



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UN ROBOT PÉNDULO INVERTIDO
MEDIANTE UN CONTROLADOR PID**

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

AUTORES: Enrique Javier Salazar Chávez
Dahimon Naiche Angulo Aviles
TUTOR: Alberto Santiago Ramírez Farfán

Guayaquil - Ecuador
2024-2025

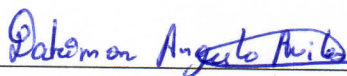
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Dahimon Naiche Angulo Aviles** con documento de identificación N° **0928909340** y **Enrique Javier Salazar Chávez** con documento de identificación N° **0957040462**; manifestamos que:

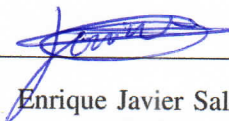
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 07 de febrero del año 2025

Atentamente,



Dahimon Naiche Angulo Aviles
0928909340



Enrique Javier Salazar Chavez
0957040462

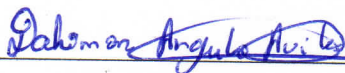
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Dahimon Naiche Angulo Aviles** con documento de identificación N° **0928909340** y **Enrique Javier Salazar Chávez** con documento de identificación N° **0957040462**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: DESARROLLO DE UN ROBOT PÉNDULO INVERTIDO MEDIANTE UN CONTROLADOR PID**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

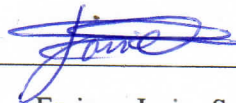
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 07 de febrero del año 2025

Atentamente,



Dahimon Naiche Angulo Aviles
0928909340



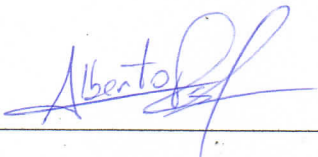
Enrique Javier Salazar Chávez
0957040462

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Alberto Santiago Ramírez Farfán**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN ROBOT PÉNDULO INVERTIDO MEDIANTE UN CONTROLADOR PID**, realizado por **Dahimon Naiche Angulo Aviles** con documento de identificación N° **0928909340** y por **Enrique Javier Salazar Chávez** con documento de identificación N° **0957040462**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 07 de febrero del año 2025

Atentamente,



Ing. Alberto Santiago Ramírez Farfán, Mg.
0923348890

DEDICATORIA

Dedico este proyecto a mi familia, especialmente a mis padres, Carlos Salazar y Mónica Chávez, cuyo amor incondicional, comprensión y apoyo constante me han brindado la fuerza para superar los desafíos. Ellos han estado a mi lado desde el inicio, esforzándose siempre para que sea una mejor persona y buscando que nunca deje de progresar. Gracias por ser mi mayor motivación, por su dedicación y por enseñarme el verdadero valor del esfuerzo y la perseverancia en cada etapa de mi vida.

A mi tutor, el Ingeniero Ramírez Farfán, por su invaluable orientación, paciencia y apoyo continuo. Su dedicación, experiencia y conocimientos han sido fundamentales para superar los desafíos que surgieron en este proyecto, brindándome las herramientas necesarias para aprender y desarrollarme tanto a nivel personal como profesional. Agradezco profundamente su compromiso en guiarme en cada paso, lo que ha sido clave en el éxito de este proceso.

Este proyecto es fruto de su constante apoyo y aliento, Gracias de todo corazón.

Enrique Salazar Chávez

DEDICATORIA

Dedico este proyecto a mis padres, por su amor incondicional, sacrificio y guía constante a lo largo de mi vida. Gracias por enseñarme el valor del esfuerzo, la perseverancia y la importancia de luchar por mis sueños. Su apoyo inquebrantable ha sido mi mayor fuente de motivación, impulsándome a seguir adelante incluso en los momentos más difíciles. Su ejemplo de dedicación y compromiso ha sido crucial para que pueda alcanzar mis metas, y por todo eso, siempre estaré agradecido.

A mi familia, por su cariño, apoyo constante y amor incondicional, que han sido mi refugio y fortaleza a lo largo de todo este proceso. Su presencia ha sido mi mayor impulso, brindándome la seguridad y confianza necesarias para enfrentar los desafíos y seguir adelante. Gracias por estar siempre a mi lado, por creer en mí y por enseñarme el verdadero valor de la unidad y el esfuerzo.

A mi abuelita, por sus sabios consejos y su amor infinito. Tus palabras, llenas de sabiduría, y tu ejemplo de vida han sido una luz que me ha orientado siempre, guiándome en los momentos de duda y fortaleciendo mi camino. Gracias por ser un pilar fundamental en mi vida.

Este logro es un reflejo de todo lo que me han dado. Gracias por estar siempre a mi lado.

Dahimon Angulo Aviles

AGRADECIMIENTO

Quiero agradecer a mi familia, especialmente a mis padres, Carlos Salazar y Mónica Chávez, por su constante apoyo, amor y comprensión. Han sido un pilar fundamental a lo largo de todo este proceso, guiándome en cada etapa con su sabiduría y paciencia. Gracias por siempre darme la fuerza para seguir adelante, por enseñarme a nunca rendirme y por estar a mi lado en cada desafío que he enfrentado.

A mi hermano, Carlos Andrés Salazar, y a mi hermana, Emely Salazar, quienes siempre me han brindado su apoyo y han estado presentes en los momentos más importantes. Su ayuda y consejos han sido cruciales para continuar creciendo y aprendiendo.

Quiero también agradecer al Ingeniero Ramírez Farfán, por su dedicación y por todo lo que me ha enseñado a lo largo de mi carrera, especialmente como miembro del club de robótica. Su orientación y enseñanzas han sido fundamentales no solo para este proyecto, sino también para el desarrollo de mis habilidades como futuro mecatrónico. Gracias a su experiencia, pude comprender de manera práctica y profunda algunos de los aspectos clave de la robótica.

Enrique Salazar Chávez

AGRADECIMIENTO

Al finalizar este trabajo de tesis, quiero expresar mi más profundo agradecimiento a todas las personas que hicieron posible su realización. En primer lugar, agradezco al ingeniero Farfán, cuya guía, paciencia y conocimientos fueron fundamentales en cada etapa de esta investigación. Su apoyo y consejos me ayudaron a superar los desafíos y a enriquecer este trabajo con una visión más amplia. A mi familia, por su amor incondicional, su apoyo emocional y su constante motivación. Su confianza en mí ha sido mi mayor impulso para seguir adelante, incluso en los momentos más difíciles.

Asimismo, agradezco a mis amigos y compañeros por su compañía, palabras de aliento y por compartir conmigo este viaje académico. Sus conversaciones y momentos de distracción fueron esenciales para mantener el equilibrio en este proceso. A mi empresa, por brindar los recursos y el espacio necesarios para desarrollar esta investigación. Finalmente, a todas aquellas personas que, de alguna manera, contribuyeron a la realización de esta tesis, mi más sincera gratitud.

Dahimon Angulo Aviles

RESUMEN

El presente proyecto se enfoca en el desarrollo de un robot péndulo invertido controlado mediante un sistema de control PID, con el propósito de participar en competencias robóticas y explorar nuevas categorías competitivas que involucren este tipo de robots. Su implementación brinda a los estudiantes la oportunidad de experimentar de manera práctica el diseño, ajuste y control de sistemas robóticos, fortaleciendo su comprensión de los principios de control automático. Para ello, se contempla la adaptación de un código PID preexistente, lo que permitirá ajustar sus parámetros en función de la dinámica del sistema y las restricciones del hardware utilizado.

Para el diseño y desarrollo del proyecto, se ha recurrido al uso de software CAD, lo que facilitó la integración de los diferentes componentes del sistema, tales como motores, sensores y microcontroladores. Durante el proceso de diseño, se priorizó una estructura equilibrada que asegurara un ensamblaje eficiente y un adecuado funcionamiento. Además, se seleccionó PLA para la impresión 3D de la estructura, debido a su resistencia y capacidad para soportar el peso de los componentes. En términos de programación, se adaptó un código PID preexistente para optimizar su integración con el hardware seleccionado.

La fase final del proyecto consistió en el ensamblaje de las piezas impresas y los componentes, verificando que contenga un diseño equilibrado. Asimismo, la estabilidad del sistema se logró mediante ajustes al código, realizados a través de pruebas iterativas que permitieron obtener los parámetros aproximados para mejorar el rendimiento. Este enfoque integral permitió superar varios desafíos técnicos, resultando en un prototipo robusto y funcional, listo para su implementación en entornos competitivos.

ABSTRACT

This project focuses on the development of an inverted pendulum robot controlled by a PID control system, with the purpose of participating in robotic competitions and exploring new competitive categories involving this type of robots. Its implementation gives students the opportunity to experiment in a practical way the design, adjustment and control of robotic systems, strengthening their understanding of the principles of automatic control. For this purpose, the adaptation of a pre-existing PID code is contemplated, which will allow adjusting its parameters according to the dynamics of the system and the restrictions of the hardware used.

For the design and development of the project, CAD software was used, which facilitated the integration of the different components of the system, such as motors, sensors and microcontrollers. During the design process, a balanced structure was prioritized to ensure efficient assembly and proper operation. In addition, PLA was selected for the 3D printing of the structure, due to its strength and ability to support the weight of the components. In terms of programming, a pre-existing PID code was adapted to optimize its integration with the selected hardware.

The final phase of the project consisted of the assembly of the printed parts and components, verifying that it contained a balanced design. In addition, system stability was achieved through adjustments to the code, performed through iterative testing to obtain the approximate parameters to improve performance. This integrated approach overcame several technical challenges, resulting in a robust and functional prototype, ready for implementation in competitive environments.

ÍNDICE

I.	INTRODUCCIÓN	1
II.	PROBLEMÁTICA	2
III.	JUSTIFICACIÓN	3
IV.	OBJETIVOS	4
IV-A.	Objetivo general	4
IV-B.	Objetivos específicos	4
V.	MARCO TEÓRICO	5
V-A.	Robótica	5
V-B.	Competencias robóticas	5
V-C.	Competencias robóticas en el Ecuador	6
V-D.	Péndulo	7
V-E.	Péndulo Invertido	8
V-F.	Control PID	9
V-F1.	Acción proporcional	11
V-F2.	Acción integral	11
V-F3.	Acción derivativa	11
V-F4.	Control PI	12
V-F5.	Control PD	12
V-G.	Ajustes de parametros de un PID	13
V-G1.	Metodo manual o prueba y error	13
V-G2.	Metodo Ziegler-Nichols	13
V-H.	Chasis	14
V-I.	Procesador	14
V-I1.	Arduino UNO	14
V-I2.	Arduino NANO	15
V-J.	TB6612FNG	16
V-K.	MPU6050	17
V-L.	Motor	18
V-M.	Batería de Lipo	19
VI.	MARCO METODOLÓGICO	20
VI-A.	Reglamentos de competencias nacionales	20
VI-B.	Diseño estructural mediante un software CAD	22
VI-B1.	Diseño de las bases	23
VI-B2.	Diseño de los componentes	24
VI-C.	Diseño electronico	28
VI-C1.	Prueba de componentes	28
VI-C2.	Diseño de placa o PCB	30
VI-D.	Integracion del Control Remoto	32
VI-D1.	Codigo	32
VI-D2.	Control Remoto	33
VI-E.	Algoritmo de trabajo	34

VII. RESULTADOS	37
VII-A. Transición del Diseño Conceptual a Modelos 3D para Impresión	37
VII-B. Prototipo 1	38
VII-B1. Ensamble	38
VII-B2. Verificación de componentes	40
VII-C. Prototipo 2	41
VII-C1. Ensamble	41
VII-C2. Verificación de componentes	42
VIII. CRONOGRAMA	44
IX. PRESUPUESTO	45
X. CONCLUSIONES	46
XI. RECOMENDACIONES	47
XII. ANEXOS	50

ÍNDICE DE FIGURAS

1.	Robot KUKA de la UPS sede cuenca [10]	5
2.	Runibot-Colombia Categorio minisumoRC [12]	6
3.	Batalla de robots categoría 120lb [15]	7
4.	Modelo de péndulo [16]	7
5.	Modelo de péndulo invertido [18]	8
6.	Robot balancín o péndulo invertido [21]	9
7.	Sistema de control básico obtenido de [25]	10
8.	Esquema de bloque PID obtenido de [25]	11
9.	Sumatoria acciones controlador PID obtenido de [24]	12
10.	Placa de arduino UNO obtenida de [30]	15
11.	Pines arduino NANO obtenida de [32]	16
12.	TB6612FNG de [33]	17
13.	MPU6050 [23]	18
14.	Motores con encoder de [34]	18
15.	Nomenclatura batería lipo obtenida de [15]	19
16.	Reglamento de Killtronic 6.0 para la categoria robot balancin	20
17.	Reglamento de la Utobot 6.0 para la categoria robot balancin	21
18.	Reglamento de la ESPE para la categoria robot balancin	22
19.	Base del robot diseñada en SolidWorks por autores	23
20.	Planos de la base elaborado por Autores	23
21.	Soporte para motores elaborado por Autores	24
22.	Plano de soporte elaborado por Autores	24
23.	Motor GA37-520 DC elaborado por Autores	25
24.	Planos de motor GA37-520 DC elaborado por Autores	25
25.	Llanta elaborado por Autores	26
26.	Dimensiones de la llanta	26
27.	Visualizacion del chasis ensamblado	27
28.	Dimensiones del robot ensamblado	27
29.	Conexion de pines MPU6050	28
30.	Conexion de pines del modulo BLUETOOTH	29
31.	Esquemático de la conexion de pines	30
32.	Esquemático mediante proteus8 diseñado por Autores	30
33.	Diagrama eléctrico proteus8 diseñado por Autores	31
34.	Visualizacion de la PCB realizado por Autores	31
35.	Variables declaradas en el código base	32
36.	Casos establecidos para las diferentes función a controlar	32
37.	Control remoto por defecto de la app	33
38.	Control desarrollado por Autores	33
39.	Algoritmo base del robot pendulo invertido	34
40.	Algoritmo desglosado del robot pendulo invertido	35
41.	Algoritmo incorporando el bluetooth	36
42.	Diseño de las bases para impresión 3D empleando UltimakerCura	37
43.	Base del robot pendulo invertido impresa por Autores	38
44.	Ensamble del prototipo 1 diseñado por Autores	39
45.	Ensamble del prototipo 1 diseñado por Autores	39
46.	Lectura del MPU6050 del prototipo 1	40
47.	Accionamiento de motores del prototipo 1	40

48.	Prueba de conduccion bluetooth	41
49.	Cableado de las conexiones del prototipo 2	41
50.	Prueba de sensores y actuadores de la PCB diseñada por Autores	42
51.	Prueba de sensores y actuadores de la PCB diseñada por Autores	42
52.	Prueba de motores del prototipo 2 diseñada por Autores	43
53.	Parámetros del código PID	50
54.	Código del control PID	51
55.	Parámetros de velocidad incorporados al prototipo	51
56.	Pines del arduino involucrados en el prototipo	52
57.	Prototipo 1	53
58.	Prototipo 2	53
59.	Soldado de PCB del prototipo 2	54
60.	Ensamblado del prototipo 2	54
61.	Prueba final del prototipo 2	55
62.	Robots balancines	56

ÍNDICE DE TABLAS

I.	Valores Kp Ti Td de Ziegler-Nichols en lazo cerrado	14
II.	Valores Kp Ti Td de Ziegler-Nichols en lazo abierto	14
III.	Arduino UNO	15
IV.	Arduino NANO	16
V.	TB6612FNG	17
VI.	MPU6050	17
VII.	Conexion MPU6050	28
VIII.	Conexion modulo BLUETOOTH	29
IX.	Conexion tb6612fng a motor y arduino	29
X.	Cronograma	44
XI.	Presupuesto	45

I. INTRODUCCIÓN

Este proyecto consiste en la implementación de un robot péndulo invertido utilizando un controlador PID preexistente, al cual se le ajustarán los parámetros de acuerdo con las características del hardware establecido, permitiendo explorar nuevas categorías competitivas, proporcionando a los estudiantes una experiencia práctica en el diseño, control y calibración de sistemas robóticos.

El club de Robótica de la Universidad Politécnica Salesiana busca habilitar nuevas categorías y, en este contexto, se ha optado por los robots de péndulo invertido debido a su capacidad para integrar conocimientos en áreas clave como el diseño, la electrónica y la mecánica. Este enfoque permite a los estudiantes aplicar estos conceptos en la práctica, mientras desarrollan habilidades técnicas en control de sistemas dinámicos, programación y modelado 3D, fundamentales para el diseño y mejoramiento de sistemas robóticos avanzados.

El controlador PID es ampliamente utilizado en la industria, tales como los sistemas de control dinámicos que requieran de estabilidad. En este proyecto, se utiliza un controlador PID preexistente, implementando el método empírico de prueba y error para ajustar el rendimiento y garantizar la estabilidad del sistema. Aunque existen métodos alternativos como Ziegler-Nichols o ajustes en lazo abierto, se opta por prueba y error debido a su conveniencia y rendimiento en este escenario específico, permitiendo ajustes directos y prácticos. Aunque todos estos métodos comparten el mismo principio de sintonización de parámetros en tiempo real, los enfoques alternativos se enfocan en la obtención de una planta, mientras que el de prueba y error se centra en la adaptación empírica de los parámetros para aproximarse a los valores reales del sistema.

La finalidad de este proyecto se enfoca en dos puntos principales: el primero es desarrollar un robot balancín que pueda participar en futuras competencias, permitiendo a los estudiantes fortalecer sus habilidades técnicas. El segundo es contribuir con documentación técnica, como planos y códigos de programación, para facilitar futuras mejoras o réplicas del proyecto, ya sea con fines educativos o competitivos.

II. PROBLEMÁTICA

La creciente popularidad de las competencias de robótica ha resaltado la necesidad de que el club de robótica de la Universidad Politécnica Salesiana, sede Guayaquil, incremente su rendimiento para mejorar en los rankings. Actualmente, el posicionamiento en estas competencias depende de la participación en múltiples categorías, lo cual exige una diversificación de prototipos. Aunque en la institución se han desarrollado robots balancín o péndulos invertidos con fines educativos, no se han creado prototipos específicamente orientados a la competencia, lo que limita las posibilidades de obtener un buen lugar en los eventos de robótica [1].

