

INSTITUTO TECNOLÓGICO SUPERIOR

VIDA NUEVA



CARRERA:

TECNOLOGIA EN ELECTROMECAÁNICA

TEMA DEL PROYECTO DE TITULACIÓN:

**CONSTRUCCIÓN DE UN ROBOT BALANCIN CONTROLADO POR UNA
APLICACIÓN MOVIL**

AUTOR:

PERÉZ VALENCIA JASON EDUARDO

TUTOR:

MSc. TITUAÑA DIAZ DARWIN VINICIO

FECHA:

AGOSTO 2020

QUITO – ECUADOR

CESIÓN DE DERECHOS DE AUTOR

Yo, PERÉZ VALENCIA JASON EDUARDO portador/a de la cedula de ciudadanía 175064922 – 8, facultado/a de la carrera **TECNOLOGÍA EN ELECTROMECHANICA** autor/a de esta obra certifico y proveo al Instituto Tecnológico Superior Vida Nueva, usar plenamente el contenido plasmado en este escrito con el tema “**CONSTRUCCIÓN DE UN ROBOT BALANCÍN CONTROLADO POR UNA APLICACIÓN MÓVIL**”, con el objeto de aportar y promover la lectura e investigación, autorizando la publicación de mi trabajo de titulación en la colección digital del repositorio institucional bajo la licencia de Creative Commons: Atribución-No Comercial-Sin Derivadas.

En la ciudad de Quito, al mes de Agosto de 2020

PERÉZ VALENCIA JASON EDUARDO

C.I.:175064922-8

CERTIFICACIÓN

En mi calidad de Tutor del proyecto: **CONSTRUCCIÓN DE UN ROBOT BALANCÍN CONTROLADO POR UNA APLICACIÓN MÓVIL**, Presentado por el estudiante **PERÉZ VALENCIA JASON EDUARDO**, para optar por el título de Tecnólogo en Electromecánica, certifico que dicho proyecto ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal que se designe.

TUTOR: MSc. TITUAÑA DIAZ DARWIN VINICIO

C.I.:1716233539

APROBACIÓN DEL TRIBUNAL EXAMINADOR

Los miembros del tribunal examinador aprueban el informe del proyecto de Aplicación Práctica, con el tema:

CONSTRUCCIÓN DE UN ROBOT BALANCÍN CONTROLADO POR UNA APLICACIÓN MÓVIL

Del Sr. estudiante: **PERÉZ VALENCIA JASON EDUARDO**

De la Carrera, Tecnología en Electromecánica

C.I.:

C.I.:

C.I.:

C.I.:

DECLARACIÓN DE AUTENTICIDAD

YO, **PERÉZ VALENCIA JASON EDUARDO**, estudiante del Instituto Tecnológico Superior Vida Nueva, declaro que he realizado este trabajo de titulación tomando en cuenta consideración citas bibliográficas que se nombran en este texto.

A través de la presente declaración cedo mis derechos de propiedad correspondiente a este trabajo de titulación al Instituto Tecnológico Superior Vida Nueva.

PERÉZ VALENCIA JASON EDUARDO

C.I.:175064922-8

AGRADECIMIENTO

El agradecimiento de mi trabajo práctico de titulación es primeramente a Dios por esta gran oportunidad y por la vida que me ha otorgado para cumplir este sueño de ser un gran profesional, a mis padres por el apoyo, y los valores que me inculcaron a lo largo de los tres años de estudio, y también a la Institución Tecnológico Superior Vida Nueva por su gran calidad de educación que otorga para ser profesionales competitivos a nivel nacional e internacional que es gracias a sus docentes con su profesionalismo y rectitud que contribuyen para compartir sus conocimientos para llegar a ser lo que somos hoy en día profesionales éticos y morales en ámbito tecnológico del país.

Igualmente quiero agradecer a mi tutor el Magister Darwin Tituaña por brindarme su valiosa colaboración y orientación para el desarrollo de este trabajo de titulación.

Pérez Valencia Jason Eduardo

DEDICATORIA

Dedico este proyecto practico a Dios y a mis padres y a mi hermano, a Dios por otorgarme todos los bienes necesarios para llegar hacer realidad mi proyecto y por estar siempre mí, luego a mis padres y hermano por el apoyo cálido y hogareño que día a día me brindaron ideales de convicción que siempre los llevare a donde quiera que vaya sin importar en que circunstancia me encuentre en mi vida, y por ultimo recalco el agradecimiento a cada uno de los docentes por su apoyo y sus conocimientos para culminar mi carrera profesional por su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento de mi capacidad.

Pérez Valencia Jason Eduardo

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR	i
CERTIFICACIÓN	ii
APROBACIÓN DEL TRIBUNAL EXAMINADOR	iii
DECLARACIÓN DE AUTENTICIDAD	iv
AGRADECIMIENTO	v
DEDICATORIA	vi
ÍNDICE GENERAL	vii
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	x
INDICE DE ANEXOS	xi
INTRODUCCION	xii
ANTECEDENTES	xiii
RESUMEN	xiv
ABSTRACT	xv
1. OBJETIVOS	1
1.1. Objetivo General	1
1.2. Objetivos Específicos	1
2. DESARROLLO O CUERPO PRINCIPAL	2
2.1. Marco Teórico – Conceptual	2
2.1.1. La robótica	2
2.1.2. Robot	2
2.1.3. La robótica móvil	3
2.1.4. Arduino	3
2.1.5. Sensores	7
2.1.6. Motores paso a paso	9

2.1.7. Batería de polímero de litio	13
2.1.8. Ruedas	14
2.2. Procedimiento – Metodología.....	15
2.2.1. Diseño.....	15
2.2.2. Construcción.....	22
2.2.3. Programación.....	26
2.2.4. Comprobación	29
2.2.5. Metodología.....	30
3. CONCLUSIONES.....	31
4. RECOMENDACIONES	32
5. REFERENCIAS	33
5.1. BIBLIOGRAFÍA	33
5.2. REFERENCIA A PÁGINAS WEBS:	33
6. ANEXOS	36

ÍNDICE DE FIGURAS

Figura 1. Robot Explorador M6 UGV.....	2
Figura 2. Un robot Segway.....	2
Figura 3. Robot Seguidor de línea.	3
Figura 4. Arduino Nano.....	3
Figura 5. Arduino Uno.....	4
Figura 6. Arduino Mega.....	5
Figura 7. Módulos de Arduino.....	6
Figura 8. Ejes de Libertad.....	7
Figura 9. Sensor MPU 6500.....	7
Figura 10. Sensor ICM20602.....	8

Figura 11. Módulos Bluetooth.....	8
Figura 12. Secuencia de encendido adecuada a las bobinas.....	9
Figura 13. Motor Nema 17.....	10
Figura 14. Motor Nema 23.....	10
Figura 15. Motor Nema 14.....	10
Figura 16. Controlador de Motor a Pasos.....	11
Figura 17. Controlador TB6612FNG.....	12
Figura 18. Controlador DRV8825.....	12
Figura 19. Batería de Lipo.....	13
Figura 20. Rueda básica.	14
Figura 21. Rueda de caucho.....	14
Figura 22. Placa de control del robot balancín.	15
Figura 23. New Project de Proteus.....	16
Figura 24. Selección de ítems de formato de trabajo de Proteus.....	16
Figura 25. Selección de elementos en Proteus.....	17
Figura 26. Diseño esquemático del circuito PCB del robot balancín.	17
Figura 27. Placa PCB.	18
Figura 28. Diseño de la estructura del robot balancín.	18
Figura 29. Diseño de base inferior de batería de lipo.....	19
Figura 30. Descarga de EZ- GUI.....	19
Figura 31. Conectividad de la aplicación con el módulo bluetooth.....	20
Figura 32. Valores PID del robot balancín.....	20
Figura 33. Calibración del SetPoint del robot balancín.....	21
Figura 34. Control de desplazamiento del robot balancín.....	21
Figura 35. Placa electrónica de control del robot balancín.....	22

Figura 36. Placas base en acrílico.....	23
Figura 37. Colocación de los motores en base inferior.....	23
Figura 38. Base de placa PCB.....	24
Figura 39. Armado de torre.....	24
Figura 40. Base de la batería.....	25
Figura 41. Montaje de las bases del robot balancín.....	25
Figura 42. Estructura del robot balancín.....	25
Figura 43. Freeware de control.....	26
Figura 44. Funcionamiento de un PID.....	27
Figura 45. Estabilización del robot balancín.....	28
Figura 46. Pruebas de funcionamiento.....	29
Figura 47. Proceso completo del robot balancín.....	29
Figura 48. Programa Freeware.....	44

ÍNDICE DE TABLAS

Tabla 1 Características del Arduino nano.....	4
Tabla 2 Características del Arduino Uno.....	5
Tabla 3 Las características del Arduino Mega.....	6
Tabla 4 Características técnicas de módulos de Bluetooth.....	9
Tabla 5 Características de Driver A4988.....	11
Tabla 6 Características del Driver TB6612FNG.....	12
Tabla 7 Características de Driver DRV8825.....	13
Tabla 8 Batería de lipo.....	13

INDICE DE ANEXOS

Anexo 1: Carcasa del robot	36
Anexo 2: Plano de base de motor y batería	37
Anexo 3: Plano de base de placa de control	38
Anexo 4: Tapa superior del robot balancín	39
Anexo 5: Soportes de los motores a pasos nema 17	40
Anexo 6: Circuito de control del robot balancín	41
Anexo 7: Programación de prueba del PID del robot balancín	42
Anexo 8: Freeware de control e interface con la aplicación EZ-GUI de Wii Master ...	44

INTRODUCCION

La investigación está encaminado a la construcción detallada de un sistema automatizado (robot) que utiliza la técnica de equilibrio con dispositivos que le permitirán desplazarse libremente por el espacio por medio de la utilización de dos ruedas móviles y sensores de giro que le permitirán evitar accidentes o choques con los diversos elementos que existen a su alrededor.

Este tipo de tecnología es uno de los más interesantes dentro del mundo actual de la electrónica y programación automática por medio software Arduino. El cual permite la programación de la placa que lleva su mismo nombre esto ahorra un gran trabajo de programación del área correspondiente al software del dispositivo.

Por otro lado correspondiente a hardware es factible la creación del robot ya que los elementos electrónicos son de relativa fácil adquisición en el mercado nacional lo que me permitirá tener un conocimiento amplio del mercado de los dispositivos disponibles en la ciudad.

Al diseñar el circuito electrónico se incorpora varios conocimientos de electrónica y el uso de softwares especializados como por ejemplo el proteus y AutoCAD.

Este tipo de robot es utilizado en operaciones de alta precisión para transportar elementos radiactivos o médicos, por ejemplo laboratorios. Esta cualidad lo hace realmente atractivo para su construcción y uso lo que permite adquirir un conocimiento de varios aspectos referentes a la tecnología de la robótica.

ANTECEDENTES

Durante una década los robots se han insertado en diferentes ámbitos de la cultura y sociedad, ocasionando que se utilice en muchas aplicaciones industriales, comerciales y residenciales. Por ejemplo una de estas aplicaciones es el transporte y movimiento de vehículos, donde está orientado este proyecto de titulación y basado en el nuevo segway: de patinete a robot presentado en el CES 2016 (Consumer Electronic Show), que se celebra anualmente en las Vegas. En un artículo de la Revista de robots (2019) detalla este proyecto como “Es la impresionante creación de un pequeño y moderno método de transporte de dos ruedas que te llevara a cualquier parte que desees y no solo eso, también estará muy atento a lo que ocurre a tu alrededor”.

A nivel nacional se ha realizado una tesis Gonzales (2018) denominada “Diseño e implementación del algoritmo de control para un robot balance, usando fuzzy logic en la plataforma de National Instruments”. Que consiste en la programación de robot balance utilizando lógica difusa que será capaz de balancearse y mantener su equilibrio sobre sus ruedas.

De igual manera a nivel internacional encontramos la tesis denominada “Diseño, construcción y control de un robot balancín”, desarrollado por (Pincin, 2016). Este diseño, se construyó, simulo e implemento en un dispositivo robot balancín para aplicaciones y estudio de técnicas avanzadas de control.

Finalmente en el libro de Baichtal (2016) se detalla el “Ball Balancing, el robot equilibrista”. (p.44). que son iguales a los Segway y son capaces de mantenerse en equilibrio sobre sus dos ruedas, pero este robot se mantiene encima de un balón.

Estas investigaciones realizadas y verificadas, aportan con el conocimiento inicial y de soporte para el proyecto de titulación que se pretende realizar, ya que permitirá brindar un buen soporte al diseño, construcción y verificación de funcionamiento del robot balancín.

RESUMEN

Para conseguir el objetivo de mantener el robot balancín en estado de balanceo fue necesario revisar un amplio estudio de la bibliografía sobre la robótica procediendo inicialmente a diseñar las partes mecánicas del robot con ayuda del programa de AutoCAD 2017 en coordenadas tridimensionales para obtener la mayor precisión posible permitiendo imprimir en 3D los soportes del motor nema 17.

Luego se diseñó y se construyó una placa electrónica a la cual se quemó la distribución de pistas para los diferentes elementos electrónicos a hacer utilizados en la placa, entre la más importante podemos citar el Arduino, el cual hace los cálculos respectivos para que el robot se balancee sin perder el equilibrio.

