



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**IMPLEMENTACIÓN DE UN ROBOT DE DESINFECCIÓN
ULTRAVIOLETA PARA MICROORGANISMOS CON REGISTRO
VISUAL PARA ASISTIR EN LA LIMPIEZA DE ÁREAS EN
CUIDADOS INTENSIVOS**

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

AUTORES: Pablo Alfredo Hernández Bonnet
 Josué Enmanuel Navarrete Solano
TUTOR: Nino Tello Vega Ureta

Guayaquil-Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Pablo Alfredo Hernández Bonnet** con documento de identificación N° **0957832355** y **Josué Enmanuel Navarrete Solano** con documento de identificación N° **0940342769**; manifestamos que:

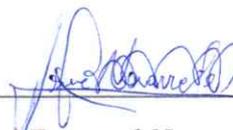
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 03 de marzo del año 2022

Atentamente,



Pablo Alfredo Hernández Bonnet
0957832355



Josué Enmanuel Navarrete Solano
0940342769

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

Nosotros, **Pablo Alfredo Hernández Bonnet** con documento de identificación N° **0957832355** y **Josué Enmanuel Navarrete Solano** con documento de identificación N° **0940342769**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: IMPLEMENTACIÓN DE UN ROBOT DE DESINFECCIÓN ULTRAVIOLETA PARA MICROORGANISMOS CON REGISTRO VISUAL PARA ASISTIR EN LA LIMPIEZA DE ÁREAS EN CUIDADOS INTENSIVOS**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Guayaquil, 03 de marzo del año 2022

Atentamente,



Pablo Alfredo Hernández Bonnet
0957832355



Josué Enmanuel Navarrete Solano
0940342769

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Nino Tello Vega Ureta**, docente de la Universidad Politécnica Salesiana , declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **IMPLEMENTACIÓN DE UN ROBOT DE DESINFECCIÓN ULTRA-VIOLETA PARA MICROORGANISMOS CON REGISTRO VISUAL PARA ASISTIR EN LA LIMPIEZA DE ÁREAS EN CUIDADOS INTENSIVOS**, realizado por **Pablo Alfredo Hernández Bonnet** con documento de identificación N° **0957832355** y por **Josué Enmanuel Navarrete Solano** con documento de identificación N° **0940342769**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 03 de marzo del año 2022

Atentamente,



Ing Nino Tello Vega Ureta, Mgs
0801602160

I. DEDICATORIA Y AGRADECIMIENTO

I-A. Dedicatoria

A mis padres, que con sus valores impecables me han ayudado a ser quien soy, también a mi hermana, mis tías, primas y todas las personas que no me han abandonado durante las distintas etapas de mi vida, sin ellos no habría podido continuar.

Pablo Alfredo Hernández Bonnet

Este trabajo es parte del esfuerzo y los valores con los que me han formado mi padres, les dedico a ellos este trabajo, también a mi familia y a todas las personas que estuvieron cerca para poder desarrollar este proyecto que tiene la finalidad de solventar problemas relacionados a la desinfección de las áreas médicas.

Josué Enmanuel Navarrete Solano

I-B. Agradecimiento

Le agradezco a Dios por darme todo lo que tengo y el privilegio de llevar a cabo mis estudios. A mi familia, por su comprensión, constante ánimo y apoyo en toda situación sin condiciones. A Giorgina Carmi y Joseph Guillén, por escucharme y aconsejarme para soportar psicológicamente los problemas que engloba cumplir una meta. A Samil Hernández y Gisella Hernández, por ilustrarme con sus conocimientos técnicos y ser ejemplos a seguir. Finalmente agradezco a mi compañero Josué Navarrete por todos los esfuerzos realizados a lo largo del trabajo de titulación.

Pablo Alfredo Hernández Bonnet

Muy agradecido con Dios por darme la iluminación y orientación día a día, por mantenerme vivo y con salud, también a mi familia que supo ayudarme y animarme para elaborar este proyecto, y a todos quienes de una u otra manera han colaborado con sus conocimiento para llevar a acabo la culminación de este trabajo de tesis.

Josué Enmanuel Navarrete Solano

II. RESUMEN

El presente trabajo de titulación tiene el propósito de diseñar e implementar un robot de desinfección UV, cuyo objetivo es asistir en el cuidado y bienestar proporcionando un método tecnológico de limpieza en las superficies de manera automática y frecuente, de este modo se facilitará la descontaminación de virus, bacterias y hongos presentes en los centros de salud.

El proyecto consiste en diseñar un robot que permita asistir a los riesgos que existen en la limpieza en la zona de trabajo médico, además se utilizará un movimiento omnidireccional así permitiendo tener un buen desempeño en el proceso de desinfección. Este tipo de robot será controlado de manera inalámbrica, estableciendo una conexión maestro esclavo entre los dos controladores, con un apoyo de software amigable, permitiendo de esta manera controlar la ubicación y estado el robot.

Se realizará la verificación de funcionalidad del robot al efectuar la desinfección en las zonas contaminadas (mediante ensayos de prueba y error) de esta manera se pretende servir de soporte a los empleados en el área de desinfección. Se ha descubierto que durante el colapso médico en los hospitales de todos los países que atravesaron la pandemia del COVID-19, se elevaron las expectativas tecnológicas en la limpieza de áreas médicas basándose así en técnicas de descontaminación de superficies por medio de luz ultravioleta y en la actualidad se continúa mejorando.

Se ha tomado en cuenta que la solución principal del proyecto de titulación radica en que la demanda de usuarios con diferentes patologías tiende a provocar aumento viral o bacteriano en las superficies de las áreas médicas, además otro de los posibles problemas estaría en el desconocimiento tecnológico de artefactos o elementos para la desinfección lo que permitirá el cuidado de una evolución de nuevas enfermedades.

Palabras clave: Desinfección UV, robot móvil, interfaz gráfica, software, trayectoria, omnidireccional.

III. ABSTRACT

The present degree work has the purpose of designing and implementing an UV disinfection robot, whose objective is to assist in the care and well-being, by providing a technological method of cleaning on surfaces automatically and frequently. For this reason can help to facilitate the decontamination of viruses, bacteria, and fungi present in health centers.

The project consists of designing a robot that allows attending to the risks that exist in cleaning in the medical work area. Moreover, an omnidirectional movement will be used, allowing so a good performance in the disinfection process. This type of robot will be controlled wirelessly, enabling a master-slave connection between the two controllers, with friendly software support, thus allowing control of the location and status of the robot.

Finally, the verification of the functionality of the robot will be carried out when carrying out the disinfection in the contaminated areas (through trial and error tests). In this way, it is intended to serve as support for the employees in the disinfection area.

It has been discovered that during the medical collapse in hospitals in all the countries that went through this pandemic (COVID-19) technological expectations were raised in cleaning expanded medical areas as well as surface decontamination techniques using ultraviolet light and currently it continues to improve. Even though, It has been taken into account that the main solution of the titling project lies in the fact that the demand of users with different pathologies caused a viral or bacterial increase in the surfaces of the medical areas.

In addition, another of the possible problems would be in the technological ignorance of artifacts or elements for disinfection which will allow the care of evolution of new diseases.

Keywords: UV disinfection, mobile robot, graphical interface, software, trajectory, omnidirectional.

ÍNDICE

I.	DEDICATORIA Y AGRADECIMIENTO	5
I-A.	Dedicatoria	5
I-B.	Agradecimiento	5
II.	RESUMEN	6
III.	ABSTRACT	7
IV.	INTRODUCCIÓN	13
V.	PLANTEAMIENTO DEL PROBLEMA	13
V-A.	Antecedentes	13
V-B.	Importancia y alcance	14
VI.	OBJETIVOS	16
VI-A.	Objetivo general	16
VI-B.	Objetivos específicos	16
VII.	FUNDAMENTOS TEÓRICOS	16
VII-A.	Robótica	16
VII-A1.	Robots móviles	17
VII-A2.	Estudio sobre ruedas	17
VII-A3.	Ruedas convencionales	17
VII-B.	Ruedas omnidireccionales	18
VII-B1.	Rueda universal	18
VII-B2.	Rueda mecanum	18
VII-C.	Lenguajes de programación utilizados en la robótica	19
VII-C1.	Lenguaje C++	19
VII-C2.	Lenguaje Python	19
VII-D.	Controladores	19
VII-D1.	Raspberry pi	19
VII-D2.	Arduino	19
VII-E.	Código QR	20
VII-F.	Robot omnidireccional	20
VII-G.	Sensores	20
VII-G1.	Sensores inductivos	20
VII-G2.	Sensores capacitivos	20
VII-G3.	Sensor de temperatura	20
VII-G4.	Sensores de distancia infrarrojos	21
VII-H.	Batería	21
VII-I.	Microorganismos	21
VII-J.	Tipos de microorganismos	21
VII-J1.	Bacterias	22
VII-J2.	Algas	22
VII-J3.	Protozoos	22
VII-J4.	Hongos	23
VII-J5.	Virus	23
VII-K.	Microbiología clínica	23
VII-L.	Microorganismos presentes en los ambientes médicos	23

VII-M.	Análisis microbiológico	24
VII-N.	Análisis microbiológico del suelo	25
VII-Ñ.	Análisis microbiológico del aire	25
VII-O.	Técnicas de análisis microbiano	25
	VII-O1. Recuento de células totales	25
	VII-O2. Recuento de células viables	26
VII-P.	Desinfección	27
VII-Q.	Herramientas para la desinfección	27
VII-R.	La radiación ultravioleta	28
VII-S.	Uso de la radiación ultravioleta para inactivar los microorganismos	28
VIII.	MARCO METODOLÓGICO	29
VIII-A.	Diseño de la estructura del robot	29
	VIII-A1. Materiales de la estructura	30
	VIII-A2. Diseño de tapa	30
	VIII-A3. Diseño de chasis	31
	VIII-A4. Diseño de Acople de motor-rueda	32
VIII-B.	Selección de ruedas	32
VIII-C.	Hardware electrónico	33
	VIII-C1. Etapa de alimentación	34
	VIII-C2. Etapa de procesamiento	37
	VIII-C3. Etapa sensorial	37
	VIII-C4. Etapa de potencia	38
	VIII-C5. Etapa de desinfección	38
	VIII-C6. Tarjeta de decodificación QR	38
	VIII-C7. Tarjeta de control	38
	VIII-C8. Tarjeta de conexión auxiliar	39
VIII-D.	Dimensionamiento de motores	40
VIII-E.	Diseño de software de control	41
	VIII-E1. Parámetros para el diseño de control	41
	VIII-E2. Diagnóstico de la señales de control	41
	VIII-E3. Control de sintonía fina de los motores	42
	VIII-E4. Diagrama de secuencia de funcionamiento	45
VIII-F.	Desarrollo del software del robot	46
	VIII-F1. Cálculo de la posición del eje del motor	46
	VIII-F2. Cálculo de la velocidad en RPM y rad/s	47
	VIII-F3. Cálculo del movimiento del robot	47
	VIII-F4. Calibración y configuración del sensor IR	48
	VIII-F5. Conexión y configuración de Raspberry PI	48
	VIII-F6. Algoritmo de lectura QR	54
	VIII-F7. Interfaz de visualización	55
	VIII-F8. Programación de movimiento y accionamiento de robot de desinfección	55
IX.	RESULTADOS	55
IX-A.	Resultados de las pruebas de desinfección	56
IX-B.	Resultados de la estructura del robot	57
IX-C.	Resultados del funcionamiento del robot	60
IX-D.	Resultados de tarjeta electrónica	61
X.	CRONOGRAMA DE ACTIVIDADES	63

XI. PRESUPUESTO	64
XII. CONCLUSIONES	65
XIII. RECOMENDACIONES	66
Apéndice A: Piezas de la estructura	69
Apéndice B: Diseño electrónico	74
Apéndice C: Procesos de montaje	78
Apéndice D: Planos de la estructura	89
Apéndice E: Algoritmo de lectura QR e interfaz de monitoreo	107
Apéndice F: Códigos de de arduino para control del sistema	112
Apéndice G: Movimiento de motores	115
Apéndice H: Implementación de control de señales mediante python	120

ÍNDICE DE FIGURAS

1.	Tipos de ruedas convencionales [22]	17
2.	Configuraciones de ruedas convencionales [13]	18
3.	Configuraciones angulares de la rueda mecanum [53]	19
4.	Tipos de microorganismos presentes en el medio y sus dimensiones [5]	22
5.	Persistencia de patógenos nosocomiales en superficies inanimadas [1]	24
6.	Cámara de recuento [35]	26
7.	Espectrofotómetro [35]	26
8.	Formas de raspado sobre el medio [49]	27
9.	Representación del espectro electromagnético. Zona de visibilidad 400 nm- 700 nm [33]	28
10.	Susceptibilidad de diferentes tipos de microorganismos [19]	29
11.	Ensamble de robot omnidireccional realizado en Inventor	30
12.	Diseño de la carcasa del robot	31
13.	Diseño 3D de chasis	31
14.	Diseño 3D de Acople del motor-rueda	32
15.	Diseño 3D de rueda tipo omnidireccional	33
16.	Diagrama de funcionamiento electrónico	34
17.	Diagrama de fuerza realizado en el software CAdesimu	35
18.	Selección de rango de distancia del sensor IR	38
19.	Diagrama de fuerza realizado en el software CAdesimu	40
20.	Señal de aproximación	42
21.	Señal de error	43
22.	Importación de librería numpy	43
23.	Parámetros para evaluar el control de sintonía fina	44
24.	Cambio del setpoint	44
25.	Señal de proceso en el control de sintonía fina	44
26.	Señal de control	45
27.	Conexión para el movimiento del motor	46
28.	Conexión para el movimiento del motor	48
29.	Raspberry Pi energizada	49

30.	Mensaje de bienvenida de Raspbian	49
31.	Configuración de región de Raspberry Pi	50
32.	Asignación de usuario y contraseña de usuario	50
33.	Actualización de software	51
34.	Actualización de programas predeterminados	51
35.	Comandos complementarios para instalación de librerías	52
36.	Instalación de librerías.	52
37.	Activación de comunicación VNC	53
38.	Selección de dispositivo mediante VNC viewer	53
39.	Librerías utilizadas para lectura QR en Python.	54
40.	Inicialización de cámara en Python.	54
41.	Bucle de decodificación QR en Python.	55
42.	Excepciones y ventana de video en Python.	55
43.	Crecimiento de bacteria en el medio	56
44.	Comparación del control con la desinfección	57
45.	Esfuerzo de Von Mises del chasis	57
46.	Coefficiente de seguridad del chasis del robot	58
47.	Perspectiva aérea de ensamble del prototipo	58
48.	Ensamble del prototipo	59
49.	Ensamble de dispositivos externos del robot	59
50.	Pruebas de interfaz gráfica	60
51.	Interfaz gráfica de monitoreo y control	61
52.	Diagrama de conexiones en software EasyEDA	61
53.	Parte trasera de tarjeta de control auxiliar	62
54.	Estructura de acrílico para aislar las placas	69
55.	chasis de la estructura del robot	70
56.	Ruedas omnidireccionales impresas en 3D	71
57.	Soporte de luces UV y tarjetas de control	72
58.	Tapa perteneciente a la estructura del chasis.	73
59.	Conexión de pines de driver L293D	74
60.	Diagrama de conexión interno de regulador Lm2596	74
61.	Cara inferior de tarjeta electrónica	76
62.	Cara superior de tarjeta electrónica	77
63.	Proceso de corte de base de acrílico	78
64.	CNC láser realizando cortes de piezas	79
65.	Proceso de impresión 3D	80
66.	Plancha de aluminio destinada a ensamblaje del robot	81
67.	Montaje de tapa ubicada en la parte superior del chasis	82
68.	Ubicación de lamparas UV dentro de la estructura	83
69.	Material destinado para la estructura del robot	84
70.	Ubicación de motores encoder en estructura del robot	85
71.	Montaje de ruedas omnidireccionales en prototipo	86
72.	Montaje de soporte de luces UV y tarjetas de control de lamparas	87
73.	Proceso de pintado de chasis de la estructura	88
74.	Plano de Tapa de chasis	89
75.	Plano de soporte principal de ruedas omnidireccionales	90
76.	Plano de base de acrílico	91
77.	Dimensiones de batería lipo 11.1V	92
78.	Plano de bocin para ruedas omnidireccionales	93
79.	Dimensiones de borneras de 2 pines	94

80.	Dimensiones de driver L293D	95
81.	Dimensiones de lamparas UV	96
82.	Dimensiones de modulo bluetooth HC-05	97
83.	Dimensiones de motor encoder CHR-GM25-370	98
84.	Dimensiones de Raspberry Pi 3A+	99
85.	Dimensiones de regulador de voltaje LM2596	100
86.	Dimensiones de relé 5V	101
87.	Dimensiones de rodillos de rueda omnidireccional	102
88.	Dimensiones de sensor IR GP2Y0A21	103
89.	Plano de soporte de luces UV y tarjetas de control de lamparas	104
90.	Plano de tapa del chasis de la estructura	105
91.	Dimensiones de ventilador DC 12V	106
92.	Codigo de interfaz y programas multihilos desarrollado en python	107
93.	Codigo de interfaz y programas multihilos desarrollado en python	108
94.	Codigo de interfaz y programas multihilos desarrollado en python	109
95.	Codigo de interfaz y programas multihilos desarrollado en python	110
96.	Codigo de interfaz y programas multihilos desarrollado en python	111
97.	Interfaz de monitoreo visual en desarrollo	111
98.	Parámetros de giro de motores	112
99.	Código que establece el sentido de giro de los motores	112
100.	Código que define la velocidad de los motores en RPM	114
101.	Código para indicar los movimientos de los motores	117
102.	Código para calibrar el sensor IR	118
103.	Código para controlar el encendido de las lámparas UV	119
104.	Cálculo del control de motores	120
105.	Control de lazo cerrado de los motores	121

ÍNDICE DE CUADROS

I.	Elementos correspondientes a la alimentación de 5V	35
II.	Elementos correspondientes a la alimentación de 11.1V	36
III.	Elementos correspondientes a la alimentación de 7.5V	36
IV.	Selección de fusibles	36
V.	Características del las luces UV	39
VI.	Características de motor encoder 140 RPM	41
VII.	Tabla de parámetros de giro del motor	42
VIII.	Cronograma de actividades del trabajo de titulación	63
IX.	Presupuesto del trabajo de titulación	64
X.	características de de regulador Lm2596 [47]	74
XI.	Comparación de motores encoder con distintos RPM	75
XII.	Especificaciones de Raspberry Pi 3A+[23]	75
XIII.	Abreviaturas de diagrama de fuerza	75
XIV.	Características de driver L293D [2]	76

IV. INTRODUCCIÓN

En la actualidad el desarrollo vertiginoso de la tecnología se ha transformado en el consumo de diversos productos, lo que conlleva a cambios en la mente, forma de vivir, hacer y pensar del ser humano.

En el área científica y tecnológica se ha podido construir robots sofisticados, demostrando una mejor precisión en los movimientos, así como que actúen en condiciones de desgaste excesivo o que puedan permanecer estáticos durante horas.

El campo de la ingeniería ha demostrado avanzar de manera tecnológica permitiendo dar una fuerza al campo de la robótica. El progreso ha permitido que actúen en condiciones de desgaste excesivo o que estén habitualmente estáticos durante horas. Para obtener todo este ascenso tanto en producción, como en la ayuda de áreas médicas, no solo se pretende seleccionar aparatos mecánicos y electrónicos que reconozcan las diferentes necesidades, es inevitable contar con modelos matemáticos que accedan en la intervención de los sistemas y tomen decisiones primordiales al momento de su implementación.

El robot de desinfección pretende colaborar con la desinfección de las áreas médicas por motivo a que existe mayor frecuencia de microorganismos virales y bacterianos en dichos lugares, permitiendo tener un ambiente seguro y libre.

Dicho robot mantiene en su estructura digital un protocolo de comunicación maestro esclavo entre dos tipos de controladores que permite generar acciones y registros de datos demostrando eficacia al efectuar el propósito de generar descontaminación, además se pretende tomar un registro de datos mediante el uso de la visualización de un código QR, lo que permite identificar las zonas donde se encuentra y de esta manera realice la acción de desinfectar.

V. PLANTEAMIENTO DEL PROBLEMA

V-A. *Antecedentes*

En la antigüedad, los médicos solían tratar las infecciones de manera empírica utilizando una cantidad imprudente de antibióticos, los cuales alteraban el sistema de los pacientes y como resultado se generaba una cepa resistente a los tratamientos antimicrobianos de esta manera aumentaba el tiempo que podía durar el microorganismo en el cuerpo humano empeorando su cuadro clínico, dando paso a enfermedades autoinmunes que afectaban al personal de salud que se encontraba ayudando en las intervenciones médicas [24] .

En zonas hospitalarias, los grandes problemas que persisten en los ambientes médicos son las infecciones por la exposición y transmisión de bacterias, virus y parásitos que constituyen factores de riesgo biológico. Esto lo convierte en una fuente potencial de contagio para los pacientes y trabajadores del lugar, provocando infecciones hospitalarias, intrahospitalarias o nosocomiales. La OIT (Organización Internacional del Trabajo) detalla que alrededor de 337 millones de personas son víctimas de riesgos en accidentes o enfermedades dentro de su ambiente laboral [52]. Las enfermedades que se producen debido a la falta de aseo en las edificaciones pueden generar inconvenientes tanto en los trabajadores del lugar como a los ciudadanos que se encontraban en aquel lugar para tratar algún trámite o labor pendiente. Por otro lado, esta información da lugar a un enfoque prioritario de los lugares donde se realizan tratamientos médicos debido a la cantidad de microorganismos nocivos que rodean el área.

Por otra parte, existe un desconocimiento de nuevas tecnologías o dispositivos médicos que contribuyen en la desinfección, de la misma manera existen dudas al considerar plantear un sistema de desinfección de alto nivel y desempeño debido a sus altos costos [30]. También se percibe desconocimiento en el manejo de tecnología UV en centros médicos, el personal se crea una idea equivocada especulando y adquiriendo preocupaciones que radican en temas como el contenido de mercurio que posee la tecnología, la alta inversión que requiere o la existencia de supuestos problemas de caídas de tensión en los sistemas eléctricos debido a la misma [36].

Basado en un estudio que evaluó el desarrollo de la tecnología innovadora de luz ultravioleta por medios electrónicos se demostró, mediante una prueba, el efecto de la radiación ultravioleta. El estudio pretendía probar la luz ultravioleta 3 habitaciones distintas ubicando la luz ultravioleta en la parte inferior de los aires acondicionados. Los resultados detallaron que en los 3 cuartos existía concentraciones ambientales de flora aerobia, sin embargo durante el accionamiento de la lámpara de luz ultravioleta se notó que la calidad del aire en los cuartos estaba descontaminada, logrando mejorar la situación ambiental de las habitaciones [14].

El momento que atravesamos debido al virus del COVID-19 ha generado un alto índice de contagios provocando de esta manera que funcionarios de las entidades hospitalarias o los pacientes internados sean un blanco fácil para infectarse. Esto ocasiona que la exigencia de sistemas de control y desinfección se establezcan como prioridad para ayudar a solucionar dicha situación. En Ecuador para precautelar estas infecciones se aplican los principios de bioseguridad que involucran a los pacientes y al personal médico, esto es el uso de material adecuado como gorros, guantes y dispositivos de protección respiratoria que impida el contacto con los microorganismos. Además, en situaciones de brotes el área de infectología del hospital utiliza bioluminiscencia para detectar el trifosfato de Adenosín, que permite determinar la suciedad biológica del área. El Ministerio de Salud Pública indica que las superficies limpias y desinfectadas consiguen reducir cerca de un 99 % de los microorganismos presentes en los suelos, en tanto que las superficies que solo fueron limpiadas disminuyen un 80 de microorganismo [20].

Teniendo en cuenta que nuestro diseño está enfocado para áreas semicríticas como son, unidad de cuidados intensivos e intermedios, neonatales, laboratorios, bancos de sangres y urgencias, este proyecto pretende garantizar el proceso de desinfección de estas áreas, convirtiéndolo de un método convencional a un procedimiento metódico automatizado para lograr seguridad de los trabajadores en las tareas que están expuesta su salud.

Por ello se concluye que se debe profundizar en la creación de prototipos que logren disminuir el riesgo del contacto directo del personal al momento de realizar un proceso de desinfección y una forma es utilizando un método de automatización de manera que contribuya con una desinfección óptima en un área constantemente expuesta a todo tipo de virus, logrando así reducir las probabilidades de transmisión de infecciones.

V-B. Importancia y alcance

Nunca hay que dejar de tener en cuenta lo importante que es el aseo para la salud del ser humano, no es nada fuera de lo común que un establecimiento que cumple con todas las medidas de salubridad transmita una sensación de seguridad a todos los ciudadanos que se encuentran realizando alguna actividad en el área. Cuando hablamos de un establecimiento donde se practica la medicina, dicha sensación debe de extenderse en cada esquina del edificio, debido a que la mínima muestra de suciedad o falta de consideración en el aseo de cualquier zona, a pesar de que no represente un área relevante para los pacientes, puede provocar una mala impresión en todos los aspectos del servicio del centro de salud, aunque estos no se encuentren fallando en sus funciones principales. La mayor parte de los establecimientos médicos tienen protocolos y reglas que se rigen con mucha intensidad y consideran que la limpieza es uno de los factores más importantes para atender ciudadanos que se encuentran en busca de atención médica. Sin embargo, todos estos protocolos toman en cuenta que el personal que está encargado del aseo de las instalaciones son seres humanos que simplemente trabajan para obtener un sueldo y poder alimentar a su familia y cubrir sus respectivas necesidades. Los trabajadores de carne y hueso presentan fatiga tanto física como psicológica, cometen errores con frecuencia y a ocasionalmente desisten de realizar sus funciones por capricho o en forma de protesta. No obstante, el mismo ser humano ya se ha puesto en marcha para solucionar sus propias falencias aplicando soluciones tecnológicas que cubren sus necesidades en algunos casos de forma perfecta. Esto también aplica en el área de aseo de las edificaciones donde realizan todas sus labores diarias.

Introducir la tecnología en el campo de la limpieza representa una ventaja enorme para la supervivencia diaria del ser humano y otorga la oportunidad de conseguir modernizar los métodos tradicionales de aseo consiguiendo el máximo nivel de pulcritud posible, esto quiere decir que se eliminaran por completo los errores que el ser

humano ocasiona al trabajar en la limpieza de un lugar debido a su fragilidad o inestabilidad tanto emocional como física. No obstante, no hay que echarle la culpa a las personas que realizan esta labor, debido a que este problema se extiende en todas las áreas existentes en la historia de la humanidad. Cuando los ciudadanos de un país observan que se lleva a cabo un procedimiento que se considera tecnológico y representa a simple vista una mejora considerable en algún servicio que es utilizado de manera frecuente aceptan utilizarlo y les genera la sensación de que su contribución de pago de impuestos al estado está siendo bien utilizado, como, por ejemplo: el desarrollo de la industria del transporte público, que evoluciono en varios países para cambiar la modalidad de buses a trenes. Por esta misma razón innovar utilizando autómatas que realicen las funciones de limpieza de los humanos en las áreas médicas siempre será estimado por los pacientes y empleados que se encuentran en la instalación, más aún teniendo en cuenta que la mayoría de los centros hospitalarios funcionan a toda hora. A pesar de ello, estos sistemas modernos de limpieza no han sido normalizados del todo y probablemente requiera que muchos centros de salud alrededor de todo el mundo los comiencen a utilizar con el objetivo de crear una tendencia tecnológica y mantenerla en continua mejora con una proyección llamativa y duradera. La forma en la que actualmente se mejora la eficiencia de las labores que requieren movimiento constante y repetitivo, como lo es la limpieza, es utilizando robots para que realicen dichas acciones monótonas de forma diaria y con un alto nivel de eficiencia y confiabilidad. Automatizar el aseo de las superficies de los edificios hospitalarios en términos generales, consiste en reemplazar y mejorar las funciones de limpieza de los trabajadores en los pasillos, salas, quirófanos, áreas de emergencia y todas las zonas que son utilizadas para ejercer la rama de salud implementando robots programados para recorrer y limpiar los lugares mencionados de manera rutinaria. Sin embargo, un problema que se busca resolver cuando se abarca el tema de la limpieza, es la desinfección de virus o bacterias que poseen un nivel de transmisión elevado, lo cual provoca que estén presentes en las superficies de los centros médicos de forma frecuente y el proceso de desinfección requiera una solución tecnológica y moderna que se mantenga constantemente eliminándolos. Tomando en cuenta este problema, una medida por la cual se puede optar es la desinfección mediante rayos ultravioleta, la cual combinada con la participación de un robot móvil que recorra el lugar, puede llegar a eliminar todos estos organismos indeseados que se cuelan en las superficies de la estructura donde se proporciona el servicio médico.

Incluso no solo puede solucionar problemas técnicos, sino que también es un recurso que puede ayudar al medio ambiente debido a que, al modernizar los métodos de aseo superficial mediante su automatización, se volvería menos necesaria la utilización de productos químicos de limpieza los cuales pueden llegar a contaminar el aire que respiramos de la misma forma, o incluso, más que los vehículos de transporte. Estos productos químicos al mismo tiempo representan un gasto necesario de los centros de salud para mantener la limpieza, por lo tanto, al eliminar dicho gasto se podría ahorrar fondos que se encuentran destinados a este sector y dedicarlos a otras áreas igual de importantes. Esta múltiple solución de problemas económicos y ambientales que existe de forma pasiva debido a la utilización de productos químicos le aporta gran valor a la investigación y el desarrollo de sistemas de limpieza modernos. Visualizando uno de los problemas más grandes que tenemos en la actualidad, la pandemia mundial del COVID-19, la tecnología empieza a tomar más importancia para la humanidad debido a la alta tasa de contagios que representa la existencia de este virus, incluso en los hospitales de alto nivel que mantienen estándares de aseo elevados, los funcionarios de entidades hospitalarias como pacientes internados son blanco fácil para infectarse, normalmente las técnicas que implantan en la descontaminación de las áreas médicas son por agentes químicos que inactivan al microorganismo, pero no lo elimina. Por otro lado, considerando que en esta situación hay que evitar a toda costa el contacto de la piel con superficies, personas y cualquier zona en la cual pueda residir el COVID-19 y que se utilizan diariamente más agentes químicos de limpieza que antes para poder intensificar la erradicación del virus, la puesta en práctica de un robot móvil de desinfección mediante rayos ultravioleta que sea controlado ya sea a distancia o automáticamente, sería una gran contribución a la lucha que se tienen día a día para poder salvar vidas contra el coronavirus.

El desarrollo de este robot puede contribuir con información acerca del recorrido que debería de tener un robot móvil de desinfección al momento de realizar la limpieza por los pasillos o salas de los centros de salud en los cuales se solicite. También es importante tener en cuenta el estudio de obstáculos a los cuales se enfrenta un robot en un establecimiento que posee aglomeración de personas. A nivel educativo el estudio de soluciones mediante la

automatización es un factor que debe de ser innovador y encontrarse en constante desarrollo, por lo tanto, a nivel personal el desarrollo de un robot es un aporte significativo tanto a la sociedad como a mi crecimiento personal e incluso el de cualquier persona que estudia ingeniería o carreras afines. Claramente al tratarse del estudio de un método de limpieza, este no solo aplica para centros médicos, sino también puede ejercerse en otros ambientes como transportes públicos, centros comerciales o restaurantes. Al mismo tiempo puede incluirse en investigaciones que abarquen métodos de eliminación de virus o bacterias y contribuye en la verificación de datos de la eficiencia de los rayos ultravioleta en labores de desinfección. También aporta en fenómenos globales e incalculables como una pandemia mundial o la existencia de una nueva enfermedad contagiosa.

VI. OBJETIVOS

VI-A. *Objetivo general*

Desarrollar un robot móvil de desinfección ultravioleta con manejo autónomo mediante el control de sistemas embebidos, con el fin de proporcionar soporte al personal de desinfección en áreas médicas.

VI-B. *Objetivos específicos*

- Diseñar los prototipos de la estructura del robot mediante el software inventor, con el fin de conocer la resistencia y estructura adecuada.
- Realizar el diagrama eléctrico y electrónico del proyecto mediante el software Proteus con el fin de seleccionar las protecciones y materiales eléctricos del sistema.
- Desarrollar el software de configuración, monitoreo y control que permita la movilización del sistema con la finalidad de obtener autonomía en su funcionalidad y transmisión de datos.
- Verificar la eficiencia del sistema desarrollado mediante pruebas en la etapa de desinfección con el fin de comprobar el funcionamiento del robot y la esterilización de organismos no visibles.

VII. FUNDAMENTOS TEÓRICOS

VII-A. *Robótica*

Tanto la automatización y la robótica en la actualidad no se muestra comúnmente en lugares como centros de salud, pero es poco probable que no existan en industrias [52]. Lo que ha cambiado es el ritmo de su evolución y utilización. Amazon, siendo una de las empresas élite en producción y comercialización de productos utilizan robots en sus almacenes aumentando de 1.400 a 30.000 dólares de ganancias en menos de tres años.

Se comprende por robótica como la disciplina encargada del diseño e implantación de un robot, los cuales son máquinas capaces de realizar tareas propias dirigidas por un sistema programado, demostrando ser idóneos en formar parte de procesos que requieran más eficacia y menos riesgo.

Podemos aclarar que la robótica fusiona múltiples disciplinas, de la cuales podemos mencionar: electrónica, mecánica, informática, inteligencia artificial, entre otras [54].

Con la robótica se pretende implementar sistemas electromecánicos de nivel lógico que demuestren la apertura tecnológica en el alivio a los trabajadores en sus tareas repetitivas las cuales generarían daños de trastornos musculares u originando daños en su salud [26].

VII-A1. *Robots móviles:* A medida que la tecnología fue avanzando las formas de movimientos en los robots aportaron una mayor eficiencia. En la actualidad nos percatamos que existe dos formas muy habituales de movimiento en robots de la cuales estaríamos hablando, de robots móviles con ruedas y robots con patas. En el caso de los robots móviles se comprenden como aquellos que pueden dirigirse desde un lugar a otro autónomamente con respecto a su movimiento.

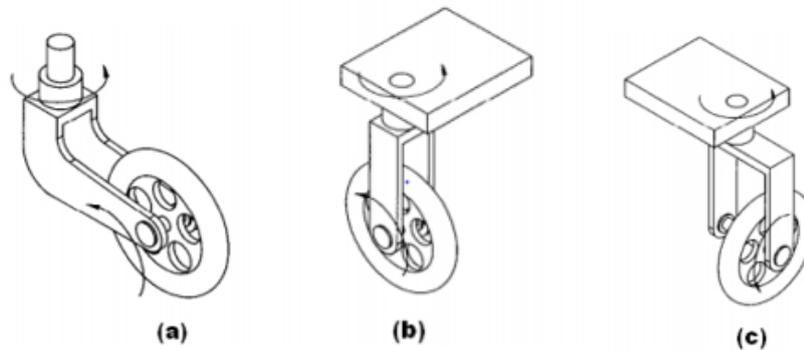
Estos robots móviles tienen la característica de moverse en un espacio de trabajo predeterminado, permitiendo de esta manera aplicarlo en diversos terrenos [38].

VII-A2. *Estudio sobre ruedas:* Es importante requerir de un tiempo en el acceder a manejar el robot en condiciones de terreno anormales, pues ello nos permitirá escoger los artículos adecuados a soportar el impacto que realiza dicho robot, es decir las ruedas.

Dependiendo de la actividad que fuere a realizar un robot de movimiento dependerá las piezas que se debe implantar.

Una de las principales piezas que implementa un robot de movimiento son sus ruedas que permiten el movimiento, desplazamiento o incluso de su elevación. Es por ese caso que se debe requerir un proceso de estudio en el impacto de las ruedas que nos permita manejar al robot durante un tiempo destinado en las condiciones de terrenos anormales [16]

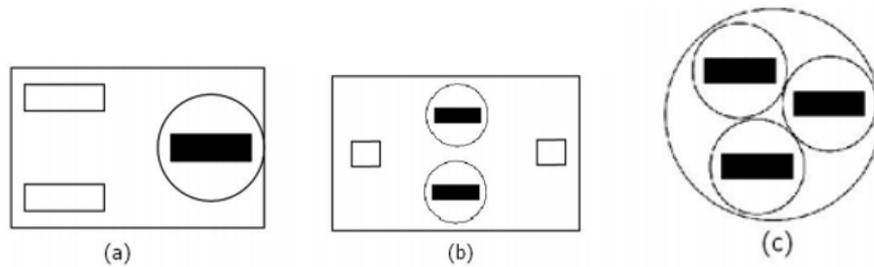
VII-A3. *Ruedas convencionales:* Este tipo de ruedas convencionales demuestran desplazamiento de tipo omnidireccional debido a su posición del eje que le permite que roten respecto de la rueda. Lo podemos visualizar como en la figura 1



(a) Tipo Forward Offset Steered (b) Tipo convencional simple (c) Tipo Lateral Offset Steered

Figura 1. Tipos de ruedas convencionales [22]

Es posible encontrar estas ruedas convencionales con configuraciones en las que se detallan, dos ruedas de tipo automóvil con motores independientes acompañada de una tercera rueda convencional que le permite un adecuado giro, otras de las configuraciones estarían usando dos ruedas convencionales y uno o dos puntos de apoyo o incluso ordenar de manera que tenga 3 ruedas convencionales [34].



(a) Dos ruedas de tipo au- (b) Dos ruedas convencionales (c) Tres ruedas convencionales
tomóvil y una rueda convencio- con dos puntos de apoyo
nal

Figura 2. Configuraciones de ruedas convencionales [13]

VII-B. Ruedas omnidireccionales

Al referirse de ruedas omnidireccionales se mantiene la postura de que este tipo de rueda permitirá que nuestro robot o artefacto pueda moverse hacia diferentes direcciones otorgando vehemencia en la rotación y orientación, tomando de referencia las ruedas convencionales que requieren girar y cambiar de dirección para llegar a un punto deseado. Se debe recalcar estos modelos de ruedas omnidireccionales me permitirán realizar trayectorias que estén dispuestas a desplazamientos y rotaciones para alcanzar el punto de destino [54].

