



**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE GUAYAQUIL**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

**PROYECTO TÉCNICO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE:**

**INGENIERO ELECTRÓNICO**

**PROYECTO TÉCNICO:**

**“DISEÑO E IMPLEMENTACIÓN DE DOS ROBOTS DE BATALLA  
DE 3 LIBRAS PARA EL CLUB DE ROBÓTICA DE LA  
UNIVERSIDAD POLITÉCNICA SALESIANA.”**

**AUTORES:**

**JOSÉ OMAR SALDARRIAGA ANDRADE  
JUAN JOSÉ MERO SIMISTERRA**

**TUTOR:**

**ING. BYRON XAVIER LIMA CEDILLO, Msc.**

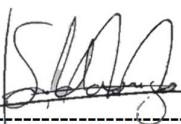
**GUAYAQUIL – ECUADOR**

**2020**

## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

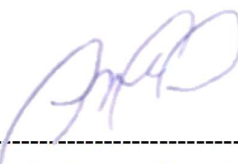
Nosotros, JOSÉ OMAR SALDARRIAGA ANDRADE y JUAN JOSÉ MERO SIMISTERRA autorizamos a la **Universidad Politécnica Salesiana** la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, se declara que los conceptos y análisis desarrollados y conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.



---

José Omar Saldarriaga Andrade  
Cédula: 1313238303



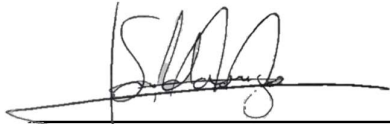
---

Juan José Mero Simisterra  
Cédula: 0802550699

## CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Nosotros, **JOSÉ OMAR SALDARRIAGA ANDRADE** y **JUAN JOSÉ MERO SIMISTERRA**, con documentos de identificación N° **1313238303** y **0802550699**, manifestamos nuestra voluntad y cedemos a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado: **“DISEÑO E IMPLEMENTACIÓN DE DOS ROBOTS DE BATALLA DE 3 LIBRAS PARA EL CLUB DE ROBÓTICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA.”** mismo que ha sido desarrollado para optar por el título de **INGENIEROS ELECTRÓNICOS**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscritos este documento en el momento que se realiza la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



José Omar Saldarriaga Andrade  
Cédula: 1313238303

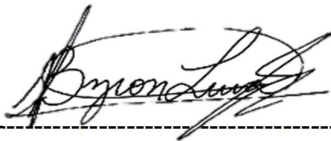


Juan José Mero Simisterra  
Cédula: 0802550699

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Ing. Byron Xavier Lima Cedillo, MSc.** Director del proyecto de titulación denominado: **“DISEÑO E IMPLEMENTACIÓN DE DOS ROBOTS DE BATALLA DE 3 LIBRAS PARA EL CLUB DE ROBÓTICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA.”** realizado por los estudiantes **JOSÉ OMAR SALDARRIAGA ANDRADE** y **JUAN JOSÉ MERO SIMISTERRA**, certifico que han sido orientados y revisado durante su desarrollo, por cuanto se aprueba la presentación de este ante las autoridades pertinentes.

Guayaquil, 20 de septiembre del 2020



---

**Ing. Byron Xavier Lima Cedillo, MSc.**  
**Cedula: 0921971768**

## **DEDICATORIA**

Este proyecto está dedicado primeramente a Dios, que siempre está a mi lado en cada una de las etapas, guiándome y protegiéndome de las adversidades brindándome sabiduría y fuerzas para seguir adelante.

A mis padres, Acenet Andrade B. y Pedro Saldarriaga Z. quienes me han apoyado siempre, en cada momento con sus consejos, sus alegrías, sus ideas y esa clara visión de que la vida se trata de superar los retos y no rendirse jamás.

A mis abuelos Carlina Basurto, Fe Zambrano y Omar Saldarriaga, que siempre con sabios consejos, me sirvieron de inspiración para cumplir con este objetivo de vida.

**José Omar Saldarriaga Andrade**

## **DEDICATORIA**

Esta tesis está dedicada principalmente a mis padres, Lorgio Mero e Isabel Simisterra, pilares fundamentales de mi vida, les dedico este triunfo como retribución a su esfuerzo, este logro sé que lo sentirán como suyo.

Dedico este trabajo cumplido a mis hijos a quienes amo con mi vida, para que tomen como ejemplo de superación y nunca se rindan y luchen por sus sueños y metas.

**Juan José Mero Simisterra**

## **AGRADECIMIENTO**

En primera instancia agradezco a todos ingenieros, que con sus sabias enseñanzas fueron mis educadores de manera incondicional, en este periodo de vida e hicieron posible este proyecto.

Considero nombrar a Ing. Luis Córdova, Ing. Luis Neira, Ing. Ricardo Cajo, Ing. Victor Huilcapi, Ing. Carlos Bosquez y el Ing. Diego Freire profesores que nos brindaron sus conocimientos técnicos y una manera distinta de desarrollar nuestras ideas, proyectándonos no solo como profesionales de tercer nivel, sino como dueños de nuestras empresas, profesionales de alto nivel con doctorados técnicos, motivándonos a generar nuestros propios sueños y cumplir nuestras metas.

De manera especial agradezco al Ing. Byron Lima y el Ing. Orlando Barcia quienes fueron mentores y amigos, brindándome todos sus conocimientos y oportunidades para aprender no solo de la academia, sino del manejo de grupos, compartir el conocimiento a quienes lo necesiten, escuchar y respetar las ideas de las demás, motivar a tu equipo a ser una familia, a festejar tus logros y sonreír con la frente en alto en las derrotas, cualidades que representan lo que es un ingeniero de la Salesiana.

Agradezco de manera especial también a mis compañeros del Club de Robótica UPS–GYE, que desde la conocida viaja guardia hasta los más, siempre seremos una familia, apoyándonos, compartiendo victorias, derrotas, risas, viajes y conocimientos que se ven reflejado en lo que representa en la actualidad el Club para la Universidad.

**José Omar Saldarriaga Andrade**

## **AGRADECIMIENTO**

Es imposible llegar a estas instancias de la vida, donde se puede gozar y cosechar los frutos del esfuerzo al que se han dedicado un tiempo importante de nuestra existencia sin poder mirar al cielo y agradecer primero a Dios, por la sencilla razón de darnos la vida y fuerzas para luchando diariamente y darnos la voluntad y herramientas necesarias para cumplir nuestros propósitos. Luego es importante agradecer a las personas que con poco o mucho han contribuido con su grano de arena para que este ciclo y logro de mi vida haya sido posible. Por lo tanto, agradezco a mis padres que desde muy pequeño supieron guiarme e inculcarme valores que han sido vitales para formar de manera acertada mi personalidad, la misma que me ha servido para enfrentar las distintas problemáticas encontradas en mi camino. Agradezco a mi novia que ha sabido siempre con paciencia alentarme, llenándome de consejos acertados con mucho amor, los cuales me han dado la calma necesaria para seguir adelante.

Agradezco de igual forma a los amigos y compañeros que han sido parte de hermosa y dura parte de mi vida, con quienes hemos compartido por largos años este camino, y entre risas y sufrimientos hemos podido culminar el cometido.

A mis profesores y tutor, ya que sin su dedicación y entrega no fuera posible el haber culminado este camino.

**Juan José Mero Simisterra**



## RESUMEN

| AÑO  | ALUMNOS  | DIRECTOR DE PROYECTO TÉCNICO         | TEMA DE PROYECTO TÉCNICO   |
|------|--|--------------------------------------|--|
| 2020 | JOSÉ OMAR SALDARRIAGA ANDRADE<br>JUAN JOSÉ MERO SIMISTERRA | ING. BYRON XAVIER LIMA CEDILLO, MSc. | DISEÑO E IMPLEMENTACIÓN DE DOS ROBOTS DE BATALLA DE 3 LIBRAS PARA EL CLUB DE ROBÓTICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA. |

El presente proyecto técnico tiene como objetivo el diseño e implementación de dos robots de batalla en la categoría de 3 libras, utilizando componentes de nivel profesional basándose en los estándares de combate a nivel nacional e internacional para futuras competencias del Club de Robótica la Universidad Politécnica Salesiana.

Mediante el uso una tarjeta microcontroladora ESP32 y de sensores, se obtuvo las magnitudes de voltajes y amperajes que fueron graficadas en tiempo real en Matlab y almacenadas en una tarjeta de memoria que se utilizó como un datalogger, para su análisis, utilizando una interfaz gráfica en Matlab.

El objetivo principal fue obtener los valores potencia de los motores de movimiento y el motor del arma para realizar un análisis del consumo de energía en cada motor y generar estrategias de combate manteniendo un uso eficiente de las baterías que es información esencial para el operador y su equipo en cada round de combate.

Palabras claves: Estrategias, Datalogger, Voltaje, Amperaje, ESP32.

## ABSTRACT

| YEAR | STUDENTS  | PROJECT DIRECTOR                                 | PROJECT THEME  |
|------|---|--|--|
| 2020 | JOSÉ OMAR<br>SALDARRIAGA<br>ANDRADE<br><br>JUAN JOSÉ MERO<br>SIMISTERRA | ING. BYRON<br>XAVIER<br>LIMA<br>CEDILLO,<br>MSc. | DESIGN AND<br>IMPLEMENTATION OF TWO 3<br>POUND BATTLE ROBOTS FOR<br>THE ROBOTICS CLUB OF THE<br>SALESIAN POLYTECHNICAL<br>UNIVERSITY . |

The objective of this technical project is the design and implementation of two battle robots in the 3-pound category, using professional-level components in the national and international combat standards for future competitions of the Robotics Club of the Salesian Polytechnic University.

Through the use of an ESP32 microcontroller card and sensors, the magnitudes of voltages and amperages were obtained that were plotted in real time in Matlab and stored on a memory card that was used as a datalogger, for analysis, using a graphical interface. in Matlab.

The main objective was to obtain the power values of the movement motors and the weapon motor to perform an analysis of the energy consumption in each motor and to generate combat strategies while maintaining efficient use of the batteries, which is essential information for the operator and its team in each round of combat.

Keywords: Strategies, Datalogger, Voltage, Amperage, ESP32.

## ÍNDICE GENERAL

|  |       |
|--|-------|
| Contenido  |       |
| CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA .....                 | II    |
| CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR.....                | III   |
| CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....        | IV    |
| DEDICATORIA.....   | V     |
| DEDICATORIA.....   | VI    |
| AGRADECIMIENTO.....  | VII   |
| AGRADECIMIENTO.....  | VIII  |
| RESUMEN.....   | IX    |
| ABSTRACT .....   | X     |
| ÍNDICE GENERAL .....   | XI    |
| ÍNDICE DE FIGURAS.....   | XV    |
| ÍNDICE DE TABLAS.....  | XVIII |
| INTRODUCCIÓN.....  | - 1 - |
| 1. EL PROBLEMA.....  | - 2 - |
| 1.1 Descripción del Problema.....                              | - 2 - |
| 1.2 Importancia y Alcance.....                                 | - 2 - |
| 1.3 Delimitación del Problema.....                             | - 2 - |
| 1.3.1 Delimitación Temporal:.....                              | - 2 - |
| 1.3.2 Delimitación Espacial.....                               | - 2 - |
| 1.3.3 Delimitación Académica.....                              | - 3 - |
| 1.4 Objetivos.....   | - 3 - |
| 1.4.1 Objetivo General.....                                    | - 3 - |
| 1.4.2 Objetivos Específicos.....                               | - 3 - |
| 2. FUNDAMENTACIÓN TEÓRICA.....                                 | - 4 - |
| 2.1 Batalla de Robots.....                                     | - 4 - |
| 2.1.1 Historia de Battlebot.....                               | - 5 - |
| 2.1.2 Principales concursos de batallas de robots.....         | - 6 - |
| 2.2 Principales materiales para la construcción de robots..... | - 8 - |
| 2.2.1 Metales.....   | - 8 - |
| 2.2.1.1 Acero 1018, acero 1020.....                            | - 8 - |
| 2.2.1.2 Acero 1045.....  | - 8 - |
| 2.2.1.3 Acero 4340.....  | - 8 - |
| 2.2.1.4 Aceros 5260.....                                       | - 8 - |
| 2.2.1.5 Acero inoxidable.....                                  | - 9 - |
| 2.2.2 Aleaciones de Aluminio.....                              | - 9 - |
| 2.2.3 Aleaciones de titanio.....                               | - 9 - |
| 2.2.4 Plásticos de alta resistencia.....                       | - 9 - |

|         |   |        |
|---------|---|--------|
| 2.2.4.1 | Policarbonato .....                                       | - 9 -  |
| 2.2.4.2 | PETG .....  | - 9 -  |
| 2.2.4.3 | Teflón (PTFE, politetrafluoretileno).....                 | - 9 -  |
| 2.2.4.4 | UHMW .....  | - 10 - |
| 2.2.4.5 | Nylon.....  | - 10 - |
| 2.2.4.6 | Fibra de carbono .....                                    | - 10 - |
| 2.3     | Tipos de robots de batalla.....                           | - 10 - |
| 2.3.1   | Rammers.....  | - 10 - |
| 2.3.2   | Sawbots.....  | - 11 - |
| 2.3.3   | Wedges .....  | - 11 - |
| 2.3.4   | Spearbots .....   | - 11 - |
| 2.3.5   | Thwackbots .....  | - 12 - |
| 2.3.6   | Thwackbots de arriba .....                                | - 12 - |
| 2.3.7   | Horizontal Spinners .....                                 | - 12 - |
| 2.3.8   | Drumbots.....   | - 13 - |
| 2.4     | Data Logger .....   | - 13 - |
| 2.5     | Batería de Lipo.....                                      | - 13 - |
| 2.5.1   | Significado de la nomenclatura de una batería LiPo .....  | - 14 - |
| 2.5.1.1 | Clasificación por número de celdas S .....                | - 14 - |
| 2.5.1.2 | Capacidad indicada en mAh .....                           | - 14 - |
| 2.5.1.3 | Tasa de descarga, número C.....                           | - 14 - |
| 3       | MARCO METODOLÓGICO.....                                   | - 15 - |
| 3.1     | Diseño Mecánico.....                                      | - 15 - |
| 3.1.1   | Arma completa .....                                       | - 15 - |
| 3.1.2   | Rodamiento .....  | - 15 - |
| 3.1.3   | Estructura del robot .....                                | - 16 - |
| 3.1.4   | Tapas superior e inferior de los robots.....              | - 17 - |
| 3.1.5   | Poleas del arma y motor .....                             | - 17 - |
| 3.1.6   | Soporte de motores brushless de arma .....                | - 18 - |
| 3.1.7   | Soporte de motores .....                                  | - 18 - |
| 3.1.8   | Llantas de robots .....                                   | - 18 - |
| 3.1.9   | Acople para motor y llantas .....                         | - 19 - |
| 3.1.10  | Patines del robot 1.....                                  | - 19 - |
| 3.1.11  | Patines del robot 2.....                                  | - 19 - |
| 3.2     | Diseño eléctrico de los robots de batalla .....           | - 20 - |
| 3.2.1   | Diagrama de flujo del funcionamiento en modo batalla..... | - 21 - |
| 3.3     | Componentes electrónicos de los robots.....               | - 21 - |
| 3.3.1   | Turnigy Evolution PRO Digital .....                       | - 21 - |
| 3.3.2   | Receptor FS-IA6B.....                                     | - 22 - |

|       |  |        |
|-------|--|--------|
| 3.3.3 | Mini interruptor de encendido FingerTech.....                                | - 23 - |
| 3.3.4 | E.S.C T-Motor F 35A 32bit 3-6S .....   | - 23 - |
| 3.3.5 | Motores brushless .....  | - 24 - |
| 3.3.6 | Controladora de motores .....  | - 25 - |
| 3.3.7 | Motores 20.4:1 Metal Gearmotor 25Dx50L mm HP .....                           | - 26 - |
| 3.4   | Datalogger de potencia de motores.....                                       | - 27 - |
| 3.4.1 | Diagrama de flujo del funcionamiento del datalogger.....                     | - 27 - |
| 3.4.2 | Diagrama de flujo de la programación del datalogger.....                     | - 28 - |
| 3.5   | Componentes electronicos del datalogger.....                                 | - 29 - |
| 3.5.1 | Placa de desarrollo Node MCU ESP32.....                                      | - 29 - |
| 3.5.2 | Bluetooth HC-05.....   | - 29 - |
| 3.5.3 | Lector SD o micro SD .....   | - 30 - |
| 3.5.8 | Batería 9V .....   | - 30 - |
| 3.5.4 | Sensor de voltaje y corriente MAX471.....                                    | - 30 - |
| 3.5.5 | Sensor de corriente ACS714 -5A a + 5A .....                                  | - 31 - |
| 3.5.6 | Sensor de corriente ACS709 -75A a + 75A.....                                 | - 32 - |
| 4.    | Resultados .....   | - 33 - |
| 4.1   | Análisis Estructural.....  | - 33 - |
| 4.1.1 | Análisis de Arma de robot #1.....  | - 33 - |
| 4.1.2 | Análisis de Arma de robot #2.....  | - 34 - |
| 4.1.3 | Análisis estructura lateral de derecha .....                                 | - 36 - |
| 4.1.4 | Análisis estructura lateral izquierda .....                                  | - 37 - |
| 4.2   | Análisis de diseño del robot 1 para considerar como estrategias de combate.- | 39 -   |
| 4.3   | Análisis de diseño del robot 2 para considerar como estrategias de combate.- | 41 -   |
| 4.3.1 | Diseño del Datalogger en robots. ....  | - 43 - |
| 4.4   | Lógica operacional del Datalogger .....                                      | - 48 - |
| 4.4.1 | Escritura de datos en la tarjeta Micro SD.....                               | - 48 - |
| 4.4.2 | Visualización en matlab de las gráficas de potencias en tiempo real... -     | 49 -   |
| 4.4.3 | Resultados de pruebas obtenidos del data logger por tarjeta SD.....          | - 51 - |
|       | Conclusiones .....   | - 53 - |
|       | Recomendaciones.....   | - 53 - |
|       | Bibliografía.....  | - 55 - |
|       | Anexos .....   | - 58 - |
|       | Anexo 1A: Arma del robot 1.....  | - 58 - |
|       | Anexo 1B: Arma del robot 2.....  | - 59 - |
|       | Anexo 1A: Planos de eje y poleas de armas.....                               | - 60 - |
|       | Anexo 2A: Planos de la estructura lateral derecha.....                       | - 61 - |
|       | Anexo 2B: Planos de la estructura lateral izquierda .....                    | - 62 - |

|  |        |
|--|--------|
| Anexo 3A: Planos de la tapa inferior .....                                 | - 63 - |
| Anexo 3B: Planos de la tapa superior .....                                 | - 64 - |
| Anexo 4: Planos de poleas para motores brushless .....                     | - 65 - |
| Anexo 5: Planos del soporte de motor Brushless.....                        | - 66 - |
| Anexo 6A: Planos del soporte izquierdo del motor DC Pololu 25Dx50L.....    | - 67 - |
| Anexo 6B: Planos del soporte derecho del motor DC Pololu 25Dx50L. ....     | - 68 - |
| Anexo 7: Planos de patines tipo gancho para el robot 2 .....               | - 69 - |
| Anexo 8: Diagrama eléctrico de conexiones .....                            | - 70 - |
| Anexo 9: Planos de las llantas Bannebot 400 23/8 .....                     | - 71 - |
| Anexo 10: Planos estructurales de los motores .....                        | - 72 - |
| Anexo 11: Planos estructurales de los acoples Bannebots T40H-Sbb1 .....    | - 73 - |
| Anexo 12: Código Arduino .....   | - 74 - |
| Anexo 12: Código MATLAB gráficos de robots de batallas .....               | - 89 - |
| Anexo 13: Código MATLAB gráficos de robots de batallas por bluetooth ..... | - 92 - |
| Anexos 14: Foto galería.....   | - 97 - |

## ÍNDICE DE FIGURAS

|  |        |
|--|--------|
| <b>Figura 1.</b> Delimitación espacial del prototipo del Proyecto Técnico Club de Robótica UPS-GUAYAQUIL .....               | - 3 -  |
| <b>Figura 2.</b> Robot Tutuchin(120lb) de UPS-GYE en Batalla de Robots CER2015. .  | - 4 -  |
| <b>Figura 3.</b> Reparaciones realizadas por equipo del Club de Robotica UPS-GYE. .  | - 4 -  |
| <b>Figura 4.</b> Mechadon .....  | - 5 -  |
| <b>Figura 5.</b> Batalla de robots en Battlebots 2019.....   | - 5 -  |
| <b>Figura 6.</b> Equipo WanHoo (2019).....   | - 6 -  |
| <b>Figura 7.</b> Robot tipo Rammers. ....  | - 10 - |
| <b>Figura 8.</b> Robot tipo Sawbots .....  | - 11 - |
| <b>Figura 9.</b> Robot tipo Wedges.....  | - 11 - |
| <b>Figura 10.</b> Robot tipo spearbots. ....   | - 11 - |
| <b>Figura 11.</b> Robot tipo Thwackbots.....   | - 12 - |
| <b>Figura 12.</b> Robot tipo Thwackbots de arriba.....   | - 12 - |
| <b>Figura 13.</b> Robot tipo Horizontal Spinners.....  | - 12 - |
| <b>Figura 14.</b> Robots tipo Drumbots. ....   | - 13 - |
| <b>Figura 15.</b> Batería de LIPO de 450MAh 7.4v 2S 75C.....   | - 13 - |
| <b>Figura 16.</b> Nomenclatura de una batería LiPo.....  | - 14 - |
| <b>Figura 17.</b> Armas del Robot 1 y Robot 2. ....  | - 15 - |
| <b>Figura 18.</b> Rodamiento rígido de bolas. ....   | - 15 - |
| <b>Figura 19.</b> Planchas de polipropileno de alta de densidad. ....  | - 16 - |
| <b>Figura 20.</b> Piezas de la estructura del robot 1 y robot 2. ....  | - 16 - |
| <b>Figura 21.</b> Tapas superiores e inferiores de los robots.....   | - 17 - |
| <b>Figura 22.</b> Poleas para el motor brushless. ....   | - 17 - |
| <b>Figura 23.</b> Soporte del motor de arma.....   | - 18 - |
| <b>Figura 24.</b> Soportes de derecha e izquierda de motores DC.....   | - 18 - |
| <b>Figura 25.</b> Llantas Bannebots T40P 2 3/8.....  | - 18 - |
| <b>Figura 26.</b> Acople de motor Bannebot T40H – Sbb1.....  | - 19 - |
| <b>Figura 27.</b> Patines del robot 1 para mantener la altura del arma y reducir la fricción. -                              | 19 -   |
| <b>Figura 28</b> Patines del robot 2 para mantener la altura del arma y como puntas para causar daño a su contrincante. .... | - 19 - |
| <b>Figura 29.</b> Diagrama de conexiones del Robot 1.....  | - 20 - |
| <b>Figura 30.</b> Diagrama de conexiones del Robot 2.....  | - 20 - |
| <b>Figura 31.</b> Diagrama de flujo de funcionamiento de los robots en modo Batalla. -                                       | 21 -   |
| <b>Figura 32.</b> Control Turnigy Evolution PRO Digital.....   | - 22 - |
| <b>Figura 33.</b> Receptor FS-IA6B de 6 canales de 2,4Ghz.....   | - 22 - |
| <b>Figura 34.</b> Mini interruptor de encendido FingerTech. ....   | - 23 - |
| <b>Figura 35.</b> Control de Velocidad electrónica T-Motor F 35. ....  | - 23 - |
| <b>Figura 36.</b> Motor sin escobillas.....  | - 24 - |
| <b>Figura 37.</b> Prop Drive 28-36 3000KV / 755W. ....   | - 24 - |
| <b>Figura 38.</b> T-Motor F80 PRO - Motor de 2500KV / 706 W.....   | - 24 - |
| <b>Figura 39.</b> Controlador de motor Sabertooth dual 12A para R / C .....  | - 25 - |
| <b>Figura 40.</b> Scorpion Mini Dual 2x6.5A. ....  | - 26 - |
| <b>Figura 41.</b> Motorreductor Metálico 25Dx50L. ....   | - 26 - |
| <b>Figura 42.</b> Diagrama de flujo de funcionamiento del datalogger en los robots... -                                      | 27 -   |
| <b>Figura 43.</b> Diagrama de flujo de la programación del datalogger en los robots. -                                       | 28 -   |
| <b>Figura 44.</b> Placa de desarrollo Node MCU ESP32.....  | - 29 - |
| <b>Figura 45.</b> Modulo Bluetooth HC-05.....  | - 29 - |
| <b>Figura 46.</b> Lectora de micro SD. ....  | - 30 - |

|   |        |
|---|--------|
| <b>Figura 47.</b> Batería de 9v. ....   | - 30 - |
| <b>Figura 48.</b> Max471 sensor de voltaje y de corriente. ....   | - 31 - |
| <b>Figura 49.</b> Sensor de corriente ACS714 -5A a + 5A. ....   | - 31 - |
| <b>Figura 50.</b> Sensor de corriente ACS709 -75A a + 75A. ....   | - 32 - |
| <b>Figura 51.</b> Modelado 3D de arma 1 visto desde solidworks. ....  | - 33 - |
| <b>Figura 52.</b> Arma de robot 1 deformada sometida a tensión. ....  | - 33 - |
| <b>Figura 53.</b> Desplazamiento de robot 1 sometida a tensiones. ....  | - 34 - |
| <b>Figura 54.</b> Modelado 3D de arma 2 visto desde solidworks. ....  | - 34 - |
| <b>Figura 55.</b> Arma de robot 2 deformada sometida a tensión. ....  | - 35 - |
| <b>Figura 56.</b> Deformación y desplazamiento del arma del robot 2 sometida a tensiones. ....                  | - 35 - |
| <b>Figura 57.</b> Modelado 3D de estructura lateral derecha visto desde solidworks. ....                        | - 36 - |
| <b>Figura 58.</b> Desplazamiento de tapa lateral derecha sometida a tensiones. ....                             | - 36 - |
| <b>Figura 59.</b> Deformación de tapa lateral derecha sometida a tensiones. ....                                | - 37 - |
| <b>Figura 60.</b> Modelado 3D de estructura lateral izquierda visto desde solidworks. ....                      | - 37 - |
| <b>Figura 61.</b> Desplazamiento de tapa lateral izquierda sometida a tensiones. ....                           | - 38 - |
| <b>Figura 62.</b> Deformación de tapa lateral izquierda sometida a tensiones. ....                              | - 38 - |
| <b>Figura 63.</b> Vista lateral derecha robot 1. ....   | - 39 - |
| <b>Figura 64.</b> Vista lateral derecha robot 1 levantado por una rampa. ....                                   | - 39 - |
| <b>Figura 65.</b> Vista frontal del robot 1. ....   | - 39 - |
| <b>Figura 66.</b> Vista frontal del robot 1 evitar recibir golpes directos en las áreas marcadas. ....          | - 40 - |
| <b>Figura 67.</b> Vista de alzado en esquina derecha del robot1. ....   | - 40 - |
| <b>Figura 68.</b> Vista superior del robot1. ....   | - 40 - |
| <b>Figura 69.</b> Vista lateral derecha robot 2. ....   | - 41 - |
| <b>Figura 70.</b> Vista lateral derecha robot 2. ....   | - 41 - |
| <b>Figura 71.</b> Vista frontal robot 2. ....   | - 41 - |
| <b>Figura 72.</b> Vista frontal del robot 2 evitar recibir golpes directos en las áreas marcadas. ....          | - 42 - |
| <b>Figura 73.</b> Vista de alzado en esquina derecha del robot2. ....   | - 42 - |
| <b>Figura 74.</b> Vista su del robot2. ....   | - 42 - |
| <b>Figura 75.</b> Perspectiva derecha del Data Logger modelado en 3D. ....                                      | - 43 - |
| <b>Figura 76.</b> Perspectiva detrás derecha del Data Logger modelado en 3D. ....                               | - 43 - |
| <b>Figura 77.</b> Vista lateral derecha del Data Logger modelado en 3D. ....                                    | - 43 - |
| <b>Figura 78.</b> Vista posterior del Data Logger modelado en 3D. ....  | - 44 - |
| <b>Figura 79.</b> Impresión en 3D de los soportes de la base soporte del data logger. ....                      | - 44 - |
| <b>Figura 80.</b> Base soporte impresa del data logger. ....  | - 44 - |
| <b>Figura 81.</b> Perspectiva Isométrica frontal y posterior del robot 1 con el datalogger. ....                | - 45 - |
| <b>Figura 82.</b> Perspectiva Isométrica frontal y posterior del robot 2 con el datalogger. ....                | - 45 - |
| <b>Figura 83.</b> Perspectiva Isométrica superior del robot 1 y robot 2 con el data logger. ....                | - 45 - |
| <b>Figura 84.</b> Perspectiva Isométrica superior del robot 1 y robot 2 con protección para el datalogger. .... | - 45 - |
| <b>Figura 85.</b> Prueba de funcionalidad de componentes. ....  | - 46 - |
| <b>Figura 86.</b> Integración de componentes del data logger. ....  | - 46 - |
| <b>Figura 87.</b> Data logger acoplados en ambos robots. ....   | - 47 - |
| <b>Figura 88.</b> Vista posterior del robot 1 y robot 2. ....   | - 47 - |
| <b>Figura 89.</b> Vista lateral del robot 1 y robot 2. ....   | - 47 - |
| <b>Figura 90.</b> Vista isométrica del robot 1 y robot 2. ....  | - 47 - |



|   |        |
|---|--------|
| <b>Figura 91</b> Robot 1 y 2 terminados .....   | - 48 - |
| <b>Figura 92.</b> Selección de robot en Matlab.....   | - 49 - |
| <b>Figura 93.</b> Selección de cámara en Matlab. ....   | - 49 - |
| <b>Figura 94.</b> Interruptor de encendido del datalogger-robot2.....   | - 50 - |
| <b>Figura 96</b> Pruebas del robot 2 valores de potencias de motores en tiempo real.-   | 51 -   |
| <b>Figura 97</b> Apertura de carpeta de datos de microSD. ....  | - 51 - |
| <b>Figura 98</b> Robot 1 gráfica de datos obtenidos de tarjeta microSD.....   | - 52 - |
| <b>Figura 99</b> Robot 2 gráfica de datos obtenidos de tarjeta microSD.....   | - 52 - |
| <b>Figura 101</b> Participación en el robochallenge obteniendo el 4to lugar en México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)..... | - 98 - |
| <b>Figura 102</b> Equipo de Robótica UPS-GYE en el robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ) .....              | - 98 - |
| <b>Figura 103</b> Equipo de Robótica UPS-GYE en el robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ) .....              | - 99 - |

## ÍNDICE DE TABLAS

|   |        |
|---|--------|
| Tabla # 1 Categorías de Batalla de Robots .....                         | - 6 -  |
| Tabla # 2 Principales concursos de Robotica a nivel nacional .....      | - 7 -  |
| Tabla # 3 Principales concursos de Robotica a nivel internacional ..... | - 7 -  |
| Tabla # 4 Comparación de motores brushless.....                         | - 24 - |
| Tabla # 5 Comparación de controladores de motores.....                  | - 25 - |

## INTRODUCCIÓN

El presente documento está dirigido a los miembros del Club de Robótica y estudiantes de la Universidad Politécnica Salesiana que durante muchos años han podido participar en eventos relacionados a competencias de robótica, obteniendo los primeros lugares muchas de las categorías competidas, la implementación de dos robots de 3 libras considerando su diseño estructural, la implementación de su electrónica, la comunicación y baterías basado en los estándares profesionales de un nivel de una competencia internacional, permitirán la participación en esta categoría que en la actualidad cuenta con muchos participantes nacionales e internacionales a un nivel profesional.

Con la implementación de una interfaz gráfica en Matlab en tiempo real y un datalogger para adquisición y almacenamiento de las magnitudes de voltaje y amperaje mediante uso de sensores, se obtuvo los valores de la potencia de los motores de movimiento y arma, para generar estrategias de combate y la optimización del uso eficiente de baterías para cada round de combate.

La implementación de los prototipos robóticos, nos permite demostrar los conocimientos adquiridos en el pensum académico, relacionando cada materia de la malla curricular con las actividades propuestas en este proyecto desde su análisis estructural en el diseño, la implementación de componentes electrónicos dimensionados a su requerimiento y el análisis de datos para cumplir los objetivos planteados.

## **1. EL PROBLEMA.**

### **1.1 Descripción del Problema.**

En las competencias de batallas de robot de 3 libras a nivel nacional e internacional, los equipos cuentan con robots que utilizan diseños y componentes de uso profesionales, sus operadores requieren obtener un alto desempeño de los motores y un consumo controlado de energía de las baterías durante cada round de batalla. Por tanto, se hace indispensable conocer y guardar los valores de las magnitudes de voltaje y amperaje de manera gráfica, para realizar un análisis de funcionamiento del robot y generar estrategias de combate en los distintos escenarios como impactos producidos por las de armas y movimientos del robot.

