

DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PARA DESINFECCIÓN DE
SUPERFICIES CON LUZ ULTRAVIOLETA

JAVIER RANGEL CASTRO

JUAN DIEGO VILLAMIZAR GARCÍA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2021

DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PARA DESINFECCIÓN DE
SUPERFICIES CON LUZ ULTRAVIOLETA

JAVIER RANGEL CASTRO

JUAN DIEGO VILLAMIZAR GARCÍA

Trabajo de grado para optar por el título de Ingeniero Mecánico

DIRECTOR

JORGE ENRIQUE MENESES FLOREZ

Magister en Ingeniería Mecánica

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2021

A mis padres Javier y Mónica y mi tía Claudia

Por apoyarme y darme la oportunidad de formarme como profesional y persona.

A mi nono Teo

Porque lo que soy es todo gracias a él.

A mis hermanos Juana, María Fernanda y Juan José

Por todo el amor y el cariño que me dan.

A Alejandra

Por ser parte de mi vida.

Javier Rangel

AGRADECIMIENTOS

A Mauricio Guadrón, por toda la motivación, ideas y apoyo que nos brindó a través de los años en el mundo de la robótica.

A Luis Gabriel Sandoval, por mostrarnos los primeros pasos y ser un apoyo incondicional en nuestro desarrollo como personas día a día.

Al director de proyecto Ing. Jorge Meneses, por motivarnos a mejorar cada vez más nuestros proyectos.

TABLA DE CONTENIDO

INTRODUCCION	14
1 PRELIMINARES	16
1.1 OBJETIVOS	16
1.1.1 Objetivo General	16
1.1.2 Objetivos específicos	16
1.2 USOS DE LA LUZ ULTRAVIOLETA EN LA DESINFECCIÓN.....	17
1.3 IDENTIFICACIÓN DEL PROBLEMA	23
1.3.1 Requerimientos del sistema para la limpieza	24
2 DESCRIPCIÓN GENERAL DEL PROYECTO	27
2.1 SISTEMA MECÁNICO ESTRUCTURAL.....	27
2.2 SISTEMA DE HARDWARE Y ELECTRÓNICA.....	30
2.2.1 Parte operativa.....	30
2.2.2 Parte de control.....	31
2.2.3 Interfaz hombre máquina	31
2.2.4 Programación y debug remotos	32
2.3 SISTEMA DE AUTONOMÍA Y SOFTWARE	34
3 DESCRIPCION DE LOS SISTEMAS	37
3.1 SISTEMA MECÁNICO ESTRUCTURAL.....	37
3.1.1 Base.....	38
3.1.2 Sistema de suspensión	44
3.1.3 Estructura vertical	46
3.1.4 Lámparas desinfectantes	48
3.1.5 Variador de altura y posición de la lámpara de las mesas	50
3.1.6 Interfaz	54
3.1.7 Sensores exteriores	55
3.1.8 Exteriores.....	55
3.2 SISTEMA DE HARDWARE Y ELECTRÓNICA.....	57
3.2.1 Batería	57

3.2.2	Circuito de carga	61
3.2.3	Circuito principal.....	62
3.2.4	Los controladores.....	63
3.3	SISTEMA DE AUTONOMÍA Y SOFTWARE	66
3.3.1	Nodo Arduino mega	69
3.3.2	Odometría	70
3.3.3	Odometría EKF (IMU)	71
3.3.4	LiDAR.....	72
3.3.5	Mapa	73
3.3.6	AMCL.....	74
3.3.7	Extractor de objetos	77
3.3.8	Posición inicial.....	77
3.3.9	Detector de personas.....	81
3.3.10	Planeador	82
3.3.11	Navegación	82
3.3.12	Base	85
3.3.13	Control de velocidad.....	87
3.3.14	Interfaz, app y status	87
3.3.15	Proceso de mapeo	89
4	PRUEBAS	91
4.1	ETAPA DE PRUEBA INICIAL	91
4.2	ETAPA DE VALIDACIÓN DE SOFTWARE Y HARDWARE	92
4.3	ETAPA DE PRUEBAS EN CAMPO REAL	93
4.3.1	Primera visita:	94
4.3.2	Segunda visita:.....	95
5	MANUAL DE OPERACIÓN	98
5.1	CARGA DE LA BATERÍA.....	98
5.2	DESCARGA Y USO DE LA APP	98
5.3	OPERACIÓN.....	99
5.4	MANTENIMIENTO:.....	102
6	COSTOS	103

7	CONCLUSIONES	108
8	RECOMENDACIONES	110
	BIBLIOGRAFIA	111
	ANEXOS	114

LISTA DE FIGURAS

Figura 1. Lámpara ultravioleta.....	14
Figura 2. Filtro UVc para agua.	18
Figura 3. Sistema de aire acondicionado con purificación UVc.....	19
Figura 4. Caja de desinfección.....	20
Figura 5. Módulos estáticos de desinfección de habitaciones.	21
Figura 6. Módulos de desinfección operado manualmente.....	22
Figura 7. Robot de desinfección autónomo comercial.	23
Figura 8. Prototipo de robot desinfectante.	27
Figura 9. Ajuste vertical y posición de guardado del brazo de los bombillos.	28
Figura 10. Foto del prototipo con los bombillos UVc encendidos.....	29
Figura 11. Estructura del sistema automatizado.	33
Figura 12. Diagrama general del software.	36
Figura 13. Perfil de aluminio tipo estrella 2040 y 2020.....	37
Figura 14. Vista inferior de la base.....	39
Figura 15. Prototipo de la base con apoyo esféricas sin suspensión.....	40
Figura 16. Prototipo de la base usando las ruedas omnidireccionales.	43
Figura 17. Diseño de sistema de suspensión.	44
Figura 18. Sistema de suspensión instalado.....	46
Figura 19. Montaje de la estructura vertical.	47
Figura 20. Estructura vertical.	48
Figura 21. Soporte para bombillo.....	49
Figura 22. Lámpara estática (a) y móvil (b).....	49
Figura 23. Acción del pistón en posición de operación (a) y guardado (b).....	51
Figura 24. Estructura móvil de la lampara.....	52
Figura 25. Interfaz.	54
Figura 26. Vista frontal-izquierda (a) y posterior-derecha (b).....	57
Figura 27. Tester USB para la medición de carga.	58

Figura 28. Esquema del circuito para el cargador.....	61
Figura 29. Esquema del circuito principal de distribución de voltaje.	62
Figura 30. Esquema de la conexión de los controladores.....	64
Figura 31. Esquema del circuito de la interfaz.	65
Figura 32. Esquema del circuito del Arduino Mega en la base.	66
Figura 33. Mapa de los nodos de control del robot.	68
Figura 34. Nube de puntos arrojados por el sensor LiDAR.....	73
Figura 35. Extractor de obstáculos.....	77
Figura 36. Búsqueda de esquinas.....	79
Figura 37. Cálculo de la posición con base a la esquina.	80
Figura 38. Representación de una persona detectada.	82
Figura 39. Proceso del nodo de navegación.....	84
Figura 40. Respuesta a una perturbación en la dirección del robot.	86
Figura 41. Representación en coordenadas x y de la respuesta a la desviación..	87
Figura 42. Primer modelo para pruebas de software iniciales.....	91
Figura 43. Habitación y ruta para la segunda etapa de pruebas.....	92
Figura 44. Robot dentro del salón de clases.....	94
Figura 45. Captura de pantalla de la aplicación móvil.....	99

LISTA DE TABLAS

Tabla 1. Alternativas de motores para la base.	41
Tabla 2. Alternativas de rueda de apoyo.	42
Tabla 3. Alternativas para el subsistema del brazo.	53
Tabla 4. Selección de alternativas de la batería.....	60
Tabla 5. Costos de electrónica general.	104
Tabla 6. Costos de la batería.....	104
Tabla 7. Costos de los controladores	105
Tabla 8. Costos de la sensorica y actuadores.....	105
Tabla 9. Costos de la estructura y componentes mecánicos.	106
Tabla 10. Costos totales.....	107

TABLA DE ANEXOS

ANEXO A. CÁLCULO DE LOS PARÁMETROS TÉCNICOS PARA LA INACTIVACIÓN DEL SARS-COV 19	115
ANEXO B. CÁLCULOS DEL EJE DE LA BASE	130
ANEXO C. PLANOS GENERALES DEL ROBOT.....	135

RESUMEN

TITULO: DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PARA DESINFECCIÓN DE SUPERFICIES CON LUZ ULTRAVIOLETA*

AUTORES: JUAN DIEGO VILLAMIZAR GARCIA Y JAVIER RANGEL CASTRO**

PALABRAS CLAVES: Robot, desinfección de salones, COVID-19, luz ultravioleta, ROS.

DESCRIPCIÓN:

En respuesta a la emergencia sanitaria producto del COVID-19 se diseña un prototipo de robot móvil con el objetivo de automatizar los protocolos de desinfección en los salones de la universidad industrial de Santander en preparación para el eventual retorno a la presencialidad. Este prototipo hace uso de lámparas de luz ultravioleta tipo C ampliamente utilizadas para el tratamiento de agua y aire en espacios cerrados pero que debido al riesgo que representan para los seres humanos apenas comienza a implementarse en otros ambientes con ayuda de robots y sistemas de automatización.

Se realizó entonces el diseño en detalle que involucra la parte mecánica, electrónica y de software basado en los requerimientos correspondientes. La fabricación del prototipo transcurre durante la emergencia sanitaria y por lo tanto se busca que en su mayoría pueda ser construido desde casa con poca ayuda de terceros y con componentes disponibles en el mercado local. Para la capa de software se hace uso de la plataforma ROS que facilita la programación de robots autónomos y se implementó una aplicación móvil para permitir el manejo remoto.

Finalmente, se realizaron las pruebas de operación en los salones de la UIS y se concluyó que el sistema realizado puede cumplir su función de forma efectiva, pero hacen falta varias mejoras tanto en hardware como en software para optimizar su desempeño y confiabilidad.

* Trabajo de grado

** Facultad de ingenierías Físico mecánicas. Escuela de ingeniería Mecánica. Director: Jorge Enrique Meneses Flórez. Magister en Ingeniería mecánica

ABSTRACT

TITLE: DESIGN AND CONSTRUCTION OF A ROBOT FOR DISINFECTION OF SURFACES WITH UV LIGHT *

AUTHORS: JUAN DIEGO VILLAMIZAR GARCIA AND JAVIER RANGEL CASTRO**

KEYWORDS: Robot, rooms disinfection, COVID-19, UV light, ROS.

DESCRIPTION:

In response to the health emergency caused by COVID-19 a mobile robot prototype is designed with the aim of automating the disinfection protocols for the classrooms of the university (UIS) in preparation for the return to the face-to-face classes. This prototype uses type C ultraviolet lights widely used for water and air treatment in closed spaces, but due to the health risk it represents for the human beings it is just beginning to be implemented in different environments using robots and automation systems.

We then proceed to the detailed design phase that involves the mechanical, electronic and software design based on the corresponding requirements. The manufacturing of the prototype is made during the health emergency and therefore aim to be as easy as possible to be made at home with little external help and with local components. The software layer is based in ROS, a specialized platform for autonomous robots and a mobile app is developed to allow remote control of the robot.

Finally, operational tests are carried out in the classrooms of UIS reaching to the conclusion that the current prototype is in fact capable of achieving the objective effectively but still many enhancements need to be made to improve the overall performance and reliability of the system.

* Degree work

** Faculty of Physical Engineering. School of mechanical engineering. Director: Jorge Enrique Meneses Flórez. Magister en Ingeniería mecánica

INTRODUCCION

El tema de la limpieza y desinfección de superficies se ha convertido en una gran problemática durante la emergencia sanitaria debido al COVID-19 y tendrá especial importancia para la reapertura de espacios de alta afluencia de personas como lo son los salones de clase de universidades y colegios.

Los protocolos de bioseguridad para dichos ambientes incluyen limpiezas frecuentes de las superficies que consumen buena cantidad de productos químicos y demandan grandes esfuerzos por parte del personal de limpieza que a su vez se convierte en un gasto económico para las instituciones.

La radiación ultravioleta de onda corta (UV-c) en altas dosis ha sido probada por numerosos estudios como un método eficaz para la inactivación de toda clase de microorganismos incluyendo los coronavirus y ha sido ampliamente utilizada para la esterilización en el campo de la medicina, el tratamiento de agua y aire y la desinfección de espacios cerrados.

Figura 1. Lámpara ultravioleta.



Fuente: Jennings oil and propane company. Disponible en:
<https://www.jenningsoil.com/what-is-a-germicidal-uv-light/>

Sin embargo, la implementación de esta tecnología implica dar solución a las grandes problemáticas que esta acarrea. En primer lugar, el riesgo biológico que la radiación UV-c representa para los seres humanos y los animales, es causante de serias quemaduras, daño ocular y se asocia al cáncer de piel, por lo tanto, es obligatorio implementar dichas aplicaciones de forma autónoma alejada de personas y animales.

En segundo lugar, la posición entre la fuente de radiación y la superficie a desinfectar. Se debe tener en cuenta que la intensidad de la radiación que recibe un punto disminuye fuertemente con la distancia a la fuente, lo cual implica que se debe situar la lámpara, o lámparas, lo más cerca posible de la superficie a desinfectar y en una posición que todas las superficies reciban su adecuada dosis de radiación directa.

1 PRELIMINARES

1.1 OBJETIVOS

1.1.1 Objetivo General: Contribuir con la misión de la escuela de ingeniería mecánica de la universidad industrial de Santander mediante el desarrollo de un sistema de desinfección de superficies empleando luz ultravioleta, fomentando la innovación y el desarrollo tecnológico en los estudiantes y proponiendo una solución innovadora para contribuir a la emergencia sanitaria a raíz del COVID-19.

1.1.2 Objetivos específicos:

- Diseñar y construir un prototipo de un robot funcional que garantice la desinfección de la superficie de mesas y sillas de los salones de clase planos de la UIS, y que cumpla con las siguientes especificaciones de diseño:
 - Base de cuatro ruedas con tracción trasera brindadas por motores 12 voltios que garantice el movimiento en superficies planas donde no intervengan escalones.
 - Ruedas de tracción con goma para la adherencia con el piso.
 - Estructura de la base con dimensiones de 45 x 45 cm máximo.
 - Sistema desplegable de dos bombillos germicidas de 15W con altura ajustable y dos posiciones (guardado y operación) para la parte superior de las mesas.
 - Lámpara fija de dos bombillos de 15W verticales para las sillas y otros elementos del salón.
 - Arquitectura hardware basada en Raspberry pi y Arduino.

- Arquitectura software basada en ROS (Robot Operating System).

- Implementar un software de navegación autónoma en el ambiente de ROS que permita al prototipo encontrar su posición y la de los objetos en el salón y seguir una trayectoria preestablecida para la limpieza con el salón ordenado.

- Elaborar un documento el cual exponga:
 - Uso adecuado del prototipo.
 - Cuidados del prototipo.
 - Datos de operación del prototipo.

1.2 USOS DE LA LUZ ULTRAVIOLETA EN LA DESINFECCIÓN

El uso de luz ultravioleta para desinfección, más específicamente la luz UV de onda corta o UVC ha sido estudiado por décadas y su eficacia en la inactivación de un gran número de microorganismos de diferentes especies incluyendo el virus SARS-CoV-2, está totalmente demostrada y documentada en numerosas fuentes como lo indica la *Illuminating engineering society*¹. Debido a la pandemia, el uso de esta tecnología ha crecido exponencialmente, así como la información disponible al respecto, en el anexo A se realizó una recopilación más a fondo de la bibliografía disponible a la fecha y con base en esta se ejecutó un análisis de los aspectos técnicos necesarios para su implementación en este proyecto.

Se realizó una recopilación de los usos actuales de la desinfección mediante el uso de luz ultravioleta en el que se revisaron aplicaciones actuales, productos en el mercado y sistemas funcionales vigentes y se elabora la siguiente lista.

¹ IES Photobiology Committee. (2020). IES Committee Report: Germicidal Ultraviolet (GUV) – Frequently Asked Questions. Consultado el 27/06/2021 [En línea] disponible en: <https://media.ies.org/docs/standards/IES-CR-2-20-V1-6d.pdf>

- Tratamiento de agua

Uno de los usos más generalizados de esta tecnología es en el tratamiento de agua tanto en el ámbito residencial como en la industria. Se venden filtros donde el agua circula alrededor de un bombillo emisor de radiación UVc para recibir su efecto esterilizante sin la necesidad de agregar químicos al líquido.

Figura 2. Filtro UVc para agua.

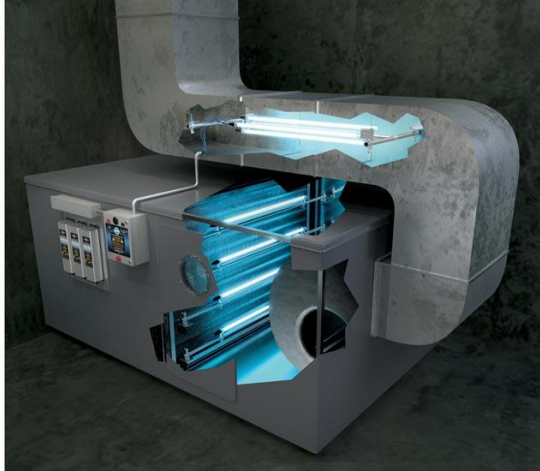


Fuente: ESP Water products. disponible en: <https://www.espwaterproducts.com/viqua-ihs22-e4-integrated-22-gpm-uv-system-with-pre-sediment-pre-carbon-with-lead-reduction-filters/>

- Tratamiento de aire

Similar a los filtros de agua, muchos sistemas de aire acondicionado, especialmente en lugares de alta contaminación como hospitales, hacen uso de filtros con luz ultravioleta para eliminar los microorganismos presentes en el aire de recirculación.

Figura 3. Sistema de aire acondicionado con purificación UVc.



Fuente: achr The News. Disponible en: <https://www.achrnews.com/articles/143318-covid-19-reveals-importance-of-uv-c-in-hvac-industry>

- Cajas para la desinfección de objetos

Uno de los artículos de mayor comercialización a raíz de la pandemia que usan la tecnología UVc son las cajas de desinfección. Consiste simplemente en un compartimiento cerrado con bombillos en su interior en el cual se introducen por unos minutos los objetos que se desea desinfectar y se activa el producto. Se consiguen en diferentes tamaños y precios.

Figura 4. Caja de desinfección.



Fuente: Mommore. Disponible en: <https://www.mommore.com/products/uv-clean-smart-sanitizer-box>

- Módulos estáticos de desinfección de habitaciones.

El objetivo de estos dispositivos es la desinfección total de una habitación incluyendo las superficies y el aire dentro de la misma. En esta categoría se encuentra un gran número de productos con diferentes características y capacidades de desinfección y debido a la pandemia su producción, diversificación y regulación se ha incrementado enormemente. El funcionamiento general consiste en colocar el dispositivo en un punto central de la habitación y programarlo para permanecer encendido por un tiempo cuando la habitación se encuentre desocupada. Muchos de estos productos incluyen sensores de movimiento para apagarse si una persona se acerca.

La principal desventaja de estos dispositivos es que, debido a la naturaleza de la luz, la acción desinfectante disminuye exponencialmente con la distancia de la fuente y por lo tanto los lugares alejados del módulo o los que no reciben radiación directa no suelen llegar a los niveles aceptables de desinfección y así mismo se requiere alta potencia y mucho tiempo para alcanzar el objetivo.

Figura 5. Módulos estáticos de desinfección de habitaciones.



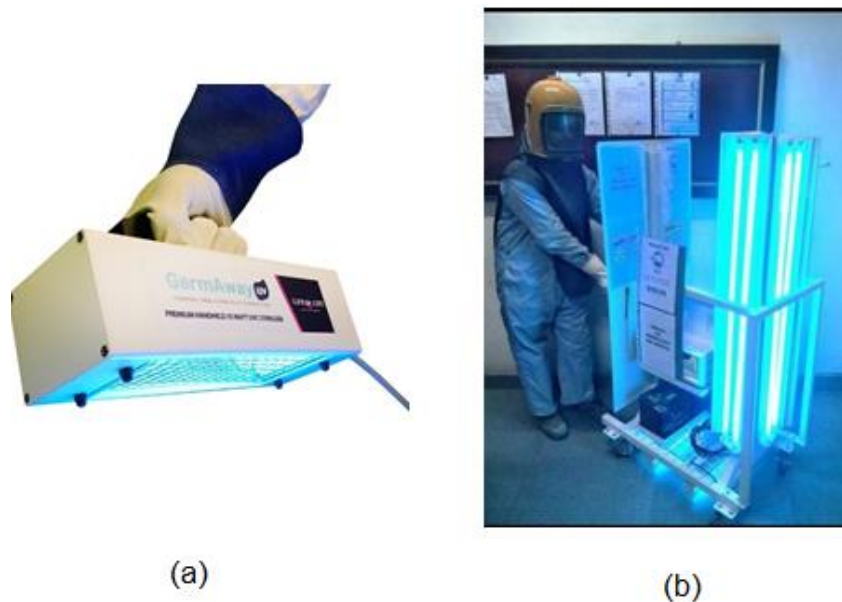
Fuente: Microchem laboratory. Disponible en: <https://microchemlab.com/test/uv-room-disinfection-devices>

- Módulos de desinfección operado manualmente

Como se ha establecido anteriormente, la exposición a la luz UVc presenta un alto riesgo para la piel y los ojos. Sin embargo, esta radiación también es fácilmente absorbida por la gran mayoría de materiales y gracias a esto existen varios dispositivos diseñados para ser operados por una persona con protección como ropa cerrada, guantes, gafas o caretas plásticas. Tienen la ventaja de que la luz se puede acercar a las superficies deseadas y por lo tanto se alcanza un nivel mucho más satisfactorio que en el punto anterior con menor tiempo y energía.

Existen dispositivos de esta categoría para llevar en la mano como en la figura 6a o tipo carreta con diferentes geometrías para empujar por el suelo como en la figura 6b

Figura 6. Módulos de desinfección operado manualmente.



Fuente: cureUV. Disponible en: <https://www.cureuv.com/collections/handheld-sanitizers/products/germawayuv-35-watt-handheld-uv-c-surface-sanitizer>

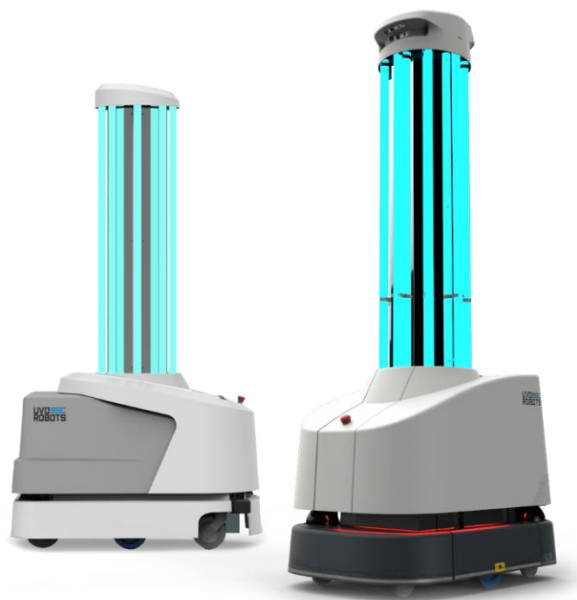
Indiamart. Disponible en: <https://www.indiamart.com/proddetail/uv-c-disinfection-trolley-22317258573.html>

- Robots autónomos para desinfección

Finalmente se llega a la categoría de los robots autónomos para desinfección como la solución más completa, pero a la vez más costosa y compleja de las presentadas. Estos robots que a la fecha de esta redacción en abril del 2021 son producidos por múltiples compañías a nivel mundial y especializadas, pero no limitadas a la desinfección de hospitales se caracterizan por su capacidad de desplazarse por los espacios previamente programados con la mínima interacción humana.

La mayoría de estos robots cuentan con una estructura vertical donde se ubican los bombillos en disposición circular y requieren de hardware y software altamente especializado.

Figura 7. Robot de desinfección autónomo comercial.



Fuente: UVD Robots. Disponible en: <https://www.uvd-robots.com/>

1.3 IDENTIFICACIÓN DEL PROBLEMA

Los salones de clase son ambientes propicios para la transmisión de enfermedades como el COVID-19, no solo por albergar a múltiples estudiantes simultáneamente sino porque cada uno de ellos está en contacto prolongado con diferentes superficies del salón de tal forma que al finalizar la clase cualquier patógeno del cual sea víctima pueda quedar en el ambiente para el próximo ocupante de su lugar.

Para la eventual reapertura de colegios y universidades no hay duda que será requisito la desinfección de los salones entre clase y clase, por lo tanto, el propósito de este proyecto es construir un sistema de desinfección móvil autónomo con luz ultravioleta germicida, que pueda ser usado para sustituir el trabajo de desinfección manual de forma efectiva reduciendo considerablemente el trabajo del personal de limpieza y el gasto de productos químicos y así mismo brinde mayores posibilidades

de desinfección donde no es posible el uso de líquidos para este fin como dispositivos electrónicos o textiles.

Este proyecto busca fomentar el desarrollo de nuevas tecnologías enfocadas en la bioseguridad y en las nuevas maneras de adecuar los espacios en respuesta a la pandemia del coronavirus, va a permitir evaluar el desempeño de la luz germicida y la robótica móvil en conjunto para la limpieza de salones, y la posibilidad de implementar esta solución en muchos otros ambientes que lo necesitan.

Debido al difícil acceso a las aulas de clase de la universidad causado por la emergencia sanitaria que afronta el país, se consultó con funcionarios de la escuela de ingeniería mecánica sobre las características de distribución dentro de los salones y los tiempos estimados de limpieza tradicional. A continuación, se muestran los análisis obtenidos con el apoyo de los funcionarios encargados de estas dichas áreas.

1.3.1 Requerimientos del sistema para la limpieza:

- Limpieza de mesones: los mesones deben ser el principal objeto de desinfección ya que se encuentra en contacto muy cercano con la cara y las manos del estudiante. Así mismo, la parte inferior de los mesones puede recibir indirectamente la infección y se le debe dedicar un esfuerzo para su desinfección.
- Limpieza de las sillas: Las sillas tienen menor riesgo de verse afectadas ya que se encuentran más alejadas de la cara y no reciben tanto contacto con las manos, sin embargo, también deben tener su dosis de desinfección con una prioridad más baja tanto en el espaldar como en el asiento.

- Limpieza de las paredes: las paredes, en el área adyacente a los mesones también deben ser desinfectadas ya que muchos estudiantes se apoyan en ellas desde sus sillas
- Duración de la limpieza en un salón: La limpieza manual de un salón (exceptuando la parte de barrido y trapeado) solamente teniendo en cuenta un aseo como el mencionado anteriormente de las mesas y sillas se estima en unos 15 minutos aproximadamente para un salón de 24 personas y un docente, se debe buscar mantener un tiempo cercano a este para no interferir con los horarios de las clases.
- Interfaz: el robot debe ser operado por el personal de limpieza correspondiente, el cual debe estar previamente capacitado debido al riesgo que representa la radiación UV-c, debe ser fácil de activar y debe indicar claramente al usuario cuando debe abandonar el salón y cuando ha concluido su tarea.
- Ubicación de los mesones y las sillas: Se evidencia que los mesones están ubicados a una distancia fija según el aula que va desde 80 centímetros a 100 centímetros entre uno y otro, y las sillas poseen de largo unos 40-50 centímetros aproximadamente dentro de este espacio.
- Limpieza de varios salones: se espera que el robot sea capaz de limpiar mínimo dos salones con una carga completa de la batería, y así mismo debe ser liviano y fácil de transportar entre un salón y otro e incluso entre pisos

Con base a los requerimientos descritos, se elaboran las siguientes conclusiones:

- Sistema de bombillos para la parte superior e inferior de las mesas.
- Sistema de bombillos para las sillas.

- Autonomía mínima de media hora de operación continua.
- Botón de arranque y apagado automático al momento de finalizar.
- Ancho de la base inferior a 50 centímetros para garantizar el paso entre mesas y sillas.
- Tiempo de limpieza total del salón entre 10 a 15 minutos.
- Bajo peso y mecanismos para desplegar los bombillos en operación y guardarlos para el transporte.
- Considerando que un solo robot no puede cumplir con toda la demanda de salones, se debe mantener el precio en el mínimo.

2 DESCRIPCIÓN GENERAL DEL PROYECTO

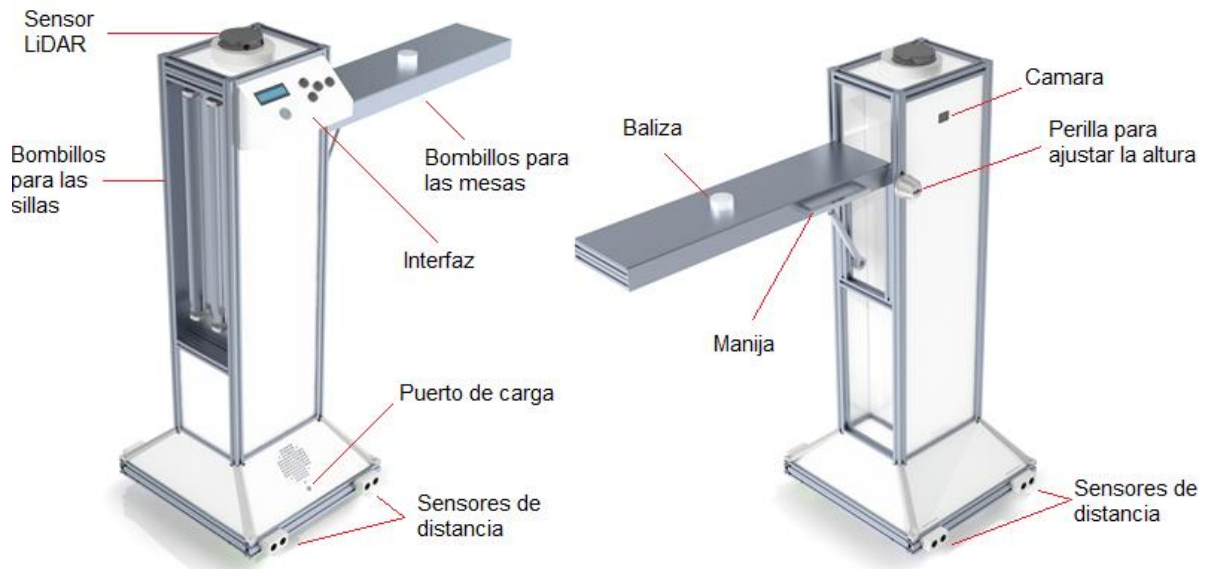
En este proyecto se realizó el diseño y construcción completa de un prototipo de robot móvil autónomo y como tal, en su desarrollo están involucrados tres grandes sistemas que corresponden a las principales áreas de la ingeniería en las que se fundamenta la robótica (mecánica, electrónica y programación):

- Sistema mecánico estructural
- Sistema de hardware y electrónica
- Sistema de autonomía y software

A continuación, se presentan de manera general estos sistemas y en la sección 3 se realiza una descripción más a fondo de los mismos con sus respectivos subsistemas.

2.1 SISTEMA MECÁNICO ESTRUCTURAL

Figura 8. Prototipo de robot desinfectante.

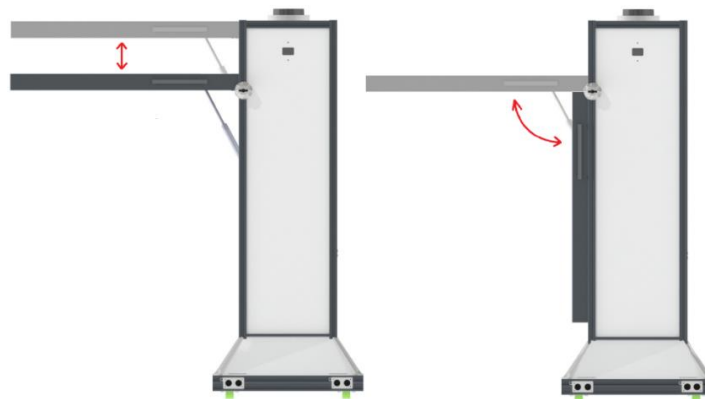


Este sistema se refiere a toda la parte externa del prototipo, las estructuras que soportan los sensores y actuadores, los mecanismos necesarios para el desplazamiento y operación y la parte estética del prototipo. En la figura 8 se muestra un renderizado del robot en vistas isométricas donde se señalan los principales componentes de este sistema.

