



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS,

ELECTRÓNICA E INDUSTRIAL

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
COMUNICACIONES**

TEMA:

CONSTRUCCIÓN DE UN ROBOT MÓVIL ESFÉRICO CONTROLADO
INALÁMBRICAMENTE PARA LA EJECUCIÓN DE PROYECTOS DE
INVESTIGACIÓN FORMATIVA Y GENERATIVA DE LA FACULTAD DE
INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL.

**Trabajo de Graduación Modalidad: Proyecto de Investigación, presentado
previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.**

Autor: Danny Santiago Pérez Camejo

Tutor: Ing. Víctor Hugo Andaluz, PhD.

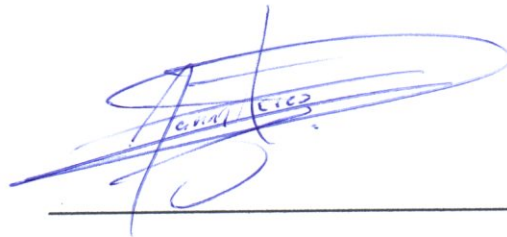
Ambato – Ecuador

Abril 2017

AUTORÍA

El presente Proyecto de Investigación titulado: CONSTRUCCIÓN DE UN ROBOT MÓVIL ESFÉRICO CONTROLADO INALÁMBRICAMENTE PARA LA EJECUCIÓN DE PROYECTOS DE INVESTIGACIÓN FORMATIVA Y GENERATIVA DE LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Abril 2017



Danny Santiago Pérez Camejo

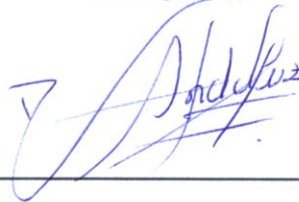
C.C. 1803801388

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: -
CONSTRUCCIÓN DE UN ROBOT MÓVIL ESFÉRICO CONTROLADO
INALÁMBRICAMENTE PARA LA EJECUCIÓN DE PROYECTOS DE
INVESTIGACIÓN FORMATIVA Y GENERATIVA DE LA FACULTAD DE
INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL, del Sr. Danny
Santiago Pérez Camejo, estudiante de la Carrera de Ingeniería en Electrónica y
Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial,
de la Universidad Técnica de Ambato, considero que el informe investigativo reúne
los requisitos suficientes para que continúe con los trámites y consiguiente
aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para
la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de
la Universidad Técnica de Ambato

Ambato, Abril 2016

EL TUTOR



Ing. Victor Hugo Andaluz, PhD.

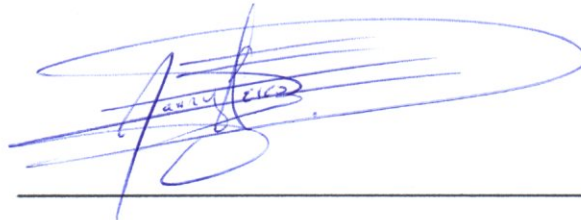
TUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Abril 2017



Danny Santiago Pérez Camejo

C.C. 1803801388

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes, Ing. Marcelo García, Mg. e Ing. Santiago Manzano, Mg., aprobó el Informe Final del Proyecto de Investigación titulado “CONSTRUCCIÓN DE UN ROBOT MÓVIL ESFÉRICO CONTROLADO INALÁMBRICAMENTE PARA LA EJECUCIÓN DE PROYECTOS DE INVESTIGACIÓN FORMATIVA Y GENERATIVA DE LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL”, presentado por el Sr. Danny Santiago Pérez Camejo, de acuerdo al numeral 9.1 de los Lineamientos Generales para la Aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

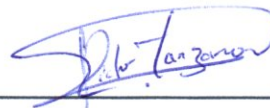


Ing. Pilar Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL



Ing. Marcelo García, Mg.
DOCENTE CALIFICADOR



Ing. Santiago Manzano, Mg.
DOCENTE CALIFICADOR

DEDICATORIA

A mis padres Héctor y Diana, por su constante lucha y sacrificio para verme como un profesional de la educación; Sus consejos y enseñanzas están plasmados en este trabajo investigativo, realizado con esfuerzo y dedicación.

A mis hermanos, familiares, mi pareja sentimental y grandes amistades que he formado a lo largo del camino educativo. Ellos quienes estuvieron pendientes en el proceso académico y con sus palabras supieron dar el aliento para continuar la senda que conduce al horizonte del éxito.

Danny Pérez

AGRADECIMIENTO

Nombrar a todos quienes me supieron brindar un apoyo incondicional me resultaría bastante extenso, por lo que lo resumo de la siguiente manera: gracias Dios, gracias Familia, gracias Ingenieros, gracias amigos y a todas las personas adecuadas en mi camino.

Danny Pérez

ÍNDICE GENERAL DE CONTENIDOS

AUTORIA.....	II
APROBACION DEL TUTOR.....	III
DERECHOS DE AUTOR.....	IV
APROBACION DE LA COMISION CALIFICADORA.....	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
ÍNDICE GENERAL DE CONTENIDOS	VIII
ÍNDICE DE FIGURA	XIII
ÍNDICE DE TABLAS	XV
RESUMEN	XVI
ABSTRACT.....	XVII
INTRODUCCIÓN	XVIII
CAPITULO I	1
EL PROBLEMA.....	1
1.1. Tema	1
1.2. Planteamiento del Problema	1
1.2.1. Contextualización.....	1
1.3. Delimitación.....	2
1.4. Justificación	3
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos	4
CAPÍTULO II.....	5
MARCO TEÓRICO.....	5
2.1 Antecedentes investigativos	5
2.2 Fundamentación teórica	6
2.2.1 Concepto	6
2.2.2 Robótica	7
2.2.4 Origen de los Robots Tipo Esfera.....	8
2.2.5 Clasificación según el mecanismo interno.....	10
Basado en péndulo	10
Mediante dos Hemisferios Independientes	11
Masas móviles.....	12

Modelo rueda de hámster (Hamster Wheel Model).....	12
2.2.6 Tecnologías de comunicación inalámbricas	13
Tecnología WLAN: Wifi	13
Tecnologías WMAN: WIMAX.....	14
Tecnologías WPAN: BLUETOOTH.....	15
Tecnología WPAN: ZIGBEE.....	16
2.2.7 Elementos de diseño de un robot móvil.....	16
Plataformas de Control.....	16
Plataforma de Control Arduino	17
Plataforma de control DEO-Nano Board.....	18
Plataforma de control Raspberry Pi.....	18
Sensores	19
Sensores empleados en la robótica	20
Servomecanismos.....	21
Componentes internos de un servo	21
Tipos de Ruedas	22
Rueda fija o común	22
Ruedas Omnidireccionales	22
Ruedas Orientales	23
Ruedas Especiales	23
Fuente de alimentación (Baterías)	24
2.2.8 Disposición de las ruedas sobre la estructura mecánica.....	25
Robot Uniciclo	25
Triciclo.....	25
Cuatriciclo.....	26
Robot Omnidireccional	27
2.2.9 Estructuras de Robots Omnidireccionales	28
Estructura Omnidireccional de 3 ruedas.....	28
Estructura Omnidireccional de 4 ruedas.....	29
2.2.10 Servidor Web	29
Lenguajes de Programación del lado servidor o cliente.....	30
2.3 Propuesta de solución.....	32
CAPITULO III.....	33
METODOLOGÍA	33
3.1. Modalidad de la investigación.....	33
3.1.1 Investigación bibliográfica	33