Posteriormente se realizó el modelo mecánico y construcción del mismo con materiales de acrílico y PLA para obtener la precisión requerida por el robot para mantenerse en equilibrio, dentro de este punto se verificó que todos los elementos tanto electrónico y mecánico estén funcionando correctamente.

El siguiente paso fue realizar la integración total de todos los elementos y se procedió a la programación de PID y en app inventor la interface de control frontend de usuario, debido a que no todas las superficies eran uniformes fue necesario realizar varias pruebas de programación hasta conseguir que el robot consiga el balanceo apropiado y no se desplome.

PALABRAS CLAVES:

Arduino, Balanceo, Google mit app inventor, Robótica, Robot, Electrónica, Motores, AutoCAD, Placa Electrónica, Circuitos, Programación, Equilibrio, Sensores, Ruedas

ABSTRACT

In order to achieve the robot to keep in a perfect balanced state, it was necessary to study an extensive literature about robotics, one of the firsts steps was to design the mechanical parts one by one, AutoCAD 2017 program was used in order to achieve this task because its facilities in three-dimensional coordinates that let obtain the most accurate precision in the 3D printing of the “nema” 17 motor mounts.

Once all the components were on place, an electronic general board was designed and burned. I installed the different electronic elements to be used on the robot on this board. Among the most important electronic components I can cite the Arduino element, which is on charge of making the necessary calculations for keeping a perfect balanced.

Subsequently, the mechanical model and its construction were carried out with acrylic materials and “PLA” to obtain the precision required by the robot to stay in balance, within this point it was verified that all the electronic and mechanical elements are working correctly.

The next step was to fully integrate all the elements and we proceeded to establish the “PID” parameters and in the “Inventor App” program design the user frontend control interface, because not all surfaces were uniform, it was necessary to carry out several programming tests until the robot gets the proper balance and doesn't crash.

KEYWORDS

Arduino, Balancing, Google mit app inventor, Robotics, Robot, Electronics, Motors, AutoCAD, Electronic Board, Circuits, Programming, Sensors, Wheels

1. OBJETIVOS

1.1.Objetivo General

Construir un robot balancín operado a través de una aplicación móvil utilizando sensores de giro, para mantener en equilibrio de su sistema mecánico.

1.2.Objetivos Específicos

- Seleccionar la bibliografía relacionada al objetivo de estudio, para selección del modelo del robot más adecuado para el proyecto.
- Diseñar un circuito electrónico para el control del robot, empleado diseño por computadora a través del software Proteus.
- Diseñar la parte mecánica del robot balancín, a través de la utilización del software AutoCAD.
- Configurar la aplicación móvil del robot balancín en la aplicación de Android EZ-GUI
- Comprobar el funcionamiento del Robot balancín, a través de pruebas en superficies planas y ásperas.

2. DESARROLLO O CUERPO PRINCIPAL

2.1. Marco Teórico – Conceptual

2.1.1. La robótica

El objeto de estudio de este proyecto es la robótica que según la revista de robots (2020) afirma que es “una ciencia que aglutina varias ramas tecnológicas o disciplinas, con el objetivo de diseñar maquinas robotizadas que sean capaces de realizar tareas automatizadas o de simular el comportamiento humano o animal, en función de la capacidad de su software”. Un ejemplo de la robótica en el campo laboral es la ilustración de la figura 1 que es un robot que indaga en ámbitos de exploración y reconocimiento de nuevos terrenos nunca antes vistos por la humanidad que adhiere perspectivas aún más mayores para robótica para un mejor proceso tecnológico.



Figura 1. Robot Explorador M6 UGV.
Fuente: Revista de robots (2019).

2.1.2. Robot

Un robot según Ocaña (2015) lo define como “autómata programable capaz de realizar determinadas operaciones de manera automática, en especial tareas repetitivas, resisas o peligrosas” (p.13). En la figura 2 por ejemplo se visualiza un robot Segway con la función de servir como medio de transporte.



Figura 2. Un robot Segway.
Fuente: Revista de robots (2019).

2.1.3. La robótica móvil

La robótica móvil es un área de investigación y desarrollo de alta exigencia., Barrientos (2007) define a la robótica móvil como “un sistema electromecánico capaz de desplazarse de manera autónoma sin estar sujeto físicamente a un solo punto”. (p.14), la figura 3, muestra en un robot su desplazamiento proporcionado mediante dispositivos de locomoción, tales como ruedas, patas, orugas, etc.



Figura 3. Robot Seguidor de línea.
Fuente: Elaboración propia. (2020).

2.1.4. Arduino

Según Arduino.cl (2020) define al Arduino como “una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra”. Esto permite establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla principalmente con cables DuPont.

Tipos de Arduino

Arduino es una de las primeras plataformas de microcontroladores open source en el mundo y se han desarrollado varias versiones de ésta.

Arduino Nano

El Arduino Nano de la figura 4 se caracteriza por ser una placa pequeña, robusta y potente de la misma capacidad que la tarjeta uno que se observa en la figura 5, se destaca sus características mostradas en la tabla 1, sus dimensiones de tamaño es de 0,73 pulgadas de anchura por 1,70 de longitud para proyectos más pequeños y dinámicos.



Figura 4. Arduino Nano.
Fuente: Iberobotics (2020).

Tabla 1*Características del Arduino nano*

Características	Descripción
Microcontrolador	ATMega328
Voltaje de operación	5v
Tensión de entrada (recomendación)	7-12v
Tensión de entrada (límite)	6 – 20v
Pines digitales de E/S	14 (6 son PWM)
Pines de entradas analógicas	8
Corriente DC por pin E/S	40miliamperio
Corriente DC por pin 3.3v	50miliamperio
Memoria flash	32KB
EEPROM	1KB
Frecuencia de trabajo	16MHz
Dimensiones	0.73”x1.70”

Nota: La tarjeta Arduino nano es una placa de desarrollo diseñada para proyectos electrónicos y de robótica que trabaja en conjunto con el software Arduino. Fuente Iberobotics. (2020).

Arduino Uno

Es la plataforma más conocida y la primera que salió al mercado de la marca Atmel que se observa en la figura 5, con circuitería de soporte, que incluye, reguladores de tensión, un puerto USB que permite programar el microcontrolador desde cualquier PC de manera cómoda y también hace pruebas de comunicación con el propio chip. En la tabla 2 se detalla las características más influyentes que permiten el trabajo de operación del Arduino.



Figura 5. Arduino Uno.
Fuente: Iberobotics (2020).

Tabla 2

Características del Arduino Uno

Características	Descripción
Microcontrolador	ATmega328
Voltaje de funcionamiento	5 V
Pines I / O digitales	14 (de los cuales 6 proveen salida PWM)
Pines de entradas análogas	6
Corriente DC por cada pin I / O	40 mA
Corriente DC en el pin de 3.3 v	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz

Nota: Contiene todo lo necesario para apoyar el micro controlador; basta con conectarlo a un ordenador con un cable USB o con un adaptador de CA o una batería (Vcc). Fuente: Iberobotics. (2020).

Arduino Mega

El Arduino Mega es una placa de mayor capacidad de toda la familia Arduino, como detalla la tabla 3 que tiene un microcontrolador Atmega 2560 y que se observa en la figura 6. Forma parte de proyectos de Arduino que involucra una gran capacidad de trabajo como por ejemplo el control de un número mayor de motores por sus pines PWM o un número mayor de pines que se requieran en trabajos de mayor cobertura que las dos placas de la figuras 4 y 5 no puedan ejercer.



Figura 6. Arduino Mega.
Fuente: Iberobotics (2020).

Tabla 3

Las características del Arduino Mega

Características	Descripción
Microcontrolador	ATmega2560
Voltaje de funcionamiento	5 V
Pines I / O digitales	54 (de los cuales 15 proveen salida PWM)
Pines de entradas análogas	16
Corriente DC por cada pin I / O	40 mA
Corriente DC en el pin de 3.3 v	50 mA
Memoria Flash	256 KB (ATmega328) de los cuales 8 KB son utilizados por el bootloader
SRAM	8 KB (ATmega328)
EEPROM	4 KB (ATmega328)
Velocidad de reloj	16 MHz

Nota: El Arduino mega es ideal para proyectos de electrónica y robótica avanzados, destacando por su elevada cantidad de pines de entrada y salida y sus 4 puertos UART por hardware. Fuente: Iberobotics (2020).

Módulos de Arduino

Arduino al ser una plataforma de software y de hardware libre, cuenta con el apoyo activo de los fabricantes de dispositivos electrónicos como se observa en la figura 7, se producen módulos especiales de varias características, ampliando la funcionalidad básica del controlador.



Figura 7. Módulos de Arduino.
Fuente: Lopacan (2020).

2.1.5. Sensores

De acuerdo con Valencia (2015) define a los sensores como “dispositivos capaces de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas”. (p.35).

Sensores de Inercia

En la robótica, los sensores los utilizamos para captar información según la providencia del entorno donde se encuentre el sensor para medir diferentes magnitudes derivadas. Según Berbel (2008) define los sensores de inercia como “aquel que se utiliza para medir las variaciones de velocidad en un cuerpo, que pueden ser lineales (medidas por los acelerómetros) o angulares (medidas por los giróscopos)”. (p.5).

Tipos de sensores de Inercia

Módulo PMU 6050

Naylampmechatronics (2020) afirma que “el MPU6050 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”, como se observa en la figura 8.

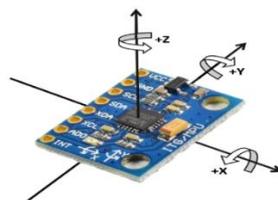


Figura 8. Ejes de Libertad.
Fuente: Naylampmechatronics (2020).

Módulo PMU 6500

Según Invensense (2020) los fabricantes del MPU6500 es más nuevo que los sensores IMU anteriores, pero no es tan popular y lo definen como “un solo chip que integra un acelerómetro de 3 ejes, un giroscopio de 3 ejes y un Digital Motion Processor™ (DMP) integrado en un pequeño paquete QFN de 3 mm x 3 mm x 0.9 mm”. Este módulo se muestra en la figura 9.

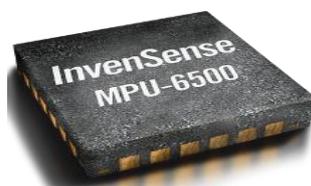


Figura 9. Sensor MPU 6500.
Fuente: Invensense (2020).

Módulo ICM20602

Esta es una de las IMU más nuevas de Invensense adecuadas para su uso en drones y presenta sensores mucho más precisos con menos ruido. También puede obtener lecturas de giroscopio a 32Khz, por lo que será una excelente opción para futuros controladores de vuelo que tengan más potencia de procesamiento como las placas F4 o F7, mostrado en la figura 10.



Figura 10. Sensor ICM20602.
Fuente: Invensense (2020).

Módulos Bluetooth

Naylampmechatronics (2020) define que “son módulos que nos permite conectar nuestros proyectos con Arduino a un Smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial”. Esenciales para todo tipo de práctica que se requiera conectividad tal como se muestra en la figura 11 y la tabla 4, parten de características muy variadas para poder trabajar con mayor sofisticación

Tipos de Módulos Bluetooth

Módulos Bluetooth HC – 05 y HC - 06

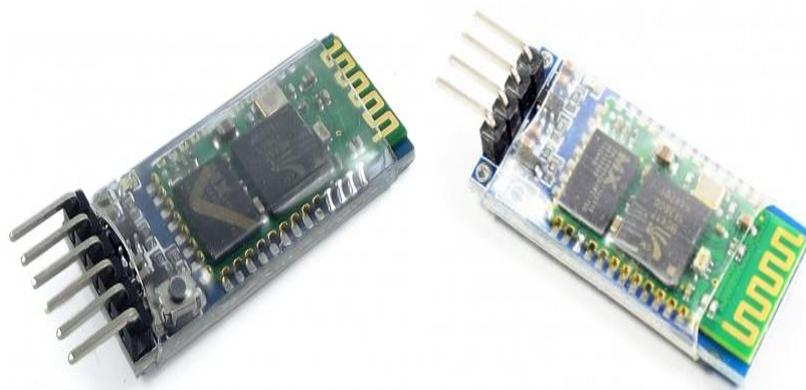


Figura 11. Módulos Bluetooth.
Fuente: Naylampmechatronics (2020).

Tabla 4

Características técnicas de Módulos de Bluetooth

Características	Descripción
Compatible	Bluetooth V2.0.
Voltaje de alimentación	3.3VDC – 6VDC.
Voltaje de operación	3.3VDC.
Baud rate ajustable	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Baud rate por defecto	9600
Tamaño	: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
Corriente de operación	< 40 mA
Corriente modo sleep	< 1mA

Nota: La comunicación Bluetooth se da entre dos tipos de dispositivos: un maestro y un esclavo Fuente: Naylampmechatronics (2020).

2.1.6. Motores paso a paso

Los motores eléctricos son dispositivos que transforman energía eléctrica en energía mecánica. Un motor paso a paso se compone de:

Una serie de imanes que tiran del rotor en una serie de rotaciones parciales, llamadas pasos que perfectamente son controladas. Estas rotaciones parciales resultan ideales para controlar y medir con precisión los giros del motor, lo que es imposible con un motor ordinario de corriente directa.