La estructura principal está fabricada con perfil de aluminio modular y paneles de acrílico blanco. Las cubiertas de los bombillos son de lámina de aluminio de dos milímetros doblada y la base esta reforzada con una placa de aluminio de tres milímetros. El prototipo mide 40 centímetros de ancho y 40 de largo en la base con una altura máxima de un metro y un peso total de 12 kilogramos.

El prototipo cuenta con 4 bombillos de luz ultravioleta tipo c de 15 watts ubicados, dos verticalmente para la desinfección de las sillas y otros elementos del salón, y dos en un brazo móvil ajustable encargados de desinfectar con gran intensidad la superficie de las mesas.

Figura 9. Ajuste vertical y posición de guardado del brazo de los bombillos.



El brazo de los bombillos para las mesas puede guardarse y desplegarse para su uso y su altura es ajustable para adaptarse mejor a las diferentes mesas como se muestra en la figura 9. A la derecha se observa el despliegue que le permite al brazo

acomodarse a 90 o 0 grados con respecto al robot para la operación y el guardado respectivamente. Este movimiento se realiza con la ayuda de un pistón de aire pasivo o amortiguador de gas que le permite mantener estas dos posiciones de forma segura y con un desplazamiento lento. A la izquierda de la figura se muestra el movimiento de ajuste de altura, para ello se debe soltar un poco la perilla, deslizar el brazo hacia arriba o abajo con ayuda de la manija y volver a apretar la perilla a la altura deseada. La altura mínima que puede tener el brazo es de 80cm con respecto al suelo y la máxima es de 93cm.

El movimiento del robot está dado por dos motorreductores de 12 voltios a 100 rpm acoplados con ruedas de silicona de seis centímetros de diámetro en la parte trasera del robot. En la parte delantera se ubican dos ruedas esféricas de apoyo metálicas con un sistema de suspensión para mejorar la tracción del robot y su desempeño en suelo con pequeñas irregularidades.

En la figura 10 se muestra una foto del prototipo final construido con los bombillos encendidos.

Figura 10. Foto del prototipo con los bombillos UVc encendidos.



2.2 SISTEMA DE HARDWARE Y ELECTRÓNICA

En este sistema se encuentran todos los componentes de hardware encargados del control y la automatización del robot destacando los sensores y controladores. También incluye los subsistemas de alimentación y distribución de potencia.

En la figura 11 se describe de manera general la estructura del sistema de hardware basada en el modelo estructural de un sistema automatizado de Josep Balcells en su libro *Autómatas Programables*².

Para comenzar, la entrada del sistema corresponde a la potencia suministrada por una batería de ion de litio de 12 voltios fabricada por los autores con una capacidad nominal de 5000 miliamperios hora que otorga una autonomía aproximada de 25 minutos a máxima operación. Para la carga de esta batería se tiene un puerto de entrada ubicado en la parte inferior del robot al cual se conecta una fuente de 19 voltios, típica para computadores portátiles. El subsistema incluye también un led que se enciende para indicar cuando se completa la carga.

El sistema después se divide en la parte operativa, de control y de interfaz, y se tienen una comunicación externa con para programación y debug remotos.

2.2.1 Parte operativa: La parte operativa corresponde principalmente al sistema mecánico explicado en el numeral anterior en el que se encuentran los sensores y actuadores del robot. Los actuadores son principalmente los bombillos de luz ultravioleta y los motores que permiten el desplazamiento del robot y los sensores, encargados de la navegación, son los expuestos a continuación:

- LiDAR Neato XV: Ubicado en la parte superior del robot, por medio de un láser giratorio realiza un escaneo de 360 grados que permite al robot ubicarse y desplazarse dentro del salón.

² Balcells, J. 1998. *Autómatas Programables*. Alfaomega. (Pág. 140)

- Sensor inercial IMU 6050: compuesto por un giroscopio y un acelerómetro, es usado principalmente para estimar la rotación del robot esencial para la navegación.
- Encoders de efecto hall: están incorporados en los motores y permiten medir la rotación de los mismos y de esta forma controlar su velocidad y estimar el desplazamiento del robot.
- Sensores de distancia ultrasónicos: por medio de sonido de alta frecuencia miden la distancia hasta los objetos que tienen enfrente. Se ubican en la parte inferior y se usan para la detección de obstáculos.

2.2.2 Parte de control: En la parte de control destaca el controlador principal del robot que es una Raspberry pi 3 encargada de los procesos pesados de computación utilizando la plataforma de robótica libre ROS en sistema operativo Linux.

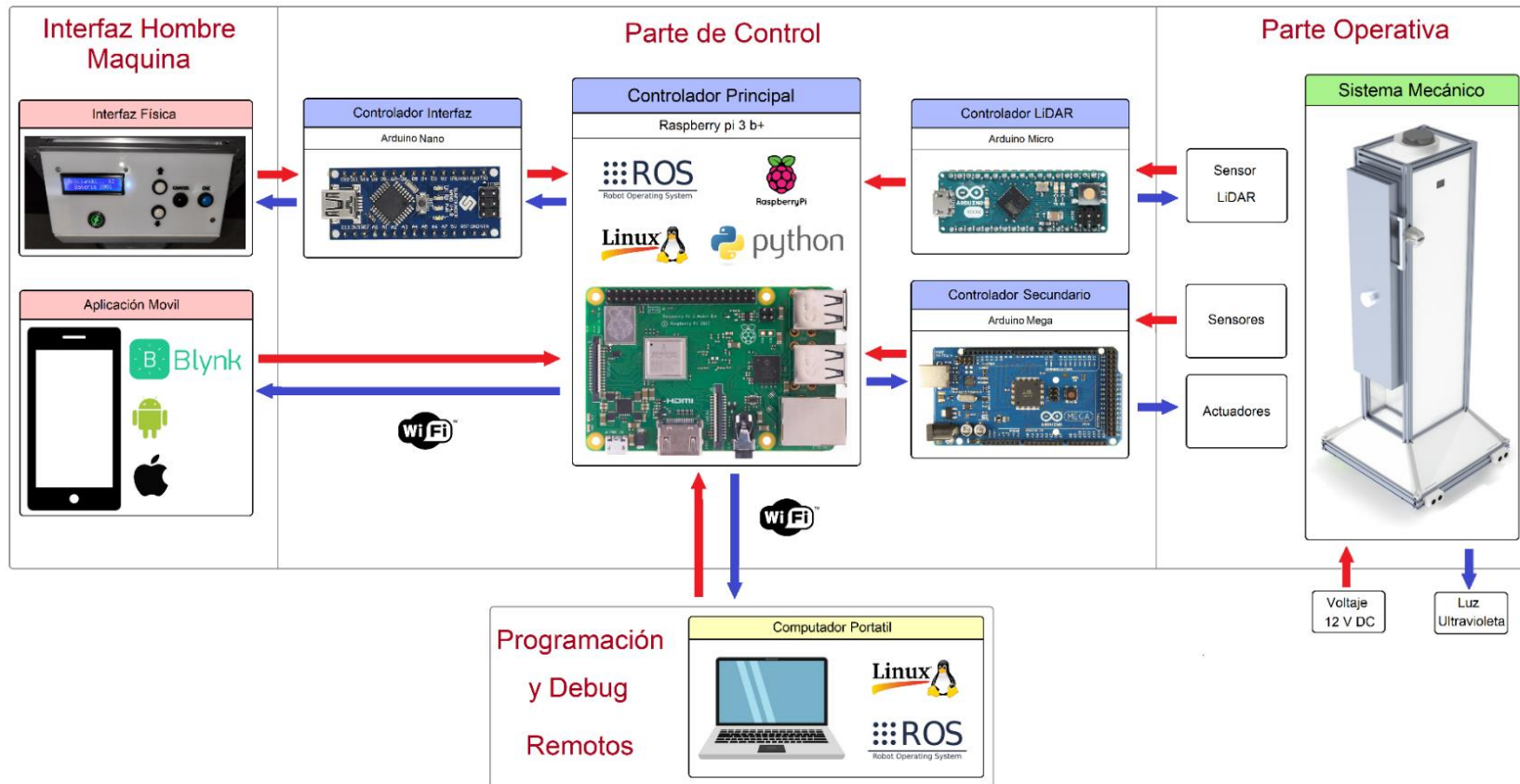
Además, se cuenta también con tres controladores basados en Arduino que se encargan de interactuar con los sensores y actuadores mientras transmiten la información hacia la Raspberry. Un Arduino mega actúa como controlador secundario encargado del control de los actuadores y de la mayoría de sensores en el robot, un Arduino micro recibe los datos del sensor LiDAR y controla su movimiento, y, por último, un Arduino nano se usa para el control de la todos los elementos de la interfaz, el sensor de voltaje de la batería y de despliegue del brazo.

2.2.3 Interfaz hombre máquina: Para la interacción del usuario con el robot se dispone de una interfaz física montada en la parte superior que cuenta con el botón de encendido principal del robot, una pantalla LCD de 16 caracteres, botones para la interacción y una baliza luminosa con buzzer para dar las alertas sobre el estado del robot. Además, se cuenta con una aplicación de celular con la plataforma Blynk que le otorga al usuario un control a distancia del robot a través de WIFI, esto incluye

la visualización de video en vivo gracias a la cámara montada en la parte frontal del robot.

2.2.4 Programación y debug remotos: Para la etapa de programación y debug se establece una comunicación a través de WIFI con un computador portátil con Linux y ROS desde la cual se puede editar en tiempo real la programación robot y acceder a todas las herramientas de visualización y debug que ofrece esta plataforma.

Figura 11. Estructura del sistema automatizado.



2.3 SISTEMA DE AUTONOMÍA Y SOFTWARE

Este sistema es el encargado de todo el comportamiento del robot donde destaca especialmente su autonomía para navegar la ruta requerida basándose en los sensores disponibles.

Los procesos fundamentales se llevan a cabo en la Raspberry pi haciendo uso de la plataforma ROS que provee librerías altamente optimizadas y herramientas de mucha utilidad para la programación de robots. De esta manera el software principal del robot se compone en su mayoría por código realizado por los autores en lenguaje Python integrado en el ambiente de desarrollo de ROS e incluye múltiples algoritmos preprogramados provenientes de las librerías que se encargan de los procesos de mayor complejidad para realizar la difícil tarea de la navegación autónoma.

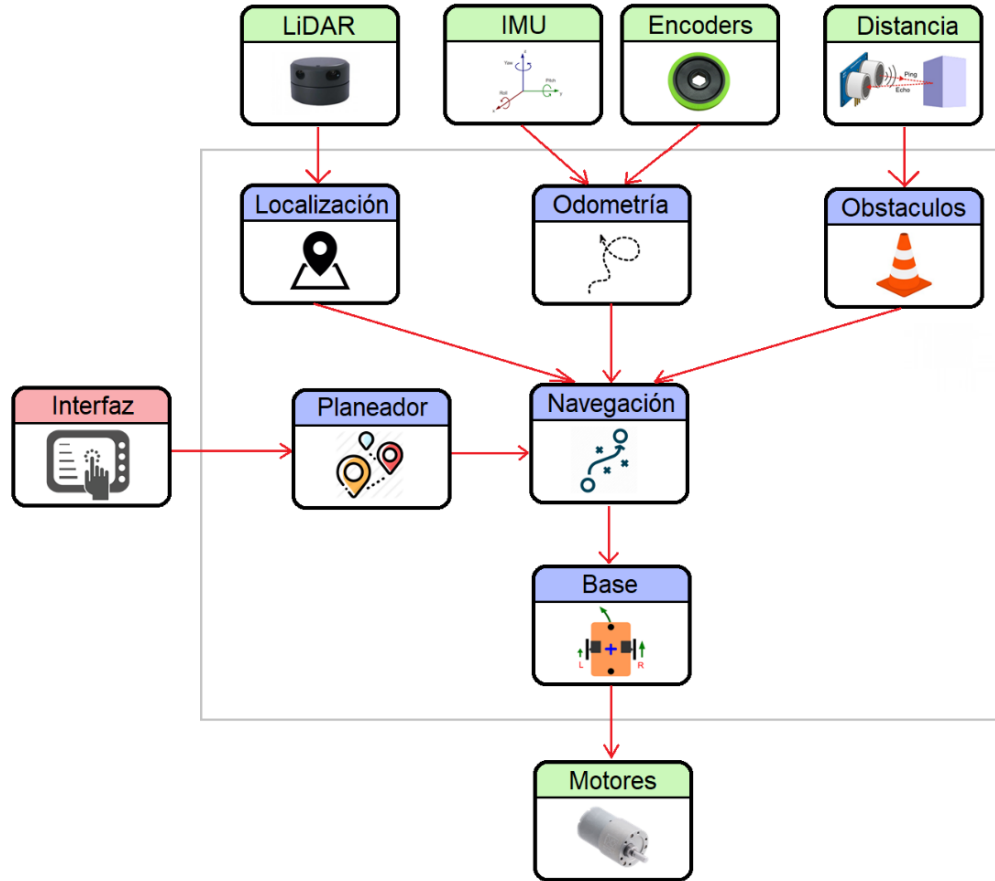
En la figura 12 se observa un esquema que representa de forma resumida los principales procesos que ejecuta continuamente el robot durante la operación para navegar efectivamente a través del salón. En verde se ilustran los sensores y actuadores con sus respectivas capas de software, en rojo la entrada proveniente de la interfaz ya sea física o a través de la aplicación móvil y en azul los algoritmos internos de procesamiento que se resumen a continuación.

- Localización: toma los datos dados por el sensor LiDAR y los compara con un mapa del salón previamente cargado para estimar la posición del robot en el salón.
- Odometría: este proceso utiliza los sensores internos, los Encoders y el IMU, para calcular la trayectoria recorrida por el robot y obtener una posición con respecto al punto de inicio.
- Detección de obstáculos: con ayuda de los sensores de distancia ultrasónicos ubicados en la base se verifica que la ruta del robot no se encuentre bloqueada por ningún obstáculo.

- **Planeador:** este proceso es el encargado de proveer la ruta que debe seguir el robot dependiendo del salón en que se encuentre y dar dirección al proceso de navegación, también recibe las órdenes dadas por la interfaz de inicio, pausa o cancelación.
- **Navegación:** uno de los procesos fundamentales del comportamiento del robot, se encarga de tomar la posición dada por los procesos de localización y odometría y la ruta dada por el planeador para computar el movimiento que debe realizar el robot para llegar al punto deseado, teniendo en cuenta también la presencia de obstáculos en la ruta.
- **Base:** este proceso toma las instrucciones de movimiento dadas por el proceso de navegación y los convierte en comandos para controlar directamente los motores de la base. Además, implementa varios sistemas de control que aseguran que el robot realiza su movimiento de manera adecuada tanto en dirección como en velocidad.

En la sección 3.3 se presentan estos procesos de forma detallada y organizada más acorde a la forma en que el robot realiza realmente los algoritmos en lugar de la visión general que se pretende presentar en esta sección.

Figura 12. Diagrama general del software.



3 DESCRIPCION DE LOS SISTEMAS

3.1 SISTEMA MECÁNICO ESTRUCTURAL

Uno de los principales obstáculos que se presenta para la fabricación de este prototipo fueron las cuarentenas y las restricciones de movilidad producto de la pandemia por el mismo COVID 19. Por esta razón, un criterio importante en el diseño fue facilitar los procesos de manufactura de tal forma que, en su mayoría, el prototipo pueda ser construido desde casa con las herramientas disponibles y con poca ayuda de terceros.

Con base en este criterio, se decide hacer uso del perfil modular de aluminio tipo estrella para la mayor parte de la estructura, el cual cuenta con gran variedad de accesorios para facilitar su ensamble de tal forma que la construcción se limita a cortar el perfil a la longitud deseada con una sierra y ensamblar con tornillos y tuercas. Además, es ideal para prototipos porque permite ajustar fácilmente muchas de las medidas durante la construcción y añadir nuevos componentes debido a su carácter modular. La figura 13 muestra las dos variantes del perfil usado, 2040 y 2020 debido a las medidas de sus lados en centímetros

Figura 13. Perfil de aluminio tipo estrella 2040 y 2020.



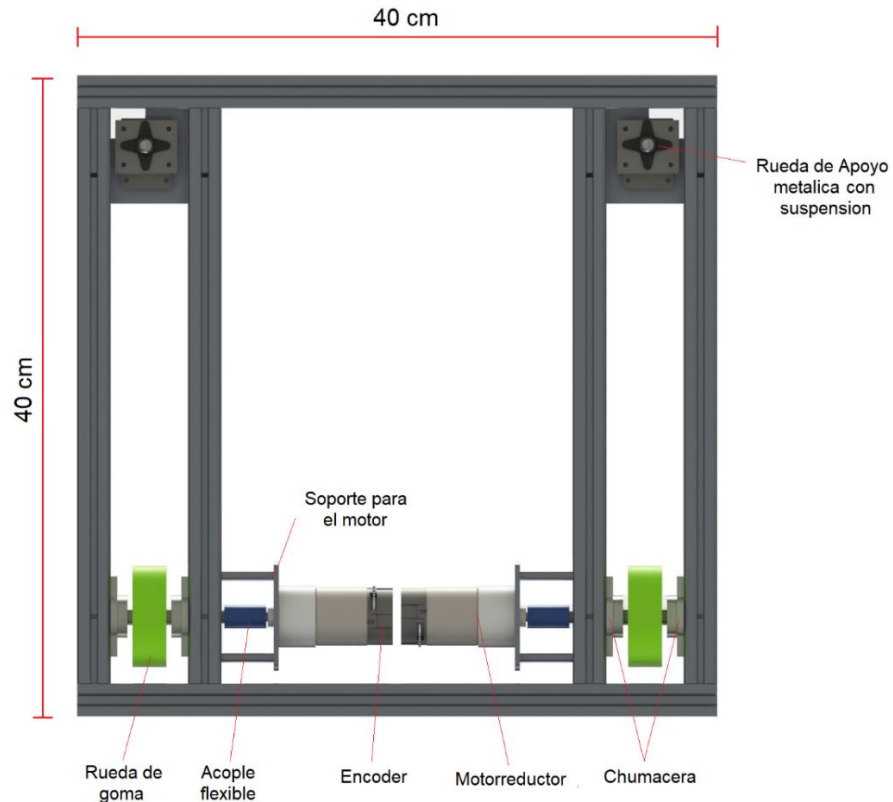
3.1.1 Base: La base del robot es la encargada del movimiento por el piso de los salones la cual cumple con las necesidades de tamaño para no tener inconvenientes en el desplazamiento entre las sillas y mesas a limpiar.

A su vez, la base cumple el papel fundamental de soportar todas las demás estructuras del robot, ya que es la pieza principal a la cual van anclados los restantes componentes, por lo cual la simplicidad y funcionalidad de dicha estructura está completamente ligada al desempeño de sus componentes.

Para la construcción de la base se emplea el perfil 2040 orientado de manera vertical para darle estabilidad a la estructura y permitir el ensamble de componentes posteriores.

En la figura 14 se presenta el modelo final de la base en una vista inferior. Sus medidas exteriores son de 40x40 cm y para ello se hace uso de seis segmentos de perfil 2040, dos de 40 cm y cuatro de 36 cm para un total de 224 cm. En la parte trasera se encuentra el sistema motriz mientras que en la delantera las ruedas de apoyo. Finalmente, se le añade una placa de aluminio de dos milímetros para cerrar la base y mejorar su rigidez como se muestra en la figura 15.

Figura 14. Vista inferior de la base.



El sistema motriz está compuesto a cada lado por un eje de acero inoxidable de 8 mm de diámetro que se apoya en dos chumaceras planas montadas a cada uno de los perfiles de aluminio con un taladrado por el que pasa el eje. Entre las chumaceras se ubica la rueda de goma de 6 cm de diámetro debidamente espaciada y hacia el centro del robot el eje se acopla mediante un acople flexible al eje de 6 mm del motor. El motor, por su parte, se ubica en una placa de soporte en aluminio montada al perfil con separadores roscados. En el anexo B se encuentran los respectivos cálculos mecánicos para la concernientes al diseño del eje.

En el otro extremo de la base se ubican entonces las ruedas de apoyo cuya función es simplemente dar la estabilidad necesaria sin ofrecer resistencia al movimiento ni a los giros. Para esta tarea, se hace uso de ruedas de apoyo de acero esféricas que

como las que se observan en la figura 15 que luego fueron montadas con un sistema de suspensión explicado en la siguiente sección.



Figura 15. Prototipo de la base con apoyo esféricas sin suspensión.



Para el diseño de la base se tuvieron en cuenta alternativas de selección de componentes que finalmente llevaron al modelo presentado anteriormente. A continuación, se presentan las alternativas propuestas para la selección de los motores y de las ruedas de apoyo con sus respectivas ventajas y desventajas y los criterios más importantes tenidos en cuenta para la selección.

3.1.1.1 Selección de alternativas de la base:

Tabla 1. Alternativas de motores para la base.

Alternativa	Ventajas	Desventajas
Motor DC con escobillas 	<ul style="list-style-type: none">• Alta eficiencia• Buen torque a altas velocidades• Fáciles de operar	<ul style="list-style-type: none">• Baja precisión• Mayor desgaste de sus componentes
Motor paso a paso 	<ul style="list-style-type: none">• Muy alta precisión• Buen torque a bajas velocidades• Mayor durabilidad	<ul style="list-style-type: none">• Alto consumo de energía• Limitado a bajas velocidades• Menor disponibilidad en el mercado

El criterio decisivo para la elección el motor con escobillas fue la eficiencia energética superior. Los motores paso a paso otorgan fácilmente una enorme precisión y control de velocidad, pero su consumo elevado de energía lo hacen poco viable para su uso en la base. La precisión de los motores DC puede ser mejorada con sensores y un sistema de control por software.

Tabla 2. Alternativas de rueda de apoyo.

Alternativa	Ventajas	Desventajas
<p>Rueda omnidireccional</p> 	<ul style="list-style-type: none"> • Buena Maniobrabilidad en cualquier dirección • Buen soporte de carga 	<ul style="list-style-type: none"> • Muy baja disponibilidad en el mercado y alto precio • Puede generar vibraciones
<p>Rueda de apoyo esférica</p> 	<ul style="list-style-type: none"> • Muy buena maniobrabilidad • Mas fácil de montar debido a su menor tamaño 	<ul style="list-style-type: none"> • Poco contacto con el suelo • Mal desempeño en obstáculos • Mayor desgaste
<p>Rueda de apoyo giratoria</p> 	<ul style="list-style-type: none"> • Alta disponibilidad en el mercado y bajo precio 	<ul style="list-style-type: none"> • Ocupa mucho espacio • Pueden afectar la maniobrabilidad en especial el movimiento en línea recta

En este caso, no se llega a una decisión a priori, sino que, gracias al bajo precio relativo de estos componentes se decide diseñar la base de tal forma que las tres opciones sean compatibles y probar su funcionamiento en el prototipo terminado.

Inicialmente se implementaron las ruedas omnidireccionales que cuentan con cilindros distribuidos en su circunferencia para permitir el deslice suave en todas las direcciones y fueron montadas como se muestra en la figura 16. Sin embargo, debido a la escasa variedad de este tipo de ruedas en el mercado, las necesidades

geométricas del presente diseño y el presupuesto, las ruedas adquiridas no contaban con la calidad suficiente para esta tarea y generaba niveles indeseables de ruido y vibración con el movimiento.

Figura 16. Prototipo de la base usando las ruedas omnidireccionales.



La siguiente alternativa fueron las ruedas de apoyo giratorias muy usadas en muchos aparatos móviles, pero desde un inicio se presentaron dificultades para adaptarlas al presente diseño debido al tamaño que ocupan. Así mismo, una vez montadas unas ruedas compatibles y después de una serie de pruebas se determina que con esta alternativa se ve afectada la dirección de movimiento del robot presentando una mayor tendencia a desviarse en el movimiento rectilíneo y a responder erráticamente al cambio de dirección. Por lo tanto, para esta aplicación donde se necesita precisión este tipo de ruedas no son la mejor alternativa.

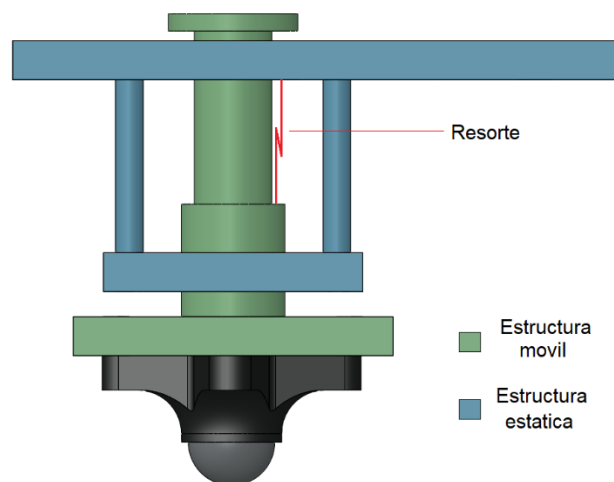
Posteriormente se reemplazaron con ruedas de apoyo esféricas metálicas que brindan un movimiento mucho más libre y suave en todas las direcciones, sin embargo, debido a su punto de apoyo limitado con el suelo y al pequeño diámetro de la esfera, con esta alternativa se presenta cierta inestabilidad y dificultades para desplazarse en superficies con irregularidades como grietas o desniveles.

De esta forma, se decide seleccionar las ruedas de apoyo esféricas e implementar un sistema de suspensión para aliviar los aspectos negativos de estas.

3.1.2 Sistema de suspensión: El objetivo de este sistema es, mediante el uso de un resorte aumentar la fuerza de contacto entre la rueda de apoyo y el suelo y permitir el desplazamiento hacia abajo de la rueda para mantener la estabilidad ante variaciones de la geometría del suelo.

Este mecanismo se diseña como un adicional teniendo ya construido el resto de la base y por lo tanto se realizó de tal forma que se adapte al diseño previo en este primer prototipo sin modificaciones considerables y por lo tanto puede ser optimizado en cuanto a materiales y procesos en futuras versiones con un rediseño general de la base.

Figura 17. Diseño de sistema de suspensión.



La figura 17 muestra el modelo propuesto para este prototipo, la estructura estática (mostrada en azul) consta de la placa superior de aluminio de 4 mm con una geometría que se adapta a la estructura existente y otra placa pequeña abajo unida mediante separadores roscados que actúa como tope y estabilizador para el desplazamiento.

La parte móvil (mostrada en verde) corresponde al eje del donde se ubica el resorte y la placa que soporta la rueda, ambas piezas son de acero inoxidable y se unen mediante soldadura. En la parte superior del eje el tope esta dado por una arandela unida con tornillo.

Para la fabricación de estas piezas se requiere el uso de un taladro de árbol para los orificios pasantes, una fresadora para las esquinas redondeadas y un torno para el eje.

Finalmente, se procede al diseño del resorte partiendo del peso total del robot que se mide aproximadamente en 12 Kg y que, teniendo cuatro puntos de apoyo (dos ruedas de tracción y dos ruedas de apoyo) cada sistema de suspensión soporta aproximadamente la cuarta parte de este peso, es decir, 3 Kg. Además, se desea desplazamiento máximo 15 mm.

Con estos requerimientos se procede a calcular y seleccionar las constantes del resorte con la siguiente formula.

$$\frac{F}{y} = \frac{G * d^4}{8 * D^3 * N} \quad (1)$$

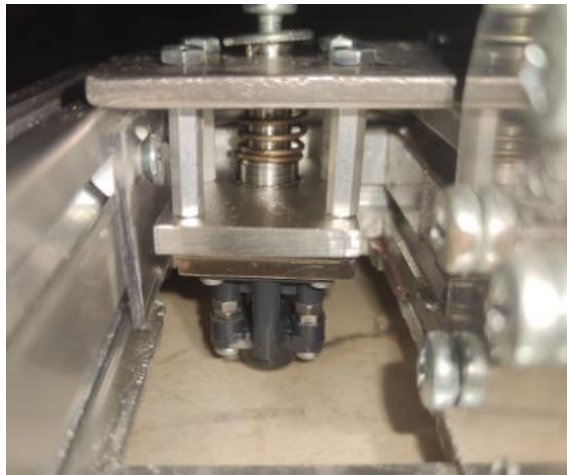
Donde:

- N es el número de espiras el cual en este caso es de 6 por cuestiones de espacio.
- G es el módulo de elasticidad en cortante del acero inoxidable de 69GPa
- d el diámetro del alambre a calcular
- D es el diámetro del eje guía que se selecciona en 9 mm
- F es la fuerza esperada de compresión de 3Kgf
- y es el desplazamiento máximo del resorte el cual es de 15mm

De la ecuación 1 despejamos d (Diámetro del alambre) para poder fabricar el resorte con las características requeridas, por lo que tenemos finalmente un resultado de 1mm para su elaboración.

En la figura 18 se muestra el sistema de suspensión instalado en el prototipo.

Figura 18. Sistema de suspensión instalado.

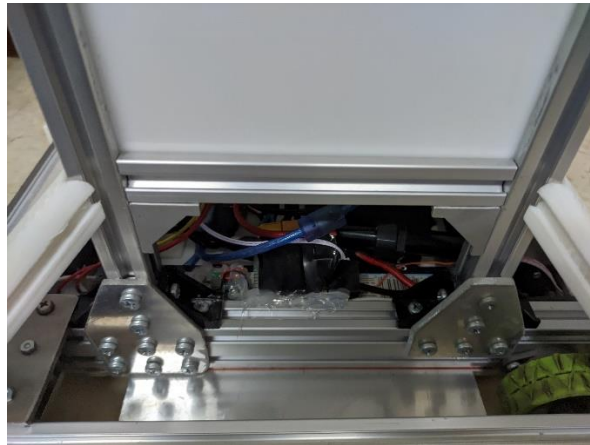


3.1.3 Estructura vertical: La estructura vertical es la encargada de soportar las dos lámparas desinfectantes y el sensor LiDAR encargado de la navegación del robot, por tanto, el diseño debe contar con espacio adecuado para la instalación de dichas lámparas y una versatilidad para la adición de acabados estéticos, garantizando una sencillez del diseño para ahorrar costos pero que no perjudiquen su funcionalidad.

La estructura vertical fue construida con perfil 2020 acoplado a 90° a los perfiles 2040 de la base como se muestra en la figura 19. Consta de cuatro perfiles verticales de 90 cm unidos con otros segmentos de perfil de 20 y 15 cm en la parte superior e

inferior para formar la estructura rectangular mostrada en la figura 20. En total son necesarios cinco metros de perfil.

Figura 19. Montaje de la estructura vertical.



El diseño de esta estructura se caracteriza por su simpleza, además de ser fácil de construir se busca reducir considerablemente el peso de la misma de tal forma que el centro de gravedad del robot se mantenga lo más bajo posible y por lo tanto mejore su estabilidad y no exista riesgo de voltearse.

Figura 20. Estructura vertical.



3.1.4 Lámparas desinfectantes: Para el diseño de las lámparas desinfectantes se cuenta con cuatro bombillos Phillips UVC de 15W con un diámetro de 2.8 cm y un largo total de 45.2 cm.

A la izquierda del robot se posicionan dos bombillos estáticos de forma vertical que serán los encargados de la limpieza de las sillas y otros elementos del salón mientras que a la izquierda se ubican los bombillos para las mesas que tienen la capacidad para cambiar de posición y variar su altura y cuyo mecanismo será descrito en la siguiente sección.

Para ambos sistemas la estructura consiste en una lámina de aluminio de 2 mm de espesor plegada en forma de U en cuyo interior se ubican los bombillos con la ayuda de los soportes mostrados en la figura 21. Estos permiten colocar y asegurar fácilmente los bombillos y realizar las conexiones eléctricas necesaria.

Figura 21. Soporte para bombillo



La lámpara de desinfección estática se ancla directamente en el perfil 2020 de la estructura vertical mediante el uso de tornillos (figura 22a) mientras que para la lámpara móvil se usan segmentos de perfil 2040 en los extremos para proveer la rigidez necesaria (figura 22b).

Figura 22. Lámpara estática (a) y móvil (b).