3.1.2 Investigación experimental.....	33
3.2. Población y muestra	33
3.3. Recolección de información	33
3.4. Procesamiento y análisis de datos	34
3.5. Desarrollo del proyecto	34
CAPITULO IV	36
DESARROLLO DE LA PROPUESTA	36
4.1 Introducción	36
4.2 Análisis de factibilidad.....	36
4.2.1 Factibilidad Técnica	36
4.2.2 Factibilidad Económica.....	37
4.3 Análisis de requerimientos	37
4.3.1 Hardware.....	37
Plataforma de control SBC.....	37
Plataforma de control basado en AVR	39
Sensor acelerómetro-giroscopio	40
Dispositivo de comunicación inalámbrica.....	41
Servomecanismos.....	42
Servomotor inteligente	42
USB2Dynamixel	44
Servomotor MG996R (High Torque)	45
Alimentación.....	45
4.3.2 Software	46
Distribuciones para Raspberry Pi	46
IDLE (Python).....	47
DreamWeaver	47
Herramienta de diseño y análisis SolidWorks	47
4.4 Análisis del Mecanismo Interno	48
4.4.1 Análisis del centro de masa del prototipo	48
Centro de Gravedad.....	48
Centro de masa.....	51
Cálculo del centro de masa del prototipo	51
4.4.2 Análisis de la unidad de manejo interno	54
Movimiento Circular	54
Posición Angular.....	55
Desplazamiento Angular	55

Velocidad Angular Media	56
Aceleración Angular	57
Relación entre velocidad lineal y velocidad angular	57
4.4.3 Modelo Cinemático	60
4.5 Consideraciones del Diseño Mecánico	62
4.5.2Diseño de las piezas internas	62
4.6 Diseño Electrónico	66
4.7 Interfaz de operación hombre-máquina	68
4.7.1 Interfaz de mando.....	68
Control de mando.....	69
4.8 Controlador PID	72
4.8.1 Estructura del controlador PID	73
4.8.2 Sintonización del controlador PID mediante el método de curva de reaccion de Ziegler-Nichols	74
Cálculo de las constantes K_p , K_i y K_d	75
Implementación del control PID	76
4.9 Pruebas de Funcionamiento.....	78
Pruebas de alimentación.....	79
Pruebas de comunicación entre los servomotores Dynamixel y el ordenador mediante el uso de RoboPlus-Wizard.....	80
Pruebas de funcionamiento de la comunicación inalámbrica a través de WiFi.....	81
Pruebas de ejecución de los servomotores de la interfaz de mando	81
Pruebas de funcionamiento del controlador PID implementado.....	82
4.9 Análisis económico del proyecto.....	83
CAPITULO V	85
CONCLUSIONES Y RECOMENDACIONES	85
5.1 Conclusiones	85
5.2 Recomendaciones.....	86
BIBLIOGRAFIA	87
ANEXOS	91
Anexo A.....	92
Guía de instalación del sistema operativo en Raspberry Pi	92
Anexo B	95
Instalación del servidor LAMP en la Raspberry Pi 2	95
B-1 Instalación del Servidor Apache.....	95
B-2 Instalación MySQL	95

B-3 Instalación de PHP	97
Anexo C	98
Instalación del servidor FTP	98
Anexo D.....	100
Instalación de la plataforma Node.js y la inclusión del módulo Socket.io.....	100
Anexo E	102
InterfazTactil.html y su respectivo código fuente en JavaScript y Canvas	102
Anexo F	109
Código fuente del archivo Index.js para entablar la comunicación entre .html y .py.....	109
Anexo G.....	111
servody.py y su respectivo código fuente para la activación de los servomotores Dynamixel.....	111
Anexo H.....	113
Código de programación del controlador PID	113
Anexo I	117
Distribución de los GPIO de la tarjeta de control Raspberry Pi 2.....	117

ÍNDICE DE FIGURA

Figura 2.1: a) Primer prototipo esférico por J.L Tate.b) Diseño esférico por Shorthouse...	9
Figura 2.2: Prototipo mecánico desarrollado por E.E Cecil.....	10
Figura 2.3: Mecanismo interno basado en el diseño pendular.	10
Figura 2.4: Dos mitades en base al diseño de Bhattachaya.	11
Figura 2.5: Diseño de dos hemisferios controlados independientemente.	11
Figura 2.6: Mecanismo interno basado en el diseño de masa móviles.....	12
Figura 2.7: Bola de Hámster de tres ruedas.....	12
Figura 2.8: Plataforma Arduino Uno y componentes.	18
Figura 2.9: Tarjeta Deo-Nano y componentes.....	18
Figura 2.10: Tarjeta Controladora Raspberry Pi y sus componentes	19
Figura 2.11: Componentes generales de un servomotor.	21
Figura 2.12: Rueda Omnidireccional con rodillos en su periferia doble.....	22
Figura 2.13: Mecanismo Omnidireccional con ruedas orientables.	23
Figura 2.14: Componentes de una rueda universal.....	23
Figura 2.15: Rueda Especial (a) Universal Doble (b) Mecanum.	24
Figura 2.16: Robot tipo Uniciclo.....	25
Figura 2.17: Robot móvil tipo Triciclo.....	26
Figura 2.18: Sistema de dirección Ackerman.....	27
Figura 2.19: Estructura de un robot omnidireccional.	27
Figura 2.20: Plataforma Omnidireccional de tres ruedas.....	28
Figura 2.21: Plataforma Omnidireccional de cuatro ruedas.....	29
Figura 2.22: Páginas dinámicas usando aplicaciones del lado del cliente	30
Figura 4.1: (a) Control Dynamixel a través del ordenador (b) Conversión a serial (c)Comunicación inalámbrica.	44
Figura 4.2: Batería Li-Po de 11.1V a 1300mAh.....	46
Figura 4.3: (a) Circuito fijador de voltaje de Carga y Descarga (b)Baterías Ion-Litio 4.7V a 5600mAh.....	46
Figura 4.4: Sistema de n partículas.....	49
Figura 4.5: Sistema de n partículas con giro de 90 grados anti-horario con respecto al eje X.....	50
Figura 4.6: Punto medio del robot esférico centros de masa de los elementos que lo conforman.....	52

Figura 4.7: Movimiento circular de partículas.....	54
Figura 4.8: Angulo entre eje de referencia y vector posición.....	55
Figura 4.9: Variación angular de una partículas.....	56
Figura 4.10: (a) Modelo de la estructura mecánica del robot móvil (b) Orientación del robot móvil.....	61
Figura 4.11: Vista en 2D de la base general con sus respectivas dimensiones.	63
Figura 4.12: Estructura interna ensamblada.	64
Figura 4.13: Bola de ejercicio para mascotas.....	65
Figura 4.14: Esquema de interconexión eléctrica de los componentes electrónicos.	67
Figura 4.15: Captura de pantalla de la interfaz táctil (a) Modo vertical (b) Modo horizontal.	70
Figura 4.16: Esquema de control PID.	73
Figura4.17:Medición del voltaje de entrada (a) Raspberry Pi (b)Servomotores Dynamixel.....	79
Figura 4.18: Reconocimiento y manejo de los dos servomotores conectados en cadena.	80
Figura 4.19: Medición de voltaje de entrada a través de Roboplus.....	81
Figura 4.20: Muestra en consola de la ejecución de los servomotores Dynamixel.	82
Figura 4.21: Acción de corrección del controlador PID con Setpoint fijo.....	82
Figura 4.22: Señal de control o variable controlada.	83

ÍNDICE DE TABLAS

Tabla 4.1: Comparación entre los distintos modelos SBC.....	38
Tabla 4.2: Especificaciones técnicas de distintas plataformas de control con AVR.	39
Tabla 4.3: Comparación de adaptadores Wi-Fi.	41
Tabla 4.4: Servomecanismos inteligentes.	43
Tabla 4.5: Especificaciones técnicas servomotor MG996R.	45
Tabla 4.6: Elementos internos con su respectiva masa y coordenadas.	52
Tabla 4.7: Vistas de las piezas que conforman el mecanismo interno.....	63
Tabla 4.8: Descripción de la funcionalidad de los botones del mando táctil.....	71
Tabla 4.9: Descripción de la funcionalidad de los botones del mando por teclado.....	72
Tabla 4.10: Reglas de sintonización Ziegler-Nichols primer método.....	75
Tabla 4.11: Costo del proyecto.	83

RESUMEN

En la actualidad, la búsqueda de información para el diseño y construcción de robots móviles está avanzando a pasos agigantados. Esto es producto de la investigación y constantes experimentos sobre la robótica y sus múltiples aplicaciones, además, existe en el mercado el hardware o elementos necesarios para su fabricación, con precios accesibles y de todas las marcas.