VII-B1. Rueda universal: El tipo de rueda universal al ser girada encima de una superficie demuestra una combinación de movimiento restringido y no restringido durante el giro. Dentro de sus especificaciones se observan pequeños rodillos en los extremos de la rueda puestos de manera perpendicular al eje de rotación de la rueda [38].

VII-B2. Rueda mecanum: La rueda de Ikon también llamada rueda mecanum en honor a su creador Bengt Erland Ikon, dentro de sus detalles se notan el contenido de rodillos que están montados en un ángulo α de $+45^\circ$, en la figura A y B se muestran dos tipos de perspectiva las cuales indican la fuerzas que se genera por la rotación de la rueda al actuar sobre la superficie debido a que el rodillo tiene contacto con el suelo.

Como se puede notar en la figura 3 sus rodillos se descomponen fuerzas paralelas a la manivela del rodillo F_1 y F_2 . La fuerza perpendicular causa rotaciones generando internamente velocidad V_r , pero las fuerzas compuestas F_1 y F_2 generan impulso sobre la rueda por lo tanto el vehículo demuestra como resultando la velocidad central V_h . Por lo tanto, la velocidad V_t (velocidad total) es la unión de V_h y V_r .

Las ruedas mecanum demuestran gran versatilidad permitiendo que el artefacto o robot pueda reubicarse y girar de forma eficiente [34].

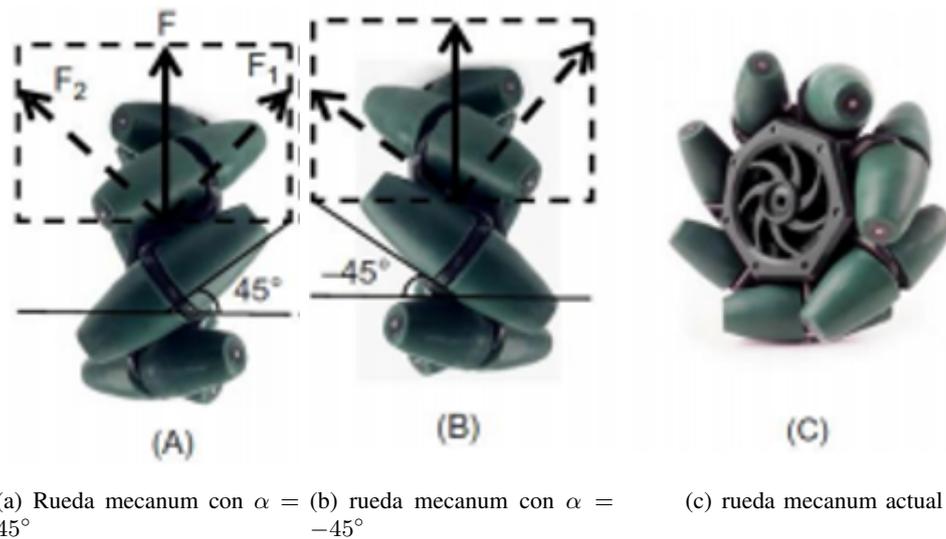


Figura 3. Configuraciones angulares de la rueda mecanum [53]

VII-C. Lenguajes de programación utilizados en la robótica

VII-C1. Lenguaje C++ : C++ es un lenguaje de programación, realizado alrededor de 1980 como una extensión del lenguaje C. Actualmente, el lenguaje C++ es reconocido por su versatilidad. Los éxitos que demuestran los programadores lo han llevado a estar en el primer puesto como instrumento de desarrollo de software en sistemas operativos famosos como Windows y Linux. Generalmente, C++ está escrito de diversos ficheros. A lo largo del proceso de compilación dichos ficheros van a ser combinados correctamente y traducidos a código objeto, facilitando un programa dispuesto a ser ejecutable [10]. El triunfo del C++ en pequeños procesadores se debe a que no es un lenguaje de tan elevado grado como lo es el PASCAL o el ADA, sino un macroensamblador enormemente portable, lo cual posibilita tener la misma flexibilidad que el lenguaje ensamblador ofreciendo casi el mismo grado de eficiencia [42].

VII-C2. Lenguaje Python: Python es un lenguaje de programación que muestra una secuencia de ventajas que lo transforman en una herramienta bastante llamativa, como que es un lenguaje bastante expresivo, un programa en lenguaje Python frecuentemente es más breve a diferencia del idioma C. Demuestra una selectiva sintaxis que posibilita la escritura de programas resultando impecable.

VII-D. Controladores

VII-D1. Raspberry pi: La PBC “Raspberry Pi” es un miniordenador con un tamaño bastante limitado que tiene la misión de actuar como un computador didáctico y accesible a alumnos e individuos que quieren emprender proyectos computacionales y de control electrónico. Tiene comúnmente un procesador de 4 núcleos, conexiones RAM, puertos USB, entradas HDMI, puertos Ethernet e inclusive puertos de tarjetas SD para guardar un sistema operativo [51].

También podría ser equipado con módulos Wi-Fi y Bluetooth para expandir las modalidades de proyectos que se pueden hacer. La placa puede inclusive ejecutar el sistema operativo Linux, por consiguiente, posibilita la codificación en diversos idiomas. Cuando el sistema operativo está instalado, es como laborar en cualquier máquina Linux.

VII-D2. Arduino: El Arduino se ofrece al mercado como una tarjeta de programación open-source manejada mediante un microcontrolador “ ATmega328P ”, este se caracteriza por poseer una cantidad decente de pines de

entrada y salida digitales a pesar de su tamaño. Esta tarjeta tiene una gran accesibilidad, ya que está abierta a ser operada mediante varios entornos de desarrollo, así como posee soporte de varias instituciones relacionadas al mundo de la programación y la electrónica. Parte de su fama consiste en su facilidad para establecer comunicación con varios dispositivos y ser programada desde los mismos. Este embebido posee 14 pines E/S, 6 entradas analógicas, sin embargo posee las mismas cualidades que el arduino UNO [6] .

VII-E. Código QR

Es un código bidimensional diseñado para ser detectado a mayor velocidad, su diseño es muy peculiar debido a que la información esta codificada en pequeños bloques que son detectados por un láser.

Su estructura consta 3 cuadrados. Cada bloque consta de cuadros negros o blancos llamados módulos, que simbolizan el 0 y el 1 en binario[27].

VII-F. Robot omnidireccional

Los robots omnidireccionales son vehículos que poseen la capacidad de moverse en cualquier dirección del plano sin la necesidad de realizar giros.

Para la construcción de un robot omnidireccional es importante tener en cuenta el número de ruedas que se utilizaran. Cada configuración estructural en las ruedas tiene ventajas y desventajas. Los robots omnidireccionales de 3 ruedas cuentan con simplicidad en dirección y control, pero presenta limitaciones en tracción y estabilidad. Los de 4 ruedas, poseen complejidad en su mecánica y controles, pero se benefician de mayor estabilidad y tracción [40].

VII-G. Sensores

El sensor es un dispositivo capaz de detectar movimientos, efectos y otros factores de los que se pretende obtener datos. Convierte un fenómeno físico en una señal analógica medible permitiendo de esta manera transmitir lectura o procesamiento adicional. Son reconocidos como un artefacto que aprovecha efectos físicos para adaptarlos en señales y de esta manera puedan ser interpretados por un dispositivo, como es el caso del termómetro de mercurio [50]. Generalizando el termino, existen dos tipos fundamentales de sensores los cuales son:

VII-G1. Sensores inductivos: Dentro de las especificaciones es que detectan materiales metálicos, la mayor parte de estos sensores son utilizados en la industria para dar el posicionamiento del objeto metálico y de esta manera detectar presencia o ausencia del objeto metálico [43].

VII-G2. Sensores capacitivos: Son sensores que detectan objetos metálicos y no metálicos. Debido a esto, la distancia de conexión será mayor en cuanto más elevada sea la constante dieléctrica. Estos tipos de sensores son frecuentemente utilizados para detectar objetos de toda clase de materiales solidos o líquidos [43].

VII-G3. Sensor de temperatura: Antiguamente existía el método de medir la temperatura por medio de un termómetro, en la actualidad existen diversos métodos y tecnología que permite detectar el efecto de la temperatura que incide en un artefacto. En los varios campos de las ingenierías han perfeccionado e innovado sensores que sostenga las variaciones de temperatura en las aplicaciones industriales [8], estos dispositivos pueden ser:

- Termocuplas
- Termorresistencias

- Termistores
- Diodos Sensores de silicio

VII-G4. Sensores de distancia infrarrojos: Este tipo de sensores funcionan mediante un haz de luz infrarrojo para recopilar la información solicitada por el usuario. Consiste en un emisor que recepta los rayos infrarrojos y de esta manera calculen el tiempo de vuelo de la luz como también la velocidad a la que se desplaza por el aire para procesar la distancia recorrida. La velocidad de la luz no se ve afectada por variaciones de temperatura en el ambiente, además al utilizar luz como principal herramienta y tener una alta frecuencia se vuelven más precisos, pero al mismo tiempo más costosos que los sensores ultrasónicos, sin embargo, la diferencia de costos no es exagerada. Cabe recalcar que la luz solar puede afectar su funcionamiento limitándolo así a interiores [41].

VII-H. Batería

La batería es un artefacto capaz de transformar la energía química en un proceso eléctrico y de esta manera poder alimentar un dispositivo esto gracias a la reacción de redox [31]. Dentro de la composición figuran pequeñas celdas las cuales permiten el almacenamiento de energía, debido a la reacción electroquímica estas se encuentran conectadas por un electrolito que lleva a cabo reacciones para la reducción/oxidación [29].

VII-I. Microorganismos

Los microorganismos son pequeños seres vivos que son imposible de percatar a simple vista, por esta razón se utilizan equipos que los detectan como los microscopios.

Los microorganismos heterogéneos (bacteria, hongos, virus, algas unicelulares) se los considera elementales para el cuidado y diversidad en el planeta. La mayor parte de estos microorganismos se forman de células por lo que se los diferencia entre microorganismos unicelulares y pluricelulares.

La OMS, declara que cerca de 55 hospitales de 14 países un 8,7% de los pacientes demuestran infecciones por causa de un microorganismo [7]. La ciencia ha demostrado que no todos los microorganismos son los causantes de enfermedades infecciosas, de hecho, existen microorganismos de agente patógeno y de agentes benéficos los cuales contribuyen en el ciclo de la materia y al equilibrio de la biosfera, proporcionando soporte en la absorción de dióxido de carbono emitiendo oxígeno, mientras que otros descomponen materia orgánica.

VII-J. Tipos de microorganismos

De acuerdo con el proceso celular de los microorganismos se conoce que existen 4 tipos de microorganismos que depende en la manera como se estructura, si es unicelular o forma colonias dando una semejanza a que fuera pluricelular [45].

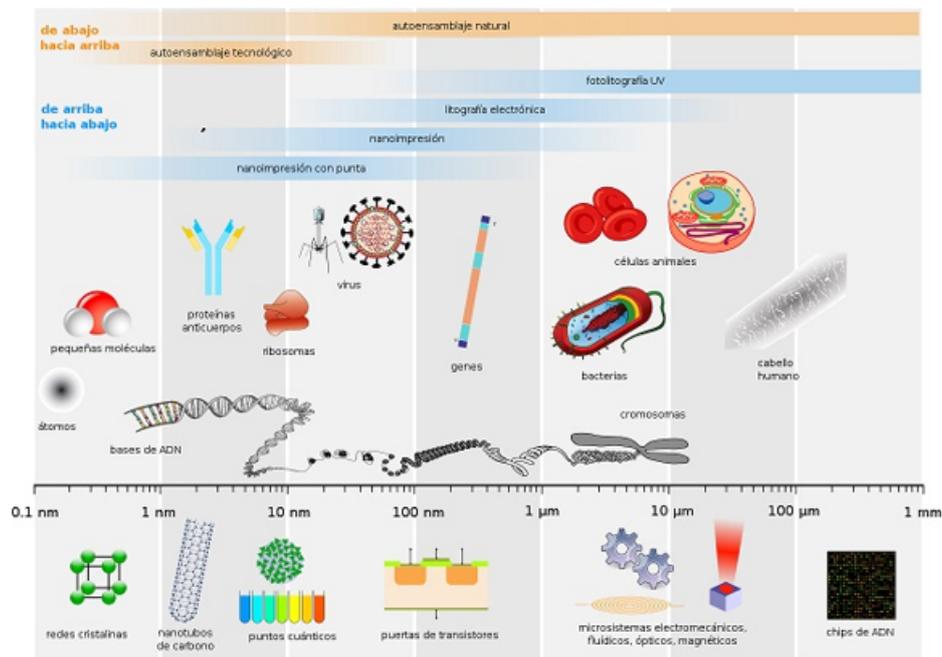


Figura 4. Tipos de microorganismos presentes en el medio y sus dimensiones [5]

VII-J1. *Bacterias*: Entre los tipos de microorganismos encontramos la bacteria que tiene la particularidad de ser un organismo procarionta unicelular, es frecuente encontrar por varias partes en la tierra. Estas bacterias son capaces de soportar altas y bajas temperatura como también presiones.

[44] estima que el cuerpo humano se encuentra expuesto a múltiples bacterias que se mantienen en las superficies, pero gracias a nuestro sistema inmunológico provocó que el efecto de la bacteria actúe nocivamente ante nosotros. Algunas de estas bacterias patógenas llegan a causar enfermedades infecciosas originando fallas hepáticas, respiratorias o estomacales en el sistema del cuerpo humano.

VII-J2. *Algas*: El alga es un microorganismo debido a que en su estructura está compuesta de microbios que se alimentan de la luz solar. El papel primordial de este tipo de microorganismo es la producir oxígeno y brindar soporte en disminuir la influencia del cambio climático.

La capacidad que tienen estas algas está en el beneficio que ofrece al consumo, en la industrial y en el beneficio del ser humano.

Se estima que anualmente 40.000 toneladas de algas se extraen para el consumo humano para la preparación del sushi en Japón. Estas algas además de ser utilizadas en el consumo están siendo parte de la elaboración de fármacos como vitaminas para el cuidado de las personas [12].

VII-J3. *Protozoos*: Son células eucariotas simples, tienen la característica de ser móviles y heterótrofos y se estima que fueron los primeros organismos celulares que existieron, son reconocidas por ser pequeñas y resistentes a las condiciones medioambientales adversas.

En, 2018 Alvarez,A [3] estimó que existen 50.000 variedades de protozoos.Existen varios casos de especies que son catalogadas de vida libre, mientras que otras se comportan como parásitos alojándose en el hombre o los animales.

Normalmente es habitual encontrar estos tipos de microorganismos en el agua, los cuales contribuyen en preservar el medio ambiente. A excepción de los protozoos parásitos cuya distribución está asociada en el hospedaje de un individuo agravando el sistema inmunológico [9].

VII-J4. Hongos: Los hongos son organismos vivos de la especie eucariota que poseen una pared celular conformada por quitina, diferenciándose de las plantas que por otro lado poseen celulosa. Poseen la característica de generar metabolitos secundarios los cuales actúan como antibióticos beneficiando en la protección de los hongos de otros microorganismos.

En la Universidad de Nacional de Salta Carrillo ha demostrado que los hongos se vuelven tóxicos cuando existen la generación de metabolitos que al fermentarse producen que el animal que consuma este vegetal demostraría las micotoxinas dando una reacción de envenenamiento [17].

VII-J5. Virus: Los virus forman parte de los tipos de microorganismos. Dentro de sus características es que son submicroscópicos, lo que significa que no son visibles por medio de un microscopio.

En su estructura se compone internamente de un ácido nucleico, llamado ADN o ARN, los cuales pueden ser de cadena simple o cadena doble, dentro de una vejiga de proteínas.

Los virus no tienden a multiplicarse por sí mismos deben adquirir células para poder incrementarse, se puede encontrar en todas partes del ecosistema terrestre [25] .

Se detalla que tanto los virus y bacterias nocivas perjudican al paciente que adquiera una de ellas, pero existe una diferencia entre ellas y está en que normalmente las bacterias se las puede combatir con antibióticos aun así sea un cuadro clínico un poco complicado, en cambio el virus es algo que no se puede eliminar, en ocasiones se propone trata de reducir la severidad del virus permitiendo que el sujeto pueda subsistir con virus sin que le cause complicaciones en el sistema inmunológico [11] .

VII-K. Microbiología clínica

En la antigüedad existieron muchas infecciones a causa de que no existía los principios metodológicos con la ayuda de herramientas que permitan demostrar el tipo de infección y de esta manera ayudar en la disminución de estos contagios. Las infecciones relacionadas con la atención médica provocan que varios pacientes aumenten su estancia en centros de salud. Este hecho se muestra como un problema en los pacientes volviéndolos vulnerables a las infecciones [48] .

La microbiología clínica estudia el análisis y diagnóstico médico a un paciente mediante distintas técnicas de identificación de los microorganismos presentes en la persona que agravan su cuadro clínico a causa de las infecciones que ocurren en nuestro organismo. El médico responsable tiene la facultad de optimización en la asistencia del paciente. Es esencial comunicar lo antes posible, al médico responsable que otorgue un excelente manejo y tratamiento, dando uso de los recursos disponibles [37] .

VII-L. Microorganismos presentes en los ambientes médicos

Es habitual escuchar que en las instituciones médicas los pacientes que ingresan a cuidados intensivos a causa de una molestia o de que su cuadro clínico es complicado, debe pasar por una rigurosa recuperación que permita demostrar un mejor estado de salud, superar y aliviar el tipo de molestia con el que ingreso además del tipo de infección presente en estos ambientes médicos [48] .

Los agentes patógenos viajan en el aire a causa de un algún huésped infectado normalmente el contagio se dan por las vías respiratorias, se ha demostrado que estos agentes patógenos viajan a través del aire debido a superficies contaminadas o incluso de pacientes enfermos.

De acuerdo a [15] Ecuador presenta una tasa de mortalidad por infecciones bacterianas y de neumonía sumamente elevadas, se estiman que aproximadamente en 1000 hospitales fallecen por situaciones como estas, a diferencia de otro países como Brasil o Perú, pero la incidencia en la contaminación es semejante.

En la figura 5 se define los tipos de agentes patógenos nosocomiales que se encuentran habitualmente en las zonas médicas y como demuestra la capacidad de resistir en las áreas médicas, debido a los minerales que tiene las superficies, por los ambientes húmedos, de temperatura baja e incluso por el mal manejo de herramientas de desinfección.

Tipos de microorganismo	Duración en el tiempo (Rango)
Bacterias Gram negativas	
<i>Acinetobacter spp.</i>	3 días – 5 meses
<i>Pseudomonas aeruginosa</i>	6 h – 16 meses
<i>Escherichia coli</i>	1,5 h – 16 meses
<i>Klebsiella spp.</i>	2 h - >30 meses
<i>Serratia marcescens</i>	3 días – 2 meses
Bacterias Gram positivas	
<i>Clostridium difficile</i>	5 meses
<i>Enterococcus spp.</i>	5 días – 4 meses
<i>Staphylococcus aureus</i>	7 días – 7 meses
Levaduras	
<i>Candida albicans</i>	1 día – 4 meses
<i>Candida parapsilosis</i>	14 días
Virus	
<i>Rotavirus</i>	2 meses
<i>Norovirus</i>	42 días

Figura 5. Persistencia de patógenos nosocomiales en superficies inanimadas [1]

Las instituciones médicas habitualmente deben someterse a un estudio microbiológico del ambiente en las diferentes áreas de los hospitales es importante conservar un control a la ausencia de agentes etiológicos en las superficies y de esta manera disminuir los índices de contaminación [15].

Ecuador, no es habitual informar la estadística sobre los casos de muertes debido a la contaminación hospitalaria. La presencia de microorganismos como *Serratia marcescens*, *Pseudomonas aeruginosa*, *staphylococcus coagulosa*, *Escherichia coli* y *Klebsiella pneumoniae* los cuales perturban el sistema inmunológico del paciente [7] .

VII-M. Análisis microbiológico

El análisis microbiológico es el método que se implementa para identificar el tipo de microorganismo que se encuentra tanto en la superficie, aire, agua para la detección es habitual el uso de compuestos biológicos, bioquímicos, moleculares o químicos.

El análisis microbiológico asiste en mantener un bajo control de la proliferación de bacterias, virus y microorganismos que son los influyentes en el desarrollo de enfermedades infecciosas y del deterioro de alimentos [55] .

En el análisis microbiológico es crucial tomar en cuenta los pasos:

- Muestreo. - Son las muestras que se realizan en la superficie que se desea analizar en base a esto se obtiene los resultados.
- Filtración.- En esta operación de filtración se maneja con la finalidad de mejorar la recuperación de microorganismos y de esta manera evitar contaminantes externos que afectarían a la muestra.
- Cultivo.- Se única la muestra y el medio de cultivo ya que exista flotamiento del microorganismo.
- Incubación.- En esta etapa final se ubica el medio de cultivo encapsulado por una caja Petri donde pasa por un periodo de incubación.

VII-N. Análisis microbiológico del suelo

Los análisis microbiológicos del suelo estudian la abundancia, el desarrollo y se representan como un método de identificación de los microorganismos del suelo, además de las actividades microbianas que favorecen al desarrollo de los cultivos. Para recurrir a esta técnica es necesario el uso de implementos de laboratorios que demuestren fielmente las condiciones de cómo se presentan los microorganismos en el medio [18] .

VII-Ñ. Análisis microbiológico del aire

El análisis microbiológico del aire es un método de estudio que identifica los microorganismos presentes en el aire. Este análisis consiste en ubicar un medio de cultivo estéril, dejándolo descubierto durante un determinado periodo, con la finalidad de que los microorganismos presentes en el aire se sedimenten ayudando a tomar propia identificación y conteo de los microorganismos los cuales afectan la persona que está expuesta a ellos [28].

VII-O. Técnicas de análisis microbiano

Son técnicas que se utilizan para evaluar el incremento o decremento microbiano.

VII-O1. Recuento de células totales:

- Método directo

Se realiza mediante el uso de la cámara de Petroff-Hausser, este artefacto permite contar las células microbianas usando como herramienta la visualización microscópica. Para este método, requiere de un portaobjetos que tienen una graduación en su superficie que permite retener el volumen del inóculo [35] .

En la figura 6 se demuestra la cámara de Petroff-Hausser donde se detalla el número y la identificación de células microbianas.

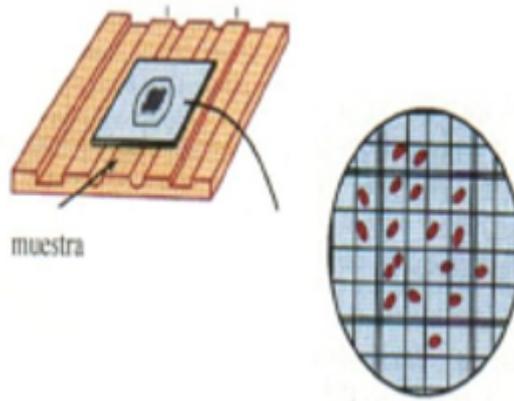


Figura 6. Cámara de recuento [35]

- Método turbidimétrico

Este método se realiza ubicando las células microbianas dentro de una cápsula, la cual debe permitir que la luz incida sobre ellas dando como resultado poder contar e identificar los microorganismos. Además, para obtener los valores de absorbencia se utiliza el espectrofotómetro donde los resultados se muestran en unidades de densidad óptica.



Figura 7. Espectrofotómetro [35]

VII-O2. *Recuento de células viables :*

- Recuento en medio solido

Radica en demostrar los cultivos que crece dentro del agar tomando en cuenta que cada célula desarrolla una colonia. Se conoce que no siempre el cultivo es formado por una sola célula [49] .

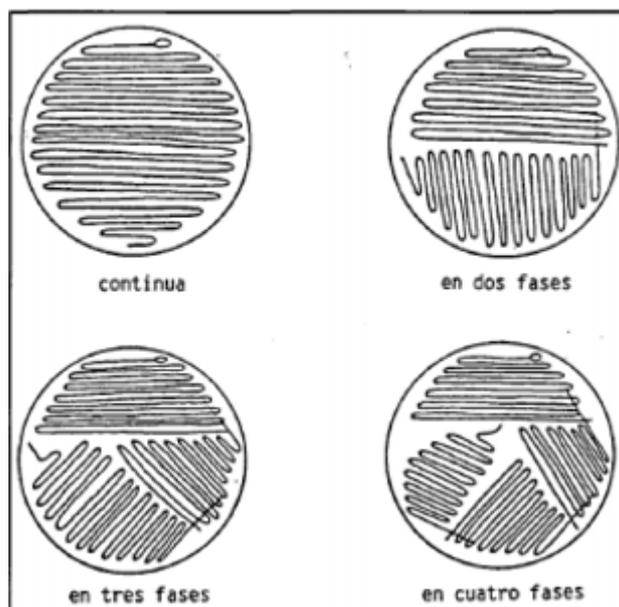


Figura 8. Formas de raspado sobre el medio [49]

- Recuento en medio líquido

Esta técnica nos ayuda a apreciar los microorganismos que se muestran en un medio líquido, alimentos o en su defecto en el suelo. Esta técnica se basa en las diluciones las cuales dependerán de la proporción de microorganismo encontrados en la muestra [18] .

VII-P. Desinfección

Para hablar de desinfección se debe primero conocer que no es lo mismo limpiar que desinfectar, pues para una limpieza será suficiente en algunos casos agua, jabón y detergente, material con el cual se logra eliminar cualquier clase de suciedad presente en objetos o superficies. [32] estima que la desinfección requiere de técnicas con aplicación de químicos para eliminar microorganismos patógenos adheridos a las superficies. La desinfección siempre será realizada después de que la superficie haya sido limpiada.

Los métodos empleados para impedir el acceso de microorganismos a un sustrato estéril es lo que se considera como asepsia, así pues, la asepsia es lo que se podría considerar como la desinfección, que para junio de 1967 llegaba por primera vez a un quirófano mejorando las condiciones sanitarias, debido a la reducción de contaminantes como virus, bacterias, hongos los cuales causaban infecciones durante las intervenciones quirúrgicas [46] .

En la actualidad la esterilización de los implementos, así como las superficies médicas se ha venido realizando con distintas técnicas relacionadas con el uso de vapor, calor seco, rayos UV y gas.

VII-Q. Herramientas para la desinfección

Dentro de las afectaciones que surgen en los centros de salud se puede notar el suministro de líquidos descontaminantes que se colocan en las superficies de los hospitales generando malestares al personal médico y pacientes.

El personal de enfermería que está expuesto a riesgos biológicos, fisicoquímicos, ergonómicos y psicosociales depende mucho de que su ambiente se encuentre aséptico, además los tipos de químicos que se utilizan deben

volatizarse rápidamente para permitir el ingreso del personal, a cambio de no exponerse a químicos que pueden perjudicar en su salud [4] .

VII-R. La radiación ultravioleta

La radiación ultravioleta es una emisión de luz emitida por el sol y fuentes artificiales, este tipo de radiación está por debajo de emisión térmica (con temperatura de 600°C), por esta razón no emite calor debido a la incandescencia [33] .

En la figura 9 se demuestra la región visible del espectro electromagnético que genera la luz UV, los rayos infrarrojos y la luz visible.

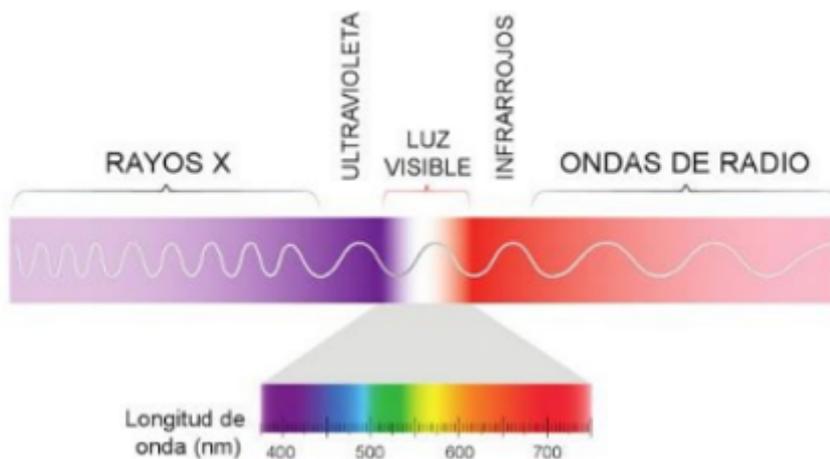


Figura 9. Representación del espectro electromagnético. Zona de visibilidad 400 nm- 700 nm [33]

El fenómeno de la luminiscencia se desarrolla en función a una fuente externa de energía que es excitada la cual absorbe el material para transformarlo en luz.

Los fenómenos de luminiscencia depende de factores que favorecen su reacción, por ejemplo, tenemos la fotoluminiscencia que se produce por la excitación del material con los fotones de radiación UV, el efecto electro lumínico es proviene de las excitaciones generadas por energía térmica, la cátodo luminiscencia es generada la excitación de los electrones y la triboluminiscencia es generada por la energía mecánica [21] .

VII-S. Uso de la radiación ultravioleta para inactivar los microorganismos

Es habitual que en personas se den infecciones a causa de microorganismos, pero en la actualidad la tecnología a demostrar dar soporte en la bioseguridad de las personas.

Las técnicas de bioseguridad que se han desarrollado implican en el uso de la radiación ultravioleta, es una onda de luz corta de 250 + 10 nanómetros (nm), es letal para bacterias, virus, protozoos, hongos, levaduras y algas. El efecto que se se manifiesta en el microorganismo, implica hacer daño a su estructura de ADN de manera que la luz germicida se encuentre a una distancia óptima garantizando la inactivación microbiana.

Las técnicas de desinfección ultravioleta que se implementan en la actualidad para la eliminación de microorganismos, tenemos los tratamientos de aguas residuales, por medio de la cual busca mejorar la capacidad del proceso

híbrido con la finalidad de eliminar los diferentes tipos de patógenos.

De acuerdo a los estudios de [19], se manifiesta que entre el 96-99 % de los microorganismos fueron eliminados por la exposición de radiación en el tratamiento de aguas residuales.

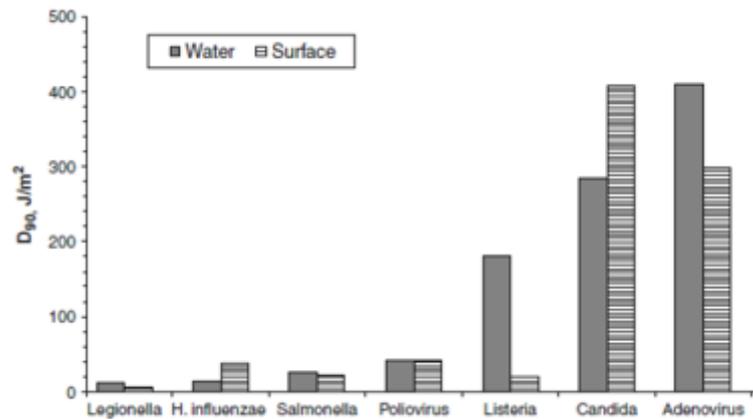


Figura 10. Susceptibilidad de diferentes tipos de microorganismos [19]

En la figura 10 se puede notar la dosis de UV en J/m² requerida por diferentes microorganismos tanto en agua como en superficies.

En efecto se han dado buenos resultados en el uso de tecnología UV-C, en cuanto a factores de desinfección. Pero dentro de las inquietudes está el factor de tiempo, distancia y la manera de como el microorganismo muestra una reducción en la concentración de estos microbios. En un estudio sobre la reducción microbiana de la escherichia coli se demostró que durante un intervalo de 15 minutos y una longitud de 5 centímetros donde se observó la reducción microbiana, se manifestó una diferencia importante en la reducción microbiana.[39] .

VIII. MARCO METODOLÓGICO

En este capítulo se describe el procedimiento mediante el cual se configuró y diseñó el robot de desinfección, así como también se muestran los planos y diagramas fundamentales para llevar a cabo el funcionamiento del hardware del proyecto.

VIII-A. Diseño de la estructura del robot

El robot está diseñado de forma que su funcionamiento no interrumpa y tampoco lastime a ninguna persona que se encuentre dentro del perímetro por donde el robot está realizando la labor de desinfección de forma automática. El robot posee una estructura tipo hexagonal para aminorar el riesgo de accidentes por cortes.

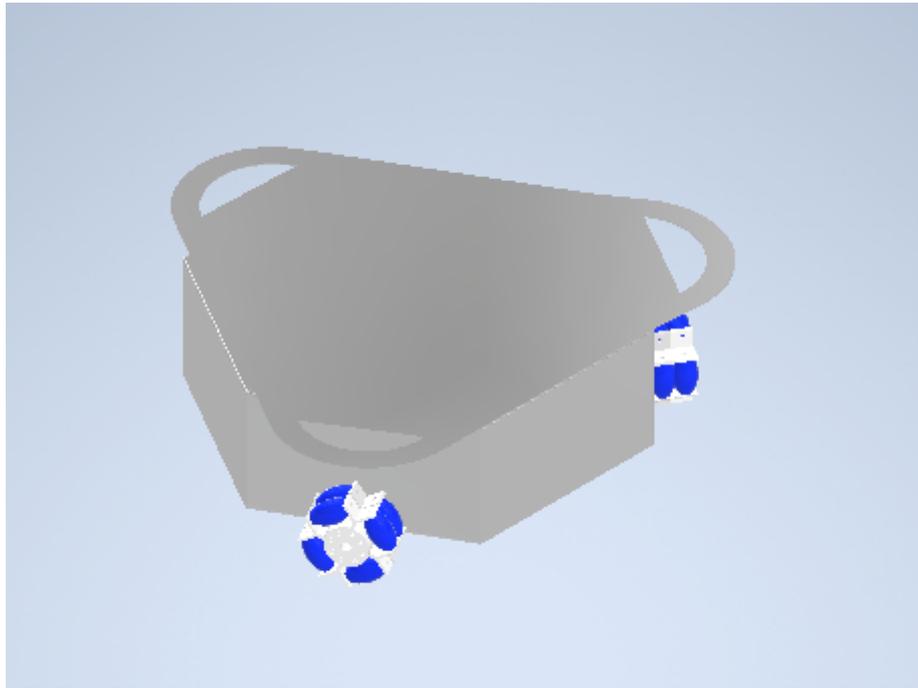


Figura 11. Ensamble de robot omnidireccional realizado en Inventor

VIII-A1. Materiales de la estructura: Para poder construir el robot se utilizaron láminas de aluminio de 1 mm de grosor, teniendo en cuenta las propiedades reflectivas que posee el aluminio y como podría beneficiar dicho efecto en el funcionamiento interno de las luces UV. También se tomó en cuenta la resistencia al impacto mínimo necesario y las características generales de disipación de calor, con las cuales se mostraba idóneo para realizar su labor. En la figura 66, ubicada en el apéndice C, se muestra la plancha de aluminio utilizada para la estructura metálica.

Teniendo en cuenta que ciertas piezas en la estructura interna del robot necesitaban la implementación de diseños que se puedan adaptar para cumplir sus necesidades, se optó por elaborar estas piezas auxiliares mediante plástico PLA, el cual posee la resistencia necesaria para soportar las cargas existentes en su superficie. Dentro del apéndice C, se muestra el proceso de impresión de las piezas antes mencionadas (figura 65). Por motivos de aislamiento eléctrico, se utilizó una lámina de acrílico para fijar en su superficie las tarjetas de control correspondientes al apartado electrónico del sistema, así se evitará que dichos componentes hagan contacto con la superficie metálica que cubre el robot. En el apéndice A, se puede observar la pieza de acrílico utilizada para el aislamiento de los componentes (figura 54).

Los investigadores Mercy González y Hernán Davilla detallan que el aluminio es un material que restringe todo tipo de choque electromagnético que son producidos por máquinas de resonancia magnética o tomográficas.[39]

VIII-A2. Diseño de tapa: Al momento de diseñar la tapa se tomó en consideración el espacio utilizado por el chasis, las ruedas omnidireccionales y las luces UV, también componentes externos de la carcasa como el espacio utilizado por la cámara. El diseño de la tapa fue realizado mediante el software inventor, el resultado se aprecia en la figura 12. Las dimensiones de esta pieza y el acabado real de la estructura se muestran en los apéndices D y A respectivamente.

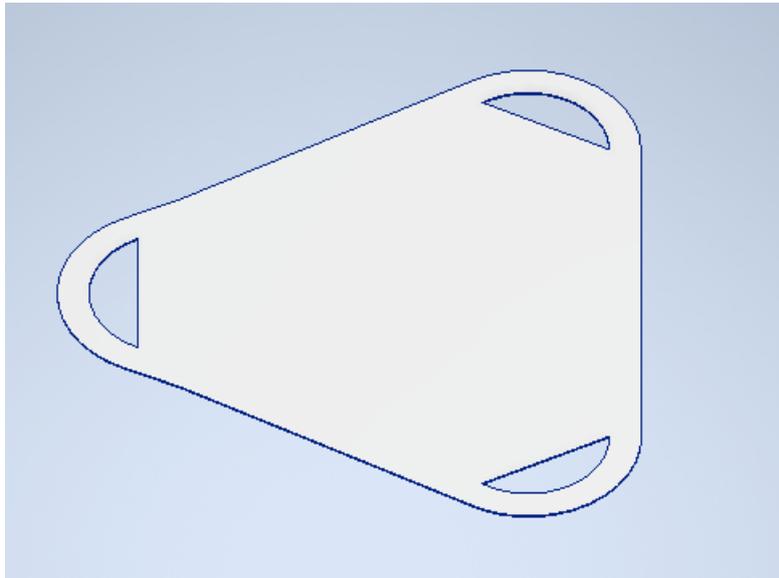


Figura 12. Diseño de la carcasa del robot

VIII-A3. Diseño de chasis: Para diseñar el chasis se analizó la cantidad de elementos que llevarían a cabo su funcionamiento dentro del mismo, en base a esto se escogieron las medidas de la estructura y los tamaños de las perforaciones necesarias para ubicar los componentes de forma que se evite situaciones de daño entre ellos como se muestra en la figura 13. Las medidas necesarias para realizar esta pieza se muestran en el apéndice D, mientras que la estructura física se encuentra ubicada en el apéndice A.