(Baichtal, 2015, p69).

En la figura 12 se muestra las etapas que un motor a pasos ejecuta parcialmente.

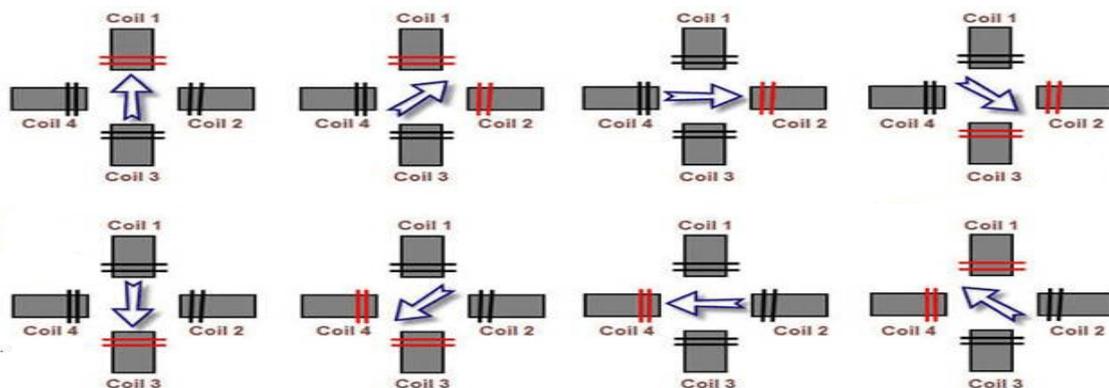


Figura 12. Secuencia de encendido adecuada a las bobinas. Fuente: Luisllamas (2016).

Tipos de Motores a pasos

Motor Nema 17

Según Polulu Robotics & Electronics (2020) define que es un “híbrido de tamaño nema 17 que puede usarse como motor paso a paso unipolar o bipolar y tiene un ángulo de paso de 1.8° (200 pasos / revolución). Cada fase consume 1.2 A a 4 V, lo que permite un par de retención de 3.2 kg-cm (44 oz-in)”. En la figura 13 se visualiza el motor nema 17



Figura 13. Motor Nema 17.
Fuente: Polulu (2020).

Motor Nema 23

Según Polulu Robotics & Electronics (2020) define que es un “híbrido de alto torque que tiene un ángulo de paso de 1.8° (200 pasos / revolución). Cada fase consume 1 A a 5,7 V, lo que permite un par de retención de 4 kg-cm (55 oz-in)”. En la figura 14 se visualiza el motor nema 24



Figura 14. Motor Nema 23.
Fuente: Polulu (2020).

Motor Nema 14

En esta clase de motor a diferencia de las dos anteriores en su capacidad Polulu Robotics & Electronics (2020) define que es un “motor paso a paso bipolar híbrido que tiene un ángulo de paso de 1.8° (200 pasos / revolución). Cada fase consume 500 mA a 10 V, lo que permite un par de retención de 1 kg-cm (14 oz-in)”. En la figura 15 se visualiza el motor nema 14



Figura 15. Motor Nema 14.
Fuente: Polulu (2020).

Controladores de Motores a pasos

Según Jiménez (2014) define como “sistemas de control que se programa para llevar una determinada acción a cabo en los diferentes circuitos que sea imprescindible el control de motores a pasos”.

Tipos de controladores de Motores de pasos a pasos

Controlador A4988 de Motor a Pasos

Según Allegro (2020) define este controlador como “controlador de motor de micro pasos completo como se observa en la figura 16, con traductor incorporado para un fácil operación” (p1).



Figura 16. Controlador de Motor a Pasos.
Fuente: Allegro (2020).

Diseñado para operar motores paso a paso bipolares completos ,medio ,cuarto, octavo y modos de decisexto paso, con una capacidad de salida hasta 35 V y 2 A, como se detalla en la tabla 5, este incluye una corriente fija de tiempo de apagado regulador que tiene la capacidad de operar lento o mixto modo de decaimiento.

Tabla 5

Características de Driver A4988

Características	Descripción
Voltaje de operación	8 a 35v
Máxima corriente por bobina	2 Amperios
Capacidad	“micro-stepping”
Interfaz	2 pines con el microcontrolador: Pines STEP y DIRECTION
Diferentes resoluciones	Paso completo, 1/2 de paso, 1/4 de paso, 1/8 de paso y 1/16 de paso
Control de corriente	ajustable mediante un potenciómetro en el módulo
Protección	por sobre-temperatura, bajo voltaje, corto y otras condiciones de error comunes

Nota. Datos técnicos del controlador del motor a pasoso. Fuente: geekfactory (2020).

Controlador TB6612FNG de Motores a Pasos

Según Llamas (2016) define el TB6612FNG como “un controlador (driver) de motores que nos permite manejar dos motores de corriente directa desde Arduino, variando tanto la velocidad como el sentido de giro”. Este controlador se muestra en la figura 17 y sus características principales en la tabla 6.



Figura 17. Controlador TB6612FNG.
Fuente: LuisLlamas (2016).

Tabla 6

Características del Driver TB6612FNG

Características	Descripción
Controlador Conducción	controlador de motor Dual tipo Puentes H dos motores de CD o un motor paso a paso bipolar
Tensión recomendada del motor	4.5 V a 13.5 V (puede operar hasta 2.5 V con un rendimiento reducido)
Voltaje de alimentación (VCC)	2.7 V a 5.5 V
Máximo de corriente de salida	3 A por canal
Corriente de salida continua	1 A por canal (puede conectarse en paralelo para entregar 2 A en forma continua)
Frecuencia máxima de PWM	100 kHz

Nota. Datos técnicos del controlador del motor a pasos. Fuente: geekfactory (2020).

Controlador DRV8825 de Motores a Pasos

Esta placa que se muestra en la figura 18 es el driver DRV8825 para Motores paso a paso bipolares y según Llamas (2016) lo describe como “controladores (drivers) que simplifican el manejo de motores paso a paso desde un autómata o procesador como Arduino”, sus características técnicas se detallan en la tabla 7.

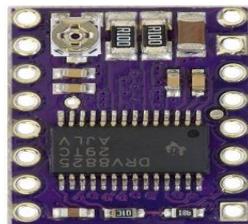


Figura 18. Controlador DRV8825.
Fuente: Luisllamas (2016).

Tabla 7
Características de Driver DRV8825

Características	Descripción
Voltaje de operación	8.2 - 45 V
Corriente de salida	1.5A por fase (A2) 2.2A (A3)
Voltaje lógico mínimo	2.5V (Max: 5.25V)
Microstepping	full, 1/2, 1/4, 1/8, 1/16 y 1/32
Protección	Ninguna

Nota. Datos técnicos del controlador del motor a pasoso. Fuente: geekfactory (2020).

2.1.7. Batería de polímero de litio

Es un artefacto compuesto por celdas electroquímicas capaces de convertir la energía química en su interior en energía eléctrica, mediante la acumulación de corriente alterna.

Esta clase de batería que se ilustra en la figura 19 es de última generación en el mundo de radio control, son baterías robustas que contiene más voltaje, amperios hora y mejor descarga. En la tabla 8 se muestra detallada las características de la batería de lipo.



Figura 19. Batería de Lipo.
 Fuente: Elaboración propia. (2020).

Tabla 8

Batería de lipo 3s 11,1v 2200mah Turnigy 35-70c Robot Drone

Características	Descripción
Capacidad mínima	2200 mAh
Configuración	3S1P / 11.1v / 3Cell
Descarga constante	35C
Pico de descarga	10 seg : 70C
Peso del paquete	185
Tamaño del paquete	105 x 35 x 26 mm
Enchufe de carga	JST-XH
Tapón de descarga	XT60

Nota. Datos técnicos de la batería de lipo. Fuente: Electrostore (2020).

De esta manera, sirven para alimentar distintos circuitos eléctricos, dependiendo de su tamaño y potencia.

2.1.8. Ruedas

Tipos de Ruedas

Las cuatro siguientes son configuraciones bastantes habituales dentro de la robótica creativa:

Rueda básica

“Las ruedas de la figura 20 son ruedas sencillas que resultan estupendas para aficionados cuyo presupuesto es muy ajustado para aplicaciones básicas” (Baichtal, 2016). Ya que sus características incorporan cubiertas ligeras de goma con banda de rodadura.

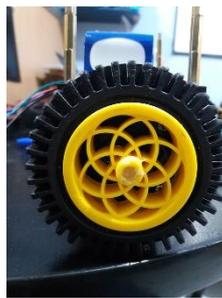


Figura 20. Rueda básica.
Fuente: Elaboración propia. (2020).

Ruedas de caucho

Según Baichtal (2016) este tipo de ruedas son “buenas para las amortiguaciones, para trabajos más sofisticados que las ruedas básicas por su incorporación robusta y versátil que unas simples ruedas de goma”



Figura 21. Rueda de caucho.
Fuente: Gayner (2020).

2.2. Procedimiento – Metodología

2.2.1. Diseño

Primeramente en el diseño se recolecto información de diferentes autores especialistas en el tema de robótica para después catalogarlo en documentación más relevante obteniendo datos más esenciales en el proyecto.

Una vez elegido la información correspondiente con todos los datos más relevantes y esenciales se prosigue al diseño de la parte electrónica y mecánica del robot balancín.

Además, se diseña la aplicación con el cual se controlara el desplazamiento del robot balancín por vía bluetooth para poder movilizarlo en línea recta adelante o atrás en diferentes tipos de tierras que se recurra probar su funcionamiento de estabilidad.

Diseño electrónico

La parte electrónica del robot está conformado de una placa PCB como se observa en la figura 22, que adhiere todos los componentes electrónicos del robot que posteriormente será el cerebro principal para ejercer desde ahí los diferentes procesos correspondientes del robot.

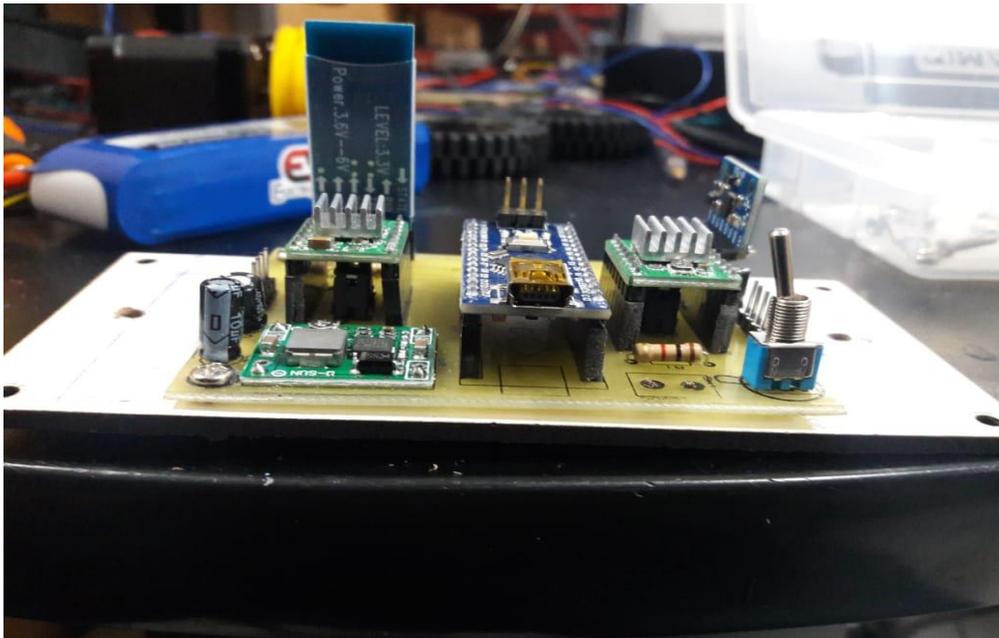


Figura 22. Placa de control del robot balancín.
Fuente: Elaboración propia. (2020).

El software utilizado en el diseño de la placa del PCB es en Proteus Design Suite donde se desarrolló el impreso PCB con el siguiente proceso:

1. Se abre el programa Proteus 8 profesional y se nos amostrara una página principal donde escogemos new Project como se muestra en la figura 23, donde se renombrara el título del proyecto y configurarlo para después trabajar en el diseño.

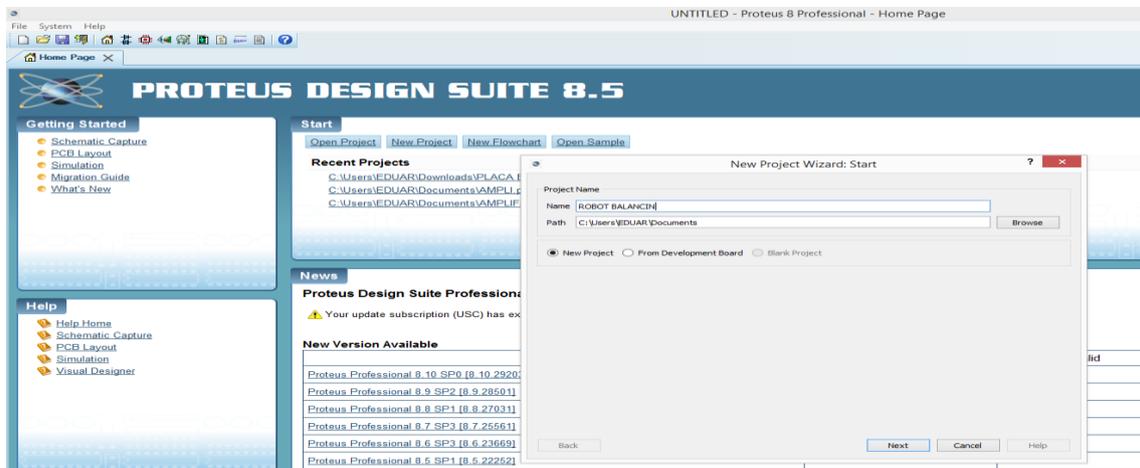


Figura 23. New Project de Proteus.
Fuente: Elaboración propia. (2020).