(a)



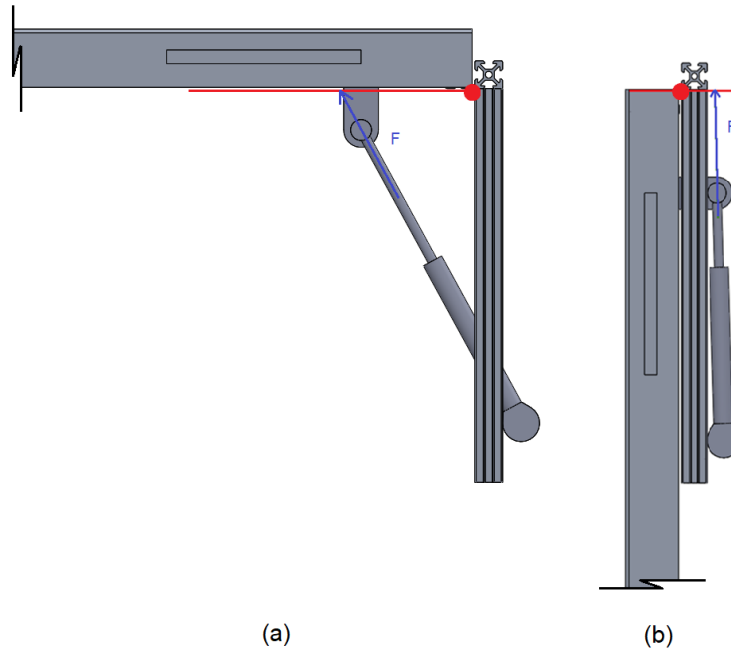
(b)

3.1.5 Variador de altura y posición de la lámpara de las mesas: Como se estableció anteriormente, la lámpara móvil ejecuta dos movimientos, uno giratorio que le permite intercambiar entre la posición de operación y la de guardado y otro lineal para ajustar la altura de la lámpara.

Para el movimiento de rotación se busca que la lámpara se posicione a 0 y 90 grados con respecto a la estructura vertical y para ello se hace de un pistón de gas amortiguador como los comúnmente usados en gabinetes de cocina u otros muebles caseros y por lo tanto se consiguen fácilmente en el mercado y a muy bajo precio.

En la figura 23 se muestra la acción que realiza el pistón en las dos posiciones, en azul se dibuja el vector de la fuerza de máximo 80 newtons que produce el pistón y que actúa sobre la lámpara formando una palanca como ilustra la línea roja, el punto marca la posición de la bisagra. Para ambos casos se observa que la fuerza F se encarga de mantener el equilibrio en dicha posición y para cambiar de una a la otra solo hace falta un pequeño empujón o jalón que varíe suficiente el ángulo para romper el equilibrio. Además, el pistón también actúa como amortiguador limitando la velocidad de este movimiento para que la lámpara se posicione de forma segura.

Figura 23. Acción del pistón en posición de operación (a) y guardado (b).



El movimiento lineal permite ajustar la altura de la lámpara y por lo tanto mueve toda la estructura vista en el punto anterior. Este mecanismo es necesario para no sobre restringir la funcionalidad del prototipo, pero no es fundamental para la operación y su uso será esporádico. Por lo tanto, el diseño de este mecanismo se realizó priorizando la simpleza en su fabricación.

La estructura de la lámpara se encarrila a la estructura vertical principal del robot gracias a las acanaladuras propias del perfil 2020 y a unas guías que permiten que la lámpara se deslice libremente hacia arriba y abajo. Para bloquear la posición a la altura deseada se usa una perilla de impresión 3D blanca mostrada en la figura 24, esta perilla gira un tornillo que pasa a través de un taladrado en el perfil exterior y aprieta una tuerca ubicada en el acanalado del perfil del marco de la lámpara produciendo el apriete deseado.

Figura 24. Estructura móvil de la lampara.



En la figura 24 también se observa la manija de aluminio montada sobre la cubierta de la lampara que permite al usuario realizar fácilmente los movimientos tanto para la posición de guardado y operación como para el ajuste de la altura.

El diseño de estos mecanismos está sujeto también a una selección de alternativas, para el movimiento de guardado y operación se elige por unanimidad el mecanismo que hace uso del pistón de aire, pero para la variación de altura fue necesaria la elaboración de la respectiva tabla de alternativas mostrada a continuación.

3.1.5.1 Selección de alternativas para el brazo :

Tabla 3. Alternativas para el subsistema del brazo.

Alternativa	Ventajas	Desventajas
Tornillo sin fin y manivela	<ul style="list-style-type: none"> • Mas comodidad para el usuario • Permite posicionar el brazo en cualquier posición 	<ul style="list-style-type: none"> • Mayor complejidad y precio
Varilla perforada y pasador para seleccionar entre ciertas alturas predeterminadas	<ul style="list-style-type: none"> • Brinda un mejor ajuste 	<ul style="list-style-type: none"> • Limita la posición del brazo a las posiciones predeterminadas • Puede presentar dificultad para el usuario en su ajuste
Perilla para asegurar la estructura móvil en cualquier posición mediante el apriete de un tornillo	<ul style="list-style-type: none"> • Mas fácil de fabricar • Permite posicionar el brazo en cualquier posición 	<ul style="list-style-type: none"> • Mayor desgaste de las piezas • Puede presentar deslizamiento y por lo tanto desajuste en condiciones anormales

En este caso se selecciona la alternativa de apriete por la simpleza en su implementación, Poder variar la altura del brazo es necesario para permitir que el prototipo sea implementado en diferentes escenarios donde la altura de las mesas sea diferente, pero no es un mecanismo esencial al menos en las primeras etapas de desarrollo, y por esta razón, se considera suficiente esta alternativa a pesar de sus desventajas.

3.1.6 Interfaz: La interfaz es la encargada de comunicar al hombre con la máquina, por lo que un diseño visual y fácil de entender es lo principal que se busca con la misma, para ello se diseña con botones completamente sencillos de entender y un medio de comunicación simple para entender que está ocurriendo al interior del software.

La estructura de la interfaz se fabrica con láminas acrílico de 3mm de espesor de color blanco unida por tornillos y sellada con silicona blanca. Se ubica en la parte superior del prototipo a un ángulo de 45° para hacer más fácil su visualización por parte del operario. En esta, se posicionan 4 botones de operación (aceptar, cancelar, arriba y abajo), un botón de encendido y apagado y un display LCD para proyección de mensajes y comunicación con el operario.

Figura 25. Interfaz.



3.1.7 Sensores exteriores: En esta categoría se ubican los sensores que interactúan con el entorno y requieren un montaje en el exterior del robot, es decir el sensor LiDAR y los ultrasonidos.

El sensor LiDAR debido a su función de mapeo necesita una visualización de 360° libre de obstáculos propios del robot. Por esta razón se ubica en la parte superior de la estructura vertical soportado por una placa de acrílico blanco de 4 mm de espesor anclada entre los perfiles 2020 que la conforman. Además, se le fabrica una cubierta en impresión 3D para proteger su mecanismo y mejorar la estética.

Por otro lado, los sensores de distancia ultrasónicos encargados de la detección de obstáculos se ubican en los perfiles 2040 frontales y posteriores de la base, para ello se les fabrica una cubierta en impresión 3D con una pestaña que se desliza por la ranura del perfil y un tornillo para ajustarlo en la posición deseada.

3.1.8 Exteriores: Por último, en los espacios sobrantes entre los diferentes perfiles estructurales del robot se ubican placas de acrílico blanco de 3 mm que protegen toda la parte interna como el cableado y la electrónica y le un aspecto mucho más estético al prototipo. Para su montaje se aprovecha la ranura de los perfiles de tal forma que las placas se mantienen en su posición sin necesidad de ningún tipo de sujeción adicional.

Para la estructura vertical la estructura de los 4 lados se monta de la siguiente manera:

- Frontal: este lado se encuentra libre y por lo tanto el acrílico cubre todo el espacio comprendido entre los perfiles verticales, superior e inferior de la estructura vertical, además sobre este acrílico se monta la cámara y por lo tanto cuenta con el agujero necesario y los tornillos para montarla.

- Posterior: en este lado se ubica la interfaz en la parte superior, la placa va entonces desde donde acaba la interfaz hasta el perfil inferior.
- Lateral izquierdo: al presentar la lámpara de los bombillos de las sillas el acrílico cubre desde el perfil inferior hasta donde inicia la lámpara y se adiciona pequeña placa horizontal para cubrir el espacio que se forma bajo los bombillos
- Lateral derecho: este lado requiere un subnivel debido a que la estructura de la lámpara de las mesas ocupa los perfiles laterales, por lo tanto, para su colocación se dispone de un segmento de perfil adicional en la parte inferior más adentro del robot, sobre este perfil reposa la placa y sube hasta la parte superior con ayuda de guías adheridas a las placas de acrílico frontal y posterior.

Para la base se utilizan placas con forma trapezoidal ubicadas en diagonal y soportadas por los perfiles 2040 exteriores de la base y los perfiles inferiores de la estructura vertical. Fue necesario añadir cuatro esquineros diagonales impresos en plástico ABS blanco mediante impresión 3D, los cuales sirven para conectar cada placa de acrílico junto con sus adyacentes como se muestra en la figura 26.

La placa posterior presenta una rejilla para permitir el flujo de aire de enfriamiento para la electrónica con ayuda de un ventilador, una cavidad para el LED indicativo de carga y la entrada del cargador del robot.

Figura 26. Vista frontal-izquierda (a) y posterior-derecha (b).



(a)



(b)

3.2 SISTEMA DE HARDWARE Y ELECTRÓNICA

3.2.1 Batería: La batería es de elaboración propia, para ella se usaron pilas de ion de litio 18650 de la marca Samsung que cuentan con un voltaje nominal de 3.6V y una capacidad de 2500 mAh. Se ubicaron en tres celdas cada una de dos pilas de tal forma que la batería final provee nominalmente 5000mAh entre 12.6 y 9.2 V. Además, se le adiciona el respectivo circuito de control conocido como BMS (Battery management system) que la protege de sobrecarga, sobre descarga, corto circuito y se encarga del balanceo de las celdas para asegurar un desempeño óptimo y mayor vida útil. Finalmente se fabrica una caja en impresión 3d para guardar las celdas y el circuito.

Figura 27. Tester USB para la medición de carga.



Para definir la capacidad adecuada de la batería, y por lo tanto el número de pilas en paralelo, se busca autonomía suficiente para la desinfección de 2 salones es decir unos 20 minutos de trabajo al máximo. Para este cálculo se asume un consumo continuo de 10 amperios y se aplica la siguiente ecuación:

$$20 \text{ minutos} = 0.33 \text{ h} \rightarrow 0.33 \text{ h} * 10000 \text{ mA} = \mathbf{3333 \text{ mAh}} \quad (2)$$

Por lo tanto, se selecciona una batería de dos pilas en paralelo para un total de 5000mAh.

Al realizar las pruebas, una vez construido el robot, se usa un *tester* como el que se muestra en la figura 27 que permite medir voltaje, la corriente y realiza la integración en el tiempo para estimar el consumo en mAh. De esta forma se obtiene una medida real de la capacidad aprovechable de la batería que resulta ser aproximadamente **3800 mAh** en lugar de los 5000 esperados. Esta disminución, aunque mayor de la esperada, puede atribuirse entre otras cosas a la protección del BMS que evita la descarga total, potencialmente riesgosa para la vida de la batería.

Con ayuda de este mismo *tester*, se pudo obtener una medida del consumo de corriente del robot de la siguiente manera: un valor base de 500mA que corresponde

a los consumidos por los controladores y sensores. Con el robot en movimiento un valor estable de 1.5A alcanzando picos de entre 2 y 3 amperios durante la aceleración, por lo tanto, se toman **2A** continuos como valor conservador para el cálculo. Por otra parte, el consumo de los bombillos no fue posible medirlo con este instrumento debido a que la corriente inicial requerida para encenderlos sobrepasa los máximos admisibles del sensor y por lo tanto fue necesario estimarla en base a los datos nominales: 15 Watts por bombillo da un total de 60W y asumiendo una eficiencia del 70% en el conjunto se obtienen 85,7W que equivalen a **7.1A** de la batería. De esta forma, sumando estos valores se estima el consumo continuo máximo en **9.1A**.

La autonomía real del robot sería entonces la capacidad real de la batería sobre el consumo continuo:

$$\frac{3800 \text{ mAh}}{9100 \text{ mA}} = 0.41 \text{ h} = \mathbf{25 \text{ minutos}} \quad (3)$$

Por lo tanto, suficiente aun para la limpieza de dos salones, pero plantea la necesidad de una batería de mayor capacidad.

La selección de este tipo de batería también fue realizada considerando las opciones disponibles en el mercado, con las cuales se elabora la siguiente tabla.

3.2.1.1 Selección de alternativas de la batería:

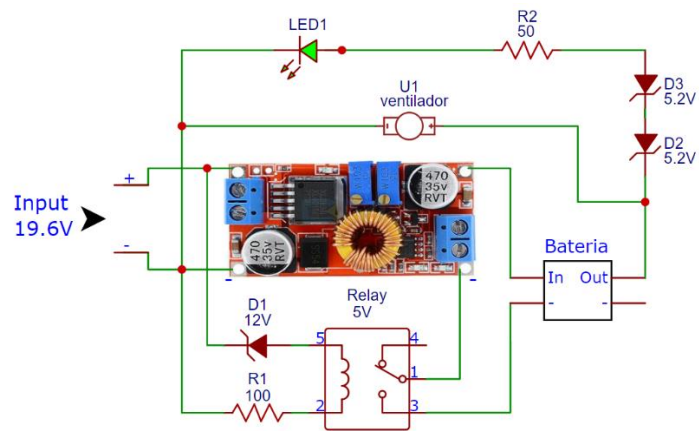
Tabla 4. Selección de alternativas de la batería

Alternativa	Ventajas	Desventajas
<p>Batería tipo Li-Po</p> 	<ul style="list-style-type: none"> • Disponibles comercialmente en packs prefabricados • Buena densidad energética • Buen tiempo de carga 	<ul style="list-style-type: none"> • Enfocadas para robótica de hobby requieren un protocolo de carga especializado • Alto precio
<p>Batería de Ion de litio</p> 	<ul style="list-style-type: none"> • Densidad Energética superior • Buen tiempo de carga • Mayor durabilidad 	<ul style="list-style-type: none"> • Se consiguen comercialmente por celdas separadas • Requieren ensamble y electrónica especializada de seguridad
<p>Batería de Acido Plomo</p> 	<ul style="list-style-type: none"> • Fáciles de usar y de cargar • Menor precio • Muy usadas para todo tipo de aplicaciones 	<ul style="list-style-type: none"> • Baja durabilidad • Poco eficientes en cuanto a peso y volumen

Se selecciona la batería de ion de litio debido a su alta densidad energética que permite reducir el peso y el espacio que ocupa y a la facilidad y rapidez de carga.

3.2.2 Circuito de carga: El reto para el uso de una batería de elaboración propia es permitir la carga de la forma más fácil y segura para el operario y con un cargador comercialmente disponible. Según las especificaciones del fabricante de las pilas, la batería debe ser cargada a un voltaje constante de 12.6V y una corriente segura de 3750 mA. Para esto, se diseña un circuito alrededor de un módulo basado en el chip XL4015 que actúa como un reductor y regulador de voltaje y corriente ideal para esta aplicación. Además, se incluye un led indicador de carga completa y un ventilador para evitar el sobrecalentamiento del módulo.

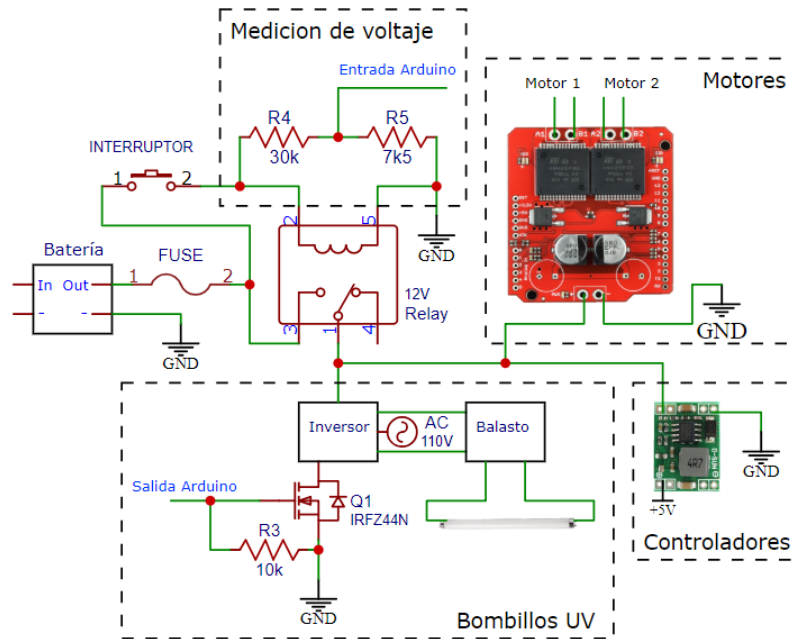
Figura 28. Esquema del circuito para el cargador.



Con este circuito, el ventilador y el led se alimentan con el voltaje de salida de la y con ayuda de los diodos Zener D2 y D3 el led se enciende solo cuando dicho voltaje se acerca al voltaje máximo de 12.6V indicando la carga completa. Finalmente se incluye un relay que desconecta todo el circuito cuando no se encuentra conectado el cargador para que estos elementos solo se enciendan mientras el robot carga y no consuman energía en otro momento.

3.2.3 Circuito principal:

Figura 29. Esquema del circuito principal de distribución de voltaje.



Este circuito mostrado en la figura 29 es el encargado de distribuir el voltaje de la batería entre los tres sistemas principales del robot: los motores, los bombillos y los controladores.

Para los controladores se tiene simplemente un módulo reductor que reduce el voltaje de la batería (que por convención se generaliza en 12V) a los 5 voltios a los que trabaja todo el sistema, estos alimentan directamente a la Raspberry pi que a su vez alimenta los demás componentes.

Para los motores se cuenta con una Shield Monster Moto Sparkfun, la cual es un driver de dos canales para el control PWM de los motores con un Arduino.

Los bombillos requieren el uso de un inversor de voltaje que convierte los 12 voltios de corriente continua de la batería a los 110 voltios de corriente alterna que necesitan los bombillos. Dicho inversor está controlado por un transistor tipo

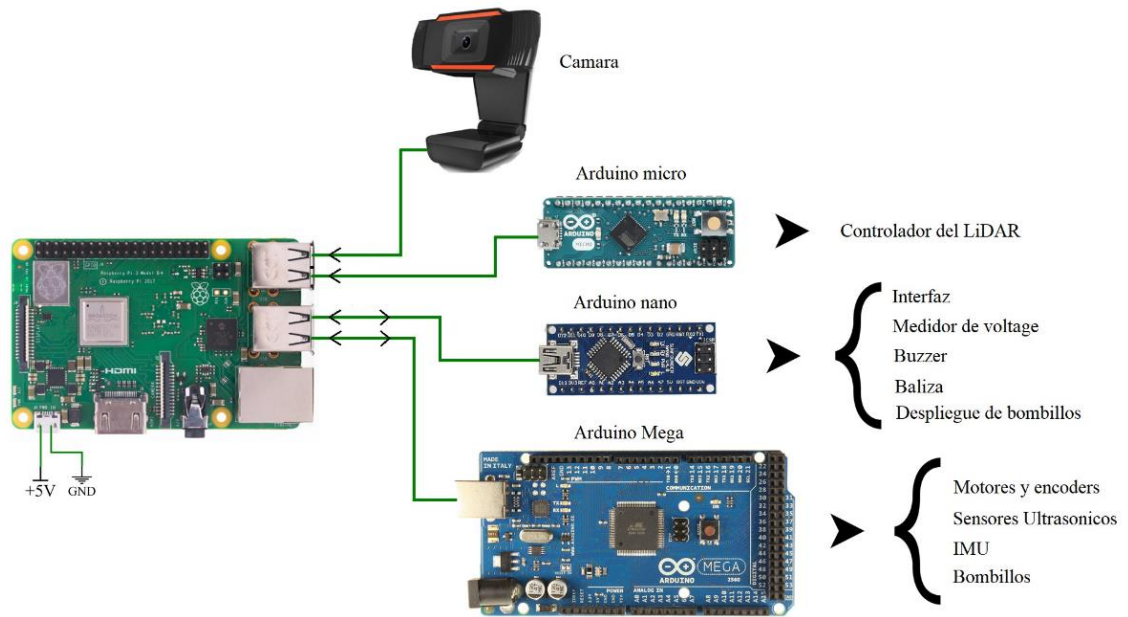
MOSFET para permitir el encendido y apagado seguro de los bombillos con una señal de Arduino. Además, es necesario otro componente conocido como balasto, que es indispensable para una correcta alimentación de los bombillos.

Así mismo, se añade un punto de medición del voltaje de la batería que es simplemente un divisor de voltaje que lo reduce a un punto inferior a 5v para que pueda ser leído por una entrada analógica del Arduino de la interfaz y de esta manera conocer la carga disponible.

Por último, todo el circuito se abre y se cierra con ayuda de un relay y el interruptor a disposición del operario ubicado en la parte superior del robot y se protege de cortocircuito con un fusible.

3.2.4 Los controladores: Como se mencionó anteriormente, el voltaje reducido a 5v de la batería alimenta la Raspberry pi que es el controlador principal del robot y se encarga de la mayoría de los procesos del mismo. Además, se tienen otros controladores basados en Arduino que principalmente se encargan de los sensores y los actuadores y se comunican con la Raspberry a través de los puertos USB. Así mismo se cuenta con una cámara que transmite a través de Wifi para brindar al operario mayor control a distancia del robot en de la aplicación para celular.

Figura 30. Esquema de la conexión de los controladores.

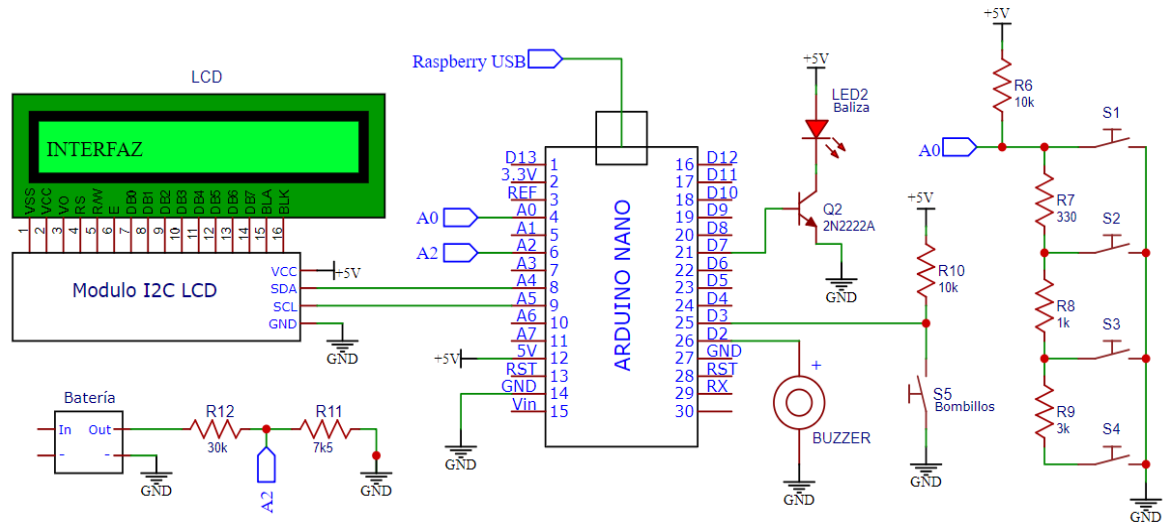


Como se mencionó anteriormente, el voltaje reducido a 5v de la batería alimenta la Raspberry pi que es el controlador principal del robot y se encarga de la mayoría de los procesos del mismo. Además, se tienen otros controladores basados en Arduino que principalmente se encargan de los sensores y los actuadores y se comunican con la Raspberry a través de los puertos USB. Así mismo se cuenta con una cámara que transmite a través de Wifi para brindar al operario mayor control a distancia del robot en de la aplicación para celular.

El Arduino micro actúa simplemente como puente entre el sensor LiDAR y la Raspberry y a su vez controla el con PID la velocidad de rotación del mismo.

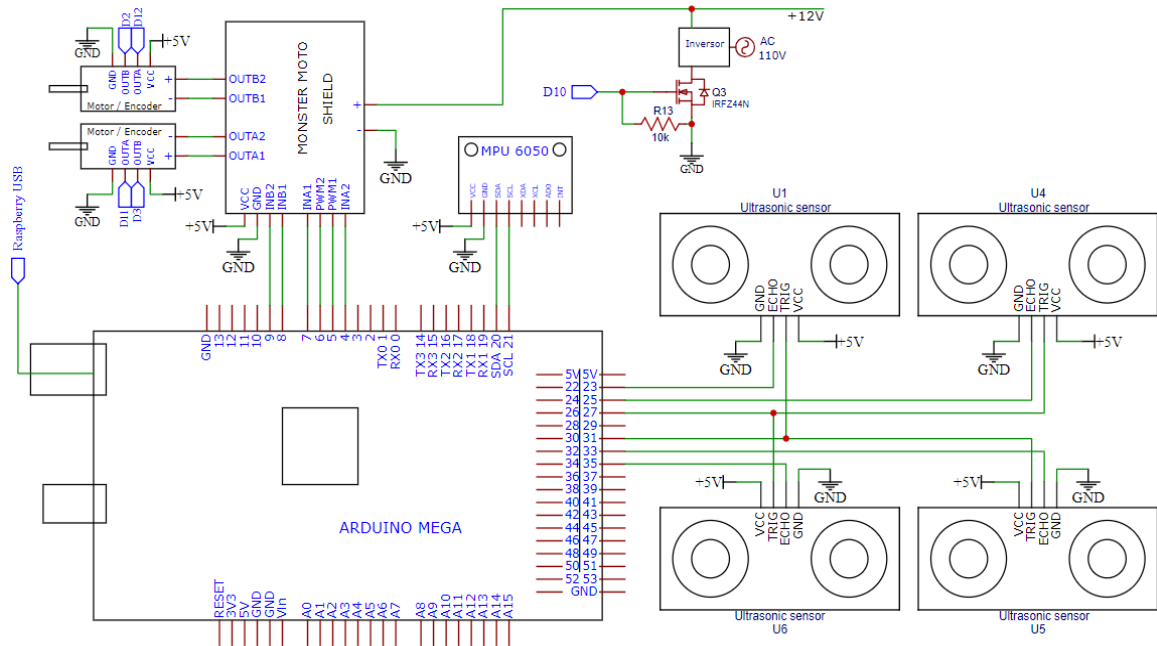
El Arduino nano, que se encuentra en la parte superior, del robot controla la interfaz (la pantalla LCD y los botones), la baliza y el Buzzer que informan los estados del robot al operario, un sensor de límite para indicar cuando los bombillos no se encuentran en la posición indicada y el medidor de voltaje de la batería. Este Arduino se comunica con la Raspberry bidireccionalmente a una frecuencia baja de 2Hz.

Figura 31. Esquema del circuito de la interfaz.



El Arduino mega se encarga de los demás sensores y actuadores: el control PWM de los motores, la lectura de sus respectivos Encoders, el sensor IMU, los sensores de distancia ultrasónicos usados para la detección de obstáculos y el transistor MOSFET para el encendido y apagado de los bombillos ultravioleta. Fue necesario un Arduino mega, debido principalmente a la cantidad de entradas y salidas digitales requeridas.

Figura 32. Esquema del circuito del Arduino Mega en la base.



3.3 SISTEMA DE AUTONOMÍA Y SOFTWARE

Como se ha establecido anteriormente, el procesador principal del robot es una Raspberry pi 3 que es lo que se conoce como un ordenador de placa reducida. Básicamente se tiene un mini computador que usa Linux como sistema operativo y sobre este se instala ROS que actúa como un ambiente de trabajo de código abierto que provee multitud de librerías y herramientas que facilitan la programación de robots.

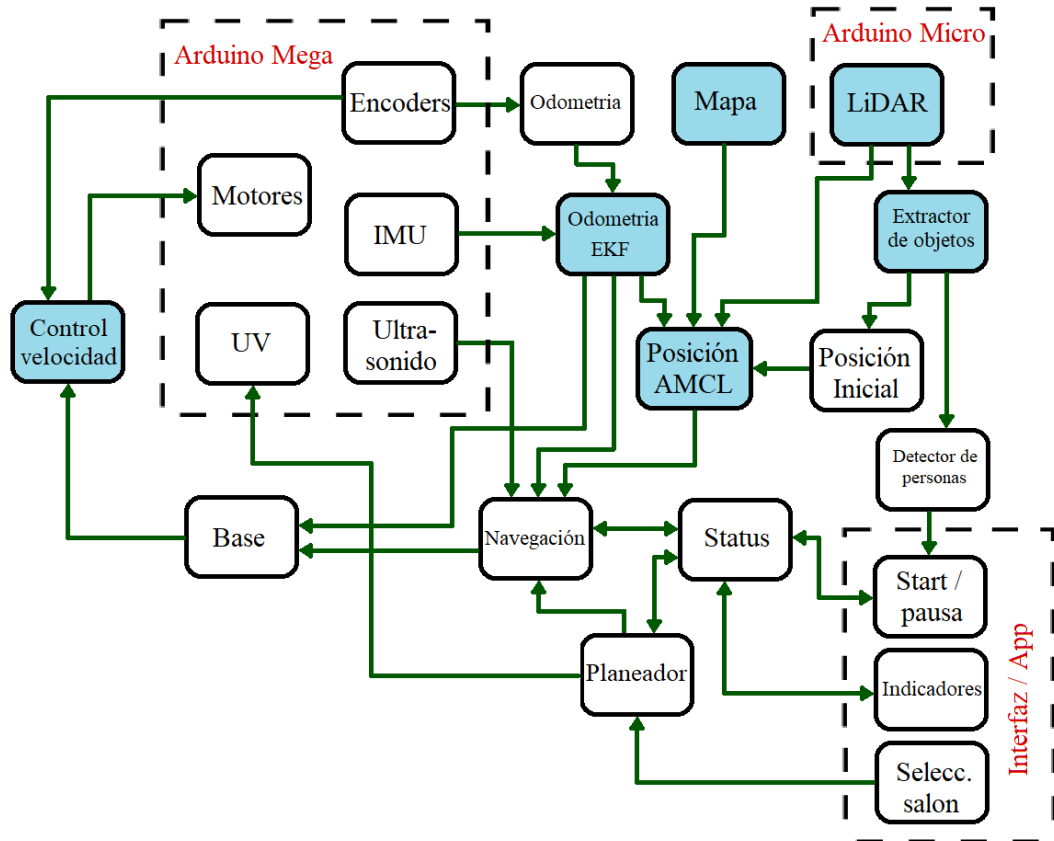
Una de las principales características de ROS es su funcionamiento modular. La mayoría de robots involucran un buen número de procesos que controlan diferentes partes del comportamiento del robot y ROS, en su nivel más bajo, ofrece una interfaz para intercomunicar estos procesos. Esta comunicación se realiza mediante el

mecanismo publicación/suscripción, es decir que cada proceso (denominado como nodo) se suscribe a diferentes mensajes de los que recibe los datos estructurados que necesita para cumplir su función y a su vez publica cierta información en otros mensajes para que se suscriban los nodos que lo requieran.

Otra de las ventajas de este entorno es la cantidad de librerías disponibles (llamadas paquetes) que proveen uno o varios nodos altamente desarrollados y listos para usar en los procesos más comunes y complejos de los robots. Gracias a la comunicación descrita anteriormente, se hace muy fácil combinar estos nodos hechos por terceros con nodos propios según lo requiera la aplicación. Solo es necesario suministrar a cada uno los mensajes que requiere y configurar algunos parámetros para su correcto funcionamiento.

En la figura 33, se muestra de forma simplificada los principales nodos y sus interacciones presentes en el robot y de esta forma resume visualmente la mayor parte del comportamiento del mismo. Los cuadros en blanco son nodos de elaboración propia programados cada uno de forma separada lenguaje Python mientras que los azules son nodos provenientes de paquetes realizados por terceros.

Figura 33. Mapa de los nodos de control del robot.