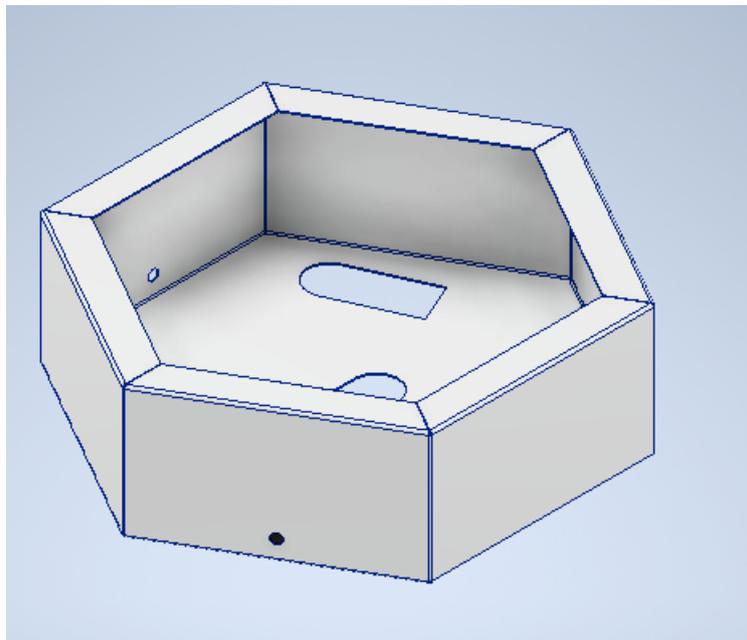


Figura 13. Diseño 3D de chasis

VIII-A4. Diseño de Acople de motor-rueda: Para obtener las dimensiones necesarias del acople fue imprescindible el uso de un calibrador vernier y medir el eje de los motores, de este modo se obtuvo una mayor exactitud para desarrollar el diseño en el software inventor. Con el fin de obtener la pieza se manufactura mediante una impresora 3D y material PLA un acabado del 90 % de relleno para adquirir resistencia y evitar rupturas en la unión motor-rueda durante el movimiento. Como se observa en la figura 14, el diseño de la pieza posee una perforación en medialuna para el acople del eje del motor, también consta de un prisionero hexagonal para sujetar el eje y proporcionar seguridad en el apriete. Los planos acotados de esta pieza se encuentran adjuntos en el apéndice D.

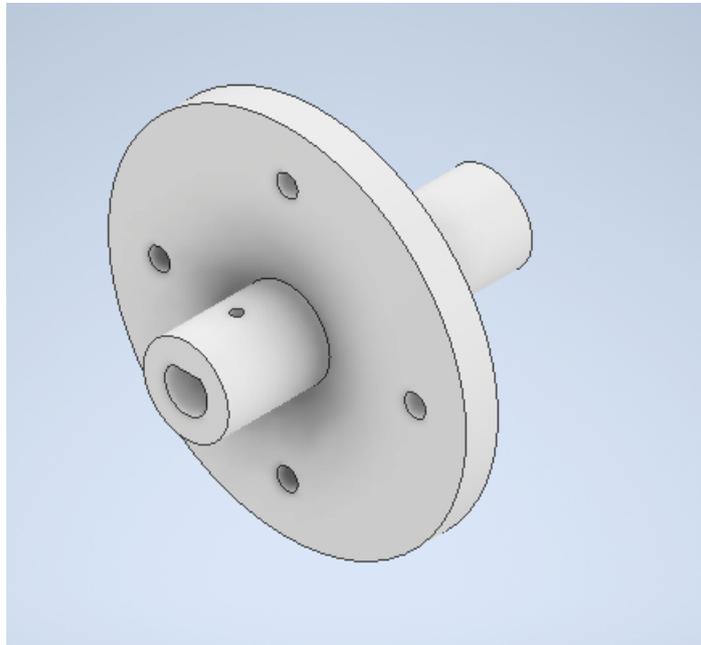


Figura 14. Diseño 3D de Acople del motor-rueda

VIII-B. Selección de ruedas

Se implementaron ruedas tipo omnidireccional para obtener el movimiento característico deseado del robot. En la figura 15, se puede observar el diseño de la estructura de la rueda, la cual consta de rodillos de goma para amortiguar el impacto de la superficie durante el movimiento, este conjunto de rodillos poseen una ubicación perpendicular respecto a la circunferencia de la rueda para así obtener deslizamiento con una fricción mínima, provocando así lo que conocemos como movimiento omnidireccional tal como se observa en los planos del apéndice D, en la figura 87. El resto de la estructura de la rueda fue manufacturada utilizando plástico ABS. Para seleccionar la rueda fue necesario analizar la resistencia a los impactos que posee cada material disponible, en este caso, una de las características más importantes del ABS es su dureza y resistencia a impactos y por esa razón se escogió esta rueda en específico. Por otro lado posee un buen acabado superficial que aporta una buena presencia y brillo, este modelo es bastante comercial y fácil de obtener.

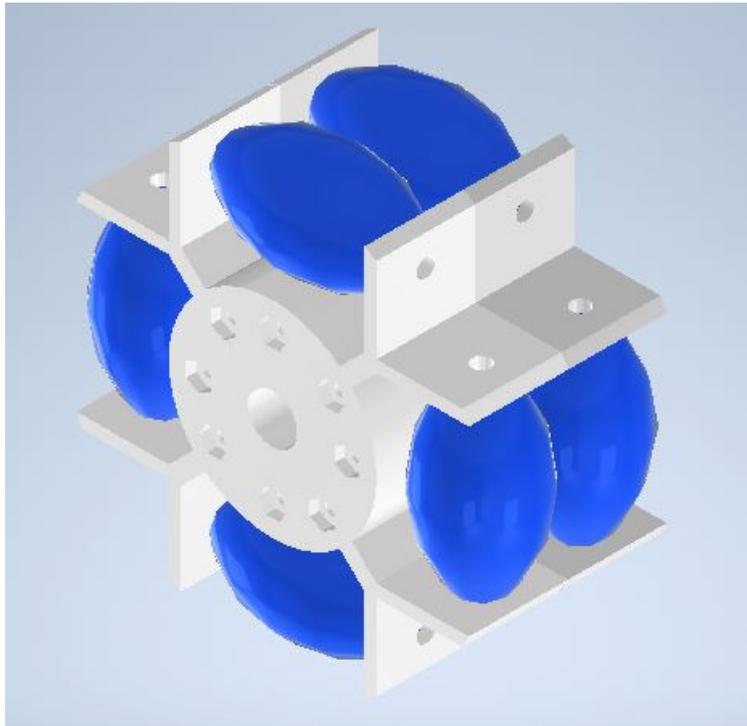


Figura 15. Diseño 3D de rueda tipo omnidireccional

VIII-C. Hardware electrónico

En esta sección se muestra el desarrollo de la parte electrónica necesaria para el correcto funcionamiento del sistema. Se detallan los dispositivos necesarios para la transmisión de datos, el movimiento de los motores, diseño de tarjetas electrónicas y diagramas de conexión. Las dimensiones de los dispositivos electrónicos utilizados en el robot se establecen en el apéndice D. En la figura 16 se muestran de manera ilustrativa las conexiones principales del hardware para la obtención del funcionamiento del robot.

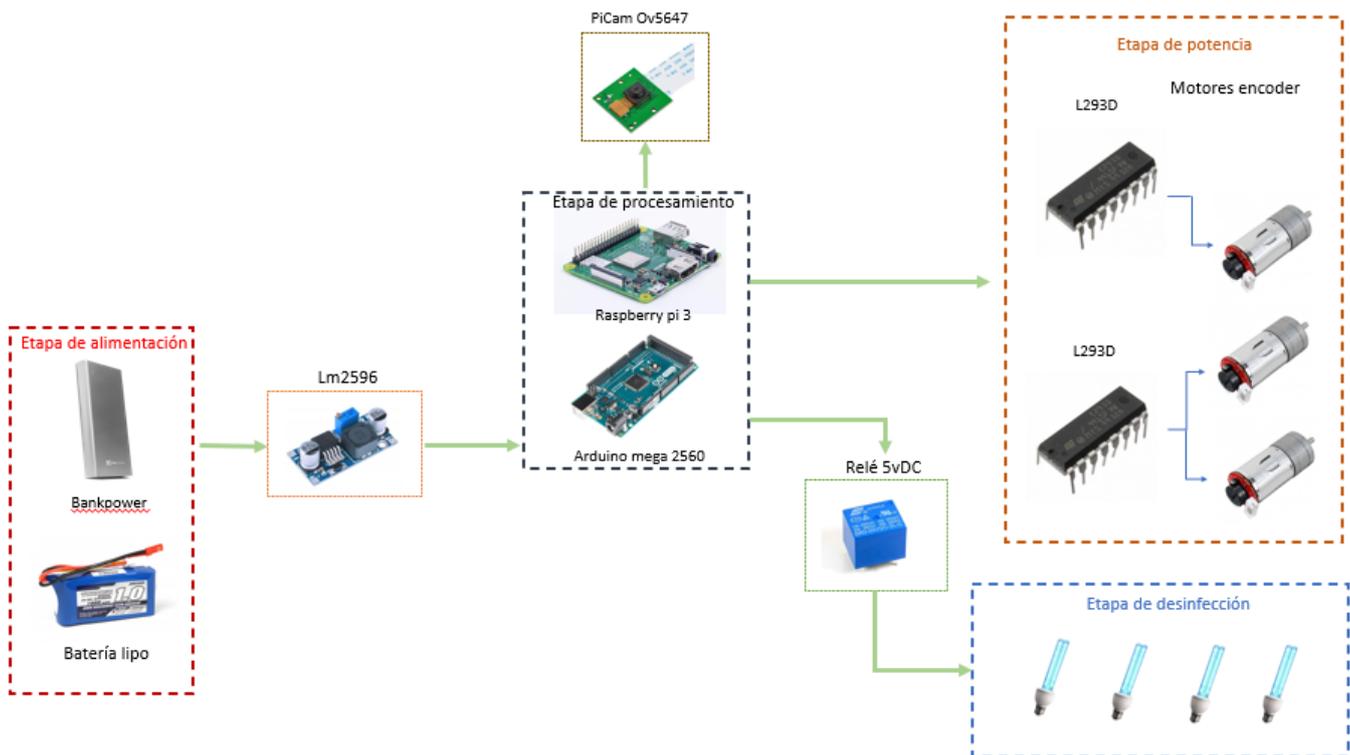


Figura 16. Diagrama de funcionamiento electrónico

VIII-C1. Etapa de alimentación: La etapa de alimentación decide la autonomía del robot y las especificaciones necesarias en cuanto a la energía requerida de los demás elementos que intervienen en el funcionamiento del sistema, por lo cual es relevante conocer el tipo de alimentación que se utilizara para el suministro de energía. En este caso la alimentación estará compuesta por una batería li-on, una batería lipo y un bankpower. Las dos baterías se encargan de abastecer de energía a los módulos, sensores, y dispositivos de fuerza que conforman el sistema, mientras que la bankpower tiene como objetivo alimentar a la etapa de procesamiento.

La bankpower pertenece a la marca “VIDVIES” y posee las características adecuadas para proteger y suministrar de una forma eficiente las cargas respectivas, sin embargo las baterías de li-on y lipo no poseen ninguna clase de protección integrada, por lo cual es necesario adaptar un sistema de protección y regulación de voltaje al circuito. Para este caso se utilizaron dos reductores de voltaje lm2596, estos dispositivos tienen la función de entregar un voltaje regulado al circuito electrónico de manera configurable a los drivers l293d, las configuraciones de conexión para este driver se muestran en el cuadro 59. Según el cuadro de características del X del apéndice B, el lm2596 posee de forma integrada protección de sobre-temperatura, protección de cortocircuito y también una protección limitadora de corriente, teniendo en cuenta dichas facultades, estos elementos representan una protección eléctrica para el robot y al mismo tiempo suministro de energía para sus componentes. En la figura 17, se muestra el diagrama de fuerza del robot, el cual consiste en separar las alimentaciones y protecciones del sistema dividiéndolas en las tres fuentes de energía correspondientes e integrando un fusible de protección para cada salida de voltaje, en el cuadro XIII se determina el significado de las abreviaturas correspondientes al diagrama.

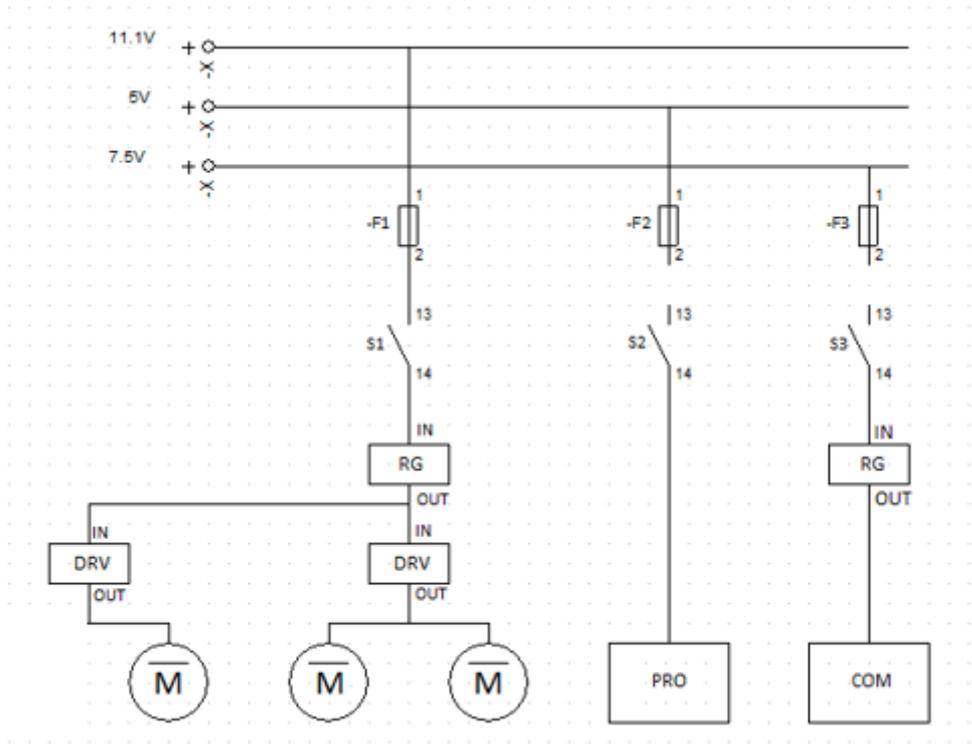


Figura 17. Diagrama de fuerza realizado en el software CADesimu

Como se indica en la figura 17, por cada canal de alimentación del sistema existe un fusible de protección para evitar de forma individual daños en el circuito. Para poder realizar la selección de cada fusible, se tomaron en cuenta las cargas máximas de cada componente dentro del sistema, el análisis se seccionó de la siguiente forma:

Cuadro I
ELEMENTOS CORRESPONDIENTES A LA ALIMENTACIÓN DE 5V

Cantidad	Nombre	Valor de corriente (A)
1	Raspberry pi 3A+	2.5A
	Total	2.5A

En el cuadro III para obtener el valor de corriente de funcionamiento en las luces UV, se procedió a realizar un cálculo en base a las características de potencia y voltaje de su placa controladora, con lo cual se obtuvo lo siguiente:

$$A = \frac{P}{V} \quad (1)$$

$$I_{l\u00e1mpara} = \frac{3,2w}{5v} = 0,5A$$

Al tratarse de 3 lámparas se debe de multiplicar el valor cuatro veces, por lo tanto el resultado total sería:

Cuadro II
ELEMENTOS CORRESPONDIENTES A LA ALIMENTACIÓN DE 11.1V

Cantidad	Nombre	Valor de corriente (A)
3	Motores encoder CHR-GM25-370	3.6A
1	Sharp GP2Y0A21YK0F	0.03A
1	Arduino nano	0.8A
	Total	4.43A

Cuadro III
ELEMENTOS CORRESPONDIENTES A LA ALIMENTACIÓN DE 7.5V

Cantidad	Nombre	Valor de corriente (A)
4	Luces UV	2A
	Total	2A

$$I_{\text{lampara}} = \frac{0,5A}{4} = 2A$$

Para obtener el valor de corriente correspondiente al Arduino Mega que se explica en el cuadro III, se debe de considerar que la corriente máxima que implica la conexión a cada pin de control de este modelo de Arduino es de 40 mA, por lo tanto:

$$I_{\text{Arduino}} = 40mA * 20 = 0,8A$$

Teniendo en cuenta los valores de corriente de cada una de las alimentaciones correspondientes en los cuadros I II III, en base a su corriente máxima de funcionamiento, se implementaron los siguientes fusibles:

Cuadro IV
SELECCIÓN DE FUSIBLES

Fuente de alimentación	Fusible
11.1V	5A/250V
7.5V	3A/250V
5V	3A/250V

Para poder conocer el voltaje de salida del regulador, se debe utilizar la fórmula de la ecuación 2, y tener en

cuenta que el de entrada de voltaje debe ser 1.2V mayor al voltaje estimado. Ahora, asumiendo que la resistencia de referencia (R1) es de 1k (usualmente se establece este valor para esta resistencia), podemos obtener el valor de R2:

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1}\right) + (i_{ADJ} * R2) \quad (2)$$

Despejamos la fórmula con el fin de obtener R2 y asumimos que i_{ADJ} tiende a 0, se obtiene:

$$R2_{5V} = \left(\frac{R1}{V_{ref}}\right) * (V_{out} - V_{ref}) \quad (3)$$

$$R2_{5V} = \left(\frac{1k\Omega}{1,2V}\right) * (5V - 1,2V) \quad (4)$$

$$R2_{5V} = 3,1k\Omega \quad (5)$$

Ajustándolo a una resistencia que tenga un valor comercial se obtiene como resultado:

$$R2 = 3k\Omega \quad (6)$$

Se aplica la misma lógica para obtener los 10V de salida en el otro regulador:

$$R2_{10V} = \left(\frac{1k\Omega}{1,2V}\right) * (10V - 1,2V) \quad (7)$$

$$R2_{10V} = 7,3k\Omega \quad (8)$$

De la misma forma, ajustándolo a una resistencia que tenga un valor comercial obtenemos como resultado:

$$R2 = 7k\Omega \quad (9)$$

Se debe de tener en cuenta que la entrada de voltaje puede variar de 4.5V a 40V y la salida se mostrará constante, sin embargo el ajuste del voltaje de salida se debe de realizar manualmente mediante el potenciómetro del dispositivo (lm2596).

VIII-C2. Etapa de procesamiento: En este proyecto se utilizaron microcontroladores que procesen la información de una forma rápida y eficiente, con el objetivo de que el robot pueda reaccionar ante los distintos escenarios a los que se enfrente. Por lo tanto, se utilizarán los sistemas embebidos Arduino Nano y Raspberry pi 3A+ como encargados de procesar la información y tomar las decisiones tanto del robot, como de los distintos actuadores que este posee. El Arduino tiene la responsabilidad de emitir órdenes a la mayoría de los actuadores como lo son los motores y leer el dato proporcionado por los sensores, mientras que la Raspberry pi procesa la información del video mediante la cámara OV5647, durante el funcionamiento del robot y así mismo decodificar los códigos QR.

VIII-C3. Etapa sensorial: La información que recolectan y emiten estos elementos, hacia el Arduino, es fundamental para que el microcontrolador tome la decisión de accionar un bloque de programación que interrumpa el funcionamiento de las luces UV como método de protección. El proceso de calibración de los sensores se muestra en la figura 18.

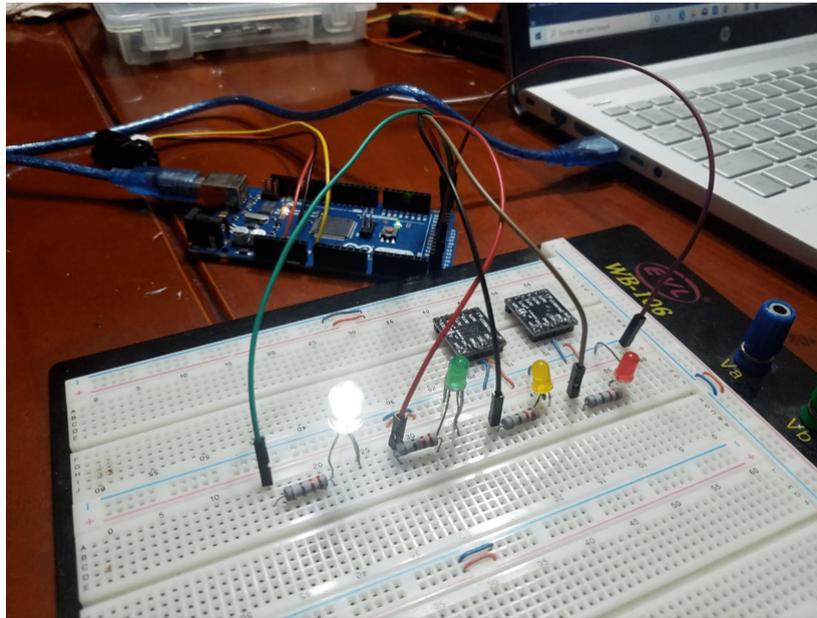


Figura 18. Selección de rango de distancia del sensor IR

VIII-C4. Etapa de potencia: El movimiento del robot se consiguió debido a las propiedades de las ruedas omnidireccionales, sin embargo, los dispositivos que generan el avance por medio de los motores encoder CHR-GM25-370, sus características se encuentran especificadas en el cuadro III. Estos motores poseen las cualidades necesarias para movilizar el robot, aún tomando en cuenta el peso total del robot que es mencionado en la ecuación 10. Adicionalmente poseen una tarjeta encoder con alimentación independiente, que facilita el control de velocidad y el planteamiento de control PID en el sistema. Debido a que estos son los componentes más exigentes de todo el autómatas, poseen una alimentación independiente y su encendido es controlado mediante un driver I293D, el cual posee las características detalladas en el cuadro XIV, se observan sus valores de corriente máxima y control de picos necesarios para contener los cambios de corriente de los motores.

En esta etapa también se incluye el funcionamiento del ventilador que mantiene regulada la temperatura de la parte interna de la estructura, este elemento se encuentra conectado directamente a la fuente de alimentación para poder funcionar inmediatamente cada que el circuito se energice.

VIII-C5. Etapa de desinfección: Esta etapa consiste en la utilización de luces UV para poder eliminar los microorganismos nocivos dentro de las áreas médicas. El control es realizado mediante un relé 5VDC para activar el paso a la alimentación de las luces, debido a que estas poseen un circuito de alta tensión que genera un arco de corriente para provocar el efecto de luz ultravioleta. Las características de las luces se encuentran en el cuadro V. Las luces cumplen con la cantidad de nanómetros necesarios para considerarse luces UV-C.

VIII-C6. Tarjeta de decodificación QR: Para el caso de la decodificación de código QR, se tomó la decisión de implementar una Raspberry Pi 3A+, cuyas características se detallan en el cuadro XII. Esta tarjeta de control se encargará de decodificar los datos capturados en tiempo real para poder emitirlos y ocasionar el funcionamiento del sistema.

VIII-C7. Tarjeta de control: Además de la tarjeta de decodificación, se utilizó la placa Arduino Mega 2560 con el propósito de proporcionar movimiento al robot. Se eligió este embebido debido a la gran cantidad de entradas analógicas que posee, lo cual facilita el control directo de los motores y sensores que conforman el sistema.

Cuadro V
 CARACTERÍSTICAS DEL LAS LUCES UV

Especificaciones de la lampara uv	
Nombre del Producto	UV disinfection Lamp
Potencia Nominal	3.8w
Tensión Nominal	DC 3.7V
Dimensiones del producto	56*56*125mm
Tiempo de vida util	30000H
Bateria	1000maH
Modelo	ST-XD-08
Voltaje de carga	DC 5.0V
Dimensiones el paquete	7.6*7.6*14.8cm

VIII-C8. *Tarjeta de conexión auxiliar:* Dentro del hardware electrónico del robot, se elaboró una tarjeta auxiliar con el objetivo de establecer conexión entre los componentes. Debido a la gran cantidad de módulos, sensores y motores, se eligió utilizar borneras de conexión y soldar los componentes a la placa para obtener una ubicación mucho más factible de cada elemento. Para disminuir gastos se evitó utilizar una tarjeta de conexiones comercial y se fabricó una de acabado semiprofesional que cumpla con la necesidad del proyecto.

Para el diseño de la tarjeta se utilizó el software Proteus v8, el cual posee las herramientas adecuadas para poder realizar los diagramas del circuito. Dentro del software inicialmente debemos de escoger los elementos utilizados en el proyecto, la estructura del diagrama se encontrará segmentada en las fases de control y fuerza que se observa en la figura 17.

La etapa de control será alimentada mediante baterías que suministrarán con 5V DC a los elementos que requieran esa cantidad de voltaje, estos componentes serán controlados mediante los pines de entrada/salida del Arduino Nano.

Por otro lado el encendido y apagado de las luces UV se accionará por medio de dos relé 5V DC. El sensor de distancia GP2Y0A21YK0F y el relé de accionamiento serán controlados mediante las entradas digitales y analógicas del Arduino Nano dependiendo de su configuración. En la siguiente figura 19 se detallan las conexiones de los componentes del hardware del sistema utilizando la herramienta proteus.

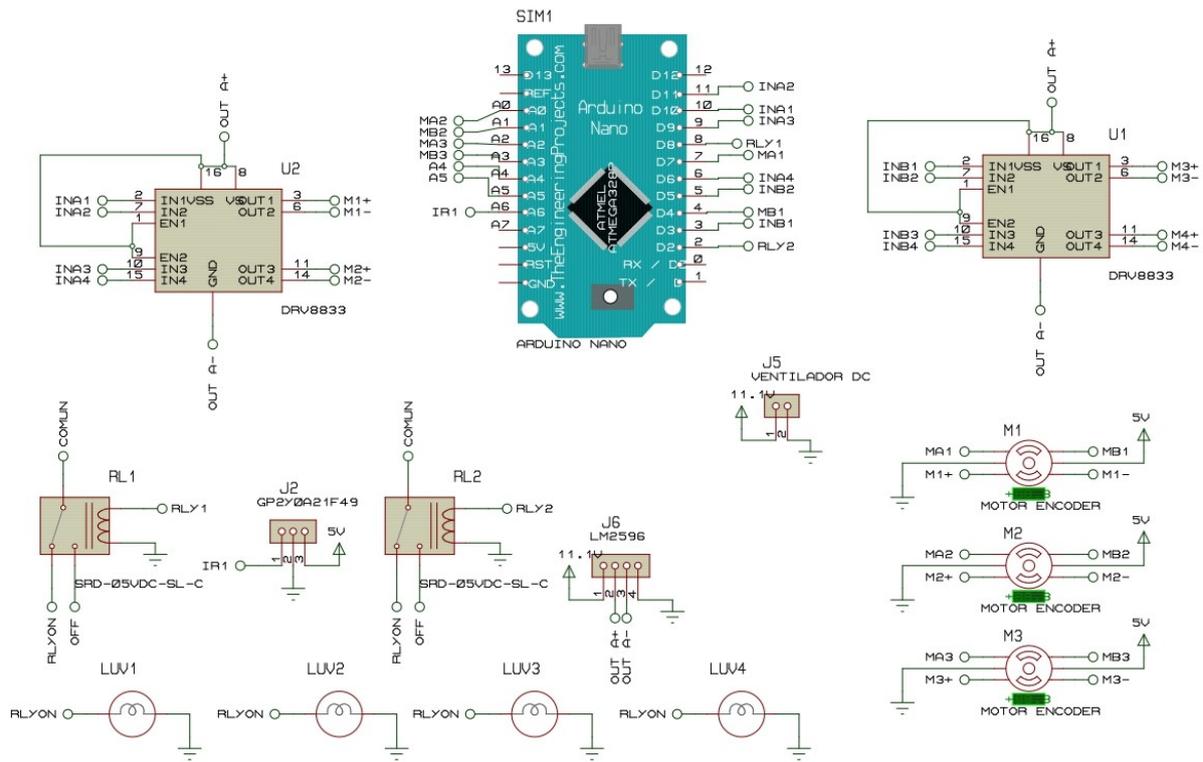


Figura 19. Diagrama de fuerza realizado en el software CADesimu

VIII-D. Dimensionamiento de motores

Para seleccionar el motor, se analizaron aspectos físicos relacionados al movimiento del sistema, como lo son la velocidad, el cálculo de la carga sobre los motores y el número de motores necesarios para generar una variación de posición en el autómata. Para este proceso se requiere de hacer uso de la 10, la cual detalla la carga máxima que el motor debe soportar y la medida del radio de las ruedas dividiendo por el número de motores que se quiere ubicar en el robot.

$$Tr = \frac{K * R_l}{n_r} \quad (10)$$

$$Tr = \frac{3,04 * 8}{3} \quad (11)$$

$$Tr = 8,10kgf * cm \quad (12)$$

Donde:

Tr = Torque instantáneo.

K = Carga del robot.

R_l = Radio de la llantas.

n_r = Número de motores.

En base a los datos obtenidos y comparándolos con el cuadro XI del apéndice B, se consideró que el motor de 140 RPM (cuadro VI) es el más adecuado, debido a que posee un torque aproximado al necesario para la movilización eficaz de la estructura del robot.

Cuadro VI
CARACTERÍSTICAS DE MOTOR ENCODER 140 RPM

Lista de parámetros motor encoder	
Tensión nominal	DC 12,0V
Sin velocidad de carga	140 RPM 0,10A
Max eficiencia	Carga 1.7 Kg.cm/110rpm/2,1W/0,33 A
Max energía	Carga 4,3 Kg.cm/70rpm/3,2W/0,75 A
Parada	Torque de parada 8,5 kg .cm Corriente de parada 1,4A
Relación de reducción del retardo	1:45
Resolución Holzer	Resolucion del motor 11 x ratio 45=495 PPR

VIII-E. Diseño de software de control

En esta sección se detalla la secuencia de funcionamiento del robot y los parámetros de configuración necesarios para la rotación de los motores, con el propósito de llevar a cabo la culminación del proceso de desinfección de una forma adecuada a las expectativas planteadas en el proyecto.

VIII-E1. Parámetros para el diseño de control: En este proceso se va a detallar cómo fueron calculados los parámetros de las ganancias integrales, proporcionales y derivativas en el proceso de control para obtener un PID de sintonía fina. Por ello se procede a identificar el sentido de giro en los actuadores que está en función de los datos que genera, como se detalla en el cuadro VII, se calculan también las velocidades máximas y mínimas, comprobando todo con los software de Arduino y Python.El código fuente de los parámetros de giro del motor se puede encontrar en el apéndice F

VIII-E2. Diagnóstico de la señales de control: En este apartado se detalla el proceso de captación de datos por medio del uso del software IDLE de Python como se detalla en la programación en la figura 104, se ingresaron librerías como “matplotlib” para la elaboración de gráficas. Para el control se generó una entrada escalón y una salida que se relaciona a los parámetros de giro del eje del motor expresados en rad/s. En este sistema se tomó un algoritmo de identificación llamado PCO, cuya función es dar de manera aleatoria valores para el modelo matemático del motor y de esta manera ir ajustando progresivamente el sistema dependiendo de la salida que hemos obtenido o la que proporcione el algoritmo de identificación. Por ello, una vez que el error sea mínimo y

Cuadro VII
 TABLA DE PARÁMETROS DE GIRO DEL MOTOR

	AB		Anterior	0	1	2	3	
	2	10	Actual	AB	00	01	10	11
+	3	11	0	00	0	-1	+1	E
	1	01	1	01	+1	0	E	-1
	0	00	2	10	-1	E	0	+1
			3	11	E	+1	-1	0

se esté aproximando a la respuesta de la curva tendremos un sistema convincente y equilibrado.

VIII-E3. *Control de sintonía fina de los motores:* Para este para este proceso se hará uso de la programación que se detalla en el apéndice H, la cual explica la respuesta de un escalón y la señal de giro de los motores de esta manera se busca reducir su error a cero.

En este proceso se toma un algoritmo de identificación llamado PCO, cuya función es dar de manera aleatoria valores para el modelo matemático del motor e ir ajustando progresivamente el sistema dependiendo de la salida que hemos obtenido o la que proporcione el algoritmo de identificación. Por ello, una vez que el error sea mínimo y se esté aproximando, tal como se observa en la figura 21, tendremos un sistema convincente y equilibrado tal como se detalla en la figura 20 .

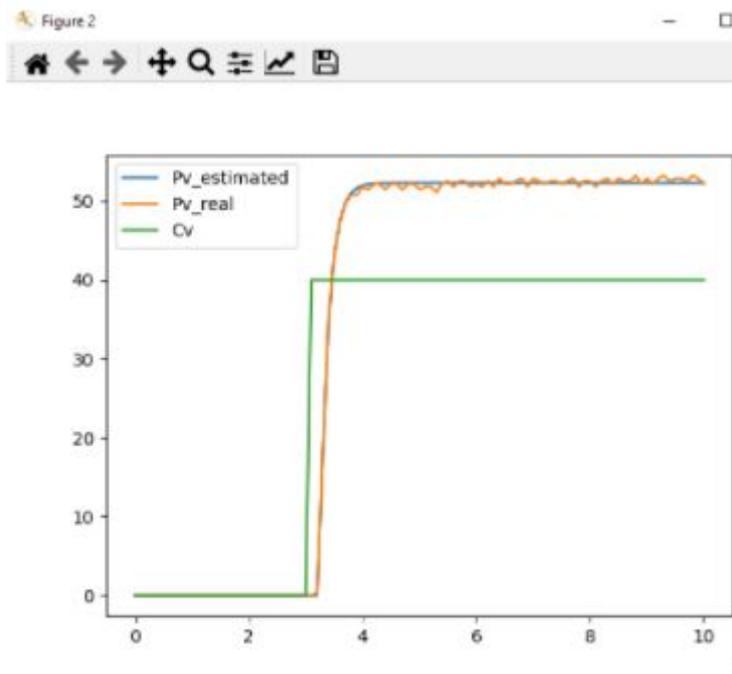


Figura 20. Señal de aproximación

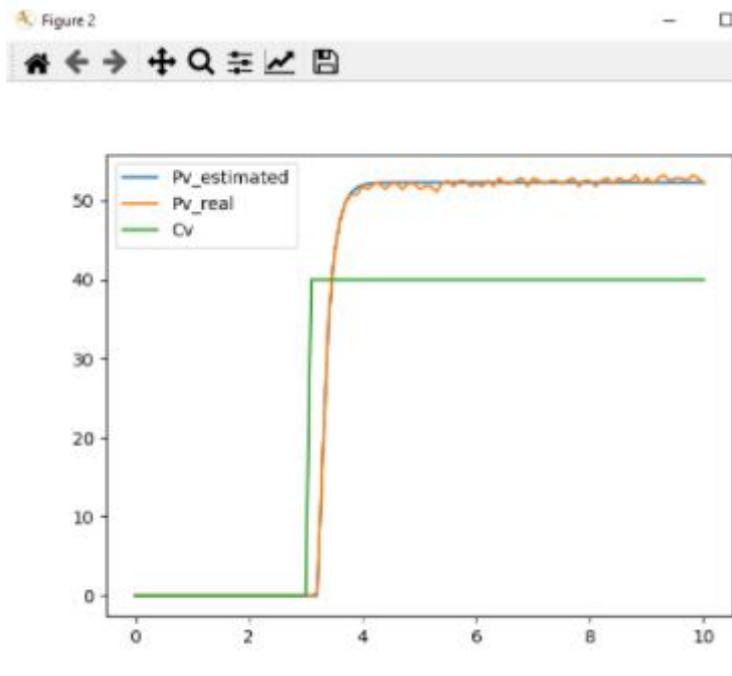


Figura 21. Señal de error

Para obtener la gráfica de la sintonía fina del robot, se hace uso de la importación numpy para demostrar arreglos matriciales como se demuestra en la figura 22.

```

from pyArduino import *
import matplotlib.pyplot as plt
import numpy as np

ts = 0.1 # Tiempo de muestreo
tf = 40 # Tiempo de simulacion
t = np.arange(0,tf+ts,ts) # Array de tiempo
N = len(t) # Numero de muestras

##### Comunicacion Serial #####

port = 'COM5' # Com Arduino
baudRate = 9600 # Baudios

arduino = serialArduino(port,baudRate,2) # Objeto serial
arduino.readSerialStart() # Inicia lectura de datos

```

Figura 22. Importación de librería numpy

En base al resultado que hemos obtenido, se debe lograr que el error decaiga rápidamente a cero disminuyendo el tiempo de integración. Luego en el código de arduino ubicamos los valores expresados en figura 23 para detallar la ganancia proporcional, derivativa e integral la cual vamos a ir disminuyendo la ganancia integral para que nuestro sistema sea estable. Para ello disminuimos el valor de 0.18 a 0.09 en la ganancia integral.

```

void setup() {

  Serial.begin(9600);
  motor.setGains(0.19,0.09,0.00); // (Kc,Ti,Td)
  motor.setCvLimits(255,40); // velocidad max 255 y 40 min
  motor.setPvLimits(8,0); // rad/s max 8.79 y min 0.7
  //motor.lambdaTunning(1.7789,0.183,0.001); // (kp,tau,delay)
}

```

Figura 23. Parámetros para evaluar el control de sintonía fina

Para poder visualizar la gráfica del control, se aplica un setpoint de trayectoria como se demuestra en la figura 24.

```

##### Setpoint Trayectoria #####
sp = 3*np.cos(0.2*t)+5|

```

Figura 24. Cambio del setpoint

En la figura 25 se presenta una prueba al sistema de control en sintonía fina, en donde la señal de set point es de color azul, y la la señal de salida (rpm), es de color naranja, se observa que el sistema sigue bastante bien la trayectoria, por lo que el control diseñado cumple con el objetivo deseado. En la figura 26 se presenta la señal de control para el actuador .