2. Se configura el formato de trabajo del proyecto y diferentes características que según se necesite se dará clic en los diferentes ítems, en caso contrario se lo deja por default y solo se dará clic en siguiente como se muestra en la figura 24.

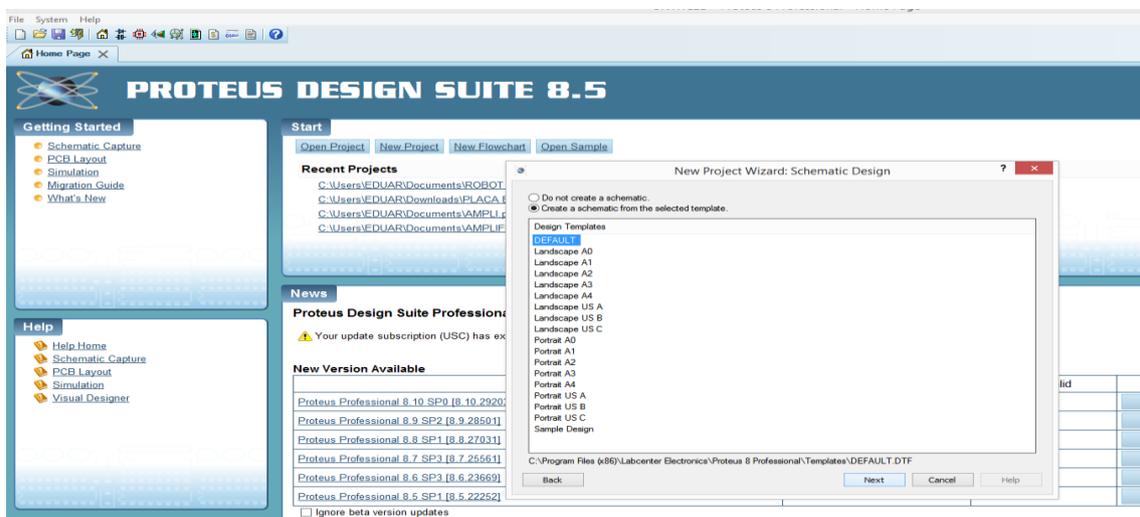


Figura 24. Selección de ítems de formato de trabajo de Proteus.
Fuente: Elaboración propia. (2020).

- Una vez establecido los parámetros con los que se va a trabajar saldrá una nueva ventana de trabajo donde daremos clic pick from libraries y se seleccionara los elementos correspondientes para el funcionamiento del robot balancín, indicado en la figura 25.

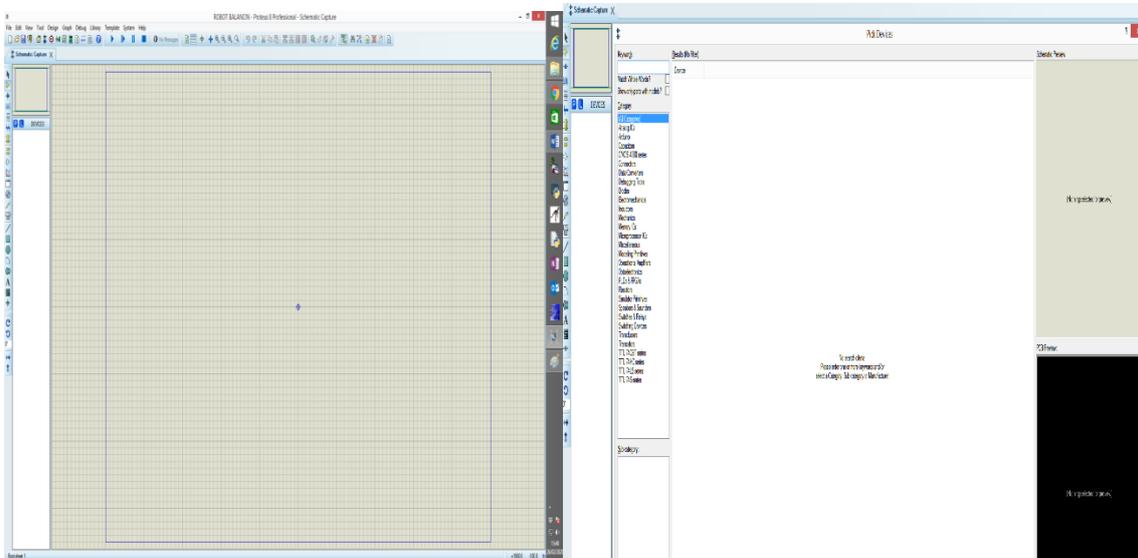


Figura 25. Selección de elementos en Proteus.
Fuente: Elaboración propia. (2020).

- Seleccionado todos los componentes necesarios para el funcionamiento del robot se prosigue a diseñar el circuito esquemático del robot balancín quedando como resultado la muestra de la figura 26, el diseño preciso para que pueda ejercer el siguiente proceso que es el diseño del PCB.

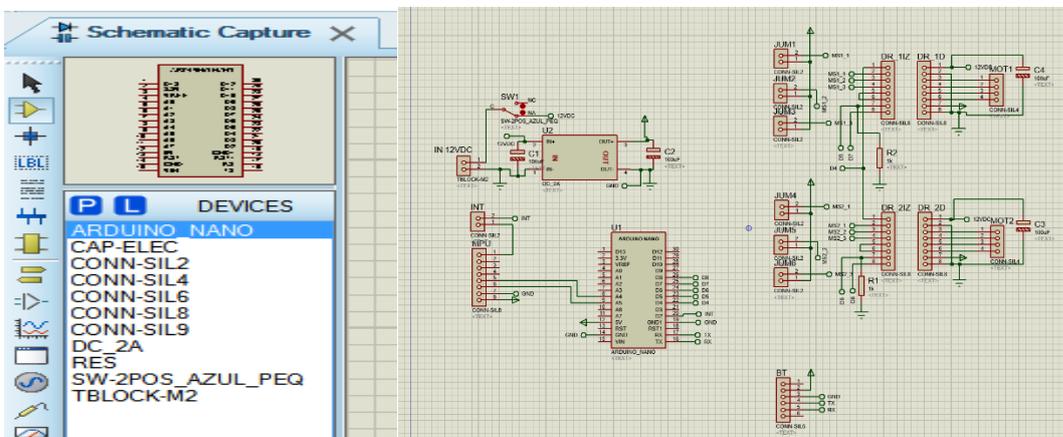


Figura 26. Diseño esquemático del circuito PCB del robot balancín.
Fuente: Elaboración propia. (2020).

5. En esta etapa final se prosigue en PCB Layout a poner los elementos establecidos por el circuito esquemático que se muestra en la figura 26, para luego transformarlo todo en un circuito PCB como se observa en la figura 27 para imprimirlo.

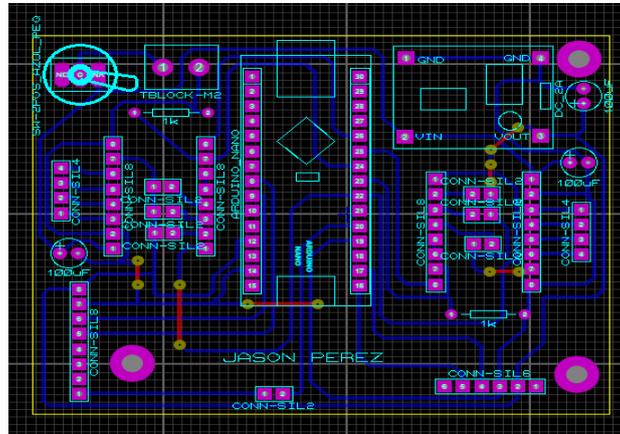


Figura 27. Placa PCB.
Fuente: Elaboración propia. (2020).

Diseño Mecánico

Para el diseño mecánico del robot balancín se utilizó AutoCAD donde se diseñó 4 bases que formaran una torre donde su longitud y ancho se determinara según los elementos que se adhieren por sección como se muestra en la figura 28 y además un diagrama en bloques de las secciones de la bases que forman una torre.

A continuación se ilustra en la imagen posterior el diseño que conforma el diagrama de bloques que se muestra en el anexo 1.

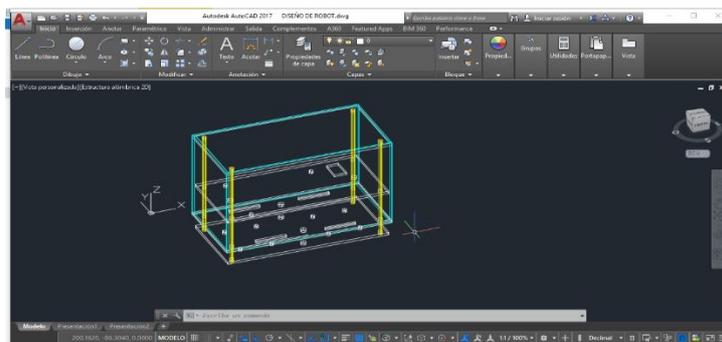


Figura 28. Diseño de la estructura del robot balancín.
Fuente: Elaboración propia. (2020).

Además, tenemos el diseño de las bases como por ejemplo la figura 29, donde se adhiere una batería de lipo a sí mismo en las diferentes bases del robot adhieren diferentes elementos como se muestra anexos 2, 3, 4 y 5 con sus respectivos valores de longitud y ancho de cada base.

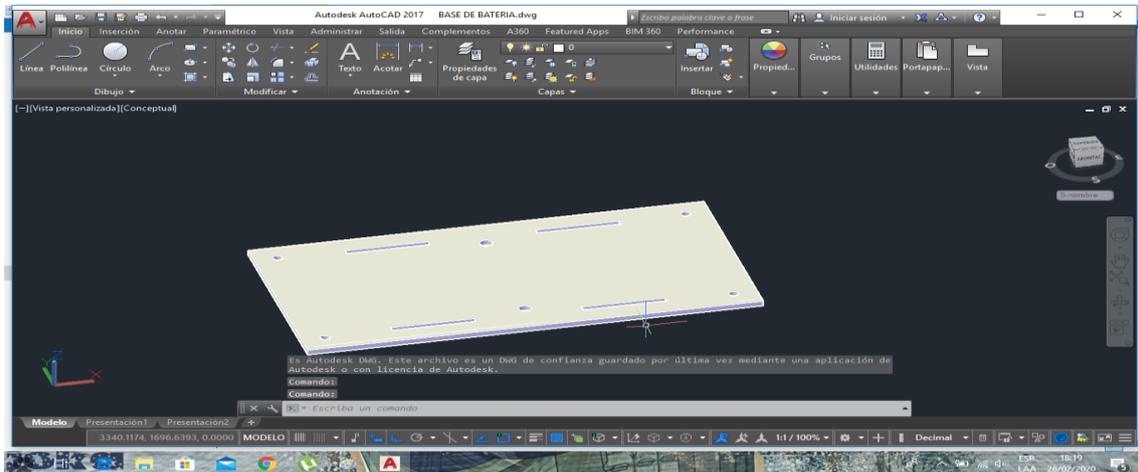


Figura 29. Diseño de base inferior de batería de lipo.
Fuente: Elaboración propia. (2020).

Software de control del robot Balancing

Para el control del robot balancín se utilizó una aplicación Android EZ-GUI que utiliza los datos disponibles de un controlador de vuelo de una manera conveniente. Con el objetivo de poder configurar y determinar la calibración PID correspondiente para un control puntual. Donde sus pasos a configurar son

1. Se dirige a la página principal de Google Play Store donde buscamos EZ-GUI para descargarlo.



Figura 30. Descarga de EZ- GUI.
Fuente: Elaboración propia. (2020).

- Una vez descargado la aplicación se procede a iniciar y enlazar con el bluetooth del circuito del robot para su comunicación como se ilustra en la figura 31.

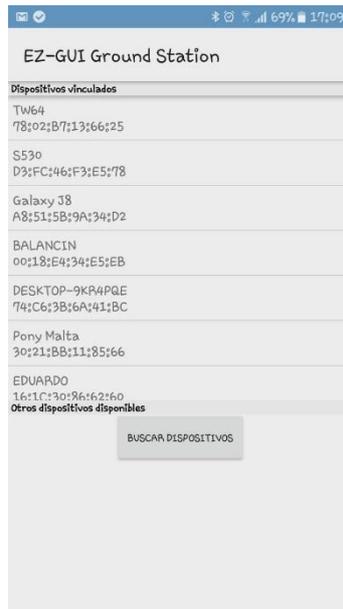


Figura 31. Conectividad de la aplicación con el módulo bluetooth.
Fuente: Elaboración propia. (2020).

