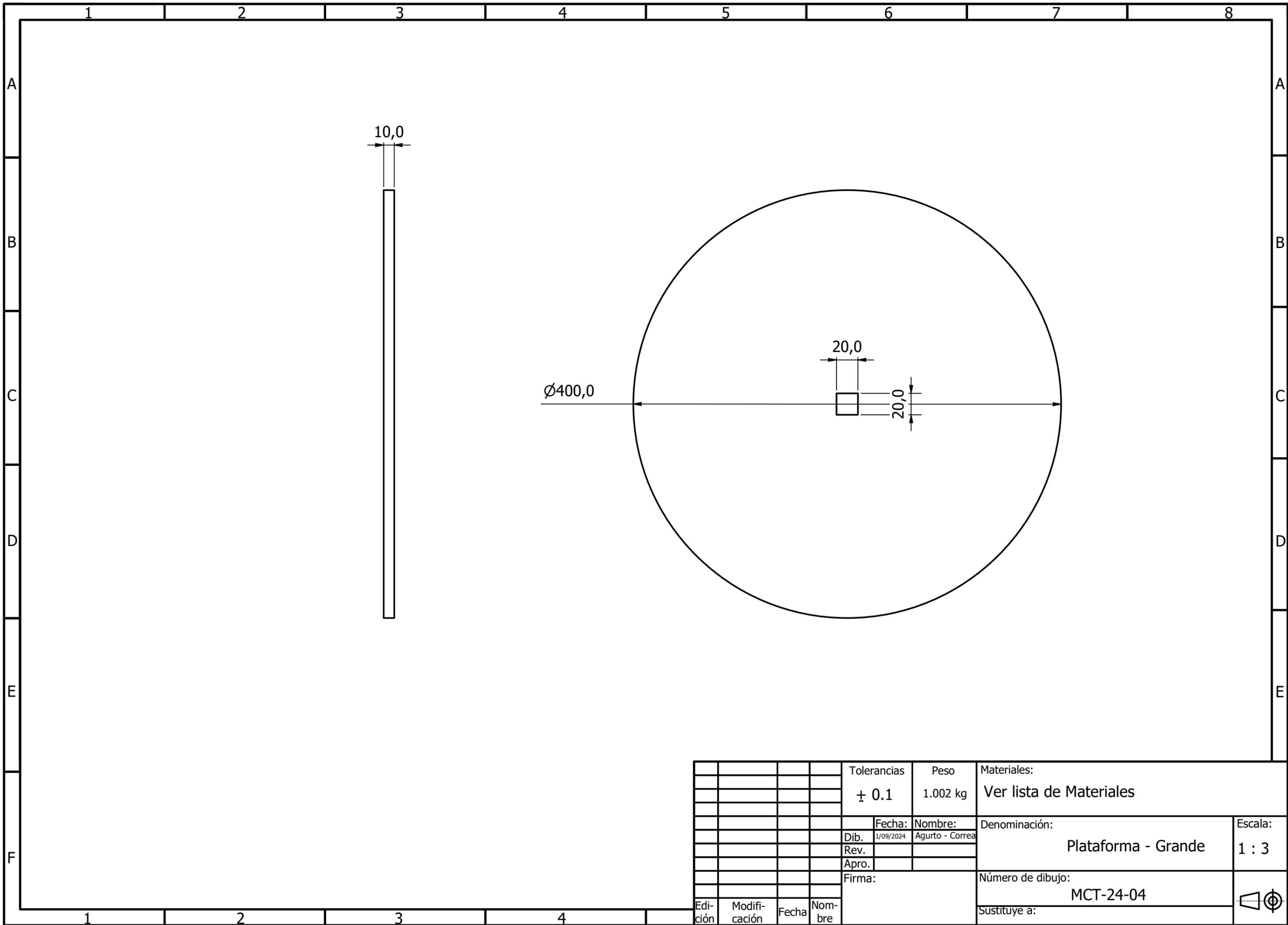


				Tolerancias	Peso	Materiales:	
				± 0.1	0.845 kg	Ver lista de Materiales	
					Fecha:	Nombre:	Denominación:
				Dib.	1/09/2024	Agurto - Correa	BASE - RUEDAS
				Rev.			
				Apro.			
				Firma:			Número de dibujo:
							MCT-24-02
							Sustituye a:
Edi- ción	Modifi- cación	Fecha	Nom- bre				