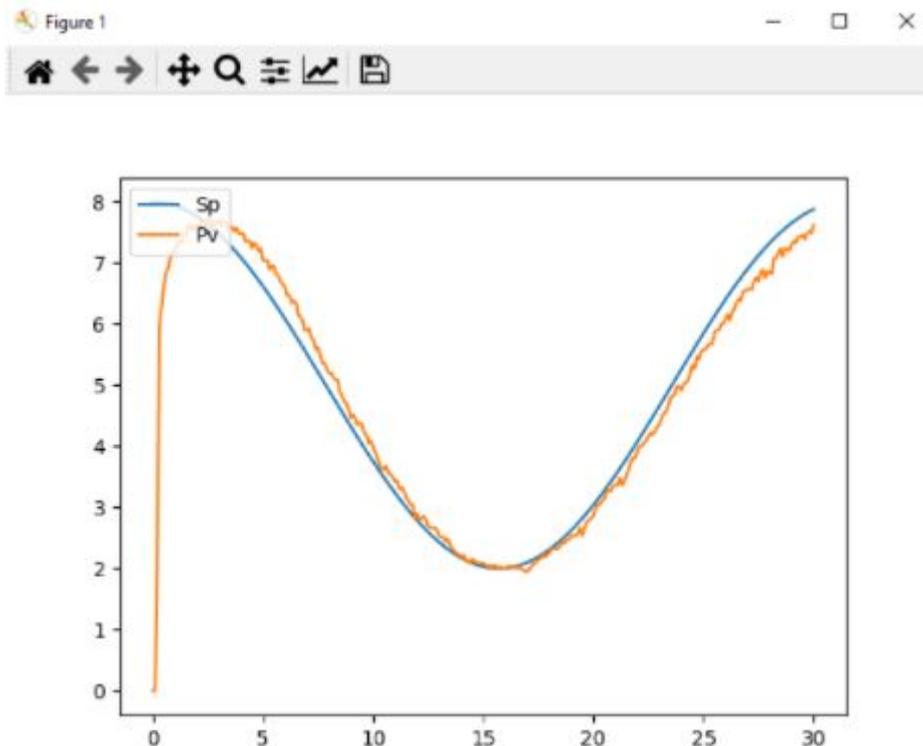


Figura 25. Señal de proceso en el control de sintonía fina

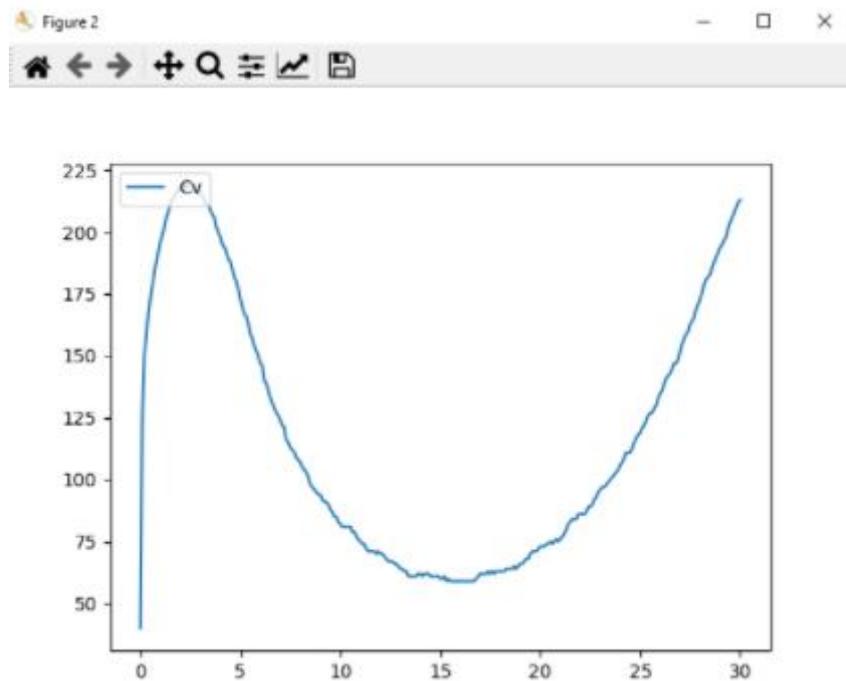


Figura 26. Señal de control

VIII-E4. Diagrama de secuencia de funcionamiento: La secuencia inicia definiendo la modalidad del robot, para este proyecto se plantea un modo automático y un modo manual. Dentro de la secuencia, se genera una bifurcación que llevan a cabo distintas acciones dependiendo de la modalidad seleccionada, sin embargo con cualquier opción se puede cumplir el proceso de desinfección.

En el caso de ser escogida el modo manual, el sistema optara por realizar una constante lectura de datos emitidos vía serial, controlando así el movimiento general del robot y cumpliendo con el proceso de desinfección mediante maniobras escogidas por el usuario. Por otro lado, si se escoge el proceso automático, se iniciará una constante lectura de código QR y transmisión de datos, provocando que el robot tome la decisión de realizar sus movimientos en base a un recorrido planteado previamente en la programación. En esta modalidad interviene el sensor IR para apagar el robot en caso de perturbaciones externas inadecuadas. En la figura 27, se desarrolla el diagrama de flujo que muestra las modalidades de funcionamiento para llevar a cabo el proceso de desinfección requerido.

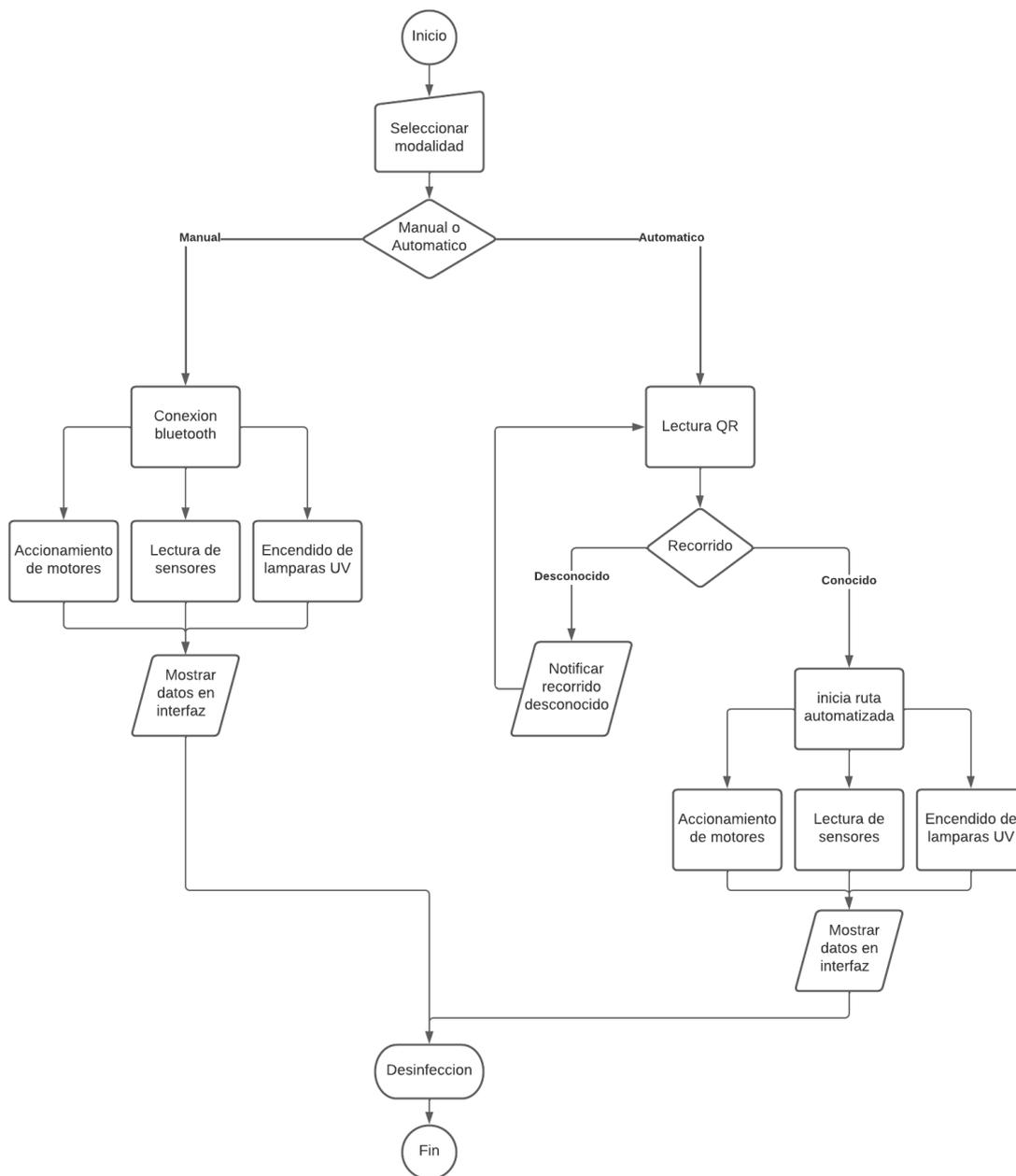


Figura 27. Conexión para el movimiento del motor

VIII-F. Desarrollo del software del robot

En este capítulo se desarrolla todo lo correspondiente a la implementación de los algoritmos de programación para las tarjetas de Arduino Mega y Raspberry Pi.

VIII-F1. Cálculo de la posición del eje del motor: Para este proceso se fue desarrollando una programación en el software de Arduino, la cual pretende estimar la posición del eje del motor en grados, haciendo uso de los sensores internos que contiene el motor encoder.

En la programación fue necesario incorporar la ecuación 13 . para poder estimar los grados de rotación, tal como se demuestra en el código que encuentra en el 99.

$$N = \frac{n * 360}{R} \quad (13)$$

VIII-F2. Cálculo de la velocidad en RPM y rad/s: Para este proceso se hace uso del código ubicado en las figuras 100 y 101, se muestra el cálculo de las RPM expresadas en rad/s de los motores, dando uso de las ecuaciones 14 y 15, las cuales determinan los valores requeridos para el funcionamiento de los actuadores.

$$N = \frac{n * 60}{t * R} \quad (14)$$

Donde:

N: Es la velocidad de rotación en revoluciones por minutos (RPM).

n : Es el número de conteo generado en un determinado tiempo.

t : Es el tiempo de generación de los pulsos en segundos(s).

R : Es la resolución del encoder para una precisión cuádruple.

$$W = \frac{2\pi * n}{t * R} \quad (15)$$

Donde:

w: Es la velocidad de rotación en radianes por segundo (rad/s).

n : Es el número de conteo generado en determinado tiempo t.

t; Es el tiempo de generación de los pulsos en segundos (s).

R: Es la resolución del encoder para una precisión cuádruple.

VIII-F3. Cálculo del movimiento del robot : Para determinar el proceso del movimiento de los actuadores fue necesario usar una programación en Arduino como se detalla en la figura 102 del apéndice F, en esta se observan los parámetros del diseño de control en el motor, como también la conexión serial del programa IDLE de Python.

Para el desarrollo del programa en Arduino, se utilizó la conexión entre un driver y la tarjeta Arduino mega, agregando el uso de alimentación por medio de una batería, como se muestra en la figura 28. El proceso se llevó a cabo para estimar un control de sintonía fina y de esta manera regular el movimiento de los motores al generar el debido proceso de desinfección.

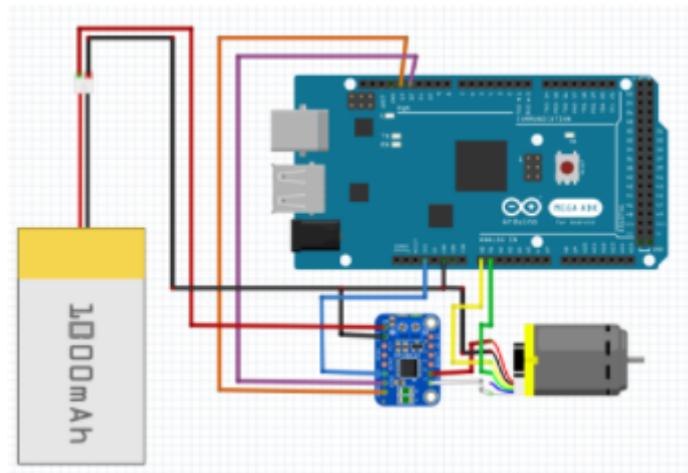


Figura 28. Conexión para el movimiento del motor

El programa de Python comprende un sistema de primer orden de lazo cerrado como se detalla en la figura 105, este sistema permite conocer los valores de k_p , τ y delay implementando el uso de la función “lambdaTunnig”, la cual proporciona una señal de control y un escalón para poder mostrar los datos antes mencionados.

Estos datos se implementaron en la programación como se observa en la figura 23, para que por medio del monitor serial, se detalle la ganancia integral, proporcional y derivativa. Este proceso es generado por medio de la librería “motorControl”, la cual estima la ganancias del actuador para sintonizar la señal hasta llegar a un proceso equilibrado.

VIII-F4. Calibración y configuración del sensor IR: Antes de implementar el sensor infrarrojo para comparación de distancia, se calibro el sensor mediante un algoritmo de prueba y error en la herramienta Arduino como se puede observar la figura 102 del apéndice F . El proceso muestra los datos de retroalimentación del sensor con los cuales se pretende obtener la posición más cercana o alejada del objeto que se le acerca, este desarrollo se propone con la finalidad de obtener un vector de posición que me permita reconocer el elemento que se le aproxima, específicamente en este proyecto es utilizado para medir la distancia con respecto a la superficie sobre la cual rueda el robot y asignar una acción programada que permita desactivar las luces UV en caso de que la estructura sea elevada una distancia preestablecida.

VIII-F5. Conexión y configuración de Raspberry PI: Para poder utilizar la Raspberry pi, como se indica en la figura 29, se realizaron distintas configuraciones iniciales con el objetivo de obtener mayor accesibilidad al momento de trabajar con la misma. Lo primero a tener en cuenta es que la Raspberry Pi necesita como elementos iniciales:

- Fuente de alimentación de 2.5A/5V DC.
- Conexión a teclado y mouse (Conexión inalámbrica o directa).
- MicroSD con sistema operativo (Raspbian).
- Pantalla con conexión tipo HDMI.

Dichos elementos se conectaron con el dispositivo y se procedió a energizar la tarjeta (Via Micro USB-B), lo cual provocará que se empiece a mostrar información en pantalla, lo primero que se observa es una pantalla de carga hasta que luego de un lapso de tiempo se habilita la interfaz del sistema operativo con un mensaje de bienvenida tal como lo muestra la figura 30.

En la figura 31 se puede observar la pantalla de bienvenida representa el inicio de la secuencia de configuraciones

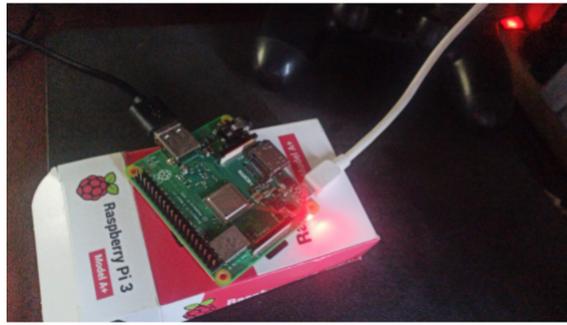


Figura 29. Raspberry Pi energizada



Figura 30. Mensaje de bienvenida de Raspbian

del sistema, a continuación de la misma, se mostrarán las configuraciones de idioma y región.



Figura 31. Configuración de región de Raspberry Pi

Luego de elegir las características correspondientes, el sistema pedirá que se asigne una contraseña de protección para poder eliminar la que viene por defecto de fábrica (figura 32).



Figura 32. Asignación de usuario y contraseña de usuario

Después se mostrarán configuraciones de ajuste de pantalla, conexión a red Wi-Fi, entre otras, sin embargo dichas

opciones no representan un paso relevante debido a que la Raspberry puede funcionar sin esto. Es recomendable ajustar cada parámetro a conveniencia del usuario. Finalmente el sistema preguntará si desea obtener las últimas actualizaciones (figura 33), es plausible aceptarlas para no tener que instalar paso a paso cada librería o programa.

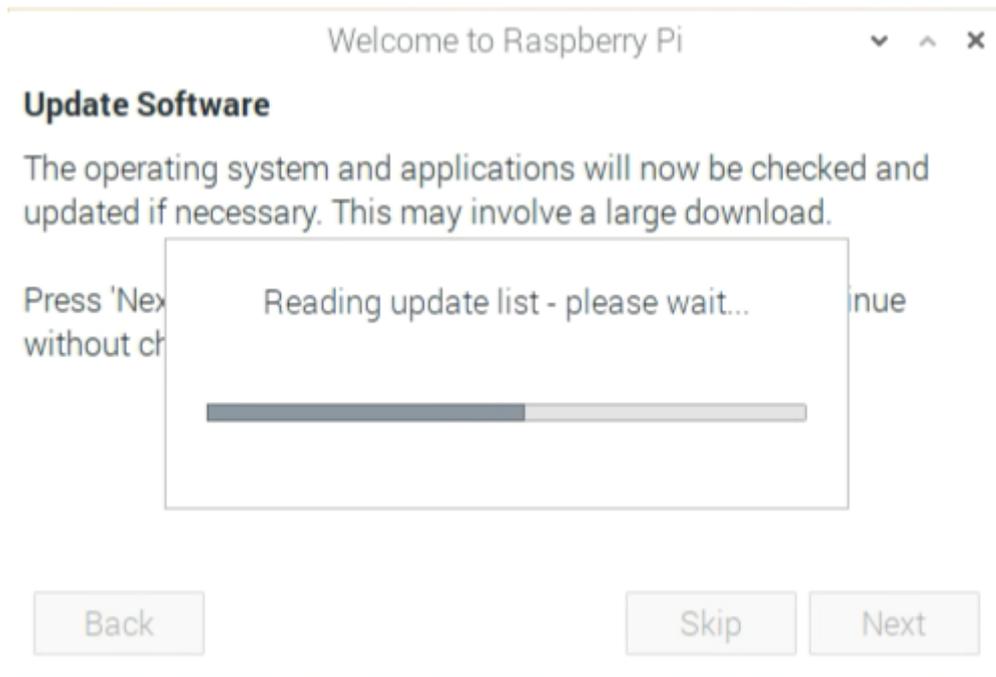


Figura 33. Actualización de software

El sistema reiniciará la tarjeta y se comenzara a trabajar sin problemas en la Raspberry pi. Teniendo en cuenta que se utilizara la IDLE de Python, se debe de instalar los programas necesarios para mejorar la accesibilidad durante el proceso. Se procede a abrir el terminal del sistema y digitar el comando “sudo apt update”, con esto se obtiene la última actualización de todas las aplicaciones, para esto se requerirá conexión a internet por lo cual se debe de conectar un cable de red o en su defecto establecer conexión con alguna red Wi-Fi local. En la figura 34 se muestra el terminal que se utiliza para instalar software y actualizaciones.

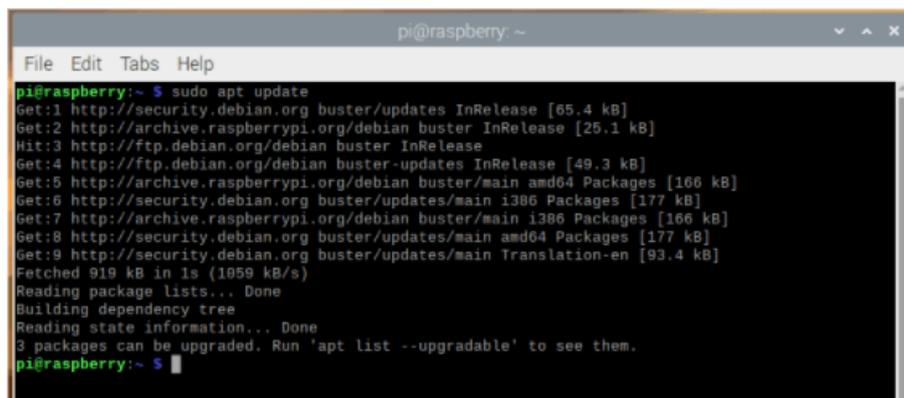


Figura 34. Actualización de programas predeterminados

Para instalar librerías se debe utilizar el comando “pip install” y de forma consiguiente escribir el nombre de la librería, para facilitar este proceso se recomienda utilizar el botón copytext de la pagina web “www.pypi.org” (figura

35), la cual proporciona las librerías de Python necesarias para el proyecto y los diferentes comandos utilizados para instalarlas.

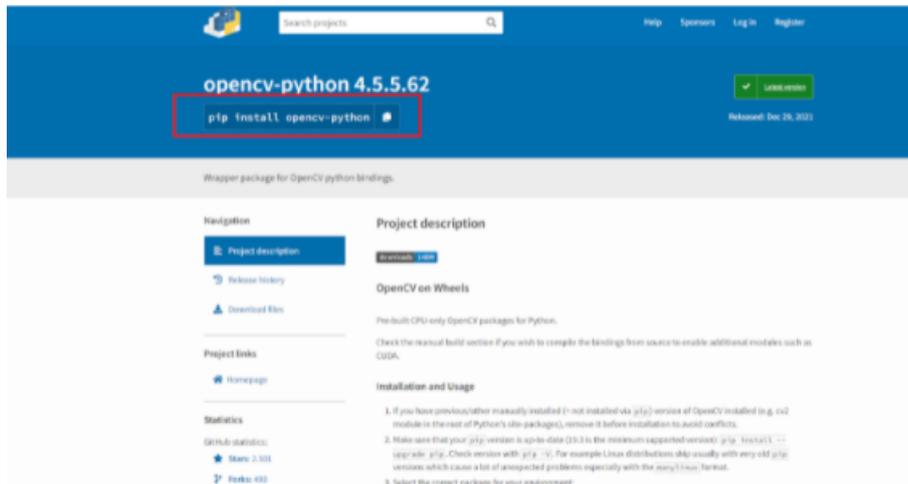


Figura 35. Comandos complementarios para instalación de librerías

Se requiere la instalación de las siguientes librerías utilizando el terminal como se muestra en la figura 36:

- opencv-Python
- pyzbar
- keyboard
- pycserial
- picamera

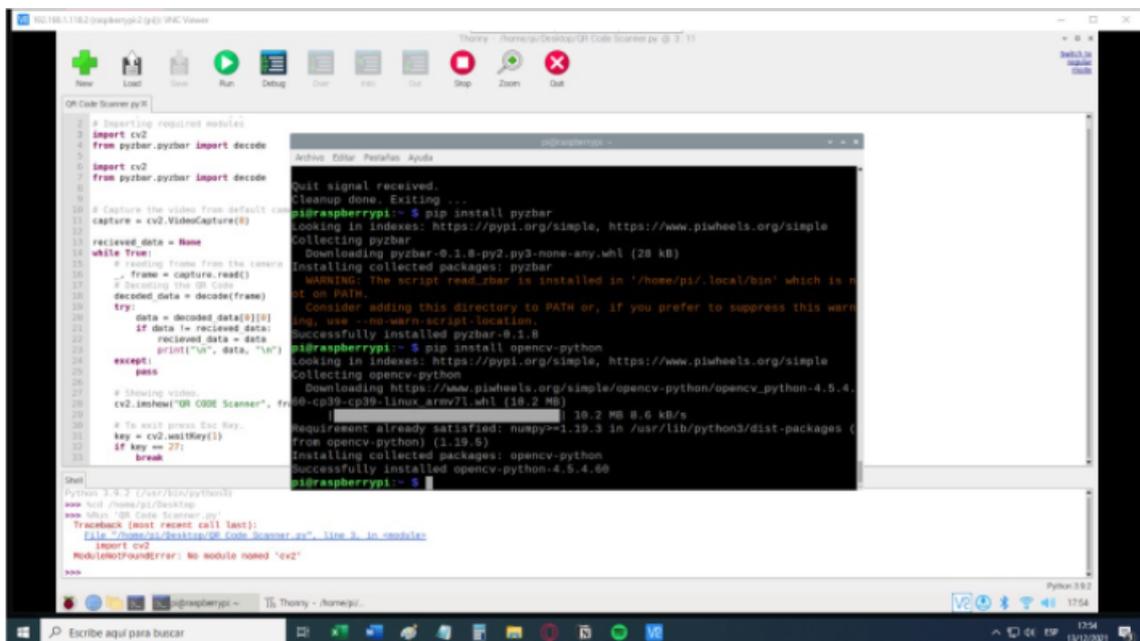


Figura 36. Instalación de librerías.

Se presenta como una incomodidad tener que conectar un teclado, ratón y pantalla a la Raspberry para poder usarla y posiblemente tener que retirar estos componentes de la computadora de trabajo principal, para solucionar este problema se utilizara el programa VNC viewer que permitirá utilizar de forma remota (Wi-Fi) la Raspberry por medio de cualquier computadora.

Lo primero que debemos hacer es ingresar a las configuraciones de la Raspberry, en el apartado de comunicaciones, y habilitar la comunicación VNC como se observa en la figura 37.

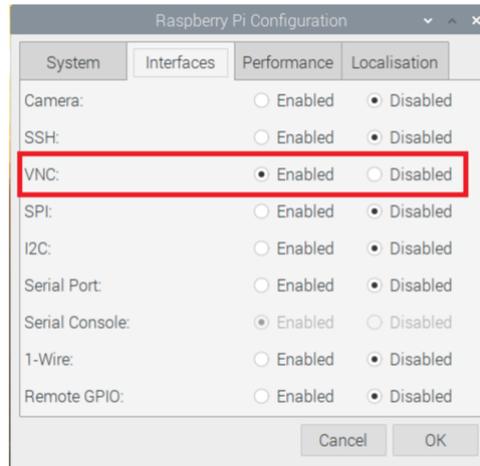


Figura 37. Activación de comunicación VNC

Luego, nos aseguraremos de que tengamos conexión a la red y colocaremos en el terminal de sistema el comando “ifconfig”, con esto se mostraran en pantalla las características de la interfaz de red de la Raspberry, de esto sencillamente extraemos el número IP del dispositivo.

Instalaremos el software VNC viewer en la computadora desde la cual manejaremos remotamente la Raspberry, crearemos una nueva conexión e introduciremos el nombre de la sesión y la ip del dispositivo externo (figura 38).

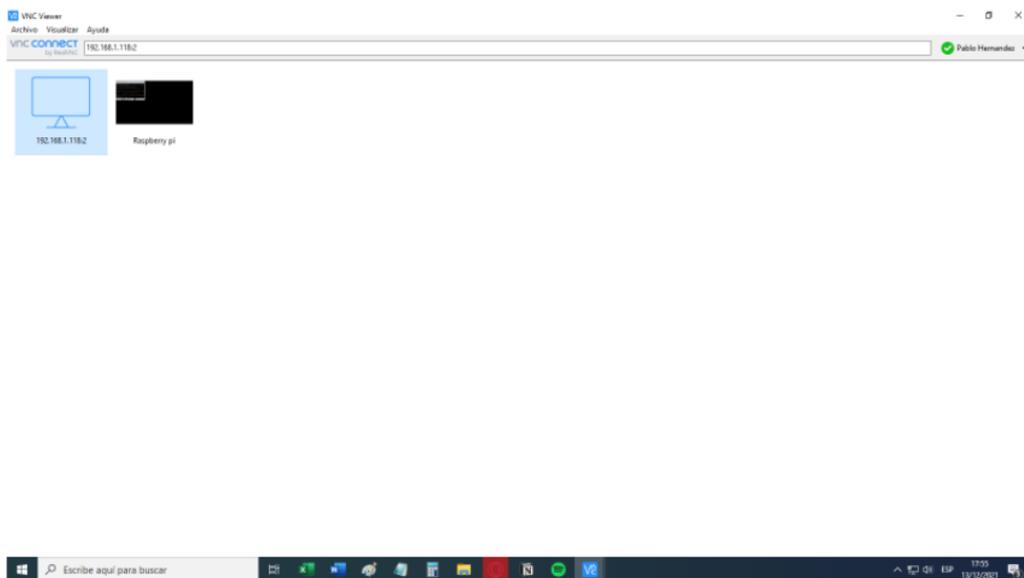


Figura 38. Selección de dispositivo mediante VNC viewer

Luego de ser creada la sesión, ingresamos a la misma y nos solicitará el nombre y contraseña de la Raspberry, las cuales configuramos previamente como indica en la figura 32, finalmente obtendremos la comunicación remota y podremos trabajar con una mayor facilidad.

VIII-F6. Algoritmo de lectura QR: La lectura de código QR es realizada mediante un algoritmos de Python que consiste en la utilización de una cámara mediante el sistema de la Raspberry pi y la combinación de funciones presentes en librerías de adquisición de datos en entornos visuales, las cuales se detallan en la figura 39 . La programación a la cual fue acoplada este algoritmo puede ser observada en el apéndice E

```
#Importar librerias
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
from pyzbar.pyzbar import decode
import os
```

Figura 39. Librerías utilizadas para lectura QR en Python.

Se procede a inicializar la cámara dentro del sistema y crear una variable destinada a la recepción de los datos que decodificó el programa de forma constante, esta variable se encontrara vacía de forma inicial en la figura 40.

```
# Inicializar la cámara
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera)
# Definicion de tiempo para renovar los datos recibidos de la camara
time.sleep(0.1)
recieved_data = None
```

Figura 40. Inicialización de cámara en Python.

Se inicia la recolección de datos, utilizando una variable para poder obtener la información de cada fotograma que emite la cámara, y así comenzar el proceso de decodificación de datos en un bucle “for” con el fin de que el proceso se esté realizando en todo momento de forma pasiva. Para poder observar y verificar mediante el terminal que la decodificación se esté realizando correctamente se utiliza el comando “print” en la variable donde se guarda el dato. El comando “decode” utilizado en ciertas líneas de la programación es utilizado para convertir la información entregada en formato “bytes” a un tipo “cadena” y facilitar la comprensión de los mensajes (figura 43).

```

# obtencion de fotogramas en videocaptura
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    # Decodificacion de codigo imagen (QR)
    decoded_data = decode(image)
    try:
        data = decoded_data[0][0]
        sala=data.decode()
        if data != recieved_data:
            recieved_data = data
            #Decode sirve para cambiar la variable de byte a cadena
            sala=data.decode()
            print(sala)
        #sala = None

```

Figura 41. Bucle de decodificación QR en Python.

Como se observa en la figura 44, se utiliza “except” para notificar al usuario en caso de errores inesperados durante la lectura QR y por motivos de visualización, se introduce una ventana mediante “cv2.imshow”, parametrizando la imagen a mostrar y el nombre de la respectiva ventana. También se establece un vaciado de cache del video capturado y un letra para poder cerrar la ventana inicializada previamente.

```

#Excepcion en caso de errores
except:
    pass
    # Mostrar una ventana con la videocaptura
    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
    # Vaciar el caché
    rawCapture.truncate(0)
    # Si presiona la letra 'q' del teclado, salga del bucle
    if key == ord("q"):
        break

```

Figura 42. Excepciones y ventana de video en Python.

VIII-F7. Interfaz de visualización: La interfaz programada mediante el lenguaje Python, la cual se muestra en el apéndice E, consiste en la utilización de las funciones implementadas en las librerías “tkinter” y “PIL” para poder plantear un conjunto de imágenes y animaciones, que permiten representar varios aspectos en el funcionamiento del robot, entre estos se encuentran la variación de temperatura del sistema, la dirección en tiempo real del robot, la sala en la que se encuentra realizando el proceso de desinfección, entre otros. En la figura 97 se muestra la interfaz de monitoreo en pleno funcionamiento.

VIII-F8. Programación de movimiento y accionamiento de robot de desinfección: Para este proceso haremos uso de una programación en la cual se detalla la funcionalidad de las lámparas por medio de conexión Arduino con relé.

El proceso pretende activar las lámparas por comandos de Arduino para que de esta manera pueda ser energizado el relé y puedan ser activadas las luces UV, dando de esta manera funcionalidad para el proceso de desinfección.

IX. RESULTADOS

En esta sección, se detalla el resultado propuesto en base a los objetivos planteados en el proyecto titulado “IMPLEMENTACIÓN DE UN ROBOT DE DESINFECCIÓN ULTRAVIOLETA PARA MICROORGANISMOS

IX-A. Resultados de las pruebas de desinfección

Para análisis de la desinfección se tomó en consideración, escoger un medio nutritivo para el crecimiento de bacterias, este proceso pretende mostrar en una perspectiva visible al ojo humano, la cantidad de bacterias que se encuentran en la superficie donde se realizó la muestra.

En el proceso de aplicación de la prueba, se escogió un lugar adecuado para realizar el análisis microbiológico, en donde se realizaron dos muestras. La primera muestra se consideró como el parámetro de control, el cual consistió en tomar los microorganismos presentes en esa área por medio de un hisopo esterilizado, se lo agrego al medio rayándolo y se lo ubico en una estufa para poder observar la cantidad de bacterias presente como se aprecia en la figura 43. La segunda prueba consistió, en utilizar el robot de desinfección sobre la misma área donde se realizó la primera muestra, para que de este modo se demuestre que el robot desinfectó el área que estaba contaminada tal como se puede detallar en la figura 44.



Figura 43. Crecimiento de bacteria en el medio



Figura 44. Comparación del control con la desinfección

IX-B. Resultados de la estructura del robot

Dentro de la simulación realizada en el software Inventor, la cual se utilizó como referencia para realizar la estructura del robot, se consideró que la estructura del chasis es la parte donde se suministra toda la carga generada por los componentes o elementos del robot. Por ello se desarrolló en el programa de inventor un análisis estático de dicha estructura, con la finalidad de observar y clarificar que la pieza pueda resistir las fuerzas existentes en su superficie y evitar deformaciones. El análisis estático, nos detalla el esfuerzo de Von Mises, la deformación y el coeficiente de seguridad, analizando de esta forma varios parámetros a tener en cuenta para el diseño. Por medio de este Inventor, se logró visualizar que el material seleccionado, puede resistir la carga impuesta, en la figura 45 se demuestra en una simulación que el material resiste y soporta a la condición que se le imponga.

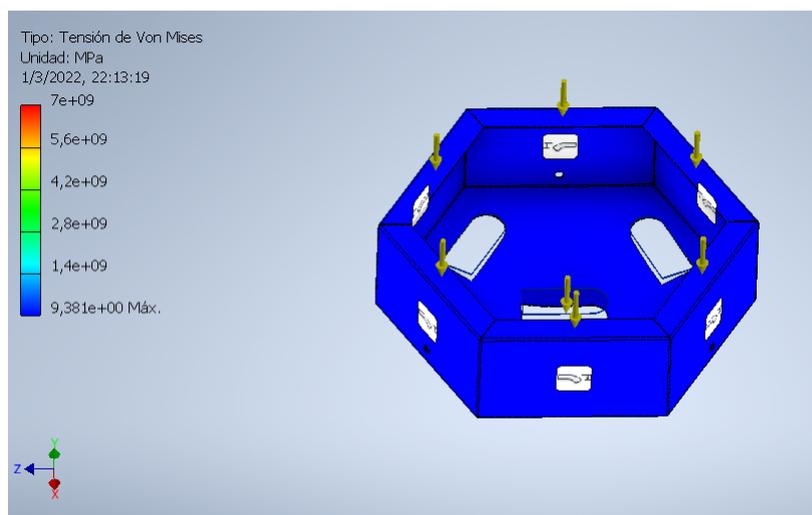


Figura 45. Esfuerzo de Von Mises del chasis

El análisis estático, detalla la deformación de la estructura y su coeficiente de seguridad, esto se muestra en la figura 46, por ello es recomendable saber el tipo de material de trabajo y sus propiedades de elasticidad que permiten dar mejor desempeño en su proceso de desinfección.

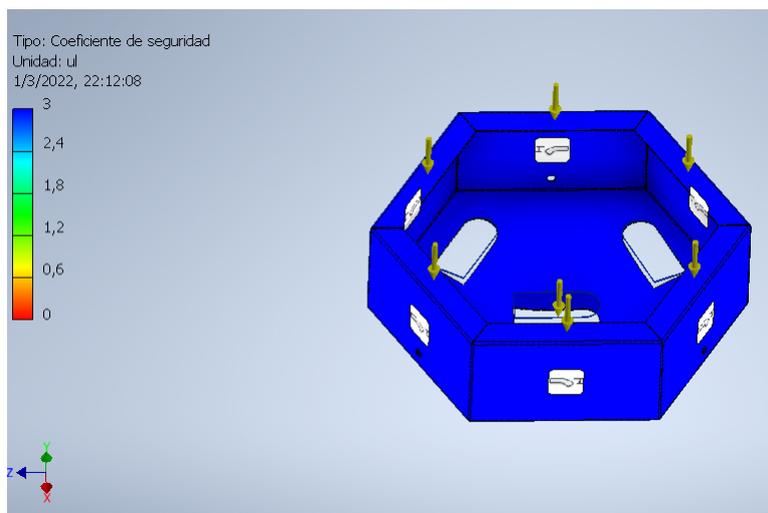


Figura 46. Coeficiente de seguridad del chasis del robot

Todos los análisis realizados mediante software, permiten la elaboración del ensamble teniendo en cuenta todos los factores que alteran el funcionamiento cinemático de la estructura del robot de desinfección como se muestra en la figura 47 y 49.