Hoy en día, los estudiantes se enfrentan a desafíos complejos que requieren integrar conocimientos de diversas áreas como Ingeniería, Matemáticas, Física y Ciencias de la Computación. La robótica se presenta como una excelente opción para combinar estos campos, y el desarrollo de un sistema de péndulo invertido es uno de los mejores ejemplos. Este proyecto involucra varias disciplinas, como control teórico, programación, matemáticas y diseño CAD. Sin embargo, la enseñanza teórica de estos temas se ve limitada por la falta de tiempo y la complejidad en los análisis prácticos, lo que genera dificultades para crear prototipos que puedan procesar datos en tiempo real y aplicar respuestas correctivas de manera efectiva [2].

El control PID, aunque adecuado para experimentos de control básicos, presenta limitaciones en investigaciones de mayor complejidad. Su simplicidad dificulta la adaptación ante variaciones en los parámetros del sistema, lo cual compromete la escalabilidad y precisión necesarias en sistemas complejos. Esta limitación hace difícil prever el comportamiento del sistema con exactitud, lo que subraya la necesidad de modelos matemáticos avanzados para lograr un control robusto bajo condiciones variables [3].

En la Universidad Politécnica Salesiana, los estudiantes se enfrentan a grandes retos al construir prototipos robóticos complejos, como el robot de péndulo invertido. La falta de experiencia en la construcción e integración de componentes para sistemas dinámicos afecta la funcionalidad de los prototipos. Esta limitación en la implementación de sistemas de control efectivos evidencia la necesidad de proyectos que permitan un aprendizaje práctico, orientado a fortalecer las habilidades necesarias para abordar los desafíos técnicos de la robótica competitiva [4].

III. JUSTIFICACIÓN

Los proyectos que integran diversas disciplinas, como el desarrollo de un péndulo invertido controlado, ofrecen una experiencia única para combinar conocimientos técnicos y científicos. Al explorar tanto los fenómenos físicos como los aspectos de programación y matemáticas, los estudiantes desarrollan habilidades que fomentan su creatividad y capacidad de adaptación. Este enfoque integral les permite enfrentar problemas complejos con una perspectiva amplia, preparándolos para aplicar soluciones innovadoras en el campo de la robótica y el control [5].

La implementación de un robot de péndulo invertido con control PID subraya los retos de estabilizar un sistema inestable, lo que demanda precisión en la coordinación de sensores y algoritmos de control. Estos proyectos brindan a los estudiantes una experiencia invaluable en el aprendizaje de métodos de control avanzados, fortaleciendo su habilidad para resolver desafíos reales de estabilidad y control en sistemas dinámicos.

El algoritmo PID es una opción adecuada para el control de un péndulo invertido debido a su simplicidad y efectividad para estabilizar el sistema mediante prueba y error, sin la necesidad de un modelo dinámico complejo. Esta característica lo convierte en una herramienta útil para la enseñanza práctica de control, ya que permite a los estudiantes experimentar directamente con los parámetros del controlador PID y observar cómo afectan el comportamiento del sistema. Aunque adecuado para situaciones básicas, para lograr un control más robusto y preciso, es necesario emplear un modelo matemático detallado, como el de dos grados de libertad, basado en los principios de la mecánica de Newton y Lagrange [6].

El desarrollo de un prototipo de robot de péndulo invertido en concursos de robótica permite a los estudiantes enfrentar retos de ingeniería, combinando control automático, electrónica y programación. Contar con una guía para su creación les proporciona un esquema claro para diseñar y optimizar el sistema, integrar componentes y seleccionar sensores y actuadores adecuados. Este enfoque no solo facilita la comprensión de conceptos complejos, sino que también prepara a los estudiantes para desarrollar soluciones innovadoras y efectivas en robótica [4].

El creciente interés de la juventud en la robótica competitiva destaca la necesidad de una formación práctica sólida. Aunque los simuladores pueden ofrecer experiencias interactivas, la experiencia real es insustituible. Conocer los diferentes componentes a implementar permite a los estudiantes adaptarse mejor durante las competencias. La construcción de prototipos proporciona un entendimiento profundo de los conceptos y desafíos técnicos, facilitando el aprendizaje de habilidades clave y preparando a los estudiantes para enfrentar los retos específicos que surgen en eventos competitivos.

IV. OBJETIVOS

IV-A. Objetivo general

Desarrollar un prototipo de robot péndulo invertido para fines educativos y competitivos, permitiendo la demostración práctica de los principios de control de sistemas dinámicos utilizando un control PID.

IV-B. Objetivos específicos

1. Desarrollar la estructura del robot de péndulo invertido, integrando sensores de posición y actuadores para el control preciso de su movimiento.
2. Construir la base del péndulo invertido, eligiendo materiales que equilibren costo y durabilidad, proporcionando un soporte estable y adecuado para las pruebas de control.
3. Validar un controlador PID que permita una maniobrabilidad precisa y estable.

V. MARCO TEÓRICO

V-A. Robótica

De acuerdo con [7], la robótica es la rama de la tecnología enfocada en el diseño, construcción, operación y control de robots. Este campo multidisciplinario combina conocimientos de mecánica, electrónica y programación, y ha experimentado un crecimiento acelerado, ampliando su alcance desde aplicaciones comerciales hasta áreas militares. Su desarrollo continúa impulsando avances significativos en sectores como la medicina, la industria y la investigación científica.

El crecimiento constante en las capacidades de los robots ha presentado un desafío fundamental para las industrias: determinar la mejor manera de aprovechar sus aplicaciones. A medida que se expande la gama de funciones que pueden desempeñar, también ha crecido el interés de las personas en los sistemas embebidos y la robótica. Este fenómeno ha captado la atención de individuos de todas las edades, desde niños hasta adultos, quienes demuestran un entusiasmo creciente por aprender y profundizar en esta tecnología [8].

En Ecuador, la robótica se ha consolidado en el ámbito educativo como una herramienta clave para motivar a estudiantes ha desarrollar habilidades en áreas multidisciplinarias. Un ejemplo de este enfoque es el Concurso Ecuatoriano de Robótica (CER), que promueve el avance tecnológico convocando cada año a instituciones de educación superior para mostrar sus habilidades en competencias nacionales e internacionales [9].



Figura 1. Robot KUKA de la UPS sede cuenca [10]

V-B. Competencias robóticas

Una competencia de robótica, según lo descrito por [11], se entiende como un evento en el que participan diversos prototipos robóticos, los cuales pueden ser diseñados o adquiridos, dependiendo de las normas establecidas por los organizadores. La configuración física y la programación de cada robot varían dependiendo de la categoría en la que compiten, lo que permite a los equipos ajustar sus diseños y estrategias para satisfacer los requisitos específicos del evento. Este tipo de competencia suele poner a prueba no solo la destreza técnica, sino también la capacidad de los participantes para innovar y adaptarse a situaciones imprevistas.

Dado que las competencias de robótica abarcan una variedad de categorías, los participantes deben ajustar tanto el diseño físico como la programación de sus robots para cumplir con los requisitos específicos de cada evento. Estos ajustes son fundamentales para garantizar que los prototipos sean competitivos en cada desafío planteado. Además, los equipos deben ser capaces de modificar y optimizar sus soluciones durante el transcurso del evento, enfrentándose a fallos de hardware o imprevistos en la programación, lo que aumenta la presión para tomar decisiones rápidas y efectivas.



Figura 2. Runibot-Colombia Categoria minisumoRC [12]

V-C. Competencias robóticas en el Ecuador

La robótica competitiva ha cobrado gran relevancia en Ecuador, evidenciada por la celebración de múltiples competencias en el país. Entre las más destacadas se encuentran UMEBOT e INTERBOT (organizadas por la Escuela Politécnica Nacional), KILLTRONIC de la Universidad Politécnica Salesiana, y RIOTRONIC de la Escuela Superior Politécnica del Chimborazo. Sin embargo, el evento más importante sigue siendo el Concurso Ecuatoriano de Robótica (CER), el cual es el evento competitivo de robótica más importante del país. Este tipo de competencias establece reglamentos específicos para cada categoría, como seguidor de línea, minisumo, batalla, balancín, entre otros. Todas estas competencias se dedican a evaluar las habilidades de los participantes en diversas categorías, fomentando la innovación y el desarrollo de destrezas técnicas y colaborativas [1].

Cada categoría tiene requisitos particulares que los participantes deben cumplir, lo que permite a los equipos ajustar sus diseños y estrategias para optimizar el desempeño de sus robots según el desafío que se les presente. La configuración física y la programación de cada robot cambian de acuerdo con la categoría en la que compiten, permitiendo a los equipos adaptarse y mejorar continuamente. Así, el CER y otros eventos similares contribuyen a consolidar el crecimiento de la robótica en Ecuador, siendo espacios clave para el aprendizaje y la creatividad en este campo [13].

El robot autoequilibrista, o péndulo invertido, ha generado un notable interés en los últimos años por su facilidad de construcción y las características de su sistema físico. Estos robots son ideales para estudios en robótica por los principios que este involucra y, por ello, se consideran una categoría digna de evaluación

en diferentes competencias a nivel mundial [14] .

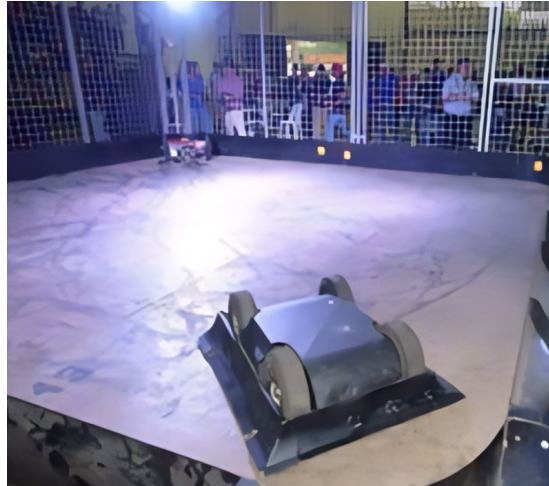


Figura 3. Batalla de robots categoría 120lb [15]

V-D. Péndulo

El péndulo simple es un sistema oscilatorio compuesto por una masa colgada de un punto fijo, que se desplaza en una trayectoria curva debido a la acción de la gravedad. La fuerza que restaura el movimiento del péndulo es proporcional al seno del ángulo de su desplazamiento, lo que genera un movimiento periódico y oscilante. Este comportamiento está regido por principios de dinámica angular y, en su forma ideal, donde se desprecia la fricción, el péndulo presenta un período de oscilación constante. Dicho período depende exclusivamente de la longitud del hilo o varilla que sostiene la masa y de la aceleración gravitatoria [16] .

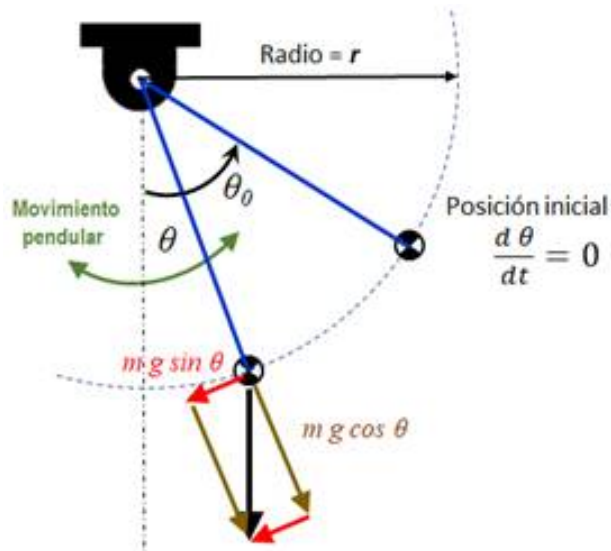


Figura 4. Modelo de péndulo [16]

En el contexto de la enseñanza de sistemas dinámicos, el péndulo simple se utiliza para ilustrar conceptos de control y estabilidad, especialmente en sistemas oscilatorios. A pesar de que el péndulo es lineal para pequeños ángulos, se puede extender a sistemas no lineales cuando el ángulo es grande, lo que permite experimentar con la aplicación de controladores para estabilizar el sistema o modificar su comportamiento. A través de la implementación de controladores como el PID [17].

V-E. Péndulo Invertido

El péndulo invertido es un sistema dinámico ampliamente utilizado en la investigación debido a su inestabilidad inherente. Su configuración básica consiste en una barra con una masa en la parte superior y un pivote en la inferior, lo que permite que la masa se mueva en el plano horizontal. El principal reto es mantener la barra en posición vertical, ya que incluso pequeñas perturbaciones pueden desestabilizar el sistema. Esto convierte al control del péndulo invertido en un desafío constante, especialmente en situaciones que requieren equilibrio estable [18].

Para abordar este desafío, se emplean métodos como el control PID. El PID ajusta continuamente la entrada de control según los errores de posición, velocidad y aceleración del sistema, permitiendo correcciones dinámicas. La implementación adecuada de este control requiere calibrar correctamente sus parámetros para mantener la estabilidad del sistema [19].

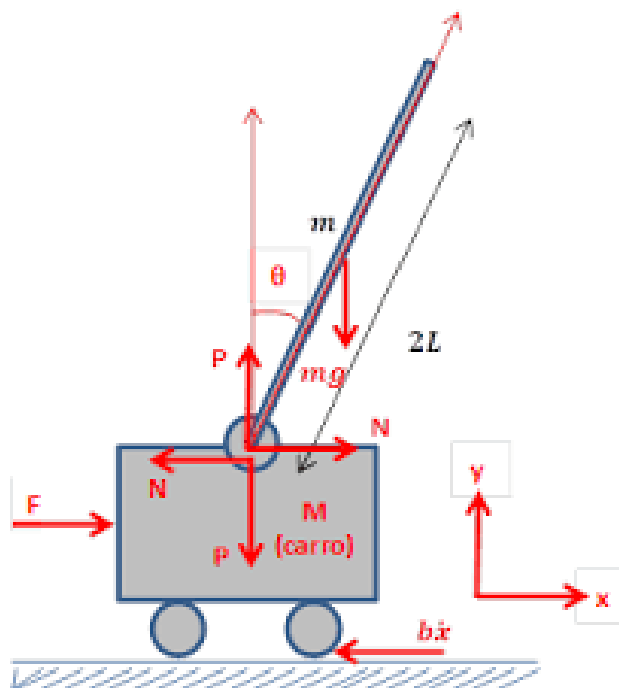


Figura 5. Modelo de péndulo invertido [18]

En los robots autoequilibristas, una versión móvil del péndulo invertido, la dificultad aumenta, ya que se debe mantener el equilibrio no solo en reposo, sino también mientras el robot se desplaza. Estos robots son utilizados para enseñar control automático y sus investigaciones incluyen el uso de sensores como

acelerómetros y giroscopios para monitorear la posición y mejorar la precisión de los controles [20].

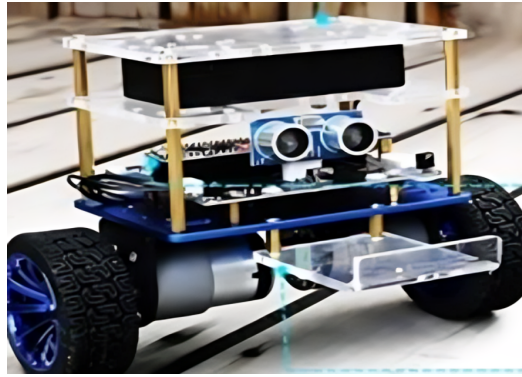


Figura 6. Robot balancín o péndulo invertido [21]

V-F. Control PID

De acuerdo con Jorge Tutiven [22], el control se define como las técnicas matemáticas que procesan variables realimentadas dentro de algoritmos y esta capacidad permite a los robots interactuar de manera más efectiva con su entorno, facilitando la realización de tareas más eficaces y complejas. Existen diversos métodos de control, como las redes neuronales y el control PID, que fortalecen esta interacción y amplían la capacidad de los robots para llevar a cabo funciones avanzadas.

El control PID (proporcional-integral-derivativo) es un método de control utilizado en diversas áreas de la industria, especialmente por su bajo costo de implementación, la facilidad de calibración y su rápida respuesta ante cambios en la señal de referencia o setpoint. Este método es uno de los más comunes en la regulación automática gracias a su flexibilidad [23].

El PID es un algoritmo clave en el control automático que se compone de tres parámetros: proporcional, derivativo e integral. El componente proporcional se basa en el error entre el valor medido y el setpoint, mientras que el integral se ocupa de la acumulación de errores pasados, eliminando el error en estado estacionario. Por su parte, el derivativo predice valores futuros, mejorando así la respuesta del sistema [24].

$$u(t) = P(t) + I(t) + D(t) \quad (1)$$

Donde:

P(t): Acción proporcional

I(t):Acción integral

D(t):Acción derivativa

Cada acción tiene su fórmula correspondiente:

$$P(t) = kp * e(t) \quad (2)$$

$$I(t) = ki * \int_0^t e(t) dt \quad (3)$$

$$D(t) = kd * \frac{d}{dt} e(t) \quad (4)$$

Reemplazamos la ecuación 2,3,4 en la ecuación 1 para obtener la fórmula general del PID:

El funcionamiento del algoritmo PID se desarrolla de la siguiente manera:

$$u(t) = kp * e(t) + ki * \int_0^1 e(t) dt + kd * \frac{d}{dt}e(t) \quad (5)$$

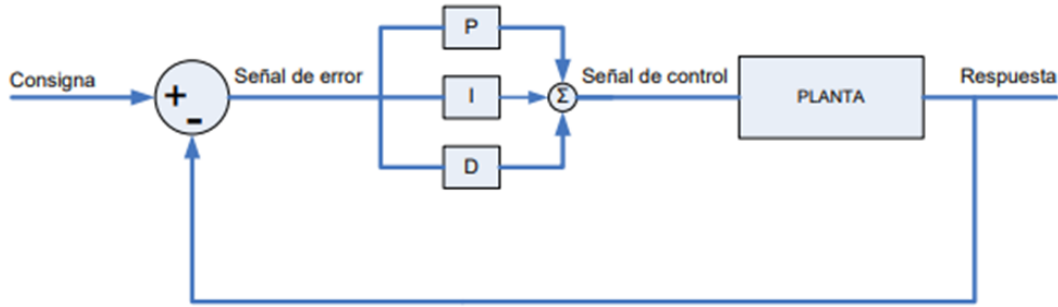


Figura 7. Sistema de control básico obtenido de [25]

Según [25], la ecuación (5) se utiliza en el ámbito académico debido a su simplicidad, pero carece de sentido en la práctica. Él menciona el caso de un controlador únicamente derivativo, donde solo K_d diferente de 0, aquí prácticamente es imposible hallar la estabilidad, Por esta razón, destaca la importancia de incluir siempre la acción proporcional, ya que esta proporciona una respuesta de control básica y directa ante el error.