### **1.2 Importancia y Alcance.**

Se diseñó e implemento dos robots de batallas de 3 libras para que los alumnos de Ingeniería Electrónica y Automatización puedan obtener datos estadísticos de estado de las baterías, y elaborar estrategias de combate que permitan mejorar su desempeño durante las competencias de batallas de robots.

Este proyecto técnico es de mucha importancia para el Club de Robótica de nuestra sede, dada la elaboración de planos mecánicos y electrónicos que servirán de base para la construcción de nuevos prototipos, manteniendo un estándar profesional.

### **1.3 Delimitación del Problema.**

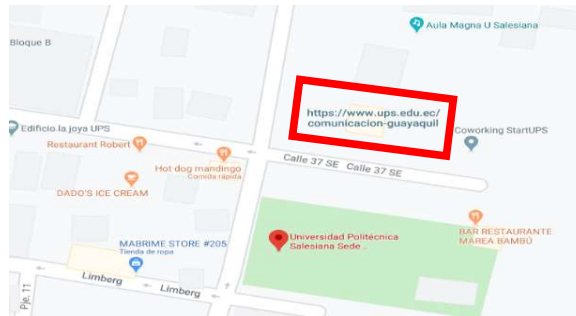
Se diseñó e implemento dos robots de batallas de 3 libras para que los alumnos de Ingeniería Electrónica y Automatización puedan obtener datos estadísticos de estado de las baterías, mediante el uso de una interfaz gráfica en tiempo real de Matlab y un datalogger, para elaborar estrategias de combate que permitan mejorar su desempeño durante las competencias de batallas de robots.

#### **1.3.1 Delimitación Temporal:**

La puesta en práctica del proyecto técnico duró en un tiempo de 18 meses y completó en el mes de agosto del 2020.

#### **1.3.2 Delimitación Espacial.**

El prototipo del proyecto va ser aprovechado en el Laboratorio del Club de Robótica de Universidad Politécnica Salesiana de la Sede Guayaquil.



**Figura 1.** Delimitación espacial del prototipo del Proyecto Técnico Club de Robótica UPS-GUAYAQUIL. [1]

### 1.3.3 Delimitación Académica.

Este proyecto tiene un alcance académico basado en los parámetros y requerimientos específicos en los reglamentos oficiales que deben cumplir los robots de competencia profesional, además se muestra un análisis técnico de eficiencia energética, mediante el uso de una interfaz gráfica en tiempo real de Matlab y un datalogger, sustentando así el conocimiento adquirido en el pensum académico generando un vínculo entre los conceptos de mecánica, física aplicada, electrónica y programación.

**Innovación:** Los prototipos cuentan con un datalogger que almacena la potencia de los motores, para realizar el análisis de sus movimientos e impactos con el uso del arma para poder optimizar el uso de las baterías en competencias.

**Impacto:** Este proyecto técnico presenta las bases para el desarrollo de prototipos motivando así a los estudiantes de colegios y universidades a participar de evento relacionados con batalla de robots y estudiar carreras afines a la electrónica.

## 1.4 Objetivos.

### 1.4.1 Objetivo General.

Diseñar e implementar dos robots de batalla de 3 libras, utilizando una tarjeta de control para almacenar datos de energía eléctrica y generar estrategias para futuras competencias.

### 1.4.2 Objetivos Específicos.

- Diseñar e implementar dos robots de batalla de 3 libras.
- Elaborar planos mecánicos y eléctricos con las cotas respectivas.
- Implementar el control, la sabertooth y el variador de frecuencia E.S.C.
- Implementar la tecnología para control inalámbrico y desplazamiento de los robots.
- Implementar un datalogger que almacene variables de potencia en los motores de desplazamiento y arma.

## 2. FUNDAMENTACIÓN TEÓRICA.

En esta sección se describen los conceptos basados al enfoque del proyecto para su diseño e implementación.

### 2.1 Batalla de Robots

Consiste en una competencia de dos robots tele operados que compiten con el uso de sus armas mecánicas o neumáticas entre sí, en un recinto cerrado. La pericia y experiencia de sus constructores y operadores se exponen en dicho encuentro para determinar cuál es el ganador del combate.



*Figura 2. Robot Tutuchin (120lb) de UPS-GYE en Batalla de Robots CER2015.*



**Figura 3.** Reparaciones realizadas por equipo del Club de Robotica UPS-GYE.

En las figuras 2 y figura 3 muestran la participación del Club de Robotica UPS-GYE, en la categoría de robot de batalla de 120 libras del Concurso Ecuatoriano de Robótica en el año 2015, en este año se obtuvo en el segundo lugar a nivel nacional en la categoría, demostrando la funcionalidad y resistencia de Tutuchin, robot que fue elaborado por David Ulloa y José Quinde.

### 2.1.1 Historia de Battlebot

BattleBots es un evento que se basa en versión original de Robot Wars creación de Marc Thorpe en 1990.

Para el año de 1999, Trey Roski y Greg Munson fundaron en San Francisco. la liga BattleBots ([www.battlebots.com](http://www.battlebots.com)), creando la competencia con mayor exposición a los medios hasta hoy. El primer evento se realizó en Long Beach, California, en agosto de 1999, con 70 robots inscritos. [1]

El evento tuvo tres clases de peso: Kilobots (25-55 lbs con ruedas, 25-83 lbs con patas), Megabots (56-109 lbs con ruedas, 84-164 lbs con patas) y Gigabots (110-200 lbs con ruedas, 165-300 lbs patas). [2]

Los robots con patas eran estructuras de 450lb, como el impresionante Mechadon de 450 lb de Mark Setrakian, marcando así una desventaja para sus contrapartes con ruedas tradicionales. Pocos robots con patas entraron en la competencia debido la complejidad de su fabricación. [2]



**Figura 4.** Mechadon [2]

En 2000, BattleBots comenzó a ser televisado, debido a su popularidad, transmitiéndose durante 9 temporadas hasta la actualidad.

Cuenta con una cobertura en más de 150 países para que los fanáticos de los combates robóticos de todo el mundo puedan disfrutar del deporte de lucha con robots más grande y único del mundo en sus pantallas. [3]



**Figura 5.** Batalla de robots en Battlebots 2019 [3]



**Figura 6.** Equipo WanHoo (2019) [4]

### 2.1.2 Principales concursos de batallas de robots.

En la actualidad se promocionan eventos de robótica a nivel nacional e internacional con categorías que van desde seguidores de línea hasta las categorías de robots de batalla.

Entre las categorías que ofrecen estos eventos destacan las siguientes:

**Tabla # 1** Categorías de Batalla de Robots [5]

| Categoría         | Peso máximo del robot  |
|-------------------|------------------------|
| Batalla de robots | 120 LB                 |
|                   | 60 LB                  |
|                   | 30 LB                  |
|                   | 12 LB                  |
|                   | 3 LB                   |
|                   | 1 LB                   |
|                   | 0.5 LB                 |
|                   | Simulada (Robot Arena) |

Los equipos nacionales e internacionales se enfrentan en las batallas para obtener premios y la adjudicación de ser los mejores en su categoría.

Para este documento se elaboró una lista de los principales eventos a nivel nacional e internacional para motivar su interés para participar en la batalla de robots y así poner a pruebas sus futuros diseños y estrategias.

Para cada competencia las reglas podrían cambiar de acuerdo a las decisiones de los jueces, estos cambios pueden darse al solicitar el uso de protecciones para los operadores y asesores de los equipos. Es importante considerar el uso de gafas de protección y guantes anti corte para las competencias, aunque el reglamento no lo establezca.



**Tabla # 2 Principales concursos de Robotica a nivel nacional**

| <b>Nombre del concurso</b>       | <b>Lugar</b>  | <b>Provincia</b>        | <b>Posibles fechas</b>    |
|----------------------------------|---|-------------------------|---------------------------|
| Copa Utabot                      | Universidad Técnica de Ambato – Ambato                | Tungurahua              | Entre abril y mayo        |
| Robot Game ZI6                   | Quito   | Pichincha               | Entre mayo y junio        |
| Riotronic                        | Escuela Superior Politécnica de Chimborazo – Riobamba | Chimborazo              | Entre junio y julio       |
| UMEBOT                           | Escuela Politécnica Nacional – Quito                  | Pichincha               | Entre junio y Julio       |
| Concurso de Robótica ESPEL       | Universidad de las Fuerzas Armadas – Latacunga        | Cotopaxi                | Entre junio y Julio       |
| Kiltronic                        | Universidad Politécnica Salesiana – Guayaquil         | Guayas                  | Entre octubre y noviembre |
| COPOL Metal Challenge            | Unidad Educativa Particular Politécnica – Guayaquil   | Guayas                  | Entre Octubre y Noviembre |
| Concurso Ecuatoriano de Robótica | La sede cambia cada año                               | La sede cambia cada año | Entre Octubre y Noviembre |

En la tabla #2 se encuentran los concursos de robótica nacionales donde se pueden adquirir acreditaciones para concursos Internacionales de alto nivel entre los cuales se mencionan en la siguiente tabla.

**Tabla # 3 Principales concursos de Robotica a nivel internacional**

| <b>Nombre del concurso</b> | <b>Lugar</b>            | <b>País</b> | <b>Posibles fechas</b>      |
|----------------------------|-------------------------|-------------|-----------------------------|
| IRONCup                    | La sede cambia cada año | Brasil      | Entre febrero y marzo       |
| Runibot                    | Bogotá                  | Colombia    | Entre abril y mayo          |
| RobochallengeMx            | La sede cambia cada año | México      | Entre mayo y junio          |
| Guerra de Robots           | Ciudad de México        | México      | Entre junio y julio         |
| Robocore Winter Challenge  | La sede cambia cada año | Brasil      | Entre octubre y noviembre   |
| Tech Games Lima            | Lima                    | Perú        | Entre noviembre y diciembre |

En la tabla #3 se muestran los principales concursos a nivel internacional, el nivel de competencia es profesional, es necesario armar estrategias de combate debido a que los operadores cuentan con mucha experiencia.

## **2.2 Principales materiales para la construcción de robots**

Existe una gran variedad de materiales que se pueden usar para la estructura, coraza y arma de un robot de combate.

El acero y el aluminio se han vuelto populares para los componentes del marco, mientras que los materiales más utilizados para las armas son el aluminio, el acero y el titanio. [6]

La elección de los materiales depende primeramente del diseño estructural del robot, sus armas, sus componentes electrónicos de control y baterías, todo lo antes mencionado influye directamente con el peso del robot, a continuación, se mostrarán los principales materiales que podría utilizar para el diseño y la elaboración de su robot.

### **2.2.1 Metales**

Los metales son el material de elección para la mayoría de las estructuras y armaduras de robots, y existen numerosos tipos de metales disponibles para robots construcción. [7] Dependiendo del tipo, pueden ser extremadamente resistentes, sin embargo, su alta densidad haría un robot totalmente de acero muy pesado.

Los siguientes son algunos de los principales tipos de acero utilizados en los robots de combate.

#### **2.2.1.1 Acero 1018, acero 1020**

Tienen un bajo contenido de carbono, alrededor del 0,18% al 0,20% en peso respectivamente. Tienen poca resistencia. Por lo general, se usan en ejes y en una variedad de componentes. Se utilizan en la estructura del robot debido a su bajo costo, sin embargo, su bajo límite elástico los hace fácilmente flexible [8]

#### **2.2.1.2 Acero 1045**

Acero con contenido medio de carbono (0,45%), se utiliza cuando mayor resistencia y dureza se desea. Se utiliza en aplicaciones de alta velocidad, engranajes, ejes y piezas de máquinas. [8]

#### **2.2.1.3 Acero 4340**

Acero con 0,40% de carbono, con níquel además de cromo y molibdeno(cromoly), con una resistencia y dureza al impacto altas. Es un material que brinda excelentes características para la fabricación de ejes y armas del robot para alcanzar una tenacidad de alto impacto [8]

#### **2.2.1.4 Aceros 5260**

Acero con 0,60% de carbono, contiene cromo y manganeso. Llamado acero para muelles, tiene una excelente tenacidad al impacto [8]

### **2.2.1.5 Acero inoxidable**

Son aceros con más del 12% en peso de cromo, que forma una película protectora que evita la corrosión. tiene una excelente combinación de tenacidad al impacto y resistencia a la corrosión, es un buen material para la armadura del robot. [8]

### **2.2.2 Aleaciones de Aluminio**

El aluminio es un metal muy ligero, tiene aproximadamente 1/3 de la densidad de los aceros, lo que hace es muy atractivo para la estructura del robot. Las aleaciones de aluminio de las series 1000, 3000 y 5000 (por ejemplo, el aluminio 1050, utilizado en equipos eléctricos, el 3003, utilizado en utensilios de cocina, y el 5052, resistente a la corrosión) son de baja resistencia y no deben usarse en combate. [8]

### **2.2.3 Aleaciones de titanio**

El titanio es uno de los mejores materiales para robots de combate. Con poco más de la mitad de la densidad de aceros (entre 4,4 y 4,6), alcanza resistencias 2,5 superiores al acero 1020.

### **2.2.4 Plásticos de alta resistencia**

Cuando piensa en la durabilidad, probablemente piense primero en los metales. Sin embargo, algunos de los plásticos ofrecen muchas ventajas sobre los metales cuando se trata de construir robots para competición. [7]

Los plásticos suelen ser útiles para la elaboración de partes estructurales debido a su peso y la flexibilidad y forma de absolver los impactos producidos por otros robots.

#### **2.2.4.1 Policarbonato**

También conocido como Lexan, es un termoplástico polimérico, transparente a las ondas de luz y a las señales de radiocontrol. Tiene tenacidad de alto impacto, y es muy ligero, con densidad 1.2. Se utiliza en robot de combate. armadura, absorbe mucha energía ya que se deforma durante un impacto.

#### **2.2.4.2 PETG**

Es un tipo modificado de PET (tereftalato de polietileno) con tenacidad al impacto en entre los valores de acrílico y Lexan. Suele utilizarse en los robots de 3 libras para la elaboración de soporte de motores, soporte de receptores, soportes de baterías. Suelen utilizarse como piezas estructurales en robot de batalla de 1 libra.

#### **2.2.4.3 Teflón (PTFE, politetrafluoretileno)**

Tiene muy baja fricción, se puede utilizar como cojinete de deslizamiento para cargas moderadas, o como un patinazo debajo del robot para deslizarse en la arena.

#### 2.2.4.4 UHMW

El polietileno de peso molecular ultra alto es un polietileno de alta densidad que también tiene muy baja fricción.

Suele usarse para la elaboración de partes estructurales como: rampas, carcasas, patines.

#### 2.2.4.5 Nylon

Son polímeros termoplásticos con alta resistencia, baja densidad y dureza relativamente alta. Son buenos para espaciadores internos en los robots, e incluso como soportes de motor, de manera similar a UHMW.

Suele usarse en las partes estructurales de los robots, poleas, tapas, rampas, además suele utilizarse para de la fabricación de aros para ruedas.

#### 2.2.4.6 Fibra de carbono

Conocida como CFRP (fibra de carbono reforzada polímero), es extremadamente rígido y ligero, se utiliza en partes externas e internas del robot debido a su alta rigidez.

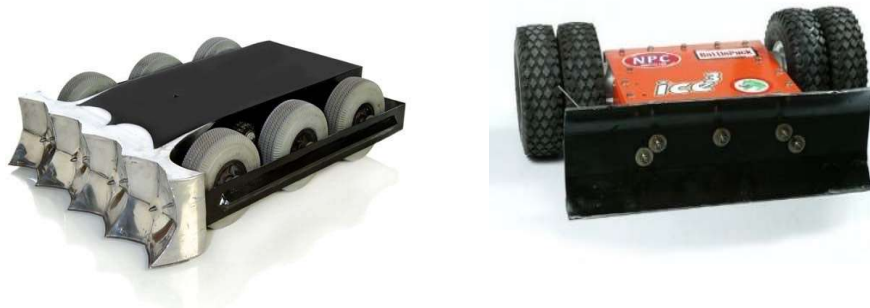
El material más resistente por volumen son los aceros, pero un robot fabricado íntegramente en acero sería muy pesado.

### 2.3 Tipos de robots de batalla.

Un robot con un buen diseño estructural, uso de armas y resistencia a golpes podría genera daño y vencer a otro robot, en la actualidad los robots de batalla se basan en sus componentes y su categoría de competencia por el peso.

Entre los tipos más comunes tenemos:

#### 2.3.1 Rammers



**Figura 7.** Robot tipo Rammers. [8]

Son robots de embestida, dañan a su oponente lanzándose hacia ellos o empujándolos a las esquinas de la arena, para generar su inmovilización.

Pueden tener entre 4 o más ruedas y pueden invertirse y seguir en competencia, suelen ser robustos y su estructura soportan muchos impactos.

### 2.3.2 Sawbots



**Figura 8.** Robot tipo Sawbots. [8]

Estos robots cuentan con discos dentados que giran a altas revoluciones por un potente motor el objetivo de su arma es atravesar la estructura de su oponente al embestirlo y atacarlo. Suelen tener entre 2 y 4 ruedas además cuentan con una coraza muy resistente a impactos además con rampas para hacer que, al impactarlos contra su contrincante esta suba hasta el arma para generar daño, pueden ser invertibles o no.

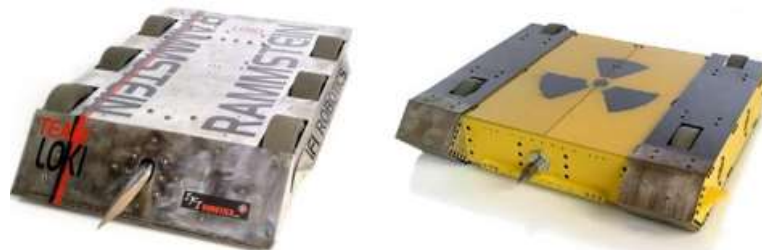
### 2.3.3 Wedges



**Figura 9.** Robot tipo Wedges. [8]

Son robots de embestida, pero cuentan con una rampa, tipo cuña para voltear a su oponente no suelen generar daños, pero puede conseguirla inmovilización y generar puntos. Por lo general tienen entre 2 o 4 ruedas con sistema de propulsión muy resistente. Pueden ser invertibles o no.

### 2.3.4 Spearbots



**Figura 10.** Robot tipo spearbots. [8]

Son robot de 4 o más ruedas cuentan con una armadura resistente a impactos y su arma es una punta que sobre sale de su estructura para trata de penetrar la coraza de su oponente.

### 2.3.5 Thwackbots



Figura 11. Robot tipo Thwackbots. [8]

Son robots de 2 ruedas, invertibles que giran su estructura en su propio eje a altas velocidades, suelen tener una varilla larga con un tipo de punta para perforar o tipo martillo para generar daño a sus oponentes al girar toda su estructura. Deben tener muy buena tracción en sus llantas para no perder velocidad y poder impactar a su oponente con mayor fuerza.

### 2.3.6 Thwackbots de arriba



Figura 12. Robot tipo Thwackbots de arriba. [8]

Son robots de dos ruedas con características similares a los Thwackbots, pero su arma cuenta con un arpón o punta vertical para generar daño a su oponente desde la parte superior de su estructura.

### 2.3.7 Horizontal Spinners



Figura 13. Robot tipo Horizontal Spinners. [8]

Son robots que generan un gran daño a sus oponentes debido a la placa de metal giratoria que cuentan en la parte superior de su estructura que al activarlas giran a altas velocidades, No son invertibles, y por lo general cuentan con dos ruedas.

### 2.3.8 Drumbots

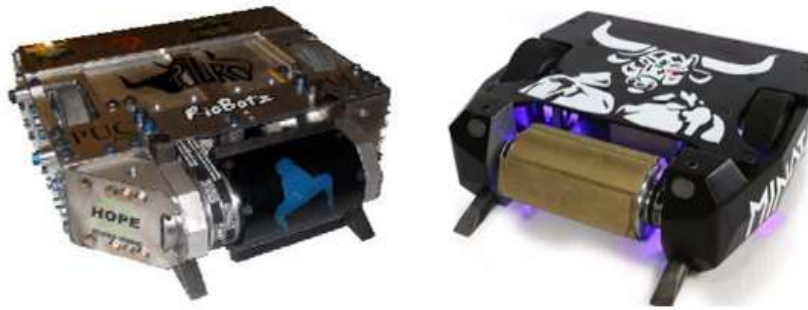


Figura 14. Robots tipo Drumbots. [8]

Estos robots tienen un tambor giratorio con dientes, que es accionado por bandas o cadenas, montado horizontalmente frente al robot. Suelen girar de tal forma que lanzan al oponente, volteándolo o provocando daños por el impacto con el arma o con el suelo. Los Drumbots son versiones más compactas de spinners verticales, con menos momento de inercia en el arma. [8]

### 2.4 Data Logger.

Data Logger o registrador de datos, son dispositivos electrónicos dedicados para registrar los datos en tiempo real tomados de las diferentes variables existentes en un proceso de control. El registrador puede contener sensores internos o externos para la adquisición de datos y pueden estar equipados con baterías para darle autonomía. [9]

Un data logger utiliza microprocesadores, están equipados con una memoria interna para almacenamiento de datos de sensores. Pueden almacenar la información y comunicarse de manera inalámbrica o alámbrica, enviando los datos en tiempo real para analizar los datos recogidos, mientras que otros tienen un dispositivo de interfaz local (teclado, pantalla LCD).

### 2.5 Batería de Lipo.

Una batería LiPo (Litio y polímero) es un tipo de batería recargable utilizada en las competencias de batalla de robots. Son muy recomendables ya que ofrecen unas prestaciones superiores a las NiCd y NiHm. [10]

Tienen una alta densidad de energía, alta velocidad de descarga y peso ligero, lo que las hacen ideales para aplicaciones de RC.



Figura 15. Batería de LIPO de 450MAh 7.4v 2S 75C. [10]

## 2.5.1 Significado de la nomenclatura de una batería LiPo

Las baterías de lipo cuentan con letras y símbolos que nos dan a conocer su capacidad de voltaje, de carga y de descarga. En la siguiente imagen se muestra las principales características que se debe analizar para adquirir una batería de acuerdo a su necesidad.



Figura 16. Nomenclatura de una batería LiPo.

### 2.5.1.1 Clasificación por número de celdas S

Las baterías LiPo cuentan con celdas conectadas en serie, cada celda tiene un voltaje nominal de 3.7 V. El número S se corresponde con el número de celdas. [10] El voltaje se define como la cantidad de celdas de la batería, también conocido como su número "S". así: [11]

- 1S = 1 celda = 3.7V
- 2S = 2 celdas = 7.4V

### 2.5.1.2 Capacidad indicada en mAh

La capacidad de una batería LiPo se mide en mAh (mili amperios hora) indica la corriente puedes consumir de la batería en una hora hasta que se descargue completamente. [11]

Por ejemplo, una batería de LiPo de 1300 mAh se descargaría completamente en una hora si se consume 1,3 A de ella. Si la corriente extraída se doblase (2.6 A), el tiempo se reduciría a la mitad.

### 2.5.1.3 Tasa de descarga, número C

Es la capacidad de la batería, con esta información podemos calcular en teoría cual es la descarga máxima segura que se puede realizar de forma continua de una batería LiPo.

$$\text{Máxima Descarga} = C \text{ Rating} * \text{Capacidad}$$

Por ejemplo, una batería de 1000mAh 1C se descargaría a 1A (amperio) en una hora. Si el número C es distinto de 1 significa que multiplicamos la tasa de descarga por un valor, reduciendo el tiempo de descarga proporcionalmente. [10]

Por ejemplo, Una batería de 1000mAh 2C se descargaría a 2A en media hora



### 3 MARCO METODOLÓGICO

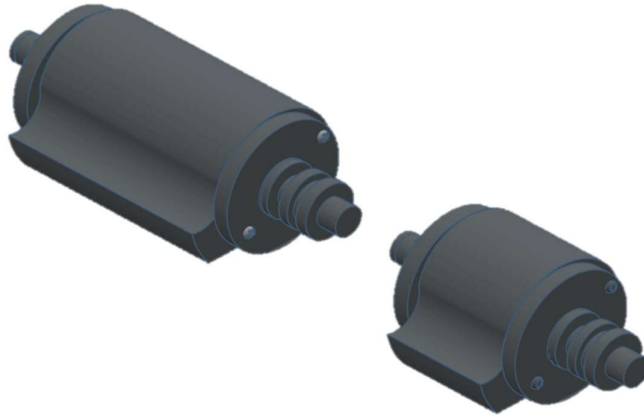
En esta capítulo veremos y conoceremos un poco más a detalles los diferentes módulos que emplearemos a lo largo del desarrollo de este proyecto técnico.

#### 3.1 Diseño Mecánico.

##### 3.1.1 Arma completa.

Las armas de los robots fueron construidas y moldeadas en acero 4340 recocido o también conocido como acero JIS SNCM439. Este tipo de acero está formado por una baja aleación de níquel-cromo-molibdeno que sumado al tratamiento recocido obtiene como resultado propiedades tales como: alta resistencia a la corrosión atmosférica, resistencia a la fatiga, alta resistencia y tenacidad.

Por sus excelentes propiedades de dureza se ha seleccionado el acero 4340, que como consecuencia del choque directo del filo tipo cuchilla y girando a altas revoluciones generen daños de consideración al robot oponente.



**Figura 17.** Armas del Robot 1 y Robot 2.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 1A y 1B de planos estructurales

##### 3.1.2 Rodamiento.

En los robots el rodamiento fue utilizado para sostener el eje del tambor del arma con el fin de disminuir el rozamiento entre las bases de apoyo y los ejes ya mencionados. Compuesto generalmente dos piezas rígidas denominadas pista externa, pista interna, y las piezas que rotan libre en medio de las pistas denominadas rodillos o bolas. [12]



**Figura 18.** Rodamiento rígido de bolas. [12]

Para los robots de batalla se utilizó el rodamiento de bolas cuya numeración es 608 ZZ, el cual es un rodamiento típico para cargas radiales y axiales, provisto de dos tapas metálicas antipolvo, jaula de acero, capaz de trabajar a altas velocidades.

### 3.1.3 Estructura del robot.

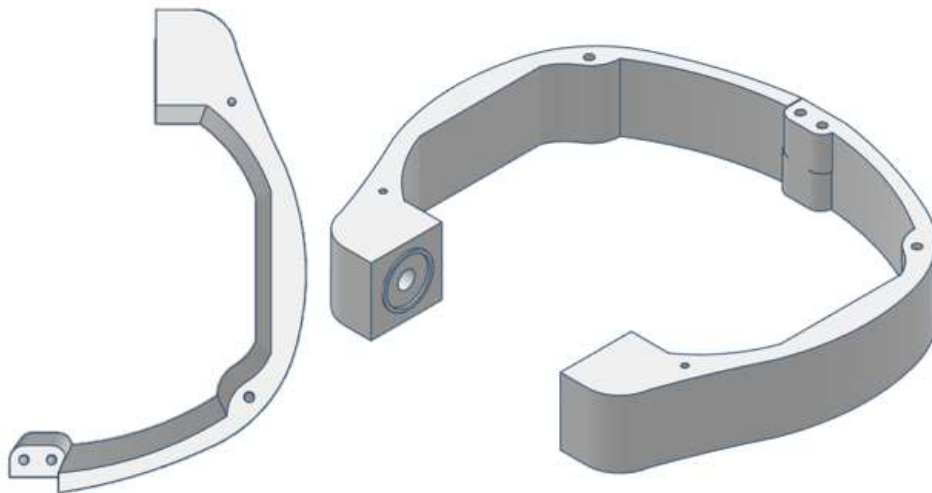
La estructura que conforma los bordes externos del robot está fabricada en polipropileno de alta densidad es un termoplástico formado por múltiples unidades de etileno.

El polipropileno de alta densidad dado sus propiedades físicas es un componente ideal para la construcción de robots de batallas de categorías de pesos bajos, y entre muchas de sus atributos tenemos: muy buena resistencia térmica y química, alta resistencia al impacto, muy ligero, fácil de procesar, facilidad para imprimir, excelente flexibilidad.



**Figura 19.** Planchas de polipropileno de alta de densidad. [13]

En los robots se utilizaron planchas polipropileno moldeadas en una CNC dando lugar a dos partes simétricas que forman los bordes exteriores del robot, las mismas que sirven de bancada para los rodamientos donde se apoya el arma.



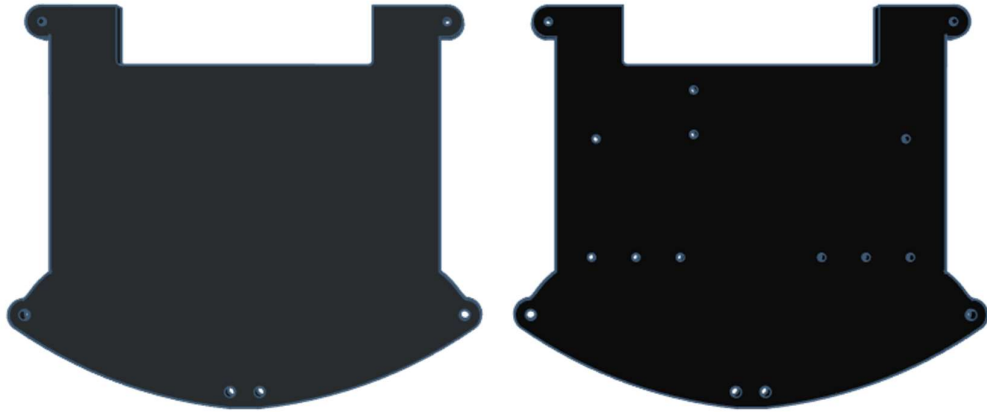
**Figura 20.** Piezas de la estructura del robot 1 y robot 2.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 2 A y 2 B de planos estructurales.

### 3.1.4 Tapas superior e inferior de los robots.

Las tapas fueron elaboradas en fibra de carbono la cual ofrece una alta resistencia a agentes externos, gran resistencia mecánica y un módulo de elasticidad elevado, resistente a las variaciones de temperatura y buena memoria mecánica.

La fibra de carbono reduce el peso de los robots, estas tapas fueron cortadas de acuerdo al diseño propuesto e importadas desde Florida, Estados Unidos.

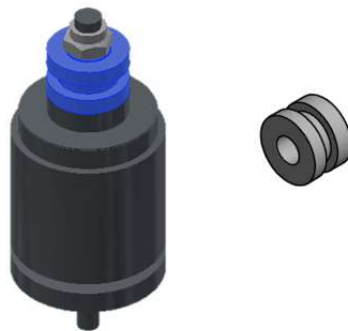


**Figura 21.** Tapas superiores e inferiores de los robots.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 3 A y 3B de planos estructurales tapa superior e inferior.

### 3.1.5 Poleas del arma y motor

Estos componentes fueron fabricados en duralón o también conocido como Nylon. Por sus características físicas y mecánicas hacen que este polímero se use en diferentes aplicaciones industriales tales como rodamientos de Nylon, poleas, engranajes, levas etc.



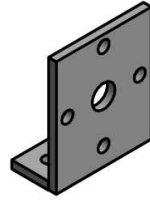
**Figura 22.** Poleas para el motor brushless.

Como sistema de transmisión desde la polea del motor hacia la polea del arma se utilizó un oring tipo B19, 3mm x 30mm.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 4 de planos estructurales.

### 3.1.6 Soporte de motores brushless de arma

Para la elaboración del soporte del motor brushless del arma se utilizó aluminio de 3mm de espesor.



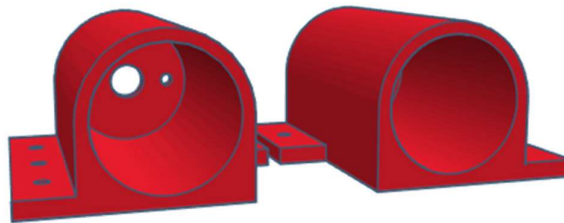
**Figura 23.** Soporte del motor de arma.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 5 de planos estructurales.

### 3.1.7 Soporte de motores.

Para los soportes de motores se utilizó Acido Poli láctico, más conocido por su abreviatura PLA. Es un polímero biodegradable, producido a partir del ácido láctico residual de la fermentación de cultivos tales como el maíz, entre otros.

Es un material altamente usado en el mundo de la impresión 3D debido a su relativa baja temperatura de fundición la cual se encuentra en el rango de 180 a 220 grados Celsius.

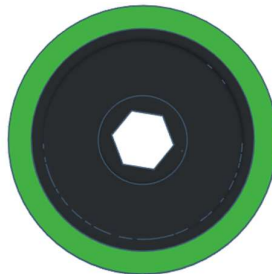


**Figura 24.** Soportes de derecha e izquierda de motores DC.

Para más detalles con respecto a las cotas de diseño, revisar el anexo 6 de planos estructurales.

### 3.1.8 Llantas de robots

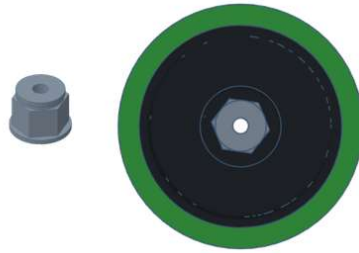
Se utilizaron en ambos robots llantas bannebots T40P-24nn 2 3/8.