Figura 47. Perspectiva aérea de ensamble del prototipo



Figura 48. Ensamble del prototipo

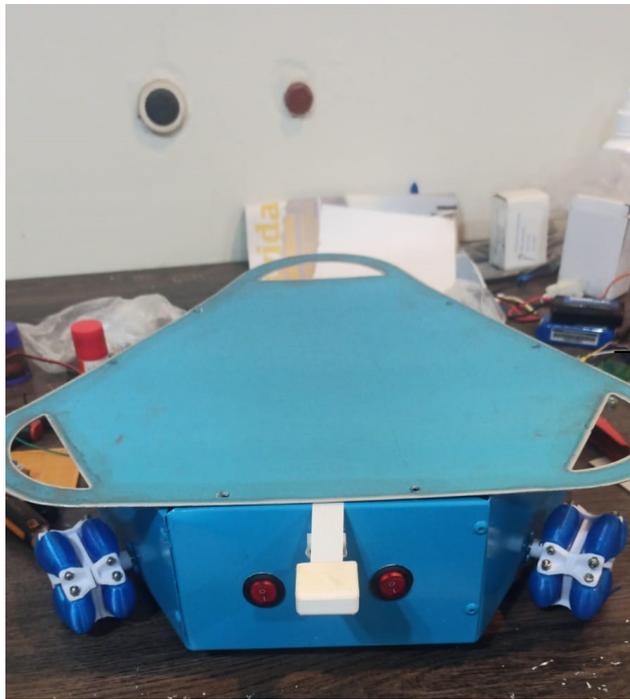


Figura 49. Ensamble de dispositivos externos del robot

IX-C. Resultados del funcionamiento del robot

En este proceso se detalla el movimiento manual y automático del robot de desinfección, haciendo uso de una interfaz gráfica para visualizar el giro de las ruedas, la temperatura interna del robot y adicionalmente controlar el movimiento del robot. 50.

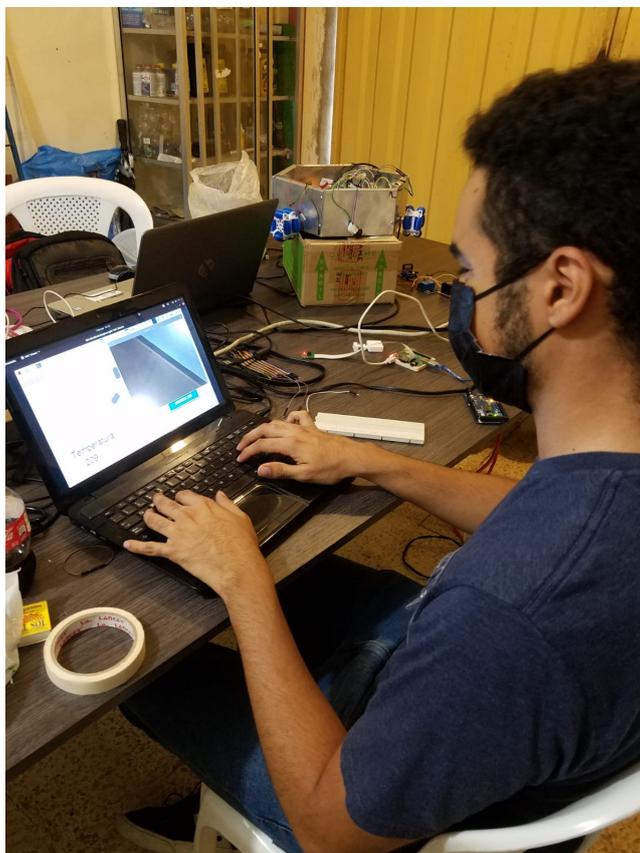


Figura 50. Pruebas de interfaz gráfica

La comunicación serial entre Raspberry Pi Y Arduino Nano, proporciona sin inconvenientes una transmisión y recepción bidireccional de datos en tiempo real que permite cumplir la función de la interfaz gráfica como medio de monitorización. En la figura 50 se muestra el resultado de la interfaz gráfica en pleno funcionamiento.

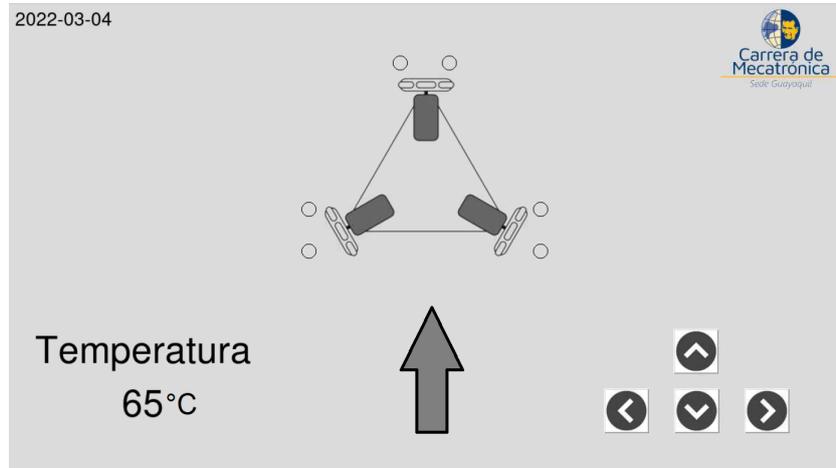


Figura 51. Interfaz gráfica de monitoreo y control

IX-D. Resultados de tarjeta electrónica

La placa electrónica PCB, fue realizada utilizando las conexiones esquemáticas presentadas en proteus (figura 18) y realizando una readaptación en el software EasyEDA para facilitar el proceso de esquematización del diagrama en PCB y por lo consiguiente su respectiva impresión mediante procesos químicos. En la figura 52 se puede observar la readaptación del esquemático.

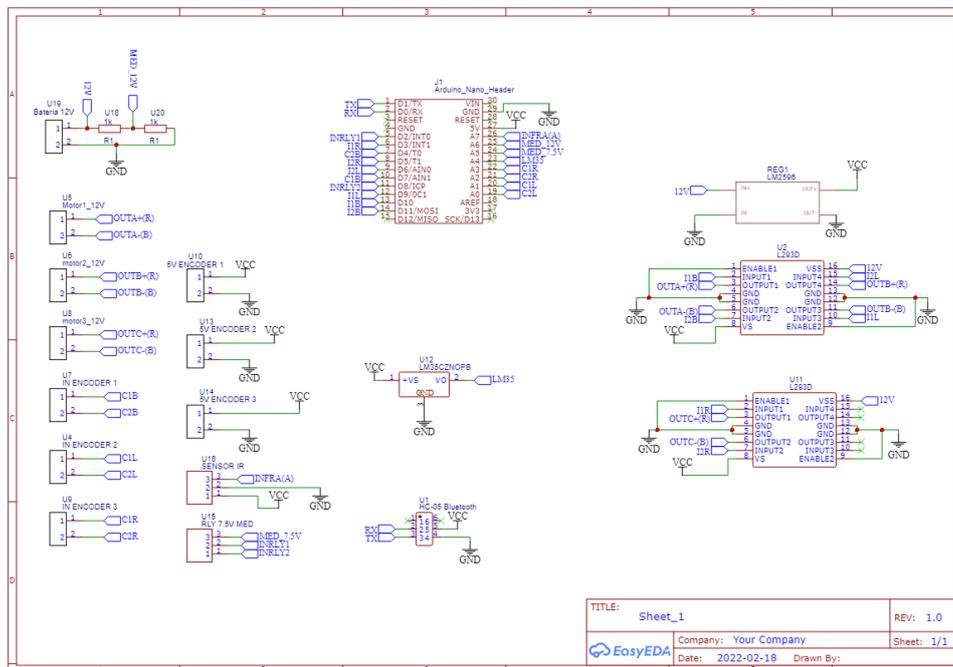


Figura 52. Diagrama de conexiones en software EasyEDA

La tarjeta, al ser dependiente de la cantidad de componentes electrónicos implícitos en el proyecto, posee unas dimensiones de 7.7cm de largo y 9.7 cm de ancho. Posee una variación entre pistas para soportar distintos valores de corriente que varían debido a los procesos de fuerza y control. En la figura 53 se observa el resultado de la impresión de la tarjeta PCB y en el anexo B se encuentra la película utilizada para la aplicación de antisolder y el recorrido de las pistas.

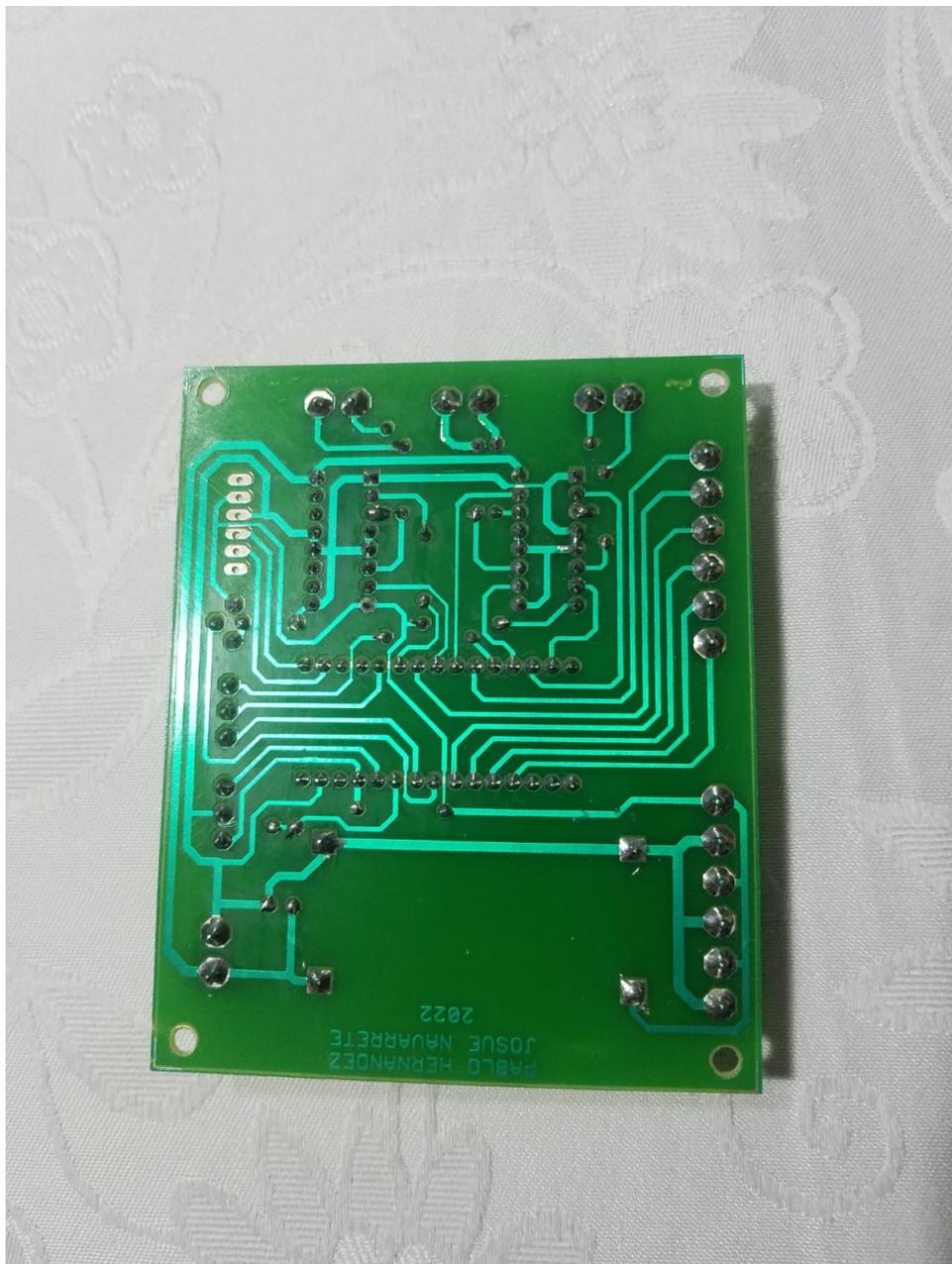


Figura 53. Parte trasera de tarjeta de control auxiliar

X. CRONOGRAMA DE ACTIVIDADES

A continuación, en el tabla X se observan el las diferentes actividades realizadas proyectadas.

Cuadro VIII
CRONOGRAMA DE ACTIVIDADES DEL TRABAJO DE TITULACIÓN

Actividad	2021				2022	
	Mayo-Junio	Julio-Agosto	Septiembre-Octubre	Noviembre-Diciembre	Enero-Febrero	
Elección del tema						
Elaboración del planteamiento del problema						
Desarrollo de la Justificación						
Redacción de los Objetivos						
Elaboración de la Hipótesis						
Progreso del Marco Teórico						
Creación de la Metodología						
Creación del diseño y estructura						
Estudio del prototipado						
Cálculos del movimiento del robot						
Selección de materiales						
Elaboración del esquema eléctrico						
Desarrollo del software						
Verificación de pruebas microbianas						
Desarrollo del libro						
Pruebas técnicas de funcionamiento						
Implementación del prototipado						
Calibración de sensores y actuadores						
Conclusiones y Anexos						
Referencias y Bibliografías						

XI. PRESUPUESTO

A continuación, en la tabla XI se observan el costo de los diferentes componentes del proyecto.

Cuadro IX
PRESUPUESTO DEL TRABAJO DE TITULACIÓN

N	ITEM	Unidad	Precio Unitario	Precio total
1	Sistema de control			
	Batería lipo	1	90	90
	Batería litio	2	32	64
	Bankpower KlipXtreme	1	40	40
	Bankpower VIDVIES	1	15	15
	Módulo de Relé	4	4.25	17
	Arduino nano	3	10	30
	Raspberry Pi 3 A+	1	45	45
	Lamparas UV	4	17	68
	Ruedas Ominiderccional	3	32	96
	Motor Encoder 140 RPM	4	18	72
	GP2Y0A21YK0F Sensor de Rango	4	12	48
	Cámara raspberry pi 5mp	1	10,50	10,50
	Driver L293D	6	7	42
	LM 2596 regulador de voltaje	3	3	6
	borneras 2 terminales	20	0,25	5
	LM35 Sensor de temperatura	3	1,85	5,55
	Header regleta simple hembra	4	0,5	2
	Header regleta simple macho	4	0,5	2
2	Materiales y Componentes			
	Impresión 3D	4	32.5	130
	Lamina de Aluminio	1	40	40
	Estructura	1	32	32
	Cortado en laser	2	15	30
	Tarjeta PCB	2	32	64
3	Materiales de oficina			
	Papel	2	10	20
	Impresiones	2	30	60
4	Internet	160 horas		150
5	Gastos variados			
	Asesorias			300
	Viáticos			200
			Subtotal	1684,05
			Imprevistos	200
			TOTAL	1884.05

XII. CONCLUSIONES

- Se diseñó la estructura con el software inventor, estableciendo la ubicación de los elementos que debían ir dentro del robot, se desarrolló un análisis estático de la pieza con el propósito de demostrar que el material con el cual fue armada la estructura soporta la carga nominal, concluyendo que el esfuerzo y el coeficiente de seguridad eran confiables.
- Mediante el software Proteus, se pudieron realizar de forma esquemática los circuitos necesarios para el funcionamiento del sistema electrónico, obteniendo así una idea de los componentes fundamentales para obtener los valores de corriente y voltaje adecuado tanto en el circuito de control de movimiento, como en el circuito de control de desinfección. Al obtener los valores de corriente de cada componente, indagando en sus características de amperaje máximo y calculando mediante la ley de ohm, se pudo realizar la selección de fusibles indispensables para proteger las fuentes de alimentación del sistema. Al finalizar los diagramas electrónicos, se ubicaron los componentes en un formato de placa de circuito impreso mediante el software EasyEDA para estimar el recorrido de las pistas y solicitar la fabricación de la tarjeta PCB.
- Se realizó la verificación de la eficacia del robot, para ello se hizo una prueba microbiológica antes del recorrido del robot sobre una superficie de experimentación, en dicha prueba se utilizó un medio de cultivo nutritivo para bacterias en la que se constató la existencia de microorganismos patógenos; posteriormente se programó una trayectoria al robot sobre la misma superficie, después de ello se utilizó un nuevo medio nutritivo con el cual se evidenció la eliminación de los microorganismos encontrados en la prueba anterior.
- La omnidireccionalidad del robot y su libertad de desplazamientos laterales sin la necesidad de realizar giros, proporcionó buenos resultados para llevar a cabo el proceso de desinfección, tanto en modo manual como automático. La interfaz visual de monitorización acompañó oportunamente el movimiento de los motores del robot. La adquisición y procesamiento de datos en tiempo real para la toma de decisiones, concluyó en un gran beneficio para conocer el estado del robot y comandar los recorridos de desinfección establecidos.

XIII. RECOMENDACIONES

- Acertar en el diseño de la estructura de un robot móvil, es un factor relevante para poder planificar los demás componentes que son parte del robot, incluso condiciona algo intangible como lo es la programación del autómata. Por esta razón, se debe de realizar un estudio exhaustivo de la elección de la estructura, utilizando todas las herramientas de simulación que sean necesarias, sin embargo, resulta conveniente el desarrollo de prototipos de bajo costo para reforzar la proyección de funcionamiento, esto nos permite conocer de mejor manera los límites que conforman la armadura del robot teniendo en cuenta distintos escenarios como choques, peso, distribución de cargas, puntos de anclaje principales, espacio disponible, entre otros.
- Al diseñar una placa PCB para contener el hardware electrónico del sistema, se debe de tener en cuenta el gran tiempo de espera que representa solicitar una tarjeta con características sobresalientes, por esta razón resulta conveniente realizar un análisis que determine si es posible realizar una placa minimalista con pistas existentes en un solo lado, y así evitar ser obligado a utilizar componentes especiales como el antisolder o mejorar el tiempo de espera y costo. Es fundamental no hacer uso de programas de *autoroute* preferiblemente diseñar el paso de las pistas de forma manual, con esto se obtiene un resultado con mucho más orden, a partir de esto se logra ser consciente de posibles errores de diseño en la placa.
- Cuando se realizan procesos de comunicaciones entre dispositivos embebidos, puede llegar a ser problemático obtener buenos resultados de forma sencilla, debido a que funcionan utilizando dos lenguajes de comunicación diferentes, en este caso Raspberry Pi y Arduino, que en proyecto se programaron utilizando Python y C++ respectivamente, no poseen grandes inconvenientes al momento de recibir y emitir datos, sin embargo si tienen ciertas limitaciones que ralentizan el proceso de desarrollo. Lo más importante a tener en cuenta, es conocer que el mejor método de comunicación en este formato es vía serial, sin embargo hay que considerar que los mensajes deben de ser emitidos y recibidos en formato "bytes".^{en} ambos dispositivos, esto implica estar constantemente codificando y decodificando las ordenes de accionamiento implícitas en la comunicación y variar utilizando formatos de codificación como utf-8 y ASCII dependiendo de la conveniencia del caso.

REFERENCIAS

- [1] L. "López Cerero, «Papel del ambiente hospitalario y los equipamientos en la transmisión de las infecciones nosocomiales,» vol. 7, págs. 459-464, 2014.
- [2] G. F. 2. -. 2021, «L293D Circuito integrado puente H,» 2021.
- [3] A. Álvarez, «Los protozoos. Características generales y su rol como agentes patógenos,» *Facultad de Ciencias Veterinarias Argentina*, vol. 8, 2018.
- [4] J. Alvarracín Rojas, «IMPLEMENTACIÓN DE UN SISTEMA DE EVALUACIÓN Y CAPACITACIÓN DE LIMPIEZA Y DESINFECCIÓN DE EQUIPOS BIOMÉDICOS EN LOS SERVICIOS DE UCI I Y II DEL HOSPITAL UNIVERSITARIO SAN JOSÉ DE POPAYÁN,» *Universidad del Rosario*, vol. 6, págs. 951-952, 1967.
- [5] Ambientech, «Microorganismo o microbio,» 2019.
- [6] Arduino, «Arduino Nano,» 2016.
- [7] C. Banchón, T. Borodulina, L. Diaz y A. Yasuji, «Aerobiología en hospitales de Guayaquil: microorganismos resistentes a cobre,» *Revista Riemat*, vol. 4, págs. 44-49, 2019.
- [8] J. Bausa, C. Gómez, B. Zaragoz, A. Martínez, D. Campos y A. Llinares, «Sensores de Temperatura Medida de Temperatura. Sensores de Temperatura.,» 2019.
- [9] A. Becerra, A. Correa y M. Torres, «Revisión histórica del registro fósil de protozoos e invertebrados marinos del estado de Tamaulipas , México.,» *Universidad Nacional Autonoma de Mexico*, vol. 9, págs. 135-143, 2020.
- [10] V. Benjumea y M. Roldán, «Fundamentos de Programación con el Lenguaje de Programación C++,» págs. 195-198, 2017.
- [11] J. Bettie, «VIRUS,» *National Human Genome Research Institute.*, 2020.
- [12] C. Biodiversidad, «Algas. Biodiversidad Mexicana.,» 2020.
- [13] T. Bräunl, «Mobile Robot Design and Applications with Embedded Systems. Embedded Robotics.,» 2006.
- [14] N. Briones Cando, M. Zambrano Torres, F. Febres calderón y K. Cuenca-león, «Luz ultravioleta para desinfección en áreas de salud, frente al covid-19.,» *Revista OACTIVA UC Cuenca*, págs. 107-114, 2020.
- [15] C. Cabrera Gia y C. Silverio Calderón, «Determinación de Microorganismos en Ambiente del Área de Neonatología de un Hospital ubicado al Sur del Ecuador.,» *Polo Del Conocimiento*, vol. 6, págs. 96-97, 2019.
- [16] Z. Calvopiña Limaico, «ALMACEN INTELIGENTE MEDIANTE ROBOT OMNIDIRECCIONAL Y VISIÓN ARTIFICIAL,» *Universidad Politécnica Salesiana*, 2018.
- [17] L. Carrillo, «Los Hongos de los alimentos y Forrajes.,» *Universidad Nacional de Salta*, 2006.
- [18] U. Cordoba, «Guía de actividades prácticas microbiología agrícola.,» págs. 33-37, 2015.
- [19] M. Correa, S. Mera, F. Guacho, E. Villarreal y S. Valencia, «Desinfección mediante el uso de luz UV-C germicida en diferentes medios como estrategia preventiva ante la COVID-19. 21.,» págs. 21-23, 2020.
- [20] T. Davila, K. Uvidia y G. Vilaña, «ESTUDIO DE LAS NORMAS DE BIOSEGURIDAD Y PROCEDIMIENTOS DE LIMPIEZA EN LAS TRES ÁREAS CRÍTICAS DEL HOSPITAL DE ESPECIALIDADES DE LAS FUERZAS ARMADAS.,» págs. 1-7, 2018.
- [21] D. Delgado, C. Ortiz, H. Daza y M. Arias, «Evaluación del uso de luz UV como alternativa para la descontaminación de equipos odontológicos,» *Congreso Internacional de Ciencia y Tecnología Para El Desarrollo Sostenible, Chiriquí, Panamá.*, 2018.
- [22] I. Doroftei, V. Grosu y V. Spinu, «Omnidirectional Mobile Robot - Design and Implementation. Bioinspiration and Robotics Walking and Climbing Robots.,» 2007.
- [23] M. Electronics, «Raspberry Pi 3 A+,» 2021.
- [24] Y. Fandiño Mendez, D. Rebellon, X. Cediél y X. Castañeda, «MICROBIOLOGIA BASADA EN EVIDENCIA PARA EL MEDICO RURAL COLOMBIANO,» *Universidad Pedagógica y Tecnológica de Colombia*, 2016.
- [25] F. Fernández Carrasco, J. Vázquez Lara, U. González Mey, J. Gómez-Salgado, T. Parrón Carreño y L. Rodríguez Díaz, «INFECCIÓN POR CORONAVIRUS COVID-19 Y LACTANCIA MATERNA: UNA REVISIÓN EXPLORATORIA. Revista Española de Salud Pública,» 2020.

- [26] A. Fernández Hernández, R. Posso y L. García Reyes, «DISPOSITIVO REMOTO PARA DESINFECCIÓN DE ENTORNOS CERRADOS MEDIANTE LUZ ULTRAVIOLETA,» *UNIVERSIDAD TÉCNICA DE AMBATO*, vol. 53, págs. 9-12, 2021.
- [27] J. Gonzalez y A. Garcia, «Codigos QR y sus aplicaciones en las ciencias de la salud.,» *Revista Cubana de Información En Ciencias de La Salud*, vol. 2, págs. 239-248, 2016.
- [28] A. Guillermo, G. Mejía y K. Zanabria, «Determinación de microorganismos en el aire de los laboratorios de microbiología de la Universidad Continental,» 2018.
- [29] A. Hernandez Romero, «Análisis económico de un sistema de almacenamiento para la disminución de desvíos de producción en un parque eólico.,» *Universidad De Sevilla*, págs. 228-230, 2016.
- [30] Z. Idrogo Maita y D. Ruíz Campos, «DETERMINACIÓN DEL DESINFECTANTE SEGURO ORTOFTALDEHIDO VERSUS GLUTARA LDEHIDO UTILIZADO POR LAS ENFERMERAS EN EL PROCESO DE DESINFECCIÓN,» *UNIVERSIDAD PRIVADA NORBERT WIENER*, vol. 3, págs. 228-230, 2016.
- [31] R. Iglesias, A. Lago, A. Noguerias y col., «Modelado y simulación de una batería de Ion-Litio comercial multicelda.,» *UNIVERSIDAD PRIVADA NORBERT WIENER*, págs. 464-469, 2015.
- [32] F. Llordachs Marqués, «Breve historia de la asepsia y antisepsia,» 2021.
- [33] N. Martínez, M. Romero y E. Rodríguez, «Materiales Luminiscentes aplicados en Técnicas de Diagnóstico y Tratamiento Médico,» *Journal of Bioprocess and Chemical Technology*, vol. 9, págs. 11-18, 2017.
- [34] S. Martínez y R. Sisto, «Control y Comportamiento de Robots Omnidireccionales,» *Universidad de la República*, 2009.
- [35] H. Menzella, «Procesos Biotecnológicos II Crecimiento microbiano. Crecimiento Microbiano-Esterilización-Tipos de Biorreactores,» *Universidad de la República*, págs. 11-20, 2016.
- [36] I. Mohamed, M. Jitka y A. Olivier, «Evaluating the impact of LED bulb development on the economic viability of ultraviolet technology for disinfection.,» *Environmental Technolog*, vol. 35, págs. 400-406, 2017.
- [37] J. Montero Robalino, «Aplicación de métodos de identificación de microorganismos patógenos humanos en el banco de cepas celulolíticas del laboratorio de investigación de la Facultad de Ingeniería Química de la Universidad Central del Ecuador,» *UNIVERSIDAD CENTRAL DEL ECUADOR*, vol. 1, 2019.
- [38] J. Ortega Noroña y L. Yapo Pillajo, «CONSTRUCCIÓN DE UN ROBOT MÓVIL HÍBRIDO OMNIDIRECCIONAL,» *Universidad Politecnica Salesiana Sede Quito*, 2017.
- [39] D. Oviedo, J. Rojas, R. Borda y M. Durango, «Efecto de la exposición a la luz ultravioleta uv-c en la viabilidad de especies de Escherichia coli y Salmonella typhimurium,» vol. 2, págs. 18-28, 2013.
- [40] J. Pérez, «Diseño e implementación de un robot omnidireccional para fines de entretenimiento,» págs. 18-28, 2020.
- [41] Prometec, «SENSORES DE DISTANCIA INFRARROJOS,» 2018.
- [42] B. Radial, S. Social y L. Flores, «Manual de Programación en Lenguaje C++,» *Revista Técnica de La Facultad de Ingeniería Universidad Del Zulia*, vol. 3, págs. 232-234, 2001.
- [43] E. Reyes, «Tipos de sensores,» *Blog*, 2007.
- [44] M. Rojas, E. Mejía, R. Alcívar y col., «Caracterización clínico-demográfica y resistencia bacteriana de las infecciones del tracto urinario en el Hospital Básico de Paute, Azuay - Ecuador,» *Archivos Venezolanos de Farmacología y Terapéutica*, vol. 2, págs. 1-5, 2018.
- [45] M. Rosero, «Monitoreo microbiológico del aire, superficies y personal del Hospital del Día de la Universidad Central del Ecuador,» *Angewandte Chemie International Edition*, vol. 6, págs. 951-952, 2019.
- [46] G. Ruíz, «Procedimiento de limpieza y desinfección de los termómetros,» *Medwave*, 2017.
- [47] N. M. SAC, «DRIVER PUENTE H L298N 2A,» 2021.
- [48] O. mundial de la Salud, «Prevención Y Control De Infecciones Asociadas a la Atención de la Salud,» *Organización Panamericana de la Salud*, vol. 51, 2018.
- [49] S. Sanz, «Prácticas de microbiología,» págs. 23-26, 2011.
- [50] G. Smith, «¿Qué es un Sensor y Qué Hace?,» 2020.
- [51] Solectro, «Curso de Raspberry Pi desde cero,» 2020.
- [52] O. I. del Trabajo, «Seguridad y salud en el centro del futuro del trabajo. Aprovechar 100 años de experiencia. In Sistema de Gestión,» 2019.

- [53] S. Tzafestas, «Mobile Robot Control V. Introduction to Mobile Robot Control,» págs. 319-384, 2014.
- [54] I. Vaca Heredia y L. Villagómez Castillo, «DISEÑO Y CONSTRUCCIÓN DE UN ROBOT INTERACTIVO Y DIDÁCTICO PARA LA EMPRESA "IPREX",» 2017.
- [55] E. VEOLIA, «Análisis microbiológico,» 2020.

APÉNDICE A
PIEZAS DE LA ESTRUCTURA

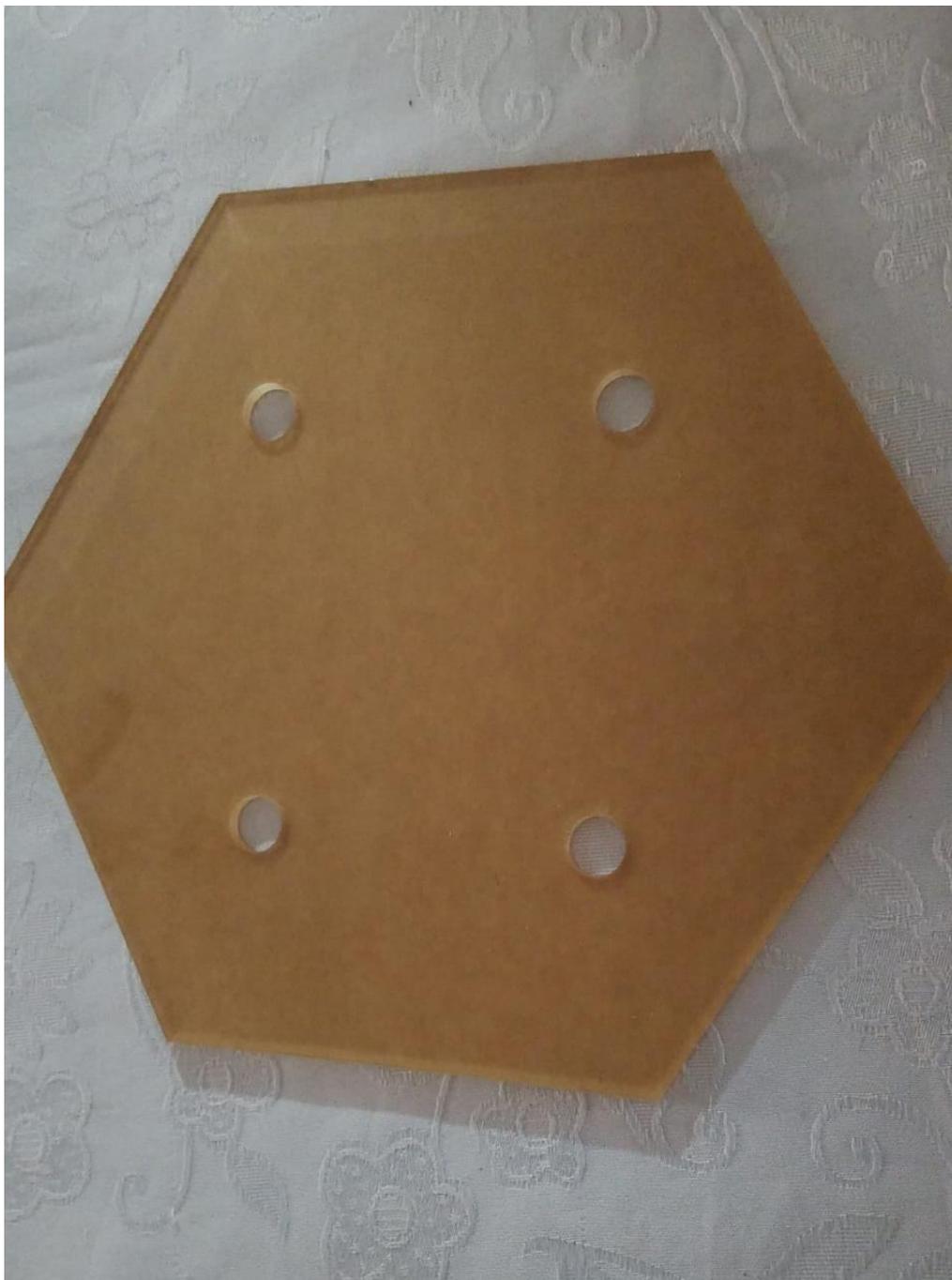


Figura 54. Estructura de acrílico para aislar las placas



Figura 55. chasis de la estructura del robot



Figura 56. Ruedas omnidireccionales impresas en 3D



Figura 57. Soporte de luces UV y tarjetas de control



Figura 58. Tapa perteneciente a la estructura del chasis.