Dado esto existe un formato estandar o ISA definida en la ecuación (6).

$$u(t) = kp * e(t) + \frac{kp}{Ti} * \int_0^1 e(t) dt + kp * Td * \frac{d}{dt}e(t) \quad (6)$$

La ecuación (6) se la puede abreviar de tal forma que quede de la siguiente manera:

$$u(t) = kp[e(t) + \frac{1}{Ti} * \int_0^1 e(t) dt + Td * \frac{d}{dt}e(t)] \quad (7)$$

Donde:

K_p : ganancia proporcional

T_i : tiempo integral

T_d : tiempo derivativo

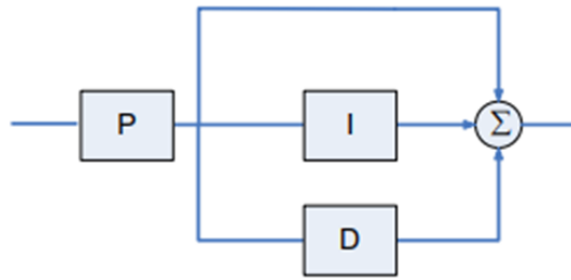


Figura 8. Esquema de bloque PID obtenido de [25]

V-F1. Acción proporcional:

En la acción proporcional, la ganancia K_p se ajusta con el propósito de reducir el error, lo que acelera la respuesta del sistema. Un valor adecuado de K_p permite que el sistema alcance el estado deseado en menos tiempo, mejorando su desempeño. Sin embargo, si este valor es demasiado alto, pueden generarse oscilaciones o incluso inestabilidad, por lo que es crucial encontrar un equilibrio en su ajuste [24].

Después de modificar K_p es fundamental evaluar el comportamiento del sistema para verificar su estabilidad. Un incremento excesivo puede hacer que el sistema responda de manera agresiva, mientras que un valor demasiado bajo podría hacer que la corrección del error sea lenta e ineficiente [26], la fórmula de la acción proporcional está dada en la ecuación (2).

V-F2. Acción integral:

La acción integral tiene como objetivo eliminar el error en estado estacionario acumulando la desviación a lo largo del tiempo y aplicando una corrección proporcional a esta acumulación. Sin embargo, un ajuste inadecuado de K_i puede generar oscilaciones excesivas o inestabilidad, por lo que es fundamental realizar una calibración precisa [26].

Para garantizar un desempeño adecuado, el ajuste de K_i debe realizarse con cuidado, equilibrando la eliminación del error en estado estacionario con la estabilidad del sistema. Si la ganancia integral es demasiado baja, el sistema tardará más en corregir el error, mientras que un valor demasiado alto puede generar respuestas inestables. Por ello, la acción integral suele combinarse con las acciones proporcional y derivativa [24].

V-F3. Acción derivativa:

La acción derivativa se ajusta cuando el sistema aún presenta inestabilidad, aplicando modificaciones graduales a la ganancia K_d para lograr una respuesta más estable. Este control actúa sobre la velocidad de cambio del error, permitiendo anticipar variaciones bruscas y aplicando correcciones que amortiguan dichas fluctuaciones. De este modo, se reduce la posibilidad de oscilaciones no deseadas y se mejora la estabilidad general del sistema [24].

Al calcular la derivada del error, la acción derivativa permite prever la tendencia de crecimiento o disminución de la señal de error, lo que facilita una respuesta más rápida y controlada. Sin embargo, un ajuste inadecuado de K_d puede amplificar el ruido en la señal de error, por lo que es esencial calibrarlo

con precisión para evitar respuestas inestables. [26].

Cuando se llevan estos sistemas al plano real, suelen aparecer limitantes que afectan la capacidad del controlador para alcanzar la respuesta deseada. en un ejemplo citado por Carmen Sanchez [24] nos dice, en un sistema de control de temperatura con una resistencia de 2000 vatios, si el controlador intenta acelerar el calentamiento incrementando la potencia, no logrará superar ese límite, lo cual restringe la velocidad de respuesta .

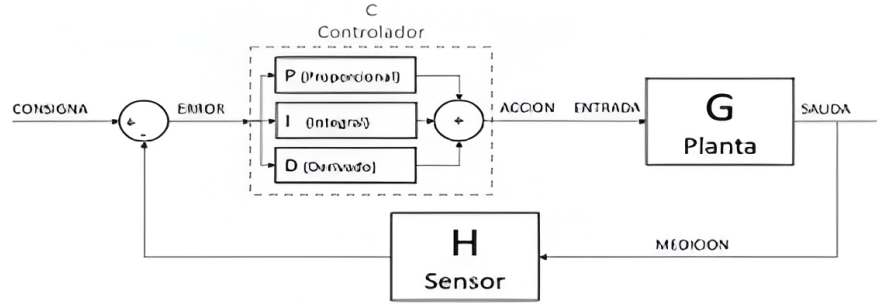


Figura 9. Sumatoria acciones controlador PID obtenido de [24]

V-F4. Control PI:

En un sistema de control PI, se busca reducir el error en estado estacionario y mejorar la respuesta del sistema. El componente proporcional corrige el error de forma inmediata, mientras que el integral acumula el error a lo largo del tiempo, compensando desviaciones persistentes. Para automatizar el ajuste, se introduce un término que suma una cantidad proporcional al área entre el tiempo y el error, permitiendo un reset. automático. Así, el valor de la desviación se ajusta dinámicamente, asegurando una corrección continua y precisa del sistema [27].

$$u(t) = kp * e(t) + \frac{kp}{Ti} * \int_0^t e(t) dt \quad (8)$$

V-F5. Control PD:

Un controlador PD (Proporcional-Derivativo) combina una acción proporcional, que responde al error actual, con una acción derivativa, que anticipa cambios futuros basándose en la tasa de variación del error. Esto permite una corrección rápida y reduce oscilaciones, mejorando la estabilidad y respuesta del sistema. Es útil en aplicaciones donde se requiere un ajuste rápido y preciso, como en sistemas de posicionamiento o control de velocidad [27].

$$u(t) = kp * e(t) + kp * Td * \frac{d}{dt}e(t) \quad (9)$$

V-G. Ajustes de parametros de un PID

El ajuste adecuado de los parámetros del controlador PID es fundamental para garantizar que el bucle de control corrija los errores de manera eficiente y rápida, permitiendo que el sistema alcance el estado estacionario con estabilidad, obteniendo el setpoint deseado.

V-G1. Metodo manual o prueba y error:

Este método consiste en fijar inicialmente los valores de K_i y K_d en 0, y ajustar únicamente el parámetro K_p hasta que se observe una oscilación en la respuesta del sistema. Una vez alcanzada esta condición, se comienza a modificar K_p de manera gradual para acercarse lo más posible al valor óptimo, buscando una respuesta con mínima oscilación [28].

Posteriormente, se ajusta el valor de K_i con el fin de reducir la magnitud de la oscilación y mejorar la estabilidad del sistema. Si es necesario, se introduce un ajuste de K_d , para atenuar posibles picos en la respuesta transitoria. Este proceso de ajuste se repite de manera iterativa hasta que se logre la estabilidad deseada en el sistema, optimizando el desempeño del controlador PID [28].

V-G2. Metodo Ziegler-Nichols:

El método de Ziegler-Nichols es una técnica práctica utilizada para ajustar los parámetros de un controlador PID sin necesidad de contar con un modelo matemático de la planta. Consiste en ajustar los valores de K_p , K_i , K_d generando una respuesta escalon, mientras se observa la respuesta transitoria del sistema, analizando aspectos como el sobreimpulso y el tiempo de establecimiento para mejorar el rendimiento y minimizar las oscilaciones [29].

Cuando se tiene acceso a un modelo matemático de la planta, se pueden emplear métodos analíticos para establecer los parámetros del controlador que cumplan con los requisitos de rendimiento. No obstante, en sistemas más complejos donde obtener dicho modelo es complicado, el enfoque analítico resulta impráctico, por lo que se opta por técnicas experimentales como Ziegler-Nichols para realizar la sintonización del controlador PID [27].

El método de Ziegler-Nichols se puede aplicar de dos formas: en lazo abierto o en lazo cerrado. Ambos enfoques utilizan un sobreimpulso del 25 % en la respuesta escalón para ajustar los parámetros del controlador.

En el método de lazo abierto, se obtiene de manera experimental la respuesta de la planta ante una entrada en escalón unitario. Cuando la planta no contiene integradores ni polos complejos, la curva resultante presenta una forma característica de "S". Sin embargo, si estas condiciones no se cumplen, el método no puede ser aplicado y según Ziegler-Nichols se sugiere usar las formulas en la tabla 1 [27].

Tabla I
VALORES KP TI TD DE ZIEGLER-NICHOLS EN LAZO CERRADO

Tipo de controlador	Kp	Ti	Td
P	T/L	∞	0
PI	0.9T/L	L/0.3	0
PID	1.2T/L	2L	0.5L

En base a esa tabla se reemplazarían los valores en la ecuación (7), quedando con la ecuación (10).

$$u(t) = 1,2 \frac{T}{L} [e(t) + \frac{1}{2L} * \int_0^1 e(t) dt + 0,5L * \frac{d}{dt} e(t)] \quad (10)$$

En el método de lazo cerrado, se configuran los valores de Ti y Td como 0 utilizando únicamente la acción de control proporcional. Luego, se incrementa gradualmente la ganancia Kp hasta alcanzar un valor crítico de Kcr en el cual la salida del sistema comienza a presentar oscilaciones sostenidas. Si no se logran observar estas oscilaciones al variar Kp el método no puede ser aplicado y según Ziegler-Nichols para este método se sugiere usar las fórmulas en la tabla 2 [27].

Tabla II
VALORES KP TI TD DE ZIEGLER-NICHOLS EN LAZO ABIERTO

Tipo de controlador	Kp	Ti	Td
P	0.5Kcr	∞	0
PI	0.45Kcr	1Pcr/1.2	0
PID	0.6Kcr	0.5Pcr	0.125Pcr

En base a esa tabla se reemplazarían los valores en la ecuación (7), quedando con la ecuación (11).

$$u(t) = 0,6Kcr [e(t) + \frac{1}{0,5Pcr} * \int_0^1 e(t) dt + 0,125Pcr * \frac{d}{dt} e(t)] \quad (11)$$

V-H. Chasis

El diseño del chasis es clave para el éxito del robot balancín, ya que impacta en la precisión de sus movimientos y el equilibrio. Además, la distribución adecuada de la electrónica y las masas es crucial para no comprometer su rendimiento [14].

V-I. Procesador

V-II. Arduino UNO: Arduino UNO es una placa microcontroladora creada en el Instituto de Diseño Interactivo en Ivrea, Italia. Está equipada con un microcontrolador ATmega, que permite ejecutar programas y controlar dispositivos. Además, cuenta con pines digitales y analógicos que facilitan la comunicación con una variedad de módulos, como sensores y actuadores, todo esto la hace ideal para proyectos de robótica [30].

Tabla III
ARDUINO UNO

ESPECIFICACIONES	
Microcontrolador	ATmega328
Voltaje operacional	5V
Voltaje de entrada	7V a 12V
Pines digitales I/O	14(6 de PWM)
Pines analogicos	8
Corriente DC para PINES I/O	40mA
Flash memory	32KB(ATmega326)
SRAM	2(ATmega326)
Clock speed	16MHz
Dimensiones	68.6mm x 53.3mm



Figura 10. Placa de arduino UNO obtenida de [30]

V-I2. Arduino NANO: El Arduino Nano es una versión compacta del Arduino Uno, equipada con los microcontroladores ATmega328 (versión 3.0) o ATmega168 (versión 2.x). Aunque conserva las mismas funcionalidades, su diseño reducido prescinde de un conector de alimentación dedicado y emplea un cable USB Mini-B para alimentación y comunicación. Creado originalmente por Gravitech, es una opción ideal para proyectos que requieren un microcontrolador potente en un espacio limitado [31].

A continuación tenemos la tabla de las especificaciones del arduino nano obtenida de [31].

Tabla IV
ARDUINO NANO

ESPECIFICACIONES	
Microcontrolador	Atmel ATmega168 or ATmega328
Voltaje operacional	5V
Voltaje de entrada	7V a 12V
Pines digitales I/O	14(6 de PWM)
Pines analogicos	8
Corriente DC para PINES I/O	40mA
Flash memory	16KB(ATmega128) o 32KB(ATmega326)
SRAM	1(ATmega128) o 2(ATmega326)
Clock speed	16MHz
Dimensiones	19mm x 44mm

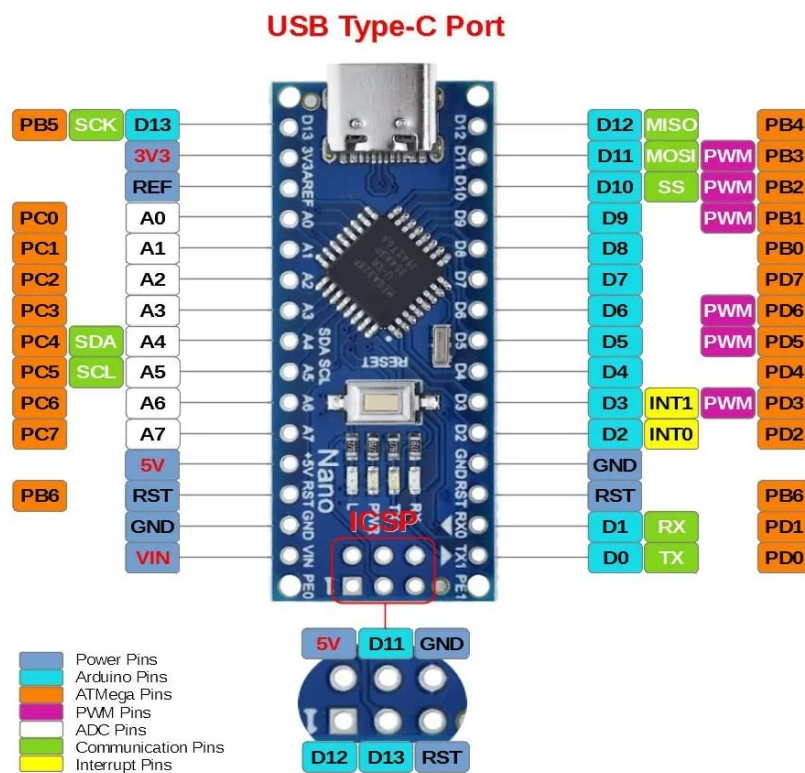


Figura 11. Pines arduino NANO obtenida de [32]

V-J. TB6612FNG

El TB6612FNG es un driver creado para controlar motores a través de señales TTL provenientes de microcontroladores como Arduino o Raspberry Pi. Este driver permite gestionar hasta dos motores DC con una corriente máxima de 1.2A por motor. Cada motor se controla mediante una señal PWM de hasta 100 kHz, proporcionando un control preciso y eficiente en aplicaciones que requieren variación de velocidad y dirección [33].

Tabla V
TB6612FNG

ESPECIFICACIONES	
VCC	5V
VM(voltaje de motor)	2.5V a 13.5V
GND	GND
AO	pins de salida motor 1
BO	pins de salida motor 2
Corriente DC para PINES I/O	40mA
AN	pins de entrada motor 1
BN	pins de entrada motor 2
PWMA	pwm de motor 1
PWMB	pwm de motor 2
Dimensiones	21mm x 18.5mm

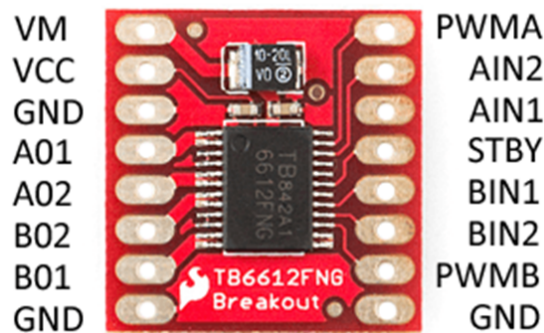


Figura 12. TB6612FNG de [33]

V-K. MPU6050

EL módulo MPU6050 es un sistema el cual esta conformado por diferentes sensores como uno de temperatura, un giroscopio y un acelerómetro, estos dos últimos sensores son los encargados de dar un posicionamiento lineal y angular en el módulo, el funcionamiento de estos sensores se da porque estan conformados por parte mecánica y pizcoelétrica, cuando el módulo se mueve , el sensor emite una señal que es recibida por el microcontrolador [23].

Tabla VI
MPU6050

Modelo	GY-521
Procesador	MPU-6050
VCC	3.3V a 5V
Rango giroscopio	± 250 500 1000 2000 $^{\circ} / s$
Rango aceleracion	± 2 ± 4 ± 8 ± 0.56 oz
Modo de comunicacion	Protocolo I2C
Dimensiones	20mm x 15mm

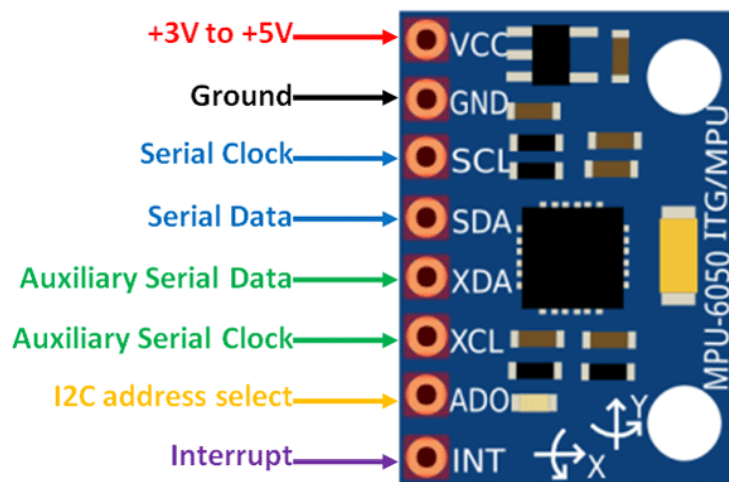


Figura 13. MPU6050 [23]

V-L. Motor

Según [34], los motores son componentes esenciales en la construcción de cualquier prototipo robótico o mecánico. Las especificaciones de los motores, como el voltaje y las RPM, deben seleccionarse en función de los requisitos específicos del prototipo. Los motores reductores, por ejemplo, se identifican con relaciones como 10:1 o 5:1, que indican la reducción en RPM y adaptan el motor a distintas necesidades de velocidad y torque, por eso la necesidad de conocer todas las carteristicas del motor que se va a emplear.