Este mapa puede parecer bastante confuso a simple vista, pero en las próximas secciones se procederá a explicar con más detalle el funcionamiento de cada uno de estos nodos por separado de tal forma el mapa cobre el significado que merece.

Por ahora, con el fin de dar un vistazo general al mapa, se puede resumir rápidamente de la siguiente manera: de los Encoders y el IMU se obtiene la odometría que es la posición relativa al punto de inicio y sin referencias externas a el robot, esta se combina mediante el algoritmo probabilístico AMCL que se basa en la comparación de lecturas del sensor LiDAR con un mapa previamente cargado para estimar la posición más probable del robot en dicho mapa. El nodo de navegación toma esta posición y envía al nodo de la base la velocidad y dirección necesaria para seguir la ruta dada por el planeador teniendo en cuenta los posibles

obstáculos detectados por los sensores ultrasónicos. En la base, estos valores se convierten en la velocidad real que debe tomar cada motor independientemente y se retroalimenta de la odometría para evitar desvíos. Por último, pasa al nodo de control de velocidad que usa un control PID para dar directamente la señal PWM a los motores. Por otra parte, el nodo status comunica bidireccionalmente los procesos de navegación y planeación con la interfaz y la aplicación móvil, y transmite las ordenes de inicio y pausa y las señales de estado como el buzzer, la baliza y el texto.

3.3.1 Nodo Arduino mega: Todos los procesos dentro del cuadro punteado son en realidad un solo nodo en ROS encargado de leer y organizar los datos de los sensores enviados por el Arduino y devolver las órdenes para los actuadores.

El Arduino por su parte, cuenta con la programación respectiva que le permite simplemente interactuar con todos los componentes y enviar los datos obtenidos a la Raspberry para que sean interpretados en el nodo de ROS. Hace uso de la librería MPU6050_tockn³ para la lectura del IMU MPU6050. Sigue los ejemplos de programación para el driver de los motores Monster Moto shield según el fabricante Sparkfun⁴. Para los Encoders usa interruptores incorporados en los pines 2 y 3 del Arduino que leen los pulsos de alta frecuencia enviados por los sensores de forma independiente a los demás procesos. Para los sensores de ultrasonido se usa la librería NewPing⁵ que permite interactuar con múltiples sensores ultrasónicos y

³ Tockn. (Julio 2019). MPU6050_tockn. Consultado el 09/02/2021 [En línea]. Disponible en: https://github.com/Tockn/MPU6050_tockn

⁴ SparkFun. (s.f.). SparkFun Monster Moto Shield. Consultado el: 18/01/2021 [En línea]. Disponible en: <https://www.sparkfun.com/products/retired/10182>

⁵ Eckel, T. (Julio 2019). NewPing Library for Arduino. Consultado el 09/02/2021 [En línea] Disponible en: <https://playground.arduino.cc/Code/NewPing/>

* Modelo matemático cuatridimensional muy usado para expresar las rotaciones en el espacio

controlar el tiempo de muestreo de cada uno para evitar interferencias entre sus lecturas.

La transferencia de datos se realiza a través del puerto serial USB a una frecuencia de 20 Hz en forma de Strings con los valores separados por comas, tanto Python como el lenguaje de Arduino cuentan con funciones que fácilmente separan y convierten estos datos nuevamente a valores numéricos.

3.3.2 Odometría: Odometría suele definirse como la estimación del desplazamiento de un vehículo con ruedas basada en la rotación de las mismas. Es una medida esencial para la navegación del robot ya que es altamente precisa a corto plazo y permite alta tasa de muestreo. En ROS el mensaje de odometría este compuesto por las tres componentes de desplazamiento y la rotación en cuaterniones* además de sus respectivas derivadas (velocidades lineales y angulares). En este caso, muchos de estos datos serán 0 ya que el movimiento del robot se limita a dos dimensiones. Este nodo se encarga entonces de construir este mensaje a partir de los valores de los dos Encoders basado en las siguientes ecuaciones.

$$D = 2\pi R \frac{\Delta E}{N} \quad (4)$$

Primero se calcula la distancia D recorrida por cada rueda en un intervalo, donde ΔE es el cambio en el valor en los “ticks” del Encoder, R es el radio de la rueda y N el número de ticks por revolución. Luego el cambio en el ángulo viene dado por la diferencia media entre la distancia recorrida por la rueda derecha D_r y la izquierda D_l y la distancia avanzada por el robot hacia adelante es simplemente el promedio entre ambas.

* Modelo matemático cuatridimensional muy usado para expresar las rotaciones en el espacio

$$\Delta\theta = \frac{D_r - D_l}{2} \quad (5)$$

$$D_m = \frac{D_r + D_l}{2} \quad (6)$$

Finalmente, el desplazamiento hacia “adelante” se divide en sus componentes x, y según el ángulo calculado, donde el origen es el punto en el que inicia el robot.

$$\Delta x = D_m \cos(\theta) \quad (7)$$

$$\Delta y = D_m \sin(\theta) \quad (8)$$

La odometría basada en Encoders tienen también dos grandes limitaciones. En primer lugar, el deslizamiento. Solo con los Encoders el robot es incapaz de diferenciar cuando el giro de las ruedas realmente se traduce en desplazamiento y por lo tanto si la rueda pierde tracción momentáneamente la odometría se desfasa. En segundo, debido al mismo concepto de odometría que consiste en integrar pequeños movimientos a través del tiempo sin ningún punto de referencia externo, los errores se acumulan rápidamente y se vuelve muy poco confiable a largo plazo, en especial cuando el ángulo se ve afectado.

Por estas razones, los siguientes nodos buscan mejorar estas problemáticas incluyendo otros sensores en la estimación de la posición.

3.3.3 Odometría EKF (IMU): Este nodo hace parte del paquete `robot_pose_ekf` creado por Wim Meeussen⁶ y busca precisamente integrar otros sensores en la estimación de la odometría, en este caso el IMU. Este sensor es simplemente la combinación de un acelerómetro y un giroscopio, del cual, con algo de matemática en la librería de Arduino, se obtiene una buena medición del ángulo del robot y una medición ruidosa de la aceleración a causa de la gravedad.

⁶ Meeussen, W. (Marzo 2012). `robot_pose_ekf`. Consultado el: 25/03/2021 [En línea] Disponible en: http://wiki.ros.org/robot_pose_ekf

Con estos datos se puede elaborar entonces un mensaje de odometría con una buena estimación del ángulo que, a diferencia de la del Encoder, no se ve afectada por el deslizamiento y es ahí donde entra este nodo a combinar ambos mensajes mediante un filtro extendido de Kalman (EKF). Dicho filtro compara no las posiciones absolutas sino los cambios en cada intervalo teniendo en cuenta la incertidumbre dada para cada una de las mediciones. De esta forma, para la componente lineal de la odometría EKF se le da mayor peso a los Encoders y para la angular al IMU.

Ahora se tiene una odometría mucho más precisa que combina lo mejor de ambos sensores y se complementan donde fallan. Sin embargo, se continua sin un punto de referencia externa y, por lo tanto, en menor medida, la tendencia a deteriorarse con el tiempo.

3.3.4 LiDAR: Para el interactuar con el LiDAR se usa un Arduino micro debido a que tiene dos puertos seriales que le permiten actuar de puente para enviar los datos recibidos directos del sensor a la Raspberry a través de USB, por lo tanto, la comunicación es diferente a la descrita para el Arduino mega, ambos controladores reciben los mismos datos en forma de Bytes y cuentan con su librería para interpretarlos.

En el Arduino se usa el programa creado por James LeRoy⁷ y se interpretan los datos únicamente para controlar con estos la velocidad del motor mediante un control PID simple. En la Raspberry son recibidos por un nodo de ROS del paquete `xv_11_laser_driver`⁸ donde se interpretan y se construye el mensaje respectivo. Dicho mensaje corresponde a los 360 valores medidos de distancia que permiten construir una imagen como muestra la figura 34, cada uno con su respectiva intensidad o confiabilidad en la lectura representada por el color. EL mensaje se

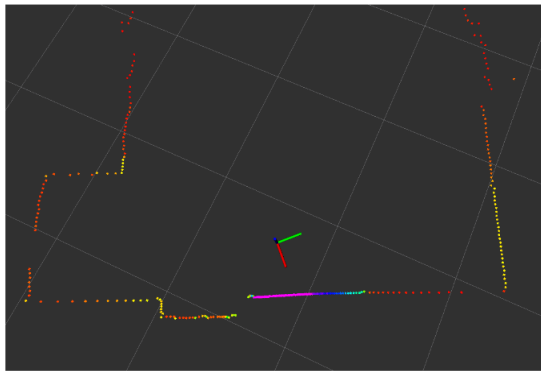
⁷ LeRoy, J. (2021). XV Lidar Controller. Consultado el: 28/03/2021 [En línea] Disponible en: https://github.com/getSurreal/XV_Lidar_Controller

⁸ Perko, E., Rockey, C., Agrawal, R., Okay, S. `xv_11_laser_driver`. Consultado el: 28/03/2021 [En línea] Disponible en: http://wiki.ros.org/xv_11_laser_driver

publica a la frecuencia de rotación del sensor, en este caso 3.33 Hz aproximadamente.

En la etapa de pruebas finales se ha detectado una problemática en la lectura de datos del LiDAR por parte de ROS que reduce la frecuencia de publicación del mensaje o produce mensajes incompletos. Dicho error no ha podido ser diagnosticado en su totalidad, aunque puede atribuirse a alguna falla en la recepción de datos de la Raspberry o a la interpretación por parte del paquete de ROS y ocurre de manera aparentemente aleatoria. Pese a ello, aunque este error puede generar cierta confusión en el robot es poco probable que esta conlleve a errores graves en su comportamiento.

Figura 34. Nube de puntos arrojados por el sensor LiDAR.



3.3.5 Mapa: El mapa usado para el algoritmo AMCL es simplemente una imagen PGM (escala de grises) que representa cada punto del ambiente tal como es visto por el LiDAR, con una resolución estándar de 5 cm² por punto. Cada pixel se representa con uno de tres colores: negro, gris claro o gris oscuro dependiendo si está ocupado por algún objeto, libre o indeterminado respectivamente. Este mapa se construye previamente con ayuda de otros procesos con ROS, dicho proceso es mejor descrito en la sección 3.3.15. En este nodo del paquete map_server se encarga solo de tomar la imagen adecuada según la selección de salón del operario, interpretar dicho mapa y organizarlo en un mensaje que pueda ser usado por los demás nodos que lo requieran.

3.3.6 AMCL: AMCL (Aaptative MonteCarlo Localización) es un algoritmo probabilístico que predice la posición y orientación de un robot usando un filtro de partículas. Dado un mapa del ambiente del robot y un sensor que provea información de ese ambiente, el algoritmo genera un numero N de partículas de forma aleatoria que representan posiciones y localizaciones probables. Luego evalúa para cada una de ellas que tanto ajusta las mediciones con el mapa y con base en esto descarta las menos adecuadas y genera más partículas cerca de las mejores. De esta manera, las partículas convergen hacia la posición más probable del robot.

Las siguientes imágenes se toman directamente de la fase de pruebas del robot y permiten observar mejor el funcionamiento del algoritmo:



Estado inicial

En un primer momento se generan N partículas aleatorias representadas como flechas repartidas por toda el área libre del mapa.



Estado 1

Cuando la odometría indica movimiento el algoritmo se actualiza y las partículas se van acumulando en grupos cada vez más pequeños.



Estado 2

Luego de varios movimientos la mayoría de grupos desaparecen y la posición del robot empieza a converger hacia los puntos más probables. Aquí se hace evidente el problema de la simetría, varios puntos del mapa pueden ser tan similares que el algoritmo no será capaz de diferenciar entre estos.



Estado 3

Finalmente, al comparar nuevas lecturas se logra mantener únicamente un punto que debe representar la ubicación real del robot, entre más concentradas se encuentren las partículas se puede decir que el algoritmo entrega una posición con menos incertidumbre. Cabe resaltar también que en este punto el algoritmo reduce el número N de partículas para reducir el consumo de recursos.



Estados siguientes

Después de alcanzar la localización precisa en el mapa, cuando el robot continúa en movimiento las partículas se mueven con él y se generan nuevas partículas más dispersas alrededor que representan la incertidumbre en la localización producto del movimiento. Es aquí

donde el algoritmo juega su papel más importante, la localización que entrega este nodo viene dada en gran parte por la odometría EKF y el algoritmo AMCL es entonces el encargado de compensar la incertidumbre que se acumula en la odometría al pasar el tiempo para finalmente obtener la posición y orientación precisa y corregida a partir del ambiente.

Es importante resaltar también que el algoritmo resulta muy efectivo para corregir esta incertidumbre y pequeños errores derivados de la odometría, pero para errores mayores como fallos de los sensores o choques no suele reaccionar rápidamente. Básicamente el algoritmo no puede predecir fácilmente posiciones que se alejen de la zona de incertidumbre representada por las partículas.

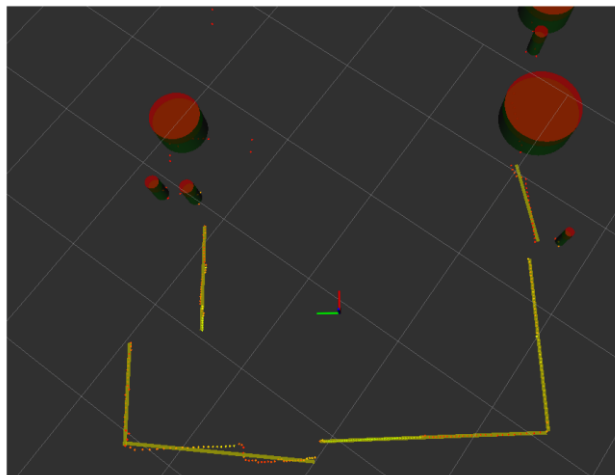
Otro problema de este algoritmo consiste en la simetría o las similitudes entre diferentes partes del mapa como se aprecia en el estado 2. Por lo tanto, dada también la naturaleza probabilística del algoritmo, no se garantiza que el estado 3 converja hacia la posición real del robot y esto puede producir fallos catastróficos en la navegación. La solución a este problema es proveer al algoritmo con la posición inicial real del robot de tal forma que comience correctamente en un punto cercano al estado 3. Esta será entonces la función de los dos siguientes nodos a analizar.

Finalmente, otro aspecto fundamental de este nodo, es que la frecuencia de actualización está limitada por el LiDAR, es decir, que tiene un promedio de 3.3 Hz, bastante inferior a la de la odometría EKF que se limita en 20 Hz. Por lo tanto,

aunque la localización dada por este nodo sea vital para la correcta navegación del robot, la odometría sigue siendo la más utilizada para los demás procesos a continuación.

3.3.7 Extractor de objetos: Es un paquete de ROS desarrollado por Mateusz Przybyla⁹ que toma las lecturas del LiDAR e implementa una serie de algoritmos que permiten agrupar los diferentes puntos geoméricamente y construir con ellos líneas o círculos que son mucho más adecuados para realizar operaciones subsiguientes.

Figura 35. Extractor de obstáculos.



3.3.8 Posición inicial: Este nodo programado por los autores busca solucionar la problemática de la posición inicial del algoritmo AMCL expuesta anteriormente. La solución más común suele ser simplemente restringir la dispersión de las partículas en un área de inicio probable. Sin embargo, esto aun requiere un gran esfuerzo por

⁹ Przybyla, M. (Febrero 2017). Online Detection and Tracking of 2D Geometric Obstacles from LRF Data. Consultado el: 12/04/2021 [En línea] Disponible en: https://github.com/tysik/obstacle_detector/blob/master/resources/ObstacleDetector.pdf

parte del algoritmo AMCL y, por lo tanto, o mantiene una probabilidad de error considerable o reduce mucho esta área complicando al operario con la posición y orientación del robot.

Por esta razón, se decide desarrollar este nodo que busca calcular de manera exacta la localización inicial restringiendo lo menos posible el área y la orientación en la que el operario debe iniciar el robot. Gracias al paquete extractor de objetos, este cálculo se realiza fácilmente y con excelente precisión.

A continuación, se describe el algoritmo y las ecuaciones implementadas:

Paso 1:

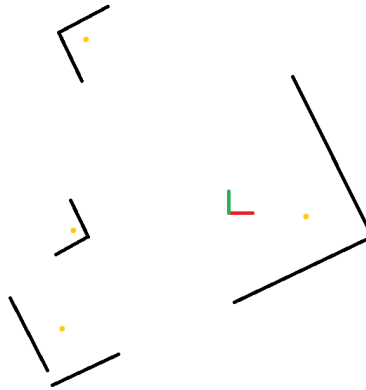
El nodo extractor de objetos entrega una lista con los segmentos de línea ordenados según son observados por el LiDAR, de forma opuesta a las manecillas del reloj dando su punto inicial y final con respecto al centro del robot. Con estos datos se halla para cada segmento el ángulo (α) que forman con el eje x del robot mediante la siguiente formula:

$$\alpha = \arctan\left(\frac{y_f - y_i}{x_f - x_i}\right) \quad (9)$$

Paso 2:

Teniendo todos los ángulos se buscan aquellos segmentos consecutivos de la lista ordenada que formen entre ellos un ángulo de $90 \pm 5^\circ$, es decir las esquinas del salón. Desde el punto de vista del robot se puede describir este estado como se muestra en la figura 36.

Figura 36. Búsqueda de esquinas.



Paso 3:

Ahora, de todas estas esquinas se debe identificar la más cercana al robot. Para esto se calcula el punto promedio de cada uno de estos pares de segmentos usando sus puntos iniciales y finales. Dichos puntos son representados aproximadamente por los puntos amarillos en la figura 36 y se calculan con la siguiente formula:

$$\bar{x} = \frac{\left[\frac{(x_{f1} + x_{i1})}{2} + \frac{(x_{f2} + x_{i2})}{2} \right]}{2} \quad (10)$$

$$\bar{y} = \frac{\left[\frac{(y_{f1} + y_{i1})}{2} + \frac{(y_{f2} + y_{i2})}{2} \right]}{2} \quad (11)$$

Paso 4:

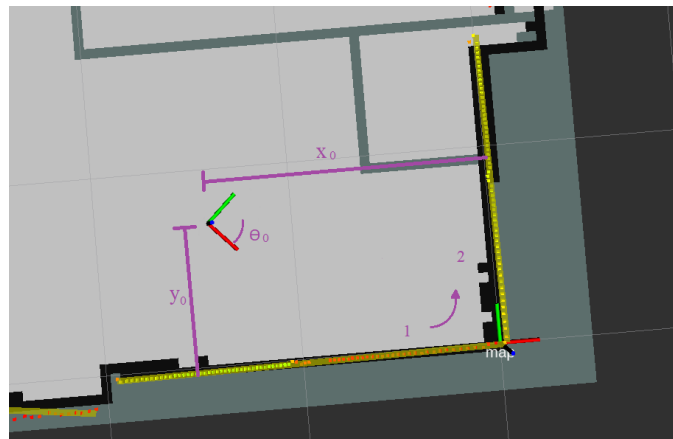
Teniendo ya los dos segmentos que representan la esquina más cercana al robot, es hora de comparar con el mapa para obtener la deseada localización inicial. Para conveniencia, el punto de origen del mapa se ubica en la esquina analizada y por lo tanto la posición (x_0, y_0) corresponde simplemente a las distancias normales del robot a los segmentos, y_0 para el primer segmento de la lista y x_0 para el segundo. La distancia normal se calcula con la siguiente ecuación:

$$d = \frac{|(x_f - x_i)(0 - y_i) - (y_f - y_i)(0 - x_i)|}{L} \quad (12)$$

Donde L representa la longitud del segmento.

Por otra parte, la orientación Θ_0 la dan los ángulos de los segmentos calculados en el paso 1. Para el primer segmento $\Theta_0 = \alpha$, para el segundo $\Theta_0 = 90-\alpha$. finalmente ambos valores obtenidos se promedian para mejorar su precisión.

Figura 37. Cálculo de la posición con base a la esquina.



De esta manera se obtiene matemáticamente la localización y orientación inicial del robot restringiendo la ubicación para el operario únicamente a que la esquina más cercana corresponda con la indicada.

Además de esta restricción, el algoritmo implementado presenta las siguientes limitaciones para su implementación general:

1. El salón debe contar con una esquina de 90 grados suficientemente definida y cercana al punto de inicio del recorrido (la cual es lo más común en los salones a implementar). Si no se cumple esta condición, se debe corregir el

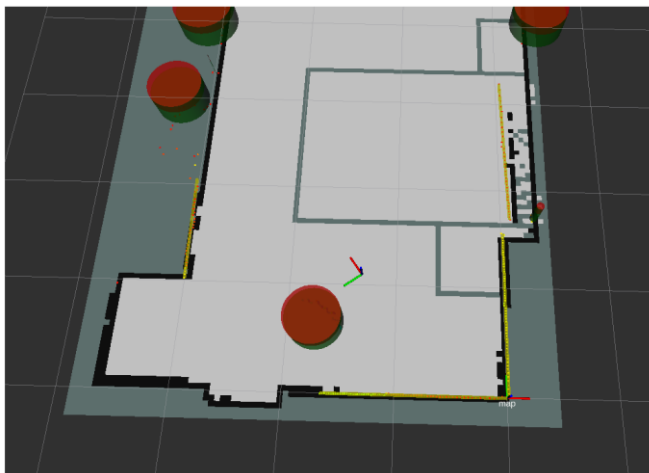
algoritmo para trabajar con alguna otra alguna otra geometría característica que se tenga disponible.

2. A la hora de realizar estas mediciones, la esquina puede estar parcialmente bloqueada por algún objeto o por el operario, pero debe respetar como mínimo una distancia mayor de 40 cm a cada lado para que el nodo extractor de objetos lo reconozca como un segmento. Estos 40 cm no deben ser necesariamente en la esquina con tal de que dos segmentos consecutivos formen 90 grados. Por lo tanto, el punto de intersección puede estar bloqueada sin problemas por un objeto que no forme una línea recta, tal es el caso de una persona.

3.3.9 Detector de personas: Este nodo es una adición de seguridad para prevenir que el robot comience su recorrido si se detecta una posible persona cerca. El extractor de objetos agrupa los puntos dados por el LiDAR y si un determinado grupo no forma una línea recta lo representa con un círculo dando la posición del centro y su diámetro.

La función de este nodo es simplemente activar una señal si se detecta un objeto circular de más de 30cm de diámetro a una distancia inferior a 2 metros, como se muestra en la figura 38. Dicho objeto tiene grandes posibilidades de representar a una persona, pero también puede tratarse de cualquier otro objeto similar o una característica propia del mapa y por lo tanto los falsos positivos ocurren frecuentemente. La señal enviada no evita totalmente el inicio del recorrido, solo previene al operario de la detección y le pide una confirmación adicional para que este verifique que el salón se encuentra vacío.

Figura 38. Representación de una persona detectada.



3.3.10 Planeador: Este nodo contiene la ruta que sigue el robot en el salón como una secuencia de los puntos (x,y) hacia los que debe desplazarse en cada etapa del recorrido. Aquí se almacenan todas las rutas para cada uno de los salones y se envía la ruta correspondiente hacia el nodo de navegación según la selección de salón realizada por el operario.

Es en este nodo también donde se envía la señal de encendido y apagado de los bombillos UV al nodo del Arduino mega según el robot avanza en su recorrido por el salón.

3.3.11 Navegación: El nodo de navegación es el encargado de controlar el movimiento general del robot. Recibe del nodo planeador un punto en el mapa al cual dirigirse y del nodo AMCL la posición y orientación en ese mapa, con esto calcula cuanto debe girar y cuanto desplazarse y aplica un control proporcional sencillo para controlar la velocidad hasta alcanzar este punto. La figura 39 explica con un ejemplo el funcionamiento de este nodo.

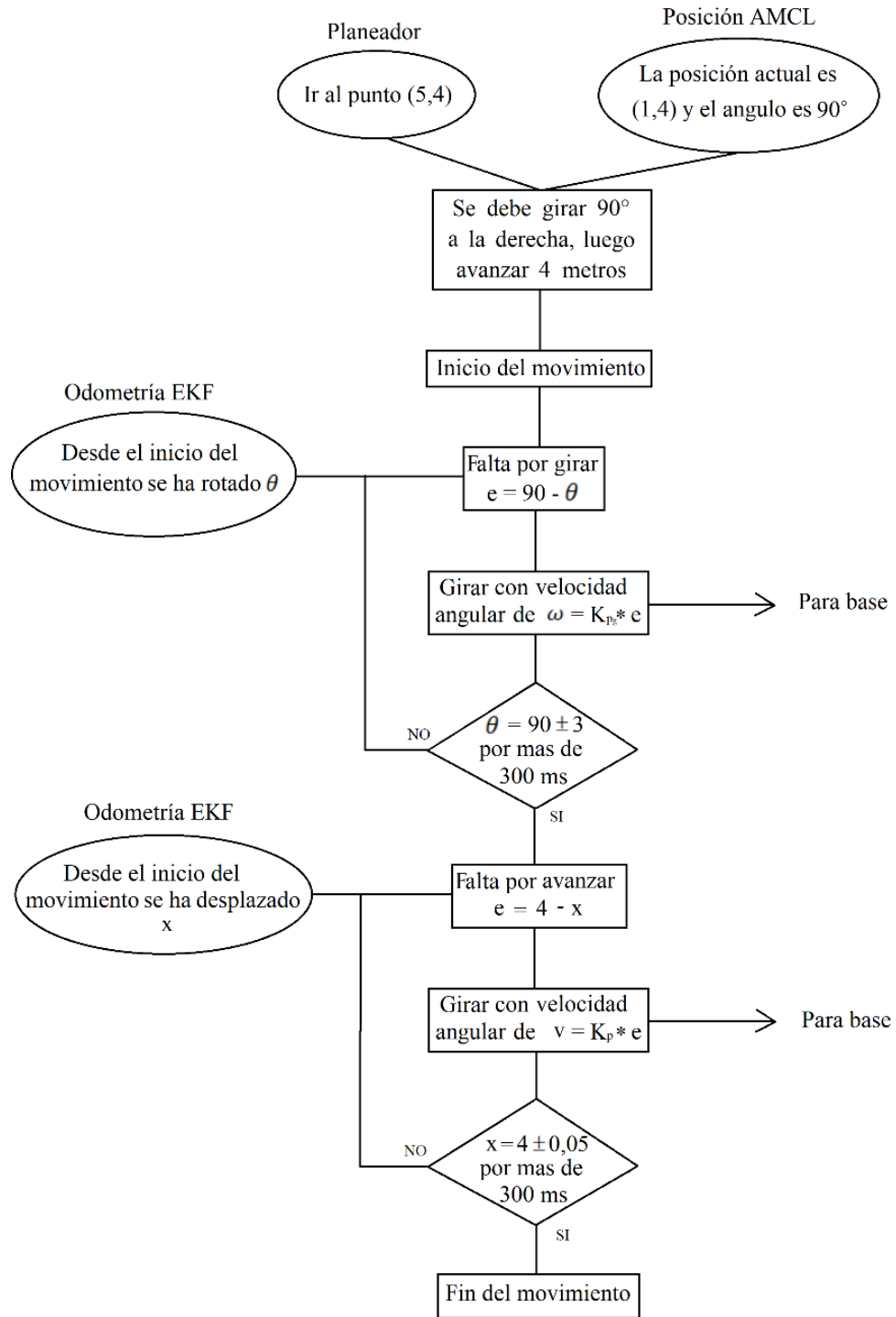
ROS cuenta con varios paquetes que implementan fácilmente algoritmos mucho más robustos de planeación y seguimiento de trayectorias ampliamente utilizados

en robot y aplicaciones similares a la presente. Sin embargo, la robustez de dichos algoritmos consiste en su capacidad para encontrar una trayectoria óptima, basada en el mapa y en las lecturas del LiDAR, para ir de un punto A a un punto B aunque esta no sea la misma siempre. Y, por lo tanto, en este caso en que el robot debe recorrer una trayectoria fija y bastante precisa para la desinfección del salón y cerca de obstáculos (mesas) estos algoritmos no son confiables y se decide implementar uno propio.

Estos algoritmos suelen tener un alto nivel de complejidad para su desarrollo, sin embargo, debido al tipo de recorrido esperado para el robot en el salón, los movimientos se pueden reducir a los más básicos: avance y retroceso en línea recta y giros sobre su propio eje, es decir, sin movimientos curvilíneos. De esta forma se simplifican enormemente los cálculos necesarios en la navegación y elimina la necesidad de algoritmos pesados de planeación de trayectorias.

En este nodo también se recoge la información de los sensores ultrasonido y se evalúa en un ciclo aparte al mostrado en el ejemplo si las detecciones representan un obstáculo en la trayectoria del robot. Si tal es el caso, este nodo envía la alarma, detiene todos los procesos y espera por la confirmación del operario de que el objeto fue removido para retomar el recorrido desde donde se detuvo. Así mismo este nodo puede recibir la señal de pausa proveniente de la aplicación para entrar en este mismo modo de espera.

Figura 39. Proceso del nodo de navegación.



3.3.12 Base: Este nodo recibe los comandos del nodo de navegación que representan la velocidad lineal del robot en m/s y la velocidad angular en rad/s y transforma estos datos en velocidades para las ruedas por separado (izquierda y derecha). Para ello se usan las siguientes ecuaciones:

$$v. derecha = \frac{(2 * v. lineal + v. angular * ancho)}{2 * radio de rueda} + correccion \quad (13)$$

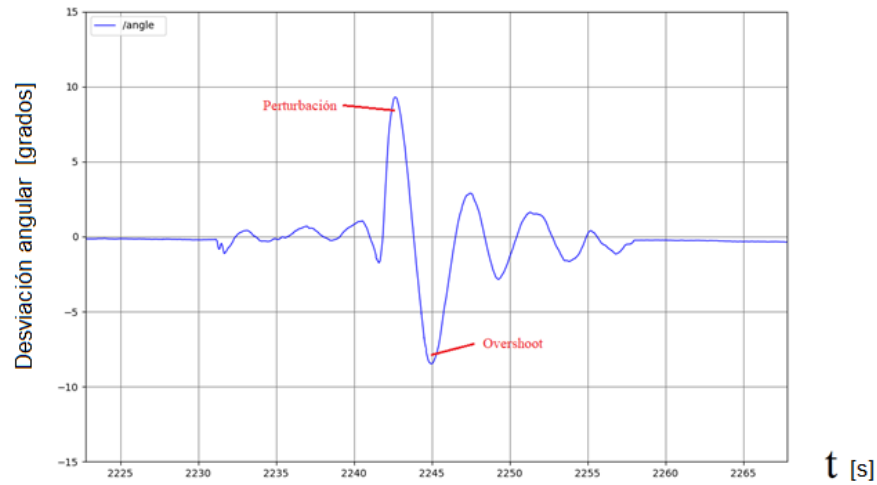
$$v. izquierda = \frac{(2 * v. lineal - v. angular * ancho)}{2 * radio de rueda} - correccion \quad (14)$$

Donde *ancho* representa la separación entre las ruedas. De esta forma se obtienen las velocidades para cada rueda en rad/s y pasan al siguiente nodo para convertirse en comandos reales para los motores.

La otra función de este nodo involucra al término adicional *corrección* que aparece en las ecuaciones con signo contrario para cada rueda y por lo tanto hace girar al robot. Este término busca mantener a el robot alineado en su trayectoria mientras avanza en línea recta y es de gran importancia para que el algoritmo de navegación mostrado en el punto anterior funcione adecuadamente.

Para obtener este término se implementa entonces un control PID que controla el ángulo del robot retroalimentado por la odometría EKF, de tal forma que se mantenga en el ángulo dado por la por el nodo de navegación. La figura 33 tomada desde la herramienta de visualización de ROS *rqt_plot* donde se grafica ángulo vs tiempo corresponde a la respuesta de este controlador frente a una perturbación considerable en el ángulo.