APÉNDICE B
DISEÑO ELECTRÓNICO

PIN	Nombre	Descripción
1	EN 1 y 2	Habilitación de los Canales 1 y 2
2	1 A	Entrada Canal 1
3	1 Y	Salida Canal 1
4	GND	Tierra o Masa
5	GND	Tierra o Masa
6	2 Y	Salida Canal 2
7	2 A	Entrada Canal 2
8	VCC 2	Alimentación del Motor
9	EN 3 y 4	Habilitación de los Canales 3 y 4
10	3 A	Entrada Canal 3
11	3 Y	Salida Canal 3
12	GND	Tierra o Masa
13	GND	Tierra o Masa
14	4 Y	Salida Canal 4
15	4 A	Entrada Canal 4
16	VCC 1	Alimentación del Circuito

Figura 59. Conexión de pines de driver L293D

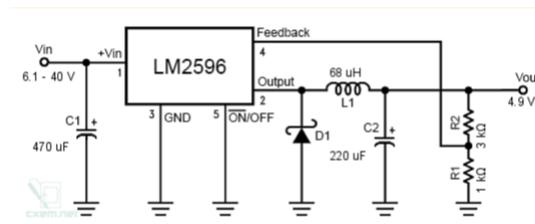


Figura 60. Diagrama de conexión interno de regulador Lm2596

Cuadro X
CARACTERÍSTICAS DE DE REGULADOR LM2596 [47]

Voltaje de entrada	4.5V a 40V DC
Voltaje de salida	1.23V a 37V DC (ajustable)
Corriente de Salida	máx. 3A
Potencia de salida	25W
Eficiencia de conversión	92%
Protección de sobre-temperatura	SI (apaga la salida)
Protección de corto circuito	SI (hasta 5A)
Protección limitadora de corriente	SI
Protección frente a inversión de polaridad	NO

Cuadro XI
COMPARACIÓN DE MOTORES ENCODER CON DISTINTOS RPM

Motor	DC 140 RPM	DC 300 RPM	DC 100 RPM
Tipo de motor	DC Potencia media	DC Potencia alta	DC Potencia media baja
Tamaño	73 mm	73 mm	73 mm
Peso	97.5 g	97.5 gr	97.5 gr
Corriente	1.2 A	3.2 A	3 A
Tensión	3.3~24.0V	3.3~24.0V	3.3~24.0V
Torque	8Kgf*cm	11Kgf*cm	6.5 Kgf*cm
Potencia de salida	25 W	28.2 W	17 W
Costo	\$ 18	\$ 20	\$ 15

Cuadro XII
ESPECIFICACIONES DE RASPBERRY PI 3A+[23]

Procesador	Broadcom BCM2837B0 64-bit SoC @ 1.4GHz
Memoria RAM	512MB LPDDR2 SDRAM
WiFi	2.4GHz y 5GHz 802.11.b/g/n/ac
Bluetooth	4.2/BLE
Entradas digitales	40 pines GPIO
Conectores	HDMI, USB 2.0, PUERTO CSI, PUERTO DSI, JACK 3.5mm, PUERTO MICRO SD
Alimentación	5V/2.5A DC

Cuadro XIII
ABREVIATURAS DE DIAGRAMA DE FUERZA

-F	Fusible
S	Interruptor
RG	Regulador de voltaje
DRV	Driver de motor
PRO	Procesadores
COM	Componentes (luces UV, módulos)
M	Motor DC

Cuadro XIV
 CARACTERÍSTICAS DE DRIVER L293D [2]

Alimentación	45 a 36 VDC
Corriente de salida	600 mA.
Corriente pico de salida	1 A por canal (no repetitiva)
Encapsulado	DIP de 16 pines
Inmunidad	Alta inmunidad al ruido eléctrico
Protección	Protección contra exceso de temperatura

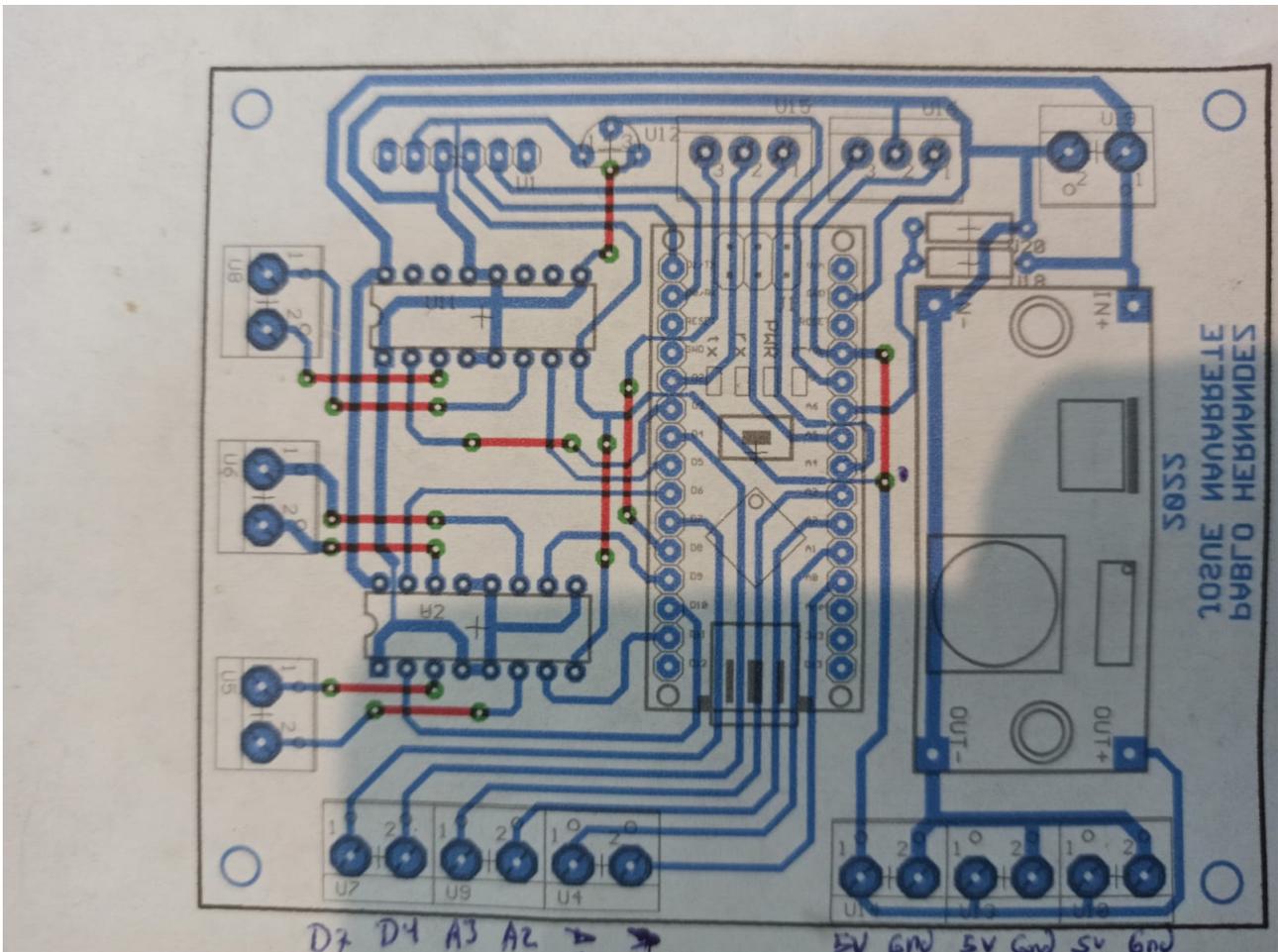


Figura 61. Cara inferior de tarjeta electrónica

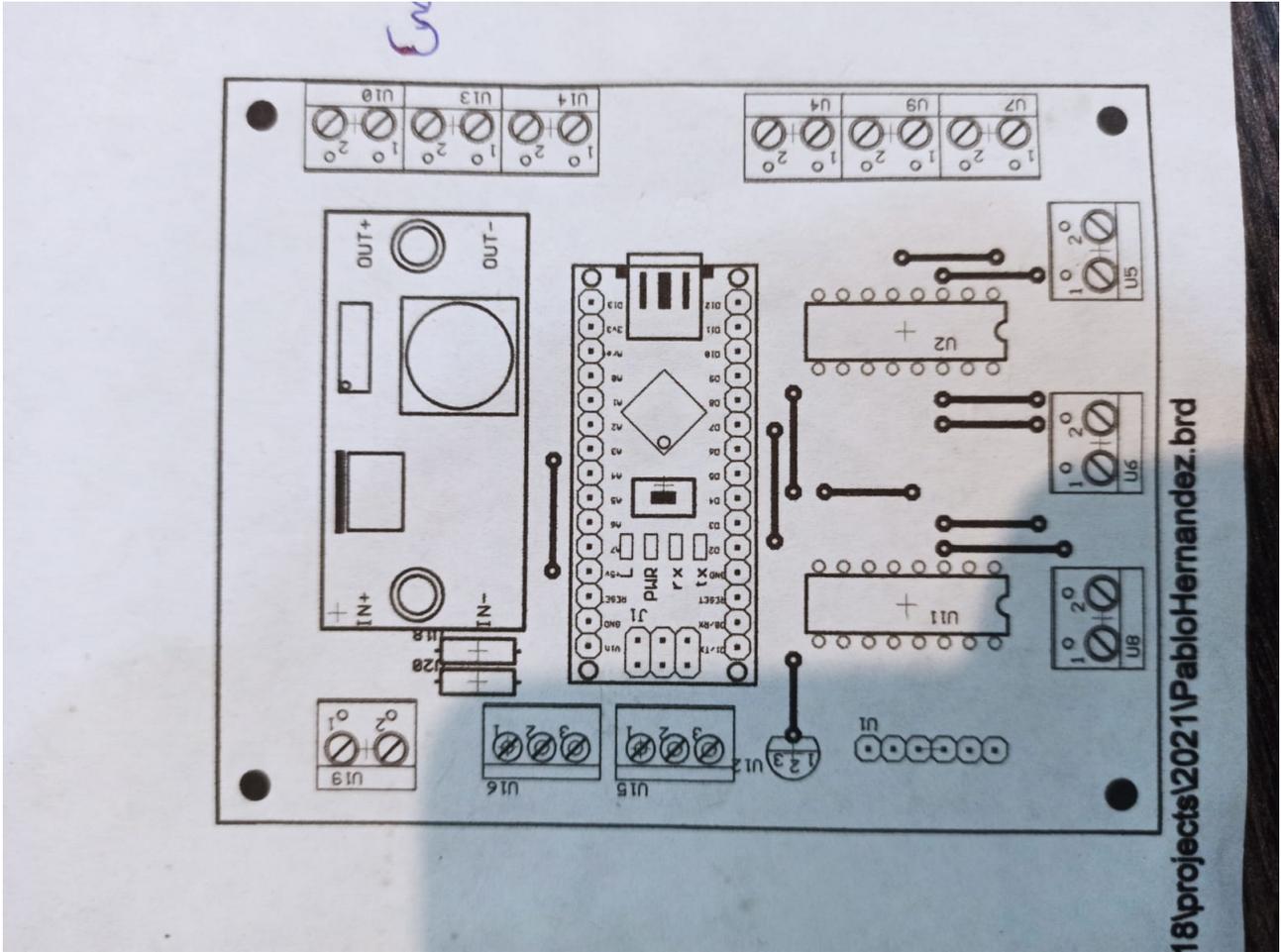


Figura 62. Cara superior de tarjeta electrónica

APÉNDICE C
PROCESOS DE MONTAJE



Figura 63. Proceso de corte de base de acrílico



Figura 64. CNC láser realizando cortes de piezas



Figura 65. Proceso de impresión 3D



Figura 66. Plancha de aluminio destinada a ensamblaje del robot



Figura 67. Montaje de tapa ubicada en la parte superior del chasis

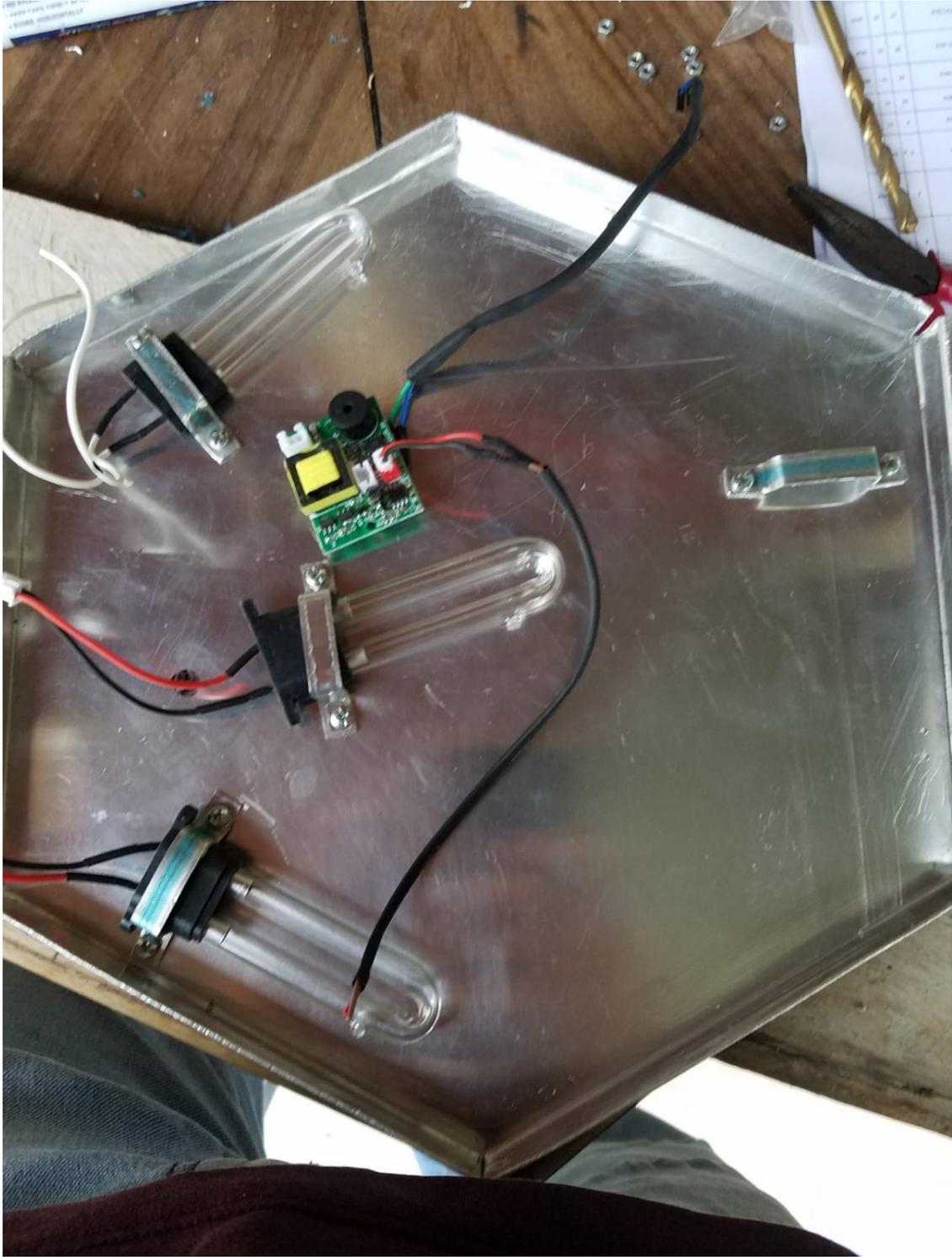


Figura 68. Ubicación de lamparas UV dentro de la estructura



Figura 69. Material destinado para la estructura del robot

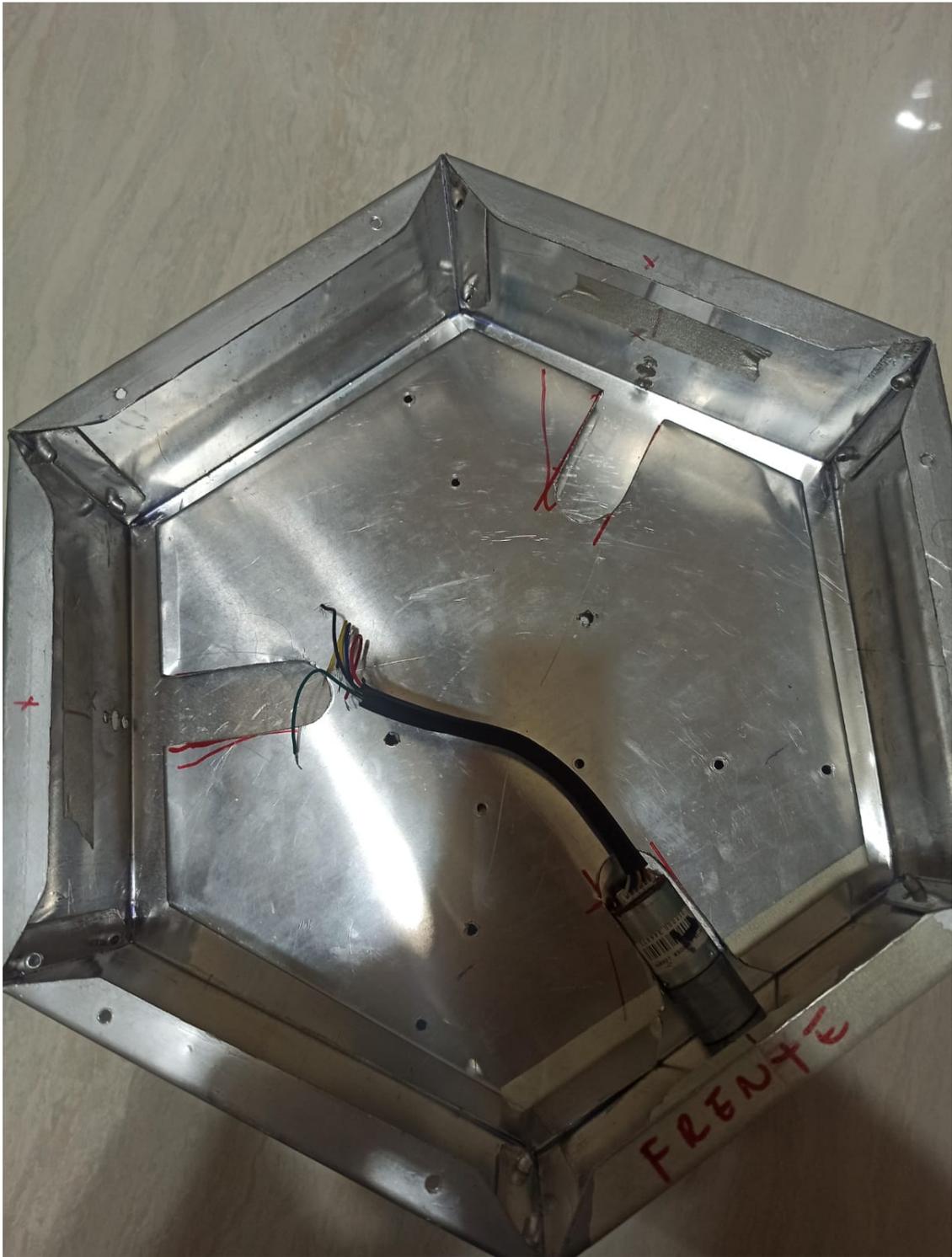


Figura 70. Ubicación de motores encoder en estructura del robot

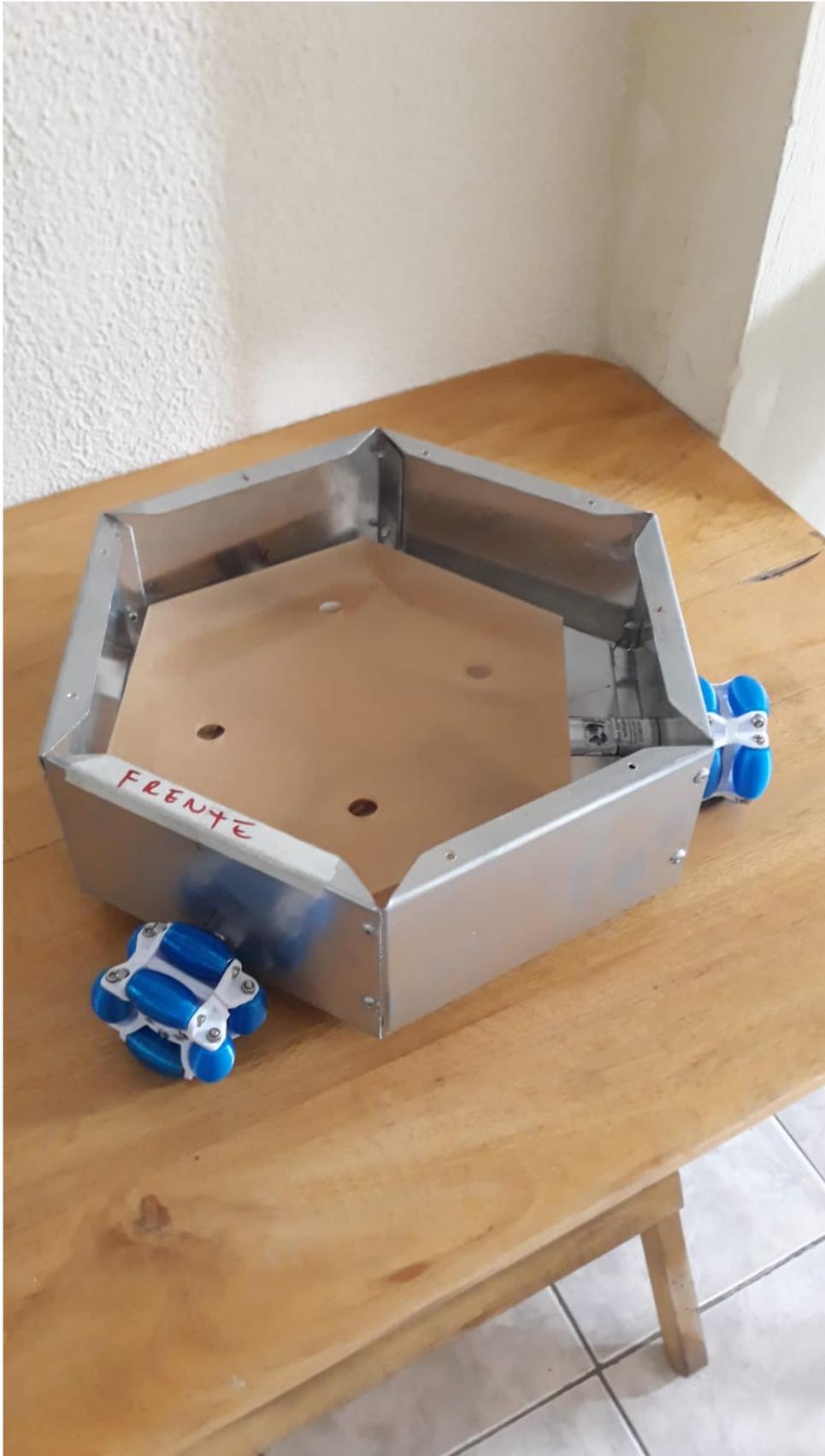


Figura 71. Montaje de ruedas omnidireccionales en prototipo



Figura 72. Montaje de soporte de luces UV y tarjetas de control de lamparas

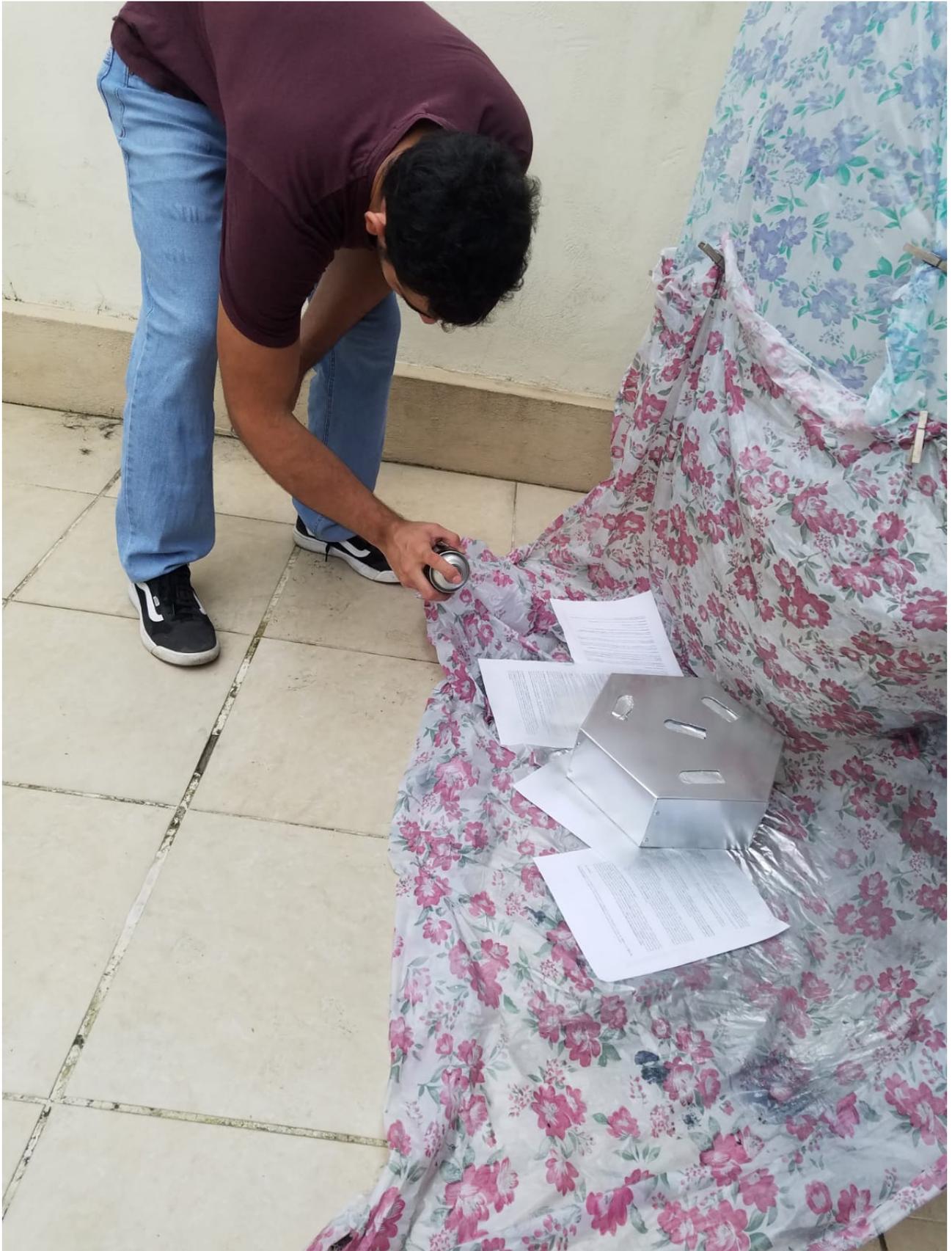


Figura 73. Proceso de pintado de chasis de la estructura

APÉNDICE D
PLANOS DE LA ESTRUCTURA

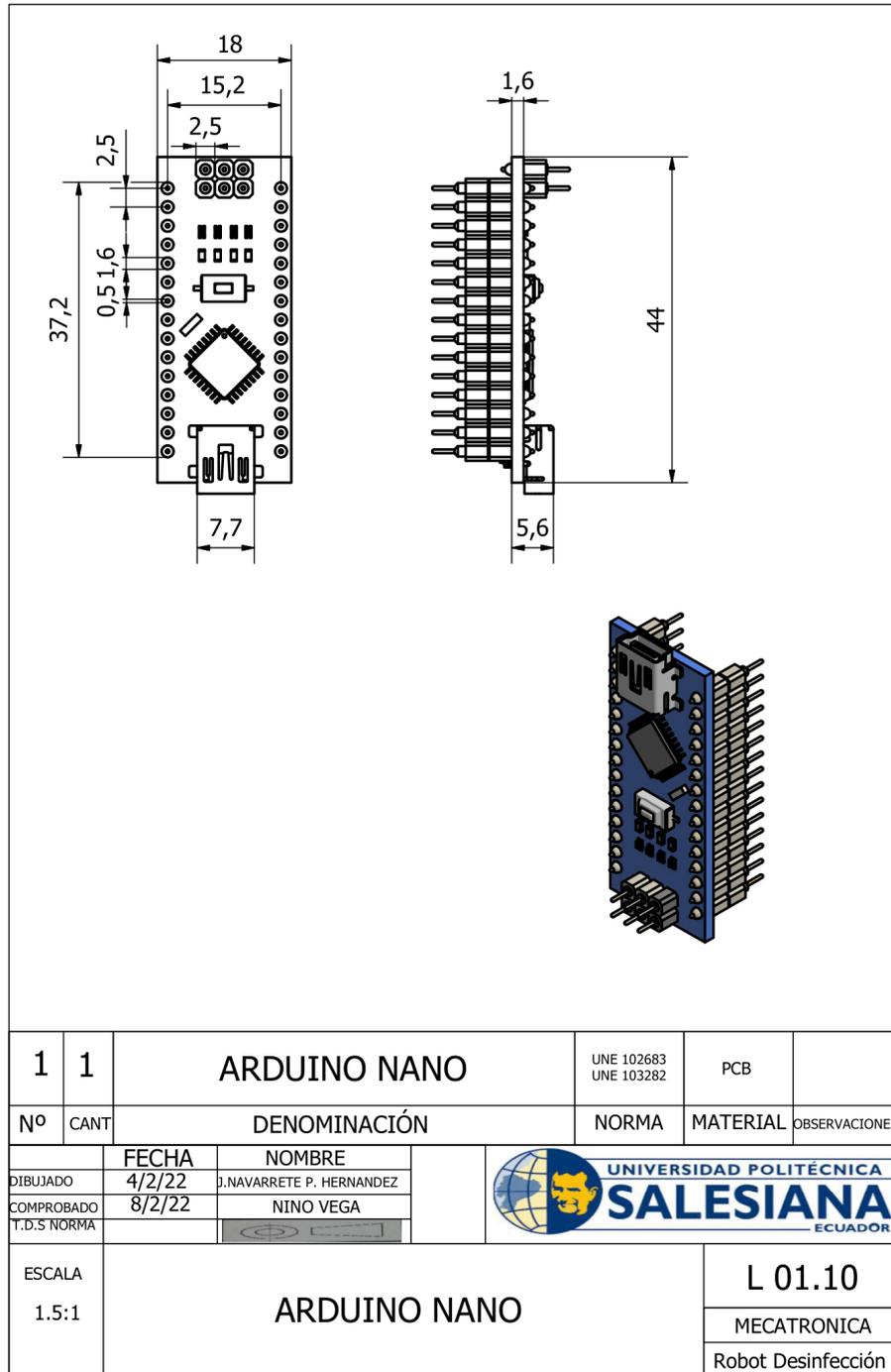
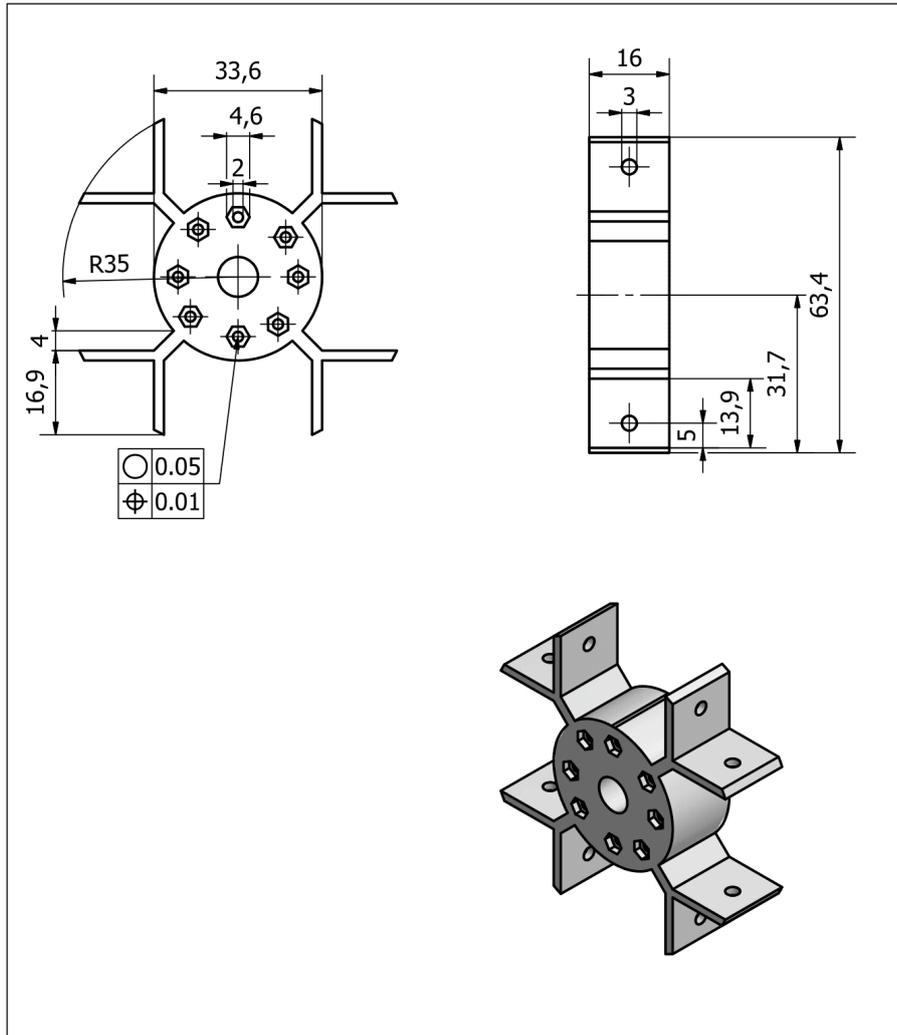
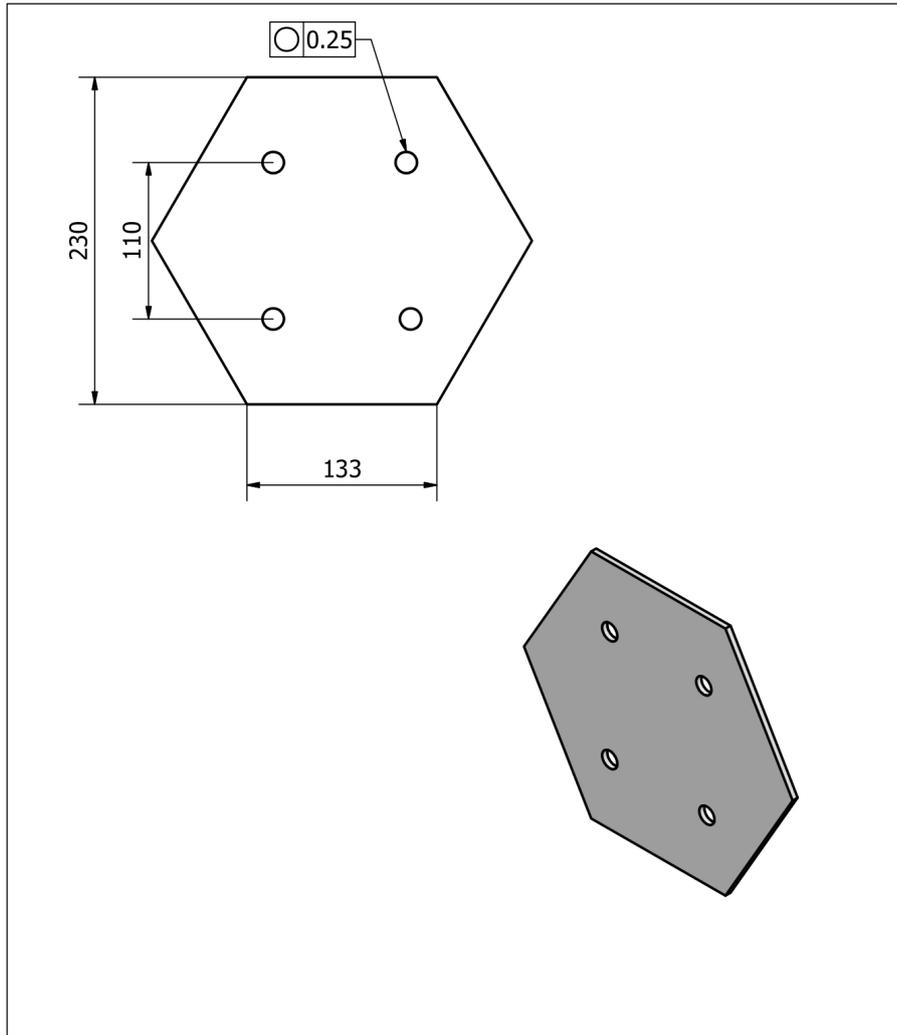


Figura 74. Plano de Tapa de chasis



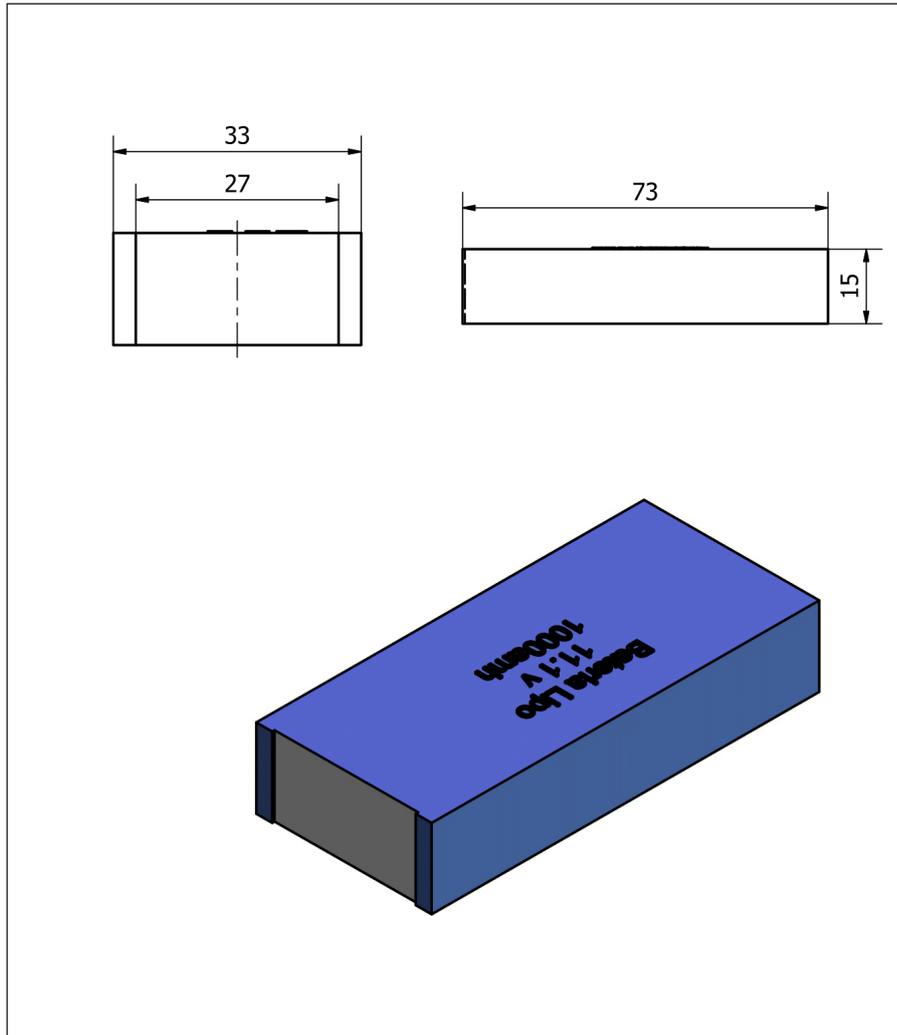
1	4	ARO DE LA LLANTA		UNE 102683 UNE 103282 ISO 2768	ALUMINIO	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	D.NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	ARO DE LA LLANTA				L 01.16	
1:1					MECATRONICA	
					Robot Desinfección	

Figura 75. Plano de soporte principal de ruedas omnidireccionales



1	1	BASE DE ACRÍLICO		UNE 102683 UNE 103282 ISO 2768 C	ACRÍLICO	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	J. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	BASE DE ACRÍLICO				L 01.03	
1:3.5					MECATRONICA	
					Robot Desinfección	

