

CARRERA DE ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT
BALANCÍN AUTONOMO Y CONTORLADO
QUE SE PUEDA MOVER EN TERRENOS IRREGULARES**

**Trabajo de Integración Curricular previo a la obtención del Título de
Tecnólogo/a Superior en Electrónica**

Autor/es: HENRY JAIR VACA CÁCERES

Tutor: ING. CÉSAR MINAYA

**MARZO, 2025
SANGOLQUI – ECUADOR**

Autor: Henry Jair Vaca Cáceres	Título a obtener: Tecnólogo/a Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: henry.vaca@ister.edu.ec
	
Dirigido por: Ing. César Minaya	Matriz: Sangolquí -Ecuador Correo electrónico: cesar.minaya@ister.edu.ec
	

TODOS LOS DERECHOS RESERVADOS.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2024 Tecnológico Universitario Rumiñahui
SANGOLQUÍ – ECUADOR

VACA CÁCERES HENRY JAIR

***DISEÑO E IMPLEMENTACIÓN DE UN ROBOT BALANCÍN AUTÓNOMO Y
CONTORLADO QUE SE PUEDA MOVER EN TERRENOS IRREGULARES***

APROBACIÓN DEL TUTOR/A

En mi calidad de Tutor/a del Trabajo de Titulación, presentado por **Vaca Cáceres Henry Jair** con CC: **1719686279** para optar por el Título de Tecnólogo/a Superior en Electrónica; cuyo título es: **Tecnólogo superior en Electrónica** se considera que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que fuere designado.

Sangolquí, a los 18 días del mes de marzo de
2025



CESAR ANDRES MINAYA ANDINO

DOCENTE-TUTOR/A

C.I. 0604887919

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.2

Sangolquí, 14 de marzo del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, **HENRY JAIR VACA CACERES** declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado **DISEÑO E IMPLEMENTACION DE UN ROBOT BALANCIN AUTONOMO Y CONTROLADO QUE SE PUEDA MOVER EN TERRENOS IRREGULARES**, de la Tecnología Superior **EN ELECTRONICA**; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.



Atentamente,
HENRY JAIR VACA CACERES
C.I.: 1719686279

FORMULARIO PARA ENTREGA DE PROYECTOS EN
BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO
RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA:

TECNOLOGÍA SUPERIO EN ELECTRONICA

AUTOR /ES:

HENRY JAIR VACA CACERES

TUTOR (METODOLÓGICO Y ACADÉMICO):

RICARDO ELICIO ROSERO YUGSI

CONTACTO ESTUDIANTE:

0963392780

CORREO ELECTRÓNICO:

henry.vaca@ister.edu.ec

TEMA:

**"DISEÑO E IMPLEMENTACIÓN DE UN
ROBOT BALANCÍN AUTÓNOMO Y
CONTORLADO
QUE SE PUEDA MOVER EN TERRENOS IRREGULARES"**

OPCIÓN DE TITULACIÓN:

UNIDAD DE INTEGRACIÓN CURRICULAR RESUMEN

RESUMEN EN ESPAÑOL:

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero

Telf: 0960052734 / 023524576 / 022331628

www.ister.edu.ec / info@ister.edu.ec

El presente trabajo de tesis se centra en el diseño e implementación de un robot balancín autónomo y controlado, capaz de moverse sobre terrenos irregulares. El proyecto se fundamenta en el modelo del péndulo invertido, donde se aplica un sistema de control avanzado para mantener el equilibrio dinámico durante el desplazamiento. La propuesta integra sensores inerciales, tales como acelerómetros y giroscopios, que capturan información en tiempo real sobre la orientación y aceleración del robot. Estos datos son procesados por un microcontrolador que ejecuta algoritmos de control, como el PID, adaptados a las condiciones variables del terreno para garantizar una respuesta estable y rápida ante perturbaciones.

El diseño contempla tanto el modo autónomo, en el que el robot toma decisiones de manera independiente basándose en las lecturas de los sensores, como el modo controlado, que permite la intervención manual mediante comunicación inalámbrica. Esta dualidad funcional amplía las aplicaciones del dispositivo, facilitando su uso en entornos complejos y de difícil acceso.

Durante la fase experimental se realizaron pruebas en escenarios reales, identificando desafíos relacionados con la irregularidad de las superficies y la necesidad de ajustar los parámetros del sistema de control. Los resultados demostraron un desempeño satisfactorio en términos de estabilidad, maniobrabilidad y adaptabilidad. Este proyecto contribuye significativamente al avance de la robótica de equilibrio, ofreciendo un marco robusto para la movilidad autónoma en condiciones adversas y estableciendo las bases para futuras investigaciones y aplicaciones en el ámbito de la ingeniería mecatrónica. Este estudio ofrece perspectivas prometedoras para la innovación tecnológica real.

PALABRAS CLAVE:

- Robot balancín
- Péndulo invertido
- Control PID
- Movilidad autónoma
- Terrenos irregulares

ABSTRACT:

This thesis focuses on the design and implementation of an autonomous and controlled rocking robot, capable of moving on uneven terrain. The project is based on the inverted pendulum model, where an advanced control system is applied to maintain dynamic balance during movement. The proposal integrates inertial sensors, such as accelerometers and gyroscopes, which capture real-time information on the robot's orientation and acceleration. These data are processed by a microcontroller that executes control algorithms, such as the PID, adapted to the varying conditions of the terrain to ensure a stable and rapid response to disturbances.

The design contemplates both the autonomous mode, in which the robot makes decisions independently based on sensor readings, and the controlled mode, which

allows manual intervention through wireless communication. This functional duality expands the applications of the device, facilitating its use in complex and difficult-to-access environments.

During the experimental phase, tests were carried out in real scenarios, identifying challenges related to the irregularity of surfaces and the need to adjust the control system parameters. The results demonstrated satisfactory performance in terms of stability, maneuverability and adaptability. This project contributes significantly to the advancement of balancing robotics, offering a robust framework for autonomous mobility in adverse conditions and laying the foundations for future research and applications in the field of mechatronic engineering. This study offers promising prospects for real technological innovation.

KEYWORDS:

- * Rocking robot
- * Inverted pendulum
- * PID control
- * Autonomous mobility
- * Uneven terrain



Firma del Estudiante (AUTOR)

Nombre del Estudiante HENRY VACA

C.I.: 1719686279

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero

Telf: 0960052734 / 023524576 / 022331628

    www.ister.edu.ec / info@ister.edu.ec

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 14 de marzo del 2025

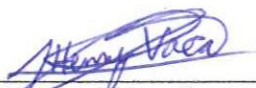
Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo **HENRY JAIR VACA CACERES** con C.I.:1719686279 alumno de la Carrera **ELECTRONICA**, cedo al **INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO**, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Firma del Estudiante

Nombre del Estudiante **HENRY VACA**

C.I.: 1719686279

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero

Telf: 0960052734 / 023524576 / 022331628

 www.ister.edu.ec / info@ister.edu.ec

DEDICATORIA

Dedico mi proyecto a mi madre y a mi padre que han estado conmigo en todo este largo proceso por el cual he pasado, he tenido muchos obstáculos y complicaciones en el trayecto de la carrera, pero ellos siempre estuvieron para apoyarme en lo que he necesitado y gracias a mi esfuerzo y constante dedicación es el resultado de este trabajo de titulación que sin su ayuda no hubiese sido posible lograrlo en cada paso que he dado me han estado brindando su apoyo y siempre me han dado ánimos para culminar con éxito mis estudios.

Vaca Cáceres Henry Jair.

AGRADECIMIENTO

Primeramente, doy gracias a Dios por permitirme llegar hasta este punto en mis estudios, también agradezco a la institución por brindarme su ayuda y conocimientos para poder poner en práctica, agradezco a mi tutor Ing., Cesar Minaya por guiarme en este proceso a la titulación y su constantes retroalimentación para cumplir con el objetivo y a mis padres, hermanos, familiares, amigos, compañeros. Deseo de todo corazón que todos los que estamos cursando por esta carrera la acabemos de manera satisfactoria.

Vaca Cáceres Henry Jair.

RESUMEN EJECUTIVO

Este proyecto de tesis aborda el diseño y construcción de un robot balancín, un sistema de control autónomo que mantiene el equilibrio sobre dos ruedas. El objetivo principal es desarrollar un prototipo funcional que aplique principios de control automático, mecatrónica e inteligencia artificial, con aplicaciones en educación, investigación y robótica móvil.

La metodología incluye el diseño mecánico del chasis, la selección de sensores (acelerómetros y giroscopios) para la estabilidad, la implementación de un microcontrolador para el procesamiento de datos y el desarrollo de un algoritmo de control basado en la teoría del péndulo invertido. Se utilizaron motores de corriente continua con control de velocidad para mantener la estabilidad y maniobrabilidad del robot.

Los resultados demostraron que el robot es capaz de equilibrarse de manera efectiva ante perturbaciones externas, validando el diseño del sistema de control. Sin embargo, se identificaron áreas de mejora, como la optimización del algoritmo de filtrado de señales y la integración de aprendizaje automático para mejorar la adaptabilidad del sistema.

En conclusión, este proyecto demuestra la viabilidad de un robot balancín como herramienta educativa y de investigación, sentando las bases para futuras mejoras en robótica de equilibrio y movilidad autónoma que servirá para competencias en el instituto.

PALABRAS CLAVE

Balancín, péndulo, microcontrolador, sensores, acelerómetro, PID, robótica, competencia, electrónica.



Ing. César Andrés Minaya Andino
TUTOR DEL TRABAJO DE TITULACIÓN

ABSTRACT

This thesis project addresses the design and construction of a self-balancing robot, an autonomous control system that maintains its equilibrium on two wheels. The main objective is to develop a functional prototype that applies principles of automatic control, mechatronics, and artificial intelligence, with applications in education, research, and mobile robotics.

The methodology includes the mechanical design of the chassis, the selection of sensors (accelerometers and gyroscopes) for stability, the implementation of a microcontroller for data processing, and the development of a control algorithm based on the inverted pendulum theory. DC motors with speed control were used to maintain the robot's stability and maneuverability.

The results demonstrated that the robot is capable of balancing effectively against external disturbances, validating the control system design. However, areas for improvement were identified, such as optimizing the signal filtering algorithm and integrating machine learning to enhance the system's adaptability.

In conclusion, this project demonstrates the feasibility of a self-balancing robot as an educational and research tool, laying the foundation for future improvements in balance robotics and autonomous mobility, which will be useful for competitions at the institute.

KEYWORDS

Self-balancing, pendulum, microcontroller, sensors, accelerometer, PID, robotics, competition, electronics.

ÍNDICE GENERAL

TODOS LOS DERECHOS RESERVADOS.....	iii
APROBACIÓN DEL TUTOR/A	iv
DEDICATORIA	vi
AGRADECIMIENTO	xi
RESUMEN EJECUTIVO	xii
PALABRAS CLAVE	xiii
ABSTRACT.....	xiv
KEYWORDS.....	xv
ÍNDICE DE FIGURAS.....	xiv
1. INTRODUCCIÓN	1
1. CAPÍTULO I	3
1.1. Planteamiento del problema	3
1.2. Preguntas de directrices.....	5
1.3. Objetivo General	5
1.4. Objetivos Especifico	6
1.5. Justificación.....	6
1.6. Alcance.....	7
2. CAPÍTULO II	8
2.1. Marco Teórico	8
2.1.1. Péndulo invertido	8
2.1.2. Arduino Nano.....	10
2.1.3. Driver Puente H TB6612FNG	12
2.1.4. Módulo Acelerómetro y Giroscopio MPU6050	14
2.1.5. Metal Gearmotor 37DX57L.....	15
2.1.6. Sensor Ultrasónico HC-SR04	17
3. CAPÍTULO III.....	20

3.1.	Fases de diseño.....	22
3.1.1.	Etapa 1: Estudio previo inicial.....	22
3.1.2.	Etapa 2: Diseño de la estructura mecánica y electrónica.....	22
3.1.3.	Etapa 3: Diseño del sistema de control y simulación	22
3.2.	Fase de implementación.....	22
3.2.1.	Etapa 1: Fabricación de la estructura mecánica.....	22
3.2.2.	Etapa 2: Ensamblaje de la tarjeta electrónica y sensores.....	23
3.2.3.	Etapa 3: Programación del sistema de control.....	23
3.2.4.	Etapa 4: Integración y pruebas iniciales	23
3.3.	Fases de evaluación.....	23
3.3.1.	Etapa 1: Pruebas de funcionamiento general.....	23
3.3.2.	Etapa 2: Prueba de comunicación y respuesta del sistema	24
3.3.3.	Etapa 3: Pruebas de movimiento y estabilidad	24
3.3.4.	Etapa 4: Evaluación según reglas de competencia	24
4.	CAPÍTULO IV.....	26
4.1.	Diseño de la propuesta	26
4.1.1.	Diagrama de flujo robot balancín	26
4.2.	Diseño 3D para piezas estructurales de robot balancín.....	27
4.2.1.	Plataforma Tinkercad Diseño 3D.....	28
4.2.2.	Diseño modular en Tinkercad.....	28
4.2.3.	Piezas para robot balancín.....	28
4.2.4.	Plataforma Tinkercad Diseño de Circuitos	40
4.2.5.	Diseño circuito robot balancín	40
4.2.6.	Control PID	43
5.	CAPITULO V.....	49
5.1.	Conclusiones	49
5.2.	Recomendaciones.....	49

5.3. Referencias	50
6. Referencias.....	50
7. ANEXOS	52

Índice de Figuras

Figura 1	Sistema de péndulo invertido	8
Figura 2	<i>Sistema de péndulo invertido sobre carrito</i>	<i>9</i>
Figura 3	<i>Arduino nano original</i>	<i>10</i>
Figura 4	<i>Descripción de pines parte superior Arduino Nano Original</i>	<i>11</i>
Figura 5	<i>Descripción de pines parte inferior Arduino Nano Original</i>	<i>12</i>
Figura 6	<i>Descripción de pines driver puente H TB6612FNG</i>	<i>13</i>
Figura 7	<i>Módulo acelerómetro y giroscopio MPU6050</i>	<i>14</i>
Figura 8	<i>Representación tridimensional módulo MPU6050</i>	<i>15</i>
Figura 9	<i>Motor a pasos metal gearmotor 100:1</i>	<i>16</i>
Figura 10	<i>Descripción detallada de medidas metal gearmotor 100.1</i>	<i>17</i>
Figura 11	<i>Sensor Ultrasónico HC-SR04</i>	<i>18</i>

INTRODUCCIÓN

El presente trabajo de titulación se centra en el diseño, desarrollo e implementación de un robot balancín, un sistema basado en el concepto de péndulo invertido, cuya estabilidad depende de un sistema de control eficiente. Este proyecto integra conocimientos de control automático, electrónica y mecánica, con el objetivo de construir un prototipo funcional que pueda mantenerse en equilibrio y responder a distintas condiciones operativas mediante un controlador Proporcional-Integral-Derivativo (PID).