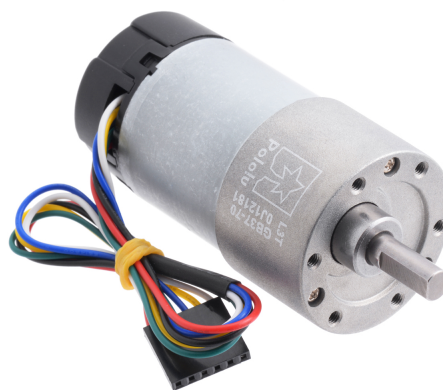


Figura 14. Motores con encoder de [34]

V-M. Batería de Lipo

Las baterías de litio-polímero (LiPo) son baterías recargables ampliamente utilizadas en competencias robóticas, destacándose por su alta densidad energética, elevada tasa de descarga y bajo peso. Cada batería LiPo presenta una nomenclatura específica que permite identificar sus características de diseño [15].

La nomenclatura de las baterías LiPo indica tres aspectos esenciales: el número de celdas (voltaje), la capacidad en mAh y la tasa de descarga (C). Estos factores ayudan a elegir una batería que ofrezca el rendimiento adecuado, permitiendo un equilibrio entre potencia y duración según las necesidades del proyecto.



Figura 15. Nomenclatura batería lipo obtenida de [15]

VI. MARCO METODOLÓGICO

Este apartado detalla el proceso completo de desarrollo del robot péndulo invertido, abarcando todas sus fases.

VI-A. Reglamentos de competencias nacionales

El robot ha sido diseñado para cumplir con los lineamientos establecidos por los reglamentos de diversas competencias a nivel nacional, lo que le permite competir en la mayoría de las competencias dentro de esta categoría. Estos lineamientos se obtienen a partir del análisis detallado de los requisitos y normas que rigen cada evento.

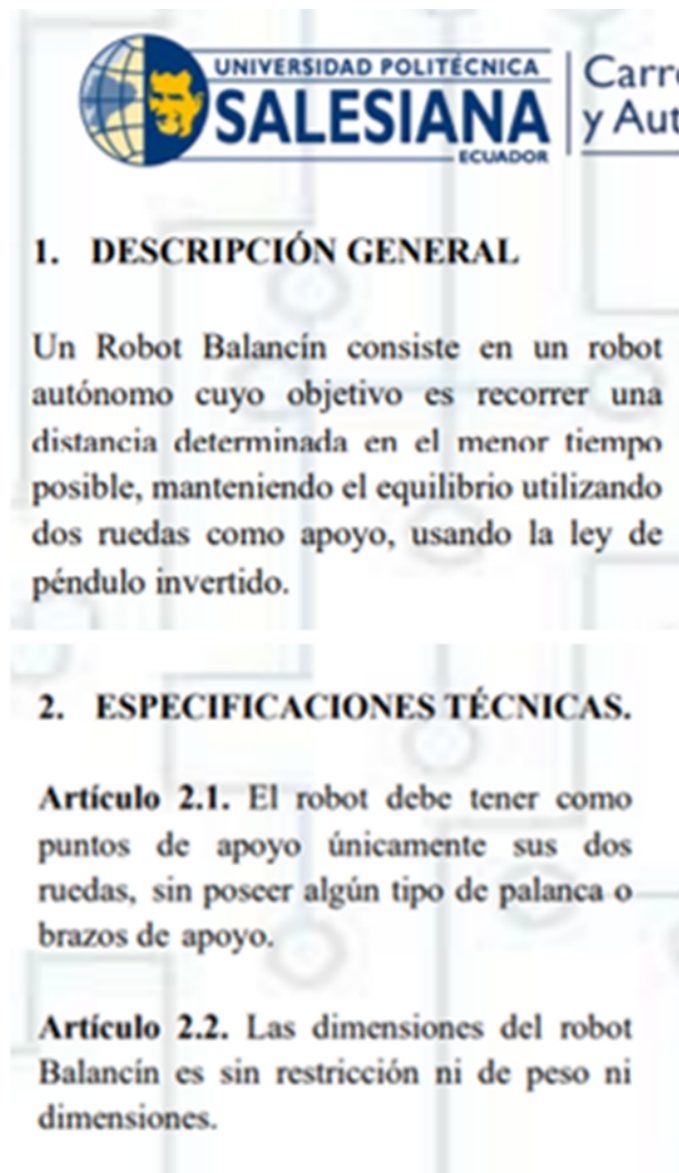


Figura 16. Reglamento de Killtronic 6.0 para la categoría robot balancin

2. Requerimientos Técnicos Del Prototipo.

Artículo 2.1 El robot Balance es un robot construido con materiales livianos, consiste en obtener y mantener el equilibrio, usando dos ruedas como apoyo, sin ayuda externa, este prototipo usa la ley de péndulo invertido con todo su desarrollo matemático y electrónico. La competencia consiste en completar un recorrido en el menor período de tiempo posible, sin perder el equilibrio.

Artículo 2.2 En el preámbulo de la competencia los robots deberán ser sometidos a verificación, si el jurado calificador observara el incumplimiento de alguno de los puntos detallados en el artículo

Artículo 2.3 se proporcionará un lapso de 3 minutos para realizar las debidas correcciones, si el incumplimiento persiste el robot será descalificado.

Artículo 2.4 La arquitectura física y el funcionamiento del robot deberá cumplir con los siguientes puntos:

Balancín con 2 puntos de apoyo

- Ancho: 20 cm
- Largo: 20cm
- Alto: 30cm
- Peso: libre
- Control: El robot balancín deberá funcionar de manera autónoma, quedando prohibido el uso de cualquier tipo de radio control excepto un botón inalámbrico de emergencia que permita parar al robot, pero que no corte de forma total el suministro de energía.

Artículo 2.5 Queda totalmente prohibido que el robot cuente con la existencia de un tercer punto de apoyo.

Artículo 2.6 Al robot se lo deberá identificar su frente y su espalda, este punto deberá ser indicado por parte del equipo en la etapa de HOMOLOGACION ante el comité evaluador.

Figura 17. Reglamento de la Utabot 6.0 para la categoria robot balancin



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



Este reglamento está basado en su totalidad en reglamento del concurso Zero Latitud 2017.

DESCRIPCIÓN GENERAL

Diseñar un robot balance, de uno o dos puntos de apoyo, que pueda mantener el equilibrio sin apoyo externo, debe completar un recorrido en el menor período de tiempo posible, sin perder el equilibrio.

1. CARACTERÍSTICAS TÉCNICAS DEL ROBOT

Por definición el robot de balance debe cumplir con los siguientes puntos:

1.1 Un robot de uno o dos puntos de apoyo, que pueda mantener el equilibrio sin apoyo externo.

1.2 El robot no debe ser construido de tal manera que pueda dañar el ambiente u otros robots. Ver las especificaciones de Seguridad.

1.3 No puede llegar a pesar más de 3 kg, ni podrá utilizar un motor de combustión interna o externa. El robot debe entrar en un cubo de 1m x 1m x 1m durante su turno.

Tipo de control de robot: Autónomo: Una vez que el robot comienza el recorrido, no puede ser tocado por un operador, o serán descalificados.

Control remoto: El equilibrio debe ser autónomo. La conducción puede ser controlada por una unidad de R/C.

Figura 18. Reglamento de la ESPE para la categoría robot balance

VI-B. Diseño estructural mediante un software CAD

Para la estructura del robot, el proceso se dividió en dos etapas: el diseño de los distintos componentes y su posterior ensamblaje:

VI-B1. Diseño de las bases:

La estructura inicial del robot se diseñó utilizando el software SolidWorks. Sus dimensiones se establecerán en función del reglamento previamente visto en las Figuras 16 ,17, 18.

Tomando como referencia estos datos, primero se diseñaron las bases donde se montarán los distintos componentes del robot. Sus dimensiones se definirán a continuación, garantizando una distribución equilibrada de los componentes a emplear:

- 122mm ancho.
- 83mm largo.
- 3mm altura.

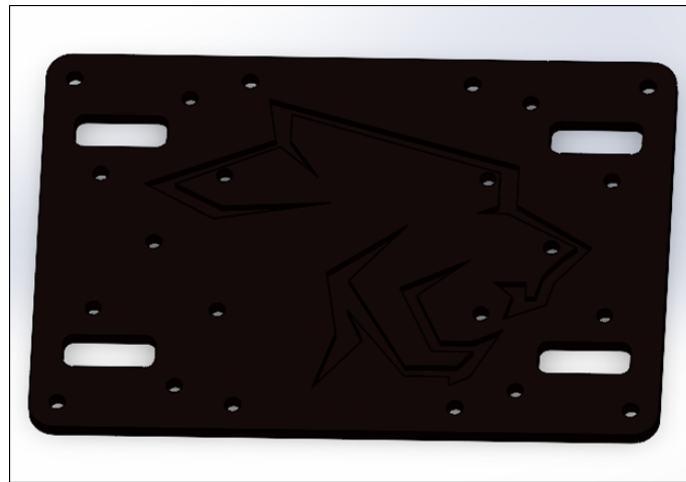


Figura 19. Base del robot diseñada en SolidWorks por autores

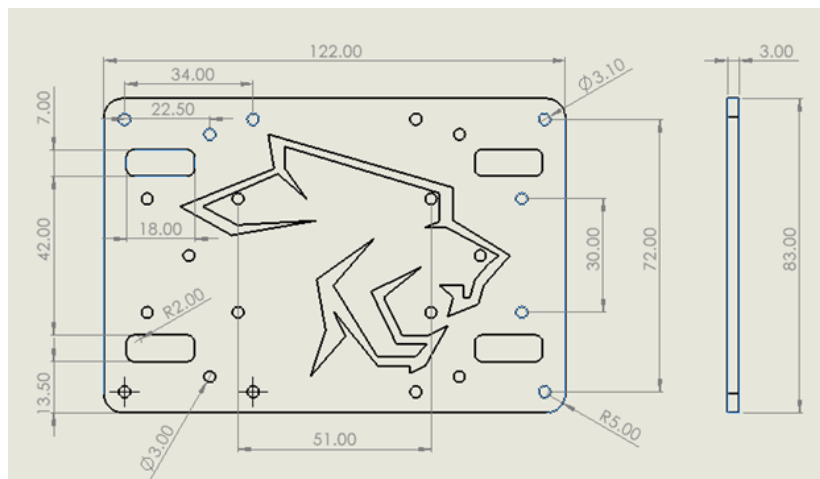


Figura 20. Planos de la base elaborado por Autores

VI-B2. *Diseño de los componentes:*

En el diseño de los distintos componentes que conforman el sistema, se tomaron como referencia sus respectivos datasheets. Para aquellos componentes que no contaban con una documentación técnica detallada, se realizaron mediciones físicas directas a fin de obtener sus dimensiones y características precisas.:

-Soportes motores.

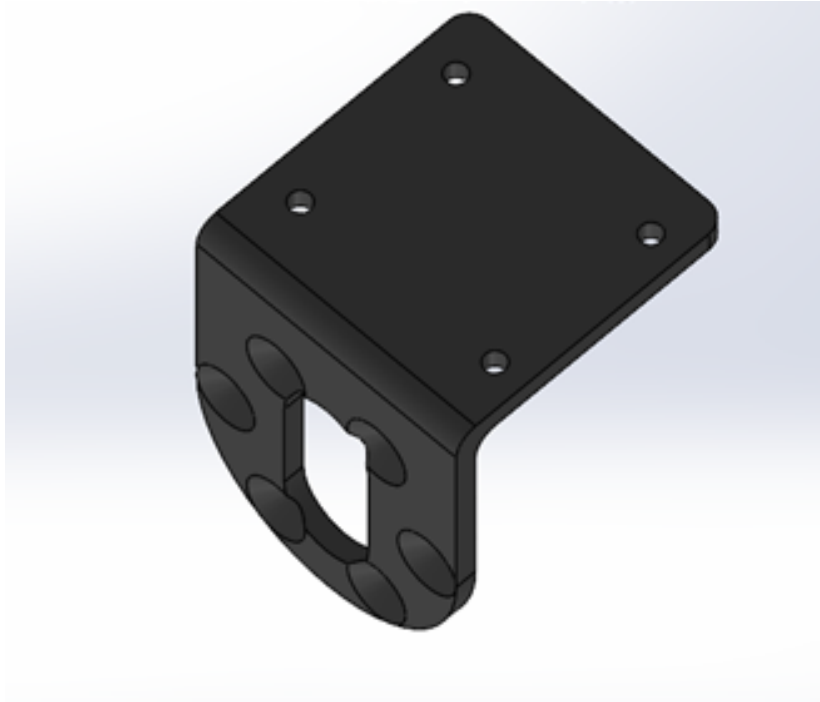


Figura 21. Soporte para motores elaborado por Autores

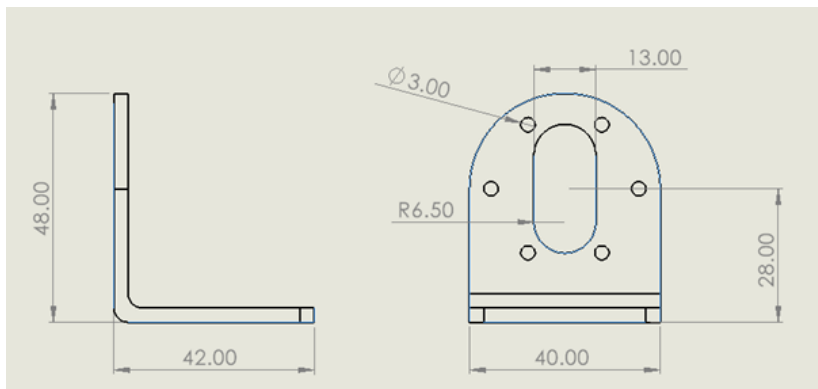


Figura 22. Plano de soporte elaborado por Autores

-Motores GA37-520 DC.

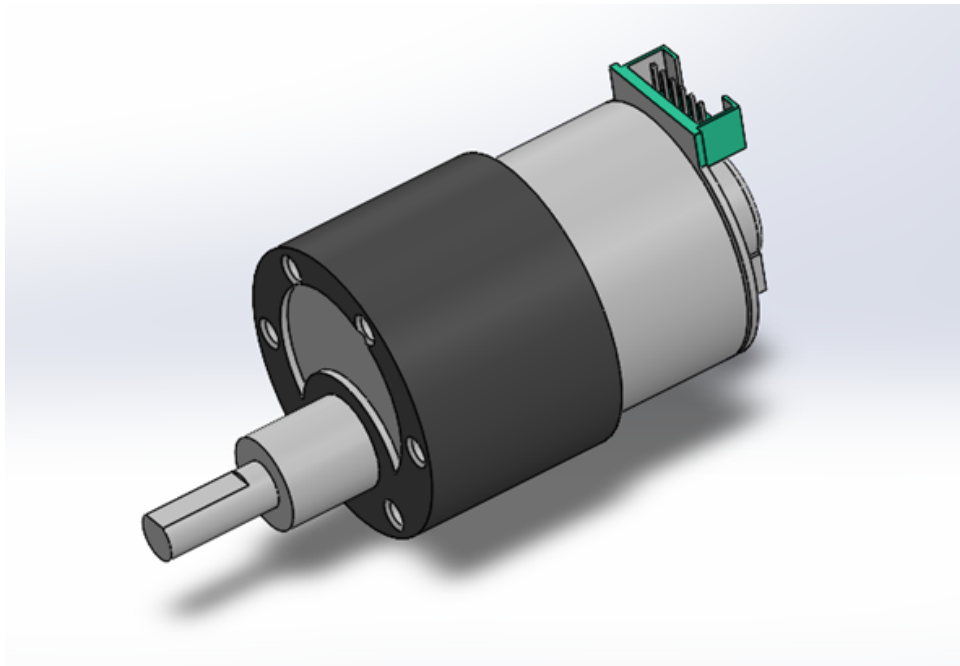


Figura 23. Motor GA37-520 DC elaborado por Autores

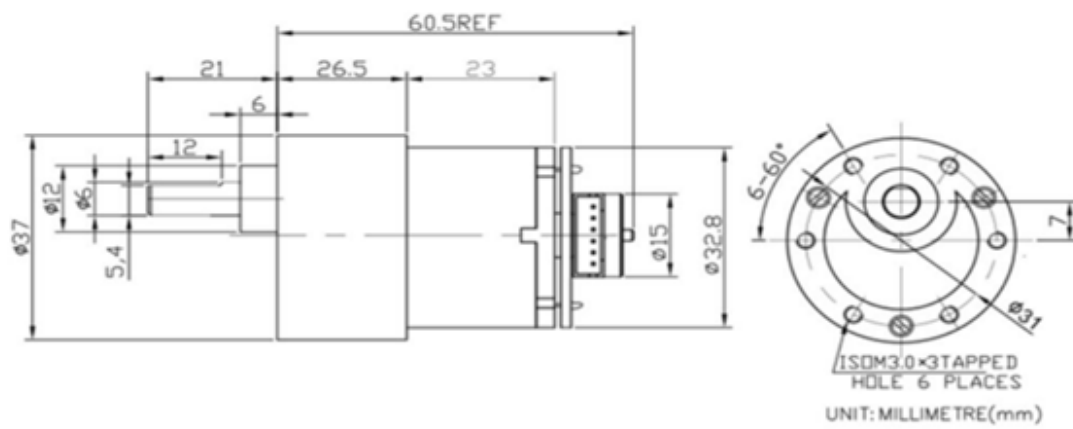


Figura 24. Planos de motor GA37-520 DC elaborado por Autores

-Llantas.