Figura 76. Plano de base de acrílico



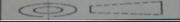
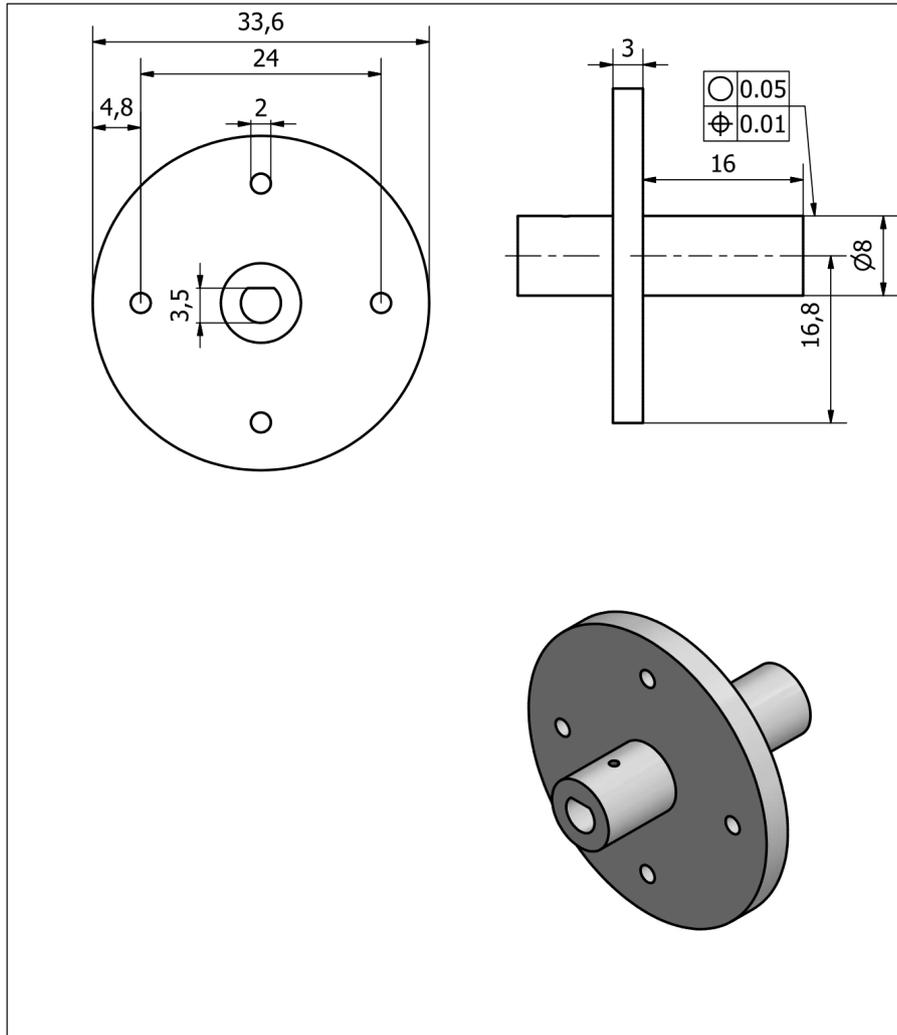
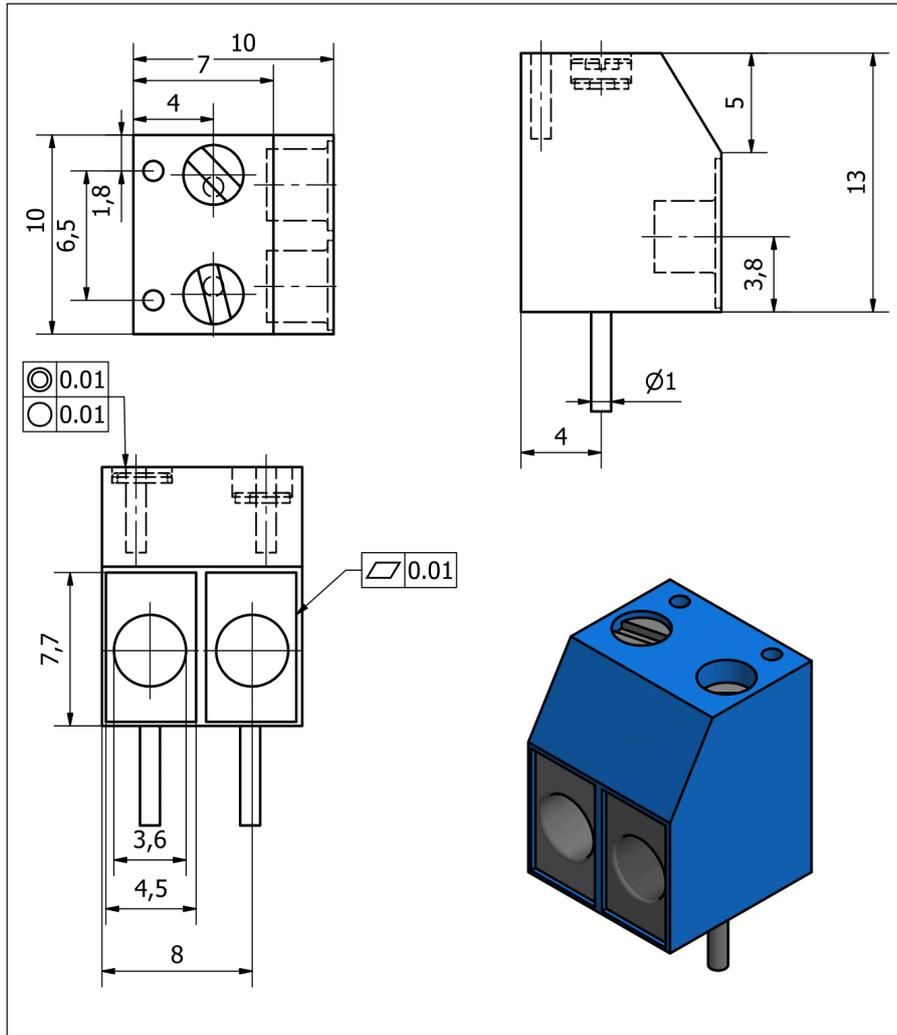
1	1	BATERIAS LIPO 11.V		UNE 102683 UNE 103282	Ion litio	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	D.NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	BATERIAS LIPO 11.V				L 01.06	
1.5:1					MECATRONICA	
					Robot Desinfección	

Figura 77. Dimensiones de batería lipo 11.1V



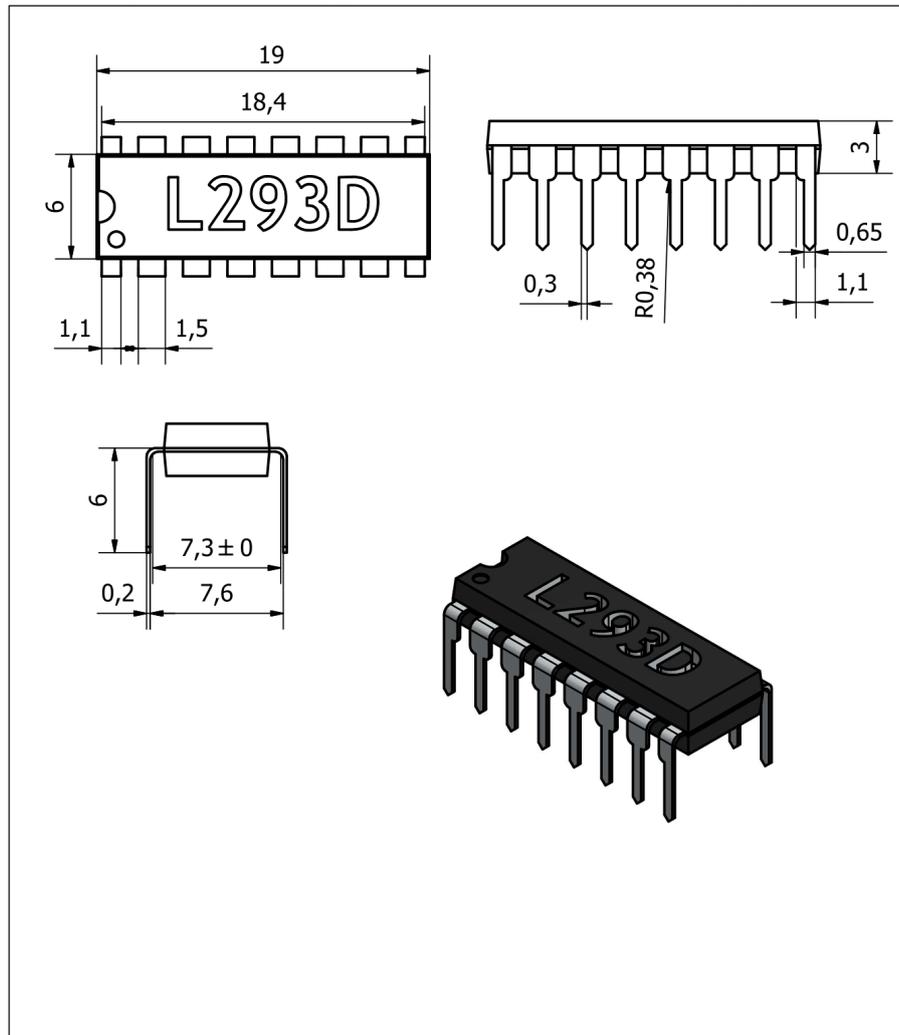
1	4	BOCIN DE LLANTAS		UNE 102683 UNE 103282 ISO 2768	PLA	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO		4/2/22	J. NAVARRETE P. HERNANDEZ			
COMPROBADO		8/2/22	NINO VEGA			
T.D.S NORMA						
ESCALA	BOCIN DE LLANTAS				L 01.08	
2:1					MECATRONICA	
					Robot Desinfección	

Figura 78. Plano de bocin para ruedas omnidireccionales



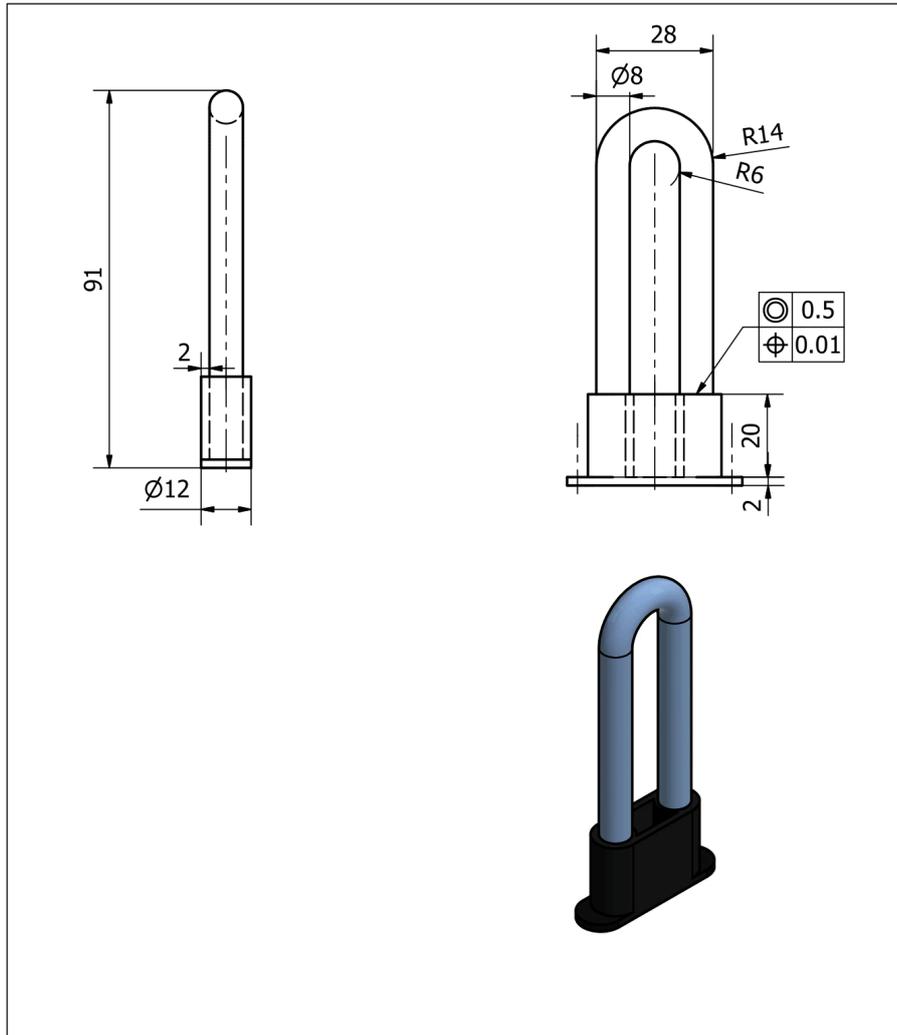
1	5	BORNERA DE DOS PINES		UNE 102683 UNE 103282 ISO 2768 C	ACERO	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO		4/2/22	J. NAVARRETE P. HERNANDEZ			
COMPROBADO		8/2/22	NINO VEGA			
T.D.S NORMA						
ESCALA	BORNERA DE DOS PINES				L 01.09	
4:1					MECATRONICA	
					Robot Desinfección	

Figura 79. Dimensiones de borneras de 2 pines



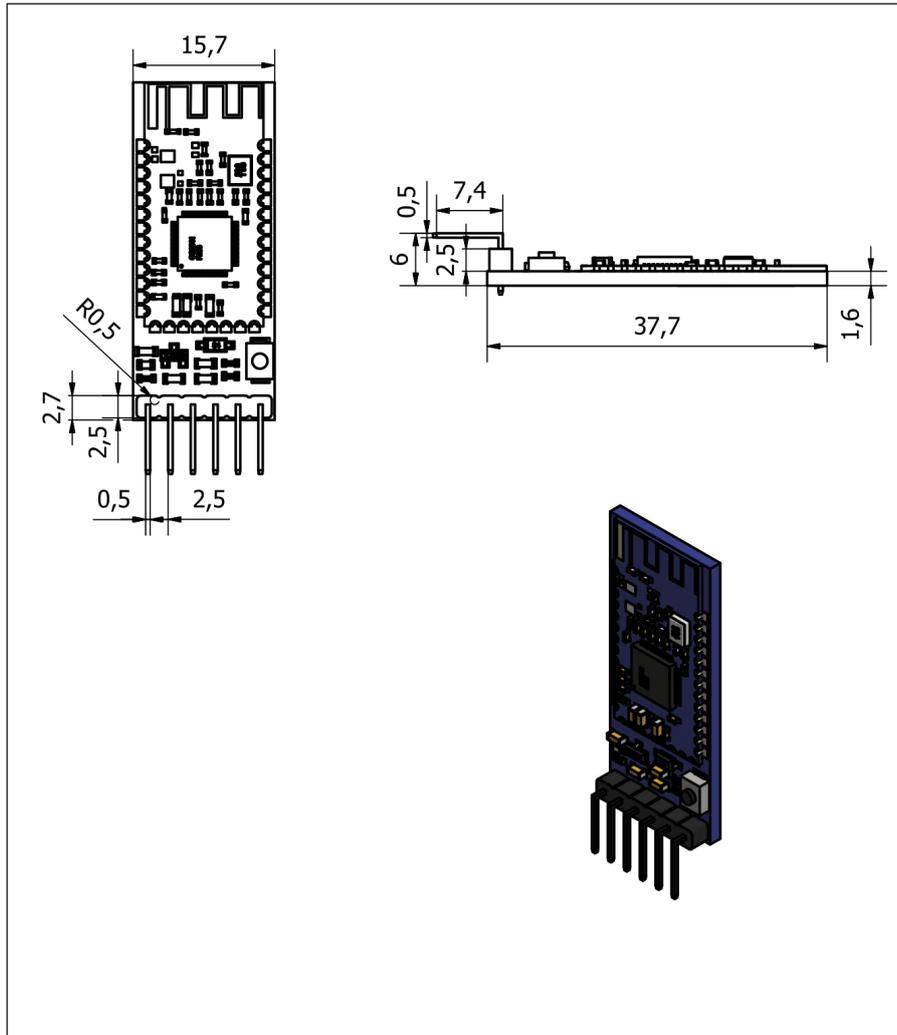
1	2	DRIVE L293D		UNE 102683 UNE 103282		
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	J. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	DRIVE L293D		L 01.12			
3.5:1			MECATRONICA			
			Robot Desinfección			

Figura 80. Dimensiones de driver L293D



1	4	LAMPARA UV		UNE 102683 UNE 103282 ISO 2768 C	Cristal Goma	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO		4/2/22	J. NAVARRETE P. HERNANDEZ			
COMPROBADO		8/2/22	NINO VEGA			
T.D.S NORMA						
ESCALA	LAMPARA UV				L 01.05	
1:1.2					MECATRONICA	
					Robot Desinfección	

Figura 81. Dimensiones de lamparas UV



1	1	MÓDULO BLUETOOTH HC-05		UNE 102683 UNE 103282	NO ESPECÍFICA	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
		4/2/22	J. NAVARRETE P. HERNANDEZ			
		8/2/22	NINO VEGA			
		COMPROBADO				
		T.D.S NORMA				
ESCALA	MÓDULO BLUETOOTH HC-05				L 01.11	
3.5:1					MECATRONICA	
					Robot Desinfección	

Figura 82. Dimensiones de modulo bluetooth HC-05

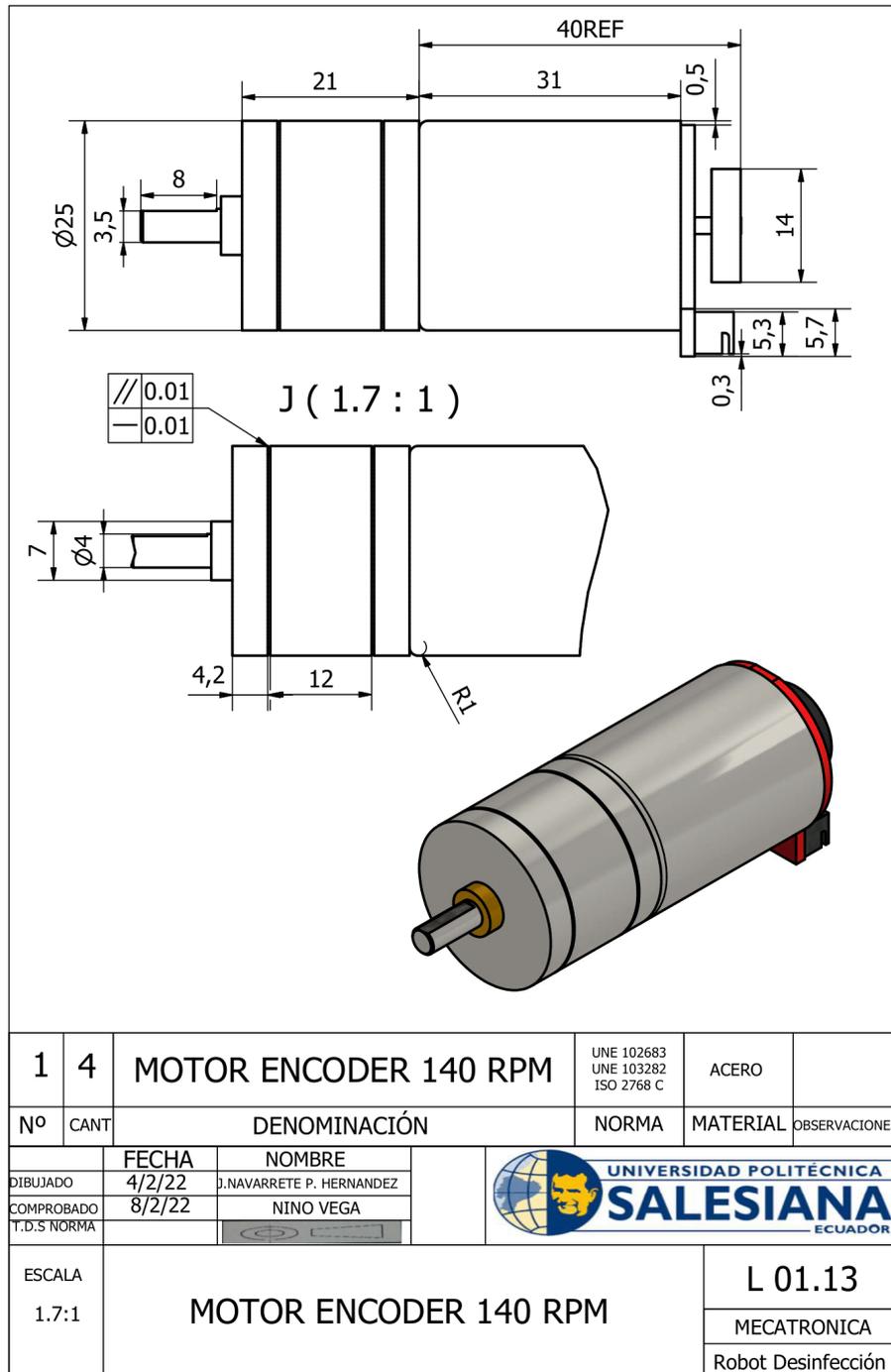
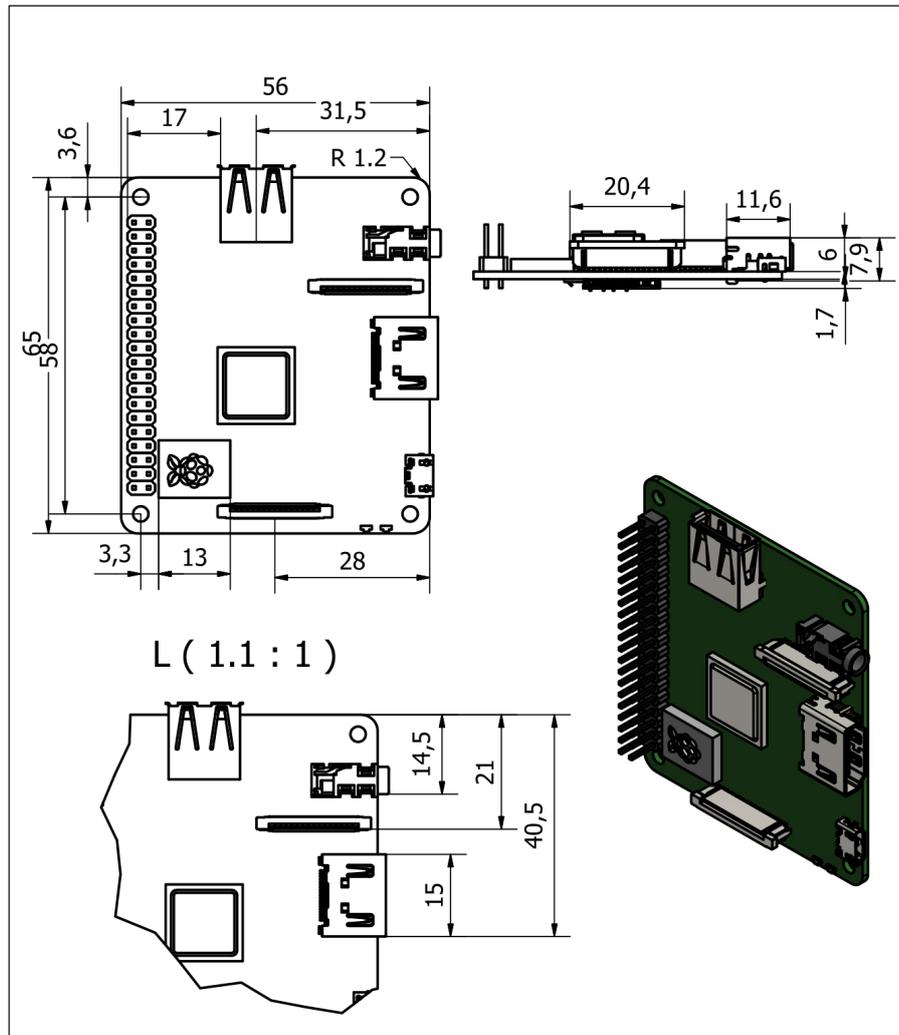
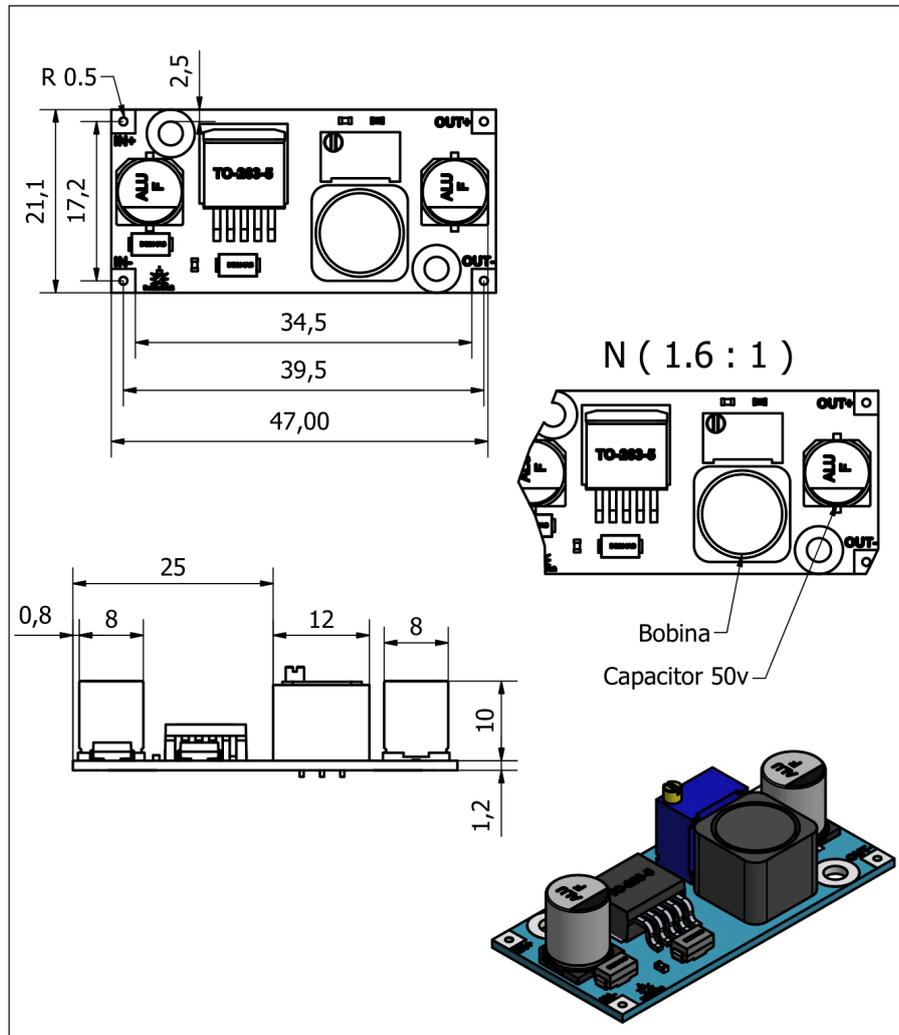


Figura 83. Dimensiones de motor encoder CHR-GM25-370



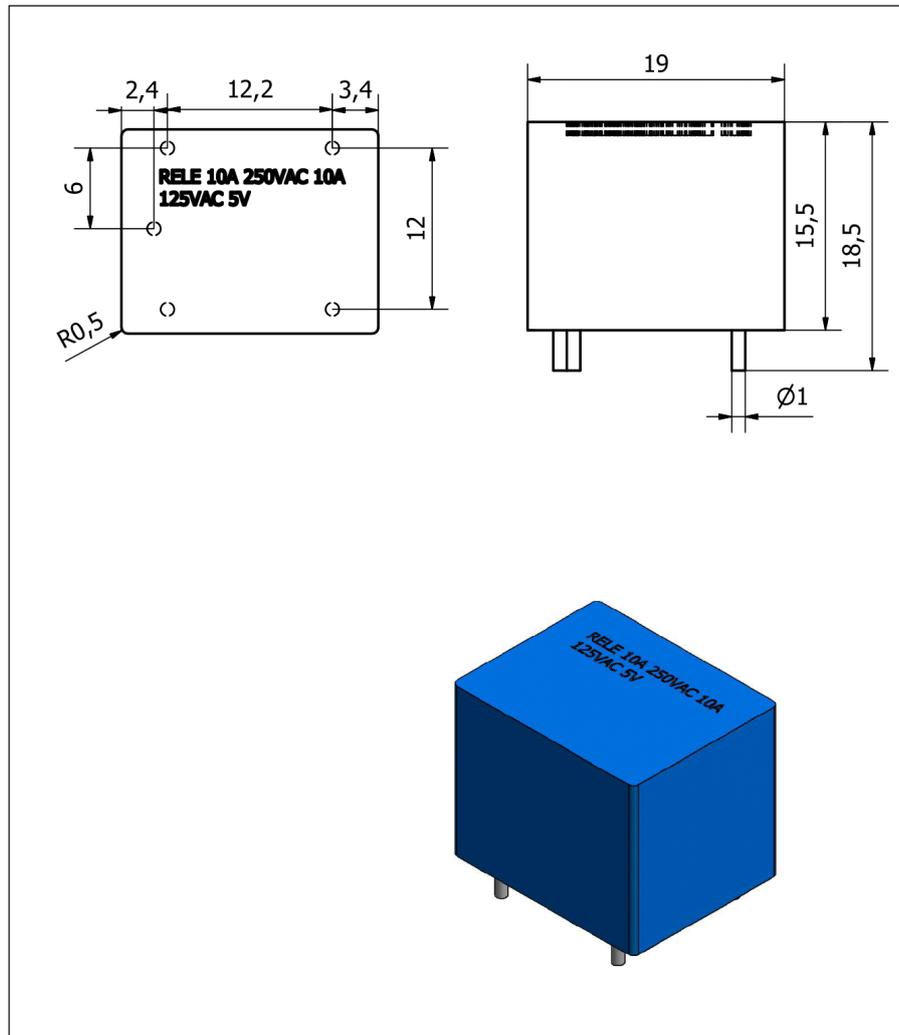
1	1	RASPBERRY PI 3 A+		UNE 102683 UNE 103282	PLACA DE PCB	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
		4/2/22	J. NAVARRETE P. HERNANDEZ			
		8/2/22	NINO VEGA			
		COMPROBADO				
		T.D.S NORMA				
ESCALA	RASPBERRY PI 3 A+				L 01.14	
1.1:1					MECATRONICA	
					Robot Desinfección	

Figura 84. Dimensiones de Raspberry Pi 3A+



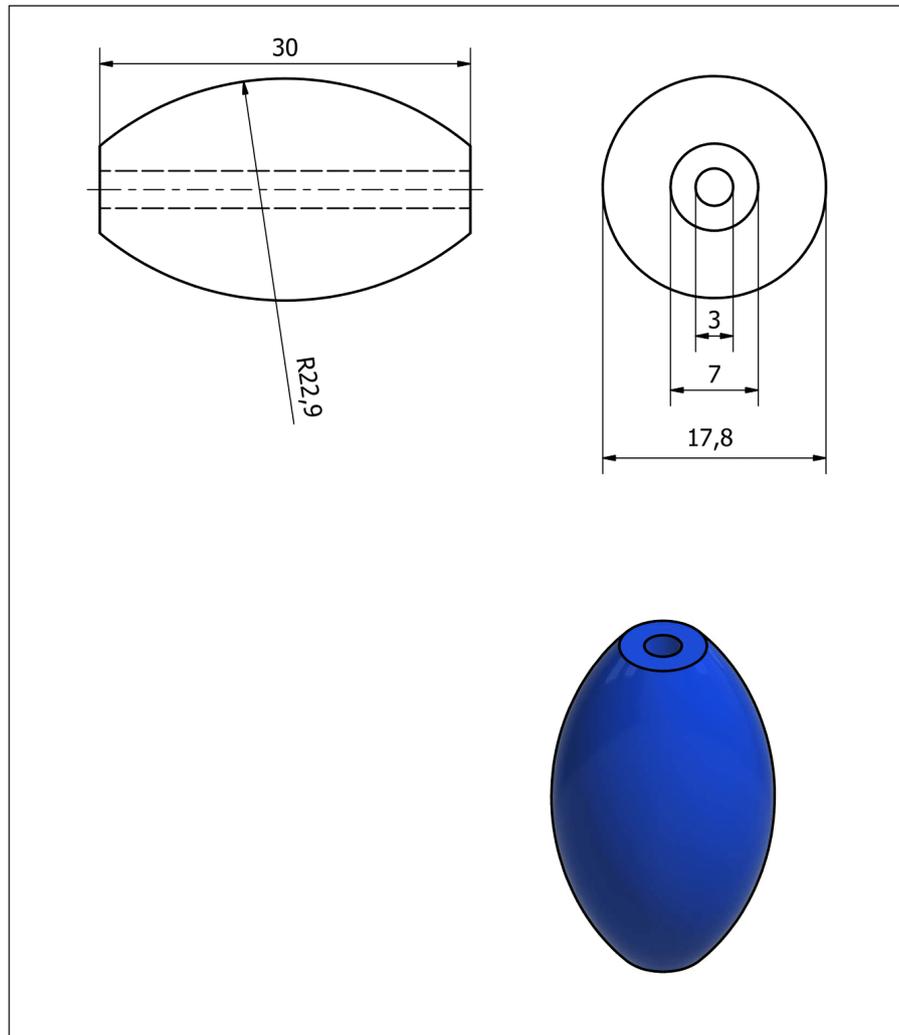
1	2	LM2596 DC-DC	UNE 102683 UNE 103282	PLACA PCB	
Nº	CANT	DENOMINACIÓN	NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE		
DIBUJADO		4/2/22	J. NAVARRETE P. HERNANDEZ		
COMPROBADO		8/2/22	NINO VEGA		
T.D.S NORMA					
ESCALA	LM2596 DC-DC			L 01.18	
1.6:1				MECATRONICA	
				Robot Desinfección	

Figura 85. Dimensiones de regulador de voltaje LM2596



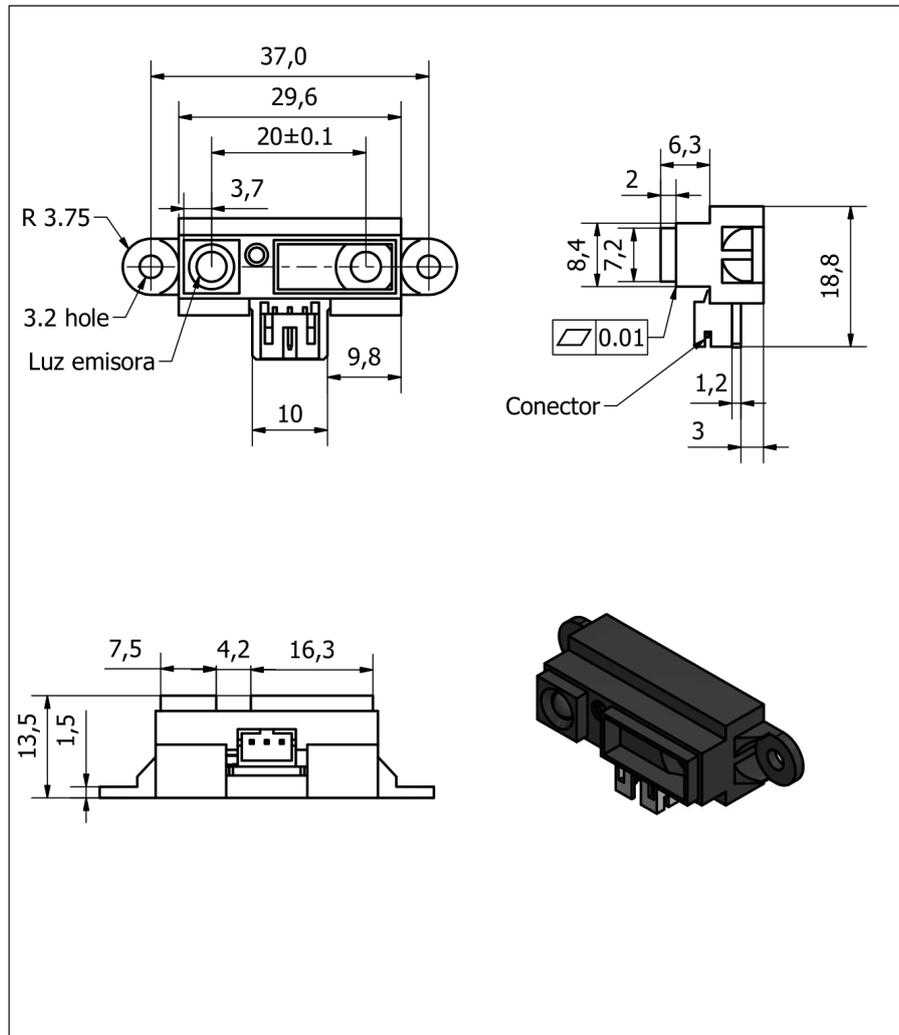
1	1	RELE 5V		UNE 102683 UNE 103282	ABS	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	J. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	RELE 5V				L 01.07	
2.7:1					MECATRONICA	
					Robot Desinfección	

Figura 86. Dimensiones de relé 5V



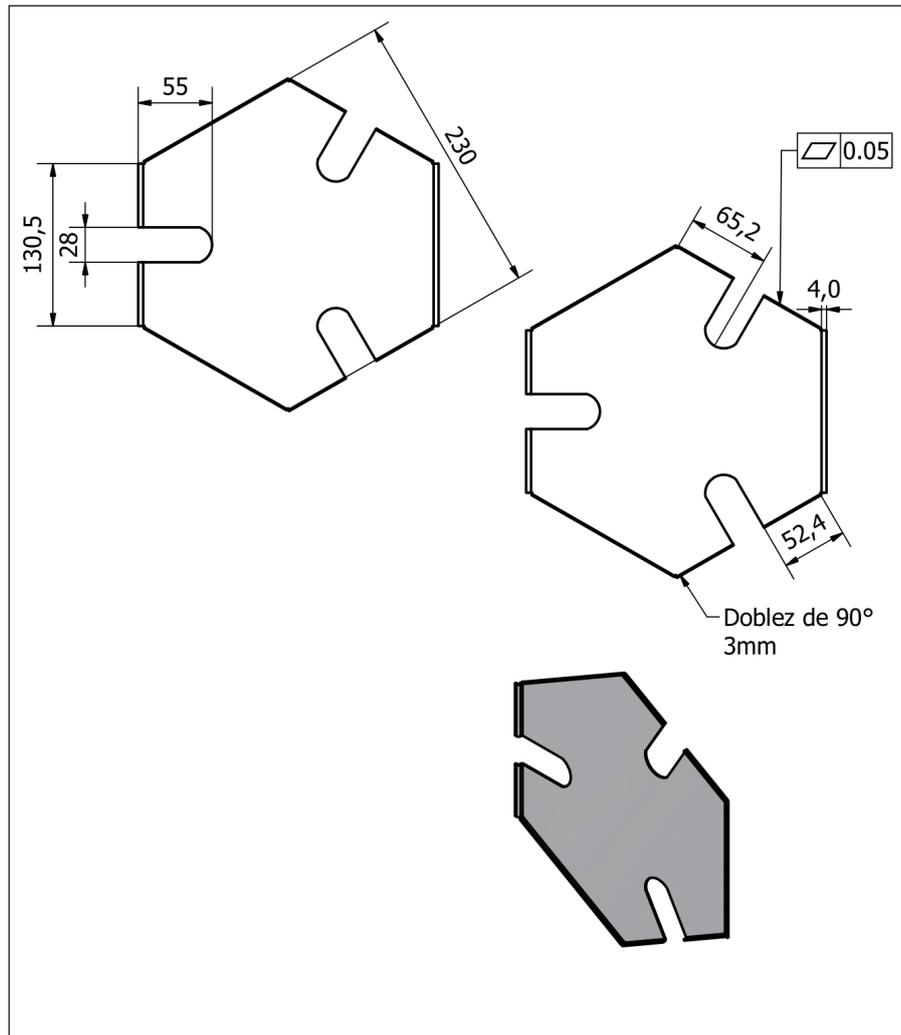
1	24	ROLLER DE LLANTAS		UNE 102683 UNE 103282 ISO 2768	TPU	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	D. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	ROLLER DE LLANTAS				L 01.15	
2.5:1					MECATRONICA	
					Robot Desinfección	