En el Capítulo I, se presenta el planteamiento del problema, donde se expone la importancia del estudio de sistemas dinámicos inestables en la ingeniería. Se definen los objetivos generales y específicos del proyecto, así como la justificación, que destaca la relevancia de este desarrollo en el ámbito educativo y en la automatización. Finalmente, se delimita el alcance del proyecto, especificando sus limitaciones y aplicaciones.

El Capítulo II abarca el fundamento teórico, proporcionando una base conceptual sobre el control PID, las características de los sistemas de retroalimentación y los principios físicos y matemáticos que rigen el comportamiento del robot balancín. También se analizan distintas estrategias de control y sus aplicaciones en sistemas dinámicos similares.

En el Capítulo III, se detalla la metodología empleada, describiendo los tipos de investigación utilizados para el desarrollo del proyecto. Se explican los métodos teóricos y experimentales aplicados, así como las técnicas e instrumentos empleados para la recolección y análisis de datos, garantizando un enfoque sistemático en la validación del diseño.

El Capítulo IV está dedicado al diseño, implementación y pruebas del prototipo. Se describe el desarrollo de la estructura mecánica y del circuito electrónico en un entorno de simulación, junto con la selección de los componentes más adecuados. Luego, se explica la implementación del algoritmo de control en un sistema embebido y el proceso de ensamblaje del robot. Finalmente, se presentan las pruebas experimentales realizadas para evaluar la estabilidad y desempeño del sistema.

En el Capítulo V, se presentan las conclusiones y recomendaciones obtenidas a partir de los resultados experimentales. Se analizan los logros alcanzados en comparación con los objetivos planteados y se sugieren mejoras para futuras versiones del robot balancín.

El documento finaliza con la sección de referencias, donde se citan las fuentes bibliográficas utilizadas, y los anexos, que incluyen información complementaria relevante para el desarrollo del proyecto.

Este trabajo no solo busca demostrar la viabilidad técnica del robot balancín, sino también servir como un recurso educativo y de consulta para estudiantes y profesionales interesados en el campo de la automatización y el control de sistemas dinámicos.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

En este capítulo se presenta el problema de investigación que da origen al desarrollo del proyecto. Se describe la importancia de estudiar sistemas de automatización como el robot balancín, analizando los desafíos técnicos que implica su diseño y control. Asimismo, se justifica la relevancia de esta investigación dentro del ámbito educativo y tecnológico, destacando su aplicación en el aprendizaje de control automático y robótica. Finalmente, se delimita el alcance del proyecto, definiendo las limitaciones y consideraciones necesarias para su desarrollo.

1.1. Planteamiento del problema

A lo largo del tiempo en la búsqueda de la automatización de procesos la teoría del control ha desarrollado grandes alternativas deterministas y heurísticas; un claro ejemplo es el control PID y el control por retroalimentación de estado, y las heurísticas las redes neuronales y lógica difusa. La eficiencia del control determina la evolución en el desarrollo de nuevos prototipos de automatización y esto parte de la base de realizar una correcta implementación del PID. Por lo general, se piensa que el tipo de automatización que se emplea no es muy eficiente debido a la tendencia de aplicar sistemas de control clásicos, por lo cual, surge la necesidad del estudio y aplicación de técnicas modernas en sistemas de control.

En una publicación de la revista Visión Electrónica, Universidad Francisco José de Celdas de Bogotá señala que la tecnología ha tenido un gran impacto en la sociedad, transformando rápidamente la forma en que se realizan diversas tareas. Por ejemplo, el robot balancín, basado en el principio del péndulo invertido, puede desplazarse sobre dos ruedas y realizar las mismas funciones que un robot humanoide (José, 2024). Por todo lo mencionado anteriormente es necesario modelar, simular y construir este tipo de proyectos donde se estudia retroalimentación de estados, y el control automático.

Durante el estudio de la formación técnica y tecnológica en el campo de la robótica, resulta esencial disponer de recursos didácticos que faciliten la comprensión tanto de los principios teóricos como prácticos de los sistemas de control. Sin embargo, en la actualidad existe una escasez notable de recursos educativos apropiados y fácilmente asequibles que abarquen todos los conceptos y aplicaciones de robótica.

Aunque los simuladores pueden ser útiles, en muchos casos no logran representar de manera exacta las complejidades presentes en los entornos reales de robótica y sus competencias. Sin embargo, a pesar de ser ilustrativos, los videos no brindan la interacción necesaria para un aprendizaje profundo. El proceso de enseñanza-aprendizaje se ve dificultado en este campo debido a las deficiencias presentes, lo cual afecta negativamente la adquisición de competencias por parte de los estudiantes. Según los resultados obtenidos por Rodríguez y Sánchez (2021), aproximadamente el 70% de los estudiantes reporta enfrentar dificultades al aplicar sus conocimientos teóricos debido a la falta de recursos adecuados.

En la actualidad, el desarrollo de habilidades prácticas esenciales para los estudiantes se ve limitado debido a la falta de recursos interactivos. Esto dificulta su capacidad para aplicar conocimientos teóricos en situaciones reales. Por otra parte, los materiales educativos actuales no se ajustan de manera flexible a los distintos estilos y necesidades de aprendizaje que tienen los estudiantes. Debido a su falta de adaptabilidad, los educadores se encuentran restringidos en su capacidad para personalizar la enseñanza y mejorar tanto la comprensión como el rendimiento de cada estudiante. En entornos educativos diversos y en constante evolución, esta situación se torna realmente desafiante debido a las variadas necesidades de los estudiantes.

Es esencial para la formación adecuada de profesionales competentes y bien preparados contar con laboratorios equipados con equipos modernos y funcionales. Frente a esta realidad, las instituciones educativas de América Latina se ven confrontadas con el reto de modernizar su material de estudio. Según la CEPAL (2020), con el objetivo de mejorar la calidad educativa y aumentar la competitividad en la región, se requiere actualizar los equipos del laboratorio e incorporar tecnologías de última generación.

En Ecuador, esta situación es fácilmente observable. A pesar de los esfuerzos del Instituto Superior Tecnológico Rumiñahui (ISTER) por brindar una educación de excelencia, la tecnología evoluciona constantemente y se reconoce la apremiante necesidad de renovar su equipamiento de laboratorio en el campo de la robótica.

El uso de un robot balancín como herramienta de aprendizaje y como robot para competencias se justifica por su capacidad para integrar conocimientos teóricos y prácticos en el campo de la robótica. Este sistema no solo ofrece a los estudiantes de tecnología y áreas afines una experiencia educativa enriquecedora y efectiva, sino que también mejora significativamente la

interactividad en el proceso de enseñanza-aprendizaje al simular situaciones reales de control y adaptarse a diversos entornos didácticos.

En resumen, el robot balancín no solo es una herramienta innovadora para el aprendizaje y la investigación, sino también un recurso estratégico que fortalece la formación técnica, su implementación representa una inversión en el futuro de los estudiantes, la institución y la sociedad en general. Esto les permitirá prepararse de una manera más efectiva para enfrentar exitosamente los desafíos del mercado laboral actual. Asimismo, al mejorar la infraestructura educativa del ISTER, el robot balancín contribuirá a consolidar la posición de la institución como referente en excelencia académica tanto a nivel local como nacional.

1.2. Preguntas de directrices

¿Cuáles son los principios que rigen el comportamiento de un robot balancín basado en un péndulo invertido?

¿Qué tipo de sensores y actuadores son los más adecuados para garantizar la estabilidad del robot balancín?

¿Cómo se puede implementar un controlador PID para optimizar la estabilidad del robot balancín en diferentes condiciones operativas?

¿Qué herramientas de simulación pueden utilizarse para predecir y analizar el comportamiento del robot antes de su implementación física?

¿Qué materiales y componentes electrónicos son los más adecuados para la construcción del prototipo, considerando factores como costo, disponibilidad y eficiencia?

¿Cuáles son las principales dificultades técnicas que pueden surgir durante la implementación del robot balancín y cómo pueden resolverse?

¿De qué manera este proyecto puede servir como material de consulta e investigación para estudiantes de electrónica y control automático?

1.3. Objetivo General

Diseñar y construir un prototipo de robot balancín estabilizado mediante dos ruedas de reacción accionadas por motores eléctricos, desarrollando e implementando estrategias de control que optimicen su desempeño, para utilizarlo como herramienta didáctica en la enseñanza de

robótica y como plataforma para competencias, promoviendo la aplicación práctica de conceptos de automatización y control.

1.4. Objetivos Especifico

- Diseñar la estructura mecánica y el circuito electrónico del robot balancín en un entorno de simulación, seleccionando los componentes óptimos para su implementación.
- Implementar y optimizar un algoritmo de control PID en un sistema embebido para monitorear y regular la estabilidad del robot balancín, garantizando su funcionamiento eficiente.
- Ensamblar el robot balancín y realizar pruebas experimentales para evaluar su estabilidad, respuesta dinámica y comportamiento ante diferentes condiciones operativas.

1.5. Justificación

El desarrollo de un robot balancín como herramienta de aprendizaje y competencia responde a la necesidad de integrar conceptos teóricos y prácticos en áreas clave como la robótica y la electrónica. Este proyecto surge como una iniciativa para modernizar los métodos de enseñanza, proporcionando a los estudiantes una experiencia educativa más inmersiva y aplicada.

A través de un enfoque que combina el diseño, la simulación, la programación y la implementación práctica, se utilizarán herramientas de simulación, estructura mecánica, componentes electrónicos adecuados y se programará un sistema para monitorear y controlar su estabilidad. Posteriormente, se realizarán pruebas y análisis de comportamiento para optimizar su funcionalidad.

Este proyecto no solo fortalecerá el aprendizaje en la asignatura de robótica, sino que también se convertirá en un recurso valioso para competencias, permitiendo a los estudiantes aplicar sus conocimientos en escenarios reales y representar a la institución en eventos académicos y tecnológicos. Además, contribuirá a la modernización de los laboratorios, fomentará la investigación en sistemas de control no lineales y preparará a los estudiantes para enfrentar los desafíos del mercado laboral actual, impulsando el desarrollo tecnológico y educativo.

Por lo antes descrito este proyecto no solo servirá como una herramienta didáctica para fortalecer el aprendizaje en la asignatura de robótica, sino que también se convertirá en un recurso para competencias de robótica, donde los estudiantes podrán aplicar sus conocimientos y representar a la institución. Además, contribuirá a la modernización de los laboratorios, fomentará la investigación en sistemas de control no lineales y preparará a los estudiantes para enfrentar los desafíos del mercado laboral actual, promoviendo así el desarrollo tecnológico y educativo.

1.6. Alcance

El proyecto abarcara el diseño, construcción y puesta en funcionamiento de un robot balancín capaz de moverse de manera autónoma y controlada mediante un aplicativo. El prototipo será estabilizado mediante dos ruedas de reacción accionadas por motor-reductores eléctricos, e implementará un sistema de control PID que permitirá estabilizar el robot, evitando que se caiga o choque, y garantizando su maniobrabilidad. Este sistema de control será programado y probado en entornos de simulación antes de su implementación en el prototipo físico, asegurando un funcionamiento óptimo.

Además, en esta etapa de construcción se llevará a cabo la selección y adquisición de los componentes adecuados, considerando factores como eficiencia, durabilidad y costo, para garantizar la viabilidad del prototipo. Una vez construido, se realizarán pruebas experimentales para evaluar su desempeño en términos de estabilidad y respuesta a perturbaciones externas.

Finalmente, el proyecto busca cooperar en el ámbito académico al convertirse en una herramienta didáctica para los estudiantes de robótica del Instituto Superior Tecnológico Rumiñahui. Asimismo, se espera que el prototipo pueda ser utilizado en futuras competencias de robótica, lo que ampliará su impacto y contribuirá al posicionamiento de la institución en eventos tecnológicos de relevancia.

CAPÍTULO II

MARCO DE REFERENCIAS

2.1. Marco Teórico

El desarrollo de sistemas robóticos autónomos ha avanzado significativamente en los últimos años. Dentro de este campo, los robots de equilibrio dinámico, como el "Robot Balancín", representan un desafío tanto en términos de estabilidad como de control en entornos dinámicos. El presente marco teórico abordará los fundamentos más importantes para la comprensión del sistema, incluyendo, la teoría de control aplicada a sistemas robóticos, y el uso de sensores y actuadores para la estabilización del robot. Además, se explorarán modelos de control como el PID, el cual es la base teórica para este tipo de dispositivos.

2.1.1. Péndulo invertido

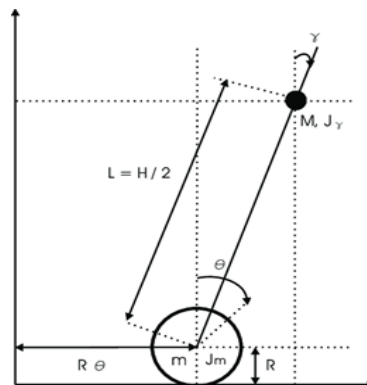
Para Luis Triviño en su trabajo menciona que, el problema de control que supone un sistema de péndulo invertido consiste en el equilibrio de un centro de masas que se sitúa por encima de un punto de pivotaje (ver Figura 1). Mientras que el sistema de péndulo clásico tiende a la estabilidad cuando este se encuentra en posición vertical, el péndulo invertido es un sistema inherentemente inestable y que debe ser controlado activamente (Macías, 2020).

Figura 1

Sistema de péndulo invertido

Figura 1

Sistema de péndulo invertido



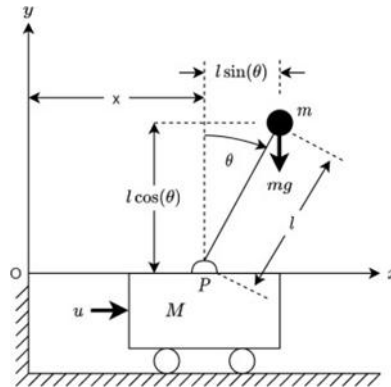
Nota: Representación con fuerzas y vectores de un péndulo invertido. Tomado de: Modelado, simulación y control de un péndulo invertido, por Luis Geovanny Triviño Macías, 2020, UA.

El problema clásico consiste en anclar una vara a un eje en un carro, lo cual restringe el movimiento de la vara a un grado de libertad (ver Figura 2).

Sistema de péndulo invertido sobre carrito

Figura 2

Sistema de péndulo invertido sobre carrito



Nota: Se muestra la complejidad del sistema de un péndulo invertido. Tomado de: Modelado, simulación y control de un péndulo invertido, por Luis Geovanny Triviño Macías, 2020, UA.