**Figura 25.** Llantas Bannebots T40P 2 3/8.

### 3.1.9 Acople para motor y llantas

Se utilizaron en ambos robots acoples comerciales Bannebots T40H-Sbb1

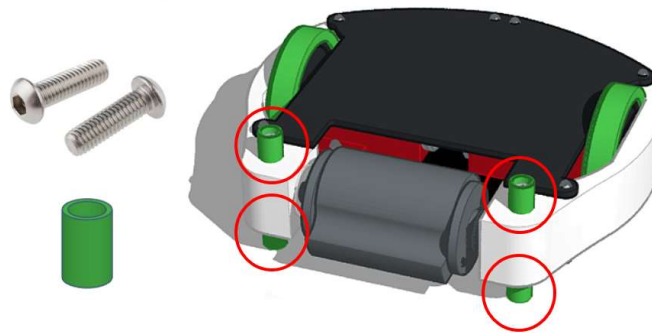


**Figura 26.** Acople de motor Bannebot T40H – Sbb1.

### 3.1.10 Patines del robot 1

En el robot 1 se utilizaron como patines dos tornillos cabeza de botón de acero inoxidable tipo hexagonales de medida M5 x 40mm.

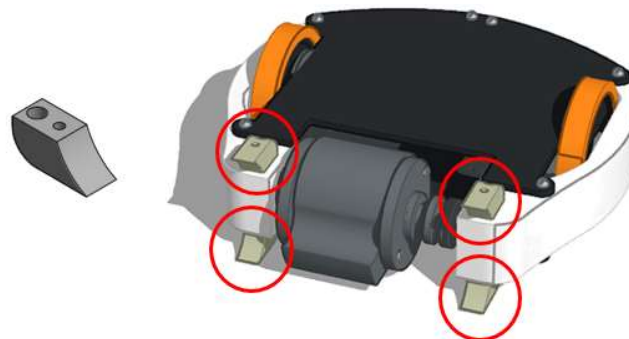
Se utilizó como recubrimiento una pieza impresa en PLA, para reducir la fricción con el piso.



**Figura 27.** Patines del robot 1 para mantener la altura del arma y reducir la fricción.

### 3.1.11 Patines del robot 2

En el robot 2 se utilizaron como patines dos piezas de aluminio de 10mm modeladas en una CNC.

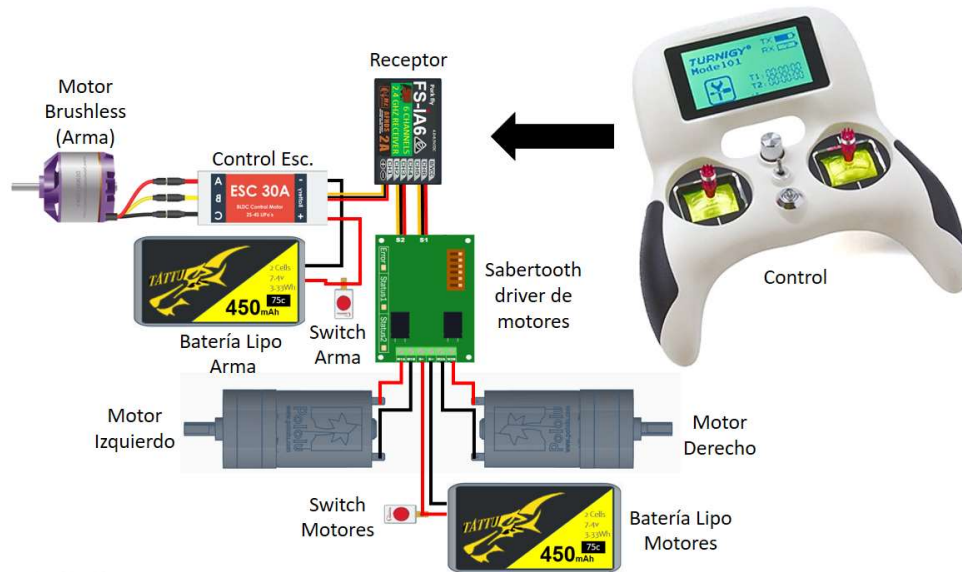


**Figura 28** Patines del robot 2 para mantener la altura del arma y como puntas para causar daño a su contrincante.

Para más detalles sobre las cotas de diseño, revisar el anexo 7 de planos.

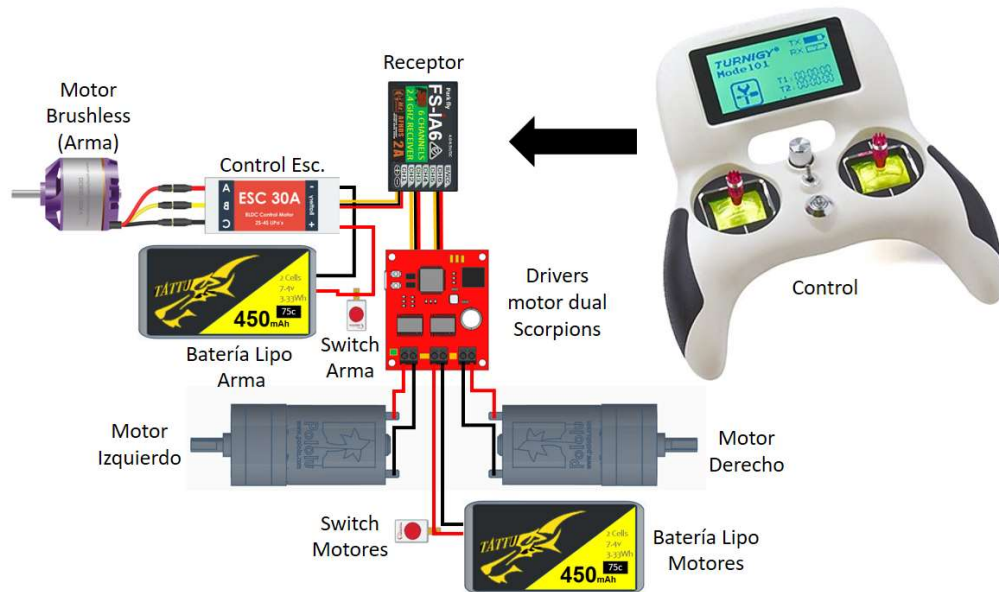
### 3.2 Diseño eléctrico de los robots de batalla

Los robots serán manipulados por unos mandos inalámbricos. La arquitectura tecnológica se la detalla en las siguientes imágenes.



**Figura 29.** Diagrama de conexiones del Robot 1.

En la figura 29 podemos observar el diagrama de conexiones del Robot 1, en este robot se utilizó como controlador de motores la sabertooth 2x12A

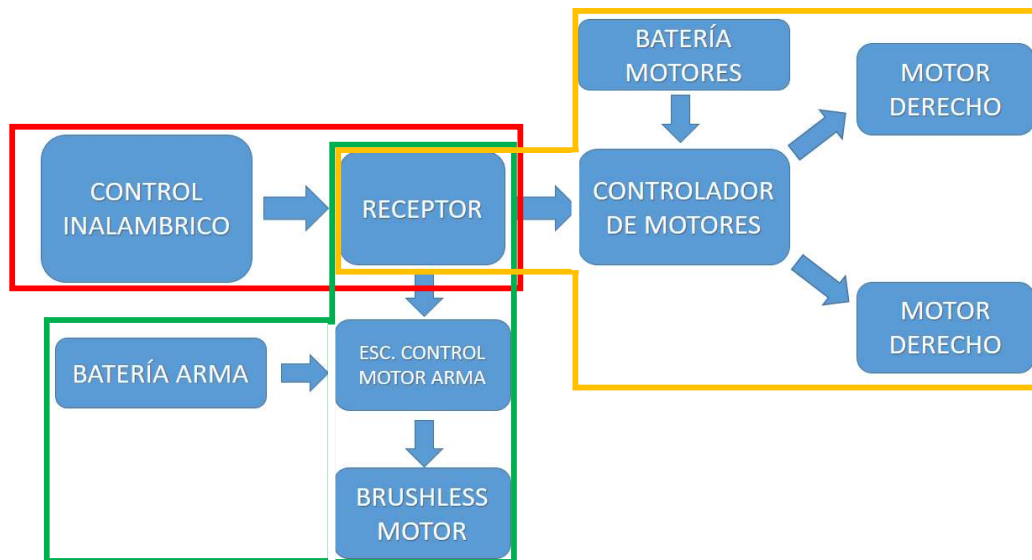


**Figura 30.** Diagrama de conexiones del Robot 2.

En la figura 30 podemos observar el diagrama de conexiones del Robot 1, en este robot se utilizó como controlador de motores Scorpion Dual de 6.5A. En el anexo 8 se detallan los planos eléctricos de conexiones.

### 3.2.1 Diagrama de flujo del funcionamiento en modo batalla

En el siguiente diagrama de flujo se muestra el funcionamiento de los robots.



**Figura 31.** Diagrama de flujo de funcionamiento de los robots en modo Batalla.

En la figura 31 muestra el funcionamiento del robot en modo batalla a continuación se detallan sus secciones.

- Sección de color rojo: El operador maniobra el robot utilizando el control inalámbrico(emisor) que envía mediante onda de radio los datos al receptor.
- Sección de color verde: El receptor recibe la señal del emisor y emite una señal PWM de 50Hz al ESC con una variación de 1ms a 2ms para controlar la velocidad del motor brushless del arma. Utiliza una batería independiente.
- Sección de color naranja: El receptor recibe la señal del emisor y envía una señal de modulación por ancho de pulsos al controlador de motores para generar movimientos como el avance, retroceso, giro a la derecha, giro a la izquierda del robot. El controlador de motores cuenta con una batería independiente y proporciona la energía al receptor.

### 3.3 Componentes electronicos de los robots

#### 3.3.1 Turnigy Evolution PRO Digital

Los robots son teleoperados a través de radio controles, en el robot 1 y 2 se implementó el Turnigy Evolution Pro Digital debido a las siguientes características detalladas:

- Protocolo RF AFHDS 2A
- Compatible con la gama de receptores iA6, iA6B, iA6C, iA10B.
- Radio de 8 canales de 2.4ghz con capacidad de telemetría.
- Palancas iluminados (ajustables) con resortes más rígidos.
- Pantalla táctil fácil de usar y LCD retroalimentada.
- Soporte SBUS / IBUS y CPPM / PW.



**Figura 32.** Control Turnigy Evolution PRO Digital. [14]

En la figura 32 se muestra el control que se implementó para ambos robots, este control muestra características necesarias para su uso en una competencia a nivel profesional además esta presentación cuenta con la opción de almacenar las direcciones de los receptores de fácil configuración, para que pueda ser utilizado como control también en el caso de que otro robot lo necesite.

### 3.3.2 Receptor FS-IA6B

Para recibir los datos transmitidos por el control es necesario implementar un receptor, en este proyecto se utilizó en ambos robots el FS-IA6B por sus características detalladas a continuación.

- Canal PWM: 6.
- Protocolo inalámbrico: AFHDS 2A.
- Alcance: 500 ~ 1500m (en el aire)
- Tipo de antena: antena de tubo de cobre dual (150 mm \* 2).
- Energía: 4.0-8.4V.
- Puerto de datos: PWM / PPM / i.bus / s.bus.



**Figura 33.** Receptor FS-IA6B de 6 canales de 2,4Ghz. [15]

En ambos robots se implementó el receptor FS-IA6B, debido primeramente a la cantidad de canales disponibles, en cada robot se utilizaron con el controlador de motores dos canales y con el variador de velocidad (E.S.C) solo un canal.

El peso y tamaño son factores esenciales en la construcción de un robot de batalla, el peso de este receptor es de 10g y sus dimensiones 47 \* 26,2 \* 15 mm, su fácil configuración y su actualización actualización con el emisor (control) hacen que sea el ideal para un robot de batalla de 3 libras.



### 3.3.3 Mini interruptor de encendido FingerTech

Es necesario tener interruptores para el control total de los robots para evitar accidentes y cumplir con el reglamento de las competencias, es por esa razón que en los robots 1 y 2 se utilizaron los interruptores de encendido FingerTech debido a uso profesional en las competencias se detallan las siguientes características:

- Material: Nylon de alta temperatura, terminales de cobre, tornillo de aluminio
- Dimensiones: 12.7 x 12.7 x 6.35 mm.
- Terminales de cobre: 3,5 mm x 1 mm.
- Peso: 2.15gr.
- Corriente nominal: 40 A.



Figura 34. Mini interruptor de encendido FingerTech. [16]

### 3.3.4 E.S.C T-Motor F 35A 32bit 3-6S

El ESC (controlador electrónico de velocidad), controla el motor Brushless generando el intercambio de polaridades en los embobinados del motor, el control se da mediante la modulación de ancho de pulsos que son suministrados por el receptor. Los ESCs de la serie F tienen drivers PWM con respuesta rápida. Integran Dshot 300/600/1200.

Las características principales se muestran a continuación:

- Firmware: BLHeli\_32.
- Admite señal PWM normal, Oneshot125, Oneshot42, MultiShot, D-Shot 1200/600/300 señal del acelerador.
- BEC: No.
- Li-Po: 3 ~ 6 S.
- Corriente continua : 35A – Max 45 A.
- Peso: 8,1 g (cable incluido).
- Dimensiones: 30 mm (largo) x 16 mm (ancho) x 5 mm (alto).

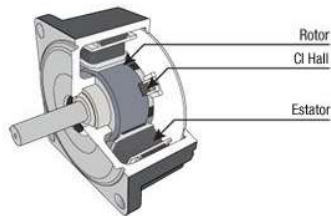


Figura 35. Control de Velocidad electrónica T-Motor F 35. [17]

Los motores del arma al romper la inercia y estar en combate generan picos altos de consumo de corriente, la implementación de un ESC con la capacidad para soportar el flujo constante de valores altos de corriente, sirve para controlar el motor y su velocidad para poder así activar el tambor del arma en el momento que sea necesario y de una manera controlada.

### 3.3.5 Motores brushless

Los motores sin escobillas o brushless no incorporan colector ni escobillas para cambiar la polaridad en el rotor; la conmutación de las bobinas se realiza electrónicamente a través de un controlador de motor. [17]



**Figura 36.** Motor sin escobillas. [18]

Se implementó para el robot 1 el brushless propdrive v2 5060 y para el robot 2 el brushless T-Motor F80 Pro 2500KV, a continuación, se detallan las características principales para cada motor:

**Tabla # 4 Comparación de motores brushless**

| Brushless            | Propdrive v2 5060 | T-Motor F80 Pro 2500KV |
|----------------------|-------------------|------------------------|
| Kv                   | 3000.00           | 2500.00                |
| Potencia ( W )       | 755,00            | 706.20                 |
| Corrientes max ( A ) | 55,00             | 47,60                  |
| Max Voltaje ( V )    | 14,80             | 14,00                  |
| Peso ( g )           | 85,00             | 36,40                  |



**Figura 37.** Prop Drive 28-36 3000KV / 755W. [19]



**Figura 38.** T-Motor F80 PRO - Motor de 2500KV / 706 W. [20]

Estos motores fueron implementados en los robots junto a un sistema de poleas que se adjunta en el anexo 4, los dos son motores de alta potencia para que el arma pueda girar a altas revoluciones y puede generar mucho daño a su oponente.

El motor Prop Drive 28-36 3000KV / 755W realizó pruebas de funcionamiento en Queretaro en el Congreso de Mecatrónica México, el robot participó del evento y se comprobó que el motor cumple con lo requerido para habilitar un arma en las batallas.

### 3.3.6 Controladora de motores

Un controlador de motor DC es un dispositivo que funciona junto con un microcontrolador. La mayoría de los controladores tiene protección contra subtensión, exceso de tensión, cortocircuito, protección de límite de corriente, protección térmica y transitorios de tensión. [21]

Se implementó para el robot 1 la Sabertooth de 2x12A y para el robot 2 Scorpion Mini Dual 2x6.5 A, a continuación, se detallan las características principales de cada controlador:

**Tabla # 5 Comparación de controladores de motores**

| Controladores               | Sabertooth de 2x12A R/C | Scorpion Mini Dual 2x6.5A |
|-----------------------------|-------------------------|---------------------------|
| Corriente max (A) por canal | 12,00                   | 8,00                      |
| Max Voltaje ( V )           | 24,00                   | 28,00                     |
| Peso (g)                    | 43,00                   | 24,00                     |
| Tamaño (mm)                 | 38,10x50,8              | 50,80x38,10               |



**Figura 39.** Controlador de motor Sabertooth dual 12A para R / C [22]

En la figura 39 se muestra la Sabertooth 2x12A que permite controlar dos motores con: voltaje analógico, radio control, serial y serial paquetizado, Esta edición especial R / C de nuestro controlador de motor viene con opciones para control exponencial, autocalibración y mezcla incorporada. [22]



**Figura 40.** Scorpion Mini Dual 2x6.5A. [23]

En la figura 40 se muestra la Scorpion Mini está diseñada para incluir la mayor capacidad de manejo de potencia, flexibilidad de control, la corriente completa y limitación de sobretensión, los motores y las baterías están protegidos contra condiciones de cortocircuito. [23]

### 3.3.7 Motores 20.4:1 Metal Gearmotor 25Dx50L mm HP

Este motorreductor consta de un motor de CC cepillado de 12 V de alta potencia combinado con una caja de engranajes de dientes rectos metálicos de 20,4: 1. [24] Las características principales de este motor se detallan a continuación:

- Velocidad sin carga a 6v: 250 rpm.
- Corriente de bloqueo a 6 V: 2800 mA.
- Peso: 85g.
- Tamaño: 25D x 50L mm.
- Par de bloqueo a 6 V:42 oz.
- Voltaje nominal:12V.



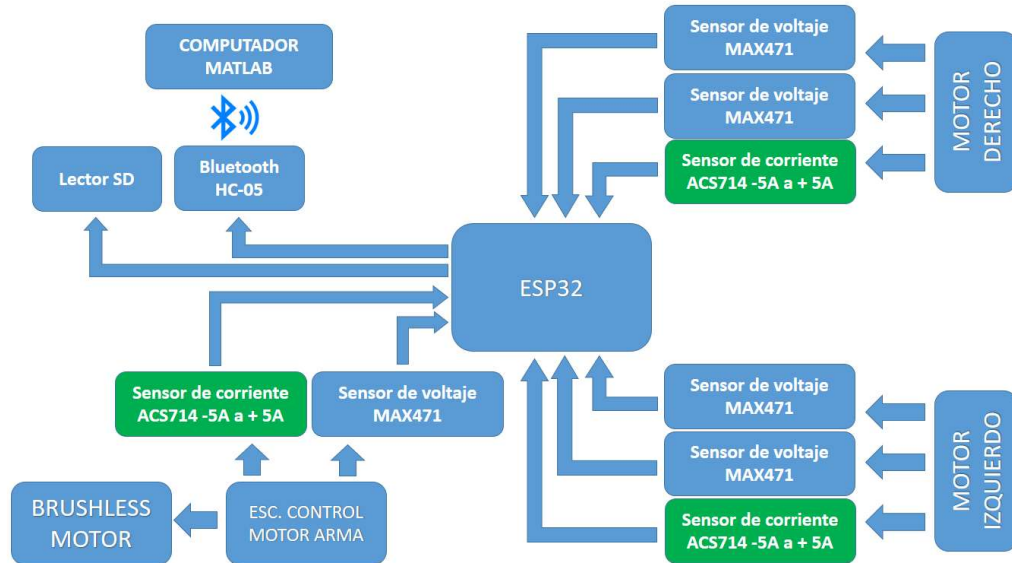
**Figura 41.** Motorreductor Metálico 25Dx50L. [24]

Se implementaron estos motorreductores a los dos robots, debido a que son motores de alto potencia con un consumo bajo de corriente, además nos ofrecen la opción de agregarles baterías de 12 V para adquirir mayor velocidad, pero esto se limita al peso y tamaño de robots. Son utilizados a nivel profesional por competidores internacionales debido a su peso de 85g.

### 3.4 Datalogger de potencia de motores

#### 3.4.1 Diagrama de flujo del funcionamiento del datalogger

En el siguiente diagrama de flujo se muestra el funcionamiento del datalogger en los robots.



**Figura 42.** Diagrama de flujo de funcionamiento del datalogger en los robots.

En la figura 42 se muestra el diagrama de funcionamiento del datalogger, para cada motor (derecha e izquierda) se utilizaron dos sensores de voltaje MAX741, debido a que el sensor mide el voltaje en un solo sentido que ingresa a sus bornes, al existir inversión de giro en los motores de necesario colocar otro sensor de voltaje. Para medir la corriente en ambos motores se utilizó un sensor que mide entre -5 A y +5 A.

Para obtener los datos de consumo de voltaje y corriente en el motor del arma, se utilizaron un sensor del voltaje Max741 y un sensor de corriente de -75 A a +75A. Todos los datos son normalizados y escalados en la ESP32, para luego ser almacenados en la lectora de Micro SD además se realizó el envío de datos por bluetooth hacia Matlab para generar las gráficas de potencia en cada uno de los motores en tiempo real.

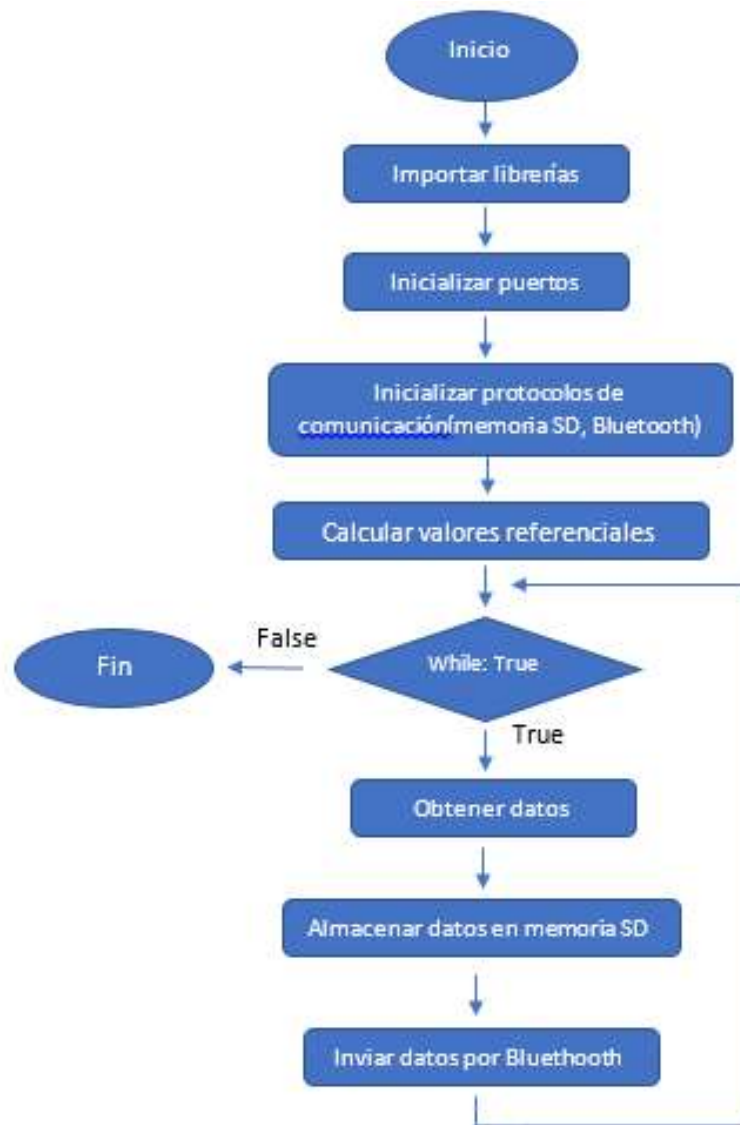
En Matlab se muestran los valores de las potencias de los motores, en tiempo real y la captura de video del robot para observar los movimientos para realizar los análisis con la gráfica en la misma interfaz.

Para realizar un análisis más detallado sobre el voltaje y corriente consumido por los motores en la microSd se guardan un archivo para cada voltaje y corriente en cada uno de los motores, estos archivos se deben cargar en Matlab mediante una interfaz para el graficar los datos.

En el anexo 8 se detallan los planos eléctricos de conexiones.

### 3.4.2 Diagrama de flujo de la programación del datalogger

A continuación se muestra el diagrama de flujo de la programación del datalogger:



**Figura 43.** Diagrama de flujo de la programación del datalogger en los robots.

En la figura 43 se muestra el diagrama de flujo que se utilizó para desarrollar la programación de la ESP32, el software que se utilizó fue el IDE de Arduino.

Se utilizaron 8 puertos analógicos de la ESP32, el puerto serie y SPI, se crearon dos librerías para la lectura de los sensores, esto reduce las líneas de procesamiento en la ESP32, luego se procede a enviar la trama de datos.

Se realizó dos programas uno para la Sabertooth y otro para la Scorpion debido a las conexiones que se realizaron para cada controlador, para verificar las conexiones revisar el anexo 8 del diagrama eléctrico.

En el anexo 9 se muestra el código principal que se utilizó en las ESP32.

### 3.5 Componentes electronicos del datalogger.

#### 3.5.1 Placa de desarrollo Node MCU ESP32

El ESP32 es un sistema de doble núcleo con dos CPU Xtensa LX6 de 32 bits de Harvard Architecture compatible con arduino. [25]

Las características principales de un ESP32 se detallan a continuación:

- Velocidad de reloj : Entre 160 Mhz y 240 Mhz
- 520 Kb de RAM
- Wifi integrado: Acces point & Station y Bluetooth 4.2 2.4 Ghz; BT 2.0 y 4.0
- 16 x Analog-to-Digital Converter (ADC) de 12 bits de resolución
- 2 x Digital to Analog converter DAC de 8 bits.
- 2 x UART o puertas serie y 2 x I2C channels y 4 x SPI channels.



**Figura 44.** Placa de desarrollo Node MCU ESP32. [26]

Este módulo se cargó la programación implementada para la lectura de las variables de corriente y voltaje procedentes de los 2 motores de tracción y motor del arma.

#### 3.5.2 Bluetooth HC-05

Es un módulo Bluetooth que puede conectarse al puerto serie de un microcontrolador, para que se comunique con otros dispositivos a través de una conexión Bluetooth. El módulo en sí puede ejecutarse tanto en modo maestro como esclavo. A continuación se detallan las características principales del módulo.

- Voltaje de suministro: 3,3 V a 6,0 V.
- Voltajes de funcionamiento: 3,3 V (todos los demás pines, excepto VCC).
- Corriente de trabajo: 30 mA.
- Rango de operación: máx. 10 m.
- Tasa de baudios admitida: 9660, 19200, 38400, 57600, 115200, 230400.



**Figura 45.** Modulo Bluetooth HC-05. [27]

Este modulo se utilizó para realizar la comunicación inalámbrica entre el módulo ESP32 y matlab para enviar la cadena de datos y graficar en tiempo real los valores de las potencias de los motores de tracción y arma.

### 3.5.3 Lector SD o micro SD

Un lector SD es un dispositivo que nos permite emplear como almacenamiento de datos una tarjeta SD. La lectura se hace a través del puerto SPI de cualquier micro. Aunque pueden disponer de otras interfaces, como bus I2C o UART, normalmente es preferible emplear SPI por su alta tasa de transferencia. [28]



**Figura 46.** Lectora de micro SD. [28]

Para guardar la cadena de datos obtenidos por las diferentes gamas de sensores de corriente y voltaje se consideró el uso del Módulo SD de ARDUINO. Una vez que el data logger sea energizado inmediatamente el módulo programable ESP32 escribirá los datos monitoreados por los sensores de voltaje y corriente en la tarjeta microSD.

### 3.5.8 Batería 9V

Para energizar el la ESP32 y los instrumentos acoplados a nuestro Data Logger se dispuso el uso de alimentación eléctrica a través de una batería de 9 voltios, con el objetivo de brindar autonomía de movimiento al robot cuando se encuentre en la fase de prueba.



**Figura 47.** Batería de 9v. [29]

En la figura 52 observamos un dibujo en 3D de una batería de 9 voltios.

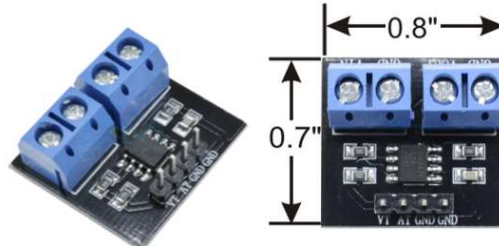
### 3.5.4 Sensor de voltaje y corriente MAX471

El MAX471 es un sensor de voltaje y corriente de alta precisión. La medición de voltaje está basada en un divisor de tensión que reduce la tensión de entrada en un factor de 1/5. [30]



A continuación se detallan las características principales del módulo.

- Voltaje mínimo: 3 V.
- Voltaje máximo: 36 V.
- Corriente de medición: -3A a +3A.
- Peso: 1,3 g.



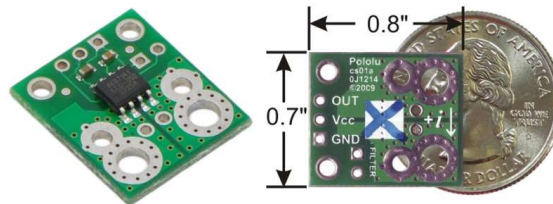
**Figura 48.** Max471 sensor de voltaje y de corriente. [30]

Para el datalogger se utilizó 5 sensores de tensión y corriente MAX471 por cada robot, 1 para el voltaje del arma y 2 por cada motor de tracción. Para los motores de tracción se usaron dos sensores de tensión MAX471 debido a que al ser voltaje DC el driver que controla el movimiento hacia los motores cambia de polaridad en sus terminales de conexión de salida, lo que ocasiona que los sensores que toman la lectura de voltaje cuando el motor gira en un sentido, no sea capaz de medir voltaje cuando gira en el sentido contrario.

### 3.5.5 Sensor de corriente ACS714 -5A a + 5A

Es un sensor de corriente lineal basado en el efecto Hall de  $\pm 5A$  ACS714 de Allegro de corriente de baja resistencia ( $\sim 1.2 \text{ m}\Omega$ ) y aislamiento eléctrico de hasta 2.1 kV RMS. [31] A continuación se detallan las características principales del sensor:

- Escala de conversión:  $0,185 \text{ V / A}$ .
- Voltaje lógico mínimo: 4,5 V.
- Voltaje lógico máximo: 5,5 V.
- Corriente de suministro: 13 mA.
- Peso: 1,3 g.
- Corriente máxima: 5A.



**Figura 49.** Sensor de corriente ACS714 -5A a + 5A. [31]

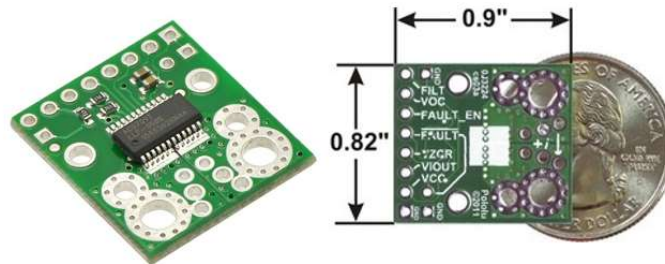
El sensor tiene una entrada de corriente bidireccional con una magnitud de corriente de -5 A y + 5A máximo generando una tensión analógica proporcional ( $185\text{mV/A}$ ). Se utilizó estos sensores para medir la corriente que pasa a través de los motores de tracción en pruebas con cargas el pico máximo de amperaje fue de 4,5 A.

### 3.5.6 Sensor de corriente ACS709 -75A a + 75A

Es un sensor de corriente lineal basado en el efecto Hall de  $\pm 75A$  ACS709 de Allegro de corriente de baja resistencia ( $\sim 1.1 \text{ m}\Omega$ ) y aislamiento eléctrico de hasta  $2.1 \text{ kV RMS}$ . [32]

A continuación se detallan las características principales del sensor:

- Escala de conversión:  $0.018 \text{ V/A}$ .
- Voltaje lógico mínimo:  $3 \text{ V}$ .
- Voltaje lógico máximo:  $5,5 \text{ V}$ .
- Corriente de suministro:  $11 \text{ mA}$ .
- Corriente máxima:  $75A$ .
- Peso:  $1,7 \text{ g}$ .



**Figura 50.** Sensor de corriente ACS709 -75A a + 75A. [33]

Para medir la corriente del motor del arma seleccionamos el sensor de corriente ACS709, ya que presenta un alto rango de medición de  $-75$  a  $+75$  amperios, ideal para soportar la carga de corriente que genera el arma al operar.

## 4. Resultados

En esta sección analizamos los resultados obtenidos en las pruebas hechas durante el desarrollo de este proyecto técnico.

### 4.1 Análisis Estructural

Para el análisis estructural se utilizó el software solidworks en donde a través de un modelado de las partes en 3D, analizamos las piezas principales que conforman la estructura del robot tal como son: armas, estructura lateral derecha, estructura lateral izquierda, con el objetivo de verificar las deformaciones máximas y puntos más susceptibles cuando son expuestas a tensiones.