En este trabajo de investigación se describe un robot móvil esférico capaz de navegar sobre superficies bidimensionales. Este robot consta de una carcasa esférica externa de 245mm de diámetro accionada por una Unidad Interna de Locomoción basado en péndulo que concentra el centro de masa del prototipo, dos servomecanismos inteligentes permiten la movilidad adelante-atrás, un servomotor convencional de alto torque mueve el péndulo de derecha a izquierda sacándolo de su estado de reposo con el propósito de alcanzar algún destino, sin perderse y sin chocar ni con obstáculos fijos, ni con otros móviles que eventualmente puedan aparecer en el camino, totalmente comandado de forma inalámbrica desde una aplicación web desarrollada con enfoque a la tecnología táctil de un Smartphone o al teclado de un ordenador. Se ha realizado el estudio cinemático del mismo basado en los conceptos de robot unicycle. El diseño y fabricación de las piezas internas del prototipo, se consideró para este trabajo el basado en Masa Pendular. Se demuestra que con la implementación de un controlador PID, el robot puede seguir básicamente la orientación deseada y estabilizarse en su estado de reposo.

Una vez que se tienen a mano el robot esférico móvil, es necesario saber manipularle, a esta actividad se llama navegación, que es el conjunto de métodos y técnicas usados para dirigir el curso de un robot móvil a medida que éste atraviesa su entorno. Se supone que, al término de una demostración robótica, se puede apreciar los resultados de su fabricación.

Palabras clave: Robots esféricos móviles, robot unicycle, servomotor inteligente, comunicación inalámbrica, servidor web

ABSTRACT

Today, the search for information for the design and construction of mobile robots is advancing by leaps and bounds. This is a product of research and constant experiments on robotics and its multiple applications, in addition, there is in the market the hardware or elements necessary for its manufacture, with affordable prices and all brands.

In this research, a spherical mobile robot capable of navigating on two-dimensional surfaces is described. This robot consists of an external 245mm diameter spherical housing driven by an internal locomotion unit based on pendulum that concentrates the center of mass of the prototype, two intelligent servomechanisms allow forward-backward mobility, a conventional high-torque servomotor moves the pendulum From right to left taking it out of its resting state in order to reach some destination, without getting lost and without colliding with fixed obstacles, or with other mobiles that may eventually appear on the road, totally commanded wirelessly from a developed web application With focus on the touch technology of a Smartphone or the keyboard of a computer. The kinematic study of the same one has been realized based on the concepts of unicycle robot. The design and manufacture of the internal pieces of the prototype, was considered for this work the one based on Pendulum Mass. It is shown that with the implementation of a PID controller, the robot can basically follow the desired orientation and stabilize in its rest state.

Once the mobile spherical robot is at hand, it is necessary to know how to manipulate it. This activity is called navigation, which is the set of methods and techniques used to direct the course of a mobile robot as it traverses its environment. It is assumed that, at the end of a robotic demonstration, the results of its manufacture can be appreciated.

Keywords: Mobile spherical robots, unicycle robot, intelligent servomotor, wireless communication, web server.

INTRODUCCIÓN

El presente proyecto se encuentra estructurado de la siguiente manera:

En el Capítulo I, se presentan las razones por las cuales es importante el desarrollo de esta investigación partiendo de un análisis macro, meso y micro hasta el planteamiento de la justificación que sirva de sustento para el desarrollo del mismo y los objetivos a satisfacer.

En el Capítulo II, se detalla una breve introducción de los robots con arquitectura esférica, donde se describe rápidamente parte de su historia y sus mecanismos de movimiento internos. Además, en este capítulo se plantea y describe de forma general todo lo concerniente a tecnologías de comunicación, estructuras mecánicas y hardware, lo cual permitirá establecer un conocimiento base para el diseño y construcción del prototipo.

El Capítulo III está compuesto por la información de la metodología utilizada para desarrollar la totalidad del proyecto presentado.

En el Capítulo IV, se describe el proceso que se llevó a cabo para la selección del hardware y software que se definen como la base fundamental para el desarrollo del robot esférico, se analiza también los fundamentos físicos que se genera al tener este tipo de diseño. En este mismo capítulo se detalla todo lo correspondiente al diseño mecánico, electrónico y la interfaz hombre-máquina.

En el Capítulo V, se mencionan las conclusiones y recomendaciones que se derivan del desarrollo de este trabajo. Finalmente, se presentan la lista de referencias consultadas y anexos.

CAPITULO I

EL PROBLEMA

1.1. Tema

Construcción de un robot móvil esférico controlado inalámbricamente para la ejecución de proyectos de investigación formativa y generativa de la Facultad de Ingeniería en Sistemas Electrónica e Industrial.

1.2. Planteamiento del Problema

1.2.1. Contextualización

Desde la aparición de la primera patente que consistía en un diseño puramente mecánico y encerrado en un cascaron esférico, ha sido motivo para que personajes en solitario y muchos grupos de investigación hayan participado a través del tiempo en el desarrollo y mejoramiento de estos cuerpos esféricos móviles.

A continuación, se presenta sus orígenes, cuyo retalo consta en el siguiente artículo titulado “Ball shaped Robots-InTech”, y dice así:

El “juguete” patentado por el Estado Unidense J. L. Tate en 1893 (U.S. Patent 508,558), presenta un vehículo esférico propulsado unidireccionalmente por un resorte cuya única aplicabilidad era el entretenimiento.

En 1906, B. Shorthouse patentó un diseño que ofrecía la posibilidad de ajustar manualmente la posición del contrapeso interno, con el fin de que dicha esfera ruede a lo largo de una trayectoria curva en vez de una trayectoria rectilínea (U.S. Patent 819.609). A partir de ello, se puede encontrar distintos mecanismos y esquemas para la movilidad de cuerpos esféricos[1].

En la actualidad, el uso de un robot tipo esfera que tiene libertad de movimiento es un punto interesante a la hora de navegar por zonas desconocidas ya que no existe la cara superior o inferior del mismo, es decir, este robot no se puede volcar, o no puede tener problemas con los que se podría encontrar por citar un ejemplo el robot Curiosity que explora Marte [2].

A nivel de Ecuador, los intentos por encontrar información en cuanto diseño y construcción de robots móviles esféricos, ha sido relativamente escaso. Sin embargo, esto no quiere decir que no exista ningún tipo de relación con esta tecnología, actualmente la marca Sphero ha alcanzado el mercado ecuatoriano con su diseño generando asombro, diversión y aprendizaje en jóvenes y niños de distintos puntos del país [3].

Por último, se examinó en el repositorio, laboratorios y el club de robótica, pertenecientes a la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, temas pertinentes a la investigación propuesta, de los cuales no existen registros relacionados al mismo.

1.3. Delimitación

Área académica: Física y Electrónica

Línea de investigación: Sistemas Electrónicos

Sublínea de investigación: Robótica

a) Delimitación del contenido

Campo: Robótica

Área: Construcción de un robot móvil

Aspecto: Robot esférico controlado

b) Delimitación espacial

Facultad de Ingeniería en Sistemas Electrónica e Industrial

c) Delimitación temporal

Esta investigación se realizará en el periodo Abril 2016 – Septiembre 2016 de acuerdo al Régimen de Titulación Vigente.

1.4. Justificación

El siguiente trabajo de investigación reviste **interés** innovación en el campo tecnológico, por cuanto se puede aprovechar su diseño mecánico y electrónico para fines didácticos.

El rediseño de un robot en forma esférica conlleva gran **importancia** para el desarrollo de algoritmos de control en el área de la robótica de la Facultad de Ingeniería en Sistemas Electrónica e Industrial, en virtud permite realizar el estudio estructural de un robot aplicando nuevos esquemas de control avanzado, aumentando así la autonomía del mismo.