Existen variaciones del problema que se extienden a dos grados de libertad, tradicionalmente mediante una bola situada debajo del sistema y controlada en todas direcciones. Adicionalmente, el problema de control del péndulo invertido ha servido como base para el desarrollo del control de sistemas más complejos. En la conferencia internacional de robots inteligentes y sistemas de 2010 se presentó un control de estabilización de postura de un robot bípedo basado en un modelo de péndulo invertido lineal. En este robot HRP-4C, desarrollado en el Instituto Nacional de Ciencias Industriales Avanzadas y Tecnología de Japón, el cuerpo se modela como un péndulo invertido. De esta manera, se controla la orientación del cuerpo de la misma forma que se controla el ángulo del péndulo invertido mediante un controlador de espacio de estados basado en asignación de polos (Calvo, 2022).

Las estrategias para controlar el problema del péndulo invertido clásico consisten en ejercer un torque en el punto de pivote, desplazar dicho punto de forma horizontal y perpendicular al eje para inducir un torque en el péndulo, o hacer que oscile verticalmente, lo que también genera un torque en el pivote. Un ejemplo típico de control mediante el movimiento del punto de pivote es el acto de equilibrar una escoba o un martillo sobre la palma de la mano.

2.1.2. Arduino Nano

Es un microcontrolador didáctico y es uno de los modelos más compactos de la familia de Arduino. Como se puede ver en la figura 3, es uno de los más replicados por su versatilidad y aplicaciones educativas. Se basa en el microcontrolador ATmega328, aunque en el mercado los clones de baja calidad todavía utilizan el modelo inferior ATmega168 (Gonzalez, 2024).

Figura 3

Arduino nano original



Nota. Microcontroladores más compactos de la familia Arduino. Tomada de Guía de modelos Arduino y sus características | BricoGeek Lab, por Oscar Gonzales, 2024, bricogeek.com.

Podemos afirmar que es una versión más compacta del Arduino UNO ya que utiliza el mismo microcontrolador ATmega328 y por ende posee las mismas funcionalidades. La mayor diferencia si lo comparamos con un Arduino UNO es su tamaño, mucho más reducido y debido a su formato, permite colocarlo directamente sobre una placa de pruebas o también conocido como protoboard (Gonzalez, 2024). En la figura 4 y 5 podemos ver la descripción de sus pines. Además, puede integrarse fácilmente en cualquier proyecto educativo, didáctico, comercial o incluso en placas PCB personalizadas.

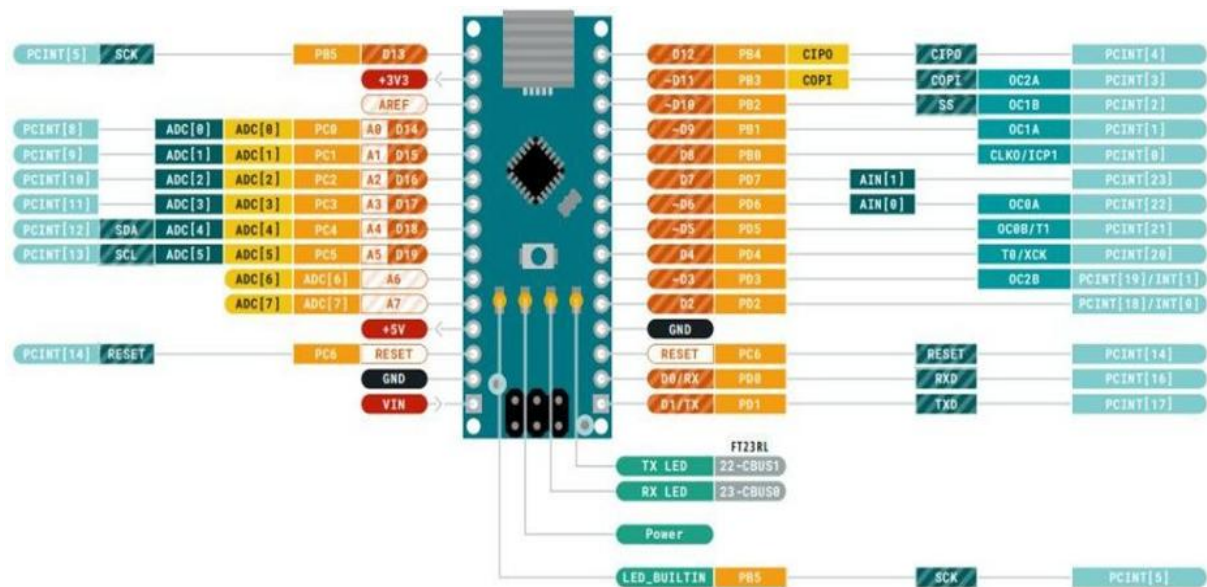
Características del Arduino Nano

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Memoria FLASH: 32 KB (2 KB usados por el bootloader)
- SRAM: 2 KB

- Velocidad de reloj: 16 MHz
- Pines analógicos: 8
- EEPROM: 1 KB
- Corriente máxima por pin: 40 mA
- Entrada de alimentación: 7 a 12 V
- Pines digitales: 22 (6 con soporte PWM)
- Salidas PWM: 6
- Consumo: 19 mA
- Tamaño: 18 x 45 mm
- Peso: 7 g
- Puerto entrada datos y voltaje: USB en tipo C, tipo micro USB, tipo mini USB

Figura 4

Descripción de pines parte superior Arduino Nano Original

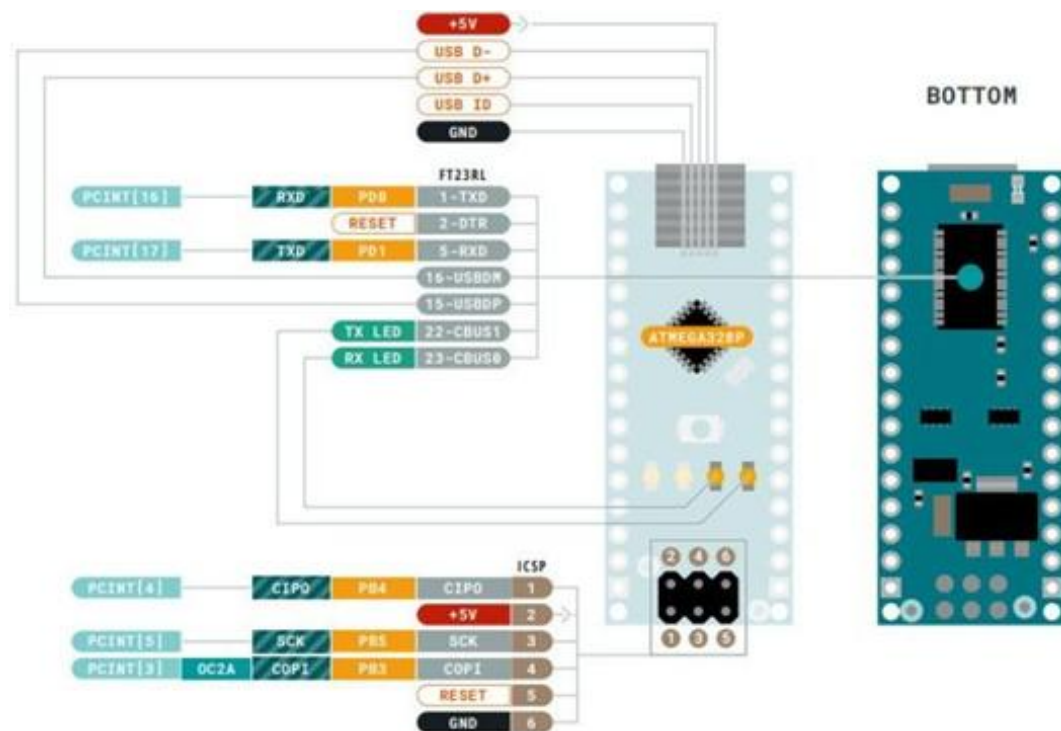


Nota. Microcontroladores más compactos de la familia Arduino. Tomada de Guía de modelos

Arduino y sus características | BricoGeek Lab, por Oscar Gonzales, 2024, bricogeek.com.

Figura 5

Descripción de pines parte inferior Arduino Nano Original



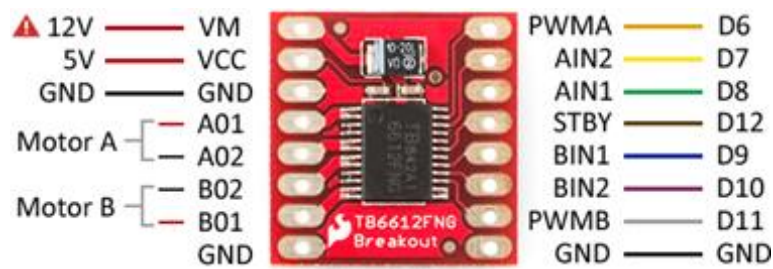
Nota. Microcontroladores más compactos de la familia Arduino. Tomada de Guía de modelos Arduino y sus características | BricoGeek Lab, por Oscar Gonzales, 2024, bricogeek.com.

2.1.3. Driver Puente H TB6612FNG

El driver para motores TB6612FNG posee dos puentes H, y puede controlar hasta dos motores DC con corriente constante de 1.2A (3.2A pico). Posee dos señales de entrada (IN1 y IN2) que pueden ser utilizadas para controlar el motor en uno de los cuatro modos posibles: CW(giro en sentido de las manecillas del reloj), CCW (en contra de las manecillas), short-brake y stop. Las dos salidas de motores (A y B) pueden ser controladas de manera separada, la velocidad de cada motor es controlada mediante una señal PWM con una frecuencia de hasta 100kHz. El pin STBY cuando es puesto en HIGH coloca al motor en modo de standby. El driver posee diodos internos de protección (Novatronic, 2020).

Figura 6

Descripción de pines driver puente H TB6612FNG



Nota. Describe a detalle cada pin del módulo. Tomada de TB6612FNG Driver para motor DC, por Novatronic, 2020, novatronicec.com

El módulo también permite controlar un motor paso a paso unipolar o bipolar, que en nuestro caso es la funcionalidad que utilizaremos para nuestros motores. El módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi y Launchpads de Texas Instruments.

Características Driver Puente H TB6612FNG

- Marca: Genérico (módulos)
- Modelo: MOD0077
- Basado en circuito integrado: TB6612FNG
- Tipo de módulo: Puente H
- Número de puentes H: 2
- Voltaje de entrada para motores: 2.5 a 13.5 VDC
- Voltaje de entrada para lógica: 2.7 a 5.5 VDC
- Capacidad nominal por puente: 1 A (ver data sheet)
- Capacidad pico por puente: 3.2 A (ver data sheet)
- Resistencia interna MOSFET: 0.5Ω (alto + bajo)
- Frecuencia de conmutación: 100 KHz
- Tecnología de fabricación: LDMOS
- Modo de “apagado” de bajo consumo

- Protección térmica y de bajo voltaje
- Pines dedicados para control PWM
- Peso: 3 gramos

Dimensiones:

- o Largo: 4 cm
- o Ancho: 3 cm
- o Alto: 0.5 cm

2.1.4. Módulo Acelerómetro y Giroscopio MPU6050

Este módulo de sensor mostrado en la figura 7, es una unidad de medición inercial o IMU (Inertial Measurement Units) de 6 grados de libertad (DoF) pues combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Este sensor es muy utilizado en navegación, goniometría, estabilización, etc (Naylampmechatronics, 2024).

Figura 7

Módulo acelerómetro y giroscopio MPU6050



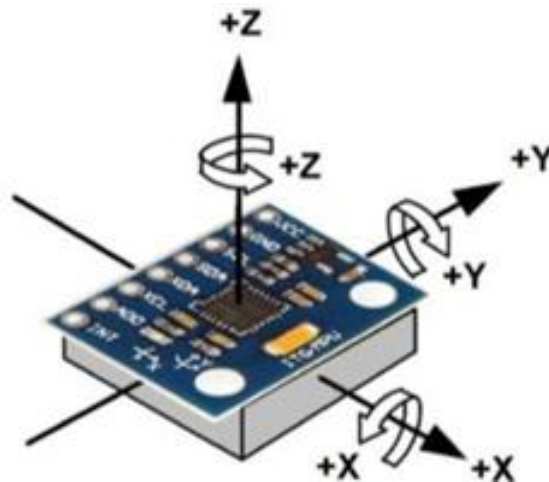
Nota. Módulo MPU6050 original. Tomada de: Tutorial MPU6050, Acelerómetro y Giroscopio, por Naylamp Mechatronics, 2024, naylampmechatronics.com

El módulo Acelerómetro MPU incluye un giroscopio de tres ejes que permite medir la velocidad angular, así como un acelerómetro de tres ejes que facilita la medición de las componentes X, Y y Z de la aceleración. Como se observa en la figura 8, la orientación de los

ejes está claramente señalada en el módulo, por lo que es fundamental prestar atención para evitar errores en la interpretación de los signos de las aceleraciones.

Figura 8

Representación tridimensional módulo MPU6050



Nota. Módulo MPU6050 original. Tomada de: Tutorial MPU6050, Acelerómetro y Giroscopio, por Naylamp Mechatronics, 2024, naylampmechatronics.com

Características Módulo Acelerómetro y Giroscopio MPU6050

- Chip: MPU-6050
- Fuente de alimentación: 3V – 5V
- Rango del giroscopio: ± 250 ± 500 ± 1000 ± 2000 $^{\circ} / s$
- Rango de aceleración: ± 2 ± 4 ± 8 ± 16 g
- Comunicación: acuerdo de comunicación estándar de la CII
- Convertidor AD de 16 bits integrado en chip, salida de datos de 16 bits
- Distancia entre pines 2,54 mm
- Comunicación: I2C

2.1.5. Metal Gearmotor 37DX57L

El motor a utilizar es el Gearmotor 37DX57L, mismo que es comercializado por la empresa Electrostore empresa, menciona que este motorreductor es un potente motor DC cepillado con 102,08: 1 caja de engranajes de metal destinado para funcionar a 12 V . La caja de cambios

está compuesta principalmente de engranajes rectos, pero cuenta con engranajes helicoidales para la primera etapa para reducir el ruido y mejorar la eficiencia. Estas unidades tienen un eje de salida en forma de D de 16 mm de largo y 6 mm de diámetro como se observa en la figura 10. Este motorreductor también está disponible con encoder integrado (Electrostore, 2024).