- Conectado la aplicación con el robot por bluetooth se empieza a configurar el PID para el proceso de adquisición de datos que se van enviar de la aplicación a la tarjeta del Arduino para mantener el robot balanceado.

PID				
	P	I	D	Tasa
ROLL	5.0	0.250	50	0.00
PITCH	10.0	0.250	50	
YAW	3.0	0.050	50	0.00
ALT	3.0	0.250	100	
Pos	2.44	1.0		
PosR	1.0	2.50	0.250	
NavR	5.0	2.00	0.100	
LEVEL	12.0	0.025	90	
MAG	2.0			

Figura 32. Valores PID del robot balancín.
Fuente: Elaboración propia. (2020).

- Configurado el PID en la aplicación se inicia a calibrar el SetPoint o la referencia de del robot cuando está parado a 90° para su respectivo funcionamiento como se muestra en la figura 33, y luego iniciar el control como se muestra en la figura 34.

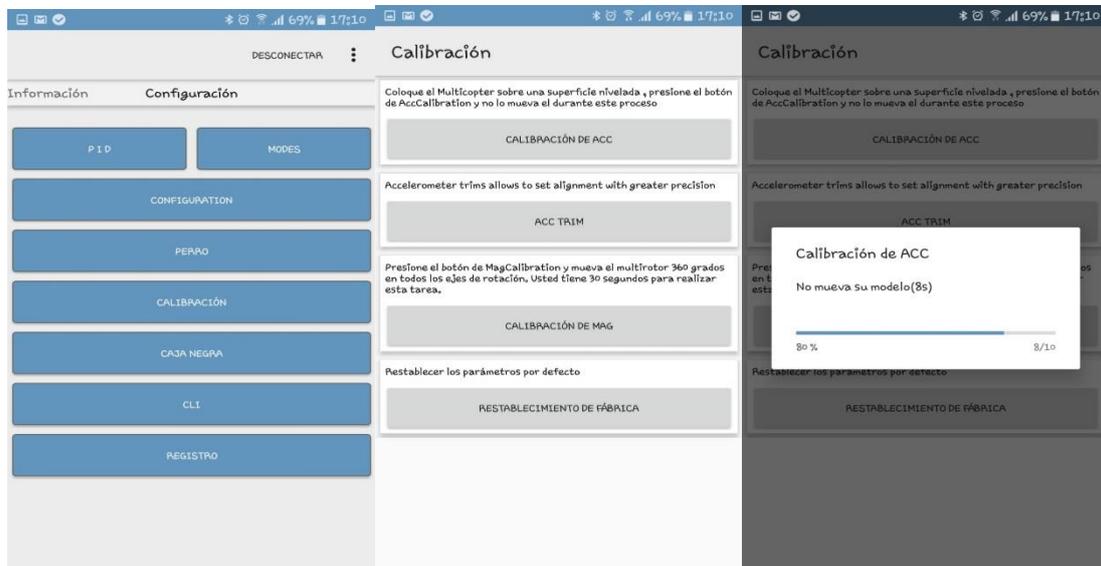


Figura 33. Calibración del SetPoint del robot balancín.
Fuente: Elaboración propia. (2020).

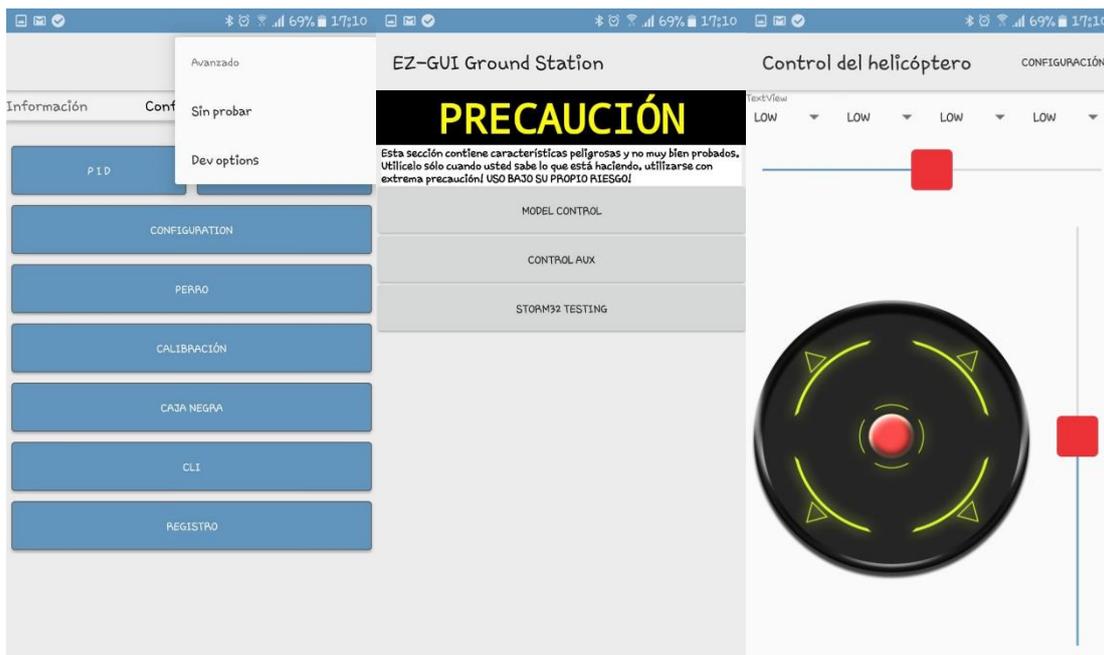


Figura 34. Control de desplazamiento del robot balancín.
Fuente: Elaboración propia. (2020).

2.2.2. Construcción

Para la construcción del robot balancín que es la etapa final y de tener culminado el diseño de la parte electrónica y mecánica del robot balancín teniendo en cuenta que también se diseñó la aplicación para controlarlo por vía bluetooth la movilidad de desplazamiento se recurre a hacer el siguiente proceso:

Construcción de la placa electrónica

1. Se comienza por la impresión del circuito de control mostrado en la figura 27 en impresora láser. con papel termotransferible para usar el método de transferencia térmica por medio de una plancha para traspasarlo a la baquelita de cobre.
2. En segundo lugar una vez traspasado el circuito a la baquelita proseguimos a quemar la baquelita de cobre con ácido sulfúrico para que coma las partes no remarcadas del circuito para dejar las pistas de conducción.
3. En tercer lugar la limpieza del circuito una vez culminado la acción de quemar el circuito con ácido sulfúrico para comenzar a taladrar los huecos donde van adherirse los elementos electrónicos del robot balancín.
4. Por último la soldadura de todos los elementos electrónicos de la placa PCB que es el control principal del robot balancín, como se muestra en su diseño final en la figura 35.

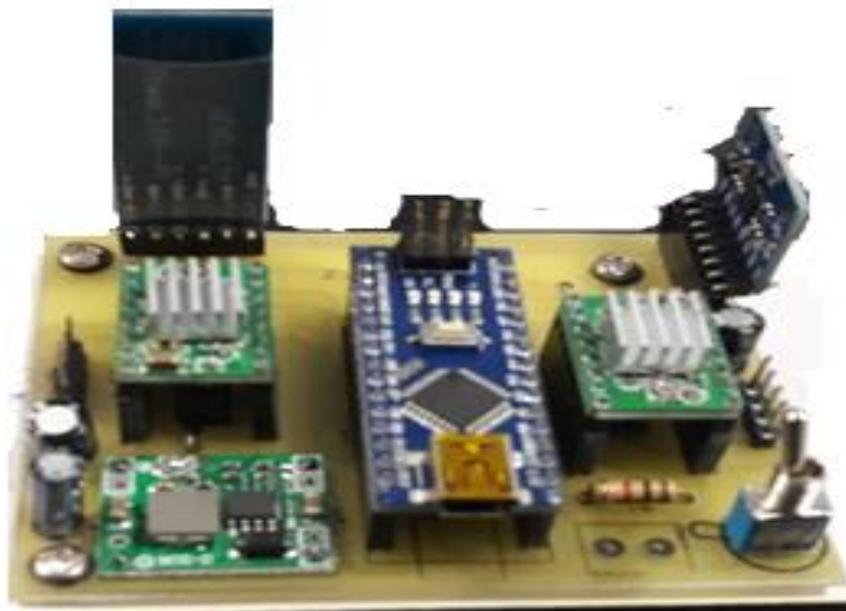


Figura 35. Placa electrónica de control del robot balancín.
Fuente: Elaboración propia. (2020).

Además, una vez culminado la placa electrónica se procederá a la realización de las placas bases en acrílico y la impresión de los soportes para los motores nema 17 en Pla con los datos correspondientes de los planos que se observan en los anexos que conforma la parte mecánica del robot balancín.

Construcción mecánica del robot balancín

La construcción mecánica del robot balancín se prosigue a la unión de las bases como se muestra en el anexo 1, esta estructura se ensambla para formar una torre y para ello se realizara el siguiente proceso:

1. Una vez obteniendo las placas bases como se observa en la figura 36, de material acrílico se prosigue armar la primera base del anexo 2 que se adhiere al motor y a la batería con su respectivo soporte del anexo 5 que se observa en la figura 37.



Figura 36. Placas base en acrílico.
Fuente: Elaboración propia. (2020).



Figura 37. Colocación de los motores en base inferior.
Fuente: Elaboración propia. (2020).

2. En la base del anexo 3 se procede a la colocación de la placa de control con la base como se muestra en la figura 38. Además, la base se adhiere con la anterior como se observa en la figura 39.

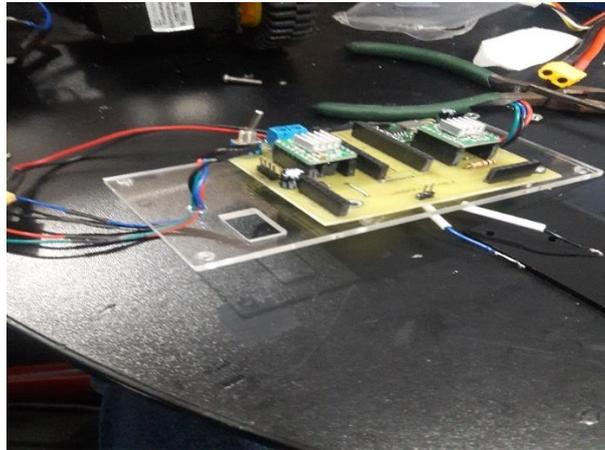


Figura 38. Base de placa PCB.
Fuente: Elaboración propia. (2020).

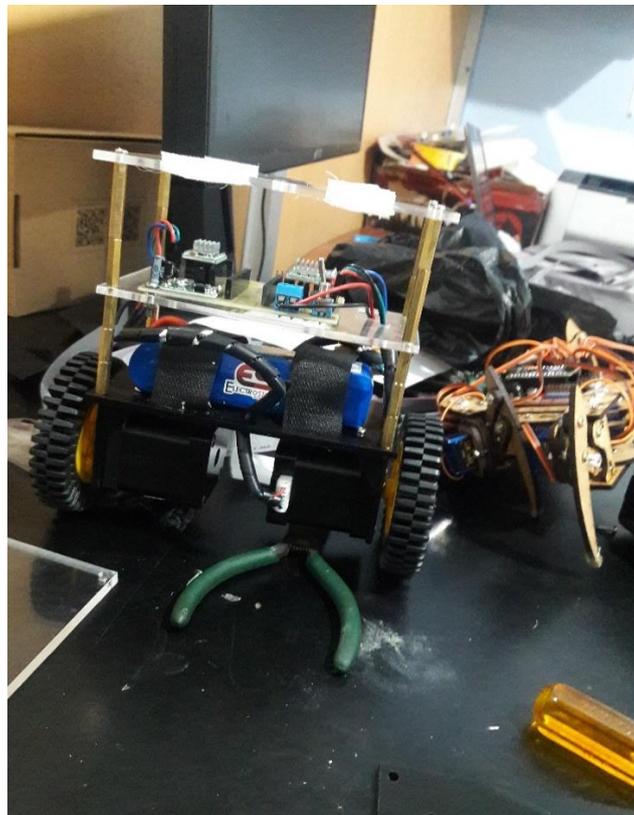


Figura 39. Armado de torre.
Fuente: Elaboración propia. (2020).

3. En la base siguiente del anexo 4 es la tapa que se adhiere al final que protege la placa de control como se muestra en la figura 40, 41 y 42.



Figura 40. Base de la batería.
Fuente: Elaboración propia. (2020).

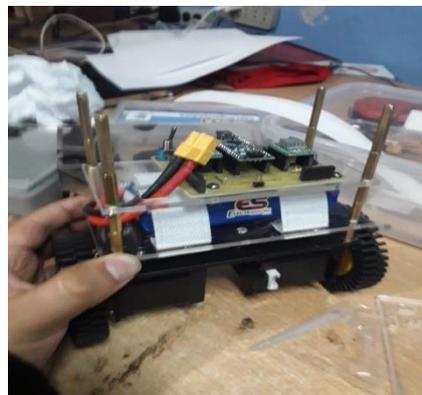


Figura 41. Montaje de las bases del robot balancín.
Fuente: Elaboración propia. (2020).