#### 4.1.1 Análisis de Arma de robot #1.

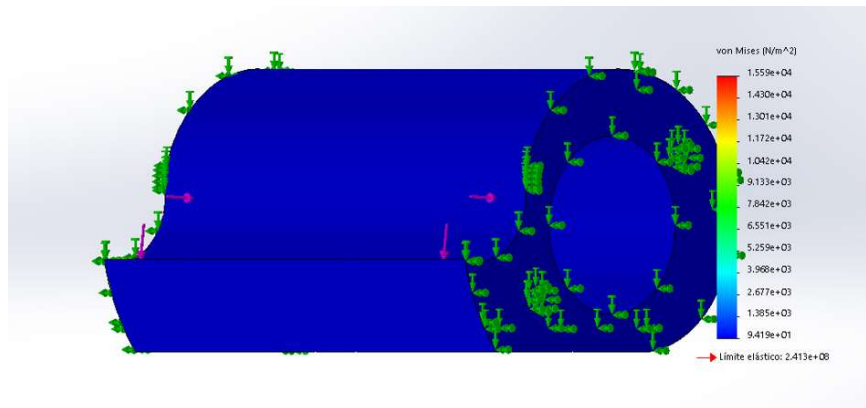


Figura 51. Modelado 3D de arma 1 visto desde solidworks.

En la figura 51 podemos observar el arma del robot 1 visto desde solidworks, la misma que en base a su material de fabricación (acero 4330) tiene un límite elástico de  $2.413e+08$  Pascal. Para nuestro análisis tomamos dos puntos estáticos, los cuales están indicados con las flechas de color verde, y aplicaremos una serie de tensiones ortogonales en el sentido que indican las flechas moradas, las cuales van en contra del movimiento del arma.

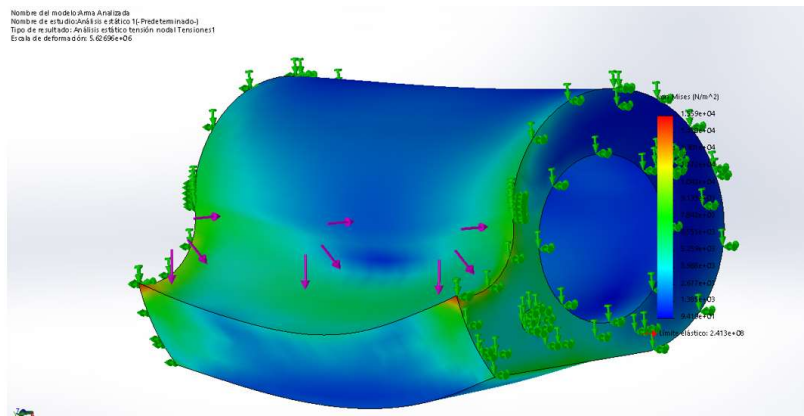
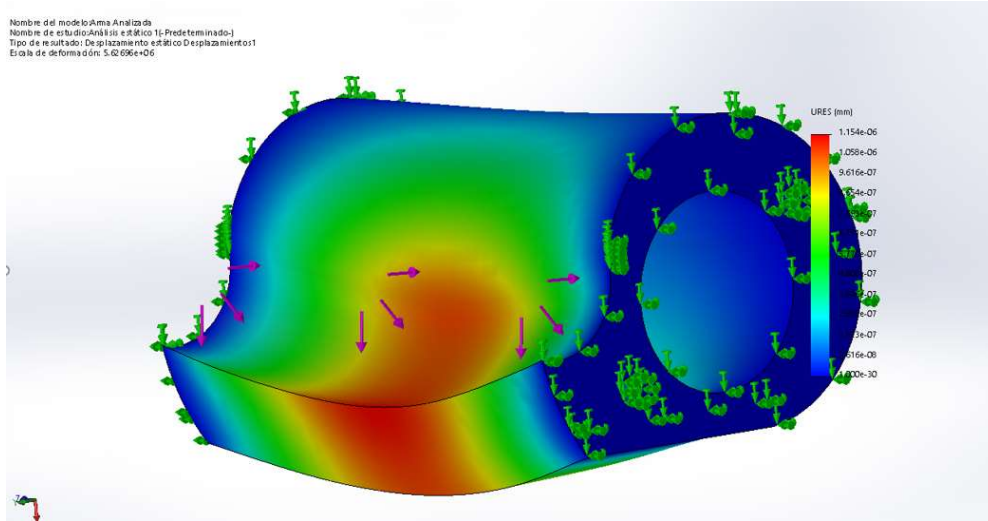


Figura 52. Arma de robot 1 deformada sometida a tensión.

En la figura 52 podemos visualizar los puntos susceptibles de deformación y el valor de la tensión aplicada medida en pascal. Se observa con el color rojo los puntos más vulnerables.

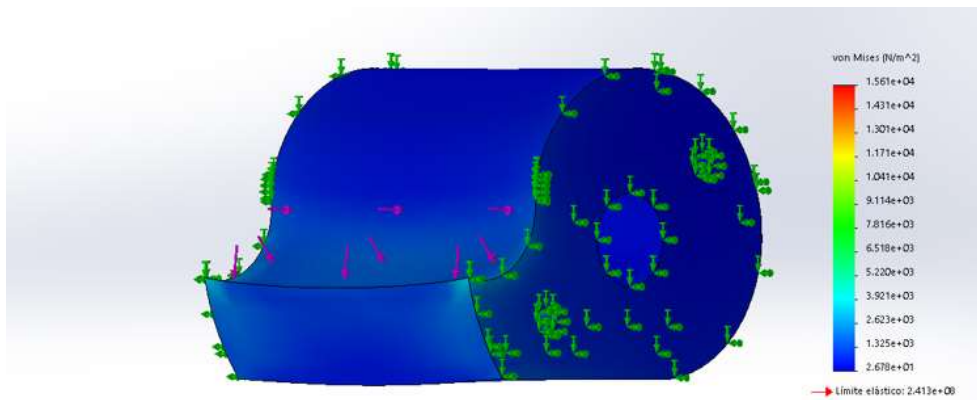


**Figura 53.** Desplazamiento de robot 1 sometida a tensiones.

En la figura 53 podemos observar la deformación y desplazamiento del arma luego de ser sometida a tensiones ya mencionadas y los puntos estáticos ya establecidos. La deformación está medida en mm (milímetros) y muestra con color rojo los puntos de mayor deformación siendo en valor más alto en el punto más intenso de color rojo de  $1.154 \times 10^{-6}$ , la misma que corresponde a  $0.001154$  micras.

Cabe resaltar que la figura muestra una deformación exagerada de la realidad de  $5.62696 \times 10^6$  para cuestiones interpretativas visuales.

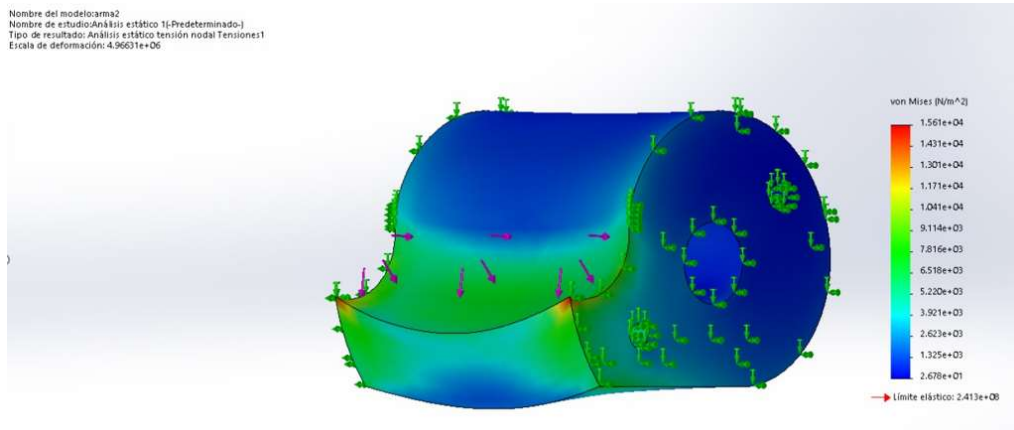
#### 4.1.2 Análisis de Arma de robot #2.



**Figura 54.** Modelado 3D de arma 2 visto desde solidworks.

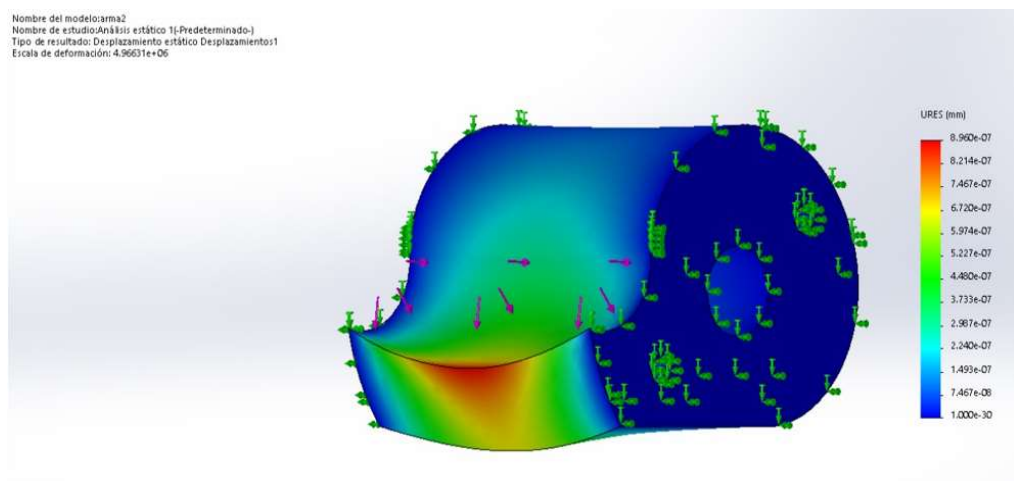
En la figura 54 podemos observar el arma del robot 2 visto desde solidworks, la misma que en base a su material de fabricación (acero 4330) tiene un límite elástico de  $2.413 \times 10^8$  Pascal.

Para el análisis se tomó dos puntos estáticos, los cuales están indicados con las flechas de color verde, y aplicaremos una serie tensiones ortogonales en el sentido que indican las flechas moradas, las cuales van en contra del movimiento del arma.



**Figura 55.** Arma de robot 2 deformada sometida a tensión.

En la figura 55 podemos visualizar los puntos susceptibles de deformación y el valor de la tensión aplicada medida en pascal. Se observa con el color rojo los puntos mas vulnerables, siendo los extremos los puntos mas susceptibles al ser sometidos a tensiones.

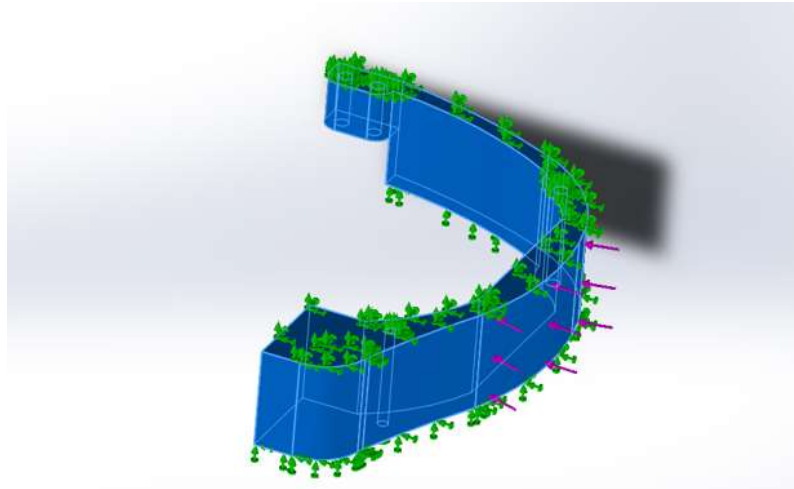


**Figura 56.** Deformación y desplazamiento del arma del robot 2 sometida a tensiones.

En la figura 56 podemos observar la deformación del arma luego de ser sometida a tensiones ya mencionadas y los puntos estáticos ya establecidos. La deformación esta medida en mm(milímetros) y muestra con color rojo los puntos de mayor deformación siendo el valor más alto en el punto más intenso de color rojo de 8.960e7, la misma que corresponde a 0.000896 micras.

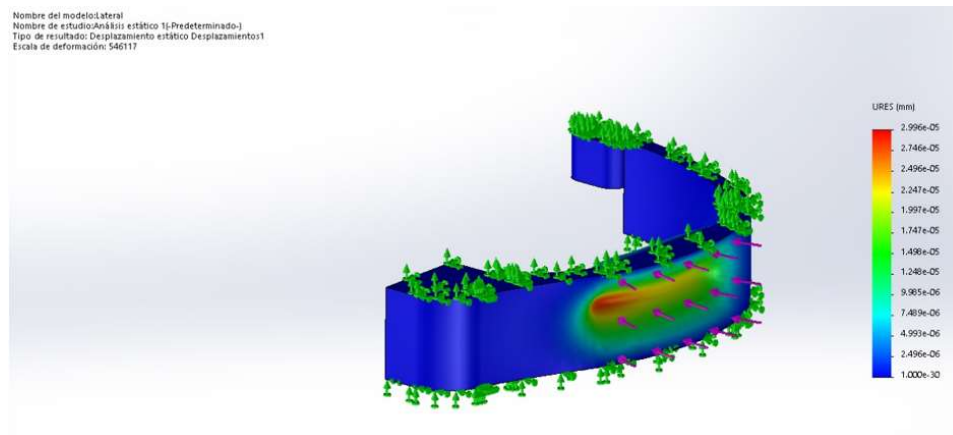
Cabe resaltar que la figura muestra una deformación exagerada de la realidad de 4.96631e+6 para cuestiones interpretativas visuales.

### 4.1.3 Análisis estructura lateral de derecha



**Figura 57.** Modelado 3D de estructura lateral derecha visto desde solidworks.

En la figura 57 podemos observar la estructura lateral derecha del robot visto desde solidworks, la misma que en base a su material de fabricación (polipropileno de alta densidad) tiene un límite elástico máximo de  $3.72e+07$  Pascal. Para nuestro análisis tomamos dos puntos estáticos, los cuales están indicados con las flechas de color verde, y aplicaremos una serie tensiones ortogonales en el sentido que indican las flechas moradas, las cuales simularán el efecto de un golpe en el sentido indicado.

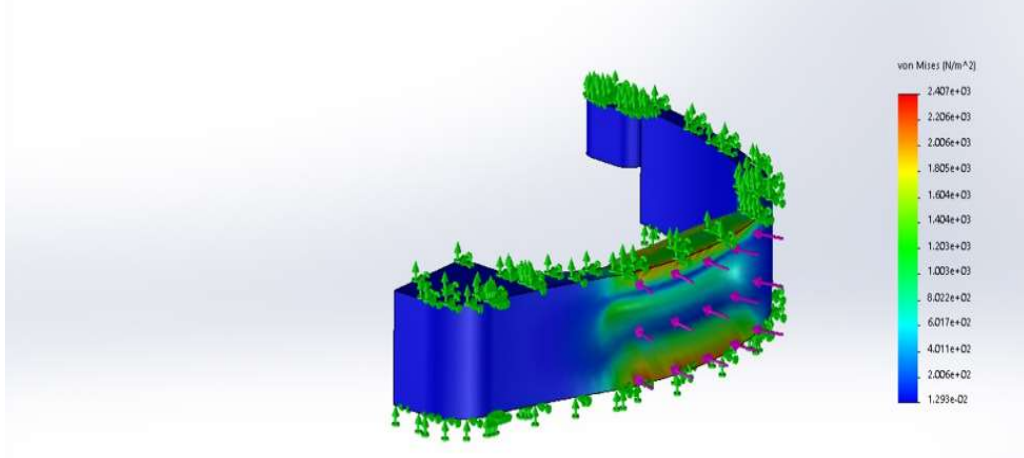


**Figura 58.** Desplazamiento de tapa lateral derecha sometida a tensiones.

En la figura 58 podemos observar la deformación de la estructura lateral derecha luego de ser sometida a tensiones vistas en la barra vertical ubicada a la derecha de la imagen. La deformación esta medida en mm (milímetros) y muestra con color rojo los puntos de mayor deformación siendo el valor más alto en el punto más intenso de color rojo de  $2.996e-5$  con respecto a la forma original de la figura, la misma que corresponde a 0.02996 micras.

Cabe resaltar que la figura muestra una deformación exagerada de la realidad de 546117 para cuestiones interpretativas visuales.

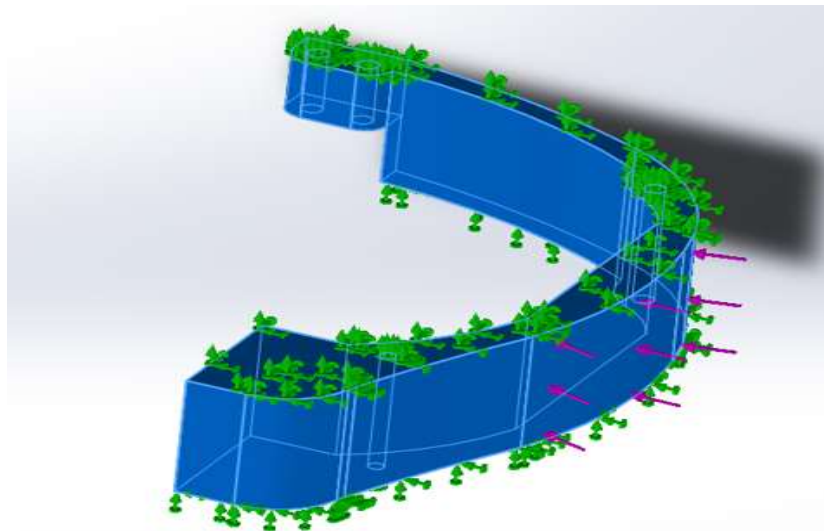
Nombre del modelo: lateral  
Nombre de estudio: Análisis estático 1j-Predefinido-1  
Tipo de resultado: Análisis: Análisis: tensión nodal Tensiones1  
Escala de deformación: 540117



**Figura 59.** Deformación de tapa lateral derecha sometida a tensiones.

En la figura 59 podemos visualizar los puntos susceptibles de deformación y el valor de la tensión aplicada medida en pascal. Se observa con el color rojo los puntos más vulnerables, siendo los extremos y las esquinas los puntos más susceptibles al ser sometidos a tensiones.

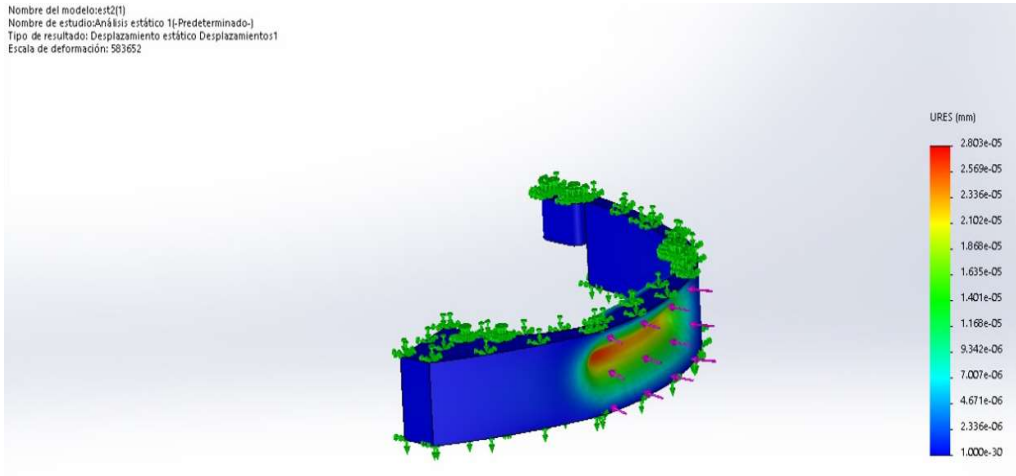
#### 4.1.4 Análisis estructura lateral izquierda



**Figura 60.** Modelado 3D de estructura lateral izquierda visto desde solidworks.

En la figura 60 podemos observar la estructura lateral izquierda del robot visto desde solidworks, la misma que en base a su material de fabricación (polipropileno de alta densidad) tiene un límite elástico máximo de  $3.72e+07$  Pascal. Para nuestro análisis tomamos dos puntos estáticos, los cuales están indicados con las flechas de color verde, y aplicaremos una serie tensiones ortogonales en el sentido que indican las flechas moradas, las cuales simularán el efecto de un golpe en el sentido indicado.

Nombre del modelo:st2[1]  
Nombre de estudio:Análisis estático 1i-Predefinido-1  
Tipo de resultado: Desplazamiento estático Desplazamientos1  
Escala de deformación: 583652

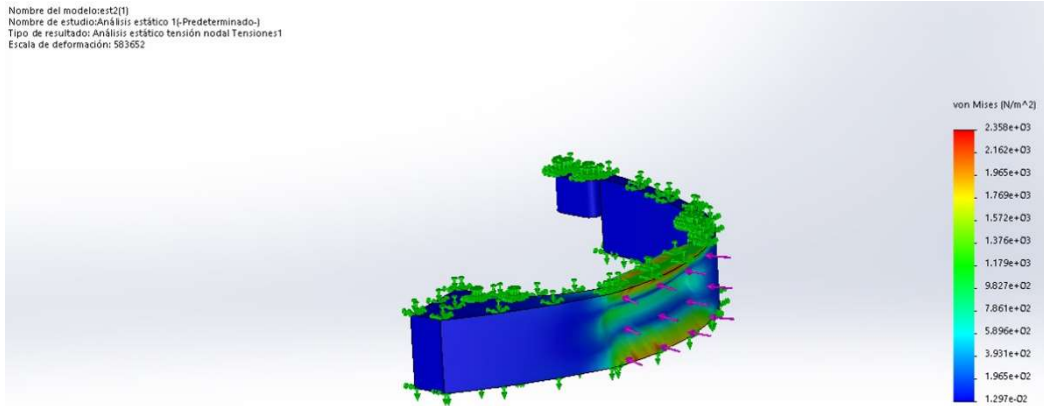


**Figura 61.** Desplazamiento de tapa lateral izquierda sometida a tensiones.

En la figura 61 podemos observar la deformación de la estructura lateral izquierda luego de ser sometida a tensiones vistas en la barra vertical ubicada a la derecha de la imagen. La deformación esta medida en mm (milímetros) y muestra con color rojo los puntos de mayor deformación siendo el valor más alto en el punto más intenso de color rojo de  $2.803e-5$  con respecto a la forma original de la figura, la misma que corresponde a 0.02803 micras.

Cabe resaltar que la figura muestra una deformación exagerada de la realidad de 583652 para cuestiones interpretativas visuales.

Nombre del modelo:st2[1]  
Nombre de estudio:Análisis estático 1i-Predefinido-1  
Tipo de resultado: Análisis estático tensión nodal Tensiones1  
Escala de deformación: 583652



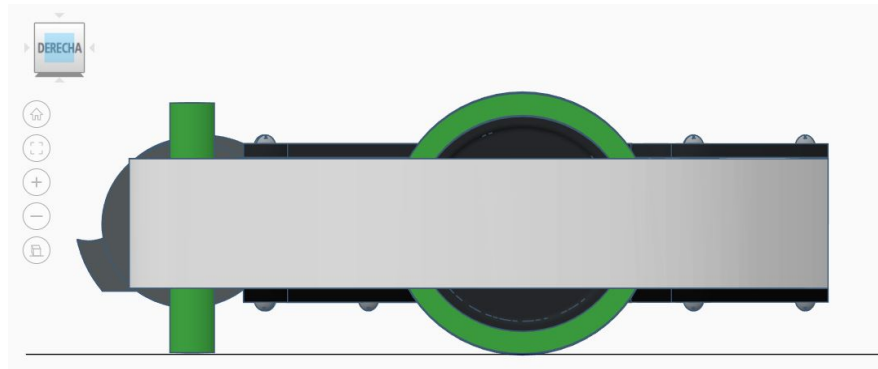
**Figura 62.** Deformación de tapa lateral izquierda sometida a tensiones

En la figura 62 podemos visualizar los puntos susceptibles de deformación y el valor de la tensión aplicada medida en pascal. Se observa con el color rojo los puntos mas vulnerables, siendo los extremos y las esquinas los puntos mas susceptibles al ser sometidos a tensiones.



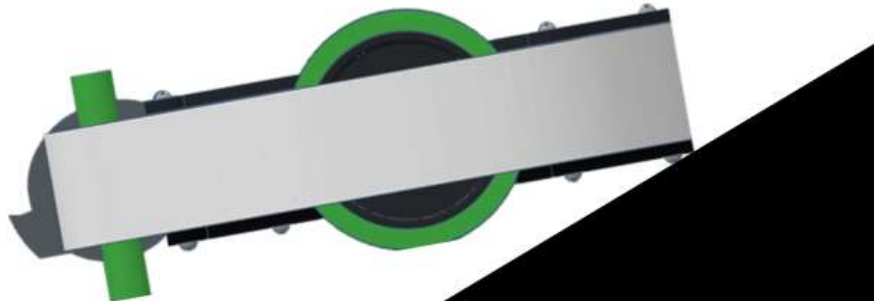
## 4.2 Análisis de diseño del robot 1 para considerar como estrategias de combate.

Se muestra continuación el modelado 3D del robot 1.



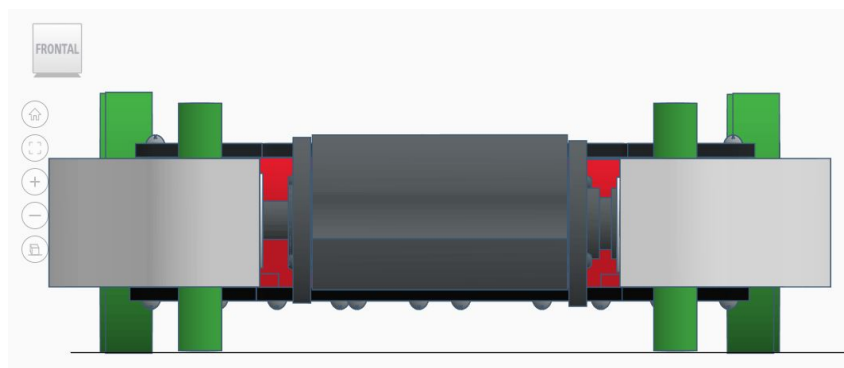
**Figura 63.** Vista lateral derecha robot 1.

En la figura 63 nos muestra que el robot puede funcionar de manera inversa sin problemas en el combate, pero se debe considerar que el arma funcionara de manera contraria, un golpe al oponente podría desplazar a su robot. Para voltearlo deberá girar sus ruedas de tracción hacia la derecha o izquierda y activar el arma al máximo el robot girará en su propio eje debido a la fuerza centrífuga.



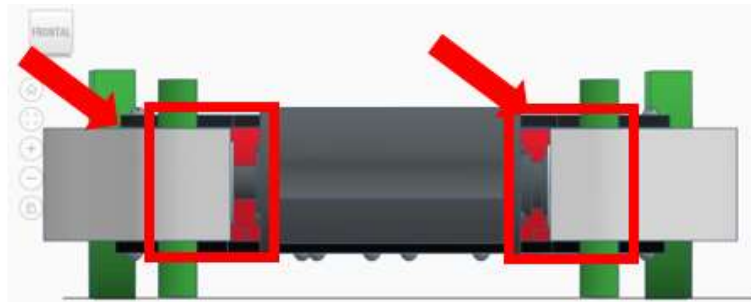
**Figura 64.** Vista lateral derecha robot 1 levantado por una rampa.

En la figura 64 se observa la altura desde el piso y la estructura es importante considerar que en combate no se debe permitir que los robots tipo rampa levanten al robot, puede provocar la pérdida de tracción, para empujarlos hacia las paredes generando inmovilización.



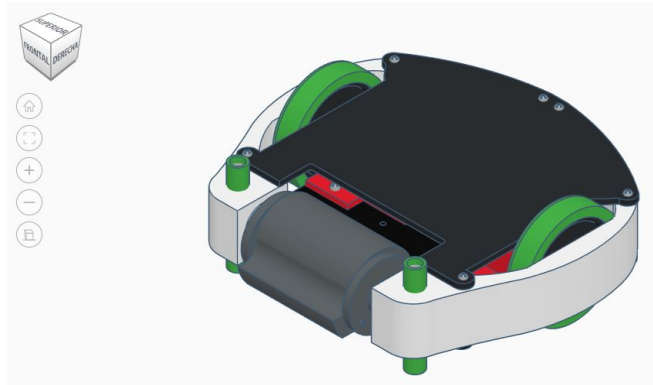
**Figura 65.** Vista frontal del robot 1.

En la figura 65 se muestra parte frontal del robot se observa que los patines sin circulares lo que le reduce la fricción con el piso para tener un mejor desplazamiento. El arma de robot 1 tiene un área de acción mayor en comparación del robot 2, en combate significa que podrá tener mayores posibilidades de brindarle un golpe a su oponente.

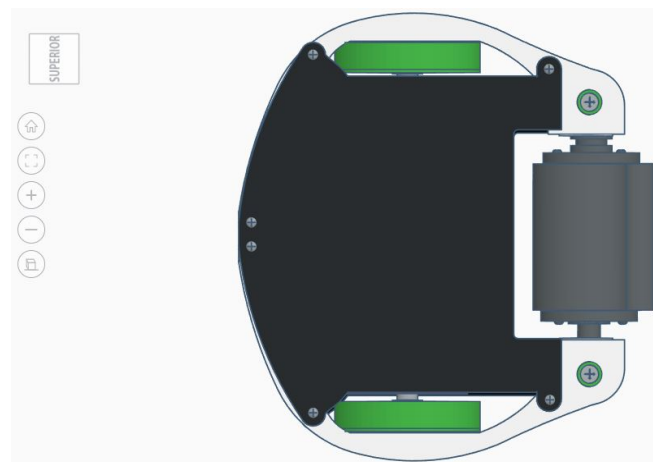


**Figura 66.** Vista frontal del robot 1 evitar recibir golpes directos en las áreas marcadas.

En la figura 66 nos muestra dos áreas en la que se debe evitar recibir la mayor cantidad de golpes debido a que son áreas donde se encuentran los rodamientos y el eje del arma.



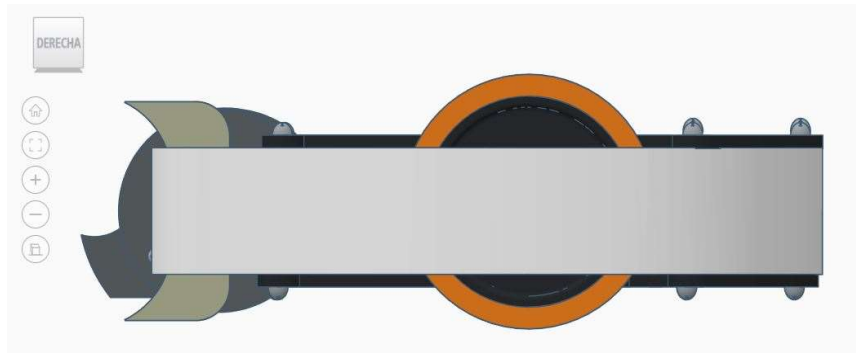
**Figura 67.** Vista de alzado en esquina derecha del robot1



**Figura 68.** Vista superior del robot1

### 4.3 Análisis de diseño del robot 2 para considerar como estrategias de combate.

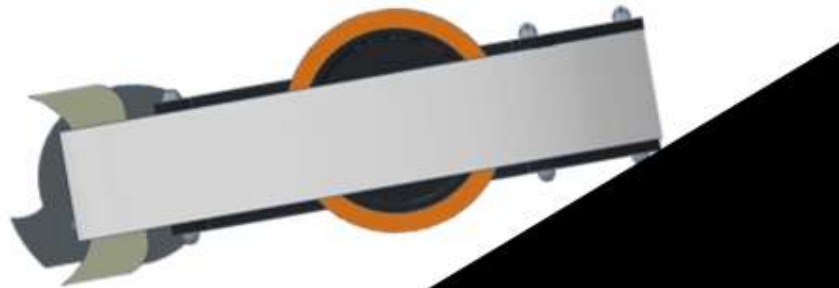
Se muestra continuación el modelado 3D del robot 2.



**Figura 69.** Vista lateral derecha robot 2.

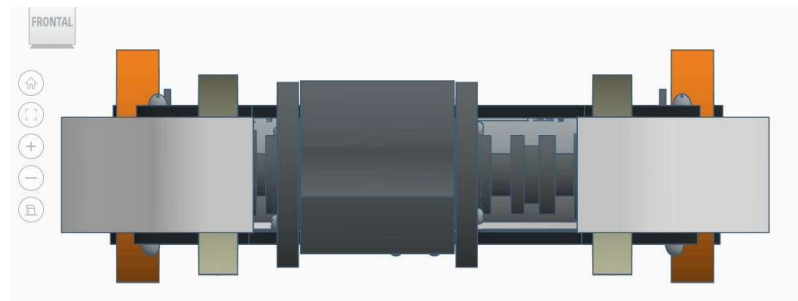
En la figura 69 se muestra que el robot puede funcionar de manera inversa sin problemas en el combate, pero se debe considerar que el arma funcionara de manera contraria, un golpe al oponente podría desplazar a su robot.