Figura 9

Motor a pasos metal gearmotor 100:1



Nota. Motor de la familia Pololu diseñado para uso educativo, comercial e industrial. Tomada de METAL GEARMOTOR 37DX57L, por Electrostore, 2024, electrostore.com/shop

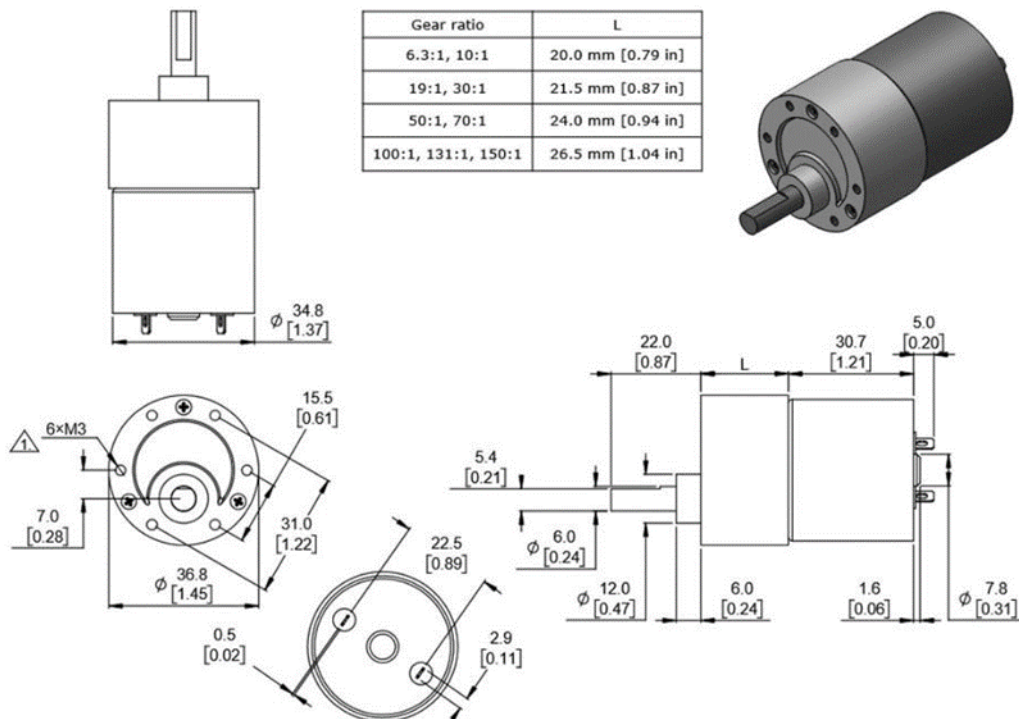
Características Metal Gearmotor 37DX57L

- Fabricante: POLOLU
- Tipo de motor: DC
- Clase de motor: con engranaje
- Corriente de trabajo máximo: 5,5A
- Superficie de eje: recorte D
- Rotaciones nominales medidas tras engranaje: 100rot./min
- Momento de rotación: max. 412mNm
- Engranaje: 102,08:1
- Diámetro de cilindro: 6mm
- Diámetro: 36,8mm
- Dimensiones de cuerpo: Ø36,8x68,2mm
- Tensión de alimentación: 12V DC
- Peso bruto: 195.78 g

- Medidas: ver ilustración 11

Figura 10

Descripción detallada de medidas metal gearmotor 100.1



Nota. Motor de la familia Pololu diseñado para uso educativo, comercial e industrial. Tomada de METAL GEARMOTOR 37DX57L, por Electrostore, 2024, electrostore.com/shop

2.1.6. Sensor Ultrasónico HC-SR04

El HC-SR04 es un sensor de distancia de bajo costo y didáctico que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. De igual forma es el más empleado en proyectos de robótica como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.

Figura 11

Sensor Ultrasónico HC-SR04



Nota. Sensor de los más comerciales en la industria de aplicaciones educativas. Tomada de Sensor ultrasonido HC-SR04 Módulo MPU6050, por Naylamp Mechatronics, 2024, naylampmechatronics.com

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación. El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido (40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar).

La distancia se puede calcular utilizando la siguiente formula:

$$\text{Distancia(m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\} / 2$$

El sensor US-016 es similar al HC-SR04, pero con salida de tipo analógico, otro sensor ultrasonido es el sensor US-100 con salida de tipo uart/serial.

Características Sensor Ultrasónico HC-SR04

- Voltaje de Operación: 5V DC
- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Rango de medición: 2cm a 450cm

- Precisión: $\pm 3\text{mm}$
- Ángulo de apertura: 15°
- Frecuencia de ultrasonido: 40KHz
- Duración mínima del pulso de disparo TRIG (nivel TTL): $10\ \mu\text{S}$
- Duración del pulso ECO de salida (nivel TTL): $100\text{-}25000\ \mu\text{S}$
- Dimensiones: $45*20*15\ \text{mm}$
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)

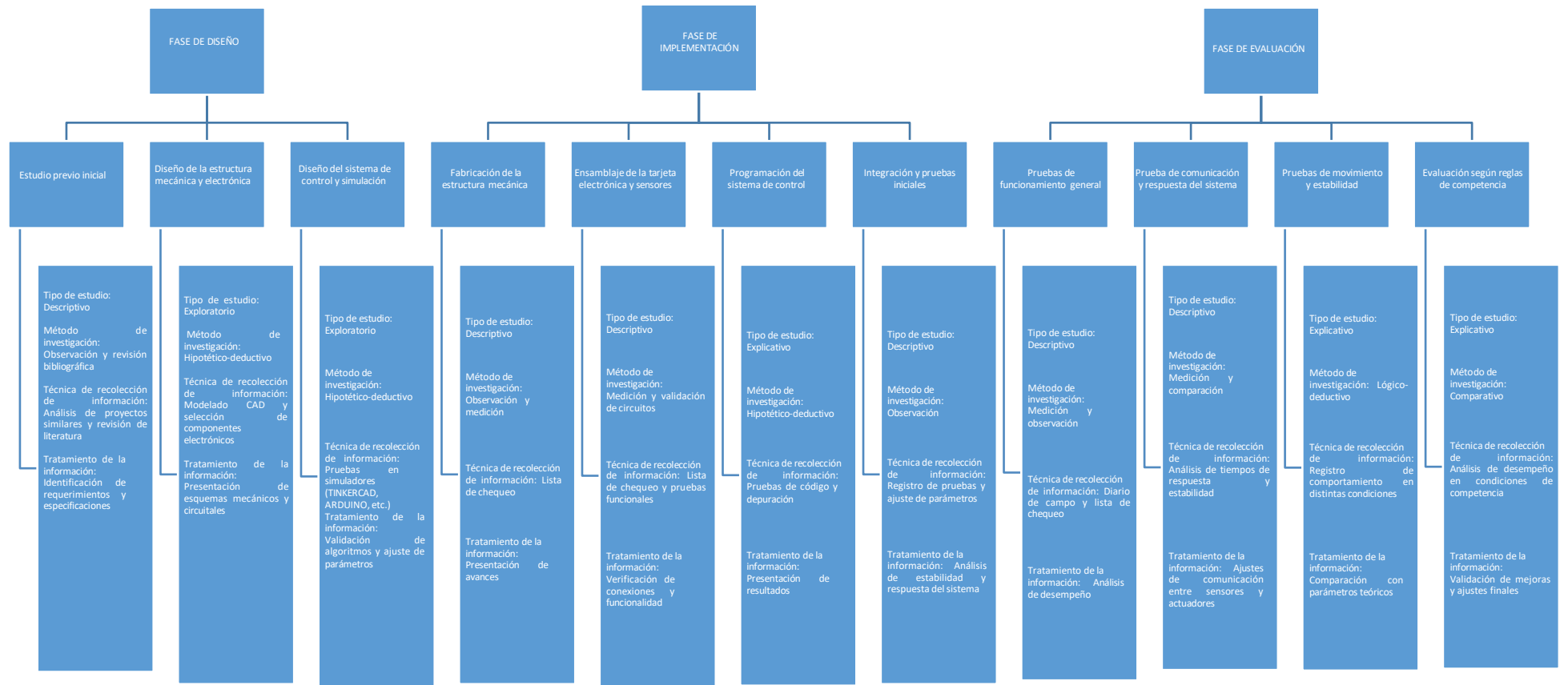
CAPÍTULO III

MARCO METODOLÓGICO

En esta investigación, se emplea fases y etapas para analizar y optimizar el desempeño del robot balancín en términos de estabilidad y control. Inicialmente, las fases de diseño, implementación y evaluación permiten observar que el robot enfrenta dificultades para mantener el equilibrio de manera estable ante diferentes perturbaciones externas. A partir de esta observación, surge la pregunta de investigación sobre cómo la mejora en los algoritmos de control y la calibración de los sensores afectan la estabilidad del robot.

Para abordar esta cuestión, se plantea la hipótesis de que la implementación de estrategias avanzadas de control, junto con una calibración precisa de los sensores, permitirá mejorar la capacidad del robot para mantener el equilibrio de forma autónoma y responder eficientemente a perturbaciones. El análisis de los resultados obtenidos permitirá validar o refutar la hipótesis, proporcionando información clave para optimizar el desempeño del robot balancín y mejorar su capacidad de adaptación en distintas condiciones operativas.

DIGRAMA DE FASES Y ETAPAS



3.1. Fases de diseño

3.1.1. Etapa 1: Estudio previo inicial

Por medio de este estudio descriptivo, se realizará un análisis previo de los requerimientos técnicos y operativos necesarios para el desarrollo del prototipo de robot balancín estabilizado. Para ello, se empleará el método de observación y la revisión bibliográfica, con el objetivo de recopilar información relevante sobre estrategias de control, estabilidad y hardware adecuado. La información obtenida permitirá establecer las bases para la selección de materiales, sensores y microcontroladores óptimos para el sistema.

3.1.2. Etapa 2: Diseño de la estructura mecánica y electrónica

Por medio de este estudio exploratorio, se llevará a cabo el diseño de la estructura mecánica y del sistema electrónico del robot balancín, definiendo la disposición de sus componentes y el ensamblaje de su chasis. Se empleará el método hipotético-deductivo, y la técnica de modelado CAD para la parte mecánica, así como el desarrollo de esquemas eléctricos para la interconexión de sensores, motores y la unidad de control. La información obtenida se documentará a través de listas de chequeo y permitirá establecer un diseño funcional y optimizado.

3.1.3. Etapa 3: Diseño del sistema de control y simulación

Por medio de este estudio exploratorio, se desarrollará y evaluará el sistema de control del robot balancín a través de simulaciones en software especializado, como MATLAB o Proteus. Se empleará el método hipotético-deductivo, permitiendo validar distintos algoritmos de control que garanticen la estabilidad y el desempeño óptimo del robot. La información será recopilada mediante pruebas en entornos virtuales, y se presentará a través de registros de simulación y análisis comparativos de estabilidad.

3.2. Fase de implementación

3.2.1. Etapa 1: Fabricación de la estructura mecánica

Por medio de este estudio descriptivo, se procederá a la construcción física de la estructura mecánica del robot balancín, asegurando que cumpla con las especificaciones establecidas en la fase de diseño. Para ello, se utilizará el método de observación y la técnica de lista de chequeo,

verificando la correcta fabricación de las piezas y su ensamblaje. Se evaluará la resistencia y compatibilidad de los materiales seleccionados, documentando cada ajuste realizado.

3.2.2. Etapa 2: Ensamblaje de la tarjeta electrónica y sensores

Por medio de este estudio descriptivo, se llevará a cabo la fabricación y ensamblaje de la tarjeta electrónica que integrará los sensores, actuadores y el microcontrolador. Se utilizará el método de medición y la técnica de lista de chequeo para verificar la correcta instalación y conexión de los componentes electrónicos. Se registrará el funcionamiento individual de cada módulo, asegurando que cumpla con los requisitos de control y estabilidad.

3.2.3. Etapa 3: Programación del sistema de control

Por medio de este estudio explicativo, se desarrollará el código del microcontrolador, implementando los algoritmos de control que permitan la estabilidad del robot balancín. Se empleará el método hipotético-deductivo y la técnica de lista de chequeo para verificar la correcta ejecución del software. Se documentará el proceso de programación, incluyendo la depuración de errores y los ajustes en los parámetros de control, asegurando la eficiencia del sistema

3.2.4. Etapa 4: Integración y pruebas iniciales

Por medio de este estudio descriptivo, se procederá a la integración de la estructura mecánica, el sistema electrónico y el software de control, verificando el correcto funcionamiento del robot balancín. Se empleará el método de observación y la técnica de registro de pruebas para analizar la estabilidad y respuesta del sistema ante distintos escenarios de prueba. La información recopilada permitirá realizar ajustes finales antes de las evaluaciones más exigentes.

3.3. Fases de evaluación

3.3.1. Etapa 1: Pruebas de funcionamiento general

Por medio de este estudio descriptivo, se evaluará el desempeño general del robot balancín, verificando que todos sus subsistemas operen de manera integrada. Se empleará el método de medición y la técnica de diario de campo para registrar el comportamiento del robot en distintos entornos de prueba. Se analizará la estabilidad, la respuesta a perturbaciones y la eficiencia del control implementado.

3.3.2. Etapa 2: Prueba de comunicación y respuesta del sistema

Por medio de este estudio descriptivo, se validará la comunicación entre los sensores, actuadores y la unidad de control del robot. Se empleará el método de medición y la técnica de lista de chequeo para verificar la correcta transmisión de datos y la respuesta del sistema en tiempo real. Se documentarán posibles fallos y se implementarán mejoras en la comunicación interna del robot.

3.3.3. Etapa 3: Pruebas de movimiento y estabilidad

Por medio de este estudio explicativo, se analizará el desempeño del robot balancín en distintas condiciones de operación, evaluando su estabilidad y capacidad de respuesta a cambios en el entorno. Se empleará el método lógico-deductivo y la técnica de registro de comportamiento, documentando datos sobre la inclinación, corrección de balance y eficiencia del sistema de control.

3.3.4. Etapa 4: Evaluación según reglas de competencia

Por medio de este estudio explicativo, se someterá al robot balancín a pruebas bajo criterios de competencias de robótica, evaluando su capacidad para mantenerse estable y seguir patrones de movimiento específicos. Se utilizará el método comparativo y la técnica de análisis de desempeño en escenarios de competencia. Se documentarán los resultados, identificando mejoras y optimizaciones necesarias para maximizar su rendimiento.