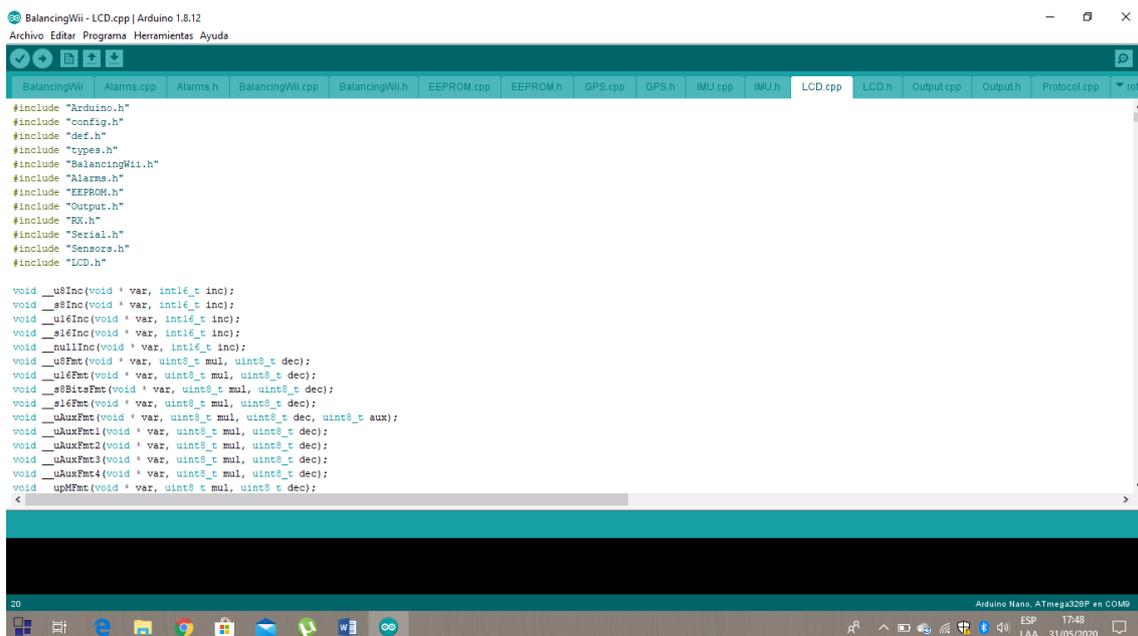


Figura 42. Estructura del robot balancín.
Fuente: Elaboración propia. (2020).

2.2.3. Programación

En esta etapa se concreta las fuentes de información utilizadas para el desarrollo del robot, para lo cual se utilizó un freeware que aporta funcionalidades de control para la configuración del robot balancín para crear un PID que garantice la retroalimentación de un bucle cerrado para que alcance la salida deseada, con la finalidad de proporcionar una estabilidad direccional del robot balancing cuando se controla su desplazamiento en su aplicación móvil EZ-GUI. Por consiguiente, estos pasos a seguir para cualquier tipo de control PID se desarrolla la teoría y práctica de la programación que dan la capacidad de ser utilizadas de varias maneras en prácticas o aplicaciones a realizar.

En la programación inicialmente fue necesario un freeware que permita gestionar una mejor apreciación en la adquisición de datos, su uso, forma para recibir o enviar datos de variables provenientes de los diversos elementos electrónicos como sensores en este caso el MPU6050, y a la vez controlar el robot con dispositivos móviles por medio del bluetooth y circuitos internos en la placa base que le dan la capacidad al robot de ser usado como un instrumento movable y balanceado para que no pierda su equilibrio. En la figura 43 se aprecia el freeware donde se tomó en cuenta el control PID del robot balancín.



```
BalancingWii - LCD.cpp | Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda

BalancingWii | Alarms.cpp | Alarms.h | BalancingWii.cpp | BalancingWii.h | EEPROM.cpp | EEPROM.h | GPS.cpp | GPS.h | IMU.cpp | IMU.h | LCD.cpp | LCD.h | Output.cpp | Output.h | Protocol.cpp | Protocol.h | rotat...

#include "Arduino.h"
#include "config.h"
#include "def.h"
#include "types.h"
#include "BalancingWii.h"
#include "Alarms.h"
#include "EEPROM.h"
#include "Output.h"
#include "RX.h"
#include "Serial.h"
#include "Sensors.h"
#include "LCD.h"

void __u8Inc(void * var, int16_t inc);
void __s8Inc(void * var, int16_t inc);
void __u16Inc(void * var, int16_t inc);
void __s16Inc(void * var, int16_t inc);
void __nullInc(void * var, int16_t inc);
void __u8Fmt(void * var, uint8_t mul, uint8_t dec);
void __u16Fmt(void * var, uint8_t mul, uint8_t dec);
void __s8Fmt(void * var, uint8_t mul, uint8_t dec);
void __s16Fmt(void * var, uint8_t mul, uint8_t dec);
void __uAuxFmt(void * var, uint8_t mul, uint8_t dec, uint8_t aux);
void __uAuxFmt2(void * var, uint8_t mul, uint8_t dec);
void __uAuxFmt3(void * var, uint8_t mul, uint8_t dec);
void __uAuxFmt4(void * var, uint8_t mul, uint8_t dec);
void __uMFmt(void * var, uint8_t mul, uint8_t dec);
```

Figura 43. Freeware de control.
Fuente: Elaboración propia. (2020).

Dado que el freeware de control necesita un control en específico se debe proporcionar valores al PID para poder utilizar correctamente el robot balancín donde la figura 44 representa el PID de cómo funciona y para qué sirve para contar con una visión abarcativa de los diferentes aspectos de esta área de la automatización y control de la robótica

```
void loop() {
  // put your main code here, to run repeatedly:

  mpu6050.update();
  unsigned long tiempoActual;
  tiempoActual = millis();
  dT=(double) (tiempoActual-tiempoAnterior);
  if(dT>=1) {

    error = ref-mpu6050.getAngleX();
    errorde=error-erroran;
    errorin +=(ki*error);
    double U =kp*error+(errorin)+kd*errorde;

    if(U>255){
      U=255;
      digitalWrite(mot16, HIGH);
      digitalWrite(mot27, HIGH);
    }
    if (U<0){
      U=150;
      digitalWrite(mot16, LOW);
      digitalWrite(mot27, LOW);
    }
    analogWrite(mot18, U);
    analogWrite(mot25, U);
    erroran=error;
    tiempoAnterior=tiempoActual;
  }
}
```

Figura 44. Funcionamiento de un PID.
Fuente: Elaboración propia. (2020).

Para finalizar, una vez comprendido de cómo funciona PID asegurando los valores KP, KI, KD para que el robot quede como se indica en la figura 45 estabilizado para después, proseguir con el firmware Balancingwii que ayuda a la interface de la aplicación de Android EZ-GUI para controlar el robot a través de la conexión Bluetooth.

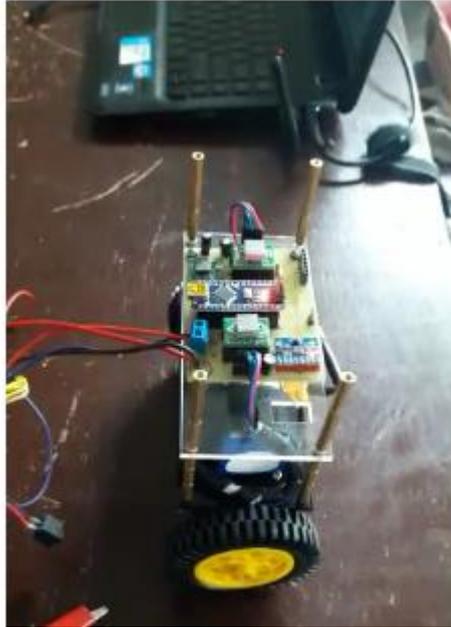


Figura 45. Estabilización del robot balancín.
Fuente: Elaboración propia. (2020).

En el anexo 7 se aprecia la programación del control PID de cómo está constituido y comprobado para su correcto funcionamiento sin ningún inconveniente al comprobar el funcionamiento total con el robot.

2.2.4. Comprobación

Finalmente culminado la programación y la calibración del PID se prosigue a la pruebas del robot balancín, la misma que se observa en la figura 46.



Figura 46. Pruebas de funcionamiento.
Fuente: Elaboración propia. (2020).

Comprobado el correcto funcionamiento de todos los componentes que componen el robot balancín se procede a realizar una prueba total con un vaso de agua en la parte superior como se muestra en la figura 47.



Figura 47. Proceso completo del robot balancín.
Fuente: Elaboración propia. (2020).

2.2.5. Metodología

La metodología que se dispone en este proyecto la investigación exploratoria, que según Arias (2016). “es aquella que se efectúa sobre el tema u objeto desconocido o poco estudiado, por lo que sus resultados constituyen una visión aproximada de dicho objeto, es decir, un nivel superficial de conocimientos” (p.23).

Esta clase de investigación utilizaremos para estudiar un problema que no está claramente definido, por lo que se lleva a cabo para comprenderlo mejor, pero sin proporcionar resultados concluyentes permitiendo que la investigación exploratoria permita obtener una gran cantidad de información valiosa de personas que sean conocedoras del tema.

Así mismo, la revisión documental en la investigación exploratoria consiste en el acercamiento al tema de estudio a través de datos bibliográficos que tengan vínculo directo con el problema de investigación y con este orden de ideas, la exploración permitirá obtener nuevos datos y elementos que puedan conducir a la construcción y pruebas que se realizaran para demostrar la eficacia del tema para obtener la confirmación o anulación de la validez del proyecto.

3. CONCLUSIONES

La selección correcta de la bibliografía ayudo a mejorar el modelo del robot para adherirse a mejoras más eficaces de funcionamiento.

El diseñar las partes del robot en el software mejoro la capacidad de desarrollo más sofisticados en la parte dinámica del robot.

La programación de los datos en Arduino reivindico el funcionamiento del robot para balancearse sin perder el equilibrio.

Android EZ- GUI fue una buena plataforma que permite configurar y ajustar los valores predeterminados para facilitar el control de desplazamiento por medio de Bluetooth.

La comprobación del robot balancín diseñado en software y armado funciono correctamente sin dificultad.

.

4. RECOMENDACIONES

La selección de la información debe ser veraz para no tener ningún inconveniente de datos erróneos al momento de elaborar el proyecto.

Diseñar de manera cautelosa para prevenir desperfectos al adherir las piezas correspondientes del robot.

Programar con datos relevantes para no tener fallas de procesamiento en el robot al momento de arrancar.

Colocar en Android EZ-GUI los valores correspondientes que facilite la comunicación con el respectivo modelo que se desee utilizar para controlarlo.

Probar en todas las superficies disponibles el funcionamiento del robot para tener en cuenta datos relevantes de funcionamiento de desplazamiento del robot.

5. REFERENCIAS

5.1. BIBLIOGRAFÍA

- Torrente, Ó. (2013). Curso de formacion. Mexico: Alfaomega Grupo Editor, S.A. de C.V.
- Blum, J. (2014). Arduino a Fondo. Madrid: Grupo Anaya, S.A.
- Baichtal, J. (2015). Construye tu Robot. Madrid: Grupo Anaya,S.A.
- Gómez, S. (2012). SolidWorks practico 1. Barcelon: Grafo,S.A.
- Mandado, E. (2015). Sistemas electrónicos digitales. Barcelona : Marcombo,Ediciones Tecnicas.
- Ocaña, G. (2015). Aula de robotica . España : Dextra Editorial S.L.
- Arias, F. (2016).El Proyecto de Investigacion . Caracas : Ediciones el Pasillo,C.A.