Para voltearlo deberá girar sus ruedas de tracción hacia la derecha o izquierda y activar el arma al máximo el robot girará en su propio eje debido a la fuerza centrífuga. Este robot cuenta con patines tipo gancho limitando un poco su movilidad por la fricción.



**Figura 70.** Vista lateral derecha robot 2.

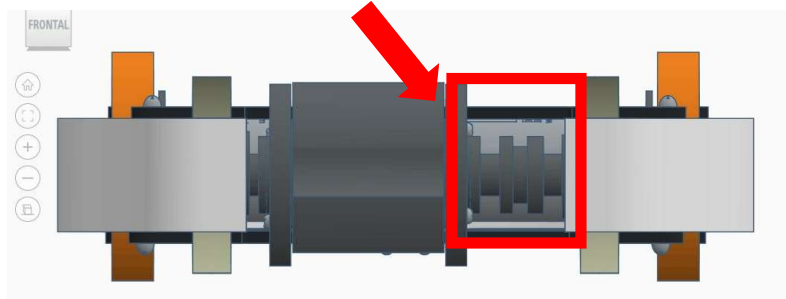
En la figura 70 se observa la altura desde el piso y la estructura, es importante considerar que en combate no se debe permitir que los robots tipo rampa levanten al robot, puede provocar la pérdida de tracción, en el caso del robot 2 cuenta con dos patines tipo gancho que reduce el desplazamiento por empuje del contrincante.



**Figura 71.** Vista frontal robot 2.

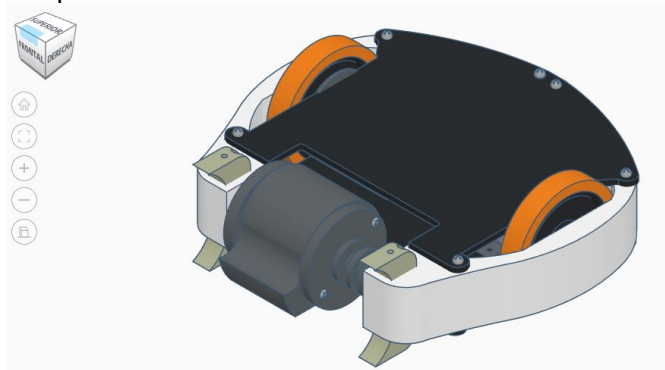
En la figura 71 se observa la vista frontal del robot 2 este robot cuenta con dos patines tipo gancho, es importante considerar el tipo de pista ya que podría reducir el desplazamiento de robot, pero también puede considerarse como un arma si el oponente trata de atacar los ganchos podría servir como defensa para destruir su arma.

El arma del robot 2 en comparación con el arma del robot 1 es más pequeña lo que limita el área de golpe, pero así mismo el impacto que se genera es mayor debido que la fuerza se concentra en un solo punto.

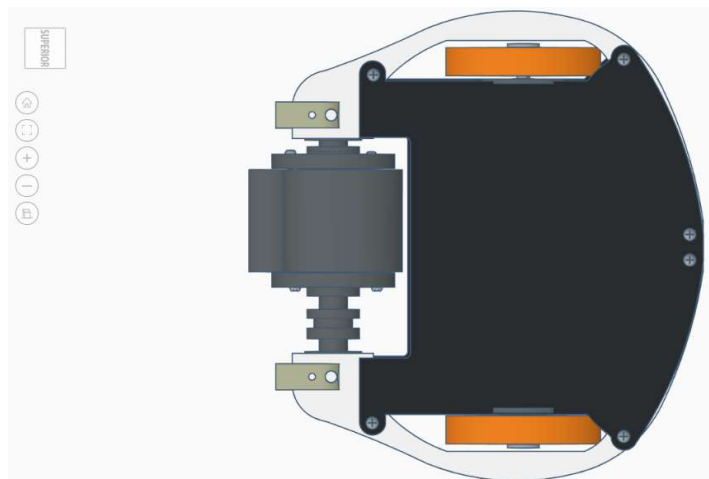


**Figura 72.** Vista frontal del robot 2 evitar recibir golpes directos en las áreas marcadas.

En la figura 72 nos muestra dos áreas en la que se debe evitar recibir la mayor cantidad de golpes de robots tipo Sawbots para evitar que corten la banda de transmisión en las poleas.



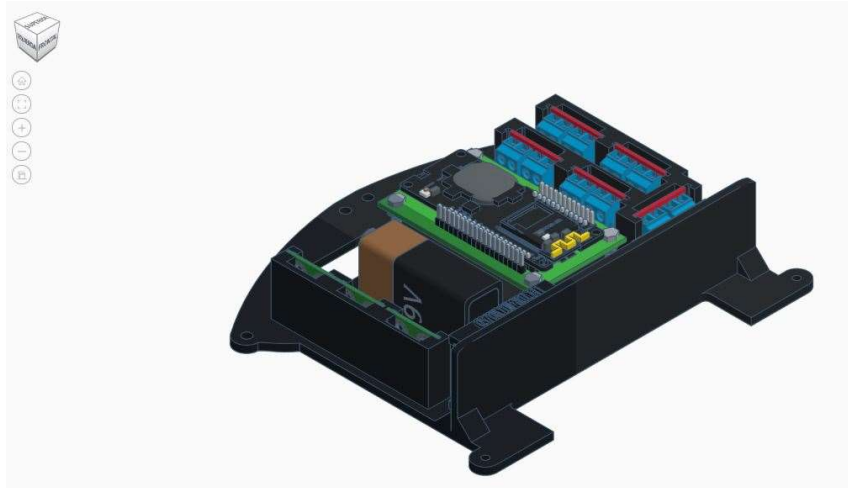
**Figura 73.** Vista de alzado en esquina derecha del robot2



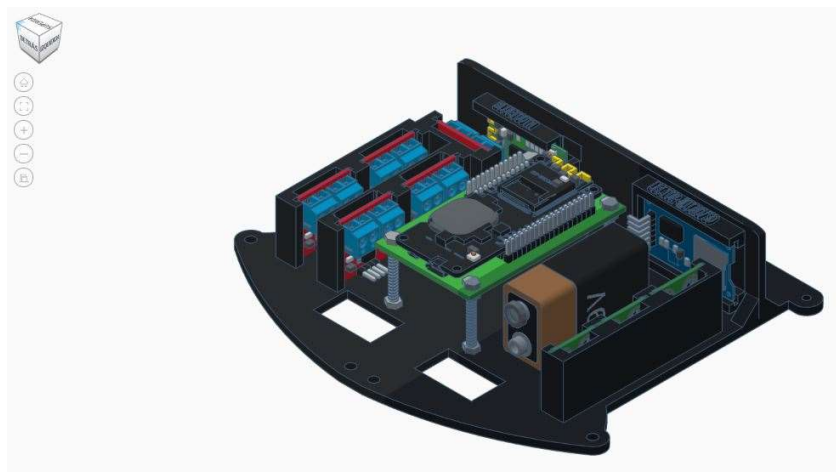
**Figura 74.** Vista su del robot2

#### 4.3.1 Diseño del Datalogger en robots.

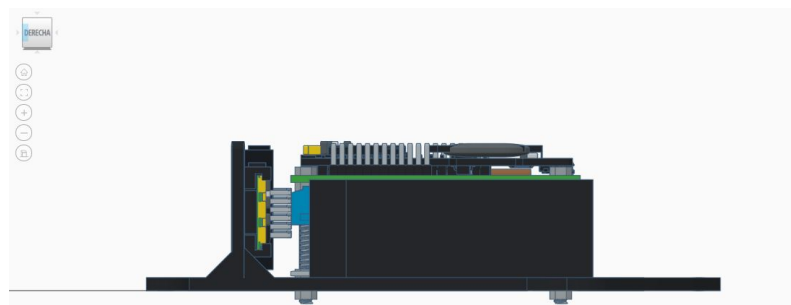
Diseño en 3D de forma y distribución de componentes que integran el datalogger para lograr el óptimo uso de espacios de acuerdo con la dimensión física de cada uno de los componentes.



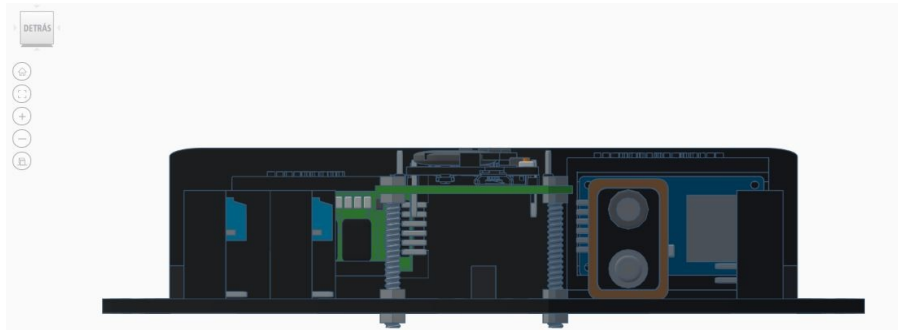
**Figura 75.** Perspectiva derecha del datalogger modelado en 3D



**Figura 76** Perspectiva detrás derecha del datalogger modelado en 3D

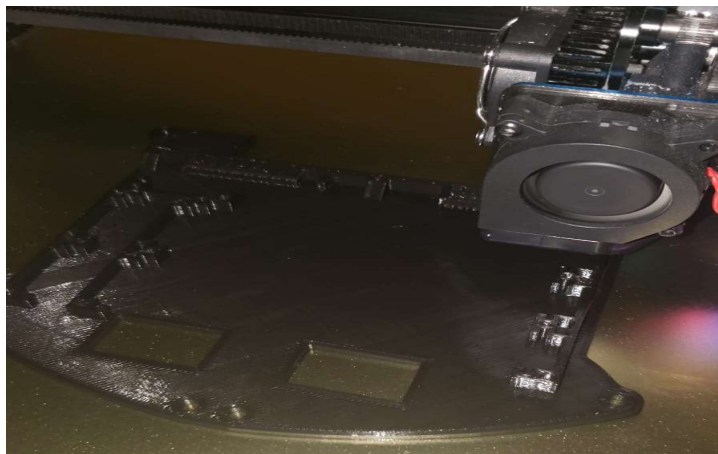


**Figura 77** Vista lateral derecha del datalogger modelado en 3D



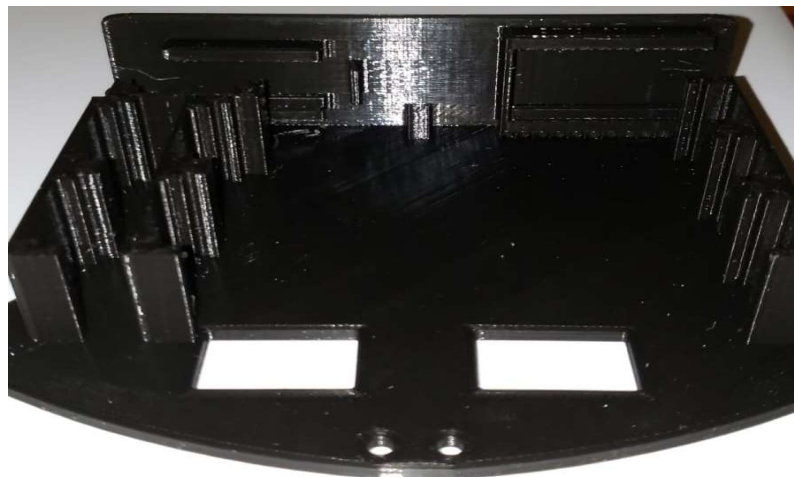
**Figura 78** Vista posterior del datalogger modelado en 3D

La elaboración de este soporte para los sensores fue desarrollada en una impresora 3D utilizando PLA.

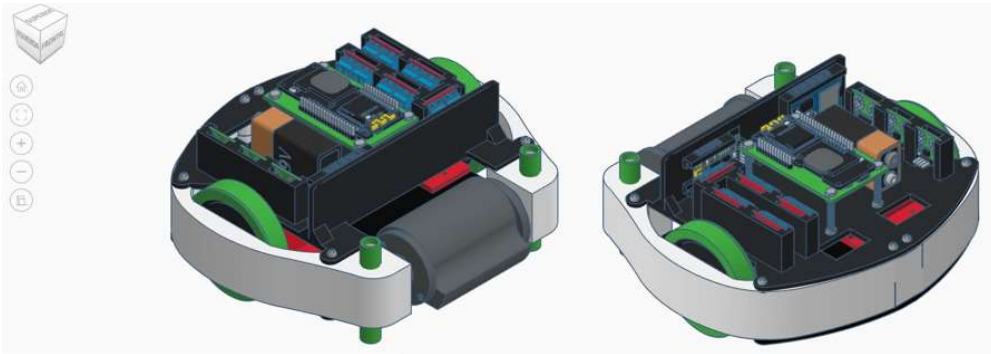


**Figura 79** Impresión en 3D de los soportes de la base soporte del datalogger.

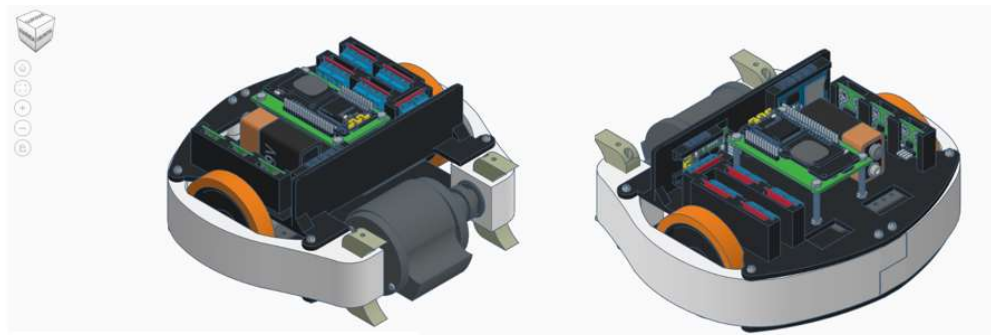
En la figura 78 podemos observar la fabricación de la base donde se soportará cada uno de los componentes del datalogger luego de haber modelado en 3D en base a las dimensiones reales de cada uno de los componentes.



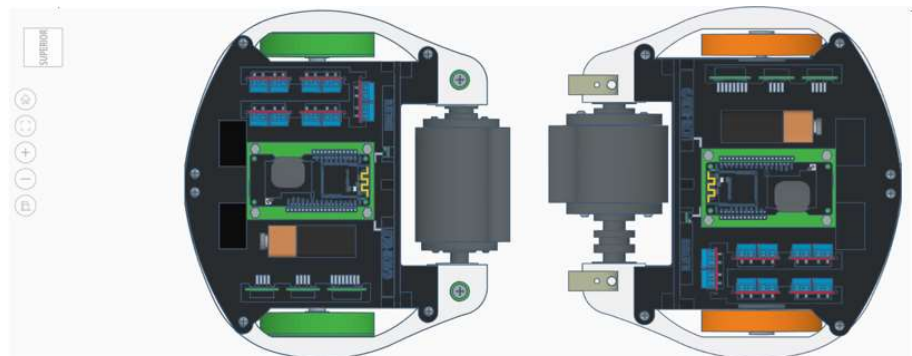
**Figura 80** Base soporte impresa del datalogger.



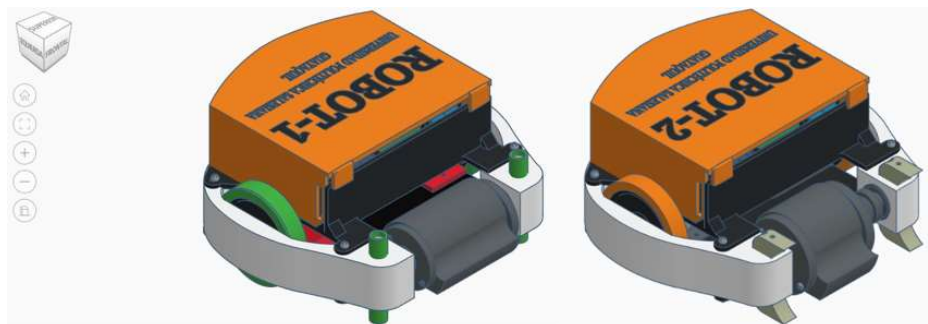
**Figura 81** Perspectiva Isométrica frontal y posterior del robot 1 con el datalogger



**Figura 82** Perspectiva Isométrica frontal y posterior del robot 2 con el datalogger

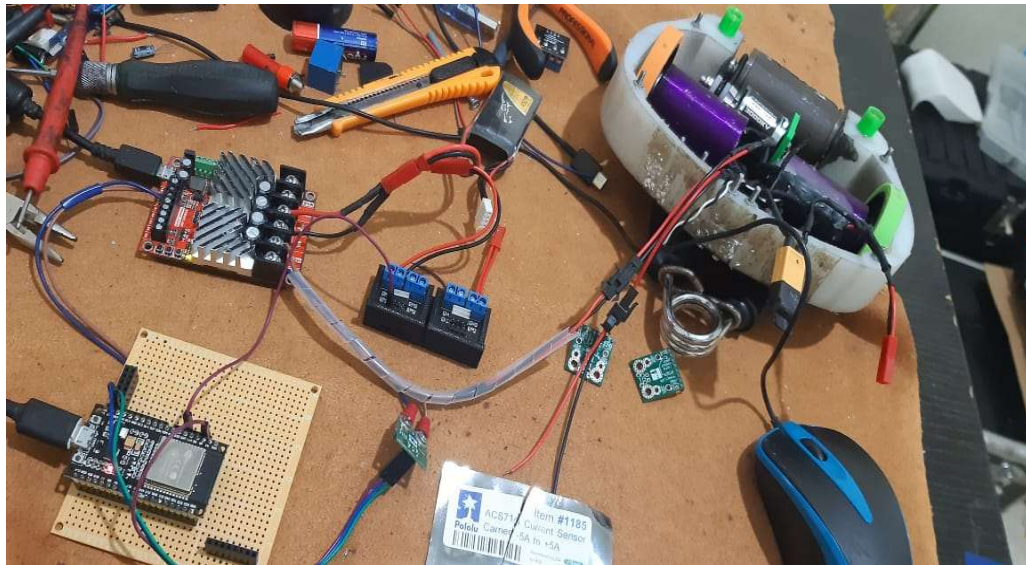


**Figura 83** Perspectiva Isométrica superior del robot 1 y robot 2 con el datalogger



**Figura 84** Perspectiva Isométrica superior del robot 1 y robot 2 con protección para el datalogger.

En las figuras 81, 82, 83 se observa al datalogger instalado en los robots es un sistema plug and play.  
En la figura 84 muestra la implementación de tapas para proteger los sensores y controlador.



**Figura 85** Prueba de funcionalidad de componentes

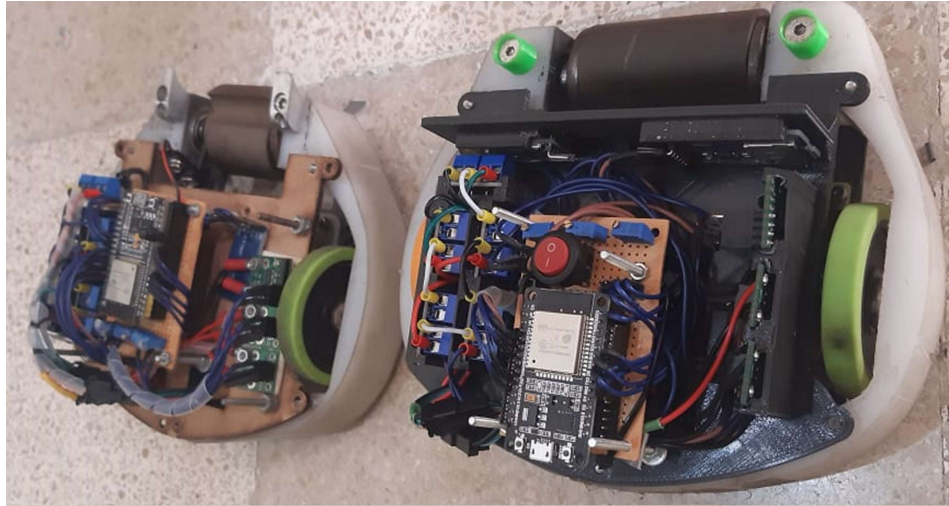
En la figura 85 podemos observar las pruebas realizadas con los sensores antes de su implementación en la base del datalogger. Se realizaron pruebas de los motores en vacío y con carga para verificar los valores de los sensores de voltaje y corrientes. Se utilizó un multímetro Fluke 117 y una RoboClaw 2x30A que cuenta con un software propio, Basic Micro para controlar el voltaje y corriente en las salidas de los motores.



**Figura 86** Integración de componentes del datalogger.

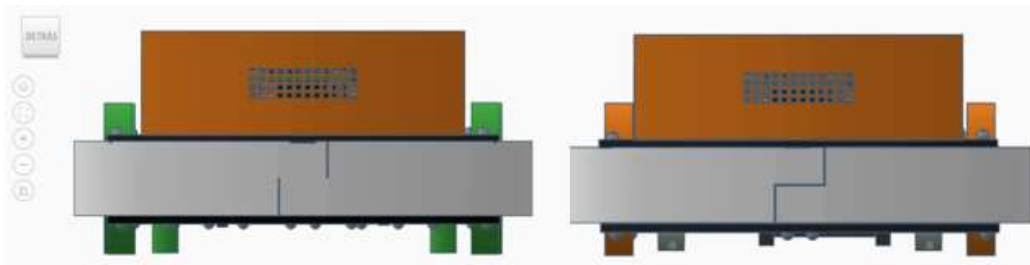
En la figura 86 observamos la integración total de los componentes del datalogger, los mismos que se encuentran totalmente instalados y atornillados a la base.





**Figura 87** Datalogger acoplados en ambos robots.

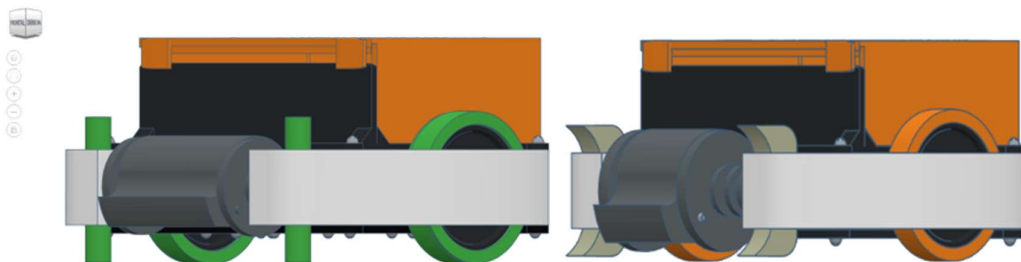
Con los datalogger totalmente implementados en ambos robots se procedió a modelar la tapa superior en 3D y se imprimieron. En la tapa superior se colocó la numeración que indica la correspondencia de cada uno de ellos.



**Figura 88** Vista posterior del robot 1 y robot 2



**Figura 89** Vista lateral del robot 1 y robot 2



**Figura 90** Vista isométrica del robot 1 y robot 2



**Figura 91** Robot 1 y 2 terminados

#### **4.2.2 Funcionalidad del datalogger**

Para comprobar la funcionalidad del datalogger acoplado a los robots, realizaremos una prueba en vacío, embancando las ruedas del robot, y otra prueba en piso simulando la funcionalidad del robot en un combate, donde obtendremos la variable de potencia de los motores de tracción y el motor del arma en tiempo real a partir de la lectura de voltaje y corriente.

#### **4.4 Lógica operacional del Datalogger**

A continuación se describirá la lógica básica de funcionamiento del Data Logger implementado. Se consideraron dos modos de uso del data Logger para darle mayor versatilidad.

##### **4.4.1 Escritura de datos en la tarjeta Micro SD.**

Cuando la interruptor que provee de energía desde la batería de 9 voltios es accionado se energiza el módulo de control ESP32 así como los sensores de voltaje y corriente. Inmediatamente el ESP32 guarda todos los cambios medidos en los sensores de corriente y voltajes en la tarjeta de memoria instalada en el módulo SD. Cuando la pruebas y obtención de datos termina, se debe poner en Off el interruptor, se desmonta la tarjeta de memoria y se la instala en una PC. Los datos obtenidos podran ser visualizados en una carpeta con extensión de archivos “txt” individualmente. Para una mayor interpretación de los datos registrados, bajo el entorno de Matlab se realizó un programa que al ejecutarlo solicita cargar los datos guardados en la microSD, y los muestra de manera gráfica. En donde se puede visualizar los voltajes, corrientes y potencias de cada motor.

#### 4.4.2 Visualización en matlab de las gráficas de potencias en tiempo real.

Para iniciar con la transferencia de datos del robot debimos activar el bluetooth de la computadora.

Luego de desde la PC y bajo el entorno de Matlab se ejecutó el programa que muestra una ventana para seleccionar el robot al cual queremos analizar cómo podemos observar en la figura 92.



**Figura 92** Selección de robot en Matlab.

Para obtener más detalles sobre el código utilizado en Matlab, revise el anexo 12.

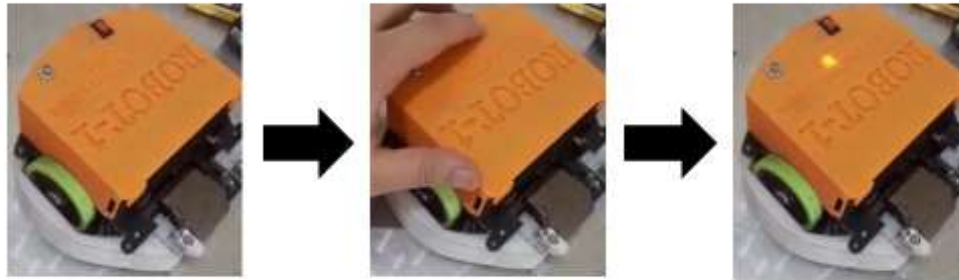
Luego que seleccionamos la cámara disponible en la computadora para visualizar al robot durante las pruebas, como podemos observar en la figura 93 se muestra la cámara.



**Figura 93** Selección de cámara en Matlab.

Paralelo al proceso de ejecución del programa de matlab, accionamos el interruptor de encendido del data logger montado sobre el robot. Una vez que el interruptor que provee de energía desde la batería de 9 voltios, el módulo ESP32, así como los sensores de voltaje y corriente adicionalmente se enciende un led en la parte superior de la tapa. Inmediatamente el módulo ESP32 receipta y procesa los datos de los sensores de voltajes y corrientes, para obtener la así la potencia consumida por cada uno de los motores.

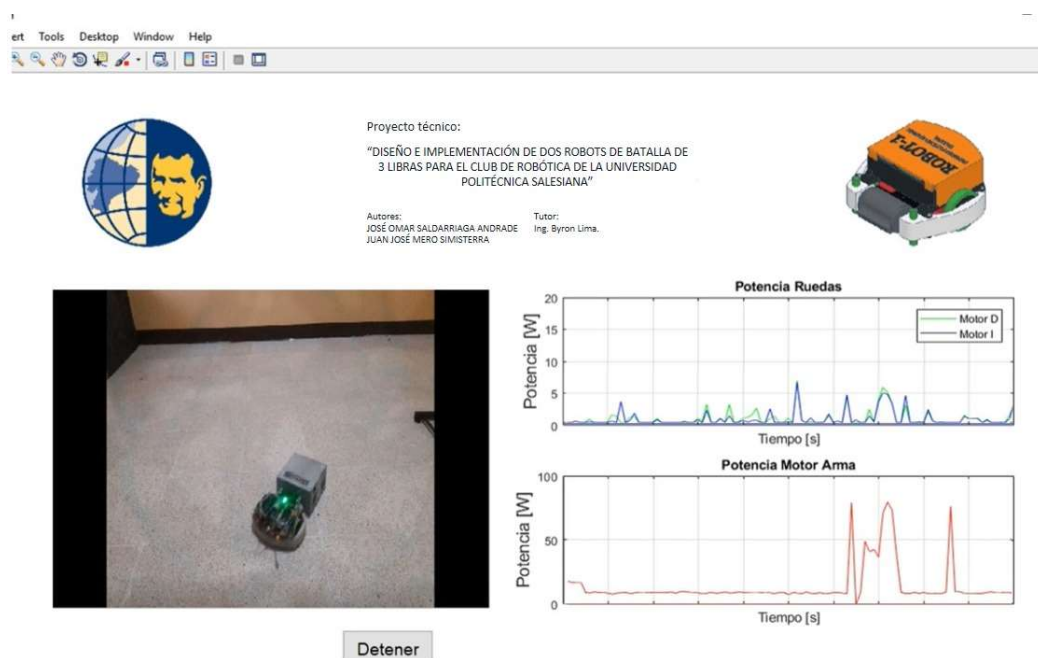
Para obtener más detalles sobre el código utilizado en Matlab, revise el anexo 13.



**Figura 94** Interruptor de encendido del datalogger-robot1.

Estos datos son enviados al computador vía bluetooth, a través del puerto serie en el computador se visualizará en tiempo real las potencias y sus distintos cambios en función del tiempo de cada uno de los motores de los robots.

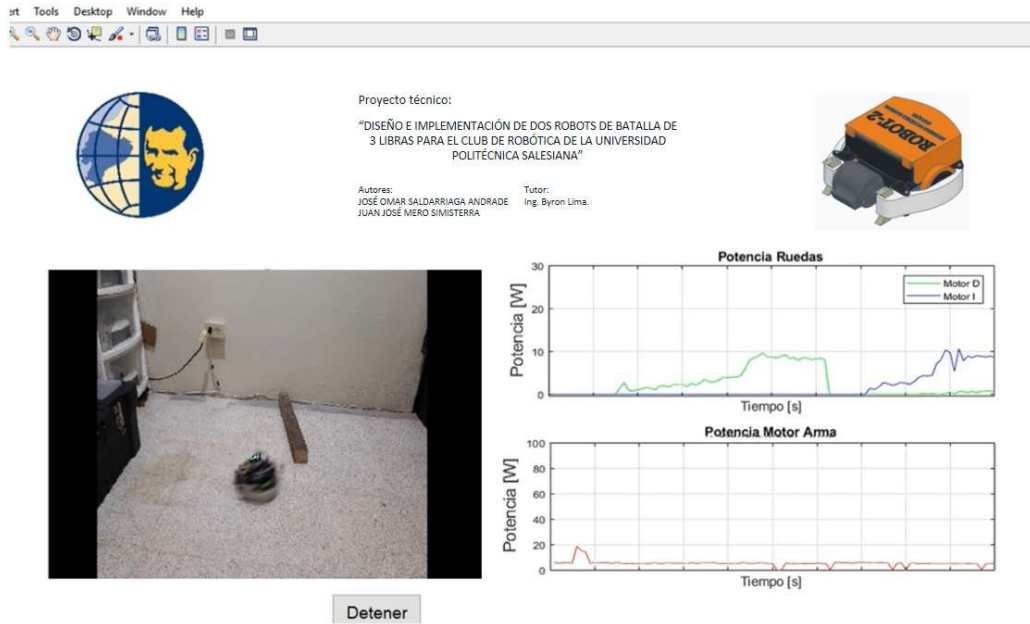
En la figura 94 podemos observar el gráfico donde se visualiza la potencia consumida por cada uno de los motores del robot seleccionado.



**Figura 95** Pruebas del robot 1 valores de potencias de motores en tiempo real.

En la figura 95 podemos observar los valores en tiempo real de las potencias de los motores de tracción y arma además esta interfaz cuenta con el acceso a cámara de fábrica de la computadora o alguna conectada a sus puertos USB, para observar los movimientos del robot en tiempo real.

En la parte superior de la interfaz se puede observar la gráfica de variación de la potencia de los motores con respecto al tiempo y en la parte inferior la gráfica de la potencia del motor del arma que, al activarse rompió la inercia y generó el pico de corriente que aumento, así su potencia.



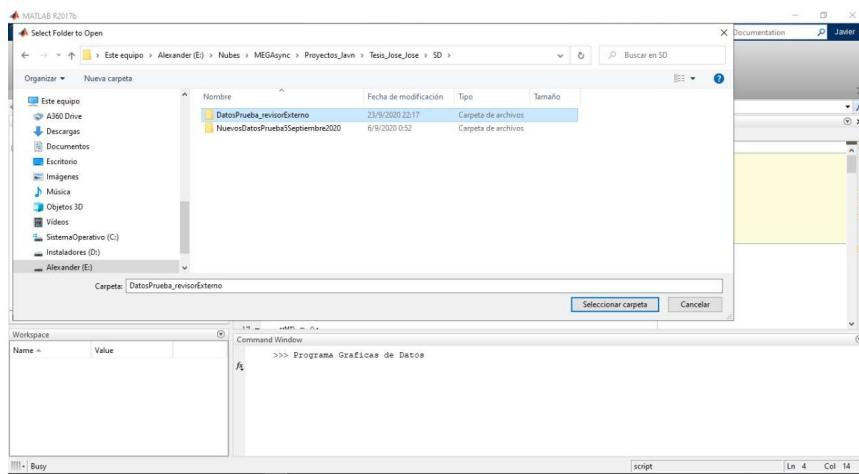
**Figura 96** Pruebas del robot 2 valores de potencias de motores en tiempo real.