Técnica e instrumento de recolección de datos

FASE	ETAPA	TÉCNICA	INSTRUMENTO
Fase de diseño	Estudio previo inicial	Observación	Análisis de proyectos similares y revisión de guías
	Diseño de la estructura mecánica y electrónica	Experimentación	Simulación

	Diseño del sistema de control y simulación	Experimentación	Simulación
Fase de implementación	Fabricación de la estructura mecánica	Observación y medición	Lista de chequeo
	Ensamblaje de la tarjeta electrónica y sensores	Medición	Simulación, pruebas funcionales
	Programación del sistema de control	Experimentación, pruebas funcionales	Simulaciones
	Integración y pruebas iniciales	Observación	Registro de pruebas

CAPÍTULO IV

DESARROLLO DEL PROYECTO TÉCNICO

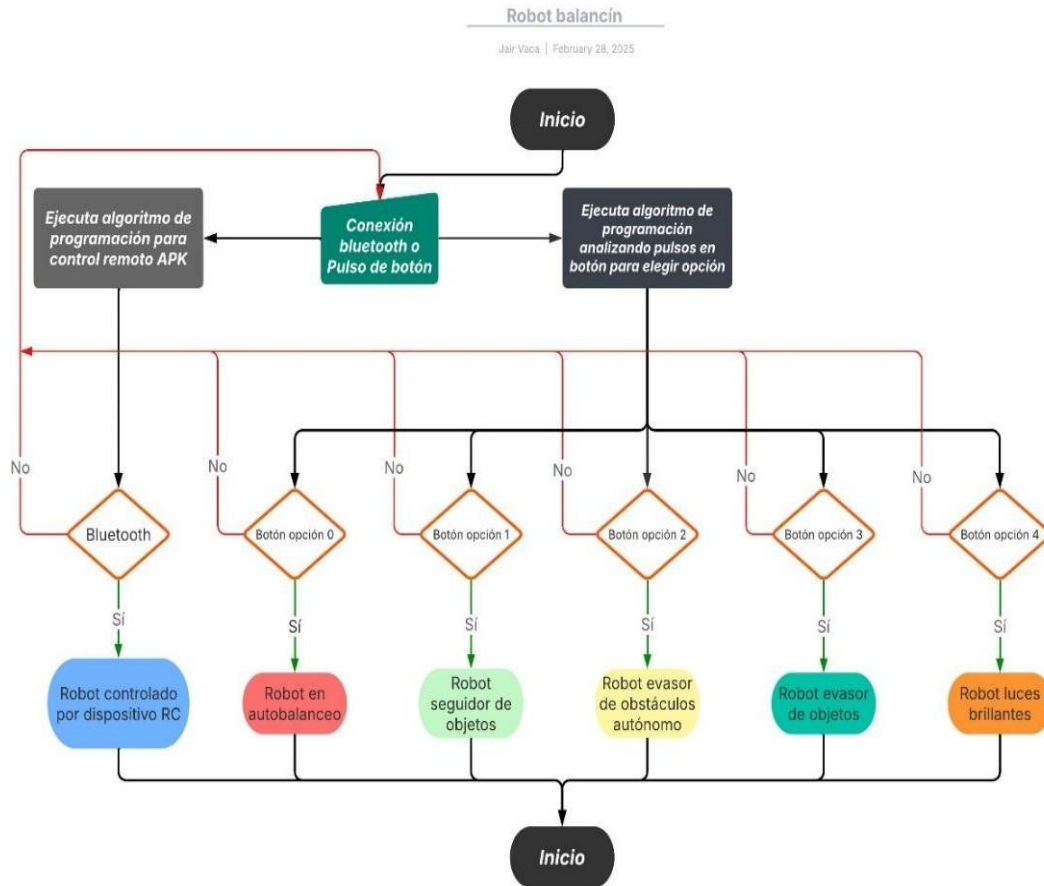
4.1. Diseño de la propuesta

4.1.1. Diagrama de flujo robot balancín

El siguiente diagrama de flujo del robot balancín fue diseñado para ilustrar el funcionamiento dual, permitiendo al usuario controlarlo a través de una aplicación vía Bluetooth o mediante un botón físico con 5 opciones predefinidas. El proceso inicia con la activación del sistema, donde el robot espera la selección del modo de control. Si se elige el control por bluetooth, el robot se conecta a la aplicación y responde a los comandos enviados desde ella. Por otro lado, si se opta por el botón físico, el usuario puede seleccionar entre las 5 opciones disponibles, cada una programada para ejecutar una secuencia específica de movimientos o acciones. El diagrama detalla cómo el robot procesa estas entradas, ejecuta las tareas correspondientes y vuelve al estado de espera para recibir nuevas instrucciones, asegurando un funcionamiento intuitivo y versátil.

Figura 12

Diagrama de Flujo Robot Balancín



Nota. Este diagrama fue diseñado en: Robot balancín | Lucidchart. (n.d).
https://lucid.app/lucidchart/91d27894-1040-4ea6-a0ba-7ab67b088c8c/edit?page=0_0&invitationId=inv_5859a238-536b-4c6a-bace-d6f1382f4e15#

El robot fue diseñado para cumplir operaciones específicas que le permitan ser utilizado en competencias de robótica en donde se califican destrezas del piloto en el caso que funcione por conexión bluetooth, y destrezas del robot en funcionamiento autónomo al cruzar una pista o laberinto con obstáculos.

Este tipo de robot puede ser utilizado también para demostraciones de equilibrio en los robots que se caracterizan por tener 2 ruedas. En el ámbito educativo puede utilizarse para la enseñanza de robótica educativa para la construcción de diagramas de flujo.

4.2. Diseño 3D para piezas estructurales de robot balancín

4.2.1. Plataforma Tinkercad Diseño 3D

Tinkercad es una herramienta de Diseño Asistido por Computador online sin pago, una de sus funciones es utilizada para crear modelos tridimensionales basado en una geometría sólida constructiva. La geometría sólida constructiva permite utilizar figuras sólidas tridimensionales como cubo, cilindro, pirámide para crear los modelos o piezas deseadas, es más fácil de utilizar que otro tipo de software CAD.

4.2.2. Diseño modular en Tinkercad

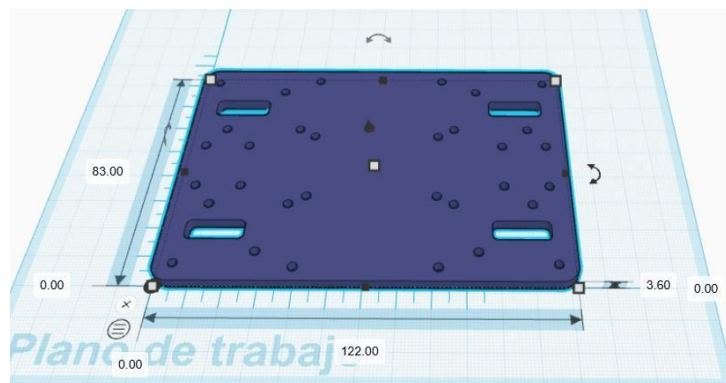
Una estrategia muy útil para un robot balancín es el diseño modular. Esto implica crear cada pieza del robot por separado, de manera que puedan ser ensambladas fácilmente y permitan ajustes individuales de ser necesario sin que afecten el diseño completo.

4.2.3. Piezas para robot balancín

4.2.3.1. Placa rectangular – base principal. Es la pieza que sirve de base para los soportes de motores y placa electrónica. Esta primera base es recomendable fabricarla en metal de 3.6 mm de espesor puesto que soportará la tracción y fuerza de los motores.

Figura 13

Diseño en Tinkercad placa base principal

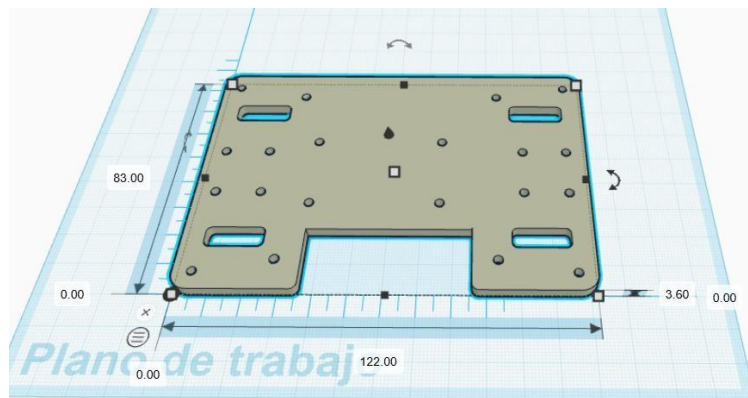


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.2. Placa rectangular – base secundaria. Esta pieza es la segunda base que servirá para soporte de la batería. Esta segunda base es recomendable fabricarla en acrílico de 3.6 mm de espesor para alivianar el peso del robot.

Figura 14

Diseño en Tinkercad placa base secundaria

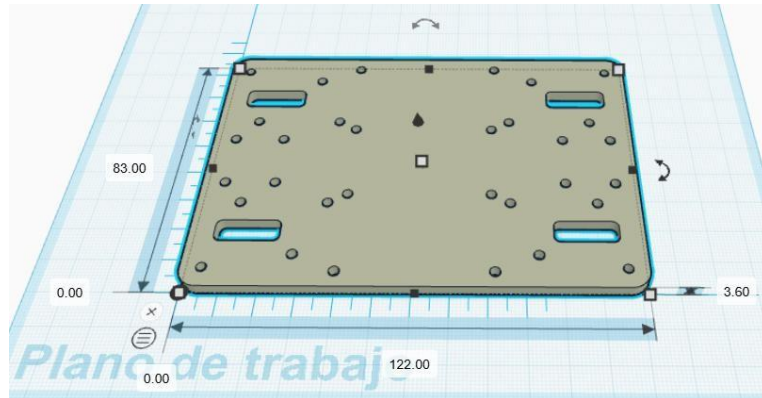


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.3. Placa rectangular – base superior. Esta pieza es la tercera base que servirá como soporte de objetos que se puedan poner sobre el robot, esto con el objetivo de realizar mediciones de estabilidad. Esta última base es recomendable fabricarla en acrílico de 3.6 mm de espesor para alivianar el peso del robot.

Figura 15

Diseño en Tinkercad placa base superior

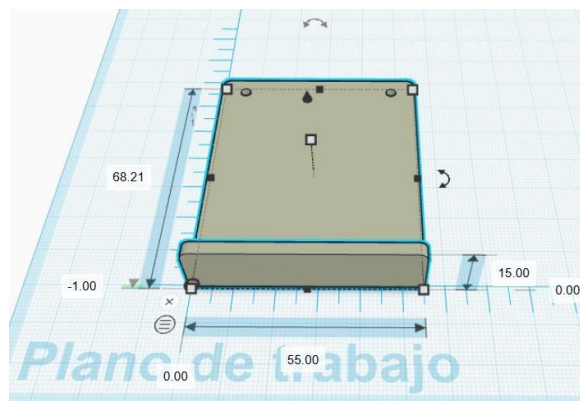


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.4. Soporte frontal. Esta pieza en forma de L servirá como soporte frontal del robot para evitar caídas bruscas cuando el robot pierda el equilibrio. Esta pieza es recomendable fabricarla en acrílico de 3.6 mm de espesor para alivianar el peso del robot.

Figura 16

Diseño en Tinkercad soporte frontal

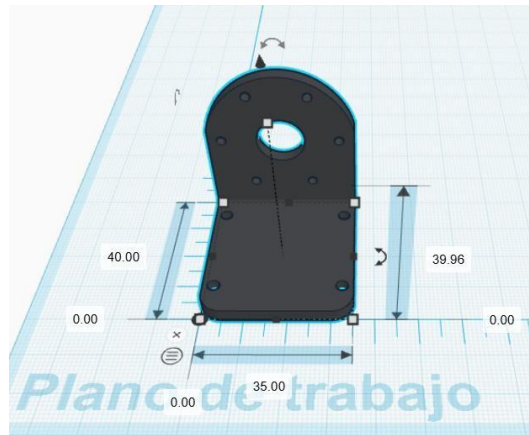


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.5. Soporte metálico de motor. Esta pieza en forma de L sirve como soporte del motor y se unirá a la placa base principal. Esta pieza es recomendable fabricarla en metal de 3.6 mm de espesor puesto que soportará la tracción y fuerza de los motores. Se necesitan dos piezas.

Figura 17

Diseño en Tinkercad soporte metálico de motor



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.6. Rueda. Esta pieza circular será utilizada como rueda para el robot, se utilizan dos piezas y son las encargadas de soportar todo el peso. Cada rueda está compuesta por un rin plástico y su neumático de caucho. Se necesitan dos piezas.

Figura 18

Diseño en Tinkercad rueda

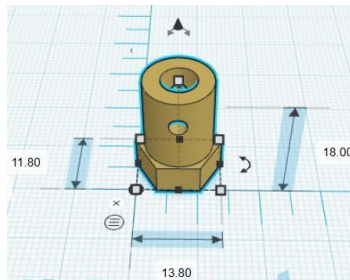


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.7. Unión hexagonal metálica. Esta pieza será utilizada para unir la rueda con el eje del motor, se caracteriza por tener un extremo hexagonal que encaja en la estructura del rin plástico de la rueda. Se necesitan dos piezas.

Figura 19

Diseño en Tinkercad unión hexagonal metálica

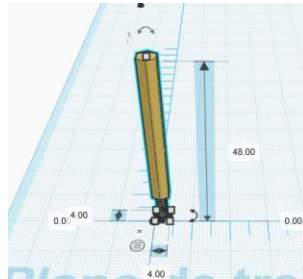


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.8. Poste metálico hexagonal. Esta pieza será utilizada para unir las placas rectangulares estructurales del robot, sirven también como separadores entre cada nivel de altura donde se implementarán las partes electrónicas. Se necesitan 8 piezas.

Figura 20

Diseño en Tinkercad poste metálico hexagonal

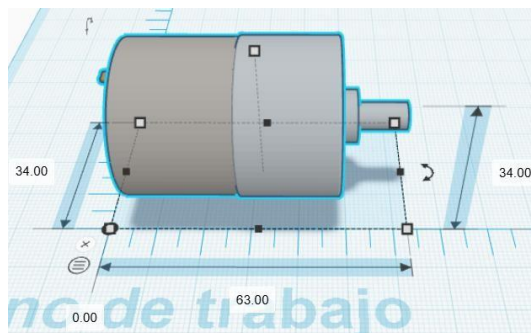


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad. <https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.9. Motor a pasos. Este elemento electrónico es la máquina que transforma los impulsos eléctricos en movimiento mecánico de su eje para en conjunto con su sensor de posición angular (encoder) y la rueda mantener el equilibrio de todo el robot. Se utilizarán dos piezas.