5.2. REFERENCIA A PÁGINAS WEBS:

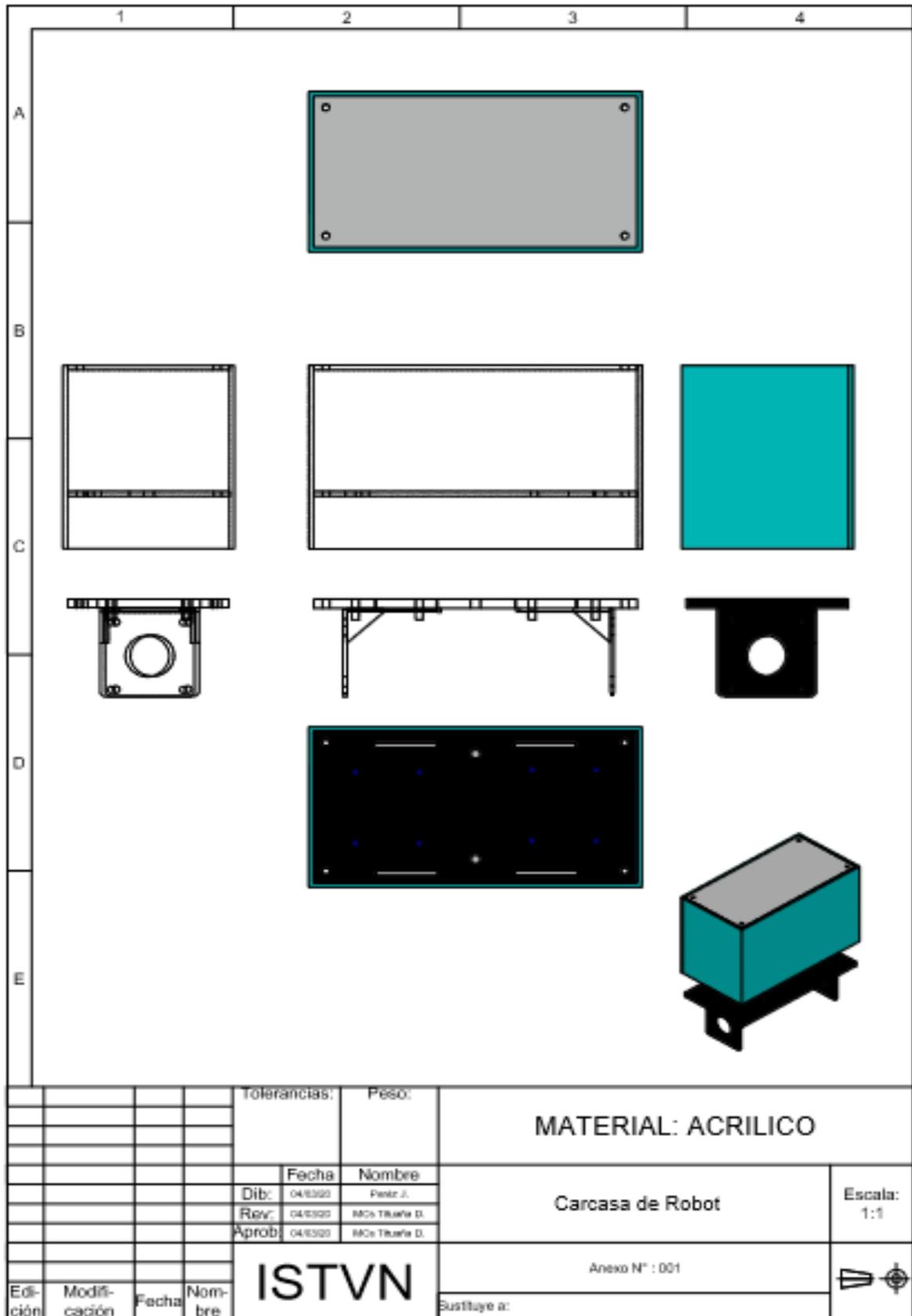
- Flores-Vázquez, C. (2017). Operación remota de un robot móvil usando un teléfono inteligente. (Articulo Científico). Universidad Politécnica Salesiana. Recuperado de <http://dspace.ups.edu.ec/handle/123456789/13819>
- Jacome, K. (2016). Diseño e implementación de un robot móvil soccer utilizando la tarjeta ARDUINO NANO y controlado mediante Bluetooth. (*Tesis de Telecomunicaciones*). Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado de <http://repositorio.ucsg.edu.ec/handle/3317/5440>
- Oña, Á. (2017). Plataforma de robótica móvil para experimento a través del sistema operativo de robots (ros): Desarrollo del meta-sistema operativo. (*Articulo Científico*). Universidad Técnica del Norte, Ibarra. Recuperado de <http://repositorio.utn.edu.ec/handle/123456789/7333>
- Ortega, J. (2017). Construcción de un Robot Móvil Híbrido Omnidireccional. (*Tesis de Electronica*). Universidad Politécnica Salesiana, Quto. Recuperado de <http://dspace.ups.edu.ec/handle/123456789/14048>

- Patricio, E. (2019). Navegación autónoma de un robot móvil omnidireccional en trayectorias predeterminadas con evitación de obstáculos. (*Tesis de Electronica Industrial*). Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial., Ambato. Recuperado de <http://repositorio.uta.edu.ec/jspui/handle/123456789/29173>
- Paúl, T. (2016). Diseño e implementacion de una plataforma robótica con traccion intercambiable. (*Articulo Científico*). Universidad de las Fuerzas Armadas ESPE, Latacunga. Recuperado de <http://repositorio.espe.edu.ec/handle/21000/10581>
- Ricardo, T. (2015). Desarrollo de un sistema de robotica colectiva con procesamimeto centralizado entre dos robots humanoides biolid premiun. (*Artículo Científico*). Universidad de las Fuerzas Armadas ESPE, Sangolqui. Recuperado de <http://repositorio.espe.edu.ec/handle/21000/7800>
- Riccillo, M. L. (2016). Robots con conciencia artificial. (*Ariticulo Científico*). La Universidad Nacional de La Plata, La Plata. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/53445>
- Sanchez, F. (2015). Robot autónomo para recorrer un laberinto. (*Articulo Científico*). La Universidad Tecnológica Nacional de Argentina, Buenos Aires. Recuperado de <http://ria.utn.edu.ar/handle/123456789/3644?show=full>
- Serrano, R. (2019). Desarrollo de un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi. (*Tesis de Telecomunicaciones*). Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado de <http://repositorio.ucsg.edu.ec/handle/3317/13361>
- Suqui, J. (2017). Desarrollo de la interfaz electrónica y eléctrica para el control de un robot humanoide. (*Tesis de Mecátronica*). Universidad UTE, Quito. Recuperado de <http://repositorio.ute.edu.ec/handle/123456789/14645>
- Vaca, W. (2018). Robot móvil para investigación en algoritmos de planeamiento de rutas: sistema de Odometría. *Articulo Científico*. Universidad Técnica del norte, Ibarra. Recuperado de <http://repositorio.utn.edu.ec/handle/123456789/7908>
- Vallejos, P. (2011). Metodología de Diseño de Robots Semi-Pasivos. (*Tesis de de Ingenieria Electrica*). Universidad de Chile, Santiago. Recuperado de <http://repositorio.uchile.cl/handle/2250/102566>

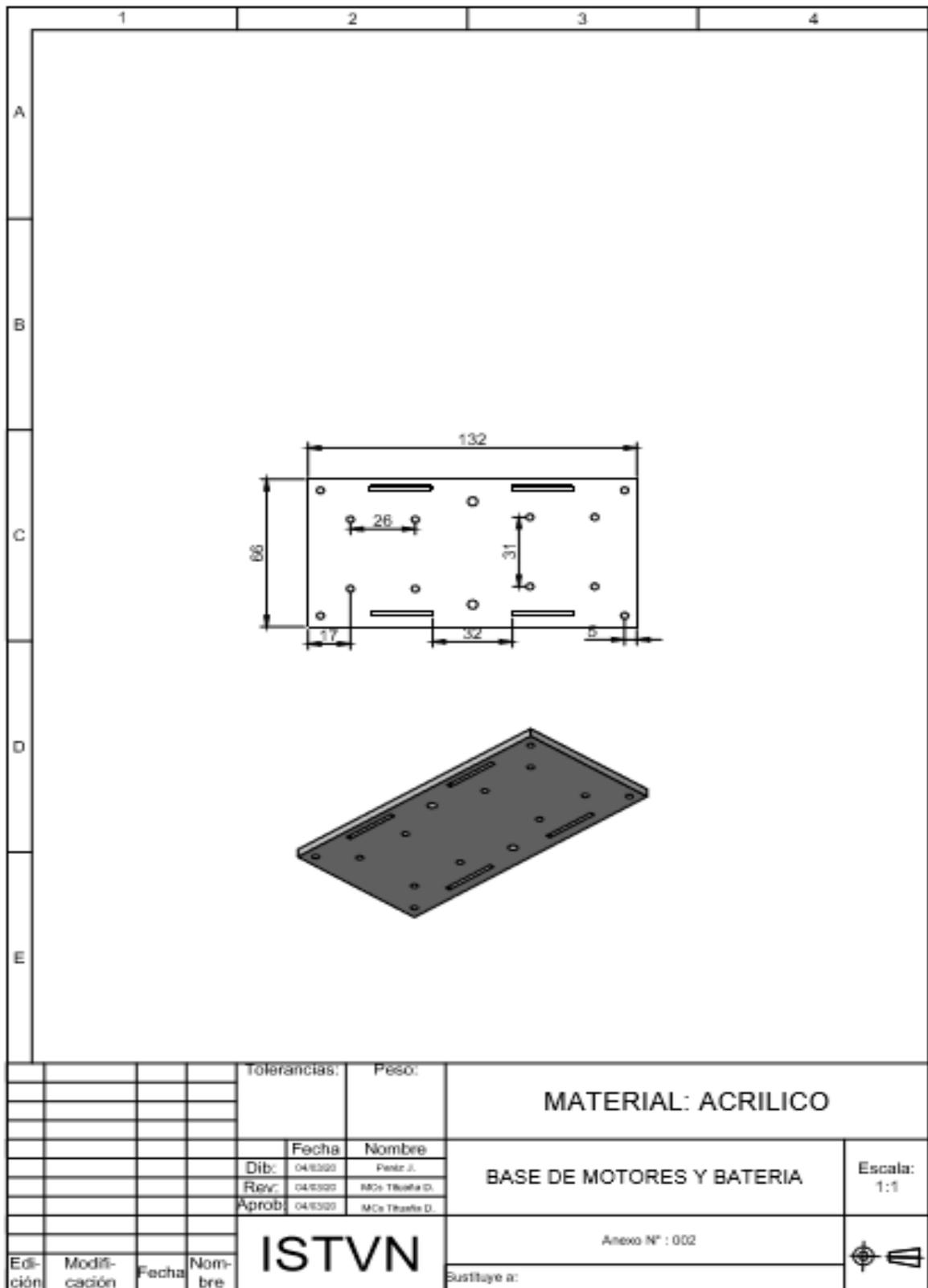
- Andrango, M. (2019). Simulación y ensamblaje de un prototipo para control y navegación de un robot móvil. (*Tesis de de Ingenieria Electrica*). Escuela Politecnica Nacional. Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/9796>.
- Rodríguez, A. (2019). Diseño y simulación de un algoritmo de coordinación entre robots móviles terrestres y cuadricópteros para el enrutamiento en la distribución de materiales dentro de un ambiente industrial. (*Tesis de de Ingenieria Electrica*). Escuela Politecnica Nacional. Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/20387>.
- Zambrano, L. (2017). Diseño e implementación de un sistema destinado a monitorear la temperatura ambiente, integrando computador de placa única raspberry pi, con tecnología móvil. (*Tesis de de Ingenieria de Sistemas*). Pontifical Universidad Catolica del Ecuador. Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/9796>.
- Sandeep. (2019). Robot de equilibrio automático de bricolaje. Recuperado de <https://electricdiylab.com/diy-self-balancing-robot/>