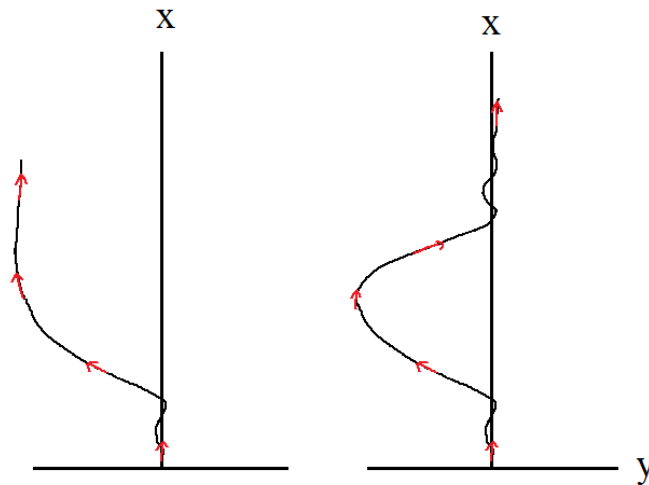
Figura 40. Respuesta a una perturbación en la dirección del robot.



Este controlador se encarga no solo de corregir el ángulo si no que de manera implícita corrige también la desviación respecto a la línea en la cual comienza su movimiento (una variable mucho más difícil de medir) y para esto el overshoot es en este caso una característica deseable en la respuesta a las perturbaciones. En la figura 41 se ilustra mejor este comportamiento representando la trayectoria x y que recorre el robot durante esta respuesta. A la izquierda un controlador sin overshoot que rápidamente corrige el ángulo, pero ahora el robot sigue avanzado en la misma dirección por un camino paralelo y por lo tanto erróneo. A la derecha el controlador implementado que responde como se muestra en la figura 40. El cual corrige más lentamente el ángulo y produce oscilaciones, pero le permite al robot regresar a la trayectoria previa a la perturbación.

De la teoría de control se sabe que la componente integral del PID se vincula fuertemente al overshoot y por lo tanto es la que predomina este controlador. El proceso de calibración se realiza manualmente y gracias a la herramienta `rqt_reconfigure` las constantes pueden editarse en tiempo real mientras que la respuesta se visualiza con `rqt_plot` como en la figura 40.

Figura 41. Representación en coordenadas x y de la respuesta a la desviación.



3.3.13 Control de velocidad: El ultimo nodo del grupo de navegación es el encargado controlar la potencia de los motores de la base, es decir, la señal PWM para mantener la velocidad indicada por el nodo anterior. Para ello implementa un control PID (uno para cada lado de la base) que se retroalimenta directamente por los Encoders.

Este nodo hace parte del *paquete* `ros_control` e involucra una serie de procesos y herramientas de ROS que modelan las partes móviles de los robots (ruedas, brazos, actuadores) y permite controlar su posición, fuerza o, en este caso, velocidad y limitar otros valores como aceleración o esfuerzo del actuador. Además, al igual que en el nodo anterior, su calibración se realiza de forma sencilla con ayuda de las herramientas de visualización y reconfiguración de parámetros de ROS.

3.3.14 Interfaz, app y status: Por último, los nodos restantes son los encargados de toda la interacción robot-operario. Primero `status` es un nodo sencillo que permite la comunicación bidireccional entre los demás nodos: la navegación, el planeador, la interfaz y la app de tal forma que todos se sincronicen respecto al estado del robot.

La interfaz y la app son dos nodos separados que realizan la misma tarea e interactúan con ROS de la misma manera, pero cada uno se encarga de sus respectivos componentes. De esta forma el operario tiene dos opciones para interactuar con el robot: la app que le permite tener control a distancia a través de WI-FI y funcionalidades adicionales o la interfaz si no se cuenta con celular o conexión a internet.

El nodo de la interfaz interactúa con el Arduino nano de una manera similar a la que lo hace el nodo del Arduino mega, es decir, con un String de valores separados por comas a través del puerto serial USB. El Arduino recibe de la Raspberry el número del estado dado por el nodo status y según este la pantalla muestra la información respectiva y la baliza y el buzzer se activan o desactivan. Desde la interfaz, basado también en el estado, se puede seleccionar el salón, dar la señal de inicio, de reanudar después de un obstáculo y de cancelar el recorrido. Además, es la encargada de medir el voltaje de la batería y calcular el porcentaje disponible y así mismo leer el sensor de límite que indica si el brazo de los bombillos fue desplegado.

Para la app se hace uso de la plataforma Blynk enfocada en lo que se conoce como internet de las cosas (IoT) y es gratuita para aplicaciones no comerciales. Cuenta con compatibilidad con la Raspberry y librerías para Python que fácilmente pueden adaptarse con las funcionalidades de ROS. La app se construye desde el celular de forma gráfica simplemente arrastrando los widgets como interruptores, espacios de texto, deslizadores, etc. Y en la Raspberry se programa para que cada widget cumpla con su función deseada. Desde la app se pueden realizar las mismas funciones que desde la interfaz, pero se tiene información del estado del robot en todo momento y la posibilidad de pausar, reanudar o cancelar el recorrido a distancia. Además, con ayuda del widget de video el operario puede visualizar el comportamiento del robot a través de la cámara.

El uso de esta app requiere que tanto el robot como el celular se encuentren conectadas a una red WI-FI, Para la visualización del video de la cámara se requiere además que la red WI-FI sea la misma para ambos mientras que las demás

funciones pueden ser usadas con cualquier red y desde cualquier lugar del mundo. Para acceder a la app de este robot, el operario debe descargar en su celular la aplicación de Blynk y a través de esta escanear un código QR provisto específico para el robot.

Para más información de la funcionalidad de la app y la interfaz véase el manual del operador en la sección 5.

3.3.15 Proceso de mapeo: Este proceso no hace parte de los nodos mostrados en el mapa de la figura 33 porque sucede antes de que comience el comportamiento principal del robot. El objetivo es crear el mapa del salón que será usado por el robot en los nodos anteriormente explicados y para ello se hace uso del paquete gmapping¹⁰ que se encarga de construir el mapa a partir de la nube de puntos dada por el LiDAR usando un filtro de partículas. El algoritmo que se aplica en este nodo es conocido como SLAM (Mapeo y localización simultanea) y permite al robot moverse a través de un espacio hasta ahora desconocido mientras va mapeando sus alrededores y actualizando su localización.

Para obtener el mapa del salón entonces se sigue el siguiente proceso:

- Mover el robot de forma manual con ayuda de un control remoto acercándolo a todos los puntos del salón y verificando que queden bien registradas las paredes y los objetos estáticos presentes.
- Revisar que las medidas que muestra el mapa se ajusten con las medidas reales del salón
- (opcional) Con ayuda de una herramienta de dibujo editar el mapa obtenido que consiste en una imagen en escala de grises para eliminar residuos que puedan quedar del proceso de mapeo y mejorar la

¹⁰ Grisetti, G., Stachniss, C., Burgard, W. (s.f.). OpenSlam. Disponible en: <https://openslam-org.github.io/gmapping.html>

aparición para su visualización en el proceso de programación de rutas y pruebas.

- Configurar la información del mapa para ubicar el punto (0,0) del mapa en la esquina de inicio del robot como se indica en la sección 3.3.8.

4 PRUEBAS

El proceso de programación y ajuste de parámetros de software se realizó a través de toda la duración de este proyecto, durante este, se destacan tres etapas de pruebas que permiten validar tanto el hardware como el software realizado hasta el momento y que concluyen con las pruebas finales en los salones de clase.

4.1 ETAPA DE PRUEBA INICIAL.

Para iniciar con el desarrollo del prototipo se construye un pequeño modelo mostrado en la figura 42 que hace uso de los mismos controladores, motores y sensores básicos de navegación que se usaran en el prototipo final. En esta etapa se configuran todos los aspectos principales de software iniciando con la configuración del sistema operativo, controladores para los sensores y actuadores y se estructura la mayor parte del comportamiento del robot explicado en la sección 3. Con esta etapa de pruebas se validan los componentes seleccionados y la metodología para el software del robot incluyendo los principales algoritmos de mapeo, movimiento de la base y navegación obteniendo resultados satisfactorios.

Figura 42. Primer modelo para pruebas de software iniciales.



4.2 ETAPA DE VALIDACIÓN DE SOFTWARE Y HARDWARE.

La segunda etapa se realizó ya con el prototipo final construido, para ello se ubica el robot en un cuarto cerrado y se tiene el objetivo de ejecutar una ruta similar a la que se espera en los salones de clase. En esta etapa se elaboran y se prueban los nodos restantes que corresponden principalmente a la navegación y posición además de las capas de interfaz. En la figura 43 se observa el robot en la habitación y la ruta programada que corresponde a la secuencia que se repite 4 veces para completar la ruta.

Figura 43. Habitación y ruta para la segunda etapa de pruebas.



Al finalizar esta etapa se da por concluida la elaboración del robot y por lo tanto se realizó una sesión de pruebas generales para evaluar el desempeño de todos sus componentes tanto de hardware como de software una vez calibrados al máximo todos sus parámetros. Estas pruebas consisten en repetir numerosas veces la ruta mencionada anteriormente y resaltar las principales fuentes de error encontradas y se elaboran las siguientes conclusiones.

- Conclusiones:
 - El robot ejecuta adecuadamente la ruta en los más de 20 recorridos realizados con una precisión adecuada.
 - No se encuentran problemas en la navegación ni en el movimiento general del robot, pero existe un nivel de vibración y ruido mayor al deseado producto del movimiento de la base que no llega a afectar su desempeño.
 - El sensor LiDAR presenta un error recurrente que disminuye su frecuencia efectiva de lectura, lo cual puede producir algunas demoras en el inicio del movimiento y la ubicación, pero no llega a afectar el desempeño de la navegación.
 - Los sensores de detección de obstáculos presentan también una baja probabilidad de lecturas erróneas, falsos positivos y falsos negativos, pese a los filtros de software implementados y calibrados. Esto genera que el robot se detenga aunque no exista ningún obstáculo que lo bloquee o que demore en detenerse y llegue a chocarse. Sin embargo, al reanudar el movimiento tras la pausa en ambos casos, el robot continúa su movimiento adecuadamente por lo que no se ve afectada la navegación, pero sí una parte importante de la funcionalidad y la autonomía del robot.

4.3 ETAPA DE PRUEBAS EN CAMPO REAL

Finalmente, con el robot correctamente validado por las pruebas anteriores se procede a realizar pruebas directamente en salones de la Universidad Industrial de Santander. Debido a las regulaciones de ingreso a la institución y factores sociales se realizaron solo las pruebas necesarias en este ambiente con objetivos claros.

Figura 44. Robot dentro del salón de clases.



4.3.1 Primera visita:

- **Objetivos:**
 - Verificar el desempeño de la parte motriz en el suelo de los salones del edificio de laboratorios de la escuela de Ingeniería Mecánica de la Universidad Industrial de Santander.
 - Realizar el mapeo del salón disponible para practicas brindado por las directivas de la institución.
 - Ajustar el mapa generado por el sensor laser en el software.
 - Construir la ruta de limpieza del salón.

- **Resumen de la jornada**

El salón se encontraba con bastante nivel de polvo y residuos de construcción en el piso, por lo que primero se procede a hacer una limpieza del suelo para tenerlo en mejores condiciones.

Luego de la limpieza general se procede a realizar las pruebas de movimiento del robot y el mapeo del salón para la navegación

Luego del escaneo se verifica el mapa en el software y con este se programa la ruta a realizar y tras una serie de ajuste de parámetros se inicia la evaluación del comportamiento del robot obteniendo las siguientes conclusiones.

- **Conclusión:**
 - El piso del aula presenta pequeños desniveles y grietas, por lo que en ciertos puntos las ruedas de apoyo presentan pérdida de contacto con el suelo y el movimiento se ve afectado ya que las ruedas motrices perdían tracción llegando incluso a detener el robot.
 - El suelo es también mucho más liso que el de las pruebas anteriores y sumado a los niveles de polvo y partículas las ruedas motrices tienden a patinar repetidamente dañando la localización del robot.
 - La navegación se ve seriamente afectada debido principalmente a los fallos en la base, además el gran tamaño del salón y la simetría de las paredes dificulta que la ubicación dada por el LiDAR corrija efectivamente los errores anteriores mediante el algoritmo AMCL (ver sección 3.3.6)
 - El mapa del salón se generó correctamente.
 - La altura y distancia de las lámparas es la adecuada para la limpieza de las superficies.
 - Se anotan las observaciones del desempeño de la base para una próxima visita luego de llevar a cabo arreglos y adaptaciones.

4.3.2 Segunda visita:

- **Objetivos**
 - Comprobar las mejoras de los arreglos realizados previamente.
 - Realizar la ruta de limpieza del salón.
 - Verificar desempeño general del prototipo.

- Resumen de la jornada:

El salón, al igual que la primera jornada, se encontraba en condiciones no adecuadas para el movimiento del prototipo, por lo que se procede a una limpieza general del aula.

Se verificó que los arreglos hechos previamente hayan generado un cambio positivo en el desempeño del prototipo.

Posteriormente, se continua con la evaluación de la navegación del robot a través de la ruta planteada.

Tras elaborar la ruta se procede a realizar pruebas y ajuste de medidas de recorridos, como velocidades y posiciones de giro y avance.

- Conclusión:

- Se logro culminar con éxito la ruta programada para el salón que permite su desinfección en un promedio de 8 minutos y 30 segundos desde que se da la orden de inicio.
- Los arreglos realizados le permitieron al prototipo mejorar significativamente el movimiento dentro del salón ya que el sistema de suspensión mejora notablemente la tracción de las ruedas, así como su respuesta a desniveles. Sin embargo, sigue existiendo una tendencia al deslizamiento y las desviaciones que afecta la precisión en el desplazamiento que al acumularse llega a generar fallos catastróficos en la navegación del robot.
- Otra fuente de error producto de las desviaciones mencionadas anteriormente es que el robot no cuenta con sensores de distancia a la altura de las mesas y en algunas ocasiones son estas las que llegan a afectar su recorrido sin que se activen a tiempo los sistemas de detección de obstáculos. Se plantea entonces la necesidad de incluir estos

sensores y software para que el robot responda adecuadamente a la posición de las mismas

- El proceso de programación y calibración de la ruta fue largo y sujeto a errores debido principalmente a la extensión del salón. Se debe plantear una manera más eficiente y automatizada para este proceso.

5 MANUAL DE OPERACIÓN

Para el uso y cuidado del prototipo se genera un manual de operación el cual le servirá al operario para dar un uso adecuado y seguro a la hora de implementar la limpieza.

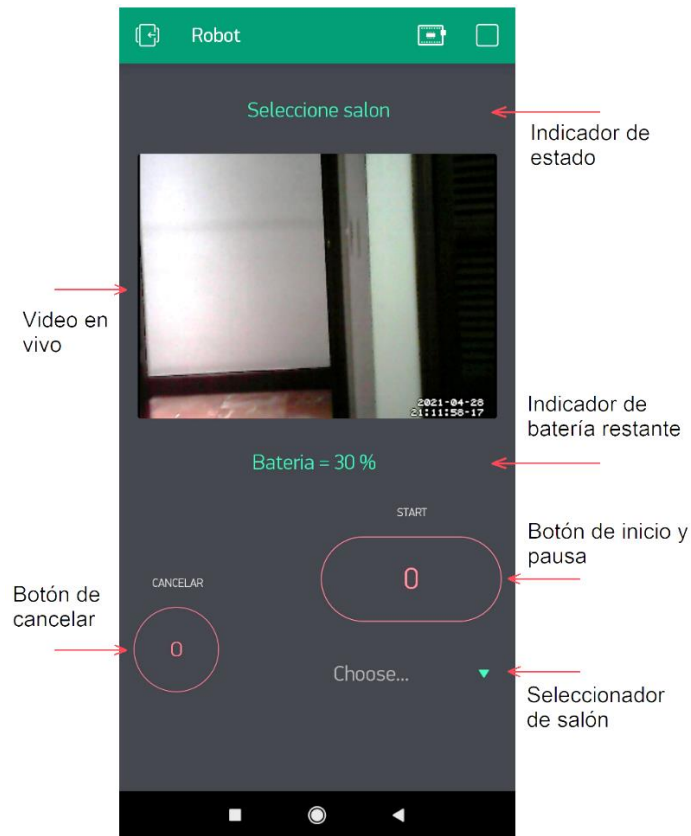
5.1 CARGA DE LA BATERÍA

- Conecte el cargador en el puerto ubicado en la parte inferior del robot. Debe poder escuchar el ventilador en funcionamiento.
- Espere a que el led ubicado encima del puerto se encienda de color verde. Aproximadamente 90 minutos.

5.2 DESCARGA Y USO DE LA APP

- Desde la tienda de aplicaciones de su celular descargue la aplicación **Blynk**.
- Inicie la app y presione el icono de QR en el inicio.
- Escanee el código QR del robot dado por los administradores.
- En la figura 45 se muestra la página principal y los elementos disponibles para el control y monitoreo del robot.

Figura 45. Captura de pantalla de la aplicación móvil.



5.3 OPERACIÓN

- Posicione el robot cerca de la esquina más cercana a la puerta del salón.
- Presione el botón de encendido en la interfaz hasta que este se ilumine de color verde.
- Espere al conteo que se muestra en el LCD de la interfaz. Ahí mismo verifique el porcentaje de carga de la batería.
- Despliegue el brazo porta bombillos.
- En este punto el procedimiento puede continuar desde la interfaz física en el robot o en la app siguiendo las instrucciones que se muestran en ambas. Los dos sistemas trabajan de forma paralela y por lo tanto se pueden ser usados indistintamente en cualquier punto de los mencionados a continuación.

Desde la interfaz:

1. Seleccione el salón en el que se encuentra utilizando los botones de navegación.
2. Espere a que se confirme que el salón fue reconocido.
3. Verifique que no se encuentre ninguna persona en el salón y que la ruta del robot se encuentra despejada.
4. Puede presionar el botón cancelar para borrar la selección de salón y regresar al punto 1.
5. Presione el botón OK para iniciar el conteo regresivo de 15 segundos marcados por el sonido de la alarma intermitente. En este tiempo, abandone el salón y cierre con llave.
6. Pasados los 15 segundos el robot comienza su rutina y se enciende la luz roja y azul de la baliza que indica que el robot está en funcionamiento. Permanezca cerca al salón para escuchar los sonidos que indican el estado del robot.
7. Si escucha 2 pitos seguidos por un silencio corto significa que el robot encontró un obstáculo en su trayectoria y se encuentra detenido. En este momento es seguro ingresar al salón. Retire el obstáculo y presione nuevamente el botón OK para reanudar la desinfección de la misma forma que en punto 4.
8. Cuando escuche un pito seguido por un silencio largo significa que el robot terminó su trabajo correctamente y puede continuar al siguiente paso.
9. Ingrese al salón y presione el botón de apagado.

Desde la app:

1. Inicie la app en su celular, si es la primera vez que hace uso de esta refiérase a la sección 5.2. Para que la app funcione correctamente, el

robot debe estar conectado a la misma red WI-FI que su celular. Puede revisar la red WI-FI del robot en la interfaz usando los botones de navegación.

2. Seleccione el salón utilizando la barra de selección en la parte inferior derecha.
3. Espere a que se confirme que el salón fue reconocido.
4. Puede presionar el botón cancelar para borrar la selección de salón y regresar al punto 2.
5. Verifique que no se encuentre ninguna persona en el salón y que la ruta del robot se encuentra despejada.
6. Salga del salón y cierre la puerta con llave.
7. Presione el botón START.
8. Si aparece el mensaje “Posible persona detectada” significa que el sensor detecto un objeto cercano similar a una persona y por seguridad requiere una confirmación extra de que el salón está vacío. De ser así, presione START nuevamente.
9. El robot comenzara su recorrido. La aplicación debe mostrar el mensaje “Desinfectando” y el botón START debe estar de color verde.
10. Mantenga la aplicación a la mano. Puede monitorear el estado del robot observando la grabación en vivo de la cámara y con los mensajes en la parte superior y pausar y reanudar la actividad del robot presionando el botón START.
11. Si recibe el mensaje “obstáculo detectado” significa que el robot encontró un obstáculo en su trayectoria y se encuentra detenido. En este momento es seguro ingresar al salón. Retire el obstáculo, salga nuevamente del salón y presione START para reanudar la desinfección.
12. Cuando reciba el mensaje “Finalizado” significa que el robot termino su trabajo correctamente y puede continuar al siguiente paso.
13. Ingrese al salón y presione el botón de apagado.

5.4 MANTENIMIENTO:

- Limpiar las ruedas motrices cada semana con un paño húmedo.
- Limpiar la superficie de la interfaz una vez cada semana.
- Lubricar las ruedas de apoyo con aceite 3 en 1 una vez cada mes.
- Si alguna llanta se llega a desajustar, utilizar una llave tipo Bristol para apretar los prisioneros.
- En caso de ser necesario, aplicar una limpieza general con un paño seco a la superficie del prototipo.
- En caso de presentarse un daño en un bombillo, reemplazar por otro con las mismas características.

6 COSTOS

Para la presentación de los costos se dividen en 5 subconjuntos según su función dentro del prototipo, los cuales son:

- Electrónica general: Se compone de todo el cableado y componentes electrónicos básicos.
- Batería: corresponde a los elementos necesarios para el armado de la batería según las especificaciones del modelo.
- Controladores: se compone de todos los dispositivos de procesamiento de datos involucrados en el prototipo.
- Sensores y actuadores: lo integran todos los elementos de hardware que se encargan de recopilar datos, los motorreductores de la base y los actuadores mecánicos y las lámparas.
- Estructura y componentes mecánicos: se compone desde las ruedas y ejes hasta toda la estructura y acabados en acrílico e impresión del prototipo.

En la tabla se referencia en la columna *fuentes* el proveedor colombiano del cual se adquirieron los respectivos componentes y de cuyas páginas se toman los precios aquí presentados. Dichos proveedores son:

[1] Didácticas Electrónicas: <https://www.didacticaselectronicas.com/>

[2] Dynamo electronics: <https://dynamoelectronics.com/>

[3] Distribuidores varios disponibles en MercadoLibre Colombia

[4] Madecentro Bucaramanga: <https://www.madecentro.com/>

[5] Tienda de iluminación local

[6] Acrílicos Metalcristal: <https://www.metalcristalacrilicos.com/>

Tabla 5. Costos de electrónica general.

Electrónica general		
Concepto	Precio	Fuente
Módulo XL4015	\$ 9 700,00	1
Módulo Buck 5V	\$ 5 000,00	1
Monster Moto Shield	\$ 37 000,00	1
Inversor	\$ 63 000,00	1
Balasto	\$ 35 000,00	5
LCD	\$ 10 000,00	1
Interruptor luminoso y botones	\$ 17 000,00	1
Relays, transistores y mosfets	\$ 6 700,00	1
Otros (resistencias, diodos, placas, cables, etc.)	\$ 35 000,00	1
TOTAL	\$ 218 400,00	

Tabla 6. Costos de la batería.

Batería		
Concepto	Precio	Fuente
Baterías 18650 X6	\$ 150 000,00	3
Tarjeta protectora BMS	\$ 18 000,00	1
Otros (caja, separadores, conectores, etc.)	\$ 10 000,00	-
TOTAL	\$ 178 000,00	

Tabla 7. Costos de los controladores

Controladores		
Concepto	Precio	Fuente
Raspberry Pi 3 B+	\$ 234 000,00	1
Arduino Mega 2560	\$ 40 000,00	1
Arduino Nano	\$ 20 000,00	1
Arduino Micro	\$ 18 000,00	1
TOTAL	\$ 312 000,00	

Tabla 8. Costos de la sensorica y actuadores.

Sensores y actuadores		
Concepto	Precio	Fuente
Bombillos ultravioletas X4	\$ 296 000,00	3
Motorreductores con Encoder	\$ 310 000,00	2
XV neato LiDAR	\$ 321 000,00	3
IMU MPU-6050	\$ 6 000,00	1
Sensores de distancia ultrasónicos HC-SR04 X4	\$ 19 000,00	1
Otros (buzzer, limite, leds)	\$ 3 000,00	1
TOTAL	\$ 955 000,00	

Tabla 9. Costos de la estructura y componentes mecánicos.

Estructura y componentes mecánicos		
Concepto	Precio	Fuente
Perfil de aluminio 2020 X7	\$ 149 100,00	1
Perfil de aluminio 2040 X3	\$ 132 000,00	1
Uniones para perfil variadas	\$ 140 000,00	1
Base de aluminio	\$ 70 000,00	-
Acople flexible X2	\$ 10 700,00	1
Ruedas Banebots X4	\$ 60 000,00	2
Ruedas de apoyo X2	\$ 23 500,00	2
Chumaceras planas X4	\$ 22 800,00	1
Ejes X2	\$ 40 000,00	-
Sistema de suspensión	\$ 80 000,00	-
Pistón resorte neumático	\$ 8 000,00	4
Cubiertas de aluminio para los bombillos	\$ 15 000,00	-
Tornillos y tuercas	\$ 100 000,00	-
Piezas en acrílico	\$ 300 000,00	6
Material usado en impresión 3d	\$ 50 000,00	-
Otros	\$ 20 000,00	-
TOTAL	\$ 1 221 100,00	

Tabla 10. Costos totales.

Costos totales	
Concepto	Precio
Estructura y componentes mecánicos	\$ 1 222 100,00
Electrónica general	\$ 218 400,00
Batería	\$ 178 000,00
Controladores	\$ 312 000,00
Sensores y actuadores	\$ 955 000,00
Adicionales	
Gastos de envío	\$ 100 000,00
TOTAL	\$ 2 985 500,00

7 CONCLUSIONES

- Se diseñó y construyó un robot autónomo de desinfección de superficies bajo las restricciones de espacio de los salones del edificio de laboratorios de la escuela de ingeniería mecánica de la Universidad Industrial de Santander, el cual logra desinfectar un aula de 10 mesones en un tiempo aproximado de 8 minutos y medio. Gracias a su batería con autonomía de 25 minutos puede desinfectar cómodamente dos salones por carga.
- Se logra realizar un robot capaz de navegar de forma autónoma y robusta por el salón guiándose por sus alrededores mientras sigue la ruta preprogramada. Con la ayuda de ROS se simplifica enormemente el trabajo de software necesario para un desarrollo de este tipo y, por lo tanto, se abren las puertas para el uso de robots autónomos en todo tipo aplicaciones similares.
- El prototipo alcanzo el objetivo propuesto al realizar efectivamente la ruta de desinfección programada, pero requiere de varias mejoras tanto en software como en hardware para llegar al nivel de precisión y confiabilidad necesarias para popularizar comercialmente su uso. A partir de este proyecto se espera generar un impulso al desarrollo de proyectos tecnológicos enfocados en combatir los grandes desafíos que enfrenta la humanidad actualmente con el COVID-19.
- Con base a la información recolectada, el uso de la luz ultravioleta para desinfección es una muy buena respuesta a la emergencia sanitaria y su uso en diversos productos ha tenido un impacto notable en el ámbito comercial a nivel internacional. Por esta razón es muy relevante continuar con el estudio de proyectos que involucren esta metodología en busca de nuevas innovaciones y su fusión con otras tecnologías de punta como en este caso la robótica.

- Gracias al uso de componentes de bajo costo disponibles en el mercado local y de un diseño enfocado en la facilidad de manufactura fue posible fabricar el prototipo, en su mayoría, desde casa y de forma manual por los autores teniendo en cuenta las restricciones de movilidad producto de la pandemia. Además, se logró mantener un presupuesto bajo que no sobrepasó los tres millones de pesos colombianos.
- Este proyecto se alinea muy de cerca con la misión de la escuela de ingeniería mecánica de la universidad industrial de Santander al buscar una solución tecnológica y competitiva con todo el compromiso social en respuesta a la inesperada situación global que se vive actualmente. Su objetivo esencial es brindar una alternativa de limpieza y bioseguridad libre de químicos para reducir el riesgo de contagio de toda la comunidad universitaria y a su vez aliviar el esfuerzo de los operarios de limpieza que serán fundamentales en el retorno a la presencialidad.

8 RECOMENDACIONES

- Cubrir la demanda de desinfección de salones de una institución de gran tamaño no es una tarea sencilla. Con el prototipo diseñado se desinfecta un salón en aproximadamente 8 minutos y medio y se pueden cubrir dos salones por ciclo de batería seguido por un periodo de 90 minutos de carga. Por lo tanto, es altamente recomendable diseñar el prototipo con batería removible de tal forma que se disponga de varias baterías previamente cargadas para reemplazarlas y cubrir más salones consecutivamente.
- Una de las principales debilidades encontradas en este prototipo fue la base, las ruedas motrices, las ruedas de apoyo y la estructura de la misma no se desempeñaron adecuadamente en el suelo liso y con suciedad de los salones de prueba afectando considerablemente la confiabilidad del robot. Para futuros prototipos es recomendable un rediseño general de esta sección del robot haciendo uso de ruedas de mayor diámetro y espesor para diferentes condiciones de trabajo.
- Debido a la altura del robot, la detección de obstáculos fue también una importante fuente de error. Se recomienda incluir más sensores de distancia a diferentes alturas para tener una mejor detección tridimensional de los alrededores, así como el uso de sensores de distancia de mayor precisión y mejor respuesta al ruido. También se plantea el uso de cámaras para la detección de obstáculos manteniendo el sistema de navegación actual.
- El proceso de programación de la ruta resulto largo y tedioso, por lo tanto, se recomienda añadir una capa adicional de software para automatizar lo más posible buscando también que pueda ser realizado y probado por los usuarios a través de una interfaz más cómoda.

BIBLIOGRAFIA

BALCELLS, Joseph y ROMERAL, José L. 1998. Autómatas Programables. Mexico: Alfaomega. ISBN: 9701502477 9789701502471

TOCKN. MPU6050_tockn. 2019. [Consultado el 27 de agosto de 2021] disponible en: https://github.com/Tockn/MPU6050_tockn

BEVERLY, Robert. Covid-19 Reveals importance of UV-C in HVAC industry. Junio 2020. En revista: The ACHR News. [Consultado el 27 de agosto de 2021] Disponible en: <https://www.achrnews.com/articles/143318-covid-19-reveals-importance-of-uv-c-in-hvac-industry>

BLYNK. (s.f.). Blynk documentation. [Consultado el 27 de agosto de 2021] Disponible en: <http://docs.blynk.cc/>

CLARK, Chris. (2016). ARW – Lecture 01 Odometry Kinematics. 2016 [Consultado el 27 de agosto de 2021] Disponible en: <https://www.hmc.edu/lair/ARW/ARW-Lecture01-Odometry.pdf>

DELLAERT, Frank., *et al.* Monte Carlo Localization for Mobile Robots. 1999 [Consultado el 27 de agosto de 2021] Disponible en: https://www.ri.cmu.edu/pub_files/pub1/dellaert_frank_1999_2/dellaert_frank_1999_2.pdf

ECKEL, Tim. NewPing Library for Arduino. Julio 2019. [Consultado el 27 de agosto de 2021] Disponible en: <https://playground.arduino.cc/Code/NewPing/>

ESP WATER PRODUCTS. Understanding Water UV Purification. 2021. [Consultado el 27 de agosto de 2021] Disponible en: <https://www.espwaterproducts.com/understanding-uv/>

GERKEY, Brian P. amcl ros package. Octubre 2020. Ros wiki. [Consultado el 27 de agosto de 2021] Disponible en: <http://wiki.ros.org/amcl>

GRISSETTI, Giorgio; STACHNISS, Cyrill y FRESE, Udo. OpenSlam. [Consultado el 27 de agosto de 2021] Disponible en: <https://openslam-org.github.io/gmapping.html>

HANDSCHIN, J. E. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. 1970. Automatica. Volumen 6. pp. 555–563. [Consultado el 27 de agosto de 2021] DOI: [https://doi.org/10.1016/0005-1098\(70\)90010-5](https://doi.org/10.1016/0005-1098(70)90010-5).