Figura 21

Diseño en Tinkercad motor a pasos



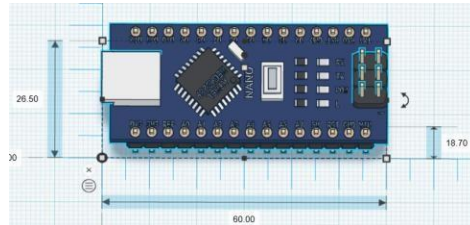
Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad. <https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot

4.2.3.10. Tarjeta de control. Esta pieza electrónica más conocida como microcontrolador será la encargada de procesar toda la información que ingrese de los sensores y según su código de programación, poder realizar la estabilización del robot mediante las señales enviadas a los motores en conjunto con la información de posición de los encoders. Se utilizará una tarjeta Arduino Nano.

Figura 22

Diseño en Tinkercad tarjeta de control Arduino nano



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad. <https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.11. Sensor ultrasónico. Esta pieza electrónica será utilizada para detectar objetos que estén frente al robot, y según la programación poder evitarlos o acercarse a estos. Se utilizará una pieza de sensor HC-SR04.

Figura 23

Diseño en Tinkercad sensor ultrasónico



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.12. Puente H. Esta pieza electrónica será utilizada para el control de accionamiento de cada motor, y según la programación poder mantener equilibrado al robot. Se utilizará una pieza módulo TB6612FNG.

Figura 24

Diseño en Tinkercad puente H



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.13. Acelerómetro y Giroscopio. Esta pieza electrónica será utilizada para detectar el posicionamiento del robot en términos de coordenadas tridimensionales X, Y, Z, y según la programación enviar las señales correctas a los motores para mantener el equilibrio del robot. Se utilizará una pieza módulo GY-521 MPU6050.

Figura 25

Diseño en Tinkercad acelerómetro/giroscopio



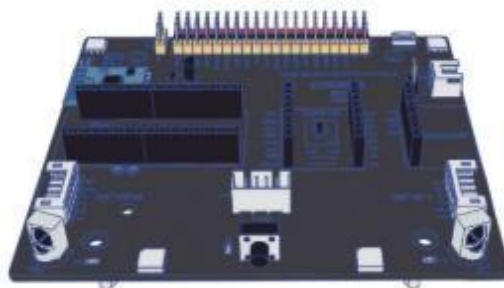
Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.

<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.14. Tarjeta Shield ELEGOO. Esta pieza electrónica está diseñada con toda la circuitería que une todos los elementos electrónicos, será utilizada para comunicar la tarjeta de control, sensores y actuadores.

Figura 26

Diseño en Tinkercad tarjeta shield Elegoo

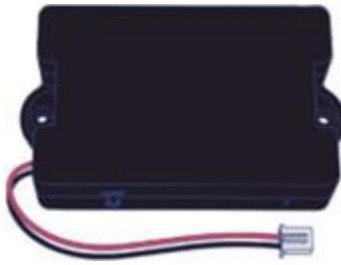


Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad. <https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.3.15. Batería. Esta pieza electrónica será utilizada para proporcionar la energía que permita el funcionamiento correcto del robot. Es de litio, 2 celdas, 7.4 voltios y 3500 mAh. Se utilizará una pieza.

Figura 27

Diseño en Tinkercad batería



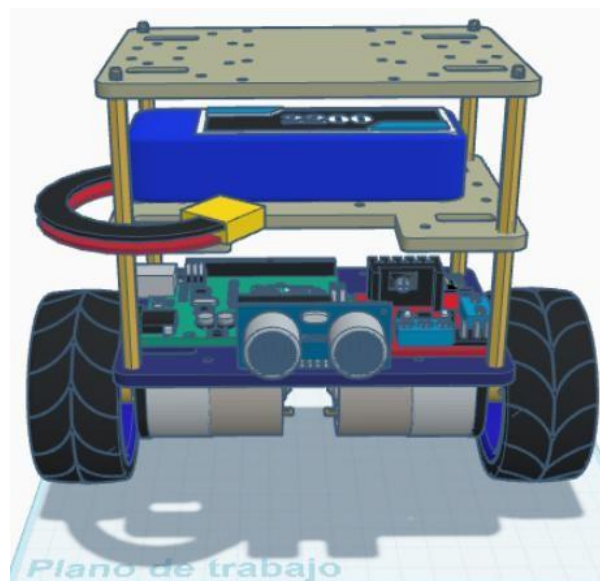
Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Robot balancín ensamblado

En un proceso posterior al diseño modular de piezas mecánicas y electrónicas del robot balancín, la plataforma CAD de Tinkercad nos permite realizar una simulación virtual del ensamblaje completo del robot. A continuación, se muestra el robot balancín totalmente armado:

Figura 28

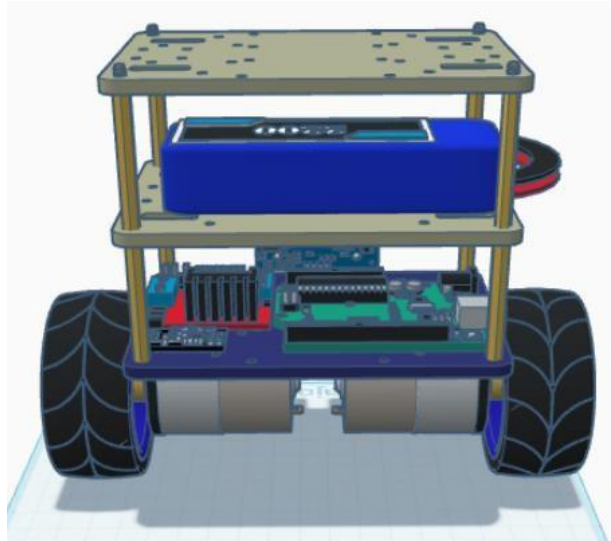
Diseño en Tinkercad robot balancín frente



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Figura 29

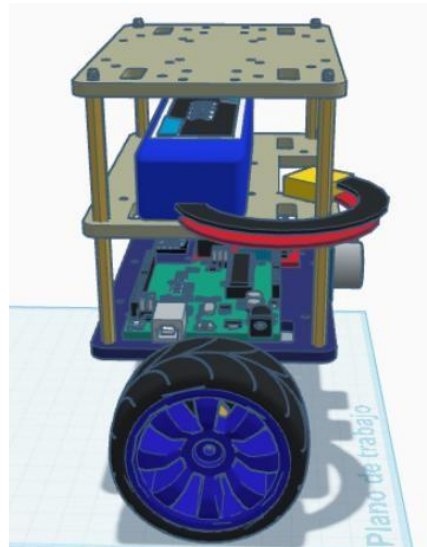
Diseño en Tinkercad robot balancín posterior



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Figura 30

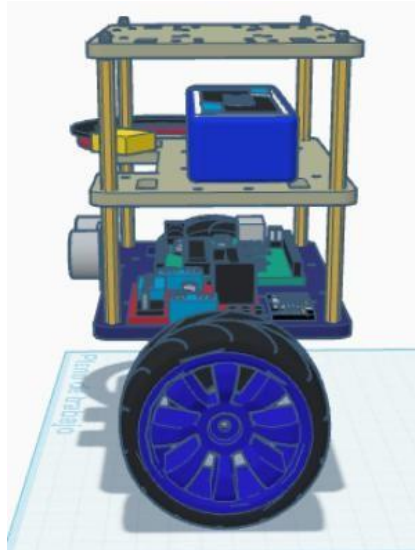
Diseño en Tinkercad robot balancín lateral derecha



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Figura 31

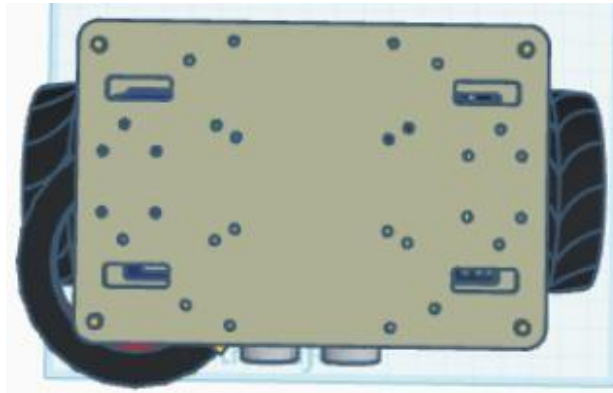
Diseño en Tinkercad robot balancín lateral izquierda



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Figura 32

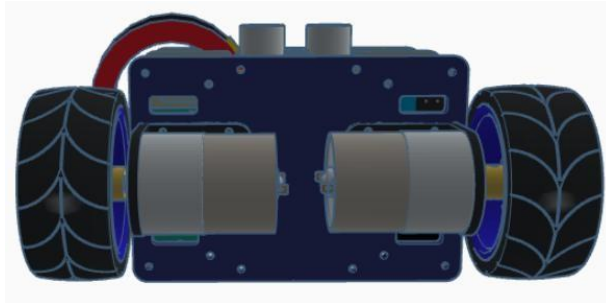
Diseño en Tinkercad robot balancín superior



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

Figura 33

Diseño en Tinkercad robot balancín inferior



Nota. Esta pieza fue diseñada en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.4. Plataforma Tinkercad Diseño de Circuitos

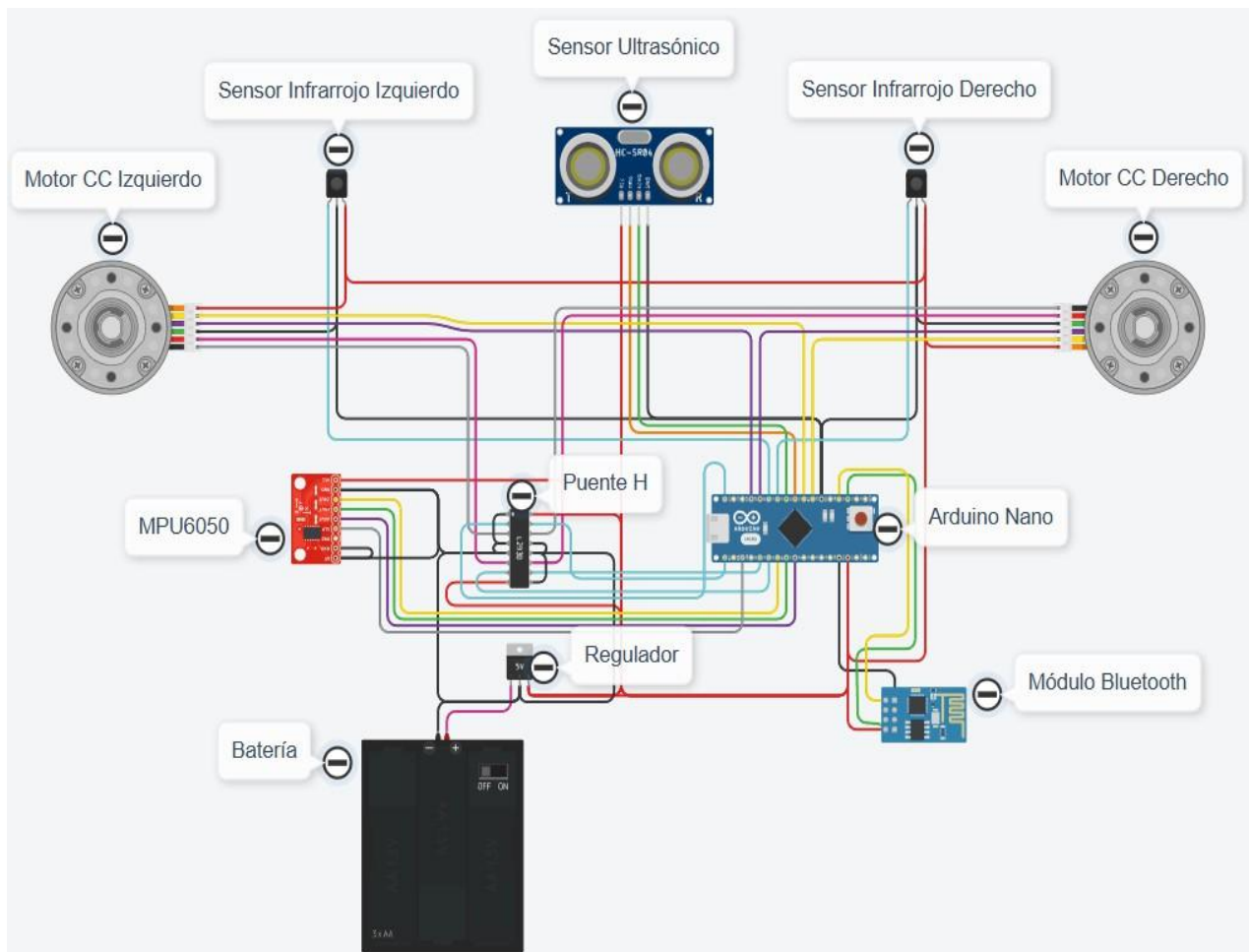
Tinkercad es una herramienta de Diseño Asistido por Computador online sin pago, una de sus funciones es utilizada para crear circuitos electrónicos basados en módulos existentes en el mercado como arduino. Su versatilidad permite realizar diseños con precisión, es más fácil de utilizar que otro tipo de software de diseño.

4.2.5. Diseño circuito robot balancín

A continuación, se presenta una simulación gráfica del circuito incluyendo su tarjeta de control Arduino Nano, puente H, motores, sensor ultrasónico y demás componentes electrónicos.