En la gráfica 96 se muestran los valores de las potencias del robot 2 en esta prueba al robot se lo sometió a giros hacia la izquierda y la derecha activando un solo motor para cada sentido, el motor del arma se activó adicionalmente con un 10% de su capacidad.

#### 4.4.3 Resultados de pruebas obtenidos del data logger por tarjeta SD.

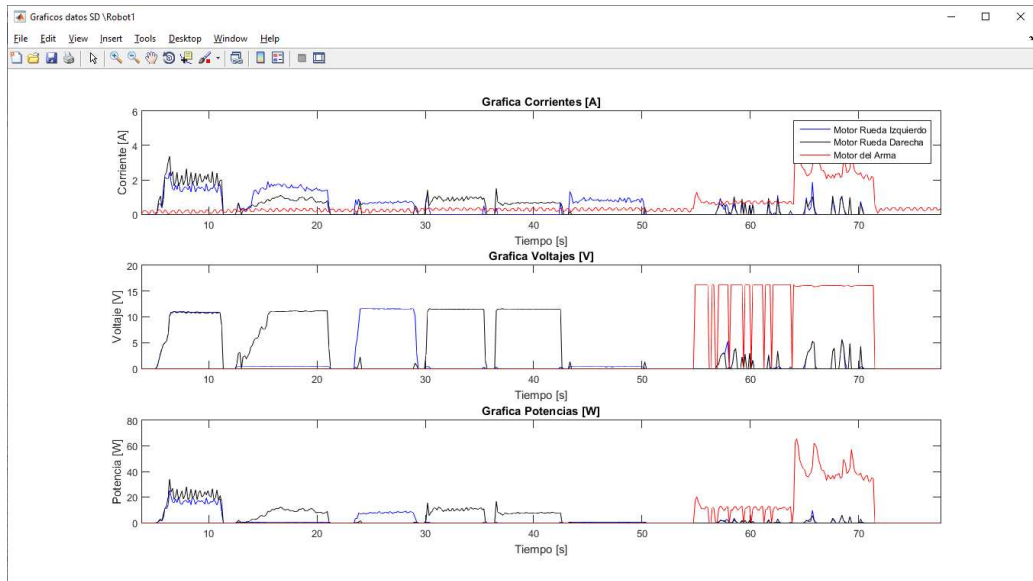
Desde la computadora y bajo el entorno de Matlab se ejecutó el programa que solicita abrir la carpeta que contiene los archivos .txt que se encuentran guardados en la tarjeta microSD.

Luego de realizar la carga de los archivos y darle clic en “SELECCIONAR CARPETA” el programa muestra la gráfica de los datos obtenidos de los valores de los voltajes, corriente, y potencia de cada motor para realizar un análisis más detallado del comportamiento de los robots.

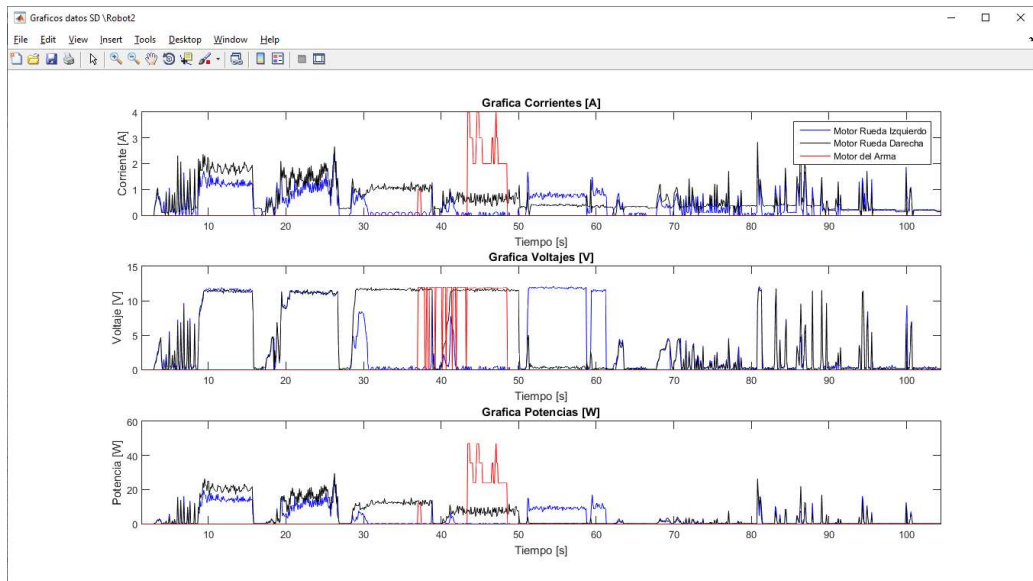


**Figura 97** Apertura de carpeta de datos de microSD.

En la figura 97 se muestra la ventana donde seleccionamos la carpeta que contiene los datos que fueron guardados en la tarjeta microSD desde el data logger.



**Figura 98** Robot 1 gráfica de datos obtenidos de tarjeta microSD



**Figura 99.** Robot 2 gráfica de datos obtenidos de tarjeta microSD

## **Conclusiones**

El cumplimiento del objetivo de Implementar un data logger que almacene las variables de potencia de los motores de desplazamiento y arma, se convirtió en el reto más difícil de nuestro proyecto, debido a que la selección de sensores de voltaje para la medición de tensión de los motores de desplazamiento no fue la adecuada inicialmente, cuya complejidad radicaba en que los motores al cambiar de giro también cambiaban la referencia a tierra de la conexión del sensor provocando un corto circuito.

Las cargas de corriente de los motores de tracción usados en nuestro proyecto demuestran su eficiencia durante la operación por su bajo consumo de corriente cuyos valores en estado de movimiento continuo con la carga de 3 libras a la cual pertenece la categoría de combate se mantienen en el orden de décimas de amperios, operando en promedio a un 40% de su capacidad. Y cuyos picos máximos registrados al romper la inercia con valores altos de voltaje llegan hasta 3 amperios, siendo este valor el 107% corriente de acuerdo a la placa del motor.

La carga de corriente consumida por el motor del arma aunque se encuentra dentro de sus parámetros operacionales es relativamente alta como para considerar contar siempre con una batería adicional cuando el robot se encuentra concursando. Los picos de corriente máximos registrados durante las pruebas es de 48 amperios, correspondiendo a un 96% del valor máximo de corriente. Mientras que luego de romper la inercia y girando sin ninguna carga adicional en el arma, la corriente registra valores entre 3 y 5 amperios, lo que corresponde a 10% la corriente máxima de acuerdo a la placa del motor.

El uso del entorno del software Matlab proporciona una herramienta muy útil a la hora de desarrollar funcionalidad, ligada a lectura de datos vía inalámbrica.

El análisis de estos datos podría ayudar a los operadores del Club de Robótica a generar estrategias de combate para evitar la descargar anticipada de corriente de las baterías.

## **Recomendaciones**

Para optimizar el uso de batería del arma cuando los robots se encuentran en combate se recomienda ejecutar el uso del arma solo cuando el objetivo se encuentre cerca, debido a que por su uso innecesario y al alto consumo de corriente la carga de la batería disminuya drásticamente.

Se recomienda activar el arma de una manera exponencial para que el robot rompa la inercia y no genere un pico alto de corriente, una vez activada el arma después de generar el golpe considere un tiempo de no activación para enfriar el motor esto puede durar algunos segundos.

Dada la condición de fácil desinstalación (plug and play) del data logger montado sobre cada robot, es necesario conectar los robot de acuerdo a las etiquetas propuestas, la desconexión de los sensores de corriente podría dejar sin el flujo de energía a los motores provocando que en combate no funcionen.

Se recomienda en cada round de batalla de robots utilizar protectores como gafas de protección certificadas para trabajo, guantes anti corte para evitar lesiones ocasionadas por los robots.

Se recomienda utilizar fija tornillos en las tuercas y tornillos de las estructuras para evitar que las vibraciones aflojen los tornillos y provoque fallas estructurales perjudicando la batalla.

Después de cada batalla se recomienda realizar un chequeo general de cables, interruptores, receptor, variador de frecuencia ESC, soporte de baterías, motores debido a que los robots son sometidos a golpes, vibraciones y altas corrientes lo que pueden provocar fallos en la electrónica de los robots



## Bibliografía

- [1] M. A. Meggiolaro, «RioBotz Combot Tutorial,» de *RioBotz Combot Tutorial*, United States, Creative Commons Attribution, 2009, p. 13.
- [2] M. Berry, «LA HISTORIA DEL COMBATE DE ROBOTS: BATTLEBOTS,» Febrero 2012. [En línea]. Available: [www.servomagazine.com](http://www.servomagazine.com) . [Último acceso: 29 junio 2019].
- [3] Battlebots, «Battlebots,» Trademark of BattleBots, Inc, [En línea]. Available: <https://battlebots.com/>. [Último acceso: 12 junio 2020].
- [4] BattleBots, «WanHoo (2019),» Trademark of BattleBots, 2019. [En línea]. Available: <https://battlebots.com/robot/wanhoo-2019/>. [Último acceso: 12 Abril 2020].
- [5] Universidad de Cuenca, «Categorías CER 2019,» 2019. [En línea]. Available: <https://cer.cedia.edu.ec/es/categorias-cer-2019>. [Último acceso: 12 Diciembre 2019].
- [6] G. Imahara, «KICKIN 'BOT: UNA GUÍA ILUSTRADA PARA CONSTRUIR ROBOTS DE COMBATE,» [En línea]. Available: <https://www.extremetech.com/gaming/55636-kickin-bot-an-illustrated-guide-to-building-combat-robots/15?print>. [Último acceso: 2 Julio 2020].
- [7] P. Miles, *Build Your Own Combat Robot*, New York: The McGraw-Hill Companies, 2002.
- [8] M. A. Meggiolaro, *RioBotz Combot Tutorial*, Brasil, 2009.
- [9] Final Test, «Dataloggers,» [En línea]. Available: <https://www.finaltest.com.mx/product-p/art-4.htm>. [Último acceso: 12 agosto 2020].
- [10] Mobus, «¿Qué es una batería LiPo?,» [En línea]. Available: <https://mobus.es/blog/que-es-una-bateria-lipo/>. [Último acceso: 12 Agosto 2020].
- [11] Promotec, «Bateria de Lipo,» [En línea]. Available: <https://www.promotec.net/elegir-bateria-lipo/>. [Último acceso: 13 Julio 2020].
- [12] J. P. P. y. A. Gardey, «Definición de Rodamiento,» 2016. [En línea]. Available: <https://definicion.de/rodamiento/#:~:text=Rodamiento%20es%20la%20denom,inaci%C3%B3n%20de,sobre%20el%20cual%20%C3%A9ste%20gira..> [Último acceso: 24 Mayo 2020].
- [13] PM Plastimecol, «Ficha Técnica Acetal,» [En línea]. Available: <http://acerplas.com/acetel.html>. [Último acceso: 14 Junio 2020].
- [14] HobbyKing, «Turnigy Evolution PRO Digital,» [En línea]. Available: [https://hobbyking.com/es\\_es/evolution-pro-8ch-digital-proportional-radio-control-system-m1-white.html?queryID=d6c78882dfe077654485eb4d94a123f3&objectID=77483&indexName=hbk\\_live\\_magento\\_es\\_es\\_products](https://hobbyking.com/es_es/evolution-pro-8ch-digital-proportional-radio-control-system-m1-white.html?queryID=d6c78882dfe077654485eb4d94a123f3&objectID=77483&indexName=hbk_live_magento_es_es_products). [Último acceso: 14 Julio 2020].
- [15] Flysky, «Product Model: FS-IA6B,» [En línea]. Available: <https://www.flysky-cn.com/ia6b-canshu>. [Último acceso: 18 Agosto 2020].
- [16] RC-Innovations, «Tmotor F 35A 32bit 3 ~ 6S,» Unmanned Technology SL., [En línea]. Available: <https://rc-innovations.es/tmotor-f-35a-32bit-3s-6s-variador-FPV-ESC-Dshot>. [Último acceso: 12 junio 2020].
- [17] Compañía Levantina de Reductores S.L, «Diferencias entre motores con escobillas y brushless,» [En línea]. Available: <https://clr.es/blog/es/diferencias-motores-con-escobillas-brushless/>. [Último acceso: 8 Enero 2020].

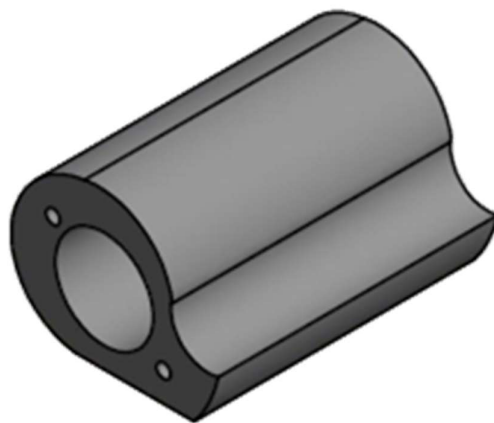
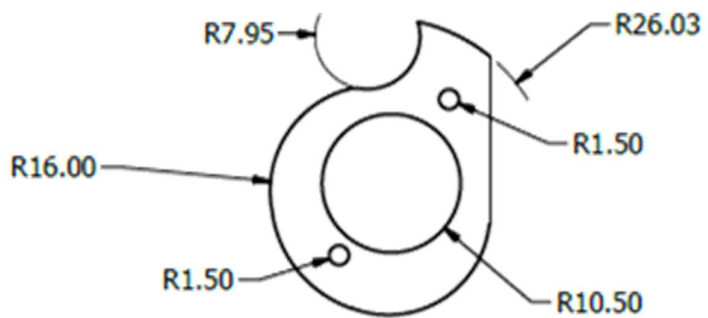
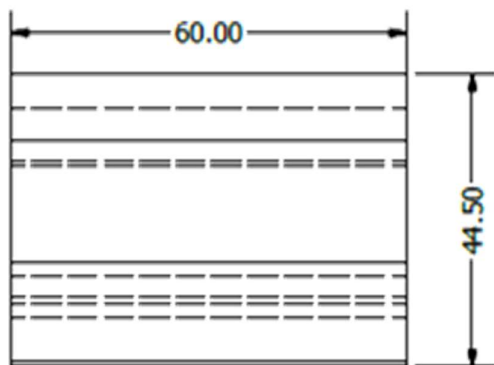
- [18] HobbyKing, «NTM Prop Drive 28-36 3000KV / 755W,» HobbyKing, [En línea]. Available: [https://hobbyking.com/es\\_es/ntm-prop-drive-28-36-3000kv-755w.html](https://hobbyking.com/es_es/ntm-prop-drive-28-36-3000kv-755w.html). [Último acceso: 12 abril 2020].
- [19] GetFPV LLC, «T-Motor F80 PRO - 2500KV Motor,» GetFPV LLC, [En línea]. Available: <https://www.getfpv.com/t-motor-f80-pro-2500kv-motor.html>. [Último acceso: 23 Julio 2020].
- [20] Amidata S.A.U., «Controladores para Motores DC,» [En línea]. Available: <https://es.rs-online.com/web/c/automatizacion-y-control-de-procesos/motores-electricos-controladores-de-motor-y-perifericos/controladores-para-motores-dc/>. [Último acceso: Enero 2020].
- [21] Dimension Engineering LLC, «Controlador de motor Sabertooth dual 12A para R / C,» [En línea]. Available: <https://www.dimensionengineering.com/products/sabertooth2x12rc>. [Último acceso: 19 Enero 2020].
- [22] Robot Power, «Scorpion Mini Product Information,» [En línea]. Available: [http://www.robotpower.com/products/scorpion\\_mini\\_info.html](http://www.robotpower.com/products/scorpion_mini_info.html). [Último acceso: 17 Julio 2020].
- [23] Pololu Corporation, «20.4: 1 Motorreductor Metálico 25Dx50L mm HP 12V,» Pololu Corporation, [En línea]. Available: <https://www.pololu.com/product/3203/specs>. [Último acceso: 19 Junio 2020].
- [24] Espressif Systems, ESP32 Technical Reference Manual, Shanghai: Espressif Systems, 2020.
- [25] Conrad Electronic International, «Joy-it Development board Node MCU ESP32 Module,» [En línea]. Available: <https://www.conrad.com/p/joy-it-development-board-node-mcu-esp32-modul-1656367>. [Último acceso: 10 Agosto 2020].
- [26] RefractOn, «Modulo Sensor de Voltaje y Corriente Arduino,» [En línea]. Available: [https://www.refractron.com/index.php?route=product/product&product\\_id=257](https://www.refractron.com/index.php?route=product/product&product_id=257). [Último acceso: 19 julio 2019].
- [27] Pololu Corporation, «ACS714 Current Sensor Carrier -5A to +5A,» [En línea]. Available: <https://www.pololu.com/product/1185/specs>. [Último acceso: 11 Marzo 2020].
- [28] Pololu Corporation, «ACS709 Current Sensor Carrier -75A to +75A,» [En línea]. Available: <https://www.pololu.com/product/2199>. [Último acceso: 16 Julio 2020].
- [29] Pololu Corporation, «ACS709 Current Sensor Carrier -75A to +75A,» [En línea]. Available: <https://www.pololu.com/product/2199/specs>. [Último acceso: 19 Julio 2020].
- [30] D. Hertz, «HC-05 Hoja de datos | Módulo transceptor Bluetooth,» EETech Media, 13 mayo 2019. [En línea]. Available: <https://maker.pro/custom/tutorial/hc-05-bluetooth-transceiver-module-datasheet-highlights>. [Último acceso: 12 Agosto 2020].
- [31] Patagoniatec, «Cómo Leer Una Tarjeta SD Con Arduino,» [En línea]. Available: <https://saber.patagoniatec.com/2014/10/como-leer-una-tarjeta-sd-micro-usd/>. [Último acceso: 10 abril 2019].
- [32] Energizer, «Energizer MAX®Baterias 9V,» [En línea]. Available: <https://energizer.com.ec/#tamanoV-productEnergizerMax9V>. [Último acceso: 19 Mayo 2020].
- [33] «Google Maps,» [En línea]. Available: <https://www.google.com/maps/place/Universidad+Polit%C3%A9cnica+Salesi>

ana+-+Guayaquil/@-2.220144,-  
79.8877792,18z/data=!4m5!3m4!1s0x902d6e4fced73235:0xb76f5008ec6c43  
45!8m2!3d-2.2201494!4d-79.8866849. [Último acceso: 24 Marzo 2019].

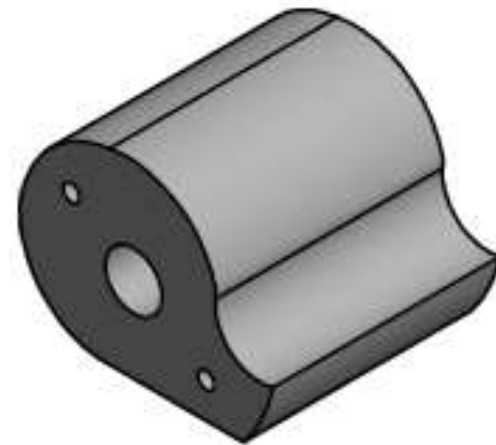
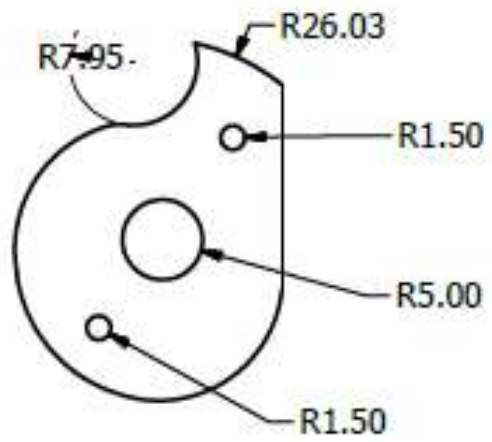
- [34] CDMX ELECTRONICA, «CDMX ELECTRONICA,» 29 AGOSTO 2019. [En línea]. Available: <https://www.cdmxelectronica.com/producto/microcontrolador-atmega328p-au-smd/>.
- [35] Tecnología de los Plásticos, «Tecnología de los Plásticos,» 11 junio 2011. [En línea]. Available: <https://tecnologiadelosplasticos.blogspot.com/2011/06/polietileno-de-alta-densidad.html>. [Último acceso: 8 Junio 2020].
- [36] Dimension Engineering LLC, «Sabertooth dual 12A motor driver,» [En línea]. Available: <https://www.dimensionengineering.com/products/sabertooth2x12>. [Último acceso: 8 Enero 2020].

Anexos

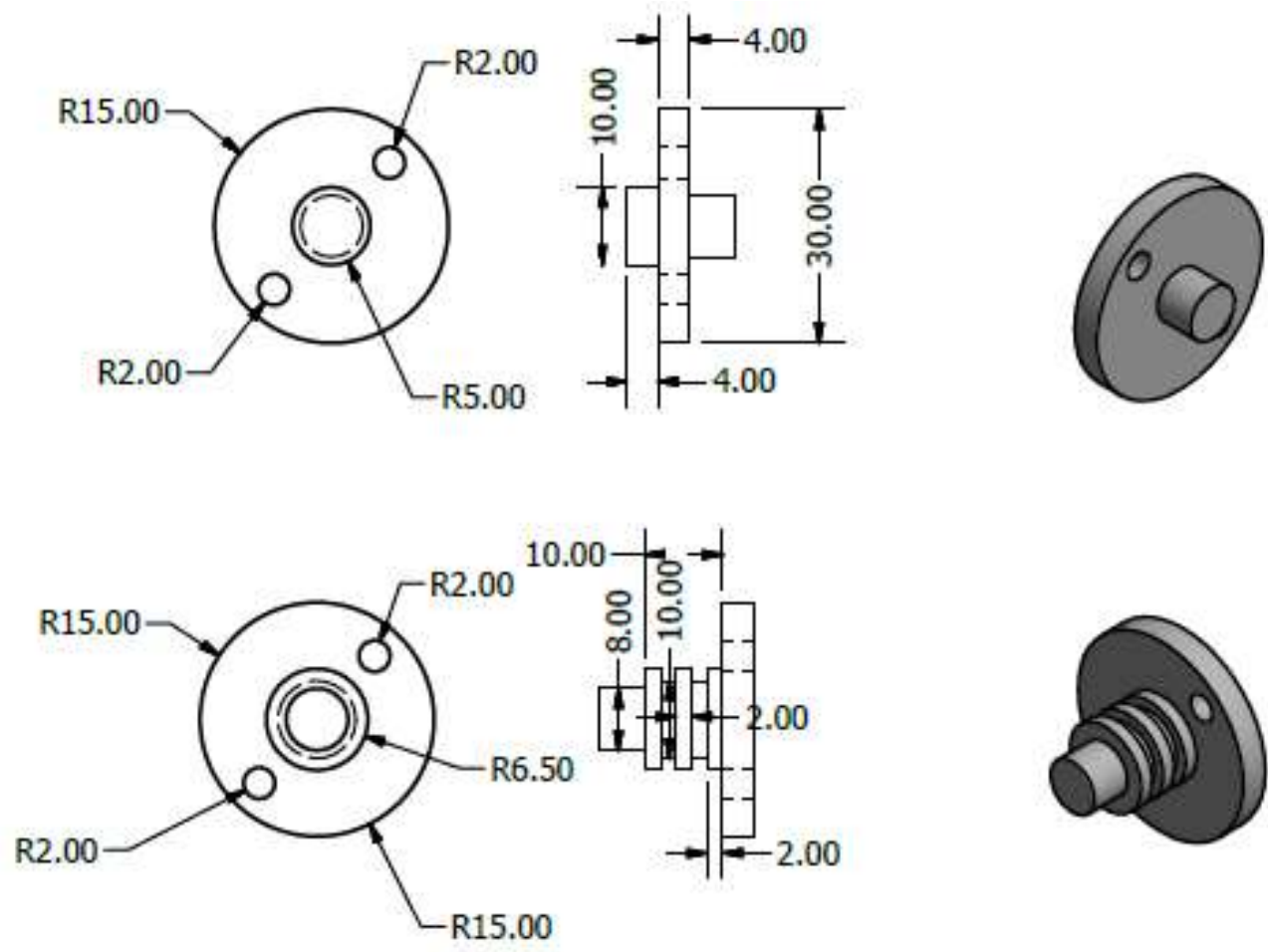
Anexo 1A: Arma del robot 1



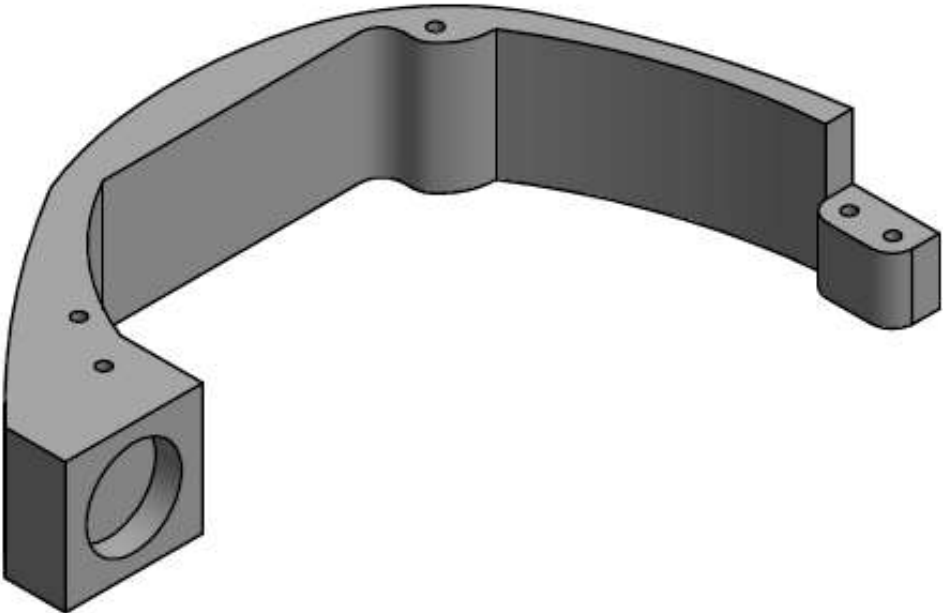
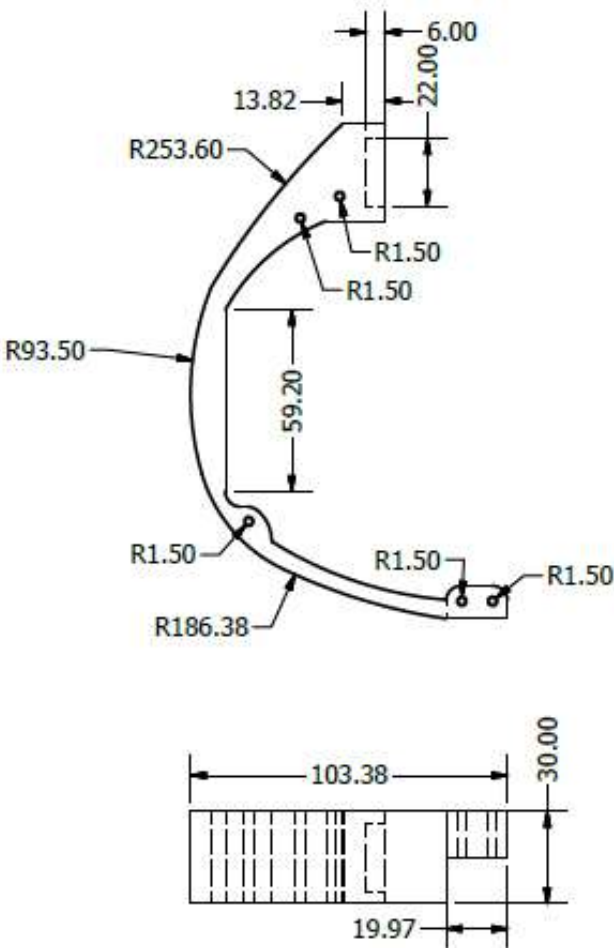
Anexo 1B: Arma del robot 2



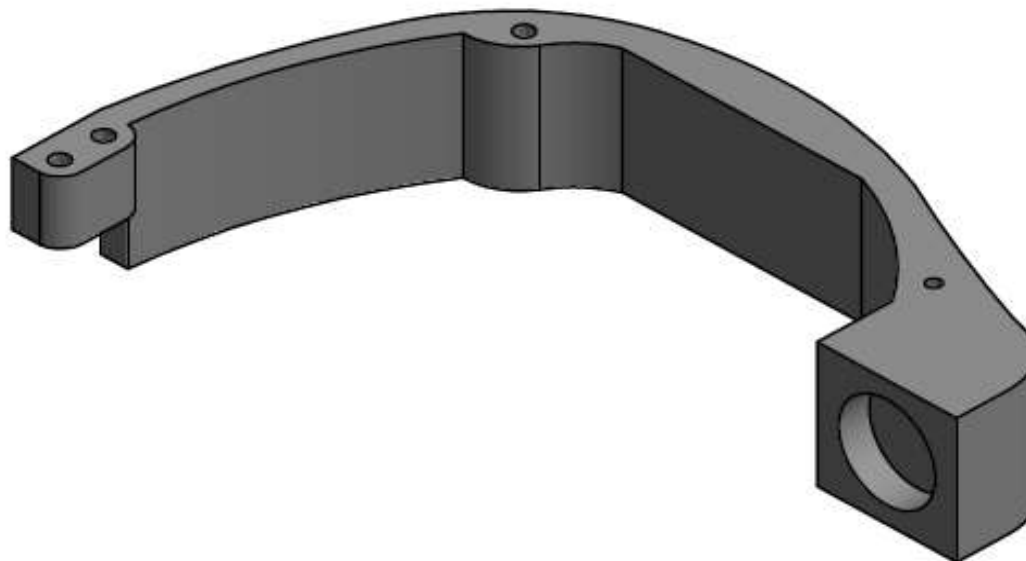
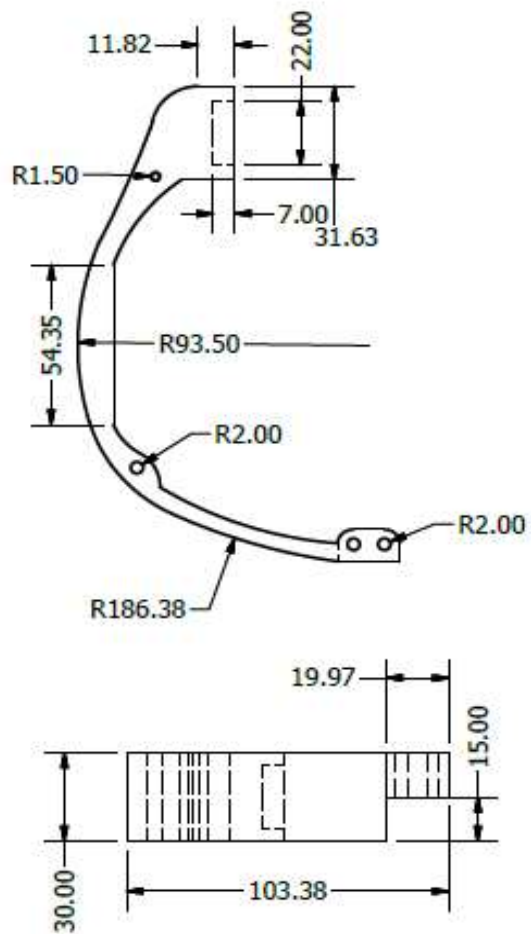
Anexo 1A: Planos de eje y poleas de armas