Frente a la propuesta de construcción de un robot móvil esférico controlado inalámbicamente, ha causado **novedad** en profesores y estudiantes, ya que es la primera vez que se valora la idea de propiciar un estudio formal en cuanto a diseños mecánicos y electrónicos de robots.

Ante el concepto de un cuerpo esférico, la construcción de un robot de dichas características **beneficiará** a estudiantes, docentes e investigadores que conforman la Carrera de Electrónica y Comunicaciones; y áreas afines, puedan tomarlo como modelo para nuevas investigaciones formativas y generativas de la Facultad.

El robot esférico causará **impacto**, por ende, al ser el primer prototipo debido a su forma particular de diseño, aportará nuevos criterios para la construcción de prototipos anteriormente no considerados.

La presente investigación es **factible**, en virtud que posee una amplia bibliografía física y electrónica, recursos humanos y tecnológicos; los cuales son económicamente alcanzables.

1.5. Objetivos

1.5.1. Objetivo General

Construir un robot esférico controlado inalámbricamente para la ejecución de proyectos de investigación formativa y generativa de la Facultad de Ingeniería en Sistemas Electrónica e Industrial.

1.5.2. Objetivos Específicos

- Describir las características cinemáticas de un robot esférico a fin de ser consideradas en el diseño de algoritmos de control.
- Construir un robot móvil esférico y acondicionar las señales eléctricas de motores y sensores para su utilización en algoritmos de control avanzados.
- Desarrollar una Interfaz Hombre Máquina que permita la comunicación inalámbrica para el control tele-operado.
- Realizar evaluaciones experimentales del desempeño del robot esférico construido y de la interfaz implementada.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos

En el proceso de indagación y revisando el repositorio de la Facultad de Ingeniería en Sistemas Electrónica e Industrial, así como documentos físicos en la biblioteca de la misma; no se han encontrado trabajos investigativos similares. No obstante, revisando el repositorio de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, se localizó un trabajo pertinente en el año 2012 titulado: “Diseño y Construcción de un robot móvil esférico con visión en tiempo real.” Cuyoprototipo cumple con las expectativas esperadas respecto a velocidad, estabilidad y tracción. En cuanto al diseño del robot esfera hace uso de unas bases de foamy espuma sobre la superficie de acrílico para así mejorar la maniobrabilidad, conducción y direccionamiento del mismo mediante una aplicación de mando desarrollada en Android. A demás, con ello se consigue obtener una buena visualización en tiempo real de la trayectoria del robot ya que, al tener menor oscilación, la imagen es perfectamente apreciable [4].

Así mismo, se tomó como referencia el siguiente artículo de investigación: “A Literature Review on the Design of Spherical Rolling Robots.”Que traducido al español significa: “Una Revisión Bibliográfica sobre el Diseño de los Robots Esféricos Móviles.” El cual presenta una revisión bibliográfica sobre el diseño de dichos robots. También, expone varios diseños diferentes con un análisis simple de la dinámica, sus aplicaciones y la prospectiva de los mismos cuyas ventajas y desventajas se discuten posteriormente a la revisión de cada diseño[5].

Otro proyecto similar publicado en el 2014 con el siguiente tema de investigación: “Mechanical Development and Control of a Miniature Spherical Rolling Robot.” Que traducido al español significa: “Desarrollo Mecánico y Control de un Robot MóvilEsférico en Miniatura.”El cual detalla el modelamiento dinámico del robot

cuyos resultados sirven en parte para la fabricación del chasis, el mismo que albergara un micro-controlador, sensores, ruedas y otros componentes de forma compacta dentro de una pequeña bola de plástico transparente. También estudia la inclusión de un control PD para mejorar el desplazamiento debido a que el robot no puede moverse en una trayectoria deseada por los distintos problemas presentes al trabajar con un control de lazo de abierto[6].

En el 2012 Juan David Hernández Vega de la Universidad Politécnica de Madrid en España realizó el proyecto denominado: “ROSPHERE: Diseño, Construcción y Aplicación de una Esfera Robótica.” Para la realización del proyecto se considera el uso de un mecanismo interno pendular de dos grados de libertad (GdL), el cual permite desbalancear el sistema y, consecuentemente, inducirle movimiento. En cuanto a software y hardware se refiere, emplea arquitecturas de bajo y alto nivel en ambos casos. Respecto al hardware de bajo nivel, utiliza un microcontrolador que realiza interfaz directa con los actuadores (dos servomotores), así como los diferentes sensores propioceptivos (IMU, GPS, etc.) y exteroceptivos (temperatura, humedad, luminosidad, etc.). El hardware de alto nivel, por otra parte, está constituido por un sistema de cómputo embebido. Respecto al software, se presenta el desarrollo de rutinas en C (bajo nivel) y Python (alto nivel) que se relacionan de manera directa con las capas de hardware.

Finalmente, se presentan las pruebas del sistema robótico, con las cuales se evalúan, cuantifican y demuestran las capacidades del robot para desplazarse en diferentes superficies, en terrenos planos y con pendientes, así como su uso como un sensor móvil de variables ambientales, tales como temperatura, humedad y luminosidad, en diferentes entornos[7].

2.2 Fundamentación teórica

2.2.1 Concepto

La fundamentación teórica de una tesis, es el estudio profundo del tema en cuestión a través de la indagación bibliográfica de lo que sobre él se ha escrito con

rigor científico y la consecuente estructuración lógica del material y el análisis crítico del mismo.

Las bases teóricas constituyen el corazón del trabajo de investigación, pues es sobre este que se construye todo el trabajo. Una buena base teórica formará la plataforma sobre la cual se construye el análisis de los resultados obtenidos en el trabajo, sin ella no se puede analizar los resultados.

En general, el marco teórico es el capítulo del trabajo en el cual se encuentran los antecedentes y las bases teóricas o la fundamentación teórica[8].

2.2.2 Robótica

De forma general, la Robótica se define como: “El conjunto de conocimientos teóricos y prácticos que permiten concebir, realizar y automatizar sistemas basados en estructuras mecánicas poli articuladas, dotados de un determinado grado de "inteligencia" y destinados a la producción industrial o a la sustitución del hombre en muy diversas tareas” [9].

De acuerdo con el proyecto de titulación “Diseño y Construcción de un robot móvil esférico con visión en tiempo real.”, desarrollado por Cañadas, R; Nazete, P. Definen a la Robótica de la siguiente manera:

“La robótica es la rama de la tecnología dedicada al diseño que combina diversas disciplinas y ciencias para proporcionar una amplia gama de prototipos que cumplan con las exigencias de la aplicación que van a desarrollar.”[4].

Para la investigadora K. Vélez, en su proyecto de titulación, plantea la siguiente definición de robótica:

“La robótica es la ciencia y tecnología de los robots enfocada en la construcción y programación de máquinas que ayuden a resolver determinadas situaciones en las que se requiere mayor fuerza física.”[10].

2.2.3 Origen de la Robótica

La robótica como concepto es una atractiva idea que viene de lejos. La idea de personas artificiales data al menos de la época de la antigua leyenda de Cadmus y el mito del Pygmalion. Posteriormente en la antigua Grecia, tres siglos antes de cristo, Ctesibius y Philon de Bizancio ya desarrollaban figuras mecánicas con el fin del entretenimiento de la gente. La robótica tal y como la concebimos hoy en día surgió con el desarrollo industrial. El término “robótica” significa en checo trabajos forzados. Fue acuñado por Karel Capek, quien, inspirado por el desempleo secundario a la mecanización de las fábricas con la revolución industrial, escribió su obra: Rossum’s Universal Robots (1921). Posteriormente lo popularizó el escritor de ciencia ficción Isaac Asimov en su obra Runaround (1942) y I, Robot (1950)[11].

Actualmente, la robótica ha tomado fuerza y se ha hecho presente en la vida cotidiana, conjuntamente con la colaboración de las matemáticas, y de otras ramas de la ingeniería como: la mecánica, la electrónica, la computación, ya que al ser una ciencia multidisciplinaria pretende alcanzar avances significativos que mejoren y faciliten las labores del hombre, tanto cotidianas como en el mundo profesional.