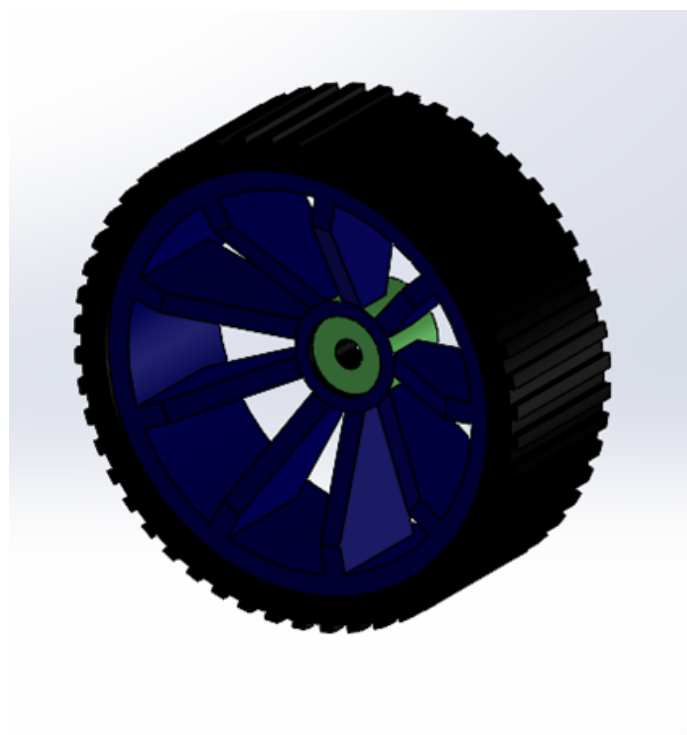


Figura 25. Llanta elaborado por Autores

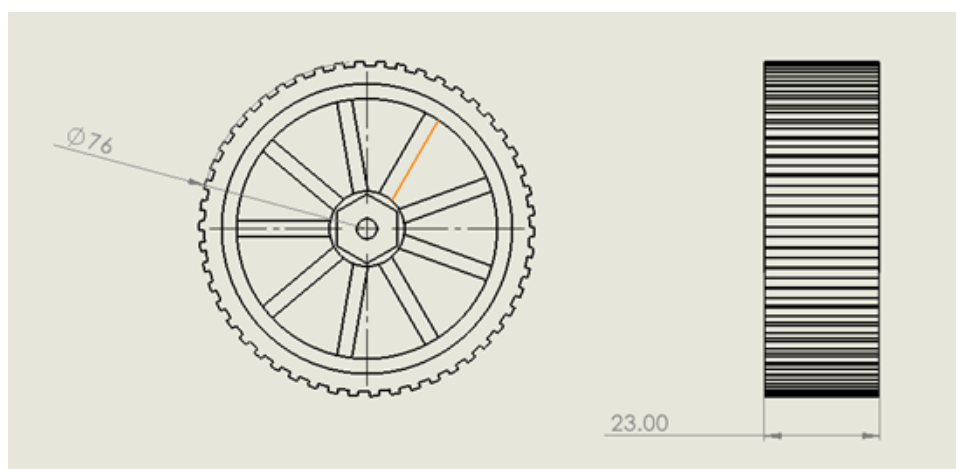


Figura 26. Dimensiones de la llanta

Una vez diseñados todos los componentes y las bases del robot, se procede con el ensamblaje, asegurando que cada pieza se ajuste correctamente a la estructura y cumpla con las especificaciones establecidas en la fase de diseño. Durante este proceso, se verifica que no existan interferencias o colisiones entre los componentes, garantizando un ensamblaje funcional y libre de obstrucciones.

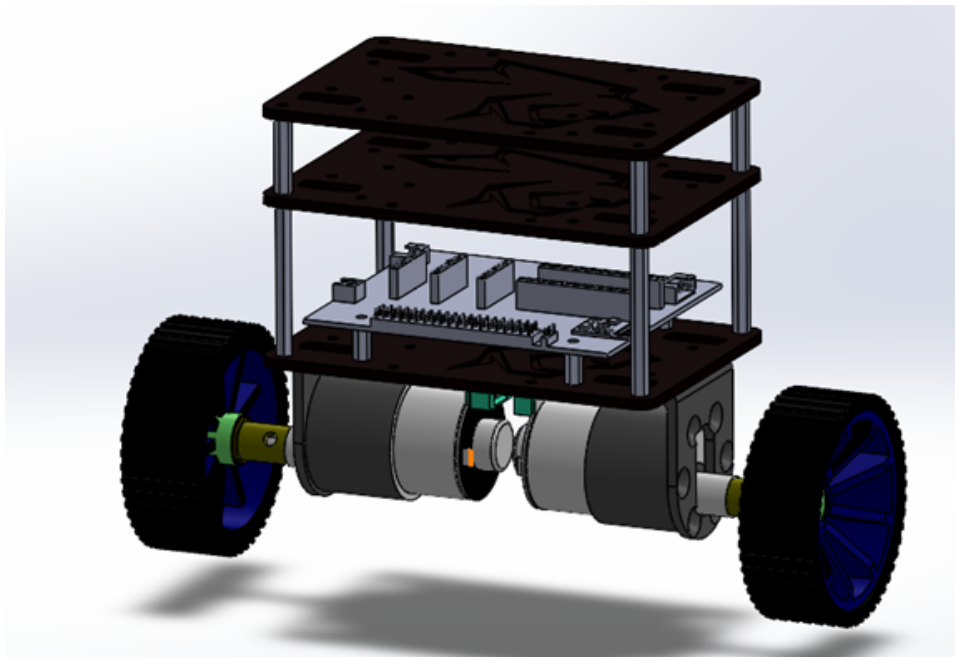


Figura 27. Visualizacion del chasis ensamblado

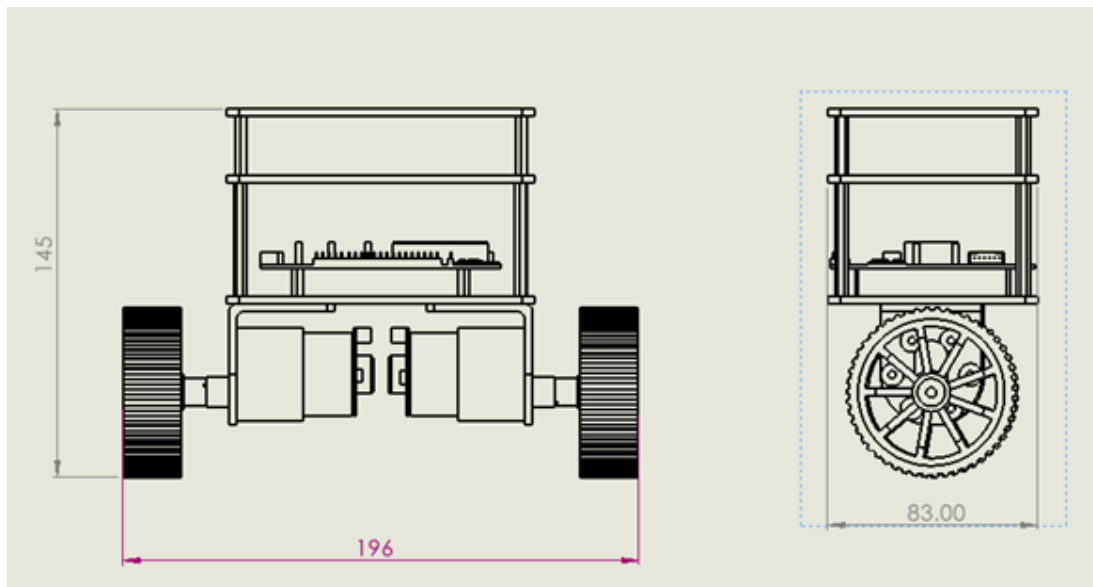


Figura 28. Dimensiones del robot ensamblado

VI-C. Diseño electrónico

VI-C1. Prueba de componentes:

Los componentes serán evaluados en tres grupos. El primero incluirá el sensor MPU6050, mientras que el segundo estará compuesto por el módulo Bluetooth, ya sea el HC-06 o el BT16-DX. Por último, el tercer grupo abarcará las llantas, los motores y el driver TB6612FNG. Todos estos elementos serán gestionados por un procesador, en este caso, un Arduino Nano.

El sensor MPU6050 es un dispositivo de posicionamiento que opera mediante comunicación I2C, lo que le permite interactuar con el Arduino Nano a través de los pines de reloj serial (SCL) y datos (SDA), correspondientes a A5 y A4, respectivamente, las conexiones se presentan en la siguiente tabla.

Tabla VII
CONEXION MPU6050

PIN MPU6050	PIN ARDUINO NANO
VCC	5V
GND	GND
A5	SCL
A4	SDA

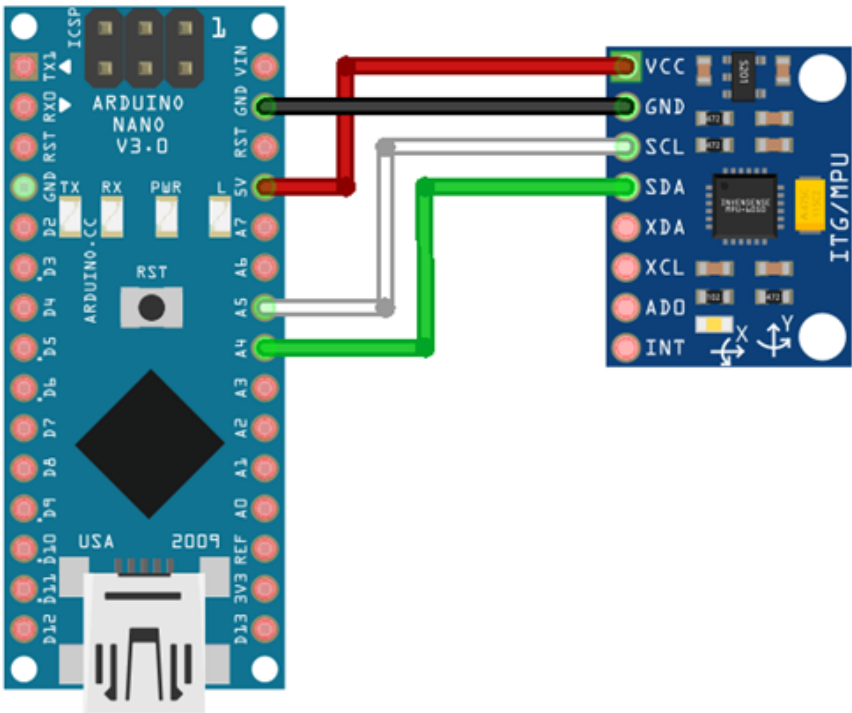


Figura 29. Conexion de pines MPU6050

El segundo grupo corresponde al módulo Bluetooth, ya sea el HC-06 o el BT16-DX. Este componente permitirá establecer la comunicación inalámbrica para controlar el robot, ejecutando las funciones asignadas según las órdenes enviadas desde el dispositivo de control, las conexiones se presentan en la siguiente tabla.

Tabla VIII
CONEXION MODULO BLUETOOTH

Modulo BLUETOOTH	PIN ARDUINO NANO
VCC	5V
GND	GND
RXD	TX
TXD	RX

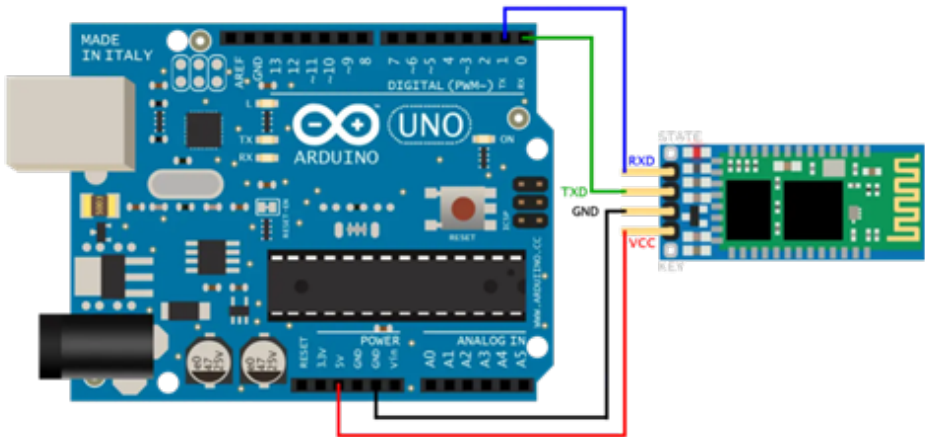


Figura 30. Conexion de pines del modulo BLUETOOTH

El último grupo, compuesto por los motores, el driver TB6612FNG y las llantas, es esencial para la estabilidad del robot mediante el controlador PID. Además de regular la dirección de giro y la velocidad de los motores, se incorporarán encoders para mejorar la precisión de lectura del sistema de control. Las conexiones se detallan en la siguiente tabla.

Tabla IX
CONEXION TB6612FNG A MOTOR Y ARDUINO

Motores con encoder	tb6612fng	PIN ARDUINO NANO
—	VIN	VIN
GND	GND	GND
5V	5V	5V
A	—	3
—	PWMA	5
—	PWMB	11
—	AIN1	4
—	BIN1	6
—	STBY	8
M+	A1	—
M-	A2	—

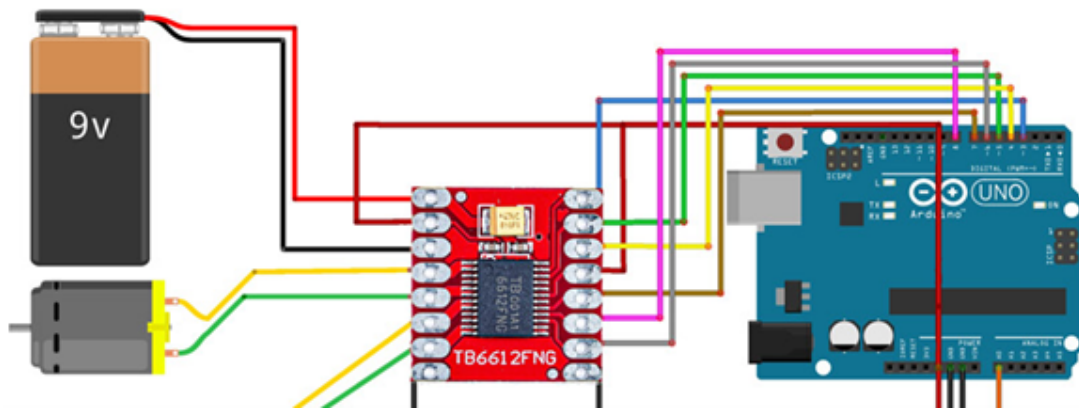


Figura 31. Esquemático de la conexión de pines

VI-C2. Diseño de placa o PCB:

Una vez definidos todos los pines del circuito electrónico del robot péndulo invertido, se procede al diseño de una placa que facilite la integración de cada componente. Esto se realiza por dos razones principales: primero, permite reemplazar fácilmente cualquier componente en caso de fallo significativo, asegurando un rendimiento óptimo para las competencias; segundo, mejora la organización y optimiza la distribución del espacio al momento de ensamblar los elementos del sistema.

En este caso, el diseño se realizó utilizando el software Proteus 8, lo que permitió no solo visualizar las conexiones eléctricas, sino también distribuir eficientemente el espacio de cada componente y obtener una representación clara de la disposición final del circuito.