Anexo 2A: Planos de la estructura lateral derecha

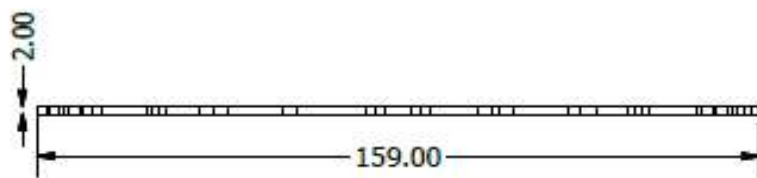
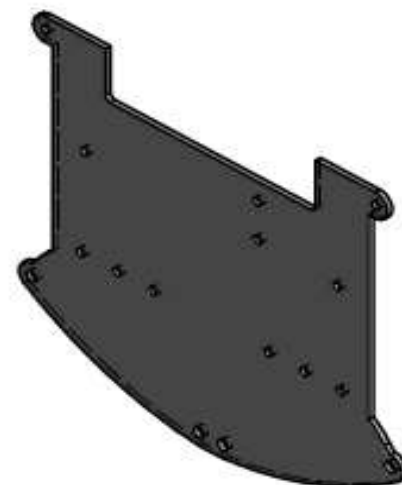
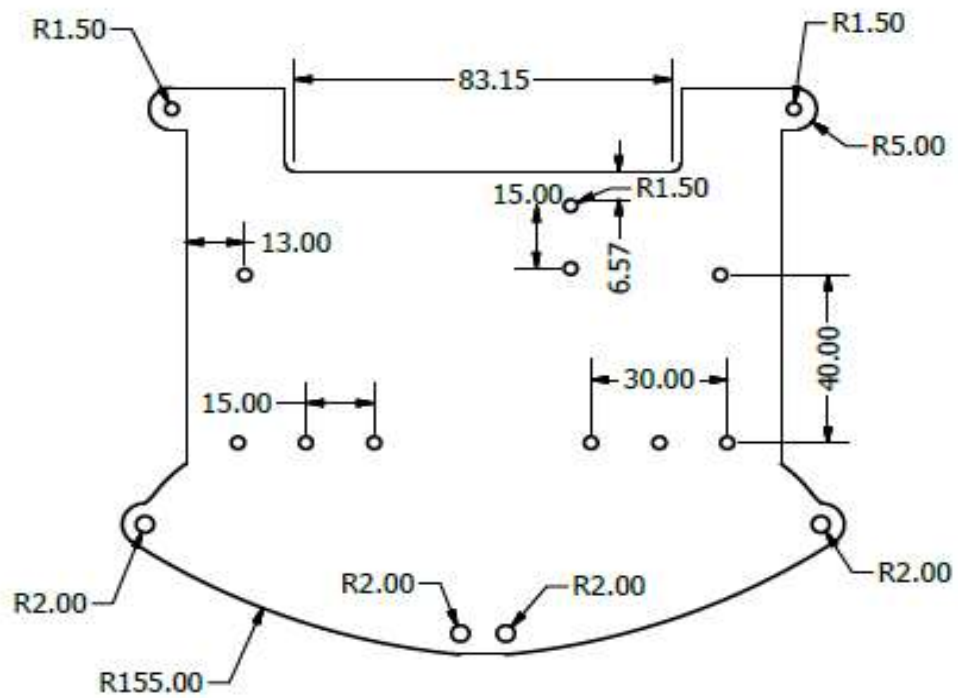


Anexo 2B: Planos de la estructura lateral izquierda

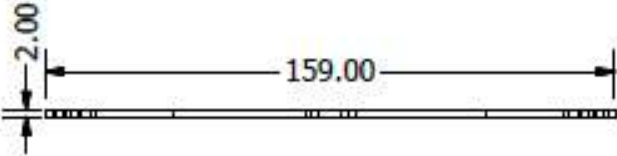
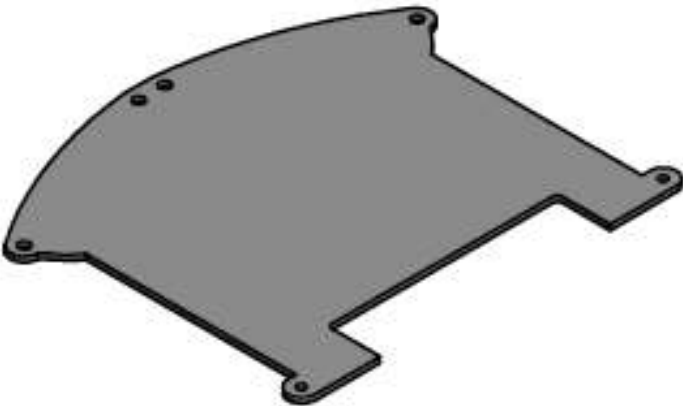
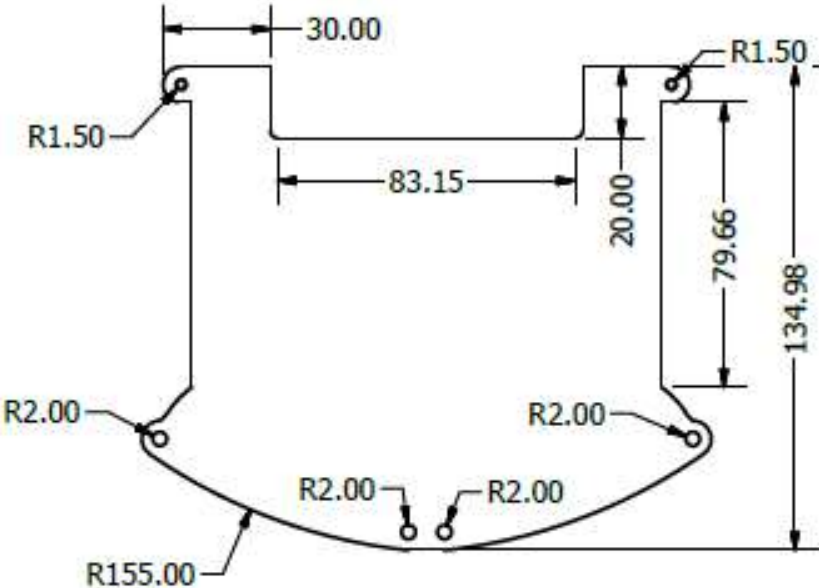




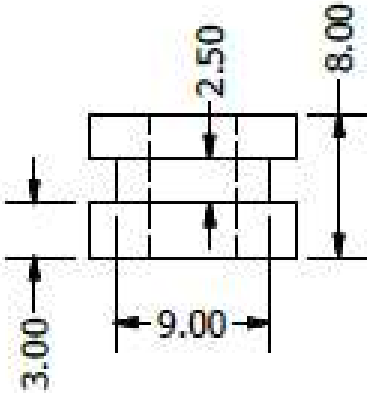
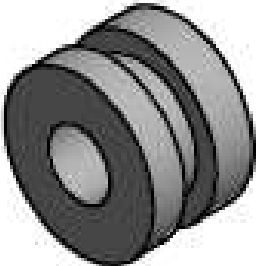
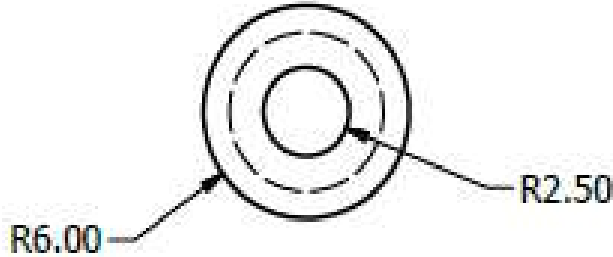
Anexo 3A: Planos de la tapa inferior



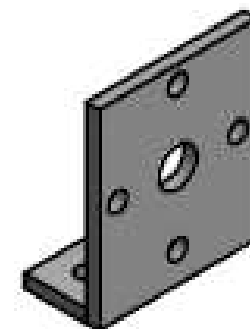
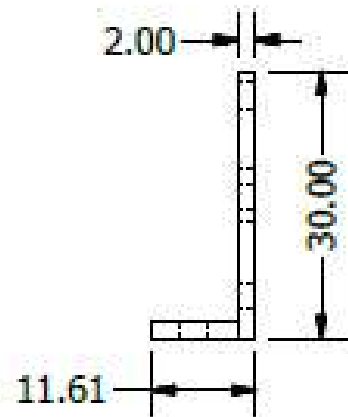
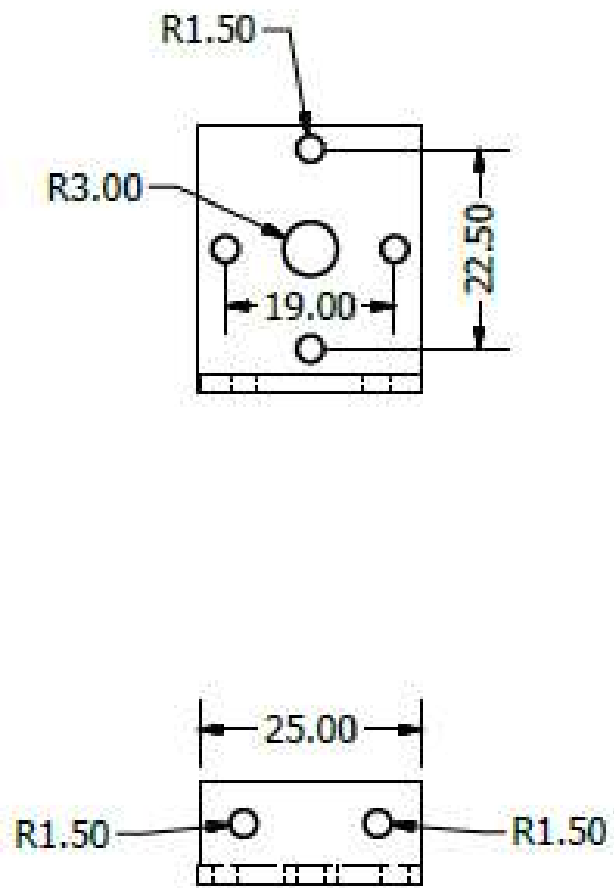
Anexo 3B: Planos de la tapa superior



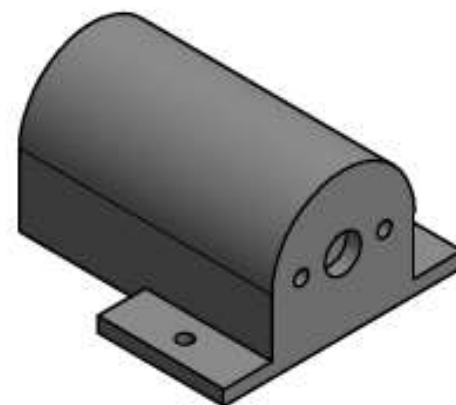
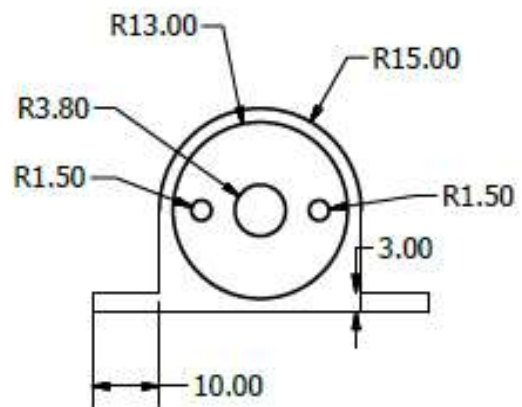
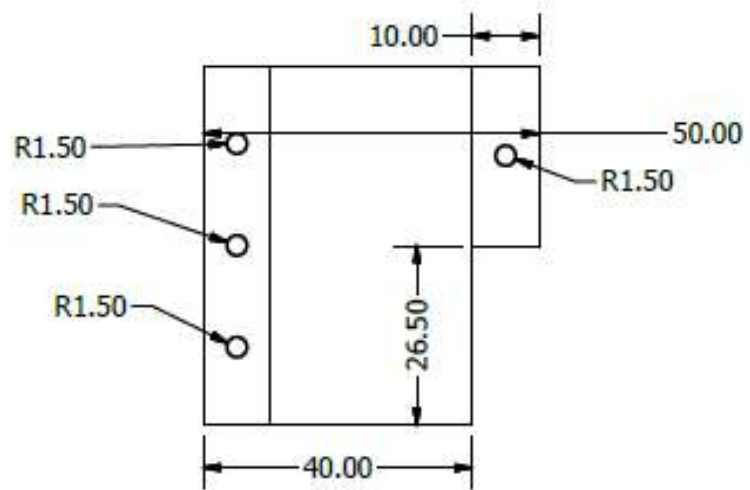
Anexo 4: Planos de poleas para motores brushless



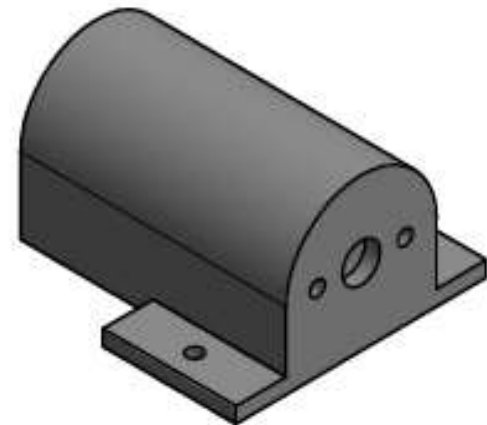
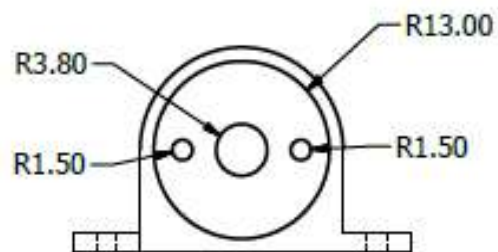
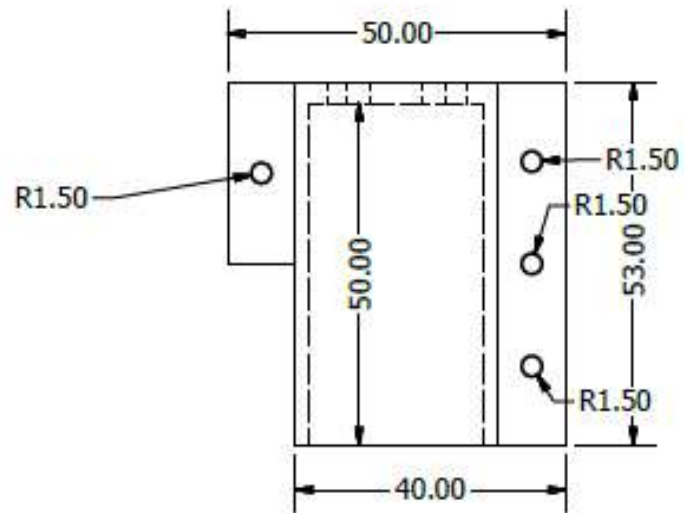
Anexo 5: Planos del soporte de motor Brushless



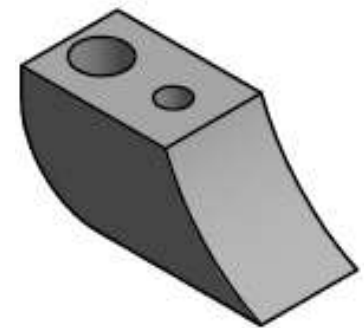
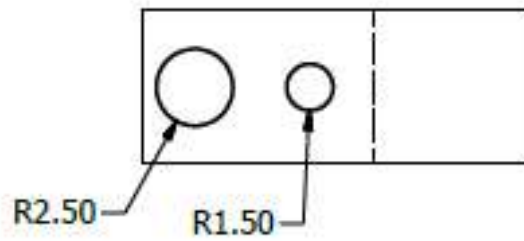
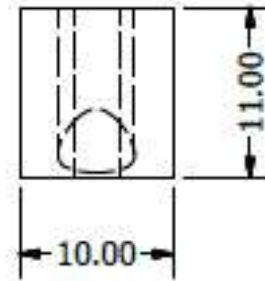
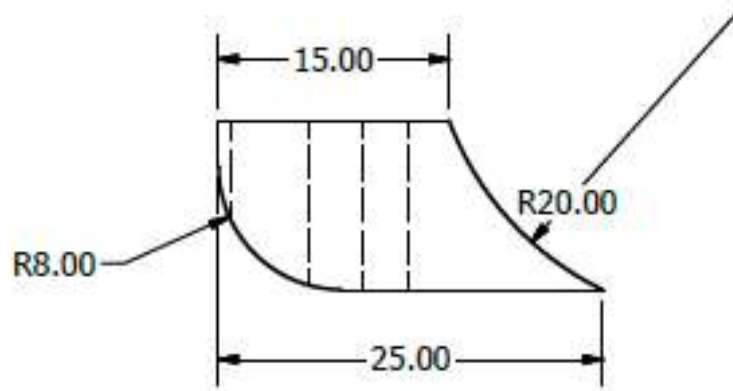
Anexo 6A: Planos del soporte izquierdo del motor DC Pololu 25Dx50L.



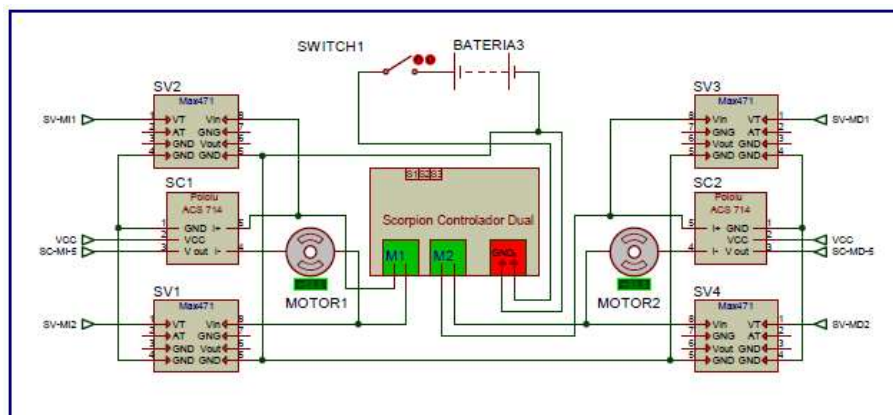
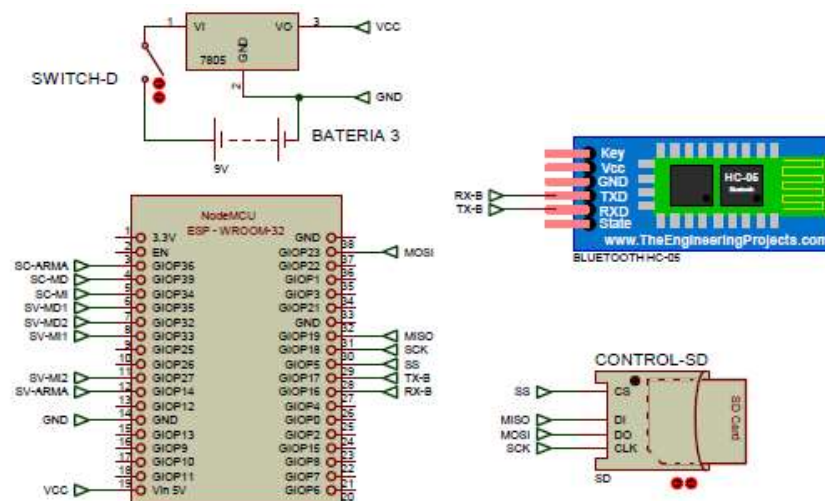
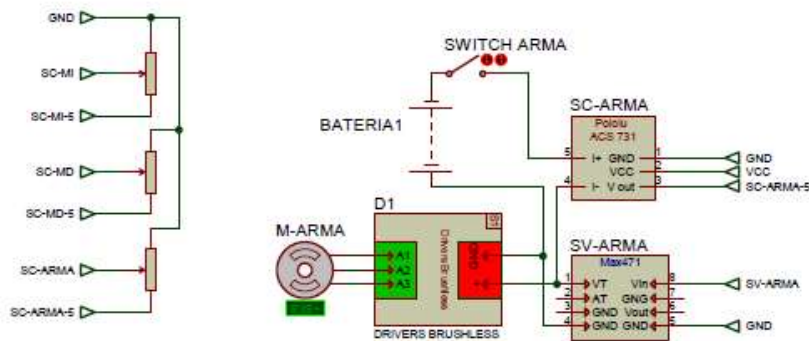
Anexo 6B: Planos del soporte derecho del motor DC Pololu 25Dx50L.



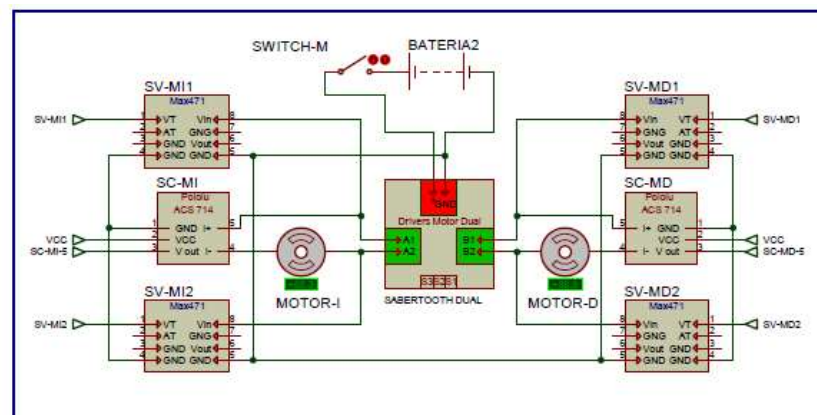
Anexo 7: Planos de patines tipo gancho para el robot 2



## Anexo 8: Diagrama eléctrico de conexiones



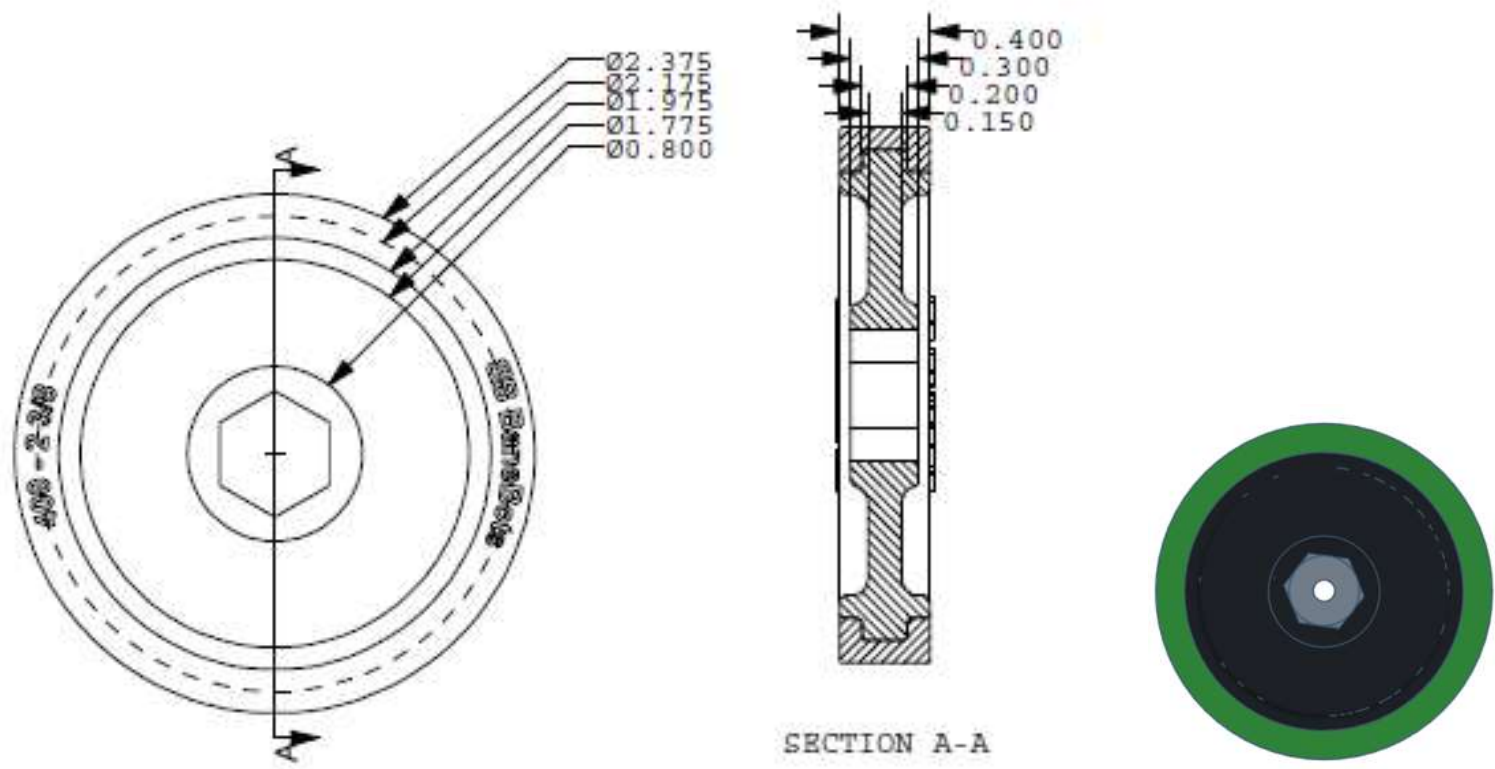
CONTROL RUEDAS ROBOT 1



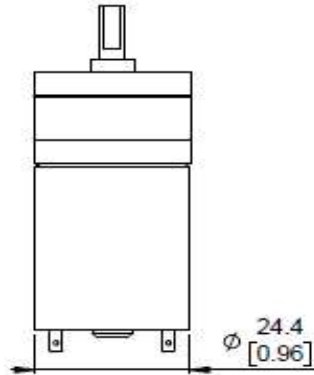
CONTROL RUEDAS ROBOT 2



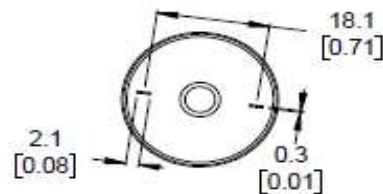
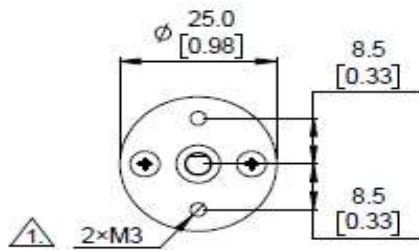
Anexo 9: Planos de las llantas Bannebot 400 23/8



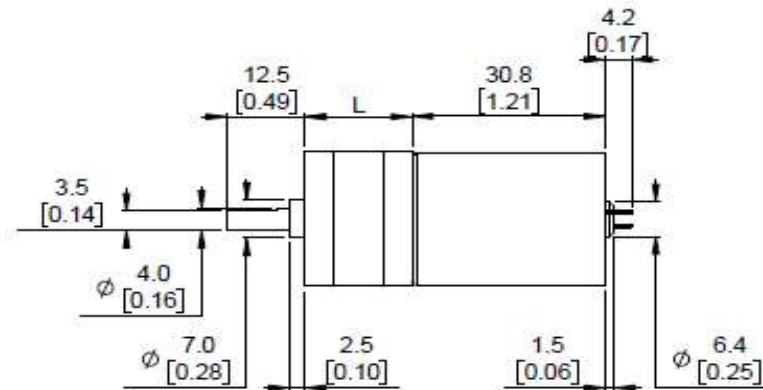
## Anexo 10: Planos estructurales de los motores



| Gear ratio | L               |
|------------|-----------------|
| 4.4:1      | 17 mm [0.68 in] |
| 9.7:1      | 19 mm [0.75 in] |
| 20.4:1     | 19 mm [0.75 in] |
| 34:1       | 21 mm [0.83 in] |
| 47:1       | 21 mm [0.83 in] |
| 75:1       | 23 mm [0.91 in] |
| 99:1       | 23 mm [0.91 in] |
| 172:1      | 25 mm [0.98 in] |
| 227:1      | 25 mm [0.98 in] |
| 378:1      | 27 mm [1.06 in] |
| 499:1      | 27 mm [1.06 in] |



1  
2 Depth of mounting holes is 6.0 mm [0.24 in].  
To get the specified scale, select 100% in print settings.



Scale: 1:1

<https://www.pololu.com/category/115/25d-mm-gearmotors>

Name:  
Metal Gearmotor 25D mm  
(without encoder)

Item number:  
1570-1577, 1581-1591, 3201-  
3207, 3225-3233, 3249-3258

Drawing date:  
8 February 2019

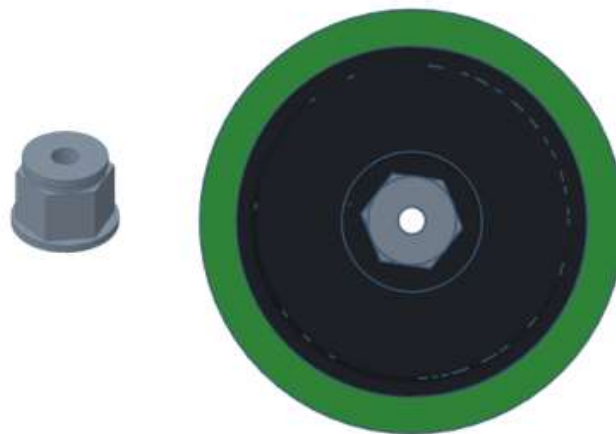
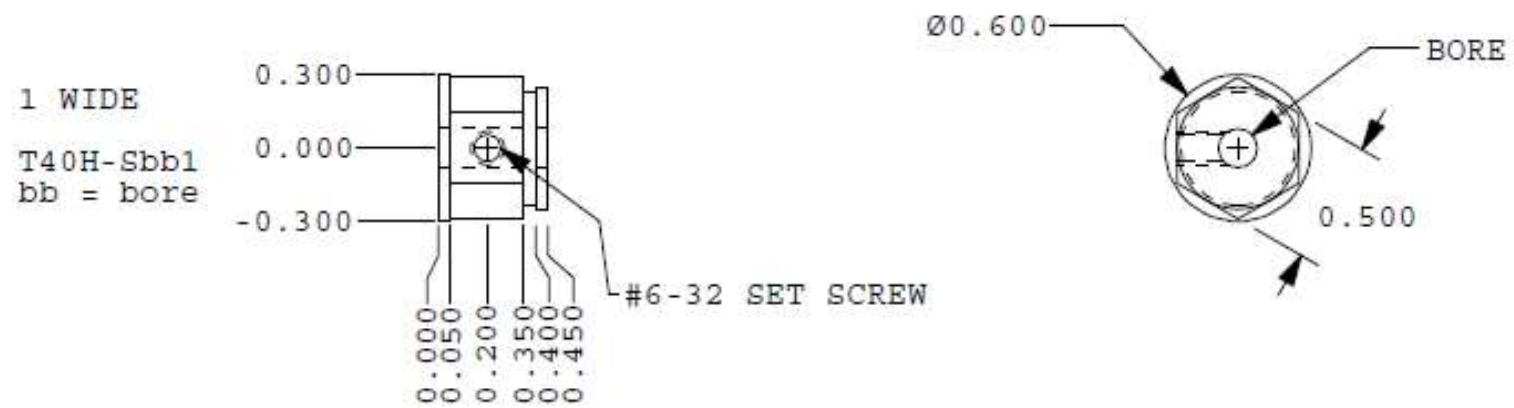
Dev code:  
—

Units: mm  
[in]

Material:  
mixed

**Pololu**  
Robotics & Electronics  
© 2019 Pololu Corporation

Anexo 11: Planos estructurales de los acoplos Bannebots T40H-Sbb1



## Anexo 12: Código Arduino

```
//ROBOT SABERTOOTH LECTURA DE SENSORES
#include "leerSensores.h"
#include "mostrarDatos.h"
#include "controlSD.h"

void setup() {

  pinMode(ledSD,OUTPUT);
  digitalWrite(ledSD,LOW);

  Serial.begin(9600);
  Serial2.begin(9600,SERIAL_8N1,16,17);

  analogReadResolution(10);
  inicioSD();
  valoresReferenciales();
}

void loop() {
  leerDatos();

  monitorSerialVoltajes();
  monitorSerialCorrientes();
  //monitorSerialPotencia();
  grabarDatosSD();
  enviarDatosBluetoothT();

  // Serial.println("");
  // delay(50);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//LIBRERÍA ISD.H
#include "FS.h"
#include "SD.h"
#include "SPI.h"

int ledSD = 12;

//-----editar archivo-----
void appendFile(fs::FS &fs, const char * path, const char * message){
  //Serial.printf("Appending to file: %s\n", path);

  File file = fs.open(path, FILE_APPEND);
  if(!file){
    Serial.println("Failed to open file for appending");
    return;
  }
  if(file.println(message))
  {
    //Serial.println("Message appended");
    file.close();
  }
}
```

```

} else {
  Serial.println("Append failed");
}
}

// -----

//-----editar archivo D-----
void appendFileD(fs::FS &fs, const char * path, float message){
  //Serial.printf("Appending to file: %s\n", path);

  File file = fs.open(path, FILE_APPEND);
  if(!file){
    Serial.println("Failed to open file for appending");
    digitalWrite(ledSD,LOW);
    return;
  }
  if(file.println(message))
  {
    //Serial.println("Message appended");
    file.close();
    digitalWrite(ledSD,HIGH);
  } else {
    Serial.println("Append failed");
  }
}

boolean CMSD = 0;
// ----- iiniciar SD -----
void inicioSD(void)
{
  if(!SD.begin())
  {
    Serial.println("Card Mount Failed");
    return;
  }
  else
  {
    uint8_t cardType = SD.cardType();
    if(cardType == CARD_NONE)
    {
      Serial.println("No SD card attached");
      //return;
    }
    else
    {
      CMSD = 1;
    }
  }
}

if (CMSD == 1)
{
  SD.remove("/Robot2-corrienteMD.txt");
  SD.remove("/Robot2-corrienteMI.txt");
}