2.2.4 Origen de los Robots Tipo Esfera

Una esfera es "el conjunto de todos los puntos en el espacio tridimensional a una misma distancia (radio) desde un punto dado (el centro)" (Enciclopedia Británica Online).

En términos de la robótica, una estructura esférica puede girar libremente en cualquier dirección y todas las posiciones son estables. La forma de una esfera proporciona simetría completa y un aspecto suave, seguro y amigable sin esquinas cortantes o protuberancias, lo cual es ventajoso cuando un dispositivo robótico está tratando con personas[1].

De acuerdo con la investigación llevada a cabo, el origen de los robots esféricos tomó lugar en el año de 1893, cuando J.L. Tate presentó el diseño de un vehículo esférico y cuya patente, se encuentra registrada en la Oficina de Patentes y Marcas

de los Estados Unidos (U.S. Patent 508,558). Este invento fue desarrollado con mira hacia el entretenimiento. En la figura 2.1a se puede apreciar el modelo patentado.

Años más tarde, B. Shorthouse también diseño y patentó un vehículo esférico que ofrecía la posibilidad de ajustar manualmente la posición interna del contrapeso, haciendo que tome una trayectoria curva en vez de solo una trayectoria recta tal y como se observa en la Figura 2.1b.

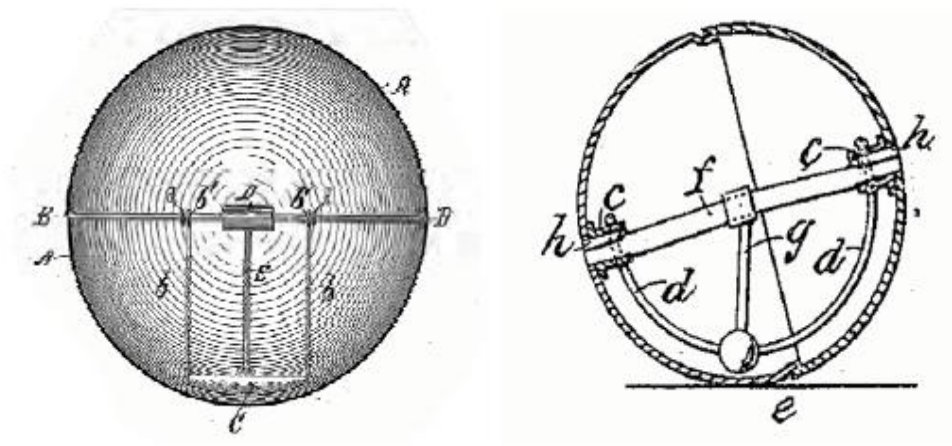


Figura 2.1: a) Primer prototipo esférico por J.L Tate. b) Diseño esférico por Shorthouse.[1]

Para el año de 1909[4], nuevamente se registra una patente cuyo prototipo se muestra en la Figura 2.2, un diseño que tomó lugar para los espectáculos de la época. Sin embargo, este atractivo diseño se trataba de una esfera con movimiento rodante tambaleante en zig-zag formado por una serie de engranajes y un péndulo; tal como lo describe su inventor en su patente (U.S. Patent 933,623).

A lo largo de la historia, los robots esfera se han ido popularizando y es la razón por la que ellos existen hasta la actualidad, con alrededor de 50 patentes registradas.



Figura 2.2: Prototipo mecánico desarrollado por E.E Cecil.[1]

Con el avance de la tecnología computacional, la evolución de las comunicaciones inalámbricas, y las cámaras digitales, permiten que estos interactúen de forma más autónoma con las personas y el medioambiente[1].

2.2.5 Clasificación según el mecanismo interno

Basado en péndulo

Este sistema está compuesto por un eje fijo al cuerpo esférico, el cual atraviesa de extremo a extremo y pasa por su CG. Dicho eje sirve como base de un sistema pendular de dos grados de libertad, donde el primero de ellos gira alrededor del eje fijo (eje transversal) y el segundo en sentido perpendicular del primero (eje longitudinal). En el extremo del sistema pendular se tiene una masa que permite mover el CM (Centro de Masa) fuera del CG (Centro de Gravedad) de la esfera (ver Figura 2.3)[12].



Figura 2.3: Mecanismo interno basado en el diseño pendular.[12]

Mediante dos Hemisferios Independientes

S. Bhattacharya desarrolló un diseño que involucra a dos rotores mutuamente perpendiculares ligados en el interior de la esfera. La esfera se ensambla en dos mitades, donde cada mitad aloja un motor y un rotor, tal como se puede apreciar en la siguiente Figura 2.4[13].

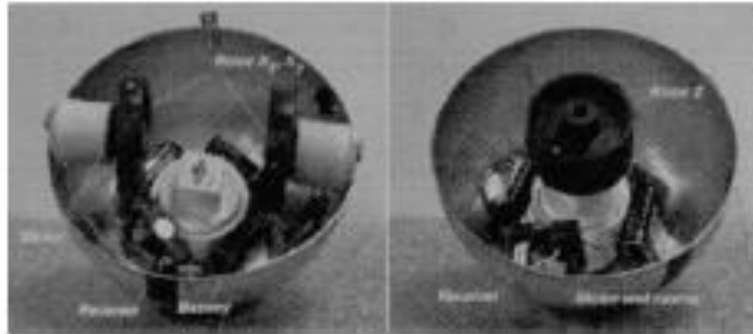


Figura 2.4: Dos mitades en base al diseño de Bhattachaya.[13]

La esfera se mueve usando la conservación del momentum en los rotores. Entre los problemas que presenta este tipo de diseño son[5]:

- Que posee una estructura mecánica compleja.
- Requiere un alto rendimiento de los motores debido a la alta velocidad de los rotores.

Para ello se puede considerar el diseño de dos hemisferios controlados de forma independiente, como se muestra en la siguiente Figura 2.5.



Figura 2.5: Diseño de dos hemisferios controlados independientemente.[5]

Masas móviles

Otra de las alternativas de mecanismos internos de esferas robóticas se denomina masas móviles. Esta cuenta con diferentes ejes que van desde el CG (Centro de Gravedad) hasta la superficie interior del cuerpo esférico del robot. Sobre estos ejes radiales se ubican y desplazan masas que permiten, por medio de su movimiento en conjunto, modificar la posición global del CM (Centro de Masa) del robot. A diferencia de los mecanismos previamente mencionados, este tipo de robot tiene la capacidad de realizar movimientos omnidireccionales (ver Figura 2.6)[14].

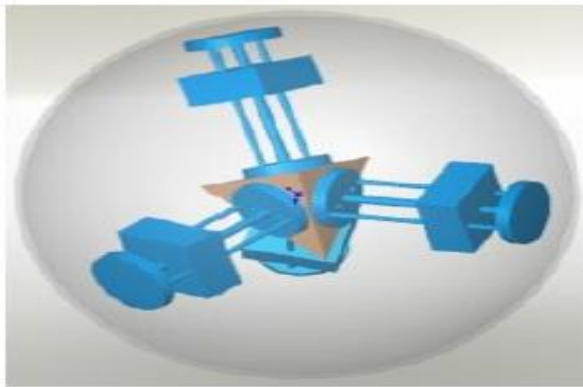


Figura 2.6: Mecanismo interno basado en el diseño de masa móviles.[14]

Modelo rueda de hámster (Hamster Wheel Model)

Se las llama así porque utilizan un vehículo en el interior del cuerpo esférico compuesto de dos a cuatro llantas para mover la superficie interna y lograr el desplazamiento del robot. El dispositivo interno se construye generalmente con la cantidad de peso y par necesario para que el sistema pueda rodar. Algunos vehículos están totalmente libres en el interior mientras que otros se apoyan en la estructura interior(ver Figura 2.7)[5].