6. ANEXOS

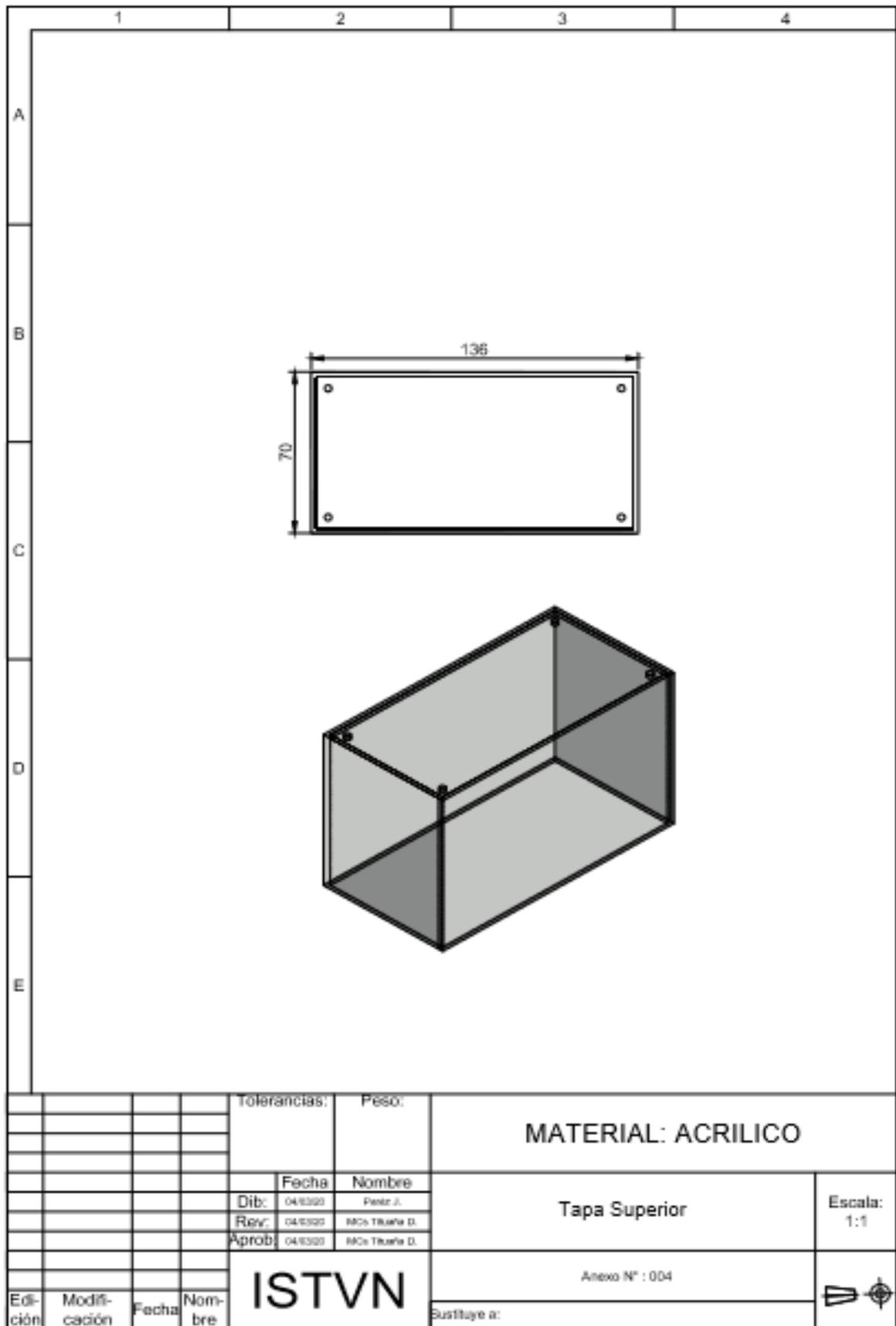
Anexo 1: Carcasa del robot



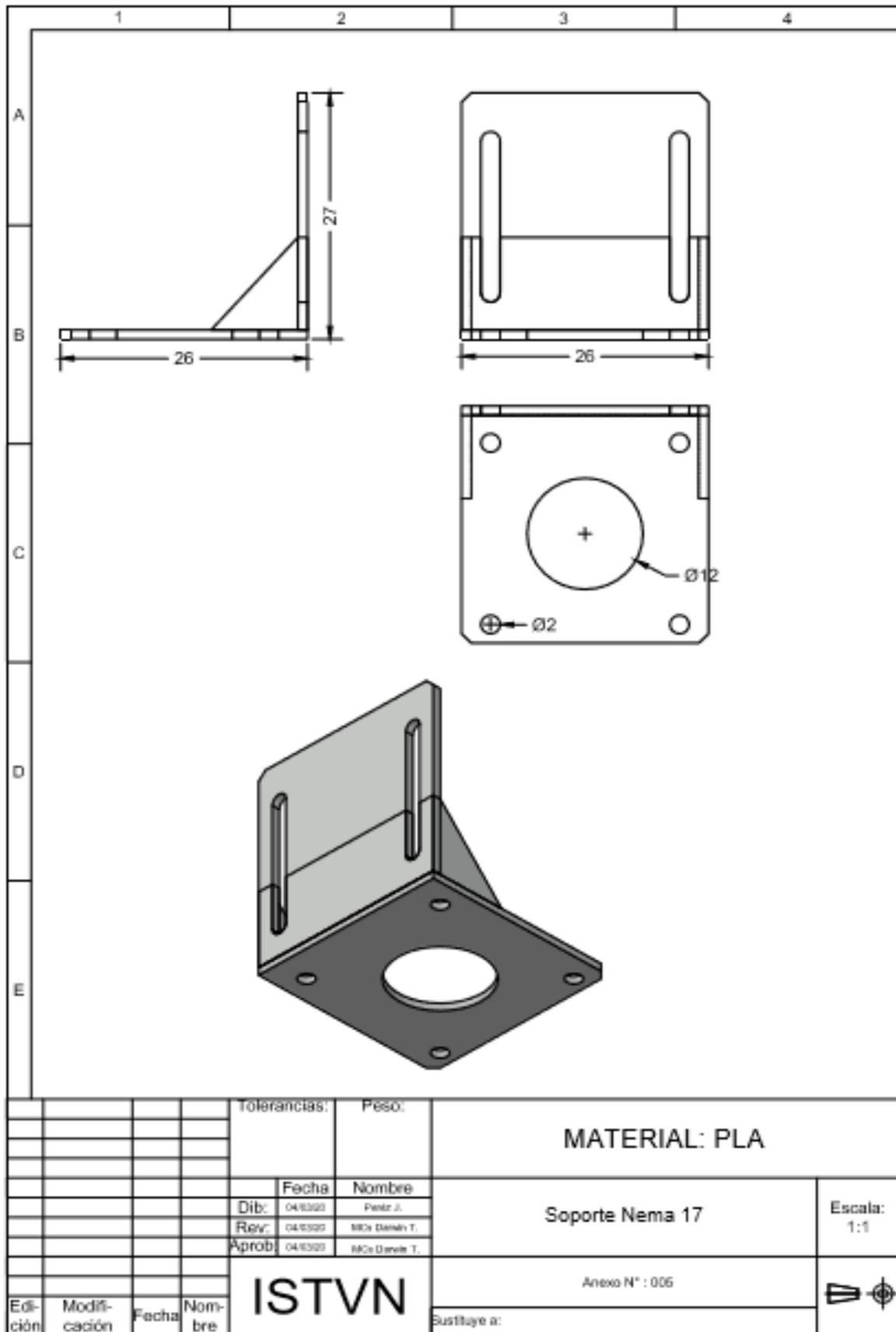
Anexo 2: Plano de base de motor y batería



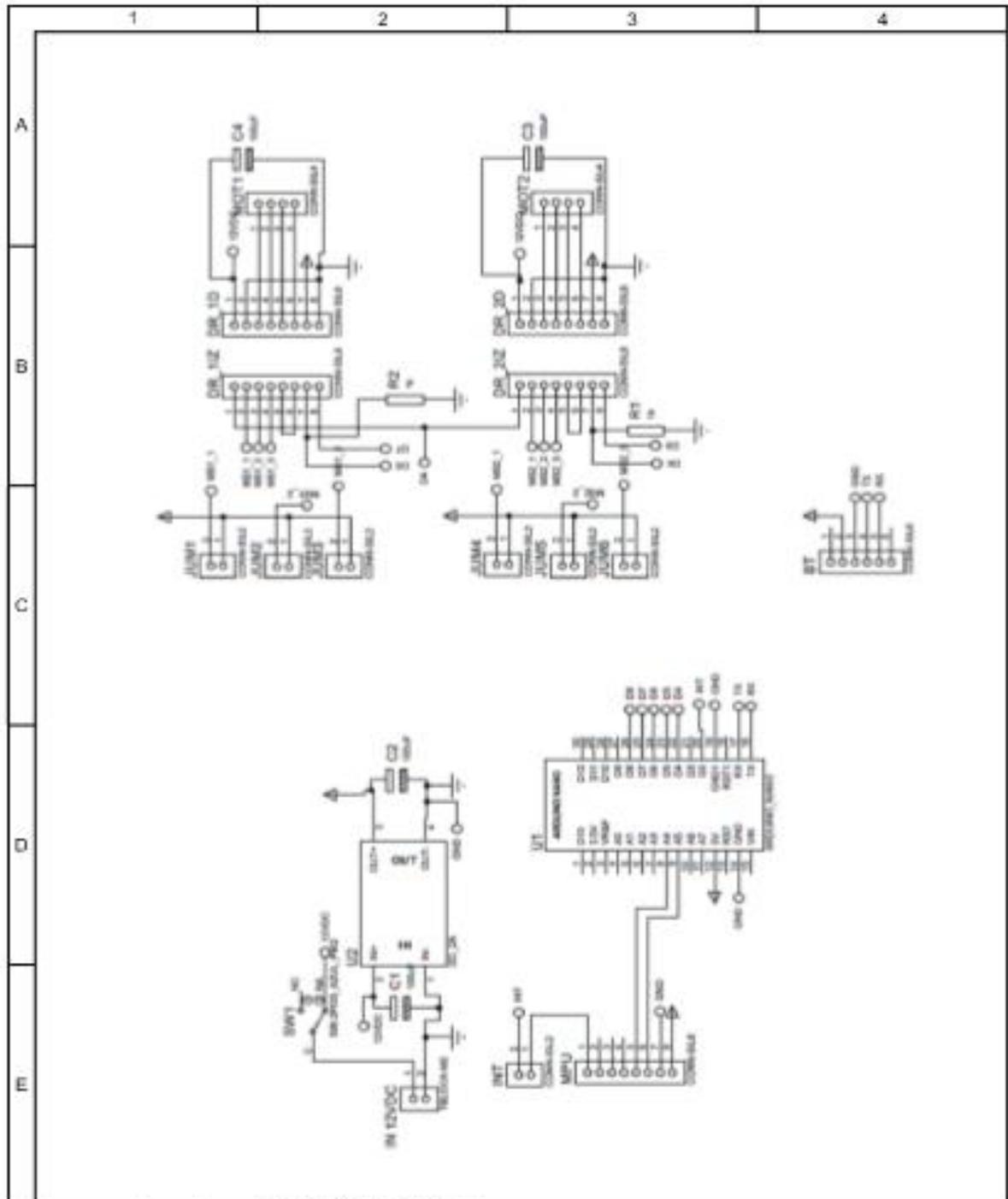
Anexo 4: Tapa superior del robot balancín



Anexo 5: Soportes de los motores a pasos nema 17



Anexo 6: Circuito de control del robot balancín



				Tolerancias:	Peso:	DISEÑO EN PROTEUS		
				Fecha	Nombre			
				Dib:	04/03/20	Circuito Electronico de control		
				Rev:	04/03/20			Escala: 1:1
				Aprob:	04/03/20			
				ISTVN		Anexo N° : 006		
Edición	Modificación	Fecha	Nombre	Sustituye a:				

Anexo 7: Programación de prueba del PID del robot balancín

```
include <MPU6050_tockn.h>

#include <Wire.h>

volatile unsigned long tiempoAnterior;

volatile double dT;

MPU6050 mpu6050(Wire);

long timer = 0;

#define mot16 8

#define mot18 6

#define mot25 5

#define mot27 7

int pasos = 200;

int stepDelay;

long tiempo;

double kp=3.8;//10.3//10.8//10.3

double ki=0.0030;//0.0024//0.0094

double kd=2.25;//8.2//12.6//12.

double ref= 2.21; // el valor del setpoint con valores decimales 38.9

double error=0;

double erroran=0;

double errorin=0;

double errorde=0;

void setup() {

    // put your setup code here, to run once:

    Serial.begin(9600);

    Wire.begin();

    mpu6050.begin();

    mpu6050.calcGyroOffsets(true);
```

```

pinMode(mot16, OUTPUT);
pinMode(mot18, OUTPUT);
pinMode(mot25, OUTPUT);
pinMode(mot27, OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
mpu6050.update();
unsigned long tiempoActual;
tiempoActual = millis();
dT=(double)(tiempoActual-tiempoAnterior);
if(dT>=1){
error = ref-mpu6050.getAngleX();
errorde=error-erroran;
errorin +=(ki*error);
double U =kp*error+(errorin)+kd*errorde;
if(U>255){
U=255;
digitalWrite(mot16, HIGH);
digitalWrite(mot27, HIGH);
}
if (U<0){
U=150;
digitalWrite(mot16, LOW);
digitalWrite(mot27, LOW);
}
analogWrite(mot18, U);
analogWrite(mot25, U);
erroran=error;
}
}

```

```
    tiempoAnterior=tiempoActual;
}
}
```

Anexo 8: Freeware de control e interface con la aplicación EZ-GUI de Wii Master

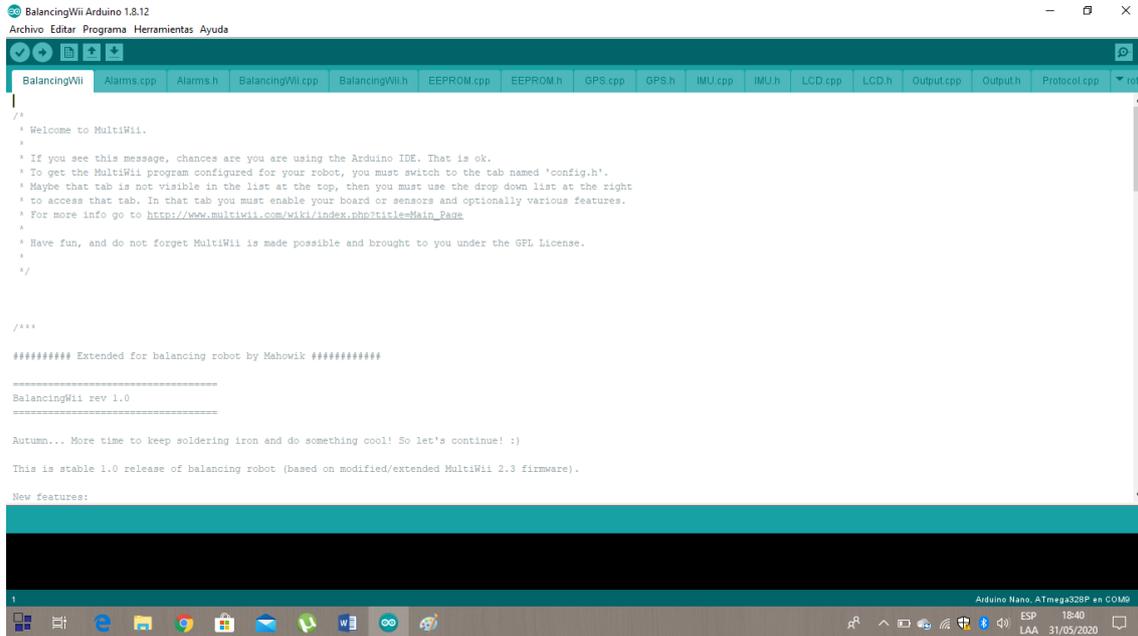


Figura 48. Programa Freeware.
Fuente: Sandeep (2019).