IES PHOTOBIOLOGY COMMITTEE. IES Committee Report: Germicidal Ultraviolet (GUV) – Frequently Asked Questions. 2020 [Consultado el 27 de agosto de 2021] disponible en: <https://media.ies.org/docs/standards/IES-CR-2-20-V1-6d.pdf>

LEROY, James. XV Lidar Controller. 2021. GitHub [Consultado el 27 de agosto de 2021] Disponible en: https://github.com/getSurreal/XV_Lidar_Controller

MEEUSEN, Wim. robot_pose_ekf. Marzo 2012 [Consultado el 27 de agosto de 2021] Disponible en: http://wiki.ros.org/robot_pose_ekf

MICROCHEM LABORATORY. (s.f.). UV room disinfection devices. [Consultado el 27 de agosto de 2021] Disponible en: <https://microchemlab.com/test/uv-room-disinfection-devices>

PERKO, Eric., *et al.* xv_11_laser_driver. 2017. GitHub [Consultado el 27 de agosto de 2021] Disponible en: https://github.com/rohbotics/xv_11_laser_driver

PHILIPS LIGHTING. TUV T8 Ficha técnica de la familia. 2020 [Consultado el 27 de agosto de 2021] Disponible en: <https://www.lighting.philips.com.mx/prof/lamparas-y-tubos-convencionales/lamparas-especiales/purificationwater-and-air/commercial-and-professional-air/tuv-t8>

PRZYBYLA, Mateusz. Online Detection and Tracking of 2D Geometric Obstacles from LRF Data. Febrero 2017. GitHub [Consultado el 27 de agosto de 2021] Disponible en:

https://github.com/tysik/obstacle_detector/blob/master/resources/ObstacleDetector.pdf

RAEISZADEH, Milad. y ADELI, Babak. A Critical Review on Ultraviolet Disinfection Systems against COVID-19 Outbreak: Applicability, Validation, and Safety Considerations. 2020. En revista: *ACS Photonics*, acsphotronics.0c01245. [Consultado el 27 de agosto de 2021] DOI: <https://doi.org/10.1021/acsphotronics.0c01245>

SPARKFUN. (s.f.). SparkFun Monster Moto Shield. [Consultado el 27 de agosto de 2021] Disponible en: <https://www.sparkfun.com/products/retired/10182>

ZHENG, Kaiyu. ROS Navigation Tuning Guide. Septiembre 2016 [Consultado el 27 de agosto de 2021] Disponible en: <https://kaiyuzheng.me/documents/navguide.pdf>

ANEXOS

ANEXO A. CÁLCULO DE LOS PARÁMETROS TÉCNICOS PARA LA INACTIVACIÓN DEL SARS-COV 19

Cálculo de los parámetros técnicos para la inactivación del SARS-CoV 19

Juan Diego Villamizar y Javier Rangel

Octubre de 2020

1. Desinfección Ultravioleta

La radiación ultravioleta de onda corta UVc ha sido ampliamente documentada y aplicada como un efectivo método de desinfección para toda clase de microorganismos [1]. Los dispositivos UVc más usados y comercialmente disponibles para aplicaciones germicidas son las lámparas de vapor de mercurio que emiten radiación ultravioleta con longitud de onda de 254nm. Sin embargo, esta radiación es tan bien conocida por causar daño agudo y crónico en los ojos y en la piel humana [2][3].

La cantidad de energía ultravioleta necesaria para la desinfección debe hallarse de forma experimental y depende del tipo de microorganismo y del porcentaje de inactivación que se quiere alcanzar, generalmente se expresa como dosis D_{90} e indica a la cantidad de energía en J/m^2 que se debe suministrar para lograr una reducción logarítmica del microorganismo presente en el área, es decir, para la inactivación del 90% del microorganismo presente, y de esta forma al duplicar la dosis se obtendría una inactivación del 99%.

2. Revisión bibliográfica para la dosis del SARS-CoV-2

Hasta la fecha, no se encuentran publicaciones que contengan datos experimentales específicos para el SARS-CoV-2, sin embargo, existen numerosos

artículos con datos para otros coronavirus [5][6][7][8][9], y otros virus como el de la influenza H1N1 [10] o el ébola [11]. Así mismo, se encuentran varias publicaciones recientes que realizan diferentes análisis para estimar el valor de la dosis para el SARS-CoV-2 con base a los múltiples datos disponibles [4][5][12].

En estos artículos se presentan los siguientes valores para D_{90} , en [5] $17 J/m^2$ en base a experimentos realizados por los autores en otros coronavirus de transmisión humana, en [12] concluyen un valor que consideran muy conservador de $30 J/m^2$ a partir de una revisión bibliográfica y experimentación con bacterias *Staphylococcus aureus*. Y por último en [4] se recomiendan $37 J/m^2$ basada únicamente en una revisión bibliográfica más amplia y reciente.

Con base a toda la bibliografía revisada se observa que determinar el D_{90} no es una tarea sencilla y es altamente dependiente de la metodología de medición y aplicación de la luz, por lo tanto, se decide tomar se decide tomar el máximo valor estimado en los documentos encontrados, es decir:

$$37 J/m^2$$

3. Irradiación a una superficie desde una fuente UV

Para calcular potencia por unidad de área de la radiación UV que recibe un punto en una superficie cercana a una lampara se deben considerar la ley el cuadrado inverso y la ley de Lambert [14], que indican los efectos de variar la distancia y el ángulo de incidencia de la fuente de luz hasta el punto en la superficie como se muestra en las figuras 1 y 2.

Figura 1. Ley del cuadrado inverso $E = I/r^2$

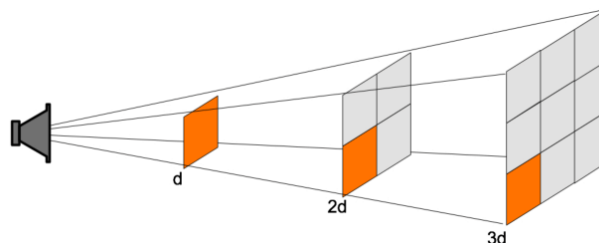
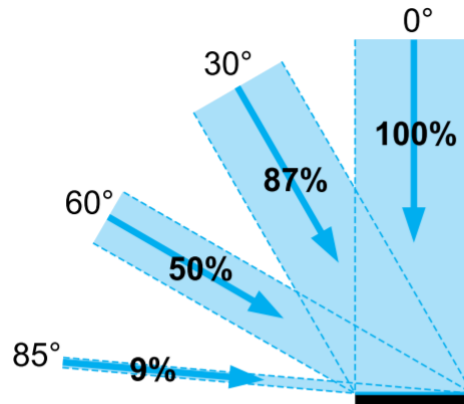
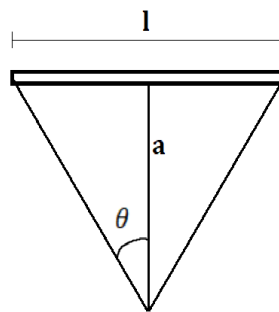


Figura 2. Ley de Lambert $E_0 = E * \cos(\theta)$



Sin embargo, para aplicar estas leyes en fuentes de luz tubulares y alargadas, como lo son los bombillos germicidas, es necesario realizar procesos de integración que permitan abordar el problema de forma bidimensional, con dicho análisis se construye la fórmula de Keitz [13]

Figura 3. Esquema para el cálculo de la irradiación UV a través de la fórmula de Keitz.



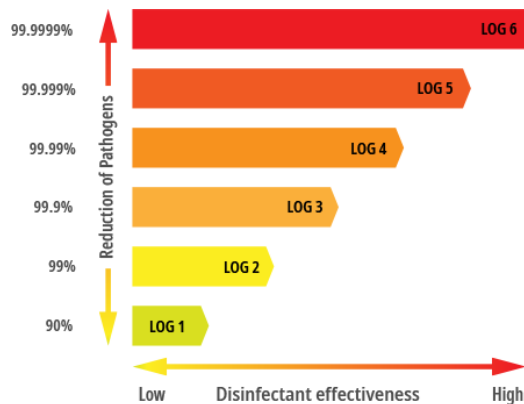
$$E_0 = \frac{\Phi}{2 * \pi * a} * (2 * \theta + \sin 2\theta) \quad (1)$$

Donde:

- φ : Valor del flujo de radiación UV-c en Watts que produce el bombillo.
- l : Longitud del bombillo.
- a : Distancia del centro del bombillo al punto de irradiación.
- θ : ángulo entre el extremo del bombillo y el punto P

Con lo cual se obtiene un valor de irradiación en un punto P en W/m^2 que al multiplicarlo por el tiempo de exposición se obtiene la energía total que debe ser comparada con el valor D_{90} para el SARS-CoV-2. Finalmente se puede expresar dicho resultado con el cociente $\frac{E}{D_{90}}$ que indica la cantidad de reducciones logarítmicas y por lo tanto el porcentaje de desinfección que se alcanza en la superficie.

Figura 4. Reducciones logarítmicas

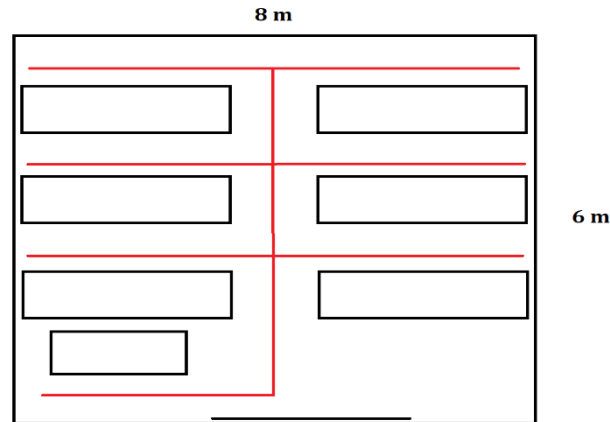


4. Metodología

4.1 Recorrido del robot

Primero se establece el recorrido que debe realizar el robot para la limpieza del salón de dimensiones 8 x 6 metros como indica la figura 5.

Figura 5. Recorrido del robot en el salón



Nótese que recorre 2 veces cada trayecto, de ida y de regreso, por lo tanto, la distancia total recorrida es de:

$$(8 * 3 + 4 + 6) * 2 = 68 [m]$$

Si se desea que la limpieza tarde 10 minutos la velocidad del robot debe ser

$$v = \frac{68}{10} \left[\frac{m}{min} \right] = 11.33 \left[\frac{cm}{s} \right]$$

4.2 Consideraciones para el desplazamiento del robot

Con la velocidad obtenida, se puede calcular el tiempo de exposición para cada punto en el recorrido teniendo en cuenta el ángulo mínimo de incidencia de la luz que en este caso se asume como 20 grados como se observa en la figura 6 y a partir de este, se elabora el diagrama de la figura 7

Figura 6. Angulo mínimo de incidencia de la luz

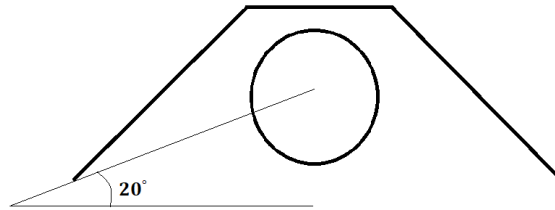
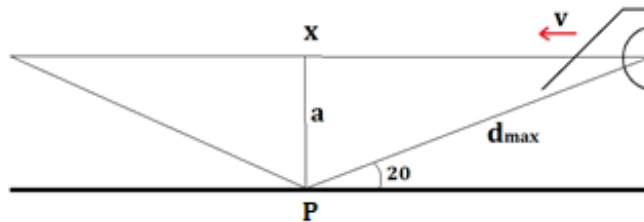


Figura 7. Diagrama para el cálculo del tiempo de exposición



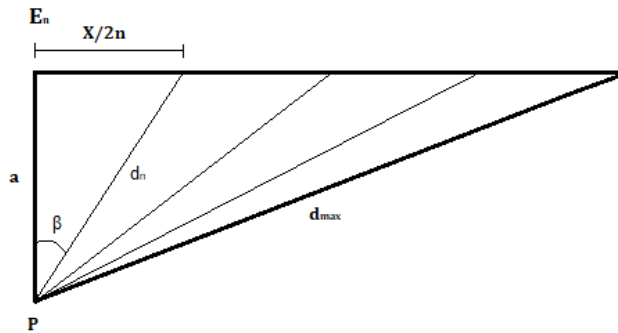
Y teniendo en cuenta que el robot recorre la distancia X dos veces (de ida y de regreso) se puede calcular el tiempo total de exposición para el punto P como:

$$t = 2 * \frac{X}{v}$$
$$t = 2 * \frac{2a}{v * \tan(20)}$$

Sin embargo, es evidente que el punto P no recibe la misma energía en todo el recorrido debido al cambio de la distancia normal y al ángulo, y por lo tanto se tiene ahora un problema que depende de las tres dimensiones geométricas y del tiempo.

Para abordar dicho problema se propone la siguiente metodología de aproximación con la formula bidimensional de keitz ilustrada en la figura 8.

Figura 8. Diagrama para la aproximación



Primero, teniendo en cuenta la simetría del trayecto X respecto al punto P se toma medio triángulo de la figura 6, después se divide el trayecto X/2 en N segmentos equidistantes, para cada segmento se calcula la energía E_n aplicando la fórmula de keitz y multiplicándola por el tiempo que tarda el robot en cruzar dicho segmento, además considerando los efectos de la variación del ángulo y la ley de Lambert en esta nueva dimensión se agrega el factor $\cos(\beta)$ a la ecuación despreciando los efectos de la tridimensionalidad en esta ley.

De esta forma la ecuación de keitz se reescribe de la siguiente manera:

$$E_{(n)} = \frac{\varphi}{2 * \pi * I * d_{(n)}} * (2 * \theta + \sin 2\theta) * \cos(\beta_n)$$

Finalmente se realiza la sumatoria para los n segmentos, se multiplica por 4 que corresponde a las veces que el robot recorre la distancia X/2 y se obtiene la energía en J/m^2 que recibe cada punto en el recorrido.

4.3 Propiedades y distribución de los bombillos a utilizar

Para este proyecto se utilizan bombillos philips de 30 y 15 watts, los cuales cuentan con las siguientes características según el catálogo del fabricante [15]:

	15 watts	30 watts
L	44 [cm]	90 [cm]
φ	4.9 [W]	12 [W]

Para la desinfección de la superficie de las mesas se usarán 2 bombillos de 15 watts en paralelo con la superficie, para la parte inferior 1 bombillo de 15 watts ubicado a 90 grados y para las sillas 1 bombillo de 30 watts que se asume paralelo a estas.

5. Resultados

Para resolver el sistema se hace uso del software EES, se debe resolver independientemente para los datos de la superficie de la mesa, la parte inferior de las mesas y las sillas

5.1 Superficie de las mesas

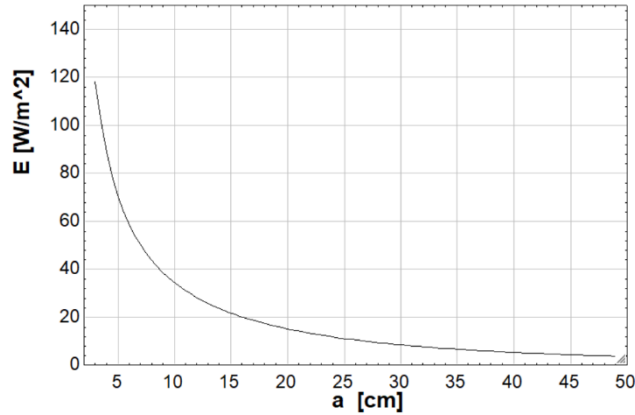
Datos:

- $L = 44$ [cm]

- $\varphi = 2 * 4.9 = 9.8$ [W]

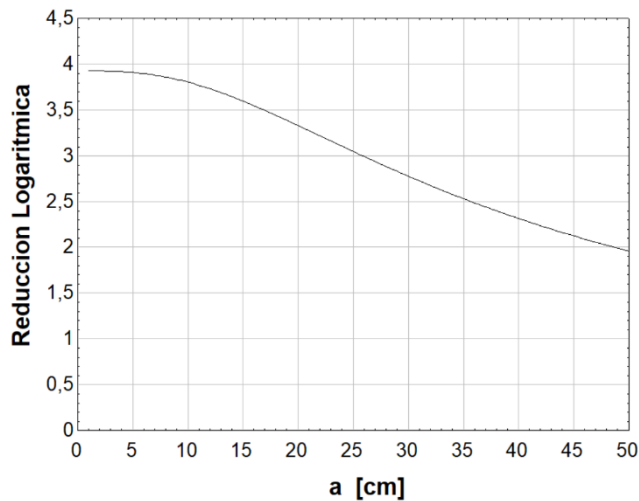
Primero se construye como referencia la gráfica de la energía aplicando la ecuación de keitz sin considerar el movimiento del robot y se obtiene la esperada curva de decaimiento como se observa en la figura 9

Figura 9. Potencia en W/m^2 de en función de la distancia “a” estacionaria



A continuación, se implementan todas las consideraciones para el desplazamiento mencionadas anteriormente, se realiza la aproximación con un $N=20$ y se construye el gráfico que muestra el número de reducciones logarítmicas (R) alcanzadas en función de la distancia.

Figura 10. Numero de reducciones logarítmicas en función de la distancia para la superficie de las mesas



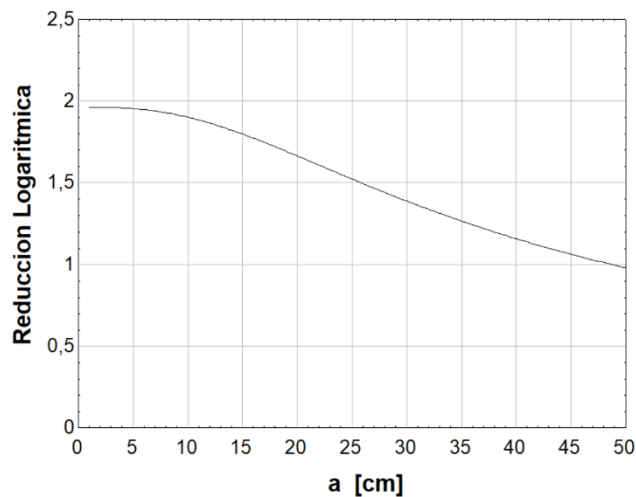
En la gráfica se observa que la reducción logarítmica llega a un máximo sin importar que tanto se acerque la fuente, esto se explica debido al movimiento del robot, entre más cerca los bombillos de la superficie más radiación absorberá, pero por menos tiempo. A partir de la gráfica se concluye que para distancias menores a 10 cm se

alcanza un valor muy cercano a 4 reducciones logarítmicas, es decir un porcentaje de inactivación del **99.99%**

5.2 Bajo las mesas

El sistema de desinfección para la parte inferior de las mesas cuenta con un solo bombillo de 15 watts por lo tanto serán los mismos datos anteriores con un φ reducido a la mitad, repitiendo el mismo procedimiento anterior se obtiene la curva de reducción logarítmica como se muestra en la figura 11.

Figura 11. Numero de reducciones logarítmicas en función de la distancia para la parte inferior de las mesas

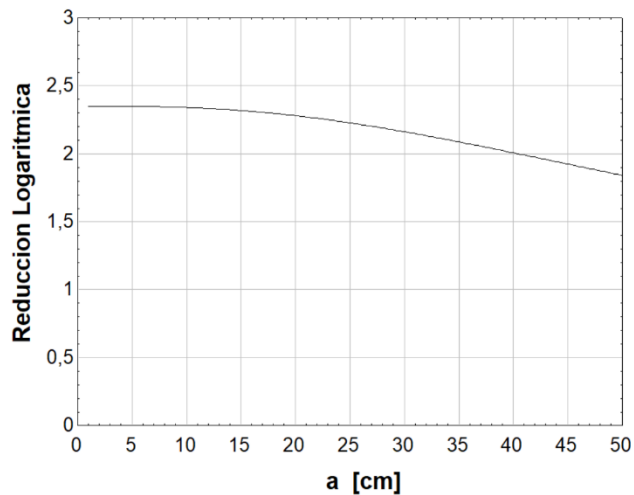


Se obtiene una curva muy similar a la anterior, pero como era de esperarse, reducida a la mitad. Además, debe mencionarse que debido al ángulo de aplicación de la luz para el revés de la mesa se produce una reducción en la energía producto de la ley de Lambert difícil de estimar matemáticamente, por eso para concluir de forma segura, se puede afirmar que se logra una inactivación mayor al **90 %**.

5.3 Sillas

Para las sillas y otras superficies se hace uso de un bombillo de 30 watts con el cual se obtiene la siguiente gráfica:

Figura 12. Numero de reducciones logarítmicas en función de la distancia para las sillas



Con este bombillo se obtiene una gráfica menos sensible al cambio en la distancia, la reducción logarítmica es muy cercana a 2 en los primeros 50 cm por lo tanto se llega a una inactivación del **99%**.

5.4 Variación de otros parámetros

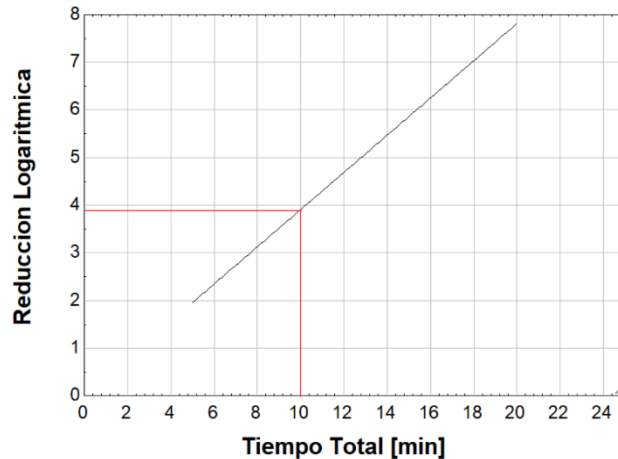
Finalmente se desea conocer la influencia sobre la reducción logarítmica que tienen otras variables clave que fueron establecidas anteriormente, el tiempo máximo de limpieza y el ángulo mínimo de incidencia de la luz

5.4.1 Tiempo total del recorrido

Hasta ahora se ha trabajado con un tiempo total del recorrido de 10 minutos, lo cual determina directamente la velocidad y por lo tanto el tiempo de exposición a la radiación de cada punto, a continuación, se elabora una gráfica que ilustra el

comportamiento de la desinfección en función al tiempo total del recorrido para la superficie de las mesas, es decir para el modelo ilustrado en la figura 10 en su punto máximo.

Figura 13. Numero de reducciones logarítmicas en función del tiempo de limpieza

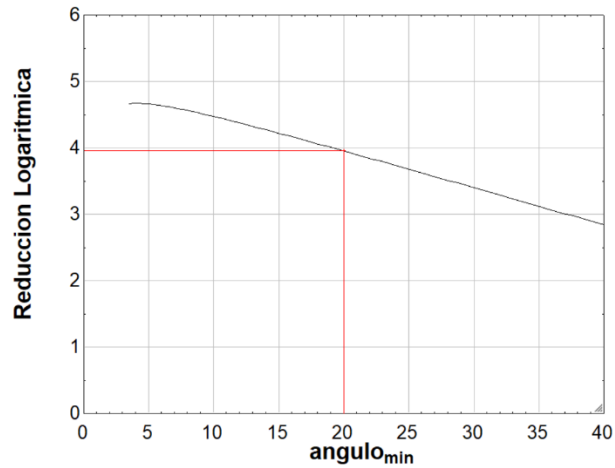


En rojo se señala el punto que fue usado para los análisis anteriores, se observa que R depende fuertemente y de forma lineal del tiempo total y por lo tanto la velocidad de desplazamiento del robot es un parámetro clave para una adecuada desinfección.

5.4.2 Angulo minimo de incidencia

Como se expuso en la sección correspondiente, el angulo minimo de incidencia de la luz determina el tiempo de exposicion para cada punto en el recorrido, pero a angulos pequeños la energia que llega a la superficie disminuye drasticamente, por esto se elabora la siguiente para ilustrar dicho comportamineto, nuevamente en el punto descrito en el numeral anterior en el que hasta ahora se habia usado un angulo de 20°.

Figura 14. Numero de reducciones logarítmicas en función del ángulo mínimo de incidencia.



Como se esperaba, entre menor sea el ángulo mejor será la desinfección de tal forma que se pueden obtener valores de R de hasta 20% mejores con ángulos menores. Sin embargo, para ángulos bajos el modelo matemático implementado puede presentar algunas inconsistencias cuando se varían otros parámetros, esto debido al método de aproximación descrito.

Físicamente este ángulo depende del montaje que se le haga al bombillo y ángulos muy pequeños pueden presentar desventajas mecánicas, debe ser optimizado con base al análisis aquí realizado.

BIBLIOGRAFÍA

- [1] KOWALSKI, Wladyslaw. Ultraviolet Germicidal Irradiation Handbook: UVGI for Air and Surface Disinfection. 1 ed. New York: Springer. 2009. [Consultado el 27 de agosto de 2021] DOI: <https://doi.org/10.1007/978-3-642-01999-9>
- [2] TREVISAN, Andrea., *et al.* Unusual High Exposure to Ultraviolet-C Radiation. En revista: Photochemistry and photobiology. 2006. [Consultado el 27 de agosto de 2021] DOI: 10.1562/2005-10-27-RA-728

- [3] ZAFFINA, S., *et al.* Accidental exposure to UV radiation produced by germicidal lamp: case report and risk assessment. 2012. En revista: *Photochemistry and photobiology*, 88(4), 1001–1004. [Consultado el 27 de agosto de 2021] DOI: <https://doi.org/10.1111/j.1751-1097.2012.01151.x>
- [4] HESSLING, Martin., *et al.* Ultraviolet irradiation doses for coronavirus inactivation -review and analysis of coronavirus photoinactivation studies. (2020). En revista: *GMS Hygiene and Infection Control*. 15. [Consultado el 27 de agosto de 2021] DOI: 10.3205/dgkh000343.
- [5] BUONANNO, Manuela., *et al.* Far-UVC light efficiently and safely inactivates airborne human coronaviruses. 2020. Preprint, Version 1. Research Square. Apr 27. [Consultado el 27 de agosto de 2021] DOI: 10.21203/rs.3.rs-25728/v1
- [6] BEDELL, Kurt; BUCHAKLIAN, Adam H. y PERLMAN, Stanley. Efficacy of an Automated Multiple Emitter Whole-Room Ultraviolet-C Disinfection System Against Coronaviruses MHV and MERS-CoV. 2016. *Infect Control Hosp Epidemiol*. [Consultado el 27 de agosto de 2021] DOI: 10.1017/ice.2015.348
- [7] KOWALSKI, Wladyslaw. Performance of the UV24 Unit Against Zoonotic Pathogens. San Fernando, CA: Medical Illumination. 2017. [Consultado el 27 de agosto de 2021] Disponible en: <http://www.medillum.com/wp-content/uploads/2017/03/Performance-of-the-UV24-Unit-AgainstZoonotic-Pathogens.pdf>
- [8] DESHMUKH, DR y POMEROY, BS. Ultraviolet inactivation and photoreactivation of avian viruses. *Avian Dis*. 1969.
- [9] NARITA, Kouji., *et al.* 222-nm UVC inactivates a wide spectrum of microbial pathogens. 2020. *Journal of Hospital Infection*. [Consultado el 27 de agosto de 2021] DOI: 10.1016/j.jhin.2020.03.030.
- [10] WELCH, D., *et al.* Far-UVC light: A new tool to control the spread of airborne-mediated microbial diseases. *Sci Rep* **8**, 2752 (2018). [Consultado el 27 de agosto de 2021] DOI <https://doi.org/10.1038/s41598-018-21058-w>

[11] LEDE, Ivar; NOLTE, Karina y KROES René. (2020). A Scalable Method for Ultraviolet C Disinfection of Surgical Facemasks Type IIR and Filtering Facepiece Particle Respirators 1 and 2. [Consultado el 27 de agosto de 2021] disponible en: <https://europepmc.org/article/ppr/ppr153995>

[12] KEITZ, H.A.E. Light Calculations and Measurements. 1971. 2nd Ed., Eindhoven, Netherlands: N.V. Philips.

[13] RYER, Alexander. The light Measurement Handbook 1997. [Consultado el 27 de agosto de 2021] Disponible en: https://cgvr.informatik.uni-bremen.de/teaching/cg_literatur/ILT-Light-Measurement-Handbook.pdf

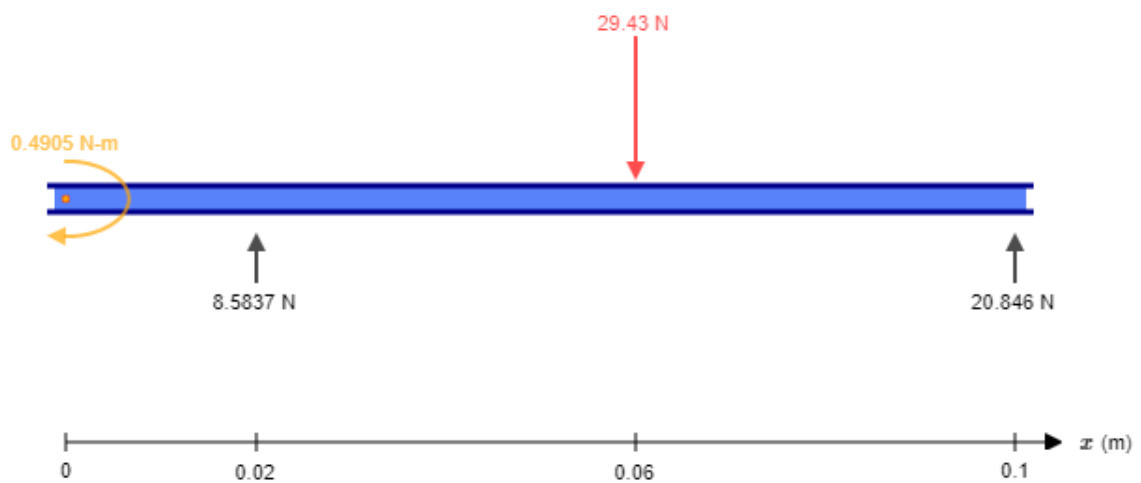
[14] PHILIPS LIGHTING. TUV T8 Ficha técnica de la familia. 2020. disponible en: <https://www.lighting.philips.com.mx/prof/lamparas-y-tubos-convencionales/lamparas-especiales/purificationwater-and-air/commercial-and-professional-air/tuv-t8>

ANEXO B. CÁLCULOS DEL EJE DE LA BASE

Para el diseño del eje que va anclado a los motores y soporta las ruedas de tracción se realiza un análisis de esfuerzos para obtener un valor de diámetro que garantice la correcta funcionalidad del sistema.

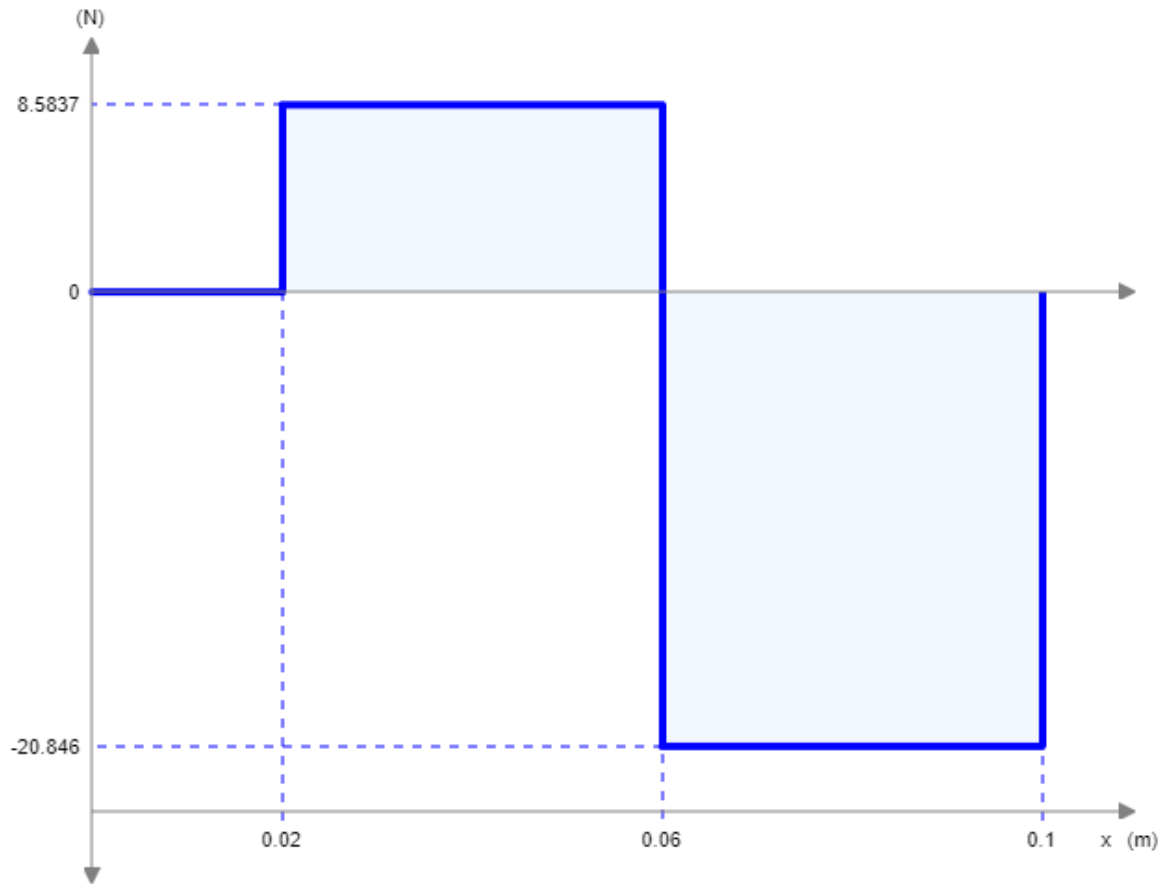
El motor utilizado ejerce un torque de 5kg-cm en la esquina del eje, posee dos apoyos en rodamientos situados a dos y diez centímetros del origen respectivamente, y una carga de 3Kg aplicada a los 6 centímetros del origen la cual equivale a la cuarta parte del peso del prototipo como se muestra en la figura.