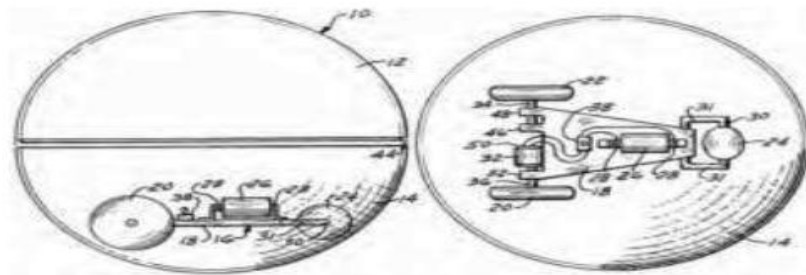


Figura 2.7: Bola de Hámster de tres ruedas. [5]

2.2.6 Tecnologías de comunicación inalámbricas

A continuación, se explicará de forma resumida cada tecnología de comunicaciones inalámbricas:

Tecnología WLAN: Wifi

En la actualidad es la que ofrece la mayor cantidad de beneficios al costo más bajo entre todas las tecnologías inalámbricas. Es económica, interoperable con equipos de diferentes fabricantes y puede ser extendida para ofrecer funcionalidades mucho más allá de las previstas originalmente por los fabricantes. Esto se debe a que WiFi utiliza estándares abiertos: enrutadores, tablet PCs, laptops y teléfonos WiFi pueden interoperar porque todos adhieren al estándar 802.11[15]. Sus velocidades de transmisión han permitido que WiFi se establezca como la tecnología predominante en el acceso inalámbrico de banda ancha a Internet, desbordando el ámbito de las aplicaciones y servicios para los que fue inicialmente concebida. A modo de resumen, conviene resaltar los siguientes aspectos técnicos, regulatorios y funcionales de WiFi[16]:

- Se trata de una tecnología madura, con un amplio abanico de productos y sistemas fiables y asequibles en el mercado y con un grado de implantación elevado y en aumento.
- Supone una conexión sin hilos en banda ancha (pocas complicaciones y bajo coste de instalación) que permita la movilidad.
- Múltiples aplicaciones (uso privado, público -Hot-Spots, hoteles, uso comunitario).
- Necesita un proyecto para cada red. En una red WiFi sin un diseño, planificación de recursos, instalación y protocolos de pruebas adecuados, pueden producirse problemas de interferencias, dificultades de protección, falta de cobertura, problemas de calidad de servicio, falta de seguridad en la red, problemas de movilidad, encarecimiento del coste final de la red, etc.

- Su viabilidad para soportar comunicaciones avanzadas (VoIP, datos e imagen) posibilita el soporte a servicios como telefonía, gestión integrada de sistemas, sistemas de información multimedia, transmisión de video para seguridad, etc.

Tecnologías WMAN: WIMAX

WiMax es la denominación de una certificación, pero que se extiende propiamente a la tecnología que cumple con esa especificación. Esto puede llevar a confusión porque productos que cumplan con la especificación, no van a tener la etiqueta de WiMax, de forma similar a lo que actualmente ocurre con IEEE 802.11 y WiFi. A pesar de la enorme difusión que presenta el estándar 802.11, presenta importantes limitaciones en la distancia a la que puede establecerse una comunicación, necesitando aumentar el número de repetidores para poder implementar una red metropolitana. Además, esta forma de incrementar la distancia solamente es aplicable en entornos urbanos[17]. Este nicho de mercado es el que quiere aprovechar WiMax. Esta tecnología permite ofrecer al usuario la transmisión a velocidades similares al ADSL o al cable/módem, sin necesidad de utilizar soporte físico de transmisión y alcanzando distancias de hasta 50 Km. Presenta a su vez una ventaja con respecto a IEEE 802.11, ofrece modos de funcionamiento donde no necesita visión directa entre emisor y receptor para hacer efectiva la comunicación (a frecuencias por debajo de 11GHz). El protocolo de comunicación digitales el denominado IEEE 802.16[18]: El estándar 802.16d para terminales fijos, y el 802.16e para estaciones en movimiento. El estándar inicial 802.16 se encontraba en la banda de frecuencias de 10-66 Gigahertzios. La nueva versión 802.16a, de marzo de 2003, usa una banda del espectro radioeléctrico más estrecha y baja, de 2-11 Gigahertzios

Como ventajas de WiMax se puede resumir[17]:

- **Banda Ancha.** Una sola estación de base puede admitir de manera simultáneamás de 60 enlaces con conectividad tipo T1/E1 o decenas de conexiones tipo DSL (ofreciendo verdaderamente decenas de megas).

- **Independencia de protocolo.** Utiliza la capa LLC, lo que le permite ser compatible con otros estándares como IEEE 802.11, Ethernet IEEE 802.3 o IEEE 802.5.
- **Ofrece QoS,** permitiendo transmitir servicios en tiempo real: VoIP, datos o video. En circunstancias ideales y sin obstáculos, la comunicación puede alcanzar una distancia a los 50 Km y la velocidad de transferencia de los datos puede llegar a los 70 Mbps.
- **Seguridad.** El estándar incluye medidas para privacidad y criptografía inherentes en el protocolo. Implementa la autenticación de los instrumentos con certificados x.509 usando DES en modo CBC. También soporta algoritmos AES.

Tecnologías WPAN: BLUETOOTH

Bluetooth es una tecnología utilizada para la conectividad inalámbrica de corto alcance entre dispositivos como PDAs (Personal Digital Assistance), teléfonos celulares, teclados, máquinas de fax, computadoras de escritorio y portátiles, módems, proyectores, impresoras, etc[19].

Durante la última semana del mes de marzo del 2002 la IEEE aprobó finalmente el estándar IEEE 802.15.1 compatible totalmente con la tecnología Bluetooth[19]. En noviembre de 2004 el SIG aprobó lo que será Bluetooth 2.0 EDR (Enhanced Data Rate), el cual presenta una velocidad de transmisión de 3 Mbps y un alcance de hasta 100 metros.

Un aspecto muy importante, dado las dimensiones del chip y al ser incorporado en dispositivos móviles y alimentado con baterías, es que tenga un consumo de potencia muy reducido (hasta un 97% menos que un teléfono móvil). Si los dispositivos Bluetooth no intercambian datos, entonces establecen el modo de “espera” para ahorrar energía, quedando a la escucha de mensajes. Dentro de sus especificaciones se resumen las siguientes[20]:

- Banda de Frecuencia: 2.4 GHz (Banda ISM).

- Potencia de transmisión: 1 m W para un alcance de 10m, 100 m W para un alcance de hasta 100m.
- Tecnología: Espectro expandido y Saltos en frecuencia (FHSS).
- Canales máximos de voz: 3 por piconet.
- Canales máximos de datos: 7 por piconet.
- Interferencia: Bluetooth minimiza la interferencia potencial al emplear saltos rápidos en frecuencia dividido 1600 veces por segundo.

Tecnología WPAN: ZIGBEE

La Alianza Zigbee es una asociación de empresas que trabajan juntas para lograr el desarrollo de un estándar para productos de monitorización y control[17]. Opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz y puede transferir datos hasta 250Kbps.

Esta organización sin fines de lucro nace en el año 2002. Desarrolla un protocolo que adopta al estándar IEEE 802.15.4 para sus 2 primeras capas[21], es decir la capa física (PHY) y la subcapa de acceso al medio (MAC) y agrega la capa de red y de aplicación.

A continuación, tenemos de forma resumida las características generales del estándar [21]:

- Ultra bajo consumo que permita usar equipos a batería
- Bajo costo de dispositivos y de instalación y mantenimiento de ellos.
- Alcance corto (típico menor a 50 metros).
- Optimizado para ciclo efectivo de transmisión menor a 0.1 %
- Velocidad de transmisión menor que 250 kbps. Típica: menor que 20 kbps

2.2.7 Elementos de diseño de un robot móvil

Plataformas de Control

Las tarjetas de control o controladoras sirven de enlace entre el ordenador y el sistema a controlar o un robot. Reciben las instrucciones del ordenador en forma

digital y tienen que convertirlas en señales, normalmente analógicas, que sean comprensibles para el robot; y viceversa, también tienen que recibir las señales del sistema robótico y enviárselas al ordenador para su procesamiento.