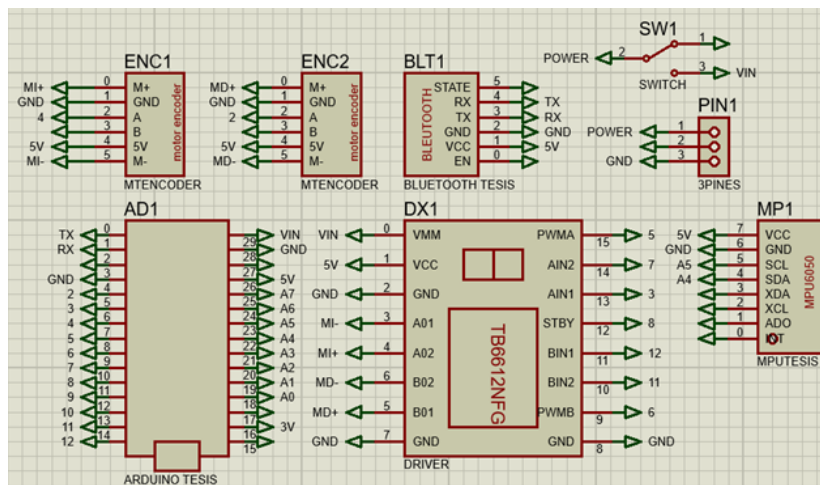


Figura 32. Esquemático mediante proteus8 diseñado por Autores

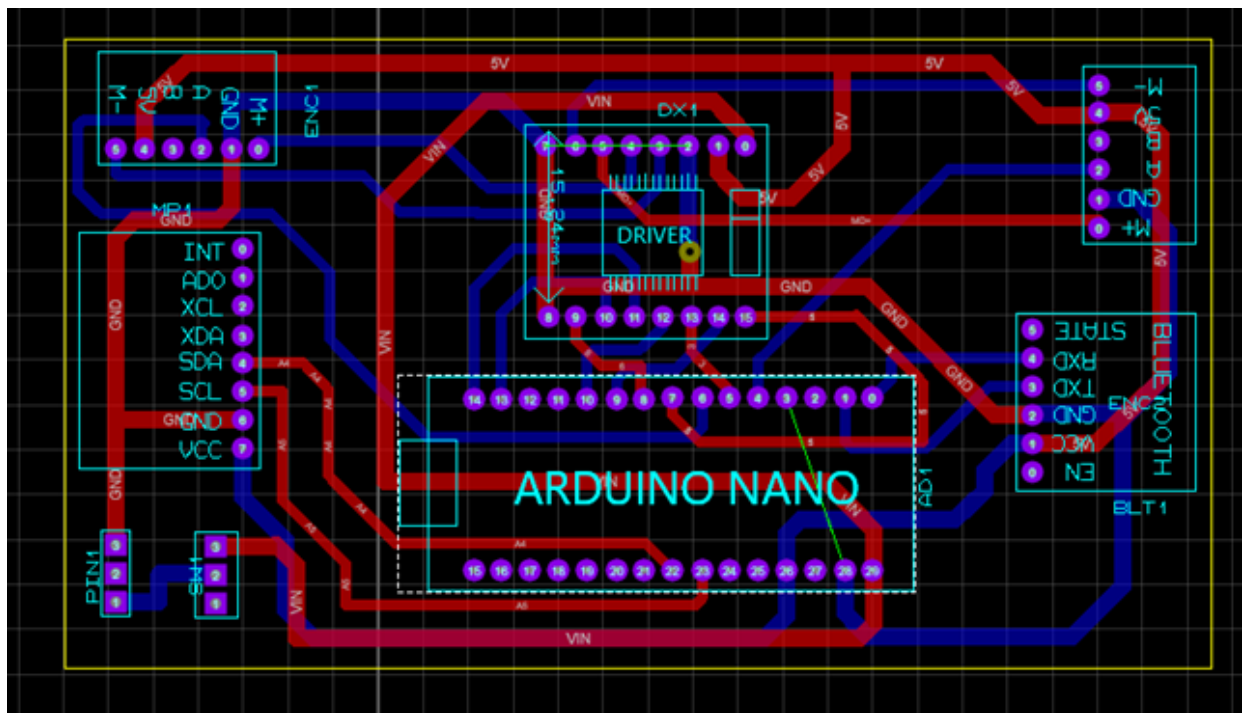


Figura 33. Diagrama electrico proteus8 diseñado por Autores

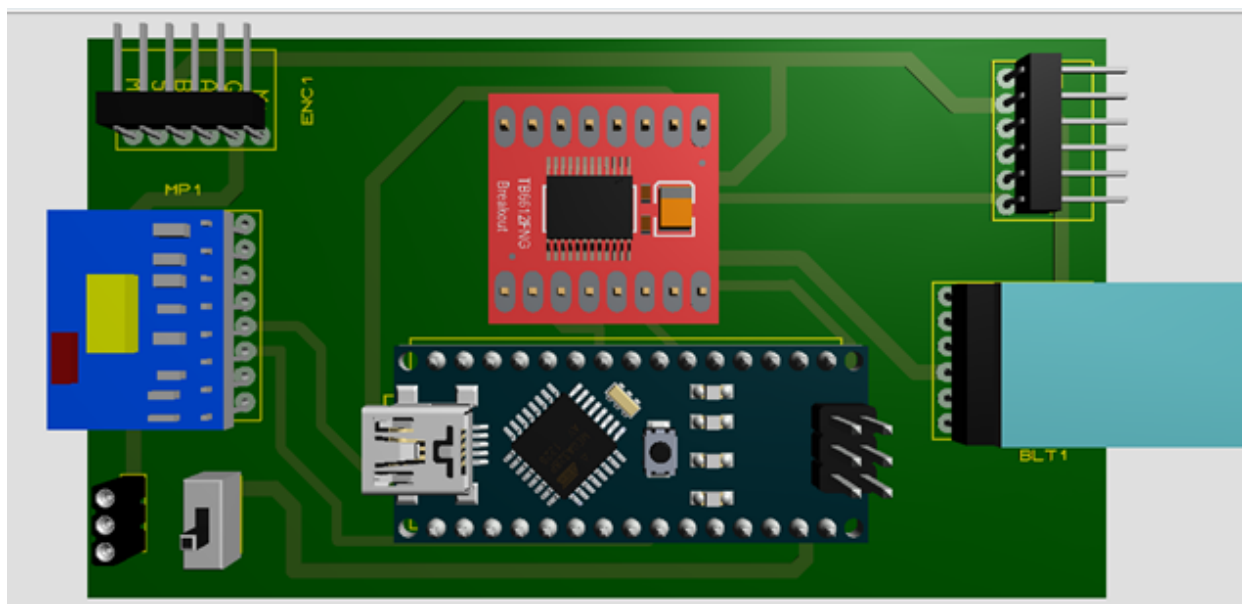


Figura 34. Visualizacion de la PCB realizado por Autores

VI-D. Integración del Control Remoto

VI-D1. Código:

El código gestiona las señales de entrada desde el control remoto para modificar en tiempo real la velocidad y la dirección de giro del robot. Estos ajustes se transmiten a través de la comunicación Bluetooth, permitiendo el control preciso del movimiento del robot.

Las funciones de velocidad están divididas en tres parámetros: SLOW, NORMAL y FAST. Para cambiar entre estas velocidades, se establecieron varias variables que podrán ser manipuladas a través de las funciones definidas en el código.

```
1/  
18 unsigned long start_prev_time = 0;  
19 boolean carInitialize_en = true;  
20 int cart_velocidad = 0;  
21 int cart_giro = 0 ;  
22 int cart_speed_slow = 20;  
23 int cart_speed_normal= 40;  
24 int cart_speed_fast= 80;  
25 int cart_turn_slow = 20;  
26 int cart_turn_normal = 40;  
27 int cart_turn_fast = 80;  
28 int cart_STBY= 0;
```

Figura 35. Variables declaradas en el código base

```
193 void keyEventHandle()  
194 {  
195     if (key_value != '\0')  
196     {  
197         key_flag = key_value;  
198  
199         switch (key_value)  
200         {  
201             case 's':  
202                 rgb.lightOff();  
203                 //motion_mode = STANDBY;  
204                 cart_velocidad = cart_speed_slow;  
205                 cart_giro = cart_turn_slow;  
206                 //delay(3000);  
207                 break;  
208             case 'd':  
209                 rgb.lightOff();  
210                 //motion_mode = STANDBY;  
211                 motion_mode = BACKWARD;  
212                 //delay(3000);  
213                 break;  
214             case 'f':
```

Figura 36. Casos establecidos para las diferentes funciones a controlar

VI-D2. Control Remoto:

El control remoto se realizará mediante la aplicación Eleego. Aunque la aplicación ya cuenta con un control por defecto, se accederá a la sección de creación personalizada para habilitar el cambio de velocidad en tiempo real y los movimientos básicos del robot. En esta sección, es posible añadir más variables, pero para este proyecto, solo se buscará ajustar la velocidad y los movimientos fundamentales.

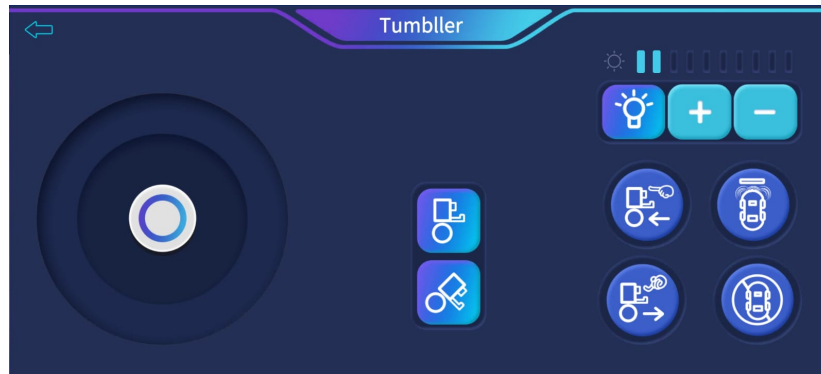


Figura 37. Control remoto por defecto de la app

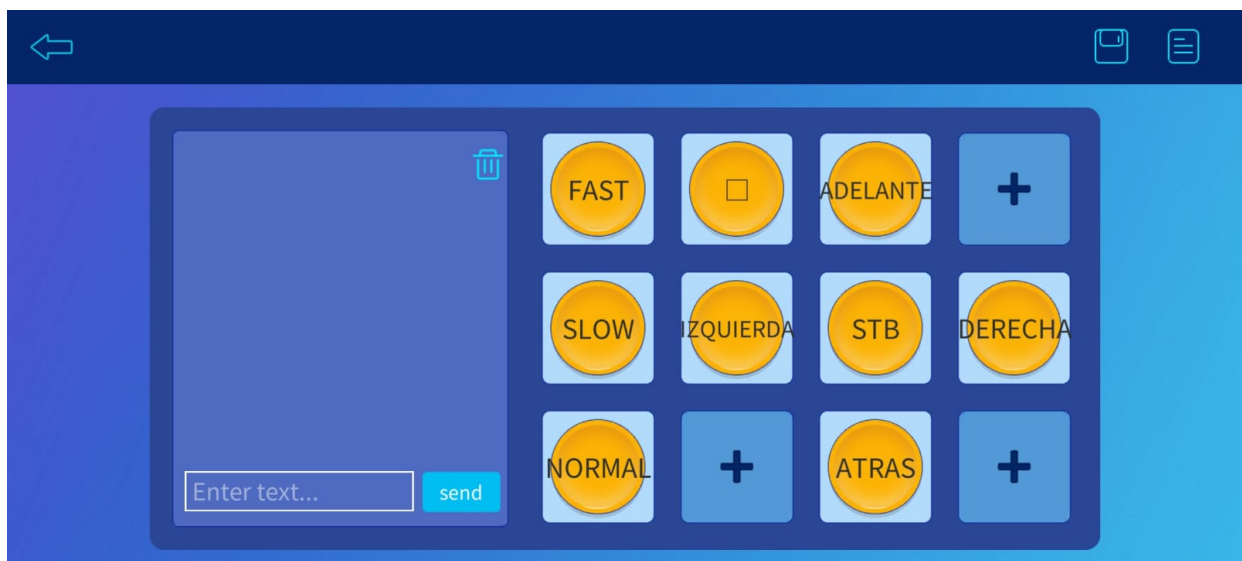


Figura 38. Control desarrollado por Autores

Para este proyecto, se deshabilitaron las funciones del control por defecto que permiten el movimiento del robot, con el fin de evitar que interfieran con nuestro sistema de control personalizado.

VI-E. Algoritmo de trabajo

El algoritmo utilizado para el control del robot péndulo invertido funciona de la siguiente manera: el sensor IMU envía la información sobre la posición angular al Arduino, el cual, en base a estos datos, genera una señal PWM que controla los motores, permitiendo así al robot ajustar su dirección y mantener su estabilidad. Además, el módulo Bluetooth permite controlar los movimientos del robot a través de un control remoto, facilitando la manipulación de la velocidad y la dirección del robot en tiempo real.

A continuación se mostraran de forma grafica los algoritmos que conforman todo el proceso desde lo mas basico hasta lo mas complejo:

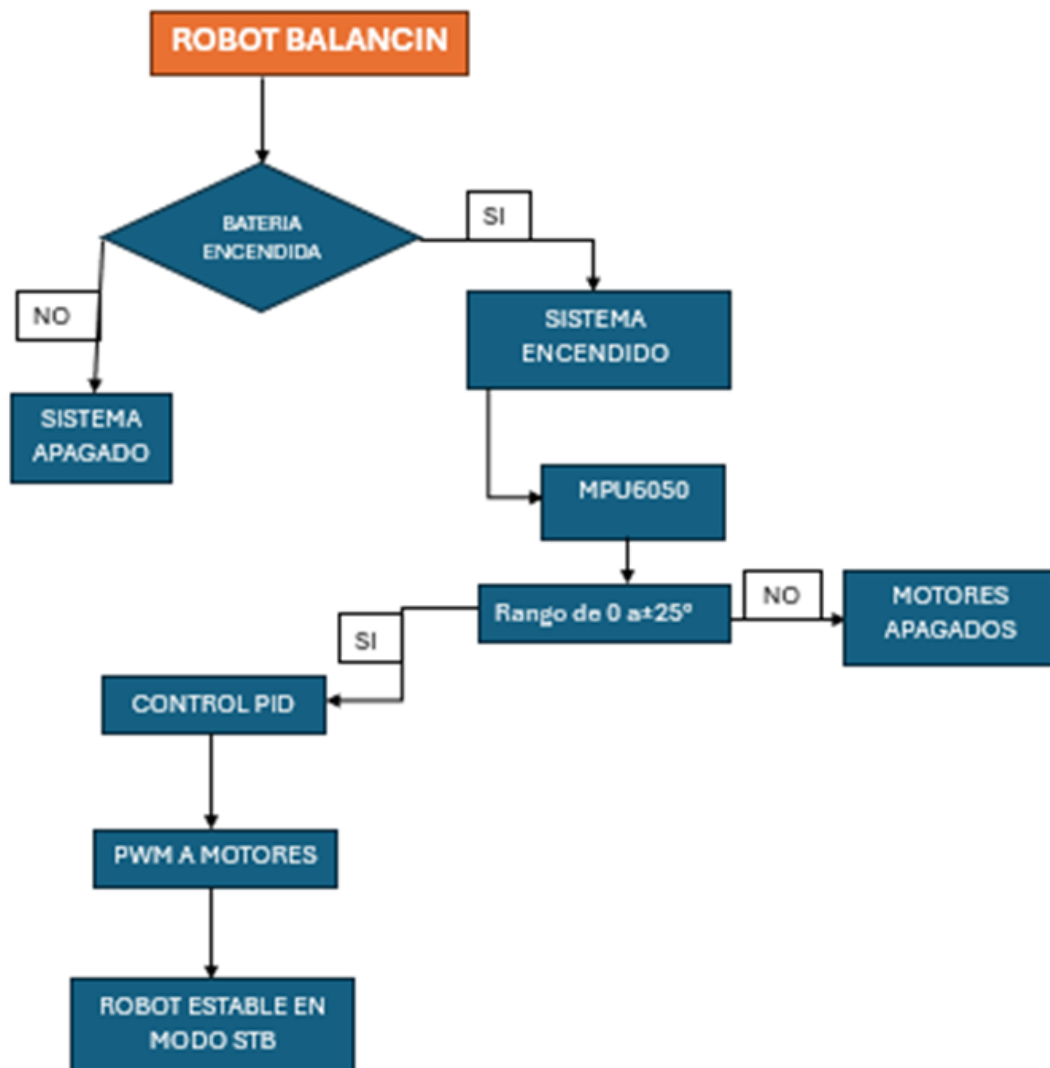


Figura 39. Algoritmo base del robot pendulo invertido

Acontinuacion tenemos el algoritmo de como trabaja todos los componentes internamente:

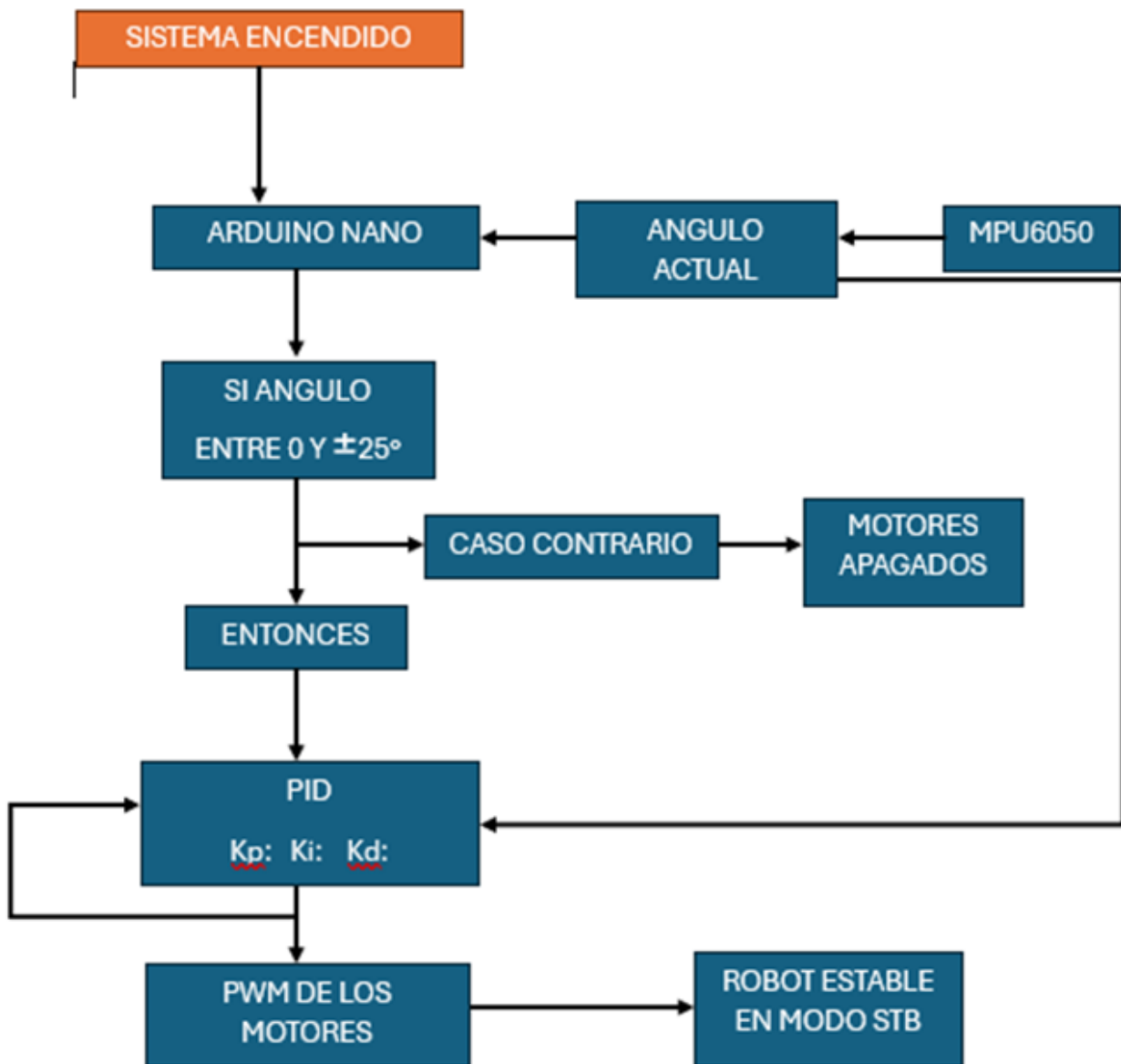


Figura 40. Algoritmo desglosado del robot pendulo invertido

Se incorpora la funcionalidad de Bluetooth al sistema, lo que permite alternar entre dos modos principales: standby y movimiento. El modo de movimiento a su vez se divide en tres submodos: fast, normal y slow. Estas variaciones se controlan mediante dos variables: VL (velocidad lineal) y VG (velocidad angular), lo que permite ajustar la velocidad del robot según la situación, pasando de una velocidad lenta a una rápida según se requiera.