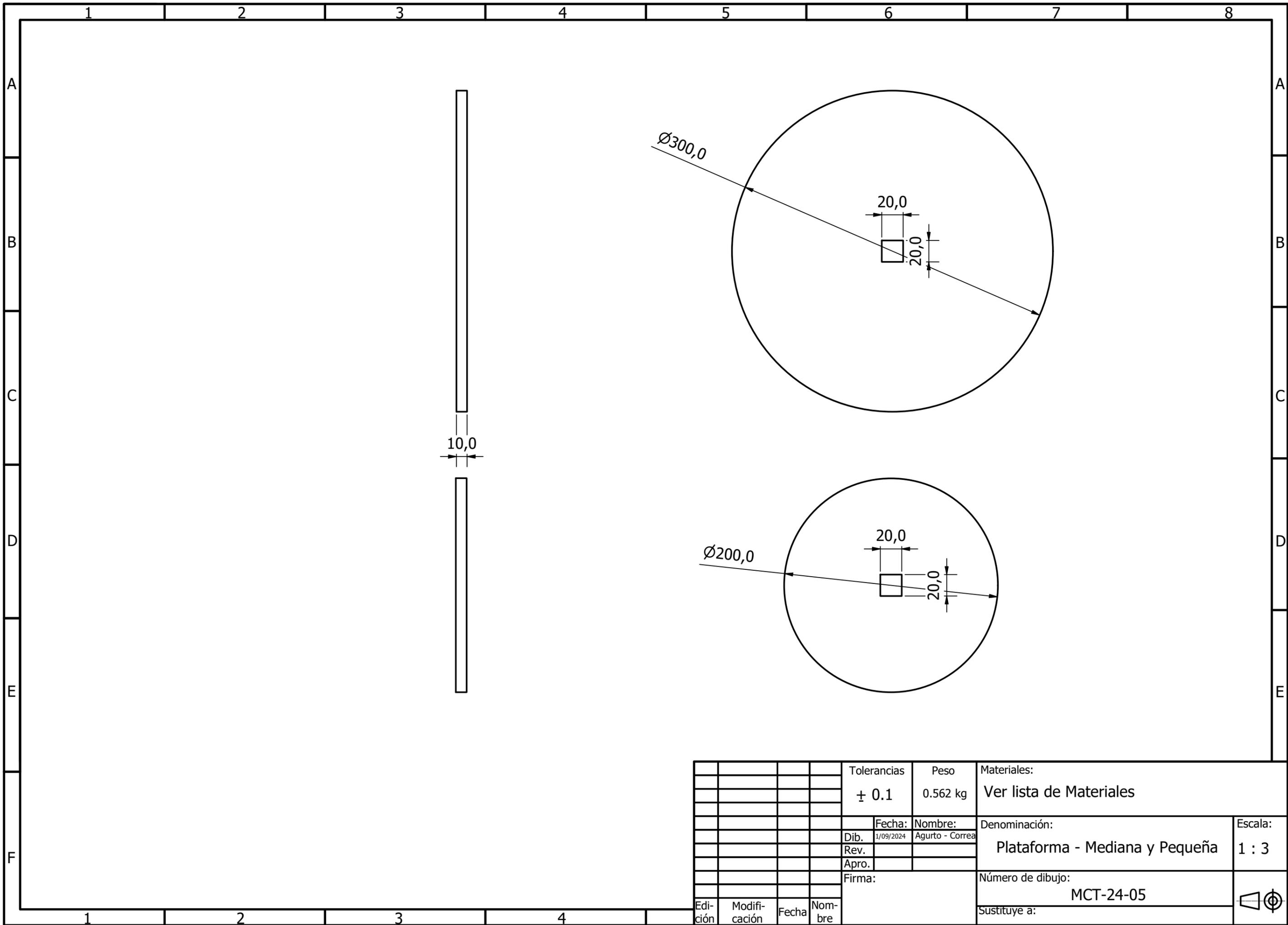


				Tolerancias	Peso	Materiales:	
				± 0.1	1.005 kg	Ver lista de Materiales	
					Fecha:	Nombre:	Denominación:
				Dib.	1/09/2024	Agurto - Correa	Base- Sensor
				Rev.			1 : 3
				Apro.			
				Firma:			Número de dibujo:
							MCT-24-03
							Sustituye a:
Edi- ción	Modifi- cación	Fecha	Nom- bre				