Figura 1. Diagrama de cargas aplicadas en el eje.



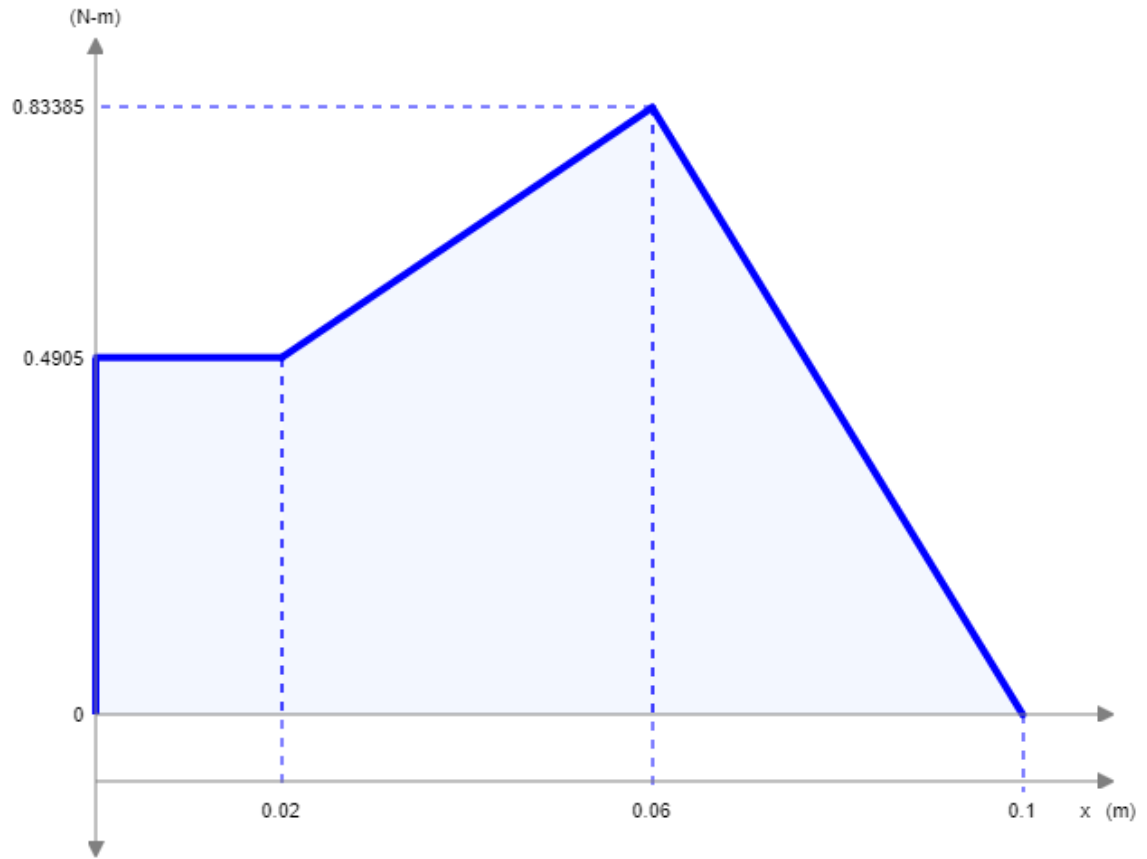
Para el análisis de las fuerzas cortantes se tiene un diagrama que muestra que la mayor carga cortante es de 20.84N aplicada en la zona desde 6 cm hasta el final del eje como se muestra en la figura.

Figura 2. Diagrama de esfuerzos cortantes en el eje.



Finalmente, en el análisis de los momentos flectores se tiene que el mayor valor se encuentra a los 6cm del origen del eje con un valor de 0.83 N-m como se muestra en la figura.

Figura 3. Diagrama de momentos flectores en el eje.



Para analizar los esfuerzos por flexión se dispone de la ecuación:

$$\sigma_m = \frac{M * c}{I}$$

Donde c corresponde al radio del eje, I a su momento de inercia y M el momento flector hallado en el diagrama.

El radio del eje es previamente seleccionado como 8mm debido a la disponibilidad de ese adaptador en el mercado, por lo que el cálculo se basará en comprobar si cumple las condiciones geométricas para garantizar el funcionamiento adecuado.

Reemplazando en la ecuación se tiene que:

$$\sigma_m = \frac{0.833N - m * 0.008m}{\frac{\pi * (0.008m)^4}{4}}$$

$$\sigma_m = 2.071 MPa$$

Ahora para hallar los esfuerzos cortantes se tiene la siguiente ecuación:

$$\tau = \frac{F}{A} + \frac{T * c}{I}$$

Donde F viene a ser la fuerza cortante de mayor magnitud sobre el eje, T el esfuerzo cortante que produce dicha fuerza y A el área transversal del eje. Reemplazando en la ecuación, se obtiene:

$$\tau = \frac{20.846N}{\pi * (0.008m)^2} + \frac{0.833 N - m * 0.008m}{\frac{\pi * (0.008m)^4}{4}}$$

$$\tau = 20.81 MPa$$

Para analizar si el diámetro elegido cumple los criterios de diseño se procede a ver los valores máximos de esfuerzo de tensión y cortantes del acero inoxidable AISI 304 con un factor de seguridad de 2

$$\sigma_{max} = \frac{585MPa}{2}$$

$$\sigma_{max} = 292.5 MPa$$

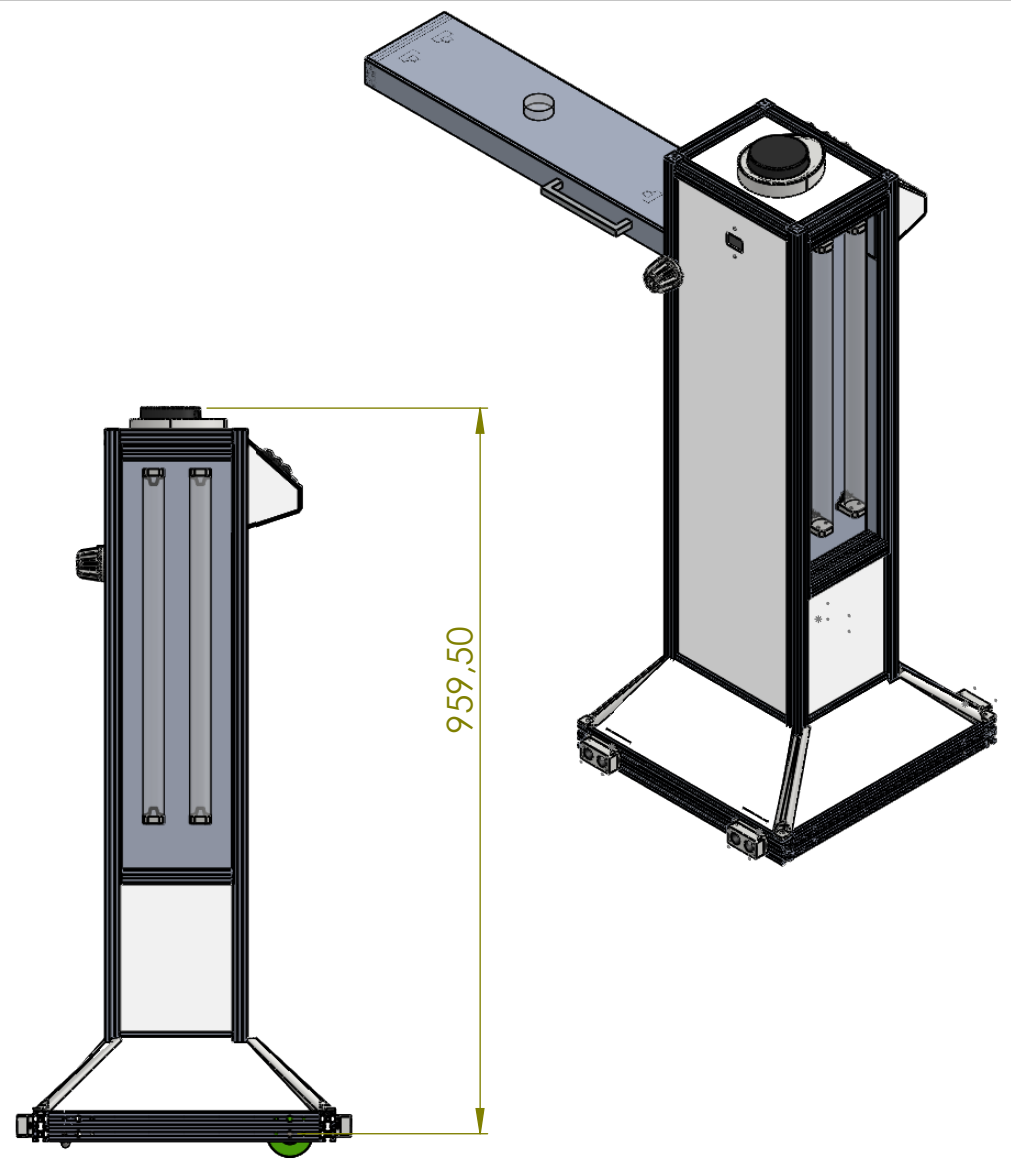
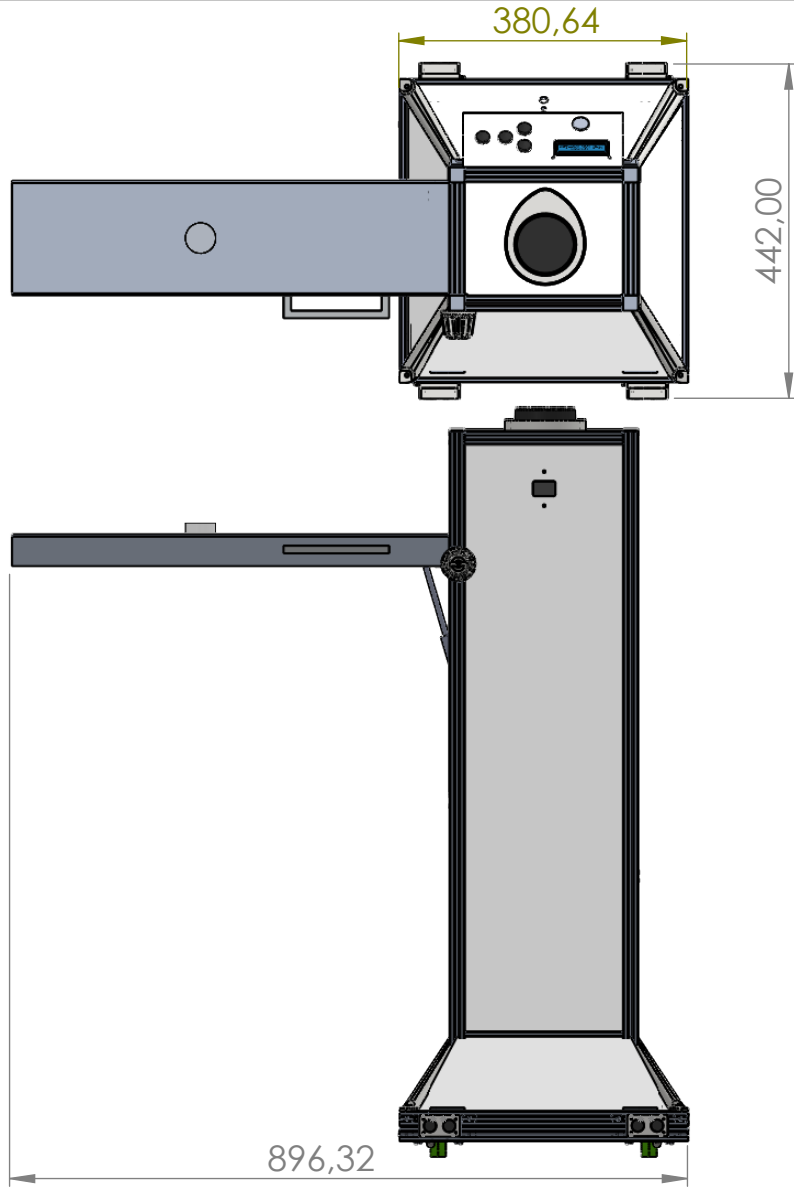
$$\tau_{max} = \frac{235MPa}{2}$$

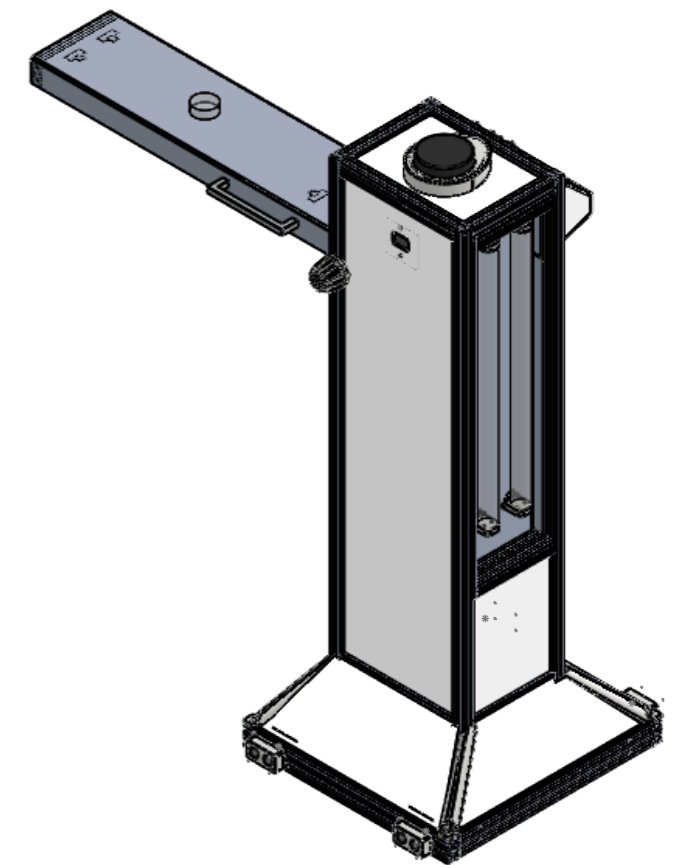
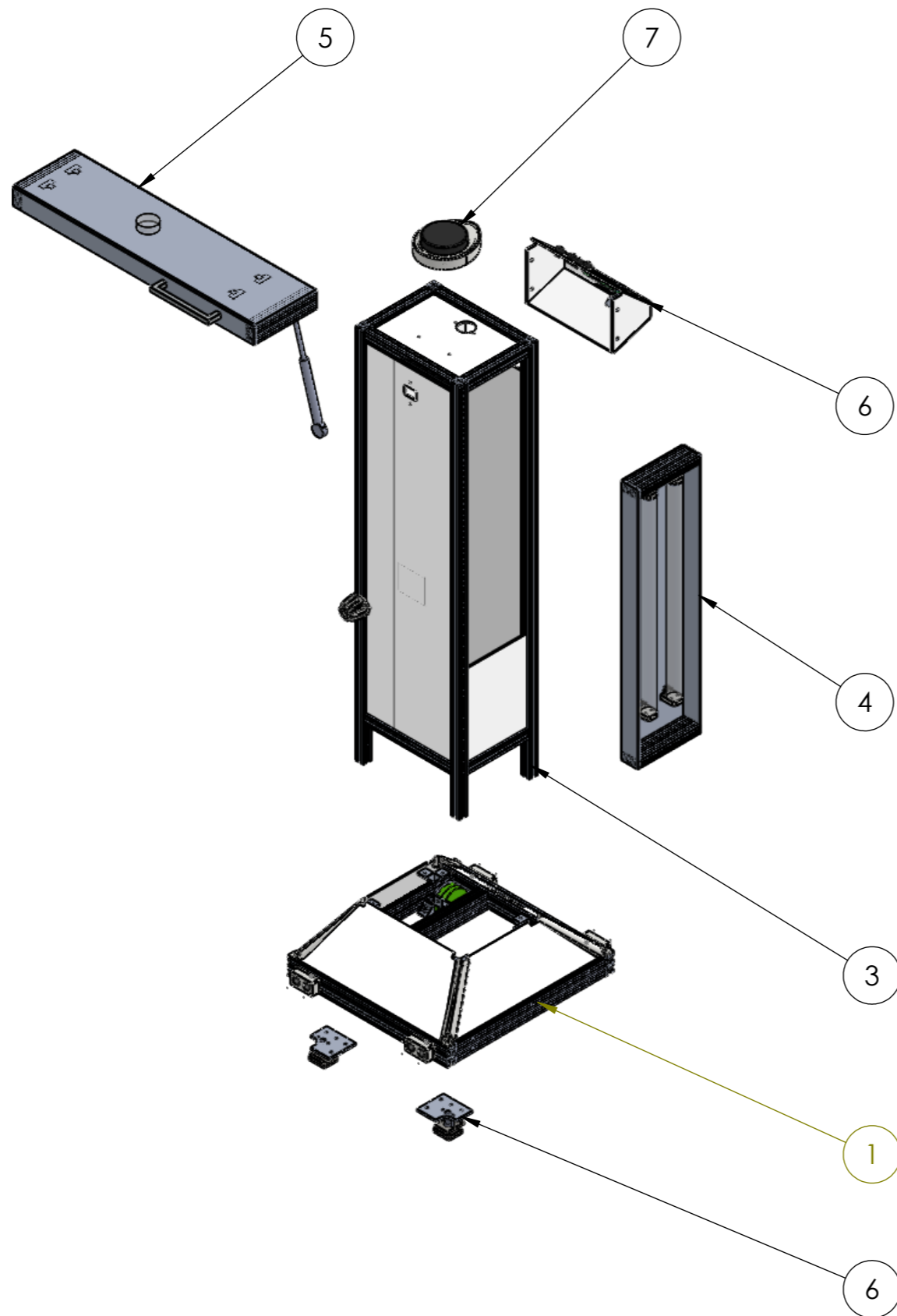
$$\tau_{max} = 117.5 MPa$$

Con los valores obtenidos de esfuerzos máximos permisibles y los valores de esfuerzos calculados que soporta el eje se procede a compararlos, por lo que se observa que los valores calculados se encuentran muy por debajo de los límites admisibles, y teniendo en cuenta que por comodidad y disponibilidad de adaptadores y rodamientos para el eje de 8mm se decide trabajar con dicho diámetro en el material Acero inoxidable AISI 304.

ANEXO C. PLANOS GENERALES DEL ROBOT

Se adjuntan los planos generales del prototipo realizado en CAD mediante el software SolidWorks, los cuales incluyen una vista general y explosionada de todo el robot y todos los subsistemas que lo componen.





7	LIDAR XV-11	1	VARIOS
6	INTERFAZ	1	VARIOS
5	LÁMPARA HORIZONTAL	1	VARIOS
4	LÁMPARA VERTICAL	1	VARIOS
3	CUERPO PRINCIPAL	1	VARIOS
2	SISTEMA DE AMORTIGUADORES	1	VARIOS
1	BASE	1	VARIOS
Nº	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:10

ROBOT DE DESINFECCION DE SUPERFICIES



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

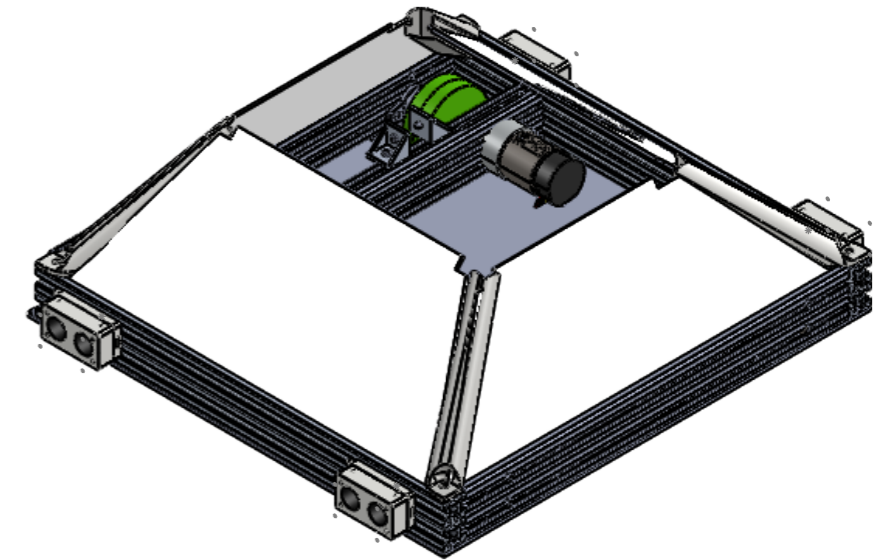
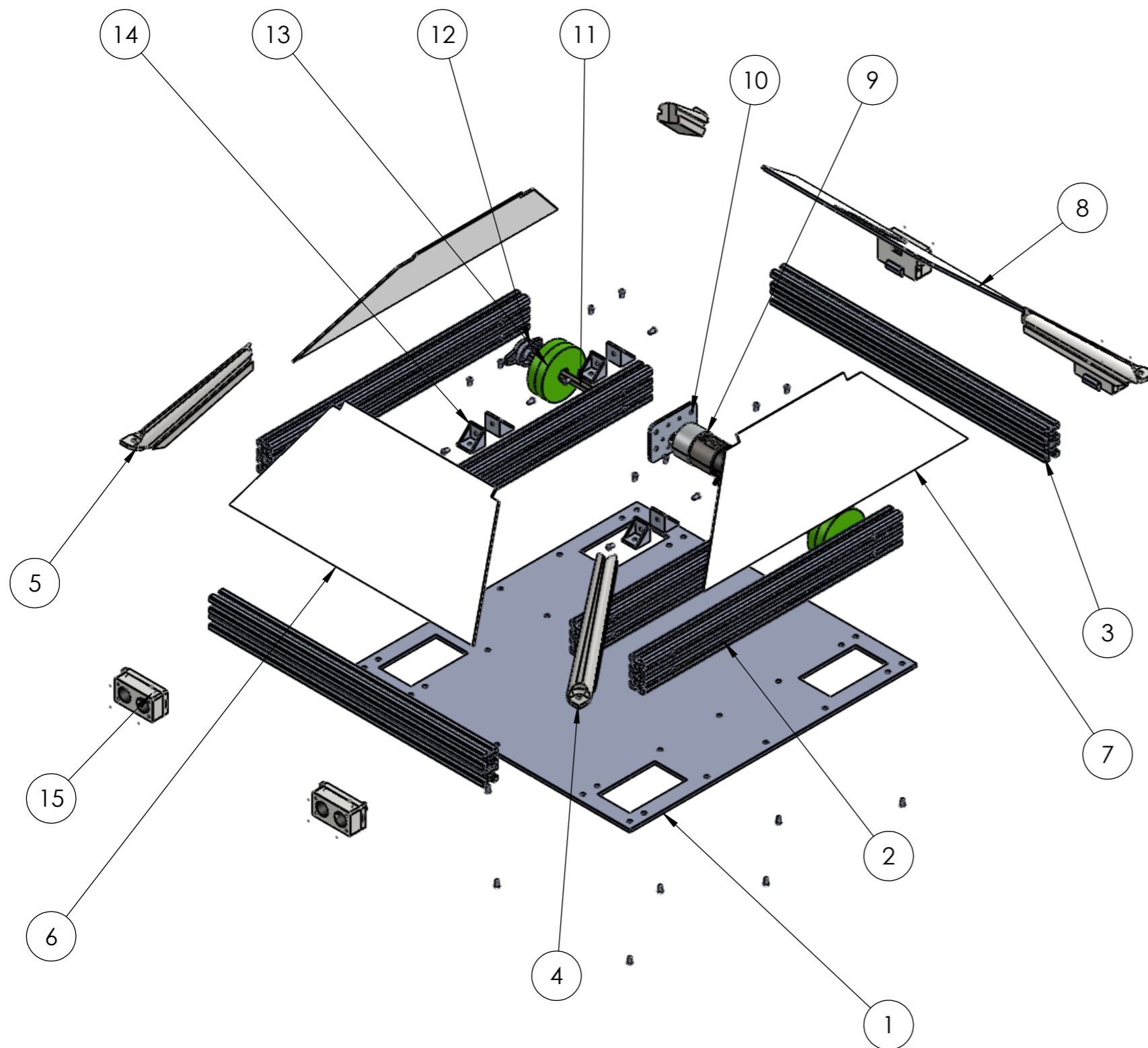
2021-07-15

HOJA: 2/47

MEDIDAS EN:
mm

MATERIAL: VARIOS

PL. #01-00-00

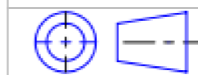


15	SENSOR ULTRASONIDO	4	VARIOS
14	UNION L	8	ALUMINIO
13	RUEDA BANEBOTS 8CM	2	VARIOS
12	CHUMACERA	4	VARIOS
11	EJE	2	ACERO INOXIDABLE
10	SOPORTE MOTORES	2	PLÁSTICO PLA
9	MOTOR 34D 5KG-CM	2	VARIOS
8	PLACA ACRÍLICO POSTERIOR	1	ACRÍLICO
7	PLACA ACRÍLICO LATERAL	2	ACRÍLICO
6	PLACA ACRÍLICO FRONTAL	1	ACRÍLICO
5	ESQUINERO IZQUIERDO	2	PLÁSTICO PLA
4	ESQUINERO DERECHO	2	PLÁSTICO PLA
3	PERFIL 2040 FRONTAL	2	ALUMINIO
2	PERFIL 2040 LATERAL	4	ALUMINIO
1	PLACA BASE	1	ALUMINIO
Nº	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:5

BASE



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

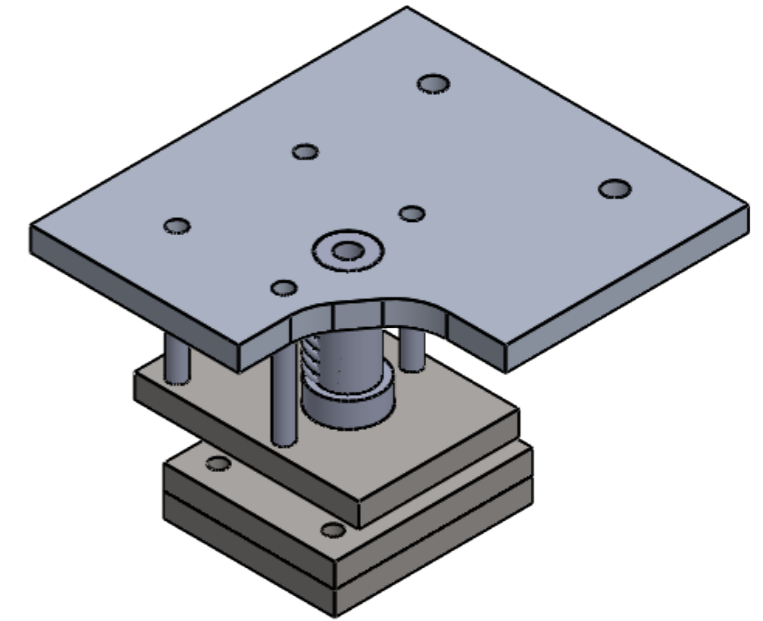
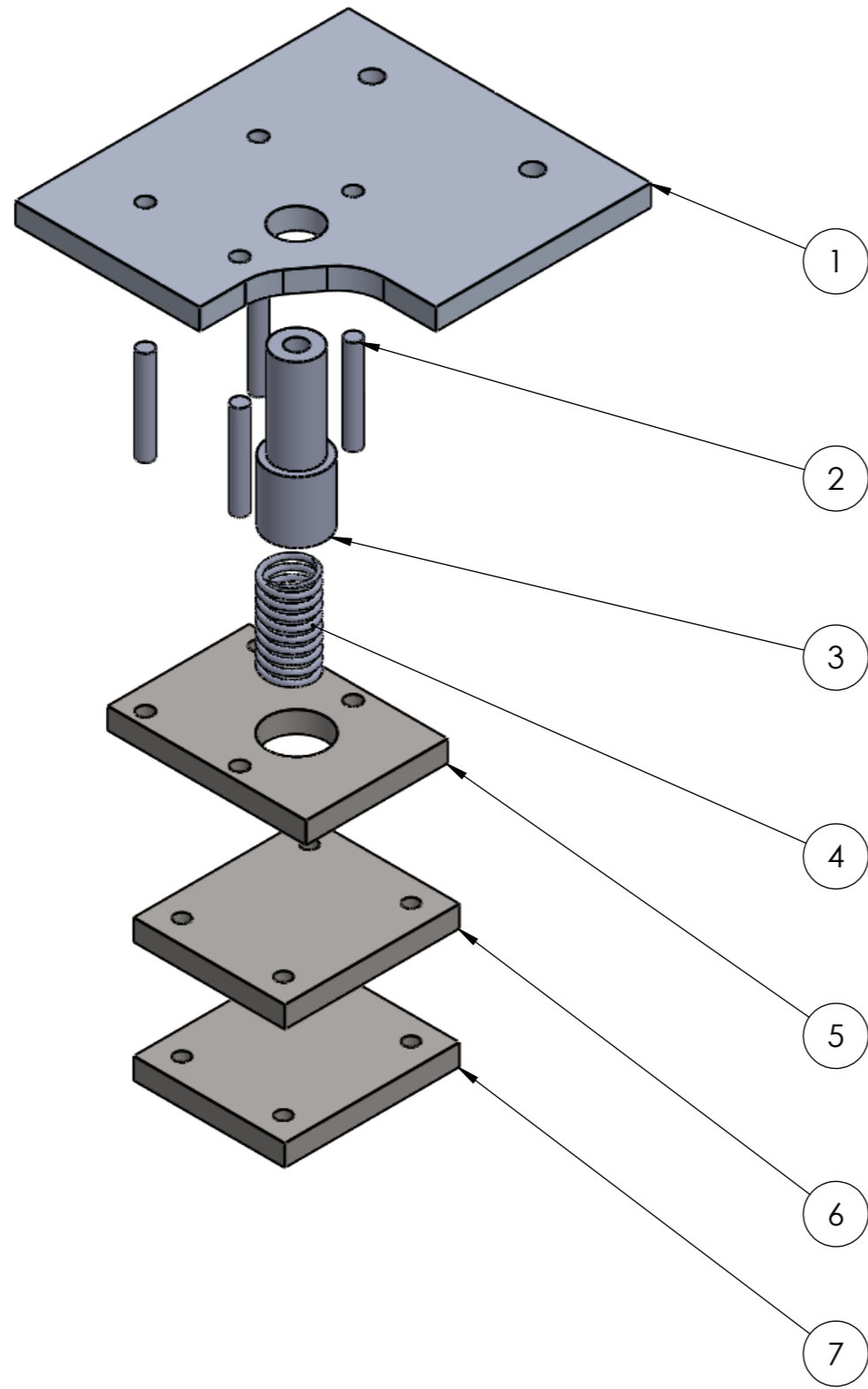
2021-07-15