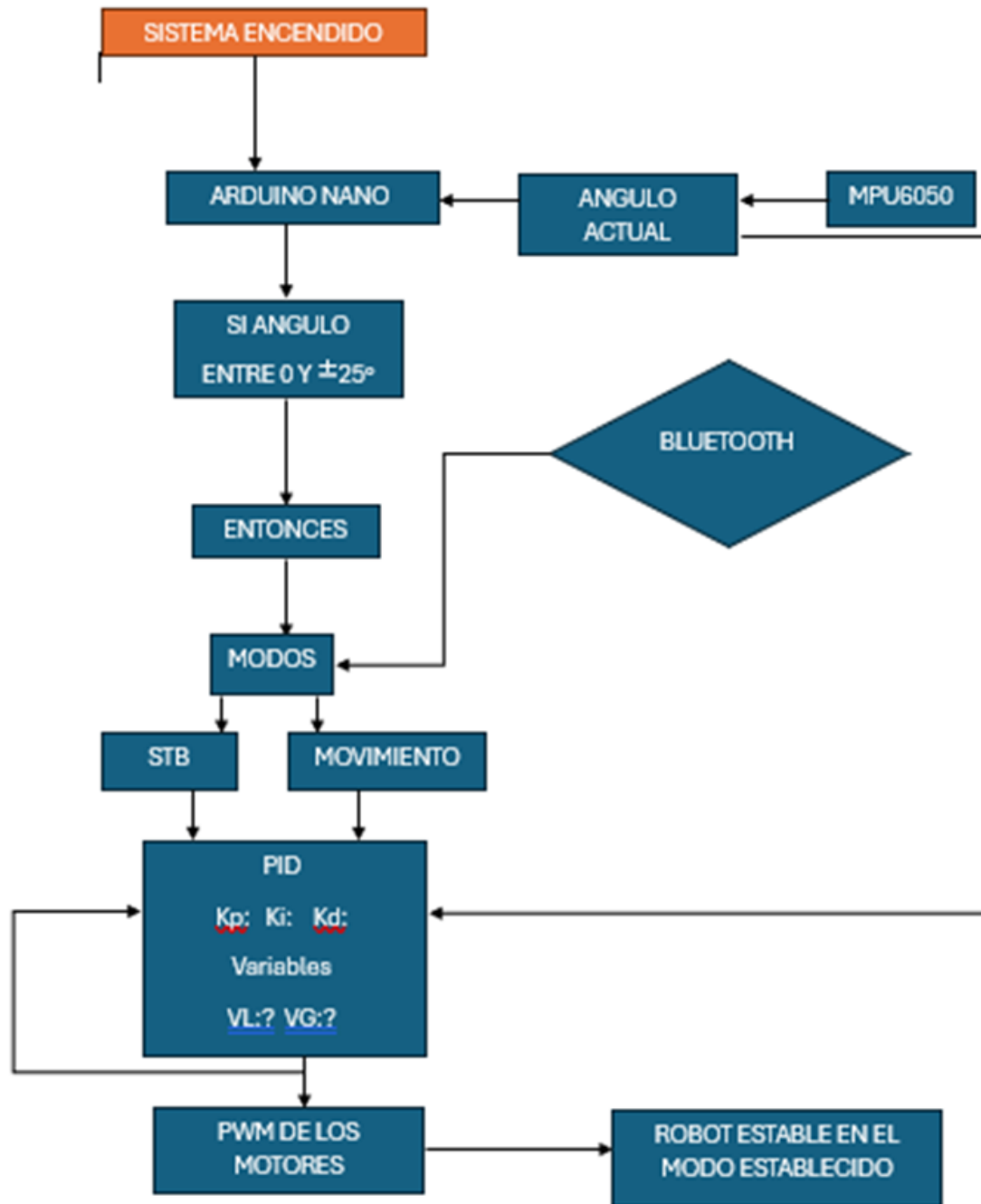


Figura 41. Algoritmo incorporando el bluetooth

VII. RESULTADOS

En esta fase se mostraran los resultados obtenidos en las diferentes etapas de desarrollo del robot péndulo invertido. Desde el diseño conceptual y la implementación del hardware hasta la programación de algoritmos de control y las pruebas de estabilidad y control de este.

VII-A. Transición del Diseño Conceptual a Modelos 3D para Impresión

Las bases previamente diseñadas, como se muestra en las figuras 16 y 17, serán convertidas a formato STL para su posterior impresión en el Club de Robótica.

Se decidió utilizar impresión 3D con material PLA debido a su alta accesibilidad y bajo costo. Este material es ampliamente disponible, lo que facilita la adquisición y reduce los tiempos de espera, permitiendo que el desarrollo del prototipo avance de manera más rápida y eficiente. Además, el PLA es fácil de trabajar con impresoras 3D, lo que facilita la creación de piezas complejas y detalladas sin complicaciones.

Otro factor clave en la elección del PLA es que cumple con las necesidades técnicas del proyecto, al ofrecer una buena relación entre rigidez, resistencia y facilidad de manipulación. Su capacidad para ser utilizado en diseños precisos y la rapidez con la que se pueden realizar ajustes e iteraciones lo convierte en la opción ideal para un prototipo como el robot péndulo invertido, donde se requiere un material confiable para pruebas y modificaciones constantes.

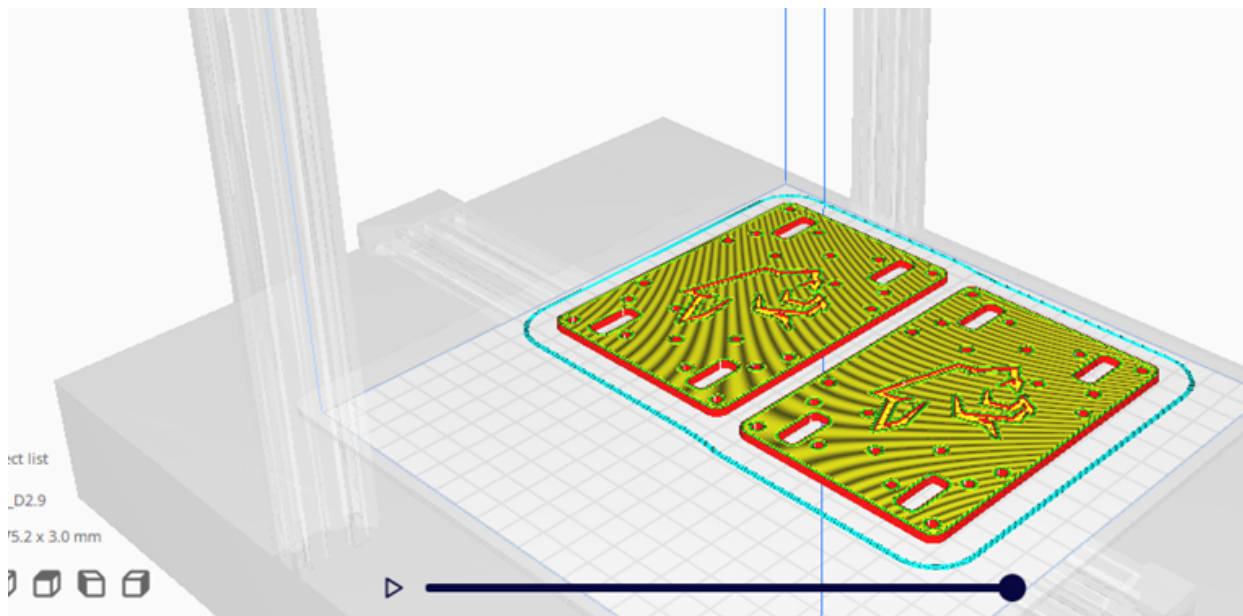


Figura 42. Diseño de las bases para impresion 3D empleando UltimakerCura



Figura 43. Base del robot pendulo invertido impresa por Autores

VII-B. Prototipo 1

VII-B1. Ensamble:

En esta fase, se procedió al ensamblaje de las piezas impresas en 3D, verificando que cada componente se integrara de manera adecuada dentro de la estructura del robot. Se prestó especial atención a la disposición y el ajuste preciso de las bases y otros elementos, asegurando que la alineación de las partes permitiera un ensamblaje eficiente y sin interferencias. Además, se realizó una comprobación detallada para garantizar que todos los componentes estuvieran correctamente posicionados y listos para funcionar de manera óptima en el prototipo.



Figura 44. Ensamble del prototipo 1 diseñado por Autores



Figura 45. Ensamble del prototipo 1 diseñado por Autores

VII-B2. Verificación de componentes:

Se llevó a cabo la verificación de las conexiones y el correcto funcionamiento de cada componente mediante pruebas simples. Se realizaron comprobaciones básicas, como el accionamiento de los motores y la lectura de los sensores, asegurando que respondieran adecuadamente a las señales enviadas. Estas pruebas permitieron identificar posibles fallos en las conexiones o en los dispositivos, facilitando ajustes y correcciones antes de continuar con el desarrollo del prototipo.



Figura 46. Lectura del MPU6050 del prototipo 1



Figura 47. Accionamiento de motores del prototipo 1

Se realizó la prueba del módulo Bluetooth para verificar su correcta comunicación con el sistema. Se estableció conexión con un dispositivo externo y se enviaron comandos de prueba para comprobar la recepción y respuesta del robot. Además, se evaluó la estabilidad de la señal y la latencia en la transmisión de datos, asegurando un funcionamiento adecuado para el control inalámbrico del prototipo.



Figura 48. Prueba de conduccion bluetooth

VII-C. Prototipo 2

VII-C1. Ensamble:

Se diseñó una placa personalizada basada en circuitos previamente utilizados y adaptada a las necesidades de este tipo de robots. Su distribución permitió una integración eficiente del sistema, reduciendo el uso de cables y mejorando la estabilidad del circuito. Además, se tuvieron en cuenta los requerimientos de control, alimentación y comunicación del robot péndulo invertido, asegurando compatibilidad con los sensores y actuadores empleados en el proyecto.

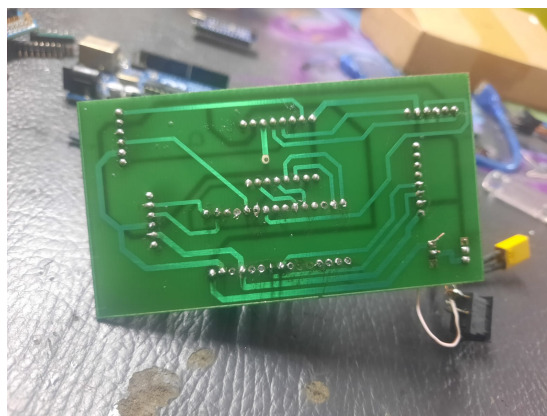


Figura 49. Cableado de las conexiones del prototipo 2

Se realizaron pruebas de funcionamiento en los motores para verificar su correcta respuesta a las señales de control. Se comprobó que giraran en la dirección esperada y con la velocidad adecuada, asegurando una respuesta estable y precisa. Además, se evaluó su desempeño bajo diferentes condiciones de carga para detectar posibles inconsistencias y realizar ajustes antes de integrarlos completamente en el prototipo.

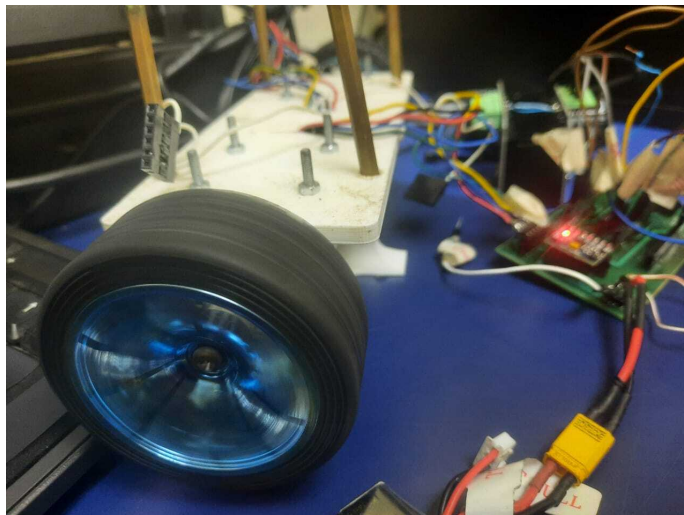


Figura 52. Prueba de motores del prototipo 2 diseñada por Autores

VIII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la Tabla X.

Tabla X
CRONOGRAMA

PROYECTO DE TESIS		DESARROLLO DE UN ROBOT PÉNDULO INVERTIDO MEDIANTE UN CONTROLADOR PID																							
ID	ACTIVIDAD	OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO				FEBRERO				MARZO			
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
1	INVESTIGACIÓN TEÓRICA	X	X	X	X	X																			
2	DISEÑO DEL PROTOTIPO					X	X	X	X																
3	ADQUISICION DE LOS COMPONENTES									X	X	X	X												
4	CONSTRUCCIÓN DEL PROTOTIPO											X	X	X	X										
5	PROGRAMACIÓN DEL PROTOTIPO													X	X	X	X								
6	AJUSTES Y CORRECCIONES																X	X	X	X					
7	DOCUMENTACIÓN Y PRESENTACION																		X	X	X	X	X	X	X

IX. PRESUPUESTO

La siguiente tabla muestra un presupuesto aproximado para un solo prototipo:

Tabla XI
PRESUPUESTO

COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO FINAL
Procesador (Arduino o Raspberry Pi)	Microcontrolador para el control del sistema	1	\$25	\$25
Driver TB66FNG	Controlador de motores para el manejo de corriente y dirección	2	\$5	\$10
Batería LiPo 3S	Batería de polímero de litio de 11.1V	1	\$25	\$25
GA37-520 DC geared motor	Motor de corriente continua con reducción	2	\$25	\$50
Jumpers	Cables de conexión para los componentes	30	\$0.10	\$3
Módulo Bluetooth	Módulo de comunicación inalámbrica	1	\$5	\$5
MPU6050	Sensor de aceleración y giroscopio	2	\$4	\$8
Par de llantas	Ruedas para el movimiento del prototipo	1	\$10	\$10
Total				\$161

X. CONCLUSIONES

-El uso del software Arduino IDE permitió una implementación sencilla y eficiente del control del robot péndulo invertido o balancín, facilitando la integración del código con los sensores y actuadores del sistema.

-En las pruebas realizadas, se observó una notable estabilidad en el sistema, lo cual demuestra el correcto funcionamiento y ajuste del controlador PID. Asimismo, el control remoto operó de manera eficiente, cumpliendo con todas las funciones establecidas de forma precisa y confiable. Estos resultados reflejan el éxito en la implementación y validación de los componentes clave del proyecto.

-El desarrollo de este tipo de proyectos brinda a los estudiantes la oportunidad de aplicar y fortalecer los conocimientos adquiridos, integrando aspectos teóricos y prácticos. Además, este proyecto sirve como base para futuras variaciones y aplicaciones, como robots seguidores de línea, robots de carga, u otros sistemas autónomos, lo que fomenta la innovación y el pensamiento creativo en el ámbito de la robótica y la ingeniería.

XI. RECOMENDACIONES

-Se recomienda verificar las especificaciones eléctricas y mecánicas de cada componente antes de su implementación, asegurando su compatibilidad con el sistema para evitar inconvenientes durante las pruebas y la fase de integración.

-Se recomienda analizar códigos preexistentes para comprender a profundidad la lógica de programación utilizada en el prototipo, facilitando su depuración y mejora.

-Se recomienda realizar una revisión exhaustiva del estado de cada componente utilizado, así como del código implementado, para garantizar su correcto funcionamiento. Esta verificación debe incluir tanto el análisis de las conexiones eléctricas y mecánicas, como el testeo de las funcionalidades del código en diversas condiciones operativas. De esta manera, se evitarán errores durante la ejecución del prototipo y se reducirá el riesgo de daños físicos a los componentes debido a un mal funcionamiento.