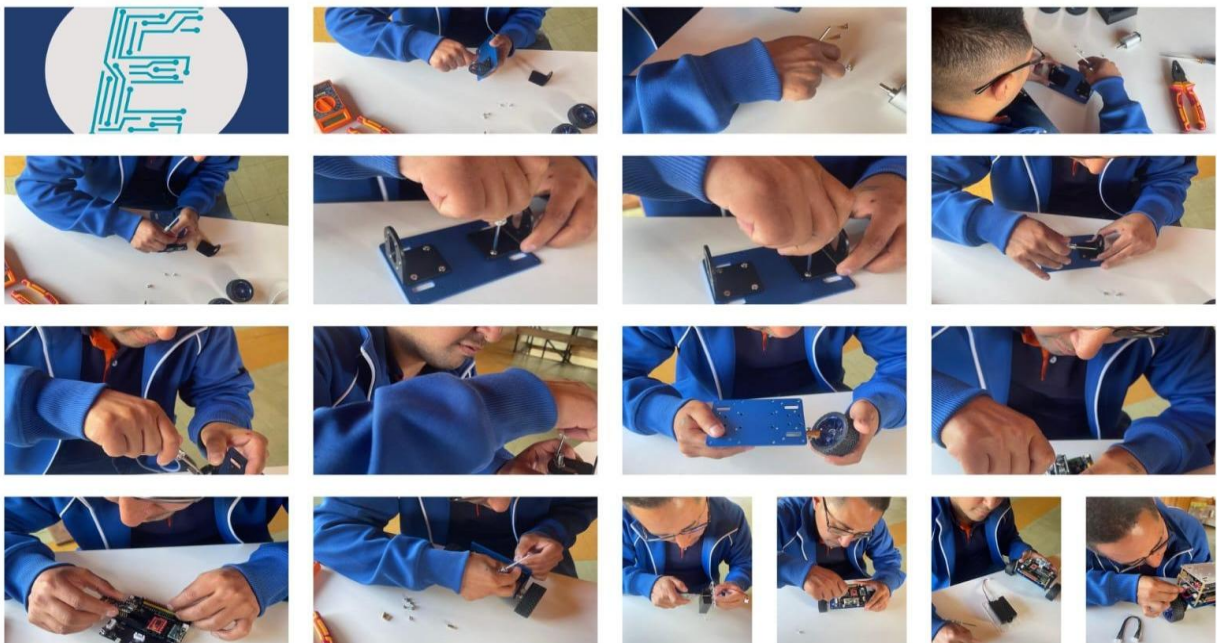
Figura 34

Diseño en Tinkercad circuito electrónico robot balancín



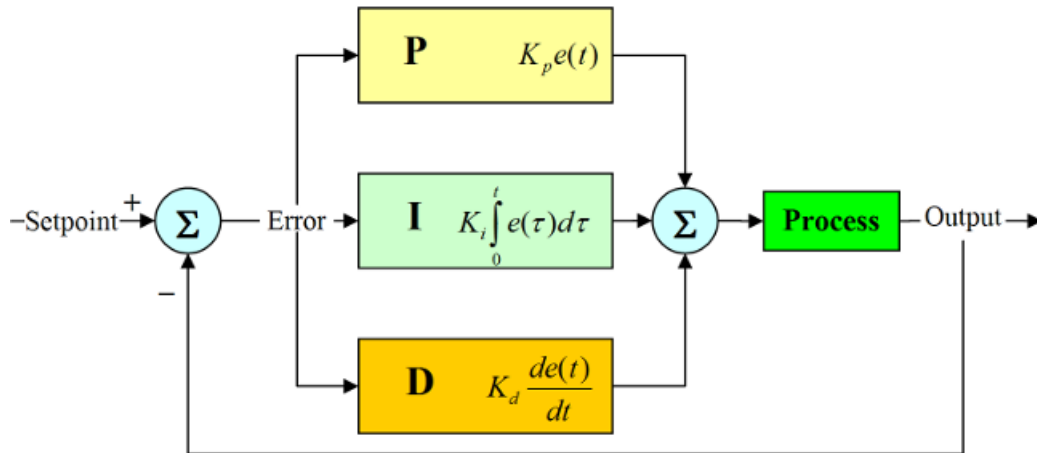
Nota. Este circuito fue diseñado en: *Login - Tinkercad.* (n.d.-c). Tinkercad.
<https://www.tinkercad.com/things/4818j5iDkqI-diseno-piezas-robot-balancin/edit?returnTo=%2Fthings%2F4818j5iDkqI-balancin-robot>

4.2.6. Construcción robot balancín



4.2.7. Control PID

La metodología que se implementará para mantener la velocidad en RPM de los motores, será el controlador PID implementado en el Arduino. El control PID está compuesto por una acción proporcional, una acción derivativa y una acción integradora.



Para la implementación en el Arduino Nano se utilizará la suma de las 3 acciones del control PID:

Acción proporcional

$$P = k_p \cdot \text{error};$$

El término proporcional realiza un cambio en la salida que es proporcional al valor actual, en donde permite mejorar el tiempo de respuesta del sistema. Por encima de un valor límite el cual es el que se debe sintonizar, el sistema comenzará a oscilar.

Acción derivada

$D = K_d * (\text{error} - \text{error_anterior});$

Esta acción proporcional a la variación de error, esta acción calcula que tan rápido se produce un cambio en el sistema.

Acción integral

Integrador_er

ror += error;

I= ki * (

Integrador_er

ror);

El término integral es el que permite el cálculo de ajuste para el valor de aterrizaje del sistema “error de estado estacionario”. Para calcular todo el controlador PID se realiza la suma de las 3 medidas:

Int error = (int

punto_entrada, set_point);

P= kp*error;

Integrador_er

ror += error;

I= ki * (

integrador_er

ror);

D= KD*(error –

error_anterior);

Error_anterior= error;

Return ((P+I+D));

Sintonización manual

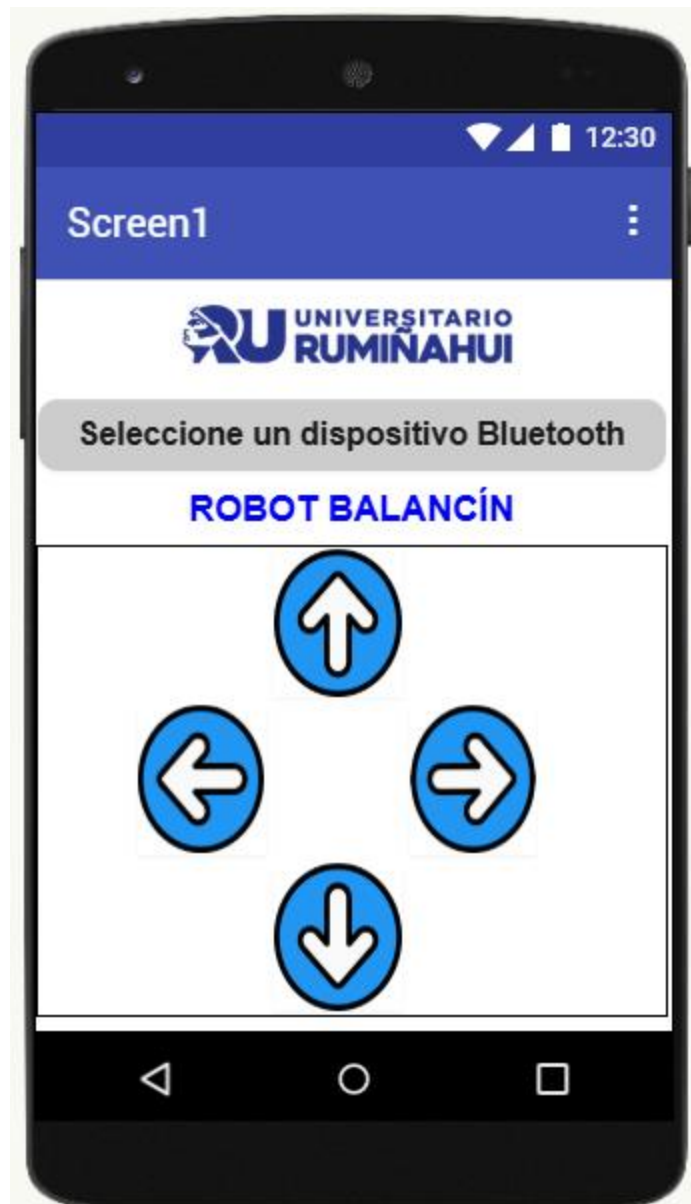
Sintonización manual, coloque las tres constantes a cero, variar el valor de KP hasta que se produce la oscilación.

Luego subir Kd hasta oscilación desaparece. Ajuste KD hasta que el sistema este críticamente amortiguado, y luego aumentar el Ki hasta que el error de estado estacionario tiende a cero en un tiempo razonable.

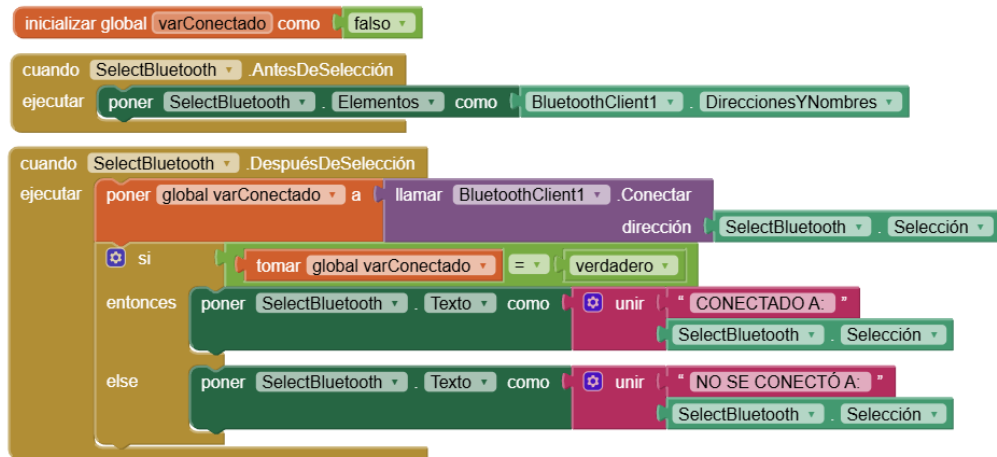
A continuación, se muestra un diagrama de flujo del funcionamiento en el proceso de control PID en el compilador Arduino Nano.

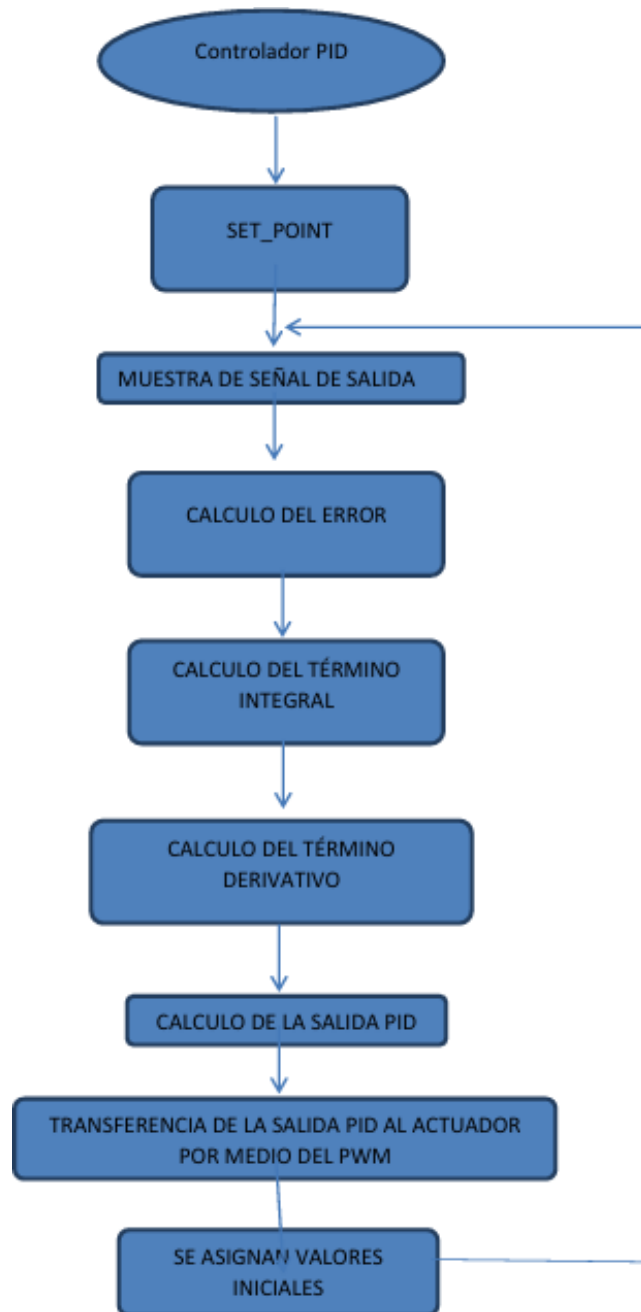
4.2.7. Diseño de aplicación para control de robot balancín

Para mejorar la experiencia de control del robot balancín, desarrollamos una aplicación en App Inventor que permite la comunicación inalámbrica mediante Bluetooth. La interfaz de la app ha sido diseñada para ser intuitiva, ofreciendo botones de dirección y una conexión estable con el robot. Durante el proceso de desarrollo, implementamos bloques de programación que envían comandos específicos al microcontrolador del robot, garantizando una respuesta precisa a cada acción del usuario. Además, se realizaron pruebas para optimizar la latencia y mejorar la estabilidad de la conexión, asegurando un control fluido y eficiente. Esta aplicación, que lleva el logo del Instituto Superior Rumiñahui, complementa el aprendizaje en robótica al permitir a los estudiantes interactuar con el robot de manera práctica e innovadora.



A continuación, tenemos el desarrollo en programación por bloques para la conexión por bluetooth.





CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El desarrollo del proyecto de titulación ha permitido la aplicación de conceptos teóricos de control automático, electrónica y robótica en un entorno práctico, fortaleciendo de manera lúdica el aprendizaje mediante la simulación y la resolución de problemas reales que ocurrían durante el desarrollo del mismo.

La implementación de un controlador PID ha demostrado ser el método más práctico para este proyecto de titulación ya que garantiza la estabilidad del robot balancín, logrando un desempeño óptimo en diferentes condiciones operativas y mejorando su capacidad de respuesta ante perturbaciones externas.

El prototipo desarrollado no solo servirá como una herramienta didáctica en la enseñanza de la robótica, sino que también se posiciona como una plataforma potencial para la participación en competencias tecnológicas, promoviendo la innovación y el desarrollo de habilidades prácticas en los estudiantes de robótica del Instituto Superior Tecnológico Rumiñahui.

Las simulaciones dieron lugar a la selección de componentes adecuados previas a la implementación física de estos, lo que ha permitido optimizar costos y garantizar la funcionalidad del robot. Además, el proyecto puede ser mejorado y ampliado en futuras versiones, incorporando nuevas estrategias de control y tecnologías emergentes.

5.2. Recomendaciones

Se sugiere explorar algoritmos de control nuevos y avanzados, como controladores diferentes o técnicas basadas en inteligencia artificial, para mejorar aún más la estabilidad y respuesta del robot en diferentes entornos.

Para mejorar la precisión y autonomía del robot balancín, se recomienda integrar sensores adicionales, cámaras o ultrasonidos, que permitan una mejor percepción del entorno y faciliten la navegación autónoma y en entornos más complejos.

Se debe realizar una mayor cantidad de pruebas experimentales en distintos escenarios y superficies, para evaluar la robustez del sistema y optimizar su desempeño en situaciones reales de operación.

Es recomendable generar material didáctico detallado sobre el diseño, implementación y pruebas del robot, con el fin de facilitar su uso en futuras investigaciones, así como su adaptación para nuevas aplicaciones y mejoras en el ámbito académico y tecnológico.