Existen diferentes tipos de controladoras, pero su apariencia es la de cualquier circuito impreso, con la particularidad que necesitan su propia fuente de alimentación[22].

Plataforma de Control Arduino

Arduino es una plataforma electrónica de código abierto basado en hardware y software fácil de usar desarrollada por David Cuartielles y Massimo Banzi[23]. A través de los años Arduino ha sido el cerebro de miles de proyectos, a partir de objetos cotidianos a los instrumentos científicos complejos. Una comunidad mundial de los fabricantes, estudiantes, aficionados, artistas, programadores y profesionales han reunido en torno a esta plataforma de código abierto, sus contribuciones añadiendo una increíble cantidad de conocimiento accesible que puede ser de gran ayuda para los principiantes como para expertos[23].

El hardware consta de una placa compuesta por un microcontrolador Atmel AVR con un conjunto de componentes electrónicos de bajo costo que facilitan la implementación. El diseño del hardware es de licencia abierta adaptable a las necesidades del programador. El microcontrolador viene precargado con el gestor de arranque de Arduino que simplifica la programación en la memoria flash de éste (ver Figura 2.8)[24]. "UNO" fue elegido para celebrar el lanzamiento del software de Arduino (IDE) 1.0. La tarjeta Uno y la versión 1.0 del software de Arduino (IDE) fueron las versiones de referencia de Arduino, evolucionando para nuevos lanzamientos.

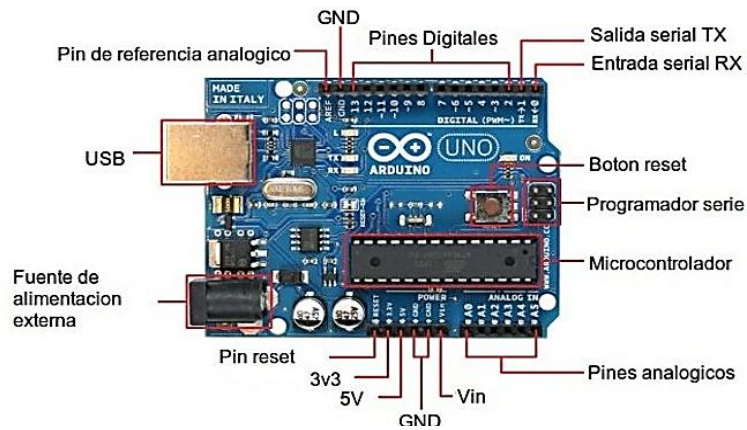


Figura 2.8: Plataforma Arduino Uno y componentes. [25]

Plataforma de control DEO-Nano Board

La tarjeta DE0-Nano presenta una plataforma de desarrollo de FPGA de tamaño compacto adecuado para una amplia gama de proyectos de enfoque portátil, como robots y proyectos móviles. La tarjeta en sí incluye un Blaster USB para la programación de la FPGA, esta a su vez puede ser alimentada ya sea desde este puerto USB o mediante una fuente de alimentación externa. Los puntos de expansión se pueden utilizar para unir varias tarjetas hijas Terasic u otros dispositivos, tales como motores y actuadores (ver Figura 2.9) [26].

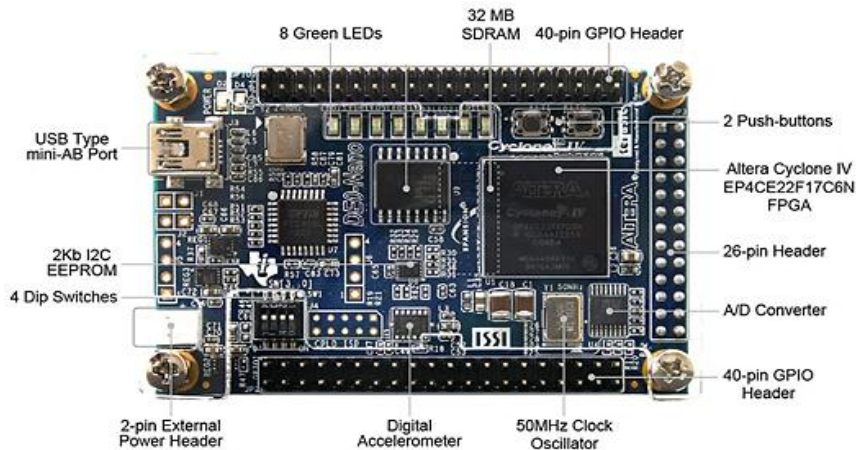


Figura 2.9: Tarjeta Deo-Nano y componentes. [26]

Plataforma de control Raspberry Pi

La Raspberry Pi es un pequeño dispositivo del tamaño de una tarjeta de crédito cuyas dimensiones son 85.6×53.98×17mm, es lo suficientemente pequeño y

liviano para cargar en la mano. Este dispositivo fue creado en el Reino Unido por los ingenieros Eben Upton, Rob Mullins, Jack Lang y Alan Mycroft[27]. Este pequeño ordenador es capaz de hacer todo lo que se espera de una computadora convencional, desde la navegación por Internet y reproducción de vídeos de alta definición, hasta la realización de hojas de cálculo, procesador de textos, y jugar juegos. Lo que, es más, la Raspberry Pi tiene la capacidad de interactuar con el mundo exterior a través de sus GPIO (General Purpose I/O Expansion Board o Entradas y Salidas de Propósito General), los cuales se han utilizado en una amplia variedad de proyectos digitales[28].

En cuanto al SO El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la Fundación Raspberry Pi (en su pagina web oficial) (ver Figura 2.10)[29].

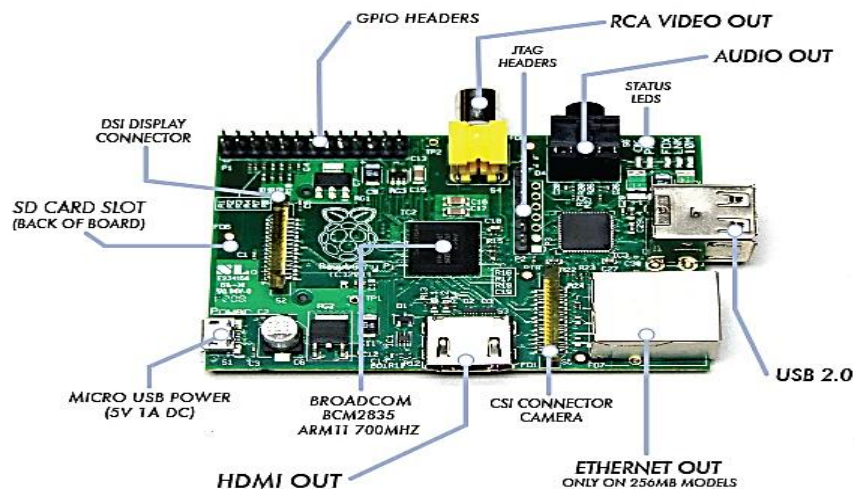


Figura 2.10: Tarjeta Controladora Raspberry Pi 2 y sus componentes. [29]

Sensores

En un sistema electrónico, el sensor es el elemento dispuesto expresamente para obtener información. La palabra sensor es el nombre popular con el que se conocen los “transductores”. Se fabrican gran variedad de transductores, capaces de convertir cualquier parámetro físico, químico o biológico en una magnitud eléctrica[30].

Sensores empleados en la robótica

Los sensores ayudan a trasladar los atributos del mundo físico en valores que la controladora de un robot puede usar. En general, la mayoría de los sensores pueden ser divididos en dos grandes grupos[30]:

- **Sensores analógicos:** un sensor analógico es aquel que puede entregar una salida variable dentro de un determinado rango. Un Sensor analógico, como por ejemplo una Fotorresistencia (estos componentes miden intensidad de luz), puede ser cableado en un circuito que pueda interpretar sus variaciones y entregar una salida variable con valores entre 0 y 5 volts.
- **Sensores Digitales:** un sensor digital es aquel que entrega una salida del tipo discreta. Es decir, que el sensor posee una salida que varía dentro de un determinado rango de valores, pero a diferencia de los sensores analógicos, esta señal varía de a pequeños pasos pre-establecidos.