Figura 87. Dimensiones de rodillos de rueda omnidireccional



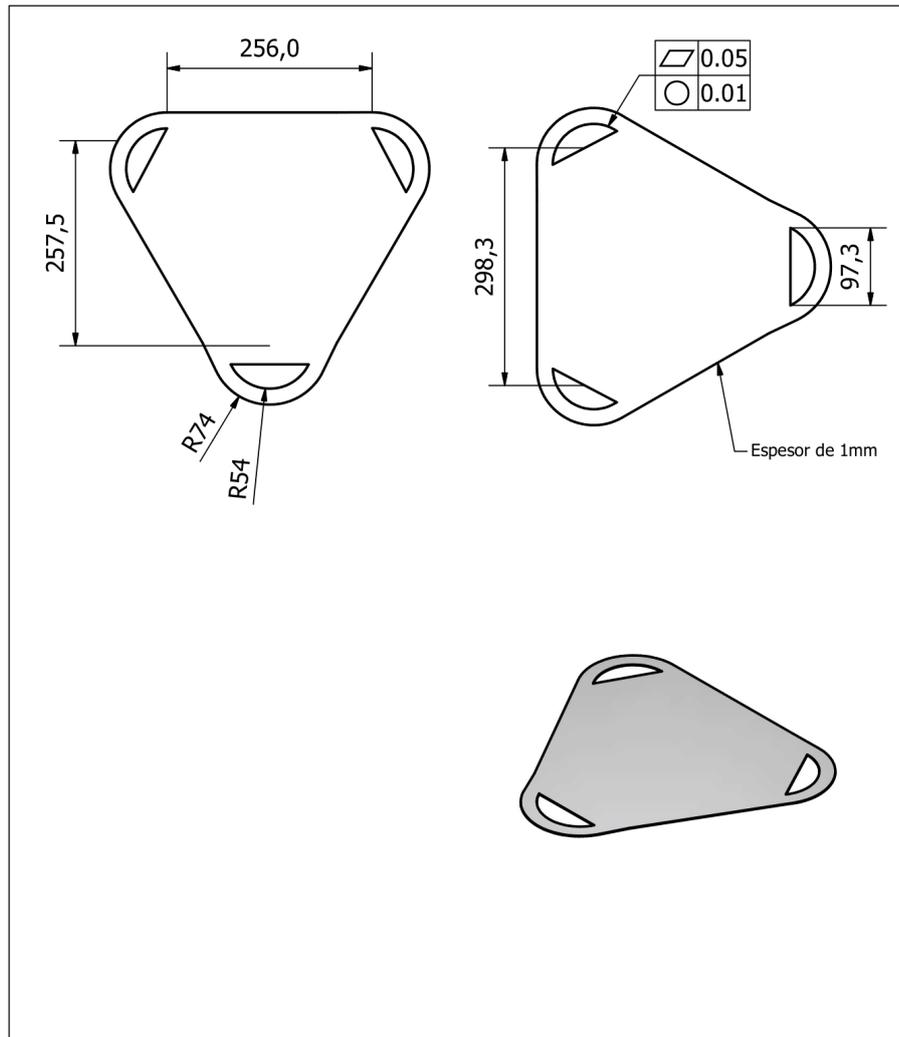
1	4	SENSOR SHARP		UNE 102683 UNE 103282 ISO 2768 C	PCB Y ABS	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
		4/2/22	J. NAVARRETE P. HERNANDEZ			
		8/2/22	NINO VEGA			
		COMPROBADO				
		T.D.S NORMA				
ESCALA	SENSOR SHARP				L 01.17	
1.5:1					MECATRONICA	
					Robot Desinfección	

Figura 88. Dimensiones de sensor IR GP2Y0A21



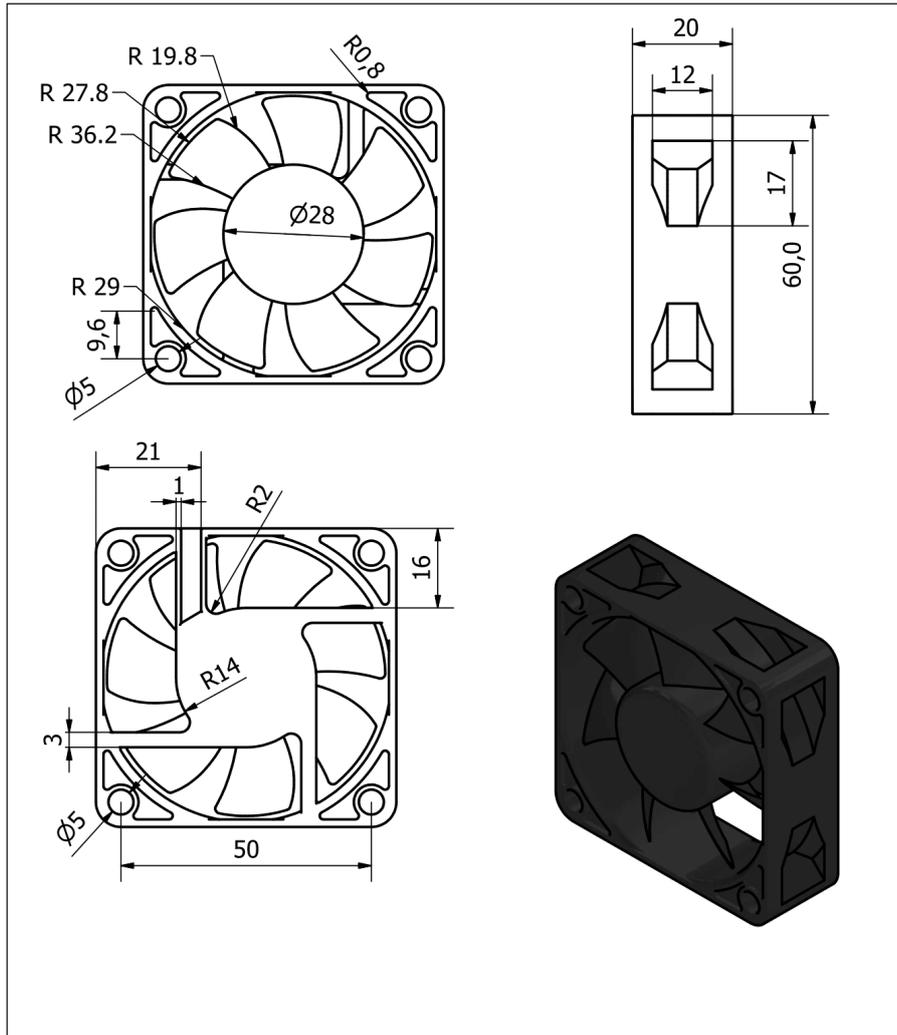
1	1	TAPA DE SENSORES		UNE 102683 UNE 103282 ISO 2768	ALUMINIO	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	D. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	TAPA DE SENSORES				L 01.04	
1:4					MECATRONICA	
					Robot Desinfección	

Figura 89. Plano de soporte de luces UV y tarjetas de control de lamparas



1	1	TAPA DEL CHASIS		UNE 102683 UNE 103282 ISO 2768 H	ALUMINIO	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	J. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	TAPA DEL CHASIS				L 01.02	
0.16:1					MECATRONICA	
					Robot Desinfección	

Figura 90. Plano de tapa del chasis de la estructura



1	1	VENTILADOR DC		UNE 102683 UNE 103282	ABS	
Nº	CANT	DENOMINACIÓN		NORMA	MATERIAL	OBSERVACIONES
		FECHA	NOMBRE			
DIBUJADO	4/2/22	J. NAVARRETE P. HERNANDEZ				
COMPROBADO	8/2/22	NINO VEGA				
T.D.S NORMA						
ESCALA	VENTILADOR DC				L 01.19	
1:1					MECATRONICA	
					Robot Desinfección	

Figura 91. Dimensiones de ventilador DC 12V

APÉNDICE E
ALGORITMO DE LECTURA QR E INTERFAZ DE MONITOREO

```
1 import tkinter
2 import time
3 from tkinter import *
4 from PIL import Image, ImageTk
5 from datetime import date
6 from datetime import datetime
7 import serial
8 import threading
9 import cv2
10 from pyzbar.pyzbar import decode
11 from picamera.array import PiRGBArray
12 from picamera import PiCamera
13 #inicializacion de puerto SERIAL
14 arduino=serial.Serial('/dev/ttyACM0',baudrate=9600, timeout = 3.0)
15 txt=''
16
17 ventana = Tk()
18 ventana.geometry("1200x800")
19 ventana.resizable(width=False, height=False)
20
21 w = Canvas(ventana, width = 500, height = 500)
22 w.place(x=350 ,y=5)
23 #Rueda de arriba
24 cir1 = w.create_oval(270, 70, 290, 90)
25 cir2 = w.create_oval(200, 70, 220, 90)
26 #Rueda izquierda
27 cir3 = w.create_oval(70, 280, 90, 300)
28 cir4 = w.create_oval(70, 340, 90, 360)
29 #Rueda derecha
30 cir5 = w.create_oval(400, 340, 420, 360)
31 cir6 = w.create_oval(400, 280, 420, 300)
32
33 #FUNCIONES
34
35 #Lectura serial
36 def lectura_serial():
37     while True:
38         lec=arduino.readline().decode()
39         #print(lec)
40
41         recv=lec.split('\r')
42         #print(recv[0])
```

Figura 92. Código de interfaz y programas multihilos desarrollado en python

```

43
44     if recv[0]=="A":
45         indi_offf()
46         indi_gir_h()
47     if recv[0]=="B":
48         indi_offf()
49         indi_gir_ah()
50     if recv[0]=="C":
51         indi_offf()
52         indi_up()
53     if recv[0]=="D":
54         indi_down()
55     else:
56         if recv[0].isnumeric()==1:
57             if int(recv[0])>11:
58                 valor_temp = Label(ventana, text =lec,font=("Helvetica",40))
59                 valor_temp.place(x=150, y=550)
60 #Lectura de codigo QR
61 def lectura_qr():
62     # Inicializar la cámara
63     camera = PiCamera()
64     camera.resolution = (640, 480)
65     camera.framerate = 32
66     rawCapture = PiRGBArray(camera)
67
68     # Load buffer
69     time.sleep(0.1)
70     recieved_data = None
71
72     # Obtener fotos
73     for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
74         image = frame.array
75         # reading frame from the camera
76         # Decoding the QR Code
77         decoded_data = decode(image)
78
79         try:
80             data = decoded_data[0][0]
81             sala=data.decode()
82             if data != recieved_data:
83                 recieved_data = data
84

```

Figura 93. Código de interfaz y programas multihilos desarrollado en python

```

85         #print(data)
86         #print("\n", data, "\n")
87         #Decode sirve para cambiar la variable de byte a cadena
88         sala=data.decode()
89         print(sala)
90         arduino.write('m'.encode('ASCII'))
91         print("modo manual enviado")
92         time.sleep(1)
93         if sala=='https://consultorio.ec':
94             arduino.write('1'.encode('ASCII'))
95             print("numero 1 enviado")
96             time.sleep(2)
97         if sala=='https://habuno.ec':
98             arduino.write('3'.encode('ASCII'))
99             print("numero 3 enviado")
100            time.sleep(0.5)
101            sala = None
102
103
104    except:
105        pass
106        # Mostrar
107        cv2.imshow("Frame", image)
108        key = cv2.waitKey(1) & 0xFF
109        # Vaciar el caché
110        rawCapture.truncate(0)
111
112
113        # Si presiona la letra 'q' del teclado, salga del bucle
114        if key == ord("q"):
115            break
116
117    #INDICADOR HORARIO
118    def indi_gir_h():
119
120        w.itemconfig(cir1, fill = "blue")
121        w.itemconfig(cir3, fill = "blue")
122        w.itemconfig(cir5, fill = "blue")
123
124    #INDICADOR ANTIHORARIO
125    def indi_gir_ah():
126        w.itemconfig(cir2, fill = "red")

```

Figura 94. Código de interfaz y programas multihilos desarrollado en python

```

127     w.itemconfig(cir4,fill = "red")
128     w.itemconfig(cir6,fill = "red")
129
130 #INDICADOR ARRIBA
131 ▼ def indi_up():
132     w.itemconfig(cir3, fill = "blue")
133     w.itemconfig(cir6,fill = "blue")
134
135
136 #INDICADOR ABAJO
137 ▼ def indi_down():
138     w.itemconfig(cir4, fill = "red")
139     w.itemconfig(cir5,fill = "red")
140
141
142
143
144 #RESET DE INDICADORES
145 ▼ def indi_off():
146     w.itemconfig(cir1, fill = "gray")
147     w.itemconfig(cir2,fill = "gray")
148     w.itemconfig(cir3,fill = "gray")
149     w.itemconfig(cir4, fill = "gray")
150     w.itemconfig(cir5,fill = "gray")
151     w.itemconfig(cir6,fill = "gray")
152
153 #CREAR FUNCION PARA APAGAR LEDS
154
155 robot_img= PhotoImage(file = "llantas_robot_2.png")
156 logo_robot = Label(ventana,image=robot_img)
157 logo_robot.place(x=450, y=100) #p
158
159 ubi_destino= PhotoImage(file = "lol.png")
160 img_ubi = Label(ventana,image=ubi_destino)
161 img_ubi.place(x=850, y=350) #p
162
163
164 logo_img= PhotoImage(file = "logo_meca_trans.png")
165 logo_ups = Label(ventana,image=logo_img)
166 logo_ups.place(x=1000, y=0) #p
167
168 #flecha_up_vec= PhotoImage(file = "flecha_ar.png")

```

Figura 95. Código de interfaz y programas multihilos desarrollado en python

```

169 #flecha_up = Label(ventana,image=flecha_up_vec)
170 #flecha_up.place(x=550, y=400) #p
171
172
173
174
175 titulo_temp = Label(ventana, text = " Temperatura ",font=("Helvetica",40))
176 titulo_temp.place(x=75, y=475)
177
178 today = date.today()
179 fecha_var = Label(ventana, text =today ,font=("Helvetica",20))
180 fecha_var.place(x=10, y=10)
181
182 #Linea para iniciar un hilo
183 ini_lectura = threading.Thread(target=lectura_serial)
184 ini_lectura.start()
185
186 #Linea para lectura de codigo QR
187 qr_cod = threading.Thread(target=lectura_qr)
188 qr_cod.start()
189 ventana.mainloop()

```

Figura 96. Código de interfaz y programas multihilos desarrollado en python

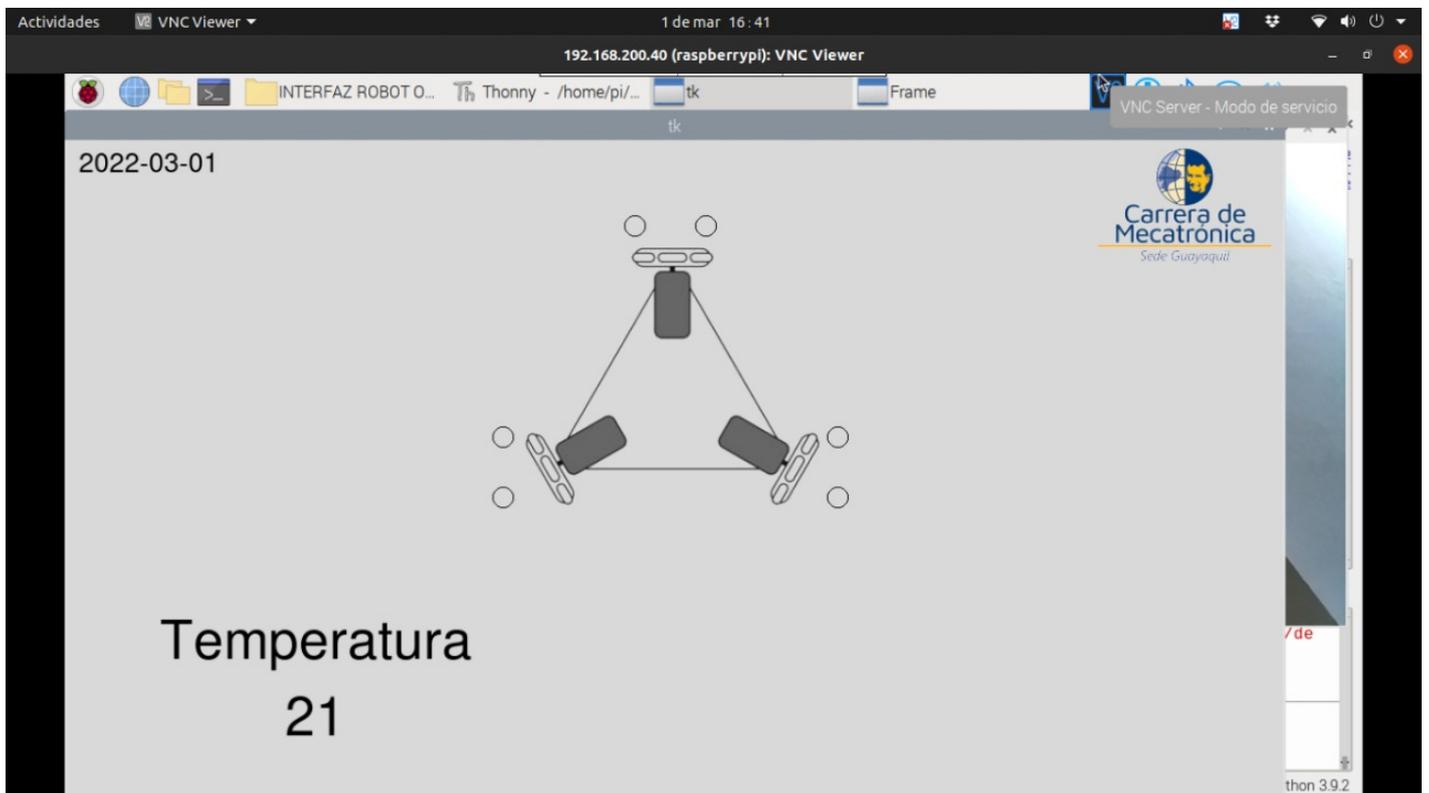


Figura 97. Interfaz de monitoreo visual en desarrollo

APÉNDICE F

CÓDIGOS DE DE ARDUINO PARA CONTROL DEL SISTEMA

```

void encoderB(void)
{
  antB=actB;
  actB=PINB & 144;

  if(antB==0 && actB==16)  nB++;
  if(antB==16 && actB==144) nB++;
  if(antB==128 && actB==0)  nB++;
  if(antB==144 && actB==128) nB++;

  if(antB==0 && actB==128)  nB--;
  if(antB==16 && actB==0)   nB--;
  if(antB==128 && actB==144) nB--;
  if(antB==144 && actB==16) nB--;
}

void encoderL(void)
{
  antL=actL;
  actL=PINL & 3;

  if(antL==0 && actL==1)  nL++;
  if(antL==1 && actL==3)  nL++;
  if(antL==2 && actL==0)  nL++;
  if(antL==3 && actL==2)  nL++;

  if(antL==0 && actL==2)  nL--;
  if(antL==1 && actL==0)  nL--;
  if(antL==2 && actL==3)  nL--;
  if(antL==3 && actL==1)  nL--;
}

void encoderR(void)
{
  antR=actR;
  actR=PINR & 12;

  if(antR==0 && actR== 4)  nR++;
  if(antR==4 && actR==12)  nR++;
  if(antR==8 && actR== 0)  nR++;
  if(antR==12 && actR== 8)  nR++;

  if(antR==0 && actR==8)  nR--;
  if(antR==4 && actR==0)  nR--;
  if(antR==8 && actR==12) nR--;
  if(antR==12 && actR==4) nR--;
}

```

Figura 98. Parámetros de giro de motores

```

#include "PinChangeInterrupt.h"

////////// Tiempo de muestreo //////////
unsigned long lastTime, sampleTime = 100;

const int C1 = A1;
const int C2 = A0;

volatile int n = 0;
volatile int ant = 0;
volatile int act = 0;

double P = 0.0;
const int R = 1980; // Resolucion del encoder R = mH*s*r

void setup() {

  Serial.begin(9600);

  pinMode(C1, INPUT);
  pinMode(C2, INPUT);

  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C1), encoder, CHANGE);
  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C2), encoder, CHANGE);

  lastTime = millis();
}

void loop()
{
  if(millis()-lastTime >= sampleTime)
  {
    lastTime = millis();
    P = n*360.0/R;
    Serial.print("Posicion en grados: ");
    Serial.println(P);
  }
}

void encoder(void)
{
  ant = act;
  act = PINC & 3;

  if(ant==0 && act==1) n++;
  if(ant==1 && act==3) n++;
  if(ant==2 && act==0) n++;
  if(ant==3 && act==2) n++;

  if(ant==0 && act==2) n--;
  if(ant==1 && act==0) n--;
  if(ant==2 && act==3) n--;
  if(ant==3 && act==1) n--;
}

```

Figura 99. Código que establece el sentido de giro de los motores

```

#include "PinChangeInterrupt.h"

////////////////////////////////// TIEMPO DE MUESTREO //////////////////////////////////
unsigned long lastTime, sampleTime = 100;

////////////////////////////////// COMUNICACION SERIAL //////////////////////////////////
String inputString = "";
bool stringComplete = false;
const char separator = ',';
const int dataLength = 1;
int data[dataLength];

////////////////////////////////// MOTOR //////////////////////////////////
const int BI1 = 10;
const int BI2 = 11;

int outValue = 0;

const int C1 = A1;
const int C2 = A0;
volatile int n = 0;
volatile int ant = 0;
volatile int act = 0;
const int pi = 3.1416;

double w = 0.0;
const int R = 1980;

void setup() {

  Serial.begin(9600);

  pinMode(BI2, OUTPUT);
  pinMode(BI1, OUTPUT);
  pinMode(C1, INPUT);
  pinMode(C2, INPUT);

  digitalWrite(BI2, LOW);
  digitalWrite(BI1, LOW);

  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C1), encoder, CHANGE);
  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C2), encoder, CHANGE);

  lastTime = millis();
}

void loop() {

  if(stringComplete)
  {
    for( int i =0; i<dataLength; i++)
    {
      int index = inputString.indexOf(separator);
      data[i]=inputString.substring(0,index).toInt();
      inputString = inputString.substring(index+1);
    }
  }
}

```

```

    if(outValue > 0) anticlockwise(BI1,BI2,outValue); else clockwise(BI1,BI2,abs(outValue));

    inputString = "";
    stringComplete = false;

}

if(millis()-lastTime >= sampleTime)
{
    w = (2*pi*1000.0*n)/((millis()-lastTime)*R);
    lastTime = millis();
    n = 0;

    Serial.print("Velocidad en rad/s: ");
    Serial.println(w);

}

}

void serialEvent()
{
    while(Serial.available())
    {
        char inChar = (char)Serial.read();
        inputString += inChar;
    }
}

void encoder(void)
{
    ant=act;
    act=PINC & 3;

    if(ant==0 && act==1) n++;
    if(ant==1 && act==3) n++;
    if(ant==2 && act==0) n++;
    if(ant==3 && act==2) n++;

    if(ant==0 && act==2) n--;
    if(ant==1 && act==0) n--;
    if(ant==2 && act==3) n--;
    if(ant==3 && act==1) n--;

}

void clockwise(int pin1, int pin2, int pwm)
{
    digitalWrite(pin1,LOW);
    analogWrite(pin2,pwm);

}

void anticlockwise(int pin1, int pin2, int pwm)
{
    digitalWrite(pin2,LOW);
    analogWrite(pin1,pwm);

}

```

Figura 100. Código que define la velocidad de los motores en RPM

APÉNDICE G

MOVIMIENTO DE MOTORES

```
#include "PinChangeInterrupt.h"
#include "motorControl.h"
#include "SoftwareSerial.h"

unsigned long lastTime, sampleTime = 10;

motorControl motor1(sampleTime);
motorControl motor2(sampleTime);
motorControl motor3(sampleTime);

//////////////////////////////////// COMUNICACION SERIAL //////////////////////////////////////
String inputString = "";
bool stringComplete = false;
const char separator = ',';
const int dataLength = 3;
double data[dataLength]; // Velocidad de referencia globales del robot
int n = 1000;
int vector[3];

//////////////////////////////////// MOTOR 1 //////////////////////////////////////
int AI1B = 10;
int AI2B = 11;
int outValueB = 0;

const int C1B = 7;
const int C2B = 4;
volatile int nB = 0;

//////////////////////////////////// MOTOR 2 //////////////////////////////////////
int BI1L = 9;
int BI2L = 6;
int outValueL = 0;

const int C1L = A1;
const int C2L = A0;
volatile int nL = 0;
volatile int antL = 0;
volatile int actL = 0;
double wL = 0;
double wRefL = 0;

//////////////////////////////////// MOTOR 3 //////////////////////////////////////
int AI1R = 3;
int AI2R = 5;
int outValueR = 0;

const int C1R = A3;
const int C2R = A2;
volatile int nR = 0;
volatile int antR = 0;
volatile int actR = 0;
double wR = 0;
double wRefR = 0;

//////////////////////////////////// VARIABLES PARA CALCULAR VELOCIDADES ANGULARES //////////////////////////////////////
double constValue = 3.1733; // (1000*2*pi)/R ----> R = 1980 Resolucion encoder cuadruple

//////////////////////////////////// ROBOT //////////////////////////////////////
double ufRobot = 0;
double ulRobot = 0;
double wRobot = 0;
double phi = 0;
double R = 3.5; //RADIO DE LA LLANTA
double L = 14.5; // DISTANCIA ENTRE EL CENTRO DEL ROBOT Y EL EJE DE LA RUEDA

int uf = 0; // inicializar todas las velocidades del robot
int ul = 0;
int w = 0;
```

```

int conec;

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

  /////////////////////////////////////////////////// SINTONIA FINA PID ////////////////////////////////////////

  motor1.setGains(0.19, 0.09, 0.00); // (Kc,Ti,Td) //0.21, 0.20, 0.04
  motor1.setCvLimits(255,0);
  motor1.setPvLimits(8,0);

  motor2.setGains(0.19, 0.09, 0.00); // (Kc,Ti,Td) //0.21, 0.20, 0.04
  motor2.setCvLimits(255,0);
  motor2.setPvLimits(8,0);

  motor3.setGains(0.19, 0.09, 0.00); // (Kc,Ti,Td) //0.21, 0.20, 0.04
  motor3.setCvLimits(255,0);
  motor3.setPvLimits(8,0);

  //////////////////////////////////////// MOTOR 1 ATRAS ////////////////////////////////////////
  pinMode(AI2B, OUTPUT);
  pinMode(AI1B, OUTPUT);
  pinMode(C1B, INPUT);
  pinMode(C2B, INPUT);

  digitalWrite(AI2B, LOW);
  digitalWrite(AI1B, LOW);

  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C1R), encoderR, CHANGE);
  attachPinChangeInterrupt(digitalPinToPinChangeInterrupt(C2R), encoderR, CHANGE);

  //////////////////////////////////////// BATERIA ////////////////////////////////////////
  // pinMode(buzzer,OUTPUT);
  // digitalWrite(buzzer,LOW);

  lastTime = millis();

  Serial.println("preparando el proceso");
}

void loop()
{
  if (Serial.available()>0){
    char Dato = Serial.read(); // mandamos el dato que necesitamos que nuestro controlador
    if(Dato == '\0'){
    }else{
      Serial.println(Dato);
    }

    if(Dato == 'm'){ // definimos el primer proceso manual y le asignamos la m
      Serial.println(Dato);
    }

    if(Dato == 'm'){ // definimos el primer proceso manual y le asignamos la m
      //Serial.println("proceso manual");
      while (true){

        char vectorR = Serial.read();

        if (vectorR == '\0') { // condicionamos nuestro vectorR con la cantidad r
        }else if(vectorR == '3'){
          movManual(vectorR); // primer proceso es entrar a la funcion de los mov.
          break;
        }else{
          //Serial.println(vectorR);
          movManual(vectorR);
          //encendidorele();
        }
      }
    }

    if(Dato == 'a'){ // condicionamos en nuestro proceso una letra que sea diferente a
      //Serial.println("proceso automatico");
      //sensorSharp(); // segundo proceso es entrar a la funcion de los lectura de sen.
      movAuto(); // primer proceso es entrar a la funcion de los movimiento de los
      //encendidorele();
    }
  }
}

```

```

void movAuto(){
}

void movManual(char accion){
// giro horario
if(accion == '1'){
uf = 0;
ul = 0;
w = 8;
motor1.reset();
motor2.reset();
motor3.reset();
}

// giro antihorario
if(accion == '2'){
uf = 0;
ul = 0;
w = -8;
motor1.reset();
motor2.reset();
motor3.reset();
}

//Serial.println(phi,2);
//Serial.println(ufRobot,2);
//Serial.println(ulRobot,2);
//Serial.println(wRobot,2);

outValueB = motor1.compute(wRefB,wB);
outValueL = motor2.compute(wRefL,wL);
outValueR = motor3.compute(wRefR,wR);

if (outValueB > 0) anticlockwise(AI2B,AI1B,outValueB); else clockwise(AI2B,AI1B,abs(outValueB));
if (outValueL > 0) anticlockwise(BI1L,BI2L,outValueL); else clockwise(BI1L,BI2L,abs(outValueL));
if (outValueR > 0) anticlockwise(AI2R,AI1R,outValueR); else clockwise(AI2R,AI1R,abs(outValueR));
}
}

void encoderB(void)
{
antB=actB;
actB=PIND & 144;

if(antB==0 && actB==16) nB++;
if(antB==16 && actB==144) nB++;
if(antB==128 && actB==0) nB++;
if(antB==144 && actB==128)nB++;

void clockwise(int pin1, int pin2, int pwm)
{
digitalWrite(pin1, LOW);
analogWrite(pin2,pwm);
}

void anticlockwise(int pin1, int pin2, int pwm)
{
digitalWrite(pin2, LOW);
analogWrite(pin1,pwm);
}

void velocityRobot(double w1, double w2, double w3)
{
ufRobot = (0.5774*R)^(w2-w3);
ulRobot = -(R/3)^(w2-(2^w1)+w3);
wRobot = -(R/(3^L))^(w1+w2+w3);
}

void velocityMotor(double uf, double ul, double w)
{
wRefB = (ul-(L^w))/R;
wRefL = -(ul+(2^L^w)-1.7321^uf)/(2^R);
wRefR = -(ul+(2^L^w)+1.7321^uf)/(2^R);
}

// Stop
if(accion == '3'){
uf = 0;
ul = 0;
w = 0;
motor1.reset();
motor2.reset();
motor3.reset();
}

wRefB = (ul-(L^w))/R;
wRefL = -(ul+(2^L^w)-1.7321^uf)/(2^R);
wRefR = -(ul+(2^L^w)+1.7321^uf)/(2^R);

////////////////////// CONTROLADOR PID ////////////////////////
if(millis()-lastTime >= sampleTime)
{
wB = constValue^nB/(millis()-lastTime);
wL = constValue^nL/(millis()-lastTime);
wR = constValue^nR/(millis()-lastTime);

lastTime = millis();

nB = 0;
nL = 0;
nR = 0;

void encoderL(void)
{
antL=actL;
actL=PINC & 3;

if(antL==0 && actL==1) nL++;
if(antL==1 && actL==3) nL++;
if(antL==2 && actL==0) nL++;
if(antL==3 && actL==2) nL++;

if(antL==0 && actL==2) nL--;
if(antL==1 && actL==0) nL--;
if(antL==2 && actL==3) nL--;
if(antL==3 && actL==1) nL--;
}

void encoderR(void)
{
antR=actR;
actR=PINC & 12;

if(antR==0 && actR== 4) nR++;
if(antR==4 && actR==12) nR++;
if(antR==8 && actR== 0) nR++;
if(antR==12 && actR== 8) nR++;
if(antR==0 && actR==8) nR--;
if(antR==4 && actR==0) nR--;
}
}

```

Figura 101. Código para indicar los movimientos de los motores

```

////////////////////////////////// SENSOR SHARK ////////////////////////////////////
unsigned long time = 0; // tiempo de ejecucion del ultimo ciclo
int period = 50; // Periodo de muestreo en ms
int sensorPin1 = A8; //Pin Analogico donde esta conectada la señal del Sensor de distancia
int sensorPin2 = A9;
int sensorPin3 = A10;
int sensorPin4 = A11;
int measure1; // Lo que mide el sensor. Son ADCs.
int measure2;
int measure3;
int measure4;
int dcal [] = { // Calibracion de ADC a Distancia
  39, 31, 23, 15, 0, -5, -10, -16, -19}; // maximo hasta -19 minimo 39
int ADCcal [] = {
  177, 194, 234, 275, 380, 417, 457, 502, 536};
int dist1; // variables para guardar la distancia en mm
int dist2;
int dist3;
int dist4;
int conec;
char buffer[10];

////////////////////////////////// conexion de sensor Sharp ////////////////////////////////////

void sensorSharp()
{
  if (millis()>time+period){ // ¿Ha transcurrido el periodo?
    time = millis();

    // Medimos DISTANCIA
    measure1 = analogRead(sensorPin1); // hacemos la lectura del sensor 1

    measure1 = constrain(measure1,ADCcal[0], ADCcal[8]); // contenemos nuestros registro en un vector de 9 posiciones antes

    for(int i =0; i<8; i++){ // Aplicamos curva de Calibracion de ADC a mm

      if (measure1 >= ADCcal[i] && measure1< ADCcal[i+1]){ // registro de datos del sensor 1
        dist1 = map(measure1,ADCcal[i],ADCcal[i+1],dcal[i],dcal[i+1]);
      }

    }

    if(1){ //Envio para PROCESSING
      Serial.print("Distancia en cm: ");
      Serial.println(dist1);
      delay(1000);
    }
  }
}
}

```

Figura 102. Código para calibrar el sensor IR

```

void encendidorele() {
for(int i=0; i<=2000;i++){
Serial.println(i);
if (i>0 && i<=11){
digitalWrite(PinRele,HIGH);
delay (500);
}
if (i>12 && i<=13){
Serial.println("apagado por 1s");
digitalWrite(PinRele,LOW);
delay (500);
}
if (i>15 && i<=17){
Serial.println("encendido de medio seg");
digitalWrite(PinRele,HIGH);
delay (50);
}
if (i>17 && i<=1500){
Serial.println("APAGADO");
digitalWrite(PinRele,LOW);
delay (500);
}
}
}

```

Figura 103. Código para controlar el encendido de las lámparas UV

APÉNDICE H

IMPLEMENTACIÓN DE CONTROL DE SEÑALES MEDIANTE PYTHON

```

from pyArduino import*
import matplotlib.pyplot as plt
import time

ts = 0.1# Tiempo de muestreo
tf = 10 # Tiempo de simulacion
t = np.arange(0,tf+ts,ts) # Array de tiempo
N = len(t) # Numero de muestras

##### Comunicacion Serial #####
port = 'COM4' # Com Arduino
baudRate = 9600 # Baudios

arduino = serialArduino(port,baudRate,1)# Objeto serial

arduino.readSerialStart() # Inicia lectura de datos

##### Señales #####
pv = np.zeros(N) # Variable de proceso (Pv)
cv = np.zeros(N) # Variable de control (Cv)

##### Loop #####
for k in range(N):

    start_time = time.time()# Tiempo actual

    # Escalon

    if k*ts > 3: # Escalon a los 3 segundos
        cv[k] = 40 # Valor escalon del 0 al 100% (40%)
    else:
        cv[k] = 0

    arduino.sendData([cv[k]]) # Enviar Cv (debe ser una lista)

    pv[k] = arduino.rawData[0] # Recibir Pv

    elapsed_time = time.time() - start_time # Tiempo transcurrido

    time.sleep(ts-elapsed_time)

    arduino.sendData([0])# Detener motor
    arduino.close()# Cerrar puerto serial

##### Guardar señales #####

with open('firstResponse.npy','wb') as f:
    np.save(f,cv)
    np.save(f,pv)
    np.save(f,t)
    np.save(f,ts)

##### Mostrar figuras #####
plt.plot(t,pv,label='Pv')
plt.plot(t,cv,label='Cv')
plt.legend(loc='upper right')
plt.show()

```

Figura 104. Cálculo del control de motores

```

from pyArduino import *
import matplotlib.pyplot as plt
import numpy as np

ts = 0.1 # Tiempo de muestreo
tf = 30 # Tiempo de simulacion
t = np.arange(0,tf+ts,ts) # Array de tiempo
N = len(t) # Numero de muestras

##### Comunicacion Serial #####

port = 'COM4' # Com Arduino
baudRate = 9600 # Baudios

arduino = serialArduino(port,baudRate,2)# Objeto serial

arduino.readSerialStart() # Inicia lectura de datos

##### Señales #####

pv = np.zeros(N) # Variable de proceso (Pv)
cv = np.zeros(N) # Variable de control (Cv)
sp = np.zeros(N) # Variable de deseada (Sp)

##### Setpoint Trayectoria #####
sp = 3*np.cos(0.2*t)+5

##### Setpoint Escalon#####
for k in range(N):
    if k*ts>3:
        #sp[k]= 7
    #else:
        #sp[k]= 0

##### Loop #####
for k in range(N):
    start_time = time.time() # Tiempo actual

##### Loop #####

for k in range(N):
    start_time = time.time() # Tiempo actual
    arduino.sendData([sp[k]]) # Enviar Sp (debe ser una lista)

    pv[k] = arduino.rawData[0] # Recibir Pv
    cv[k] = arduino.rawData[1] # Recibir Cv

    elapsed_time = time.time() - start_time # Tiempo transcurrido

    time.sleep(ts-elapsed_time) # Esperar hasta completar el tiempo de muestreo

    arduino.sendData([0]) # Detener motor
    arduino.close() # Cerrar puerto serial

##### Mostrar figuras #####
plt.figure()
plt.plot(t,sp,label='Sp')
plt.plot(t,pv,label='Pv')
plt.legend(loc='upper left')

plt.figure()
plt.plot(t,cv,label='Cv')
plt.legend(loc='upper left')

plt.show()

```

Figura 105. Control de lazo cerrado de los motores