HOJA 3/47

MEDIDAS EN:
mm

MATERIAL: VARIOS

PL. #01-01-00



7	RUEDA LOCA	1	VARIOS
6	PLACA APOYO EJE	1	ACERO INOXIDABLE
5	PLACA SECUNDARIA	1	ACERO INOXIDABLE
4	RESORTE	1	ACERO INOXIDABLE
3	EJE VERTICAL	1	ACERO INOXIDABLE
2	SEPARADORES	1	ACERO INOXIDABLE
1	PLACA PRINCIPAL	1	ACERO INOXIDABLE
N°	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:1

SISTEMA DE AMORTIGUADORES



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

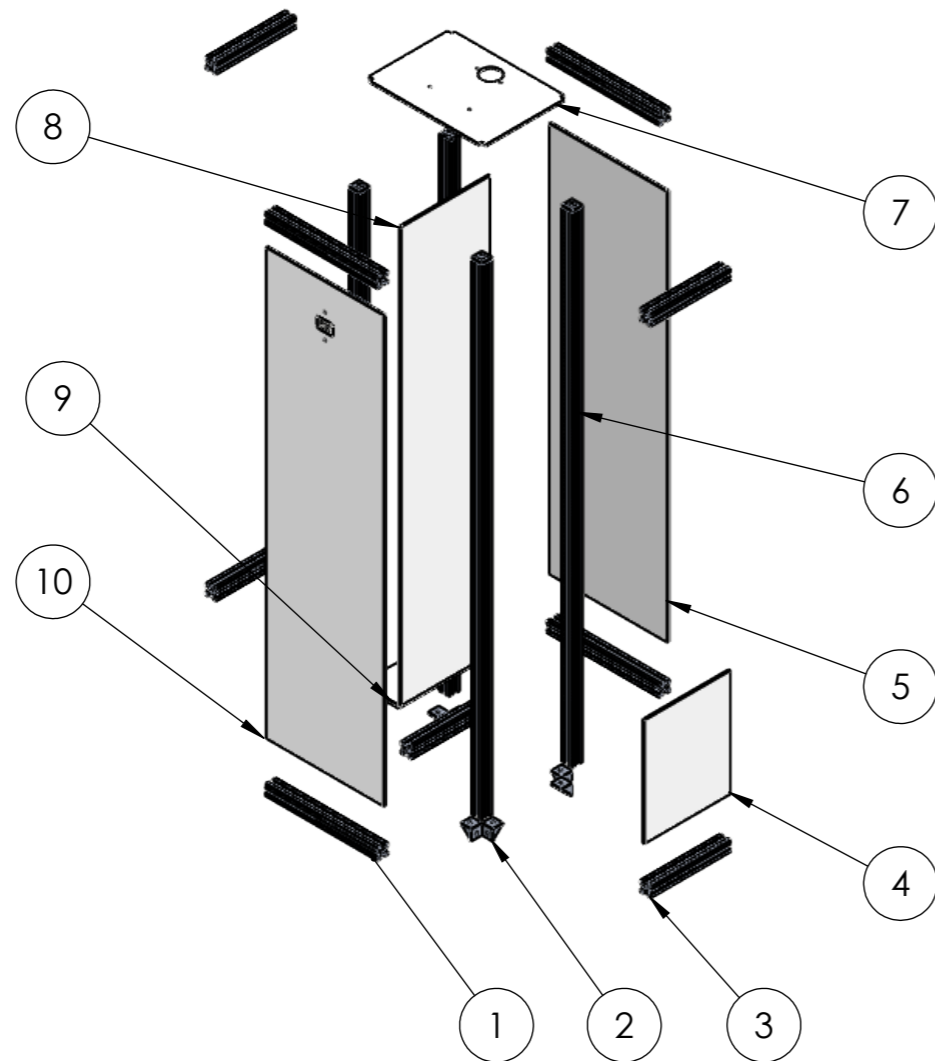
2021-07-15

HOJA 14/47

MEDIDAS EN:
mm

MATERIAL: VARIOS

PL. #01-02-00

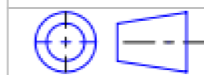


10	PLACA ACRÍLICO VERTICAL FRONTAL	1	ACRÍLICO
9	PLACA ACRÍLICO ACOSTADA	1	ACRÍLICO
8	PLACA ACRÍLICO VERTICAL LARGA	1	ACRÍLICO
7	PLACA ACRÍLICO LIDAR	1	ACRÍLICO
6	PERFIL 2020 VERTICAL PRINCIPAL	4	ALUMINIO
5	PLACA ACRÍLICO VERTICAL POSTERIOR	1	ACRÍLICO
4	PLACA ACRÍLICO VERTICAL CORTA	1	ACRÍLICO
3	PERFIL 2020 LATERAL	4	ALUMINIO
2	UNION L	8	ALUMINIO
1	PERFIL 2020 FRONTAL	4	ALUMINIO
Nº	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:10

CUERPO PRINCIPAL



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

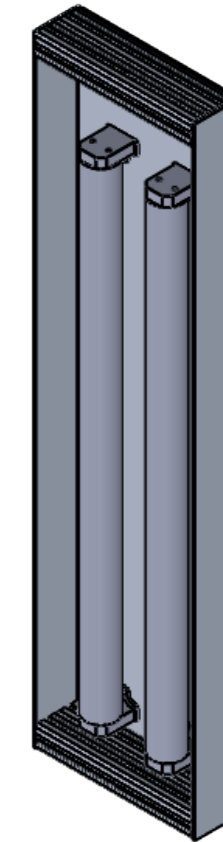
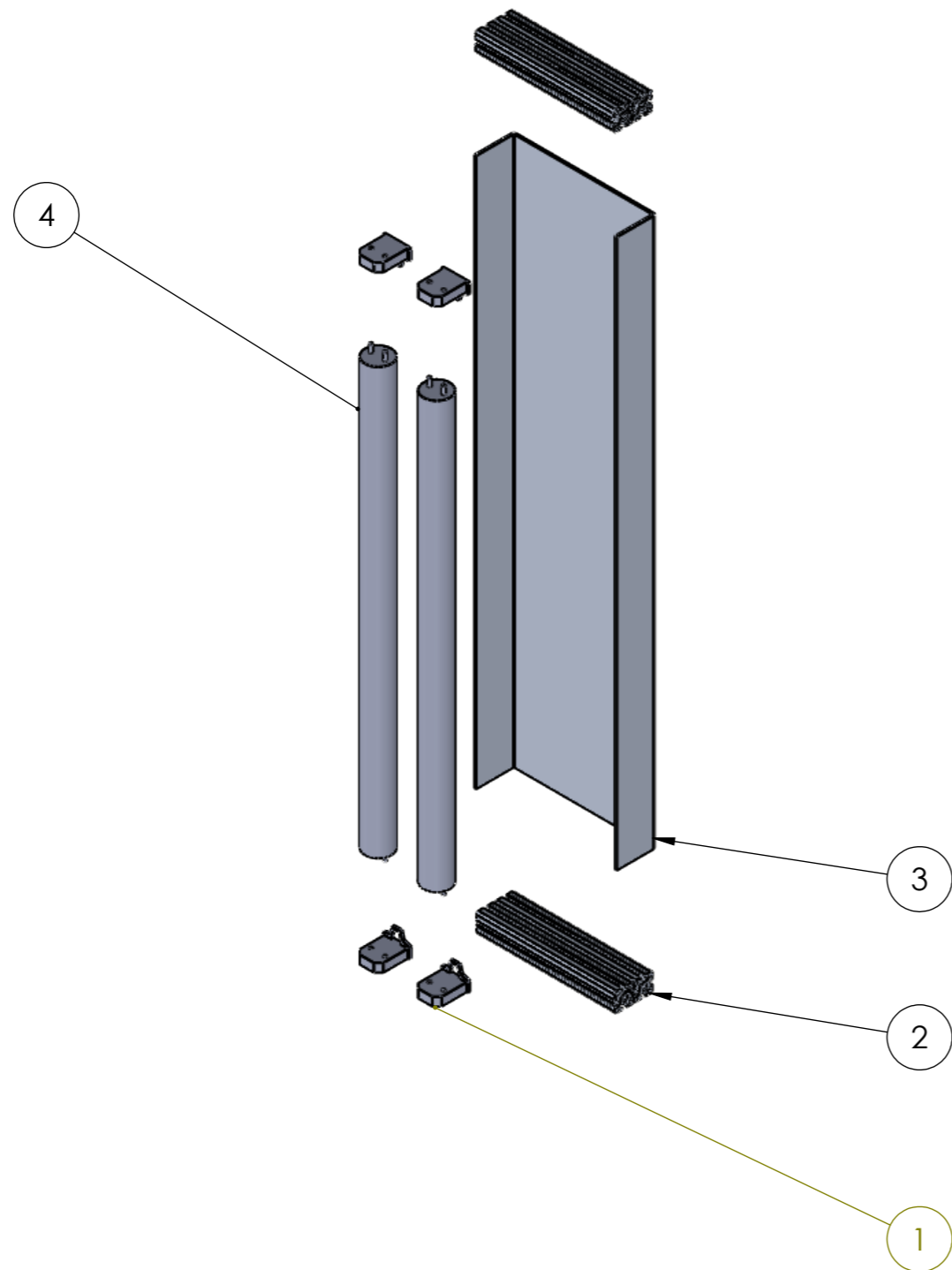
2021-07-15

HOJA 22/47

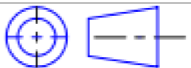
MEDIDAS EN:
mm

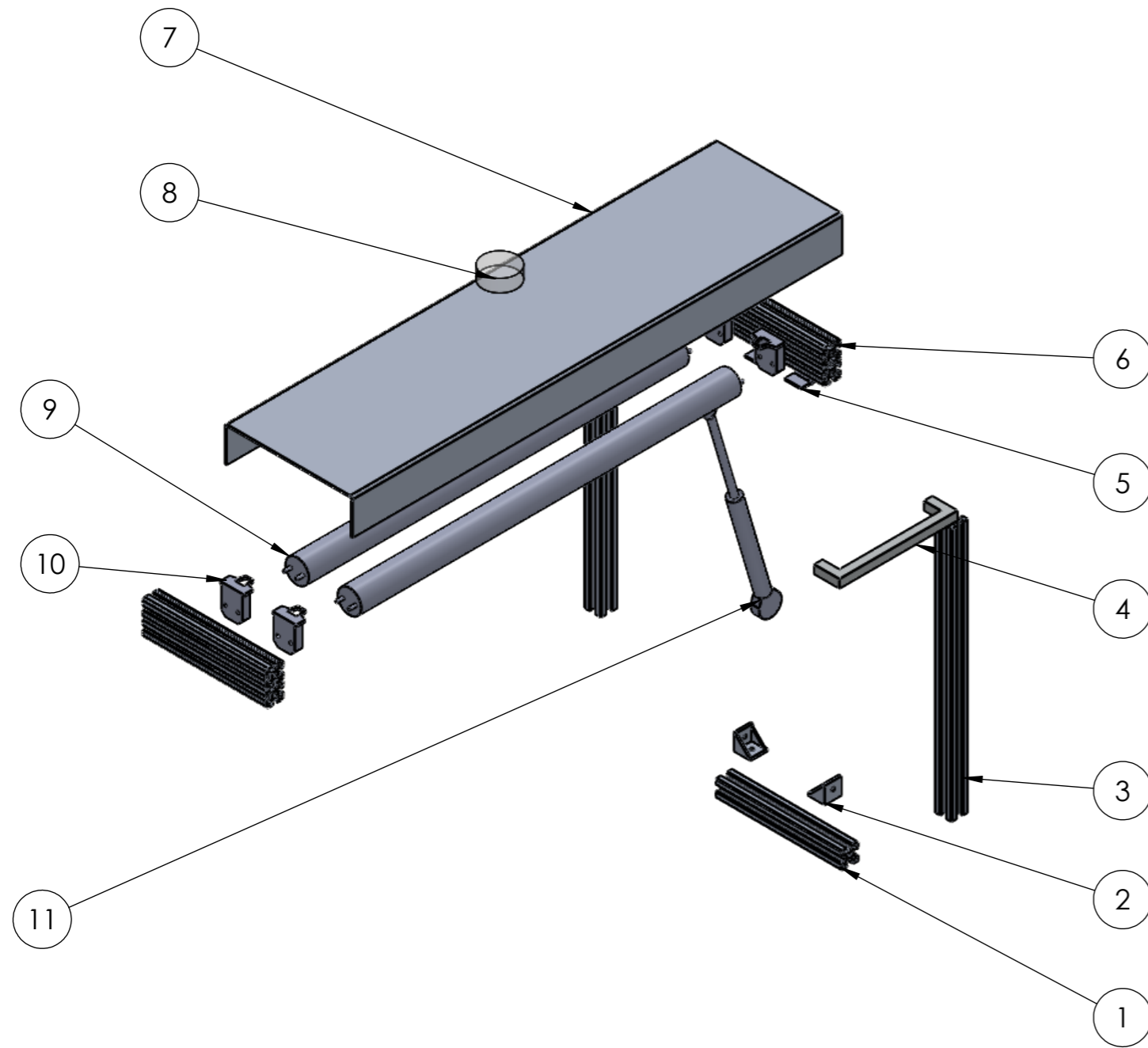
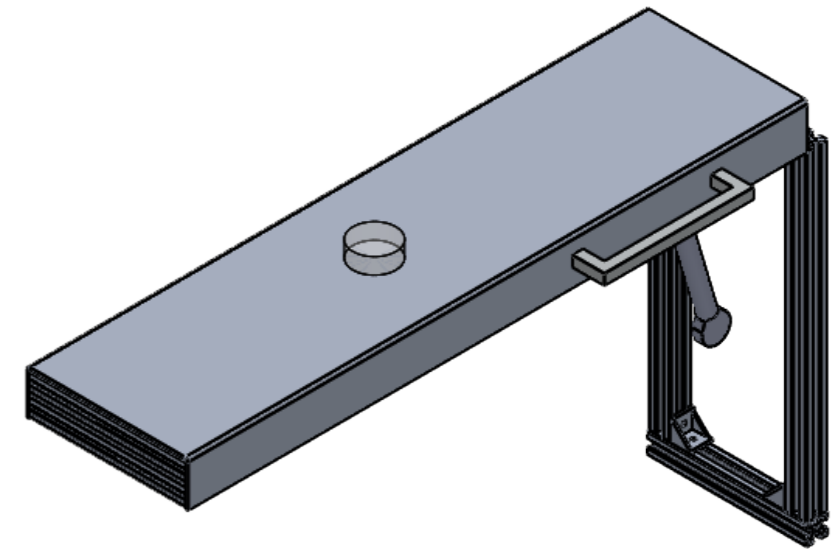
MATERIAL: VARIOS

PL. #01-03-00



N°	NOMBRE	CANT	MATERIAL
4	BOMBILLO UV-C 15W	2	VARIOS
3	CUERPO LÁMPARA	1	ALUMINIO
2	PERFIL 2040 SOPORTE	2	ALUMINIO
1	SOCKET T8	4	PLÁSTICO

ESCUELA DE INGENIERÍA MECÁNICA			
ESCALA: 1:5		LÁMPARA VERTICAL	
	DIBUJADO POR: JAVIER RANGEL JUAN VILLAMIZAR	2021-05-22	
FORMATO: A3	REVISADO POR: ING. JORGE MENESES	2021-07-15	HOJA 32/47
MEDIDAS EN: mm	MATERIAL: VARIOS		PL. #01-04-00



11	CILINDRO DE GAS	1	VARIOS
10	SOCKET T8	4	PLÁSTICO
9	BOMBILLO UV-C 15W	2	VARIOS
8	BALIZA	1	ACRÍLICO
7	CUERPO LÁMPARA	2	ALUMINIO
6	PERFIL 2040 TOPE	2	ALUMINIO
5	BISAGRA	2	ALUMINIO
4	MANIJA	1	ACERO INOXIDABLE
3	PERFIL 2020 GUIA	2	ALUMINIO
2	UNION L	2	ALUMINIO
1	PERFIL 2020 UNION	1	ALUMINIO
Nº	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:5

LÁMPARA HORIZONTAL



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

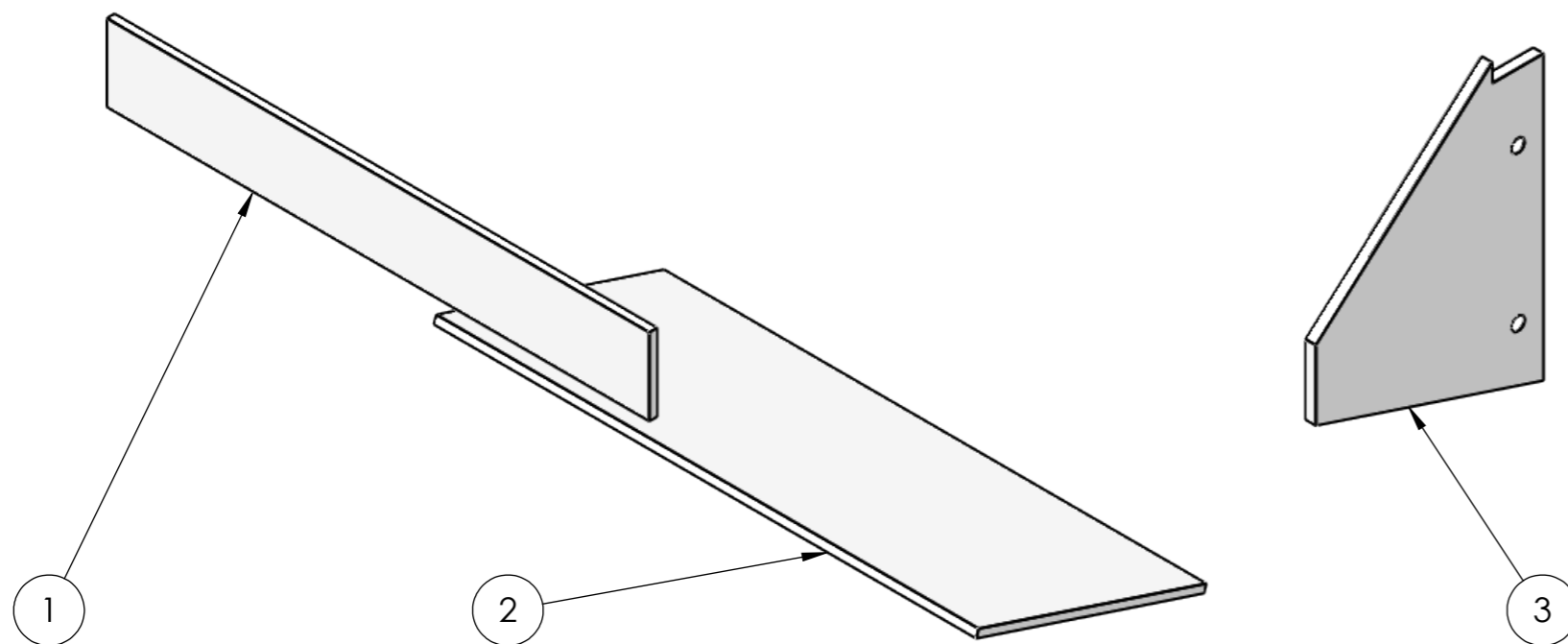
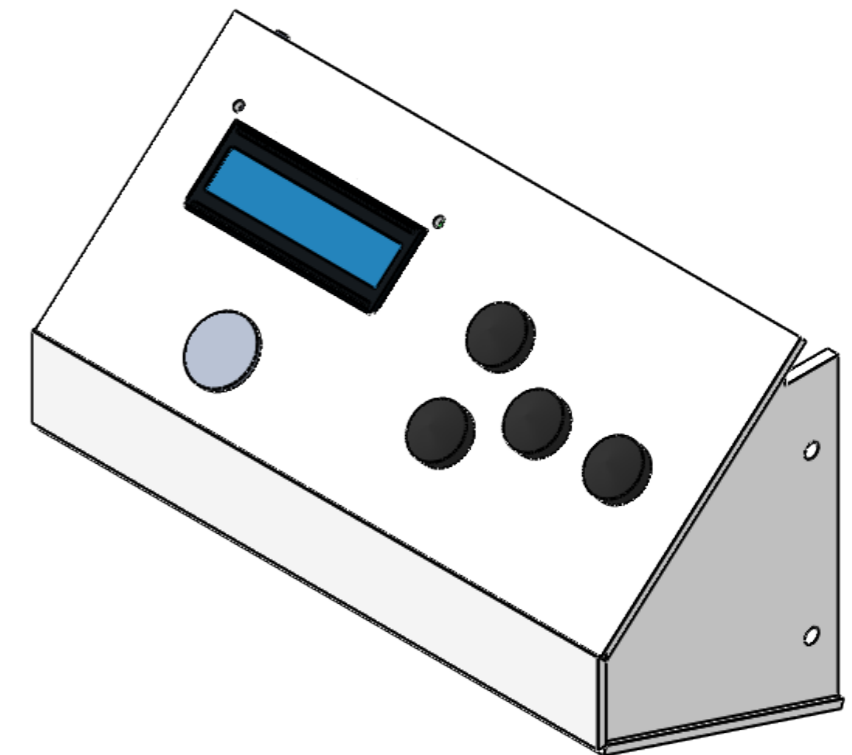
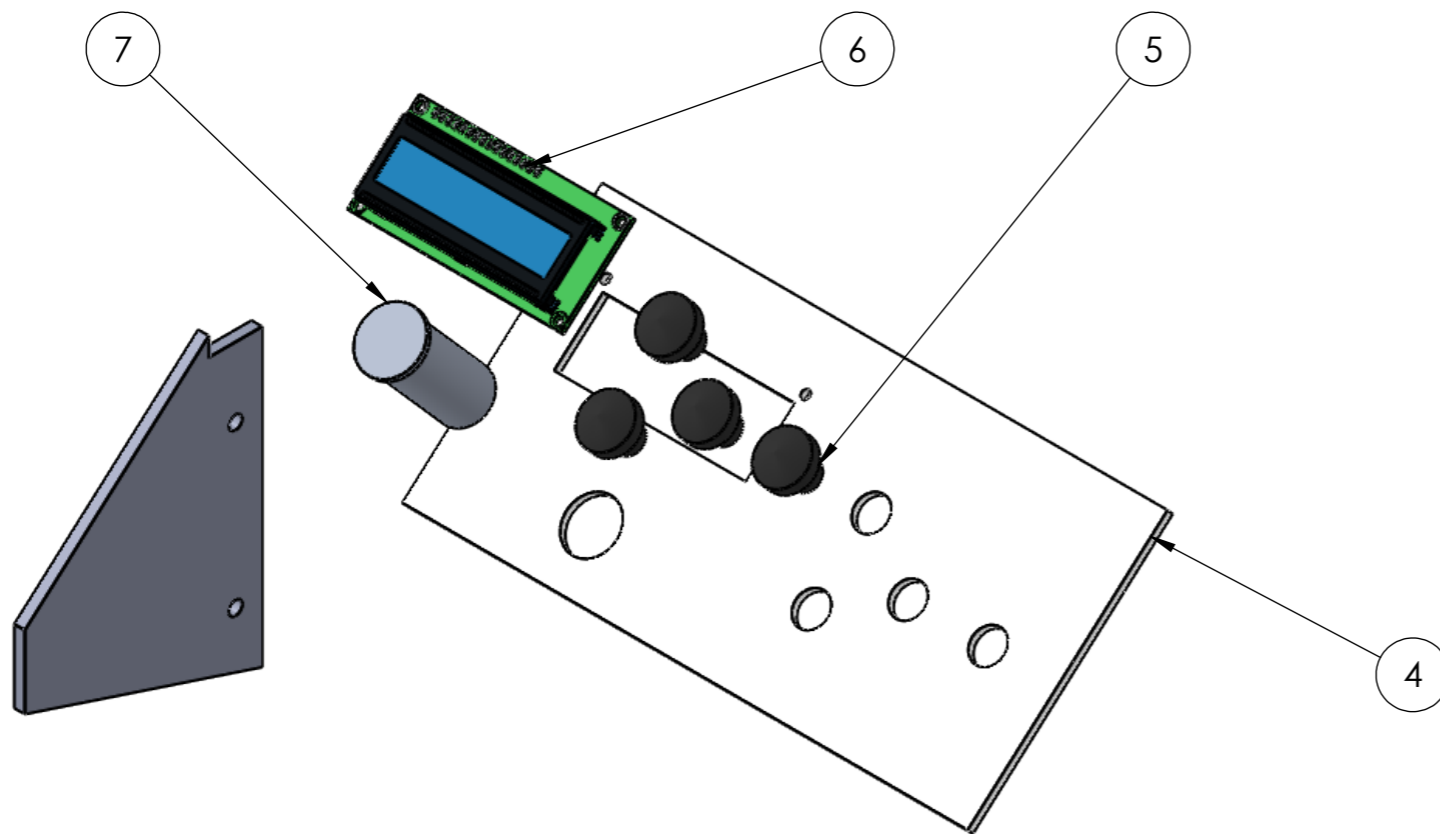
2021-07-15

HOJA 35/47

MEDIDAS EN:
mm

MATERIAL: VARIOS

PL. #01-05-00

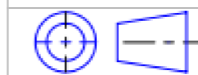


7	BOTON ENCENDIDO	1	VARIOS
6	DISPLAY LCD	1	VARIOS
5	BOTON	4	VARIOS
4	ACRILICO PRINCIPAL	1	ACRILICO
3	ACRILICO SOPORTE LATERAL	2	ACRILICO
2	ACRILICO INFERIOR	1	ACRILICO
1	ACRILICO COBERTOR	1	ACRILICO
N°	NOMBRE	CANT	MATERIAL

ESCUELA DE INGENIERÍA MECÁNICA

ESCALA: 1:2

INTERFAZ



DIBUJADO POR:
JAVIER RANGEL
JUAN VILLAMIZAR

2021-05-22

FORMATO:
A3

REVISADO POR:
ING. JORGE MENESES

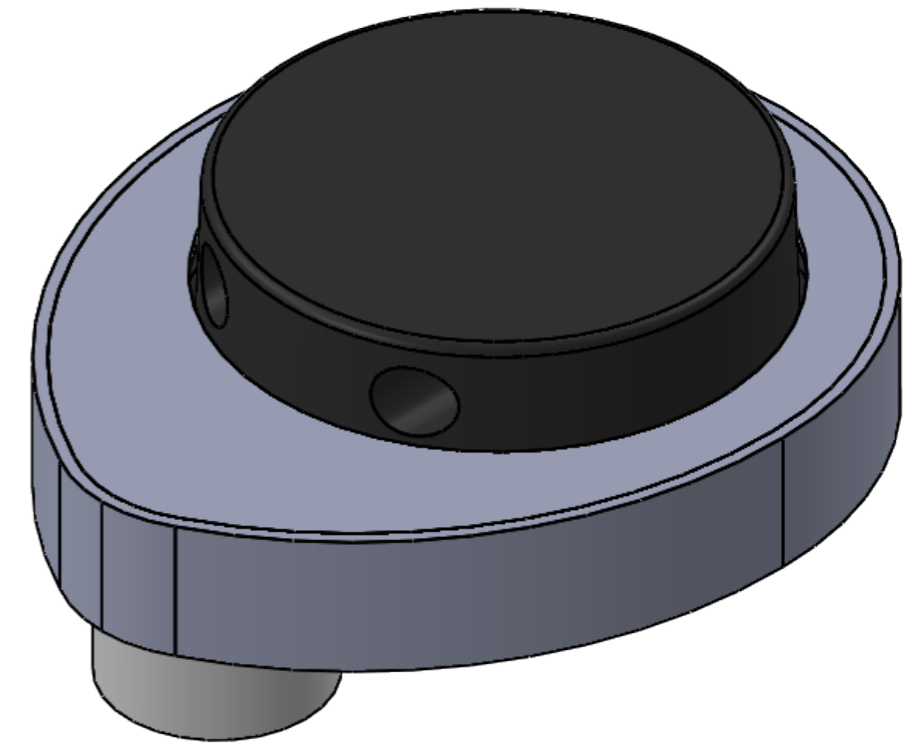
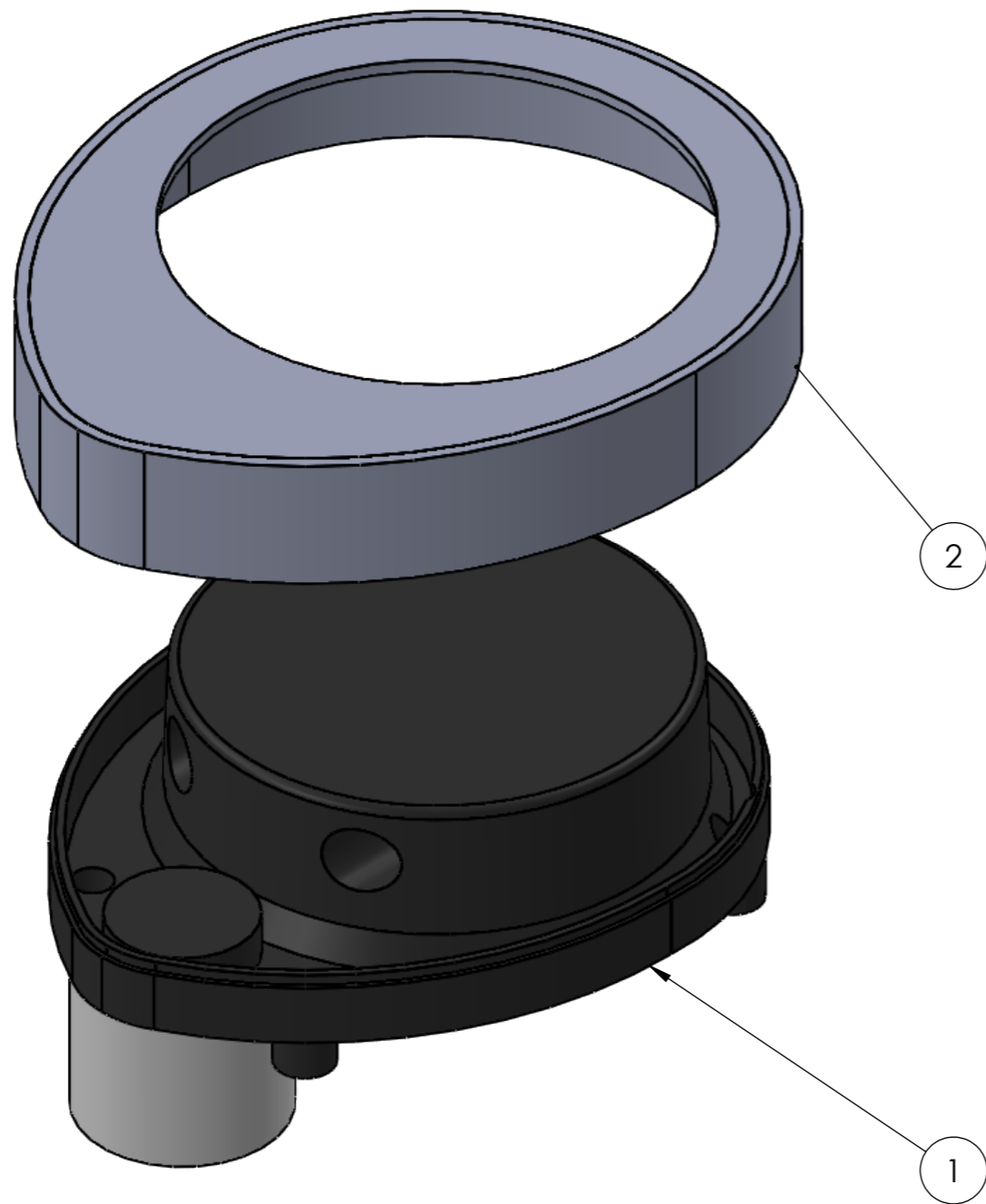
2021-07-15

HOJA 41/47

MEDIDAS EN:
mm

MATERIAL: VARIOS

PL. #01-06-00



2	CARCASA LIDAR	1	PLASTICO PLA
1	LIDAR XV-11	1	VARIOS
N°	NOMBRE	CANT	MATERIAL
ESCUELA DE INGENIERÍA MECÁNICA			
ESCALA: 1:1	LIDAR XV-11		
	DIBUJADO POR: JAVIER RANGEL JUAN VILLAMIZAR	2021-05-22	
FORMATO: A3	REVISADO POR: ING. JORGE MENESES	2021-07-15	HOJA 46/47
MEDIDAS EN: mm	MATERIAL: VARIOS		PL. #01-07-00