Magnitudes físicas que es necesario medir para que un robot tenga algún conocimiento del entorno:

- Luz (con su gama de espectro: visible, infrarroja, ultravioleta)
- Sonido y ultrasonido
- Gravedad (inclinación, posición)
- Temperatura
- Humedad
- Presión y/o fuerza
- Velocidad
- Magnetismo
- Ubicación
- Proximidad
- Distancia

Servomecanismos

Un servo es un dispositivo pequeño que tiene un eje de rendimiento controlado. El servomotor puede ser llevado a posiciones angulares específicas al enviar una señal codificada(PWM)[31], dentro de un rango de operación (generalmente de 180°) y mantenerse estable en dicha posición. Los servos se suelen utilizar en robótica, automática y modelismo (vehículos por radio-control, RC) debido a su gran precisión en el posicionamiento[32].

Las señales PWM (Pulse Width Modulation) utilizadas para controlar los servos están formadas por pulsos positivos cuya duración es proporcional a la posición deseada del servo y que se repiten cada 20ms (50Hz). Todos los servos pueden funcionar correctamente en un rango de movimiento de 90°, que se corresponde con pulsos PWM comprendidos entre 0.9 y 2.1ms. Sin embargo, también existen servos que se pueden mover en un rango extendido de 180° y sus pulsos de control varían entre 0.5 y 2.5ms[32].

Componentes internos de un servo

Los servos están compuestos por un amplificador, el servo motor, piñonería de reducción o juego de engranajes y un potenciómetro de realimentación; todo incorporado en el mismo conjunto, tal como se aprecia en la Figura 2.11.

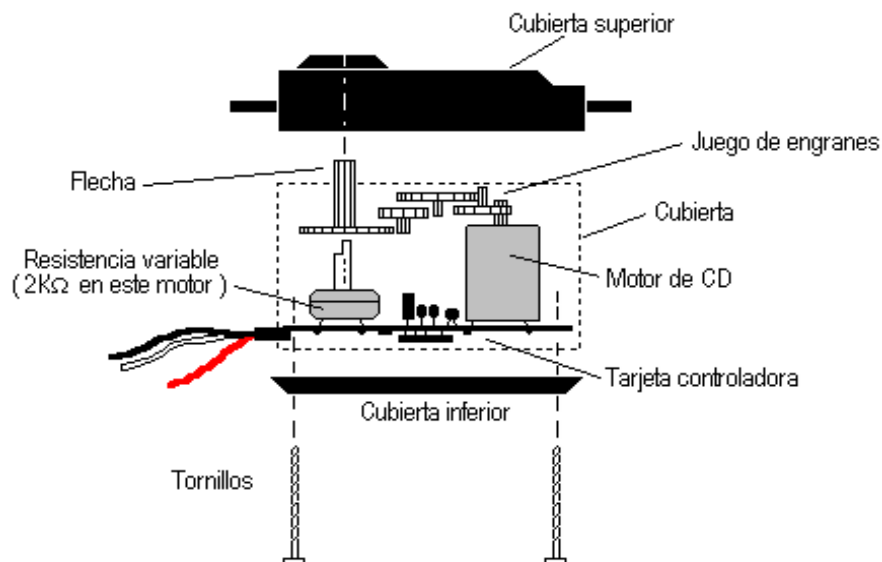


Figura 2.11: Componentes generales de un servomotor. [33]

Tipos de Ruedas

Entre los robots terrestres con ruedas su movilidad está caracterizada por dos factores[34]: el tipo de ruedas que poseen y su disposición sobre una estructura mecánica. Dentro del tipo de ruedas que están compuestos los robots terrestres, se encuentran: las ruedas convencionales fijas y el grupo de las ruedas omnidireccionales.

Rueda fija o común

El eje de la rueda está fijo a la estructura del robot. En general está asociada al sistema de tracción del robot[34].

Ruedas Omnidireccionales

Las ruedas omnidireccionales en todas sus configuraciones se basan bajo un mismo principio general, que tienen que ver con la inclusión de rodillos alrededor de su periferia, aumenta un grado de libertad adquiriendo movilidad en el eje perpendicular al sentido rodadura que tiene una rueda común.

La Figura 2.12 muestra la estructura básica de una rueda omnidireccional, de la cual se desarrollan ruedas omnidireccionales más complejas que mejoran el desplazamiento, el posicionamiento, la orientación, capacidad de carga, así como reducir la complejidad del controlador[35].



Figura 2.12: Rueda Omnidireccional con rodillos en su periferia doble. [36]

Para el caso de un vehículo móvil omnidireccional basado en ruedas, se divide en dos grupos importantes como son: ruedas orientables y ruedas especiales.

- **Ruedas Orientales**

Dentro de las ruedas orientales, aquellas que permiten un desplazamiento omnidireccional son las ruedas orientables centradas, ya que cada rueda posee un motor que la orienta (dirección) y otro motor que le permite girar (tracción). Ver figura 2.13.



Figura 2.13: Mecanismo Omnidireccional con ruedas orientables. [37]

- **Ruedas Especiales**

Las ruedas especiales están basadas en la idea de incluir: una componente activa, que provee tracción en una dirección, y una componente pasiva, que así mismo provee tracción, pero en una dirección diferente a la activa tal como se presenta en la figura 2.14[35].



Figura 2.14: Componentes de una rueda universal. [36]

En este sub-grupo de las ruedas especiales, se incluyen las ruedas *Universales* y las ruedas *Mecanum*; ambas presentan rodillos ubicados en su periferia, pero dispuestos en ángulos diferentes, tal como se ilustra en la Figura 2.15.



Figura 2.15: Rueda Especial (a) Universal Doble (b) Mecanum. [38]

Los rodillos ubicados en la periferia de la rueda universal son los que añaden una componente pasiva perpendicular a la componente activa. Cuando tres o más de estas ruedas son utilizadas en un robot, sus componentes activas y pasivas combinadas permiten realizar movimientos omnidireccionales. En las Ruedas Mecanum, la componente pasiva forma un ángulo de 45 grados con respecto a la activa[35].

Fuente de alimentación (Baterías)

Una batería es un dispositivo que convierte la energía química contenida en sus materiales activos en energía eléctrica a través de una reacción electroquímica. Las baterías pueden ser[39]:

- **Primarias**, las baterías primarias son utilizadas solamente una vez, debido a que sus reacciones químicas internas, que suministran la corriente eléctrica, son irreversibles.
- **Secundarias**, en las baterías secundarias las reacciones químicas, que suministran la corriente eléctrica, son reversibles; es decir, que haciendo circular una corriente eléctrica continua a través de la batería, en sentido contrario a la corriente de descarga, se pueden recuperar los compuestos químicos internos originales que suministran la corriente eléctrica en una batería cargada; de esta manera este tipo de baterías pueden ser usadas y nuevamente cargadas para su reutilización.

2.2.8 Disposición de las ruedas sobre la estructura mecánica

A continuación, se presenta un resumen sobre el diseño de diversos tipos de robots móviles basado en ruedas, que se diferencian entre sí por su grado de maniobrabilidad.

Robot Uniciclo

El robot tipo uniciclo es, en general, el elegido por los investigadores a la hora de probar nuevas estrategias de control por tener una cinemática sencilla. Es una estructura que consta de dos ruedas fijas convencionales sobre el mismo eje, controladas de manera independiente y una rueda loca que le confiere estabilidad (Figura 2.16). El sistema de tracción-dirección asociado al robot le permite independizar las consignas de la velocidad lineal y angular respectivamente. Las ventajas que se derivan de la estructura mecánica y de la electrónica de control hacen de esta configuración la preferida para robots de laboratorio[37].