5.3. Referencias

Referencias

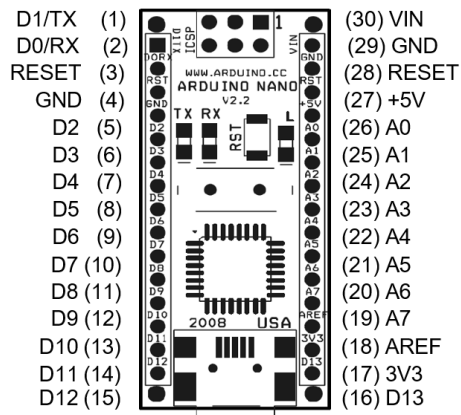
- Calvo, R. C. (Septiembre de 2022). *UNED*. Obtenido de https://www.uned.es/universidad/facultades/dam/jcr:497e4f13-8e2c-4a2f-bf9d-1cced8aa6e7c/PFM_Ricardo_Camacho_Calvo.pdf
- Electrostore. (2024). *grupoelectrostore.com*. Obtenido de <https://grupoelectrostore.com/shop/>
- Gonzalez, O. (2024). *Bricogeek.com*. Obtenido de <https://lab.bricogeek.com/tutorial/guia-de-modelos-arduino-y-sus-caracteristicas/arduino-nano>
- José, Y. C. (2024). *Repositorio.upse.edu.ec*. Obtenido de <https://repositorio.upse.edu.ec/bitstream/46000/11198/4/UPSE-TET-2024-0006.pdf>
- Macías, L. G. (2020). *Universidad Autónoma de Barcelona*. Obtenido de https://ddd.uab.cat/pub/tfg/2020/234238/TFG_LuisGeovannyTrivinoMacias.pdf
- Naylampmechatronics. (2024). *naylampmechatronics.com*. Obtenido de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html

Novatronic. (2020). *Novatronic.com*. Obtenido de
<https://novatronic.com/index.php/product/tb6612fng-driver-para-motor-dc/>

ANEXOS

Datasheet Arduino Nano

Arduino Nano Pin Layout



Código de Programación

```
1  //Librerías
2  #include <Arduino.h>
3  #include "PinChangeInt.h"
4  #include "Pins.h"
5  #include "mode.h"
6  #include "Command.h"
7  #include "BalanceCar.h"
8  #include "Rgb.h"
9  #include "Ultrasonic.h"
10 #include "voltage.h"
11
12 unsigned long start_prev_time = 0;
13 boolean carInitialize_en = true;
14
15 //Modos de funcionamiento
16 void functionMode()
17 {
18     switch (function_mode)
19     {
20     case IDLE:
21         break;
22     case IRREMOTE:
23         break;
24     case OBSTACLE:
25         obstacleAvoidanceMode();
26         break;
27     case FOLLOW:
28         followMode();
29         break;
30     case BLUETOOTH:
31         break;
32     case FOLLOW2:
33         followMode2();
34         break;
35     default:
36         break;
37     }
38 }
39
```

```

40 void setMotionState()
41 {
42     switch (motion_mode)
43     {
44     case FORWARD:
45         switch (function_mode)
46         {
47         case FOLLOW:
48             setting_car_speed = 20;
49             setting_turn_speed = 0;
50             break;
51         case FOLLOW2:
52             setting_car_speed = 20;
53             setting_turn_speed = 0;
54             break;
55         case BLUETOOTH:
56             setting_car_speed = 80;
57             break;
58         case IRREMOTE:
59             setting_car_speed = 80;
60             setting_turn_speed = 0;
61             break;
62         default:
63             setting_car_speed = 40;
64             setting_turn_speed = 0;
65             break;
66         }
67         break;
68     case BACKWARD:
69         switch (function_mode)
70         {
71         case FOLLOW:
72             setting_car_speed = -20;
73             setting_turn_speed = 0;
74             break;
75         case FOLLOW2:
76             setting_car_speed = -20;
77             setting_turn_speed = 0;
78             break;
79         case BLUETOOTH:
80             setting_car_speed = -80;
81             break;
82         case IRREMOTE:
83             setting_car_speed = -80;
84             setting_turn_speed = 0;
85             break;
86         default:
87             setting_car_speed = -40;
88             setting_turn_speed = 0;
89             break;
90         }
91         break;

```

```

92     case TURNLEFT:
93         switch (function_mode)
94         {
95             case FOLLOW:
96                 setting_car_speed = 0;
97                 setting_turn_speed = 50;
98                 break;
99             case FOLLOW2:
100                 setting_car_speed = 0;
101                 setting_turn_speed = 50;
102                 break;
103             case BLUETOOTH:
104                 setting_turn_speed = 80;
105                 break;
106             case IRREMOTE:
107                 setting_car_speed = 0;
108                 setting_turn_speed = 80;
109                 break;
110             default:
111                 setting_car_speed = 0;
112                 setting_turn_speed = 50;
113                 break;
114         }
115         break;
116     case TURNRIGHT:
117         switch (function_mode)
118         {
119             case FOLLOW:
120                 setting_car_speed = 0;
121                 setting_turn_speed = -50;
122                 break;
123             case FOLLOW2:
124                 setting_car_speed = 0;
125                 setting_turn_speed = -50;
126                 break;
127             case BLUETOOTH:
128                 setting_turn_speed = -80;
129                 break;
130             case IRREMOTE:
131                 setting_car_speed = 0;
132                 setting_turn_speed = -80;
133                 break;
134             default:
135                 setting_car_speed = 0;
136                 setting_turn_speed = -50;
137                 break;
138         }
139         break;
140     case STANDBY:
141         setting_car_speed = 0;
142         setting_turn_speed = 0;
143         break;

```



```

144 case STOP:
145     if (millis() - start_prev_time > 1000)
146     {
147         function_mode = IDLE;
148         if (balance_angle_min <= kalmanfilter_angle && kalmanfilter_angle <= balance_angle_max)
149         {
150             motion_mode = STANDBY;
151             rgb.lightOff();
152         }
153     }
154     break;
155 case START:
156     if (millis() - start_prev_time > 2000)
157     {
158         if (balance_angle_min <= kalmanfilter_angle && kalmanfilter_angle <= balance_angle_max)
159         {
160             car_speed_integral = 0;
161             setting_car_speed = 0;
162             motion_mode = STANDBY;
163             rgb.lightOff();
164         }
165         else
166         {
167             motion_mode = STOP;
168             carStop();
169             rgb.brightRedColor();
170         }
171     }
172     break;
173 default:
174     break;
175 }
176 }
177

```

```

178 void keyEventHandle()
179 {
180     if (key_value != '\0')
181     {
182         key_flag = key_value;
183
184         switch (key_value)
185         {
186             case 's':
187                 rgb.lightOff();
188                 motion_mode = STANDBY;
189                 break;
190             case 'f':
191                 rgb.flashBlueColorFront();
192                 motion_mode = FORWARD;
193                 break;
194             case 'b':
195                 rgb.flashBlueColorback();
196                 motion_mode = BACKWARD;
197                 break;
198             case 'l':
199                 rgb.flashBlueColorLeft();
200                 motion_mode = TURNLEFT;
201                 break;
202             case 'i':
203                 rgb.flashBlueColorRight();
204                 motion_mode = TURNRIGHT;
205                 break;
206             case '1':
207                 function_mode = FOLLOW;
208                 follow_flag = 0;
209                 follow_prev_time = millis();
210                 break;
211             case '2':
212                 function_mode = OBSTACLE;
213                 obstacle_avoidance_flag = 0;
214                 obstacle_avoidance_prev_time = millis();
215                 break;
216             case '3':

```

```

217     rgb_loop:
218         key_value = '\0';
219         rgb.flag++;
220         if (rgb.flag > 6)
221         {
222             rgb.flag = 1;
223         }
224         switch (rgb.flag)
225         {
226             case 0:
227                 break;
228             case 1:
229                 if (rgb.theaterChaseRainbow(50) && key_value == '3')
230                     goto rgb_loop;
231                 break;
232             case 2:
233                 if (rgb.rainbowCycle(20) && key_value == '3')
234                     goto rgb_loop;
235                 break;
236             case 3:
237                 if (rgb.theaterChase(127, 127, 127, 50) && key_value == '3')
238                     goto rgb_loop;
239                 break;
240             case 4:
241                 if (rgb.rainbow(20) && key_value == '3')
242                     goto rgb_loop;
243                 break;
244             case 5:
245                 if (rgb.whiteOverRainbow(20, 30, 4) && key_value == '3')
246                     goto rgb_loop;
247                 break;
248             case 6:
249                 if (rgb.rainbowFade2White(3, 50, 50) && key_value == '3')
250                     goto rgb_loop;
251                 break;
252                 break;
253             default:
254                 break;
255         }
256         break;
257     case '4':
258         function_mode = IDLE;
259         motion_mode = STOP;
260         carBack(110);
261         delay((kalmanfilter_angle - 30) * (kalmanfilter_angle - 30) / 8);
262         carStop();
263         start_prev_time = millis();
264         rgb.brightRedColor();
265         break;

```

```

266     case '5':
267         if (millis() - start_prev_time > 500 && kalmanfilter_angle >= balance_angle_min)
268         {
269             start_prev_time = millis();
270             motion_mode = START;
271         }
272         motion_mode = START;
273         break;
274     case '6':
275         rgb.brightness = 50;
276         rgb.setBrightness(rgb.brightness);
277         rgb.show();
278         break;
279     case '7':
280         rgb.brightRedColor();
281         rgb.brightness -= 25;
282         if (rgb.brightness <= 0)
283         {
284             rgb.brightness = 0;
285         }
286         rgb.setBrightness(rgb.brightness);
287         rgb.show();
288         break;
289     case '8':
290         rgb.brightRedColor();
291         rgb.brightness += 25;
292         if (rgb.brightness >= 255)
293         {
294             rgb.brightness = 255;
295         }
296         rgb.setBrightness(rgb.brightness);
297         rgb.show();
298         break;
299     case '9':
300         rgb.brightness = 0;
301         rgb.setBrightness(rgb.brightness);
302         rgb.show();
303         break;
304     case '0':
305         function_mode = FOLLOW2;
306         follow_flag = 0;
307         follow_prev_time = millis();
308         break;
309     case '*':
310         break;
311     case '#':
312         break;
313     default:
314         break;
315 }
316 if (key_flag == key_value)
317 {
318     key_value = '\0';
319 }
320 }
321 }

```

```

322
323 void setup()
324 {
325
326     Serial.begin(9600);
327     ultrasonicInit();
328     keyInit();
329     rgb.initialize();
330     voltageInit();
331     start_prev_time = millis();
332     carInitialize();
333 }
334 void loop()
335 {
336     getKeyValue();
337     getBluetoothData();
338     keyEventHandle();
339     getDistance();
340     voltageMeasure();
341     setMotionState();
342     functionMode();
343     checkObstacle();
344     rgb.blink(100);
345     static unsigned long print_time;
346     if (millis() - print_time > 100)
347     {
348         print_time = millis();
349         Serial.println(kalmanfilter.angle);
350     }
351     static unsigned long start_time;
352     if (millis() - start_time < 10)
353     {
354         function_mode = IDLE;
355         motion_mode = STOP;
356         carStop();
357     }
358     if (millis() - start_time == 2000) // Ingreso péndulo
359     {
360         key_value = '5';
361     }
362 }

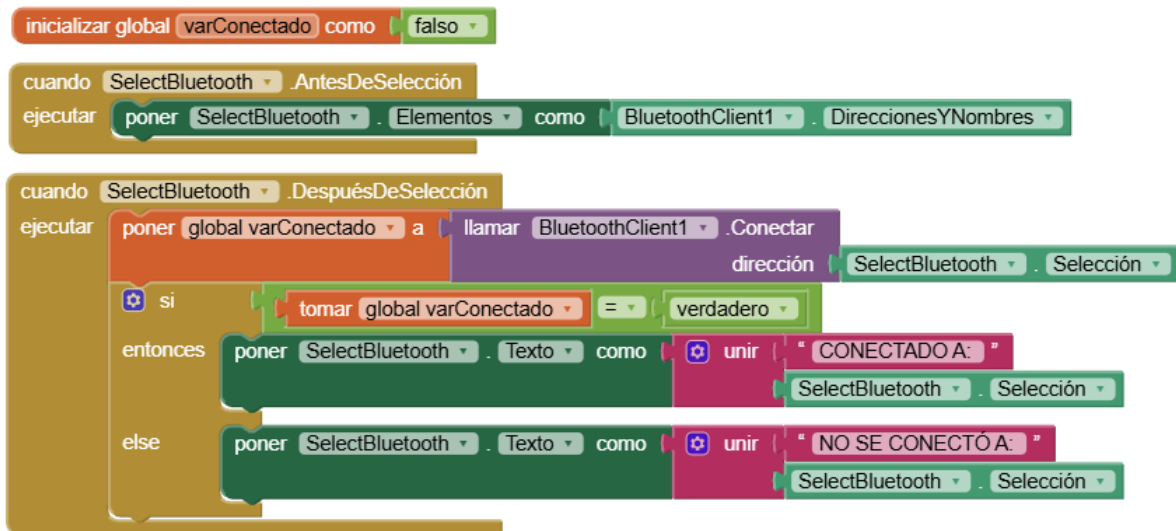
```

Componentes PID

```
8  #include "MsTimer2.h"
9  #include "KalmanFilter.h"
10 #include "I2Cdev.h"
11 // #include "MPU6050_6Axis_MotionApps20.h"
12
13 #include "MPU6050.h"
14 #include "Wire.h"
15 MPU6050 mpu;
16 KalmanFilter kalmanfilter;
17
18 //Setting PID parameters
19
20 double kp_balance = 55, kd_balance = 0.75;
21 double kp_speed = 10, ki_speed = 0.26;
22 double kp_turn = 2.5, kd_turn = 0.5;
23
24 //Setting MPU6050 calibration parameters
25 double angle_zero = 0; //x axle angle calibration
26 double angular_velocity_zero = 0; //x axle angular velocity calibration
27
28 volatile unsigned long encoder_count_right_a = 0;
29 volatile unsigned long encoder_count_left_a = 0;
30 int16_t ax, ay, az, gx, gy, gz;
31 float dt = 0.005, Q_angle = 0.001, Q_gyro = 0.005, R_angle = 0.5, C_0 = 1, K1 = 0.05;
32
33 int encoder_left_pulse_num_speed = 0;
34 int encoder_right_pulse_num_speed = 0;
35 double speed_control_output = 0;
36 double rotation_control_output = 0;
37 double speed_filter = 0;
38 int speed_control_period_count = 0;
39 double car_speed_integral = 0;
40 double speed_filter_old = 0;
41 int setting_car_speed = 0;
42 int setting_turn_speed = 0;
43 double pwm_left = 0;
44 double pwm_right = 0;
45 float kalmanfilter_angle;
46 // char balance_angle_min = -27;
47 // char balance_angle_max = 27;
48 char balance_angle_min = -22;
49 char balance_angle_max = 22;
```

Configuraciones APP Inventor – Aplicación móvil

Configuración bluetooth



Configuración Botones para movimientos de robot