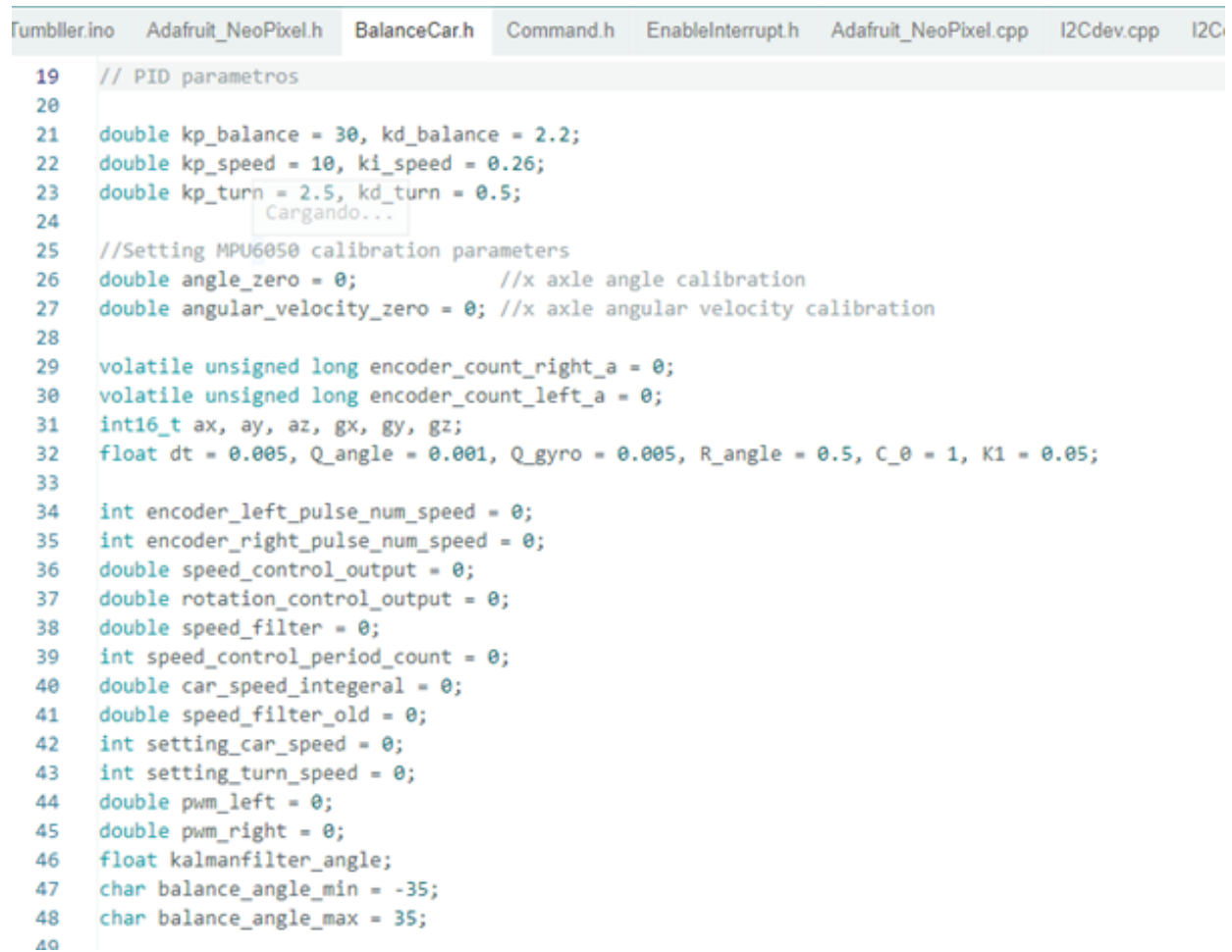
REFERENCIAS

- [1] L. A. Aldaz Andrade y R. D. Orellana Torres, «Desarrollo de un prototipo de robot móvil de competencia multi-categoría,» B.S. thesis, 2017.
- [2] S. Daza-Suárez y J. S. Quintero-Guerrero, «Construcción de un prototipo de un sistema de péndulo invertido estabilizado,» 2023.
- [3] M. Kuzbari Mora, «Estudio comparativo entre el control Pid clásico y el control Pid fraccionario aplicado al sistema del péndulo invertido,» 2020.
- [4] Á. R. Choca Varela y L. D. Moya Yanchatipan, «Diseño e implementación de un robot batalla de 30 libras,» B.S. thesis, 2022.
- [5] C. L. Cifuentes Briones, «Diseño de un algoritmo de control para el péndulo invertido simple marca feedback modelo 33-936s usando variables de estado.,» B.S. thesis, 2018.
- [6] J. C. Baas, J. R. Martínez, E. O. Díaz y F. M. Mena, «Implementación de un controlador para robot de tipo péndulo invertido,» *Revista Perspectivas*, vol. 6, n.º 2, 2024.
- [7] B. R. Pila Acosta, «Construcción de un robot de sumo con el kit lego MINDSTORMS home ev3 considerando las reglas de la competencia ROBOT GAMES ZERO LATITUD®,» 2021.
- [8] D. A. C. Benavides, J. L. L. Arias, G. A. D. Martínez, J. E. Salazar y A. F. R. Álvarez, «SISTEMA EMBEBIDO ROBOT MINISUMO SIT-UV: UNA HERRAMIENTA DE EDUCACIÓN,» *Encuentro Internacional de Educación en Ingeniería*, 2020.
- [9] V. M. G. Macías y E. Intriago, «La robótica en el ámbito educativo de Ecuador,» *Serie Científica De La Universidad De Las Ciencias Informáticas*, vol. 15, n.º 8, págs. 84-93, 2022.
- [10] P. A. R. Garcia, H. J. V. Montiel y H. G. R. Macas, «Robots Mini Sumo RC de competencia para el club de robótica del Instituto Superior Tecnológico Tsa chila: Mini Sumo RC robots for competition for the robotics club of the Instituto Superior Tecnológico Tsa chila,» *Boletín Científico Ideas y Voces*, págs. 875-908, 2023.
- [11] L. Toca, D. Romo, C. Ruiz y B. Machay, «Diseño de robots de Competencia en Categorías Seguidor de Línea y Robot Laberinto,» *NEXOS CIENTÍFICOS-ISSN 2773-7489*, vol. 3, n.º 2, págs. 28-42, 2019.
- [12] B. F. Cabrera Sarmiento y M. J. Martinez Avilés, «Diseño de una herramienta de acople para el proceso de soldadura MIG aplicada al robot KUKA KR5,» B.S. thesis, 2022.
- [13] H. R. Guaigua Cisneros, «Diseño e implementación de robots de competencia: diseño e implementación de un robot de competencia categoría robot soccer,» B.S. thesis, Quito: EPN, 2023., 2023.
- [14] F. R. C. Aguayo y D. G. A. Vilema, «Robot equilibrista sobre dos ruedas en configuración diferencial para concurso de robótica Riotronic 2018,» *Revista Caribeña de Ciencias Sociales (RCCS)*, n.º 11, pág. 23, 2018.
- [15] J. O. Saldarriaga Andrade y J. J. Mero Simisterra, «Diseño e implementación de dos robots de batalla de 3 libras para el Club de Robótica de la Universidad Politécnica Salesiana,» B.S. thesis, 2020.
- [16] A. E. Salazar Andrade, «Estudio comparativo del desempeño de tres tipos de controladores para el Péndulo Invertido Furuta.,» B.S. thesis, 2018.
- [17] N. E. G. Díaz, «Péndulo simple,»
- [18] C. E. Infante Mejia y J. A. Navarrete Burgos, «Diseño de un controlador inteligente para un sistema inestable tipo péndulo invertido con monitorio remoto por IoT,» B.S. thesis, 2024.
- [19] D. K. Solórzano Peñafiel, «Diseño e implementación de un controlador óptimo LQG, para un sistema de péndulo invertido aplicado en un equipo Lego Mindstorms.,» B.S. thesis, 2018.
- [20] L. A. Mendoza Reyes, «Diseño, implementación y control de un mecanismo de péndulo invertido con volante de inercia,» 2023.

- [21] T. José e Y. Claribel, «Diseño e implementación de un sistema de control lineal cuadrático Gaussiano en un sistema no lineal electromecánico denominado Robot Balancín.,» B.S. thesis, La Libertad: Universidad Estatal Península de Santa Elena, 2024, 2024.
- [22] J. E. Tutiven Reyes y D. E. Villagómez Galarza, «Diseño e Implementación de SLAM y control PID de un robot autónomo Robomaster S1 utilizando Python,» B.S. thesis, 2021.
- [23] M. Á. Uvillus Chancusig, «Robot móvil para las competencias de insecto con obstáculos, sin obstáculos y trepador con corrección de trayectoria,» B.S. thesis, 2023.
- [24] J. M. Sarmiento Cedeño y J. S. Cedeño Cruz, «Diseño de un controlador pid difuso para un robot auto balanceado con Tecnología PITSCO TETRIX PRIME Y MYRIO,» B.S. thesis, 2020.
- [25] F. Zayas-Gato, H. Quintián, E. Jove, J.-L. Casteleiro-Roca y J. L. Calvo-Rolle, *Diseño de controladores PID*. Universidade da Coruña, Servizo de Publicacións, 2020.
- [26] A. Valenzuela Cobos y F. G. Escandón, «Diseño e implementación de módulos para prácticas de controladores PID convencionales,» mar. de 2019.
- [27] J. C. Corredor Carreño y O. A. González Triana, «Diseño de un controlador adaptativo basado en los parámetros de sintonización de ziegler y nichols para plantas de primer orden,» 2022.
- [28] J. I. Canduela Ilundain et al., «Sistema de control PID de temperatura para la docencia de asignaturas de regulación automática,» 2022.
- [29] R. P. Rivera, «Métodos de sintonía de controlador PID para un motor K90-T2,» *Alternativas*, vol. 24, n.º 1, págs. 5-16, 2023.
- [30] J. A. Cartagena Izquierdo y J. A. Ramirez De La Torre, «Diseño e implementación de un Control difuso para un Robot seguidor de línea con autobalanceo utilizando lego Mindstorms,» B.S. thesis, 2022.
- [31] A. Nano, «Arduino nano,» *A MOBICON Company*, pág. 30, 2018.
- [32] A. D. Ñacata Paucar, «Implementación de un sistema WNCS mediante una plataforma basada en la nube para el control de un péndulo invertido,» B.S. thesis, 2022.
- [33] C. O. Abrajan Arias, «Diseño y construcción de un robot seguidor de línea evasor de obstáculos empleando Arduino Nano.,» 2020.
- [34] E. A. Chango Quishpe, «Diseño y construcción de un robot mini sumo de competencia controlado con radio control.,» 2023.

XII. ANEXOS

Se incorporaron los códigos principales del sistema, tales como el control PID, la gestión de los modos de operación y la asignación de pines.



```
19 // PID parametros
20
21 double kp_balance = 30, kd_balance = 2.2;
22 double kp_speed = 10, ki_speed = 0.26;
23 double kp_turn = 2.5, kd_turn = 0.5;
24 Cargando...
25 //Setting MPU6050 calibration parameters
26 double angle_zero = 0; //x axle angle calibration
27 double angular_velocity_zero = 0; //x axle angular velocity calibration
28
29 volatile unsigned long encoder_count_right_a = 0;
30 volatile unsigned long encoder_count_left_a = 0;
31 int16_t ax, ay, az, gx, gy, gz;
32 float dt = 0.005, Q_angle = 0.001, Q_gyro = 0.005, R_angle = 0.5, C_0 = 1, K1 = 0.05;
33
34 int encoder_left_pulse_num_speed = 0;
35 int encoder_right_pulse_num_speed = 0;
36 double speed_control_output = 0;
37 double rotation_control_output = 0;
38 double speed_filter = 0;
39 int speed_control_period_count = 0;
40 double car_speed_integral = 0;
41 double speed_filter_old = 0;
42 int setting_car_speed = 0;
43 int setting_turn_speed = 0;
44 double pwm_left = 0;
45 double pwm_right = 0;
46 float kalmanfilter_angle;
47 char balance_angle_min = -35;
48 char balance_angle_max = 35;
49
```

Figura 53. Parámetros del código PID

```

{
    sei();
    encoder_left_pulse_num_speed += pwm_left < 0 ? -encoder_count_left_a : encoder_count_left_a;
    encoder_right_pulse_num_speed += pwm_right < 0 ? -encoder_count_right_a : encoder_count_right_a;
    encoder_count_left_a = 0;
    encoder_count_right_a = 0;
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    kalmanfilter.Angle(ax, ay, az, gx, gy, gz, dt, Q_angle, Q_gyro, R_angle, C_0, K1);
    kalmanfilter_angle = kalmanfilter.angle;
    double balance_control_output = kp_balance * (kalmanfilter_angle - angle_zero) + kd_balance * (kalmanfilter.Gyro_x - angular_velocity_zero);

    speed_control_period_count++;
    if (speed_control_period_count >= 8)
    {
        speed_control_period_count = 0;
        double car_speed = (encoder_left_pulse_num_speed + encoder_right_pulse_num_speed) * 0.5;
        encoder_left_pulse_num_speed = 0;
        encoder_right_pulse_num_speed = 0;
        speed_filter = speed_filter_old * 0.7 + car_speed * 0.3;
        speed_filter_old = speed_filter;
        car_speed_integral += speed_filter;
        car_speed_integral += -setting_car_speed;
        car_speed_integral = constrain(car_speed_integral, -3000, 3000);
        speed_control_output = -kp_speed * speed_filter - ki_speed * car_speed_integral;
        rotation_control_output = setting_turn_speed + kd_turn * kalmanfilter.Gyro_z;
    }

    pwm_left = balance_control_output - speed_control_output - rotation_control_output;
    pwm_right = balance_control_output - speed_control_output + rotation_control_output;
}

```

Figura 54. Código del control PID

```

7  /*
3  #include <Arduino.h>
9  #include "Pins.h"
9  #include "mode.h"
1  #include "Command.h"
2  #include "BalanceCar.h"
3  #include "Rgb.h"
4  #include "Ultrasonic.h"
5  #include "voltage.h"
5  #include "EnableInterrupt.h"
7
3  unsigned long start_prev_time = 0;
9  boolean carInitialize_en = true;
9  int cart_velocidad = 0;
1  int cart_giro = 0 ;
2  int cart_speed_slow = 20;
3  int cart_speed_normal= 40;
4  int cart_speed_fast= 80;
5  int cart_turn_slow = 20;
5  int cart_turn_normal = 40;
7  int cart_turn_fast = 80;
8  int cart_STBY= 0;

```

Figura 55. Parámetros de velocidad incorporados al prototipo

```

1  | #ifndef Pins_h
2  | #define Pins_h
3  |
4  | #define VOL_MEASURE_PIN A2
5  | #define ECHO_PIN A3
6  | #define TRIG_PIN A2
7  | #define RECV_PIN A1
8  | #define NUMPIXELS 4
9  | #define RGB_PIN 9
10 | #define AIN1 7
11 | #define AIN2 3
12 | #define PWMA_LEFT 5
13 | #define BIN1 12
14 | #define BIN2 11
15 | #define PWMB_RIGHT 6
16 | #define STBY_PIN 8
17 | #define ENCODER_LEFT_A_PIN 2
18 | #define ENCODER_RIGHT_A_PIN 4
19 | #define IR_SEND_PIN 9
20 | #define LEFT_RECEIVE_PIN A0
21 | #define RIGHT_RECEIVE_PIN A1
22 | #define KEY_MODE 10
23 | #endif
24 |

```

Figura 56. Pines del arduino involucrados en el prototipo

A continuación, se presentan los prototipos finalizados, los cuales han sido ensamblados y verificados para garantizar su correcto funcionamiento. Estos modelos están listos para ser evaluados en diferentes pruebas, permitiendo analizar su desempeño y realizar los ajustes necesarios para su optimización.



Figura 57. Prototipo 1



Figura 58. Prototipo 2



Figura 59. Soldado de PCB del prototipo 2

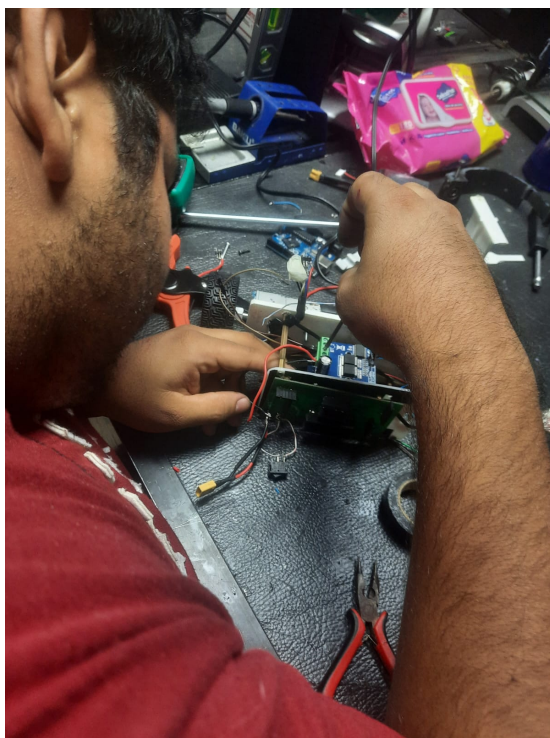


Figura 60. Ensamblado del prototipo 2



Figura 61. Prueba final del prototipo 2

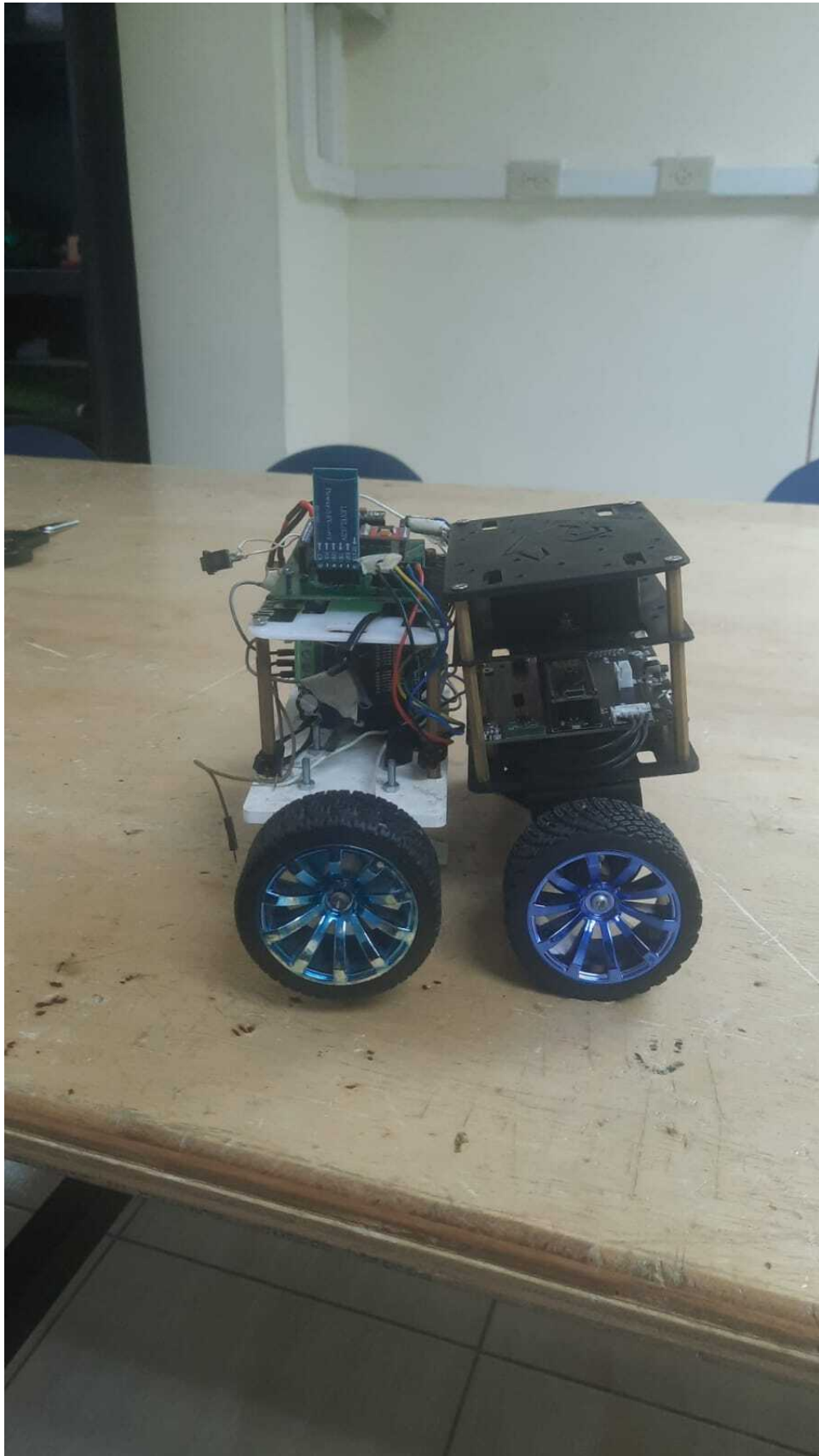


Figura 62. Robots balancines