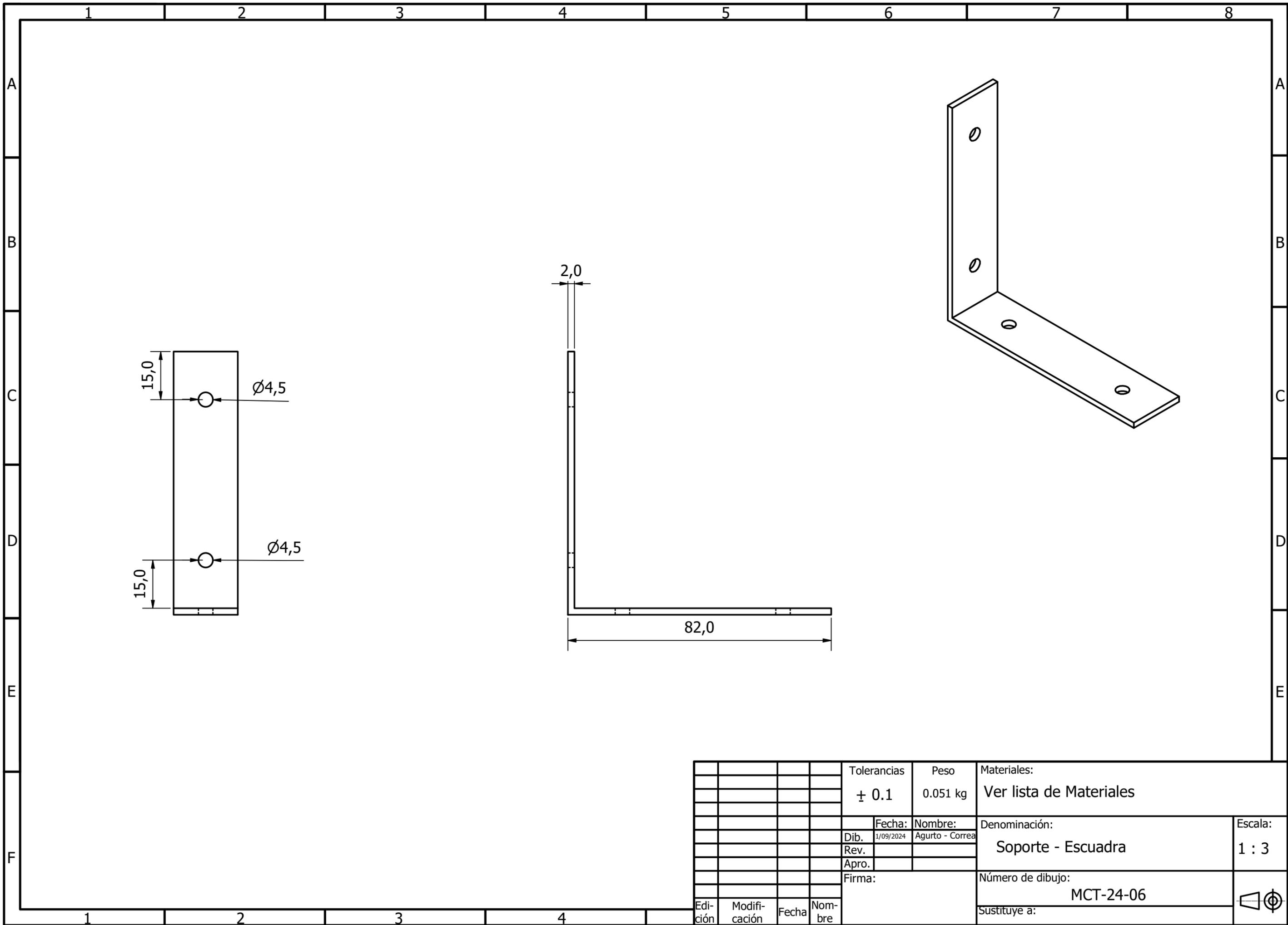
				Tolerancias	Peso	Materiales:	
				± 0.1	1.002 kg	Ver lista de Materiales	
					Fecha:	Nombre:	Denominación:
				Dib.	1/09/2024	Agurto - Correa	Plataforma - Grande
				Rev.			
				Apro.			Número de dibujo: MCT-24-04
				Firma:			
Edi- ción	Modifi- cación	Fecha	Nom- bre				Sustituye a:





				Tolerancias	Peso	Materiales:	
				± 0.1	0.562 kg	Ver lista de Materiales	
					Fecha:	Nombre:	Denominación:
					1/09/2024	Agurto - Correa	Plataforma - Mediana y Pequeña
							Escala:
							1 : 3
					Firma:	Número de dibujo:	
						MCT-24-05	
Edi- ción	Modifi- cación	Fecha	Nom- bre			Sustituye a:	





				Tolerancias	Peso	Materiales:	
				± 0.1	0.051 kg	Ver lista de Materiales	
					Fecha:	Nombre:	Denominación:
				Dib.	1/09/2024	Agurto - Correa	Soporte - Escuadra
				Rev.			
				Apro.			
				Firma:			Número de dibujo:
							MCT-24-06
							Sustituye a:
Edi- ción	Modifi- cación	Fecha	Nom- bre				