```

```

SD.remove("/Robot2-corrienteMA.txt");
SD.remove("/Robot2-voltajeMD.txt");
SD.remove("/Robot2-voltajeMI.txt");
SD.remove("/Robot2-voltajeMA.txt");
SD.remove("/Robot2-tiempo.txt");
SD.remove("/name.txt");

Serial.println("datos Borrados");

File F = SD.open("/name.txt",FILE_APPEND);F.print("Robot2");F.close();

appendFile(SD, "/Robot2-corrienteMD.txt", "CorrienteMD");
appendFile(SD, "/Robot2-corrienteMI.txt", "CorrienteMI");
appendFile(SD, "/Robot2-corrienteMA.txt", "CorrienteMA");
appendFile(SD, "/Robot2-voltajeMD.txt", "VoltajeMD");
appendFile(SD, "/Robot2-voltajeMI.txt", "VoltajeMI");
appendFile(SD, "/Robot2-voltajeMA.txt", "VoltajeMA");
appendFile(SD, "/Robot2-tiempo.txt", "Tiempo");

digitalWrite(ledSD,HIGH);

}
}

// ----- Guardar Datos SD -----

void grabarDatosSD(void)
{
  if (CMSD == 1)
  {
    appendFileD(SD, "/Robot2-corrienteMD.txt",CM1);
    appendFileD(SD, "/Robot2-corrienteMI.txt",CM2);
    appendFileD(SD, "/Robot2-corrienteMA.txt",CMA);
    appendFileD(SD, "/Robot2-voltajeMD.txt",VM1);
    appendFileD(SD, "/Robot2-voltajeMI.txt",VM2);
    appendFileD(SD, "/Robot2-voltajeMA.txt",VMA);
    appendFileD(SD, "/Robot2-tiempo.txt",millis());
  }
}

/////////////////////////////////////////////////////////////////

//LEER SENSORES
//----- pines sensores corriente -----
int ScM1 = 39;
int ScM2 = 34;
int ScMA = 36;

//----- pines sensores voltaje -----
int SvM1A = 35;
int SvM1B = 32;
int SvM2A = 33;
int SvM2B = 14;
int SvMA = 27;

```



```

while(i < 400)
{
  i++;
  value = (float)analogRead(ScM1);
  adcScM1= value + adcScM1;
  value = (float)analogRead(ScM2);
  adcScM2= value + adcScM2;
  value = (float)analogRead(ScMA);
  adcScMA= value + adcScMA;
  delay(1);
}

adcScM1 = adcScM1/i;
adcScM2 = adcScM2/i;
adcScMA = adcScMA/i;

vc1 = (adcScM1 * vRC)/ adcRC;
vc1 = (vc1 *5.05)/3.0;
vc2 = (adcScM2 * vRC)/ adcRC;
vc2 = (vc2 *5.05)/3.0;
vcA = (adcScMA * vRC)/ adcRC;
vcA = (vcA *5.05)/3.0;
}

// ----- Leer sensores -----

void leerDatos(void)
{
  float value = 0;
  int i = 0;
  while(i < 400)
  {
    i++;
    value = (float)analogRead(ScM1);
    adcScM1= value + adcScM1;
    value = (float)analogRead(ScM2);
    adcScM2= value + adcScM2;
    value = (float)analogRead(ScMA);
    adcScMA= value + adcScMA;

    value = (float)analogRead(SvM1A);
    adcSvM1A = adcSvM1A + value;
    value = (float)analogRead(SvM1B);
    adcSvM1B = adcSvM1B + value;
    value = (float)analogRead(SvM2A);
    adcSvM2A = adcSvM2A + value;
    value = (float)analogRead(SvM2B);
    adcSvM2B = adcSvM2B + value;
    value = (float)analogRead(SvMA);
    adcSvMA = adcSvMA + value;
  }

  adcScM1 = adcScM1/i;
  adcScM2 = adcScM2/i;

```



```

adcScMA = adcScMA/i;

adcSvM1A = adcSvM1A/i;
adcSvM1B = adcSvM1B/i;
adcSvM2A = adcSvM2A/i;
adcSvM2B = adcSvM2B/i;
adcSvMA = adcSvMA/i;

// --- conversion a voltaje
vC1 = (adcScM1 * vRC) / adcRC;
vC2 = (adcScM2 * vRC) / adcRC;
vCA = (adcScMA * vRC) / adcRC;

vC1 = (vC1 * 5) / 3.0;
vC2 = (vC2 * 5) / 3.0;
vCA = (vCA * 5) / 3.0;

vM1A = (adcSvM1A * 0.614) / 165;
vM1B = (adcSvM1B * 0.614) / 165;
vM2A = (adcSvM2A * 0.614) / 165;
vM2B = (adcSvM2B * 0.614) / 165;
VMA = (adcSvMA * 0.614) / 165;

vM1A = (vM1A * 5.0);
vM1B = (vM1B * 5.0);
vM2A = (vM2A * 5.0);
vM2B = (vM2B * 5.0);
VMA = (VMA * 5.0);

// --- procesamiento de datos -----

VM1 = abs(vM1B - vM1A);
// if(vM1A > vM1B){VM1 = vM1A;}
// else{VM1 = vM1B;}
VM2 = abs(vM2A - vM2B);
// if(vM2A > vM2B){VM2 = vM2A;}
// else{VM2 = vM2B;}

CM1 = ((vC1 - vc1)/Sensibilidad1) + 0.15;
CM1 = abs(CM1);
CM2 = ((vC2 - vc2)/Sensibilidad2) + 0.17;
CM2 = abs(CM2);
CMA = ((vCA - vcA)/Sensibilidad3) + 0.68;
CMA = abs(CMA);

if(CM1 < 0.08){CM1 = 0; VM1 = 0;}
if(CM2 < 0.08){CM2 = 0; VM2 = 0;}
if(CMA < 0.7){CMA = 0; VMA = 0;} // corrientes menores a 0.7 A no detecta

// --- calculo de potencias -----

```

```

    PM1 = CM1 * VM1;
    PM2 = CM2 * VM2;
    PMA = CMA * VMA;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//MOSTRAR DATOS
char dato;

void monitorSerialVoltajes(void)
{
    Serial.print(" Vm1: ");Serial.print(VM1);
    //Serial.print(adcSvM1A);
    Serial.print(" Vm2: ");Serial.print(VM2);
    //Serial.print(adcSvM1B);
    Serial.print(" VmA: ");Serial.print(VMA);

    //Serial.print(analogRead(ScMA));
    //Serial.println("");
}

void monitorSerialCorrientes(void)
{
    Serial.print(" Cm1: ");Serial.print(CM1);
    // Serial.print(" VCm1: ");Serial.print(vC1);
    Serial.print(" Cm2: ");Serial.print(CM2);
    // Serial.print(" VCm2: ");Serial.print(vC2);
    Serial.print(" CmA: ");Serial.print(CMA);
    // Serial.print(" VCmA: ");Serial.print(vCA);
    Serial.println("");
}

void monitorSerialPotencia(void)
{
    Serial.print(" Pm1: ");Serial.print(PM1);
    Serial.print(" Pm2: ");Serial.print(PM2);
    Serial.print(" PmA: ");Serial.print(PMA);
    //Serial.println("");
}

void enviarDatosBluetoothT(void)
{
    if(Serial2.available()) {
        dato = Serial2.read();
        // Serial.println(dato);
    }
    if(dato == 'D'){
        // Serial2.print("-a");Serial2.print(abs(VM1));
        // Serial2.print("-b");Serial2.print(abs(CM1));
        Serial2.print("-c");Serial2.print(abs(PM1));
        // Serial2.print("-d");Serial2.print(abs(VM2));
    }
}

```

```

// Serial2.print("-e");Serial2.print(abs(CM2));
  Serial2.print("-f");Serial2.print(abs(PM2));
// Serial2.print("-g");Serial2.print(abs(VMA));
// Serial2.print("-h");Serial2.print(abs(CMA));
  Serial2.print("-i");Serial2.print(abs(PMA));
  Serial2.println("-");

  Serial.flush(); dato = ' ';
}

}

```

```

void enviarDatosBluetooth(void)
{
  if(Serial2.available()) {
    dato = Serial2.read();
    Serial.println(dato);
  }
  // if(dato == 'A'){Serial2.print(abs(VM1),2);}
  // else if(dato == 'B'){Serial2.print(abs(VM2),2);}
  // else if(dato == 'C'){Serial2.print(abs(VMA),2);}
  // else if(dato == 'D'){Serial2.print(abs(CM1),2);}
  // else if(dato == 'E'){Serial2.print(abs(CM2),2);}
  // else if(dato == 'F'){Serial2.print(abs(CMA),2);}
  // else if(dato == 'G'){Serial2.print(abs(PM1),2);}
  // else if(dato == 'H'){Serial2.print(abs(PM2),2);}
  // else if(dato == 'I'){Serial2.print(abs(PMA),2);}

  if(dato == 'A'){Serial2.print("111.11");}
  else if(dato == 'B'){Serial2.print("222.22");}
  else if(dato == 'C'){Serial2.print("333.33");}
  else if(dato == 'D'){Serial2.print("444.44");}
  else if(dato == 'E'){Serial2.print("555.55");}
  else if(dato == 'F'){Serial2.print("666.66");}
  else if(dato == 'G'){Serial2.print("777.77");}
  else if(dato == 'H'){Serial2.print("888.88");}
  else if(dato == 'I'){Serial2.print("999.99");}
  Serial.flush(); dato = ' ';
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//ROBOT SCORPIO LECTURA DE DATOS
#include "leerSensores.h"
#include "mostrarDatos.h"
#include "controlSD.h"

void setup() {

```

```

pinMode(ledSD,OUTPUT);
digitalWrite(ledSD,LOW);

Serial.begin(9600);
Serial2.begin(9600,SERIAL_8N1,16,17);

analogReadResolution(10);
inicioSD();
valoresReferenciales();
}

void loop() {
  leerDatos();

  monitorSerialVoltajes();
  monitorSerialCorrientes();
  //monitorSerialPotencia();
  grabarDatosSD();
  enviarDatosBluetoothT();

  // Serial.println("");
  // delay(50);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//CONTROL ISD.H
#include "FS.h"
#include "SD.h"
#include "SPI.h"

int ledSD = 13;

//-----editar archivo-----
void appendFile(fs::FS &fs, const char * path, const char * message){
  //Serial.printf("Appending to file: %s\n", path);

  File file = fs.open(path, FILE_APPEND);
  if(!file){
    Serial.println("Failed to open file for appending");
    return;
  }
  if(file.println(message))
  {
    //Serial.println("Message appended");
    file.close();
  } else {
    Serial.println("Append failed");
  }
}
// -----

//-----editar archivo D-----
void appendFileD(fs::FS &fs, const char * path, float message){
  //Serial.printf("Appending to file: %s\n", path);

```

```

File file = fs.open(path, FILE_APPEND);
if(!file){
  Serial.println("Failed to open file for appending");
  digitalWrite(ledSD,LOW);
  return;
}
if(file.println(message))
{
  //Serial.println("Message appended");
  file.close();
  // confirmacion de SD OK -----
  digitalWrite(ledSD,HIGH);

} else {
  Serial.println("Append failed");
}
}

```

```
// -----
```

```

boolean CMSD = 0;
// ----- iiniciar SD -----
void inicioSD(void)
{
  if(!SD.begin())
  {
    Serial.println("Card Mount Failed");
    return;
  }
  else
  {
    uint8_t cardType = SD.cardType();
    if(cardType == CARD_NONE)
    {
      Serial.println("No SD card attached");
      //return;
    }
    else
    {
      CMSD = 1;
    }
  }
}

```

```

if (CMSD == 1)
{
  SD.remove("/Robot1-corrienteMD.txt");
  SD.remove("/Robot1-corrienteMI.txt");
  SD.remove("/Robot1-corrienteMA.txt");
  SD.remove("/Robot1-voltajeMD.txt");
  SD.remove("/Robot1-voltajeMI.txt");
}

```

```

SD.remove("/Robot1-voltajeMA.txt");
SD.remove("/Robot1-tiempo.txt");
SD.remove("/name.txt");

Serial.println("datos Borrados");

File F = SD.open("/name.txt",FILE_APPEND);F.print("Robot1");F.close();
appendFile(SD, "/Robot1-corrienteMD.txt", "CorrienteMD");
appendFile(SD, "/Robot1-corrienteMI.txt", "CorrienteMI");
appendFile(SD, "/Robot1-corrienteMA.txt", "CorrienteMA");
appendFile(SD, "/Robot1-voltajeMD.txt", "VoltajeMD");
appendFile(SD, "/Robot1-voltajeMI.txt", "VoltajeMI");
appendFile(SD, "/Robot1-voltajeMA.txt", "VoltajeMA");
appendFile(SD, "/Robot1-tiempo.txt", "Tiempo");

digitalWrite(ledSD,HIGH);

}
}

// ----- Guardar Datos SD -----

void grabarDatosSD(void)
{
  if (CMSD == 1)
  {
    appendFileD(SD, "/Robot1-corrienteMD.txt",CM1);
    appendFileD(SD, "/Robot1-corrienteMI.txt",CM2);
    appendFileD(SD, "/Robot1-corrienteMA.txt",CMA);
    appendFileD(SD, "/Robot1-voltajeMD.txt",VM1);
    appendFileD(SD, "/Robot1-voltajeMI.txt",VM2);
    appendFileD(SD, "/Robot1-voltajeMA.txt",VMA);
    appendFileD(SD, "/Robot1-tiempo.txt",millis());
  }
}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//LEER SENSORES

//----- pines sensores corriente -----
int ScM1 = 39;
int ScM2 = 34;
int ScMA = 36;

//----- pines sensores voltaje -----
int SvM1A = 32;//35
int SvM1B = 33;
int SvM2A = 27;
int SvM2B = 14;
int SvMA = 35;

//----- Variables -----
// ---- sensibilidad sensores de corriente-----
float Sensibilidad1 = 0.187;

```



```

{
  i++;
  value = (float)analogRead(ScM1);
  adcScM1= value + adcScM1;
  value = (float)analogRead(ScM2);
  adcScM2= value + adcScM2;
  value = (float)analogRead(ScMA);
  adcScMA= value + adcScMA;
  delay(1);
}

adcScM1 = adcScM1/i;
adcScM2 = adcScM2/i;
adcScMA = adcScMA/i;

vc1 = (adcScM1 * vRC)/ adcRC;
vc1 = (vc1 *5)/3;
vc2 = (adcScM2 * vRC)/ adcRC;
vc2 = (vc2 *5)/3;
vcA = (adcScMA * vRC)/ adcRC;
vcA = (vcA *5)/3;
}

// ----- Leer sensores -----

void leerDatos(void)
{
  float value = 0;
  int i = 0;
  while(i < 400)
  {
    i++;
    value = (float)analogRead(ScM1);
    adcScM1= value + adcScM1;
    value = (float)analogRead(ScM2);
    adcScM2= value + adcScM2;
    value = (float)analogRead(ScMA);
    adcScMA= value + adcScMA;

    value = (float)analogRead(SvM1A);
    adcSvM1A = adcSvM1A + value;
    value = (float)analogRead(SvM1B);
    adcSvM1B = adcSvM1B + value;
    value = (float)analogRead(SvM2A);
    adcSvM2A = adcSvM2A + value;
    value = (float)analogRead(SvM2B);
    adcSvM2B = adcSvM2B + value;
    value = (float)analogRead(SvMA);
    adcSvMA = adcSvMA + value;
  }

  adcScM1 = adcScM1/i;
  adcScM2 = adcScM2/i;
  adcScMA = adcScMA/i;
}

```



```

adcSvM1A = adcSvM1A/i;
adcSvM1B = adcSvM1B/i;
adcSvM2A = adcSvM2A/i;
adcSvM2B = adcSvM2B/i;
adcSvMA = adcSvMA/i;

// --- conversion a voltaje
vC1 = (adcScM1 * vRC) / adcRC;
vC2 = (adcScM2 * vRC) / adcRC;
vCA = (adcScMA * vRC) / adcRC;

vC1 = (vC1 * 5) / 3;
vC2 = (vC2 * 5) / 3;
vCA = (vCA * 5) / 3;

vM1A = (adcSvM1A * 2.246) / 652.0;
vM1B = (adcSvM1B * 2.246) / 652.0;
vM2A = (adcSvM2A * 2.246) / 652.0;
vM2B = (adcSvM2B * 2.243) / 648.0;
VMA = (adcSvMA * 3.03) / 960.0;

vM1A = (vM1A * 5.0);
vM1B = (vM1B * 5.0);
vM2A = (vM2A * 5.0);
vM2B = (vM2B * 5.0);
VMA = (VMA * 5.0);

// --- procesamiento de datos -----

VM1 = 0;
if(vM1A > vM1B){VM1 = vM1A;}
else{VM1 = vM1B;}
VM1 = abs(VM1);
VM2 = 0;
if(vM2A > vM2B){VM2 = vM2A;}
else{VM2 = vM2B;}
VM2 = abs(VM2) ;

CM1 = ((vC1 - vc1)/Sensibilidad1) - 0.05;
CM1 = abs(CM1);
CM2 = ((vC2 - vc2)/Sensibilidad2) - 0.02;
CM2 = abs(CM2);
CMA = ((vCA - vcA)/Sensibilidad3) - 0.08;
CMA = abs(CMA);

if(CM1 < 0.08){CM1 = 0; VM1 = 0;}
if(CM2 < 0.08){CM2 = 0; VM2 = 0;}
if(CMA < 0.6){ VMA = 0; VMA = 0;} // corrientes menores a 0.7 A no detecta
else{vma1 = VMA; }

```

```

// --- calculo de potencias -----
PM1 = CM1 * VM1;
PM2 = CM2 * VM2;
PMA = CMA * VMA;

}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//MOSTRAR DATOS
// char dato;

void monitorSerialVoltajes(void)
{
  Serial.print(" Vm1: ");Serial.print(VM1);
  Serial.print(" Vm2: ");Serial.print(VM2);
  Serial.print(" VmA: ");Serial.print(VMA);
  //Serial.print(" VCmA: ");Serial.print(vCA);
  // Serial.println("");
}

void monitorSerialCorrientes(void)
{
  Serial.print(" Cm1: ");Serial.print(CM1);
  Serial.print(" Cm2: ");Serial.print(CM2);
  Serial.print(" CmA: ");Serial.print(CMA);
  // Serial.print(" CmA: ");Serial.print(vcA);
  // Serial.print(" VCmA: ");Serial.print(vCA);
  Serial.println("");
}

void monitorSerialPotencia(void)
{
  Serial.print(" Pm1: ");Serial.print(PM1);
  Serial.print(" Pm2: ");Serial.print(PM2);
  Serial.print(" PmA: ");Serial.print(PMA);
  //Serial.println("");
}

void enviarDatosBluetoothT(void)
{
  // dato = 'D';
  if(Serial2.available()) {
    dato = Serial2.read();
    Serial.println(dato);}

if(dato == 'D'){
// Serial2.print("-a");Serial2.print((VM1));
// Serial2.print("-b");Serial2.print((CM1));
  Serial2.print("-c");Serial2.print((PM1));
// Serial2.print("-d");Serial2.print((VM2));
}

```

```

// Serial2.print("-e");Serial2.print((CM2));
  Serial2.print("-f");Serial2.print((PM2));
// Serial2.print("-g");Serial2.print((VMA));
// Serial2.print("-h");Serial2.print((CMA));
  Serial2.print("-i");Serial2.print((PMA));
  Serial2.println("-");

  Serial.flush(); dato = ' ';
}
}

```

```

void enviarDatosBluetooth(void)
{
  if(Serial2.available()) {
    dato = Serial2.read();
    Serial.println(dato);
  }
  // if(dato == 'A'){Serial2.print(abs(VM1),2);}
  // else if(dato == 'B'){Serial2.print(abs(VM2),2);}
  // else if(dato == 'C'){Serial2.print(abs(VMA),2);}
  // else if(dato == 'D'){Serial2.print(abs(CM1),2);}
  // else if(dato == 'E'){Serial2.print(abs(CM2),2);}
  // else if(dato == 'F'){Serial2.print(abs(CMA),2);}
  // else if(dato == 'G'){Serial2.print(abs(PM1),2);}
  // else if(dato == 'H'){Serial2.print(abs(PM2),2);}
  // else if(dato == 'I'){Serial2.print(abs(PMA),2);}

  if(dato == 'A'){Serial2.print("111.11");}
  else if(dato == 'B'){Serial2.print("222.22");}
  else if(dato == 'C'){Serial2.print("333.33");}
  else if(dato == 'D'){Serial2.print("444.44");}
  else if(dato == 'E'){Serial2.print("555.55");}
  else if(dato == 'F'){Serial2.print("666.66");}
  else if(dato == 'G'){Serial2.print("777.77");}
  else if(dato == 'H'){Serial2.print("888.88");}
  else if(dato == 'I'){Serial2.print("999.99");}
  Serial.flush(); dato = ' ';
}
}

```

## Anexo 12: Código MATLAB gráficos de robots de batallas

```

%% grafica Datos Robot Batallas
%% ----- Inicializacion -----
clc
clear all
warning off;

close.figure(10),figure(11),figure(12))
disp(' >>> Programa Graficas de Datos ')

```

```
%% ----- Inicializacion -----
```

```
cMI = 0;  
cMD = 0;  
cMA = 0;
```

```
vMI = 0;  
vMD = 0;  
vMA = 0;
```

```
pMI = 0;  
pMD = 0;  
pMA = 0;
```

```
c = 0;  
v = 0;
```

```
%% ----- Ingreso Datos -----
```

```
ruta = uigetdir('Seleccione Carpeta');  
length(ruta);  
if(length(ruta) > 1)
```

```
    archivoname = strcat(ruta, 'name.txt');  
    nRobot=strcat('\',fileread(archivoname));
```

```
    archivoR = (strcat(ruta,[nRobot '-tiempo.txt']));  
    text = fileread(archivoR);  
    D = importdata(archivoR);  
    dTiempo = D.data(:,1);  
    dTiempo = (dTiempo / 1000);  
    xmin = min(dTiempo);  
    xmax = max(dTiempo);
```

```
    archivoR = (strcat(ruta,[nRobot '-voltajeMD.txt']));  
    text = fileread(archivoR);  
    D = importdata(archivoR);  
    vMD = D.data(:,1);
```

```
    archivoR = (strcat(ruta,[nRobot '-voltajeMI.txt']));  
    text = fileread(archivoR);  
    D = importdata(archivoR);  
    vMI = D.data(:,1);
```

```
    archivoR = (strcat(ruta,[nRobot '-voltajeMA.txt']));  
    text = fileread(archivoR);  
    D = importdata(archivoR);  
    vMA = D.data(:,1);
```

```
v = 1;
```

```
    archivoR = (strcat(ruta,[nRobot '-corrienteMD.txt']));  
    text = fileread(archivoR);
```

```

D = importdata(archivoR);
cMD = D.data(:,1);

archivoR = (strcat(ruta,[nRobot '-corrienteMI.txt']));
text = fileread(archivoR);
D = importdata(archivoR);
cMI = D.data(:,1);

archivoR = (strcat(ruta,[nRobot '-corrienteMA.txt']));
text = fileread(archivoR);
D = importdata(archivoR);
cMA = D.data(:,1);

c = 1;

end% end if ruta

if((c ==1)&&(v == 1))
    nci = length(cMI);
    ncd = length(cMD);
    nca = length(cMA);

    nvi = length(vMI);
    nvd = length(vMD);
    nva = length(vMA);

    nvt = length(dTiempo);

    n = nci;
    if(n > ncd)
        n = ncd;
    else if(n > nca)
        n = nvi;
    else if (n > nvd)
        n = nvd;
    else if (n > nva)
        n = nva;
    else if (n > nvt)
        n = nvt;
    end
    end
    end
    end
end

% cMI = abs(cMI);
% cMD = abs(cMD);
% cMA = abs(cMA);

pMI = abs(cMI(1:n)) .* vMI(1:n);
pMD = abs(cMD(1:n)) .* vMD(1:n);
pMA = abs(cMA(1:n)) .* vMA(1:n);

grafico1 = figure(1);

```

```

set(grafico1,'units','normalized','outerposition',[0 0 1 1])
set(grafico1,'color','white')
set(grafico1,'NumberTitle','off')
set(grafico1,'Name',['Graficos datos SD ' nRobot])

subplot(3,1,1)
plot(dTiempo(1:n),cMI(1:n),'color','b');hold on
plot(dTiempo(1:n),cMD(1:n),'color','k');
plot(dTiempo(1:n),cMA(1:n),'color','r');hold off
xlim([xmin xmax])
title('Grafica Corrientes [A]')
legend('Motor Rueda Izquierdo','Motor Rueda Derecha','Motor del Arma')
xlabel('Tiempo [s]')
ylabel('Corriente [A]')
subplot(3,1,2)
plot(dTiempo(1:n),vMI(1:n),'color','b');hold on
plot(dTiempo(1:n),vMD(1:n),'color','k');
plot(dTiempo(1:n),vMA(1:n),'color','r');hold off
xlim([xmin xmax])
title('Grafica Voltajes [V]')
xlabel('Tiempo [s]')
ylabel('Voltaje [V]')
subplot(3,1,3)
plot(dTiempo(1:n),pMI(1:n),'color','b');hold on
plot(dTiempo(1:n),pMD(1:n),'color','k');
plot(dTiempo(1:n),pMA(1:n),'color','r');hold off
xlim([xmin xmax])
title('Grafica Potencias [W]')
xlabel('Tiempo [s]')
ylabel('Potencia [W]')

end

```

### Anexo 13: Código MATLAB gráficos de robots de batallas por bluetooth

```

clc
close all
clear all
warning off;

%% MENU SELCCIONAR ROBOT BLUETOOTH -----
nName =menu('Seleccione Robot','Robot 1','Robot 2');
nRobot = {'Robot1' 'Robot2'};
%% SELECCIONAR CAMARA STREAMING -----
Camaras = imaqhwinfo('winvideo'); % indica puertos de camaras conectadas
nCamaras = length(Camaras.DeviceIDs);
nameCamaras = {' ' ' '};
for i=1:nCamaras

    infoCamaras = imaqhwinfo('winvideo',Camaras.DeviceIDs{i}); % muestra
informacion de la camara
    nameCamaras{i} = infoCamaras.DeviceName;
end

```

```

%disp(nameCamaras)
%menu(nameCamaras)

selCamara = chooseCamara(nameCamaras);
%%----- PROGRAMA STRIMING -----
if(nName ~= 0)

    B = Bluetooth(nRobot{nName},1);          % CREA VARIABLE BLUETOOTH
    fopen(B);                                % ABRE PUERTO SERIAL BLUETOOTH

    disp('Programa datos Bluettoth iniciado...')

    VM1 = 0; CM1 = 0; PM1 =0;
    VM2 = 0; CM2 = 0; PM2 =0;
    VMA = 0; CMA = 0; PMA =0;

    [logoUPS,map1] = imread('logoUPS.jpg');
    if(nName == 1)
        [logoRobot,map2] = imread('logoRobot1.jpeg');
    else
        [logoRobot,map2] = imread('logoRobot2.jpeg');
    end
    [titulo,map3] = imread('titulo.jpg');
%% LINEA DE POTENCIA
fr=50;
nmuestra= 100;

voltaje1 = zeros(nmuestra,1);
voltaje2 = zeros(nmuestra,1);
voltajeA = zeros(nmuestra,1);

corriente1 = zeros(nmuestra,1);
corriente2 = zeros(nmuestra,1);
corrienteA = zeros(nmuestra,1);

potencia1 = zeros(nmuestra,1); %Vector de Datos
potencia2 = zeros(nmuestra,1); %Vector de Datos
potenciaA = zeros(nmuestra,1); %Vector de Datos

ylim([-0.5 30]);
xlim([0 (nmuestra-1)/fr]);
grid

uicontrol('Style','Pushbutton','String','Detener',...
'FontSize',14,...
'Position', [500 5 100 40],...
'Callback','parar = 0;');%Boton Parar

grafico1 = figure(1);

set(grafico1,'units','normalized','outerposition',[0 0 1 1])
set(grafico1,'color','white')

```

```

set(grafico1,'NumberTitle','off')
set(grafico1,'Name','Streaming Robot Data')

subplot(3,4,2:3)
    imshow(titulo,map3);
    axis off
subplot(3,4,1)
    imshow(logoUPS,map1);
subplot(3,4,4)
    imshow(logoRobot,map2);

disp('inicando Camara..')
subplot(3,4,[5 6 9 10])
if(selCamara ~=0)
camara = videoinput('winvideo',selCamara); % seleccion de camara
camara.ReturnedColorSpace = 'rgb';
camaraRes = get(camara,'VideoResolution');
nBands = get(camara,'NumberOfBands');
hImage = image(zeros(camaraRes(2), camaraRes(1), nBands ));
preview(camara, hImage);
title(nRobot{nName})
drawnow
disp('camara iniciada..')
else
    imshow(logoRobot)
    axis off
end%end if sel camara
n1=1;
%% -----

n1=0;
nDatos = 1;
parar = 1;
op = 2;
tipoG = 1;
while (parar == 1)
    n1= n1 +1;
    fwrite(B,'D','uchar');
    Dato = fgets(B); %disp(Dato)
    n = length(Dato)-2;
%%     interpretar datos seriales
    if(n > 0)
        temp = "";
        d1 = '0';
        tDato = "";
        for i=1:n
            indato = Dato(i);
            if ((indato ~= 'a')&&(indato ~= 'b')&&(indato ~= 'c')&&(indato ~=
'd')&&(indato ~= 'e')&&(indato ~= 'f')&&(indato ~= 'g')&&(indato ~= 'h')&&(indato ~=
'i')&&(indato ~= '-'))
                d1 = strcat(d1,indato);
            else if(indato ~= '-')
                tDato = indato;
            end% if indato '-'

```



```

end% if indato

if(indato == '-')
    switch tDato
        case 'a'
            VM1 = str2num(d1);
        case 'b'
            CM1 = str2num(d1);
        case 'c'
            PM1 = str2num(d1);
        case 'd'
            VM2 = str2num(d1);
        case 'e'
            CM2 = str2num(d1);
        case 'f'
            PM2 = str2num(d1);
        case 'g'
            VMA = str2num(d1);
        case 'h'
            CMA = str2num(d1);
        case 'i'
            PMA = str2num(d1);
        otherwise

            end%end swithc
            d1 = '0';
        end

end%for i
end % end if n > 0 interpretar datos

if (nDatos <= nmuestra)

    voltaje1(nDatos) = VM1;
    voltaje2(nDatos) = VM2;
    voltajeA(nDatos) = VMA;

    corriente1(nDatos) = CM1;
    corriente2(nDatos) = CM2;
    corrienteA(nDatos) = CMA;

    potencia1(nDatos) = PM1;
    potencia2(nDatos) = PM2;
    potenciaA(nDatos) = PMA;
    nDatos = nDatos + 1;
else
    voltaje1(1:end-1) = voltaje1(2:end);voltage1(end) = VM1;
    voltaje2(1:end-1) = voltaje2(2:end);voltage2(end) = VM2;
    voltajeA(1:end-1) = voltajeA(2:end);voltageA(end) = VMA;

    corriente1(1:end-1) = corriente1(2:end);corriente1(end) = CM1;
    corriente2(1:end-1) = corriente2(2:end);corriente2(end) = CM2;
    corrienteA(1:end-1) = corrienteA(2:end);corrienteA(end) = CMA;

```

```

potencia1(1:end-1) = potencia1(2:end);potencia1(end) = PM1;
potencia2(1:end-1) = potencia2(2:end);potencia2(end) = PM2;
potenciaA(1:end-1) = potenciaA(2:end);potenciaA(end) = PMA;
end

```

```

figure(1)

```

```

subplot(3,4,[7 8])
plot(potencia1,'color','g');hold on
plot(potencia2,'color','b');hold off
ylim([0 20]);
grid
title('Potencia Ruedas')
legend('Motor D','Motor I')
xlabel('Tiempo [s]')
ylabel('Potencia [W]','FontSize',14)
set(gca,'XTickLabel',[])

```

```

subplot(3,4,[11 12])
plot(potenciaA,'color','r');
ylim([0 100]);
grid
title('Potencia Motor Arma')
xlabel('Tiempo [s]')
ylabel('Potencia [W]','FontSize',14)
set(gca,'XTickLabel',[])

```

```

drawnow

```

```

disp( {VM1 CM1 PM1 VM2 CM2 PM2 VMA CMA PMA} )
disp(n1)
pause(2/1000)
end % while loop()
closepreview
close (figure(1))
fclose(B)
clear B
disp('Programa datos Bluetooth terminado...')
else
disp('Robot no seleccionado')
end%end if elegir nombre robot

```

```

%% funcion *****

```

```

function choice = chooseCamara(datos)
datos = {' ' datos{:}};
d = dialog('Position',[300 300 250 150],'Name','Camara');
txt = uicontrol('Parent',d,...
'Style','text',...

```

```

        'Position',[20 80 210 40],...
        'String','Seleccione Camara');

popup = uicontrol('Parent',d,...
    'Style','popup',...
    'Position',[55 70 135 25],...
    'String',datos,...
    'Callback',@popup_callback);

btn = uicontrol('Parent',d,...
    'Position',[89 20 70 25],...
    'String','Seleccionar',...
    'Callback','delete(gcf)');

choice = 0;

% Wait for d to close before running to completion
uiwait(d);

function popup_callback(popup,event)
    idx = popup.Value;
    popup_items = popup.String;
    %choice = char(popup_items(idx,:));
    choice = get(popup,'Value')-1;
end
end

```

#### Anexos 14: Foto galería



Participación en el robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)



Participación en el robochallenge obteniendo el 4to lugar en México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)



Equipo de Robótica UPS-GYE en el robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)



Equipo de Robótica UPS-GYE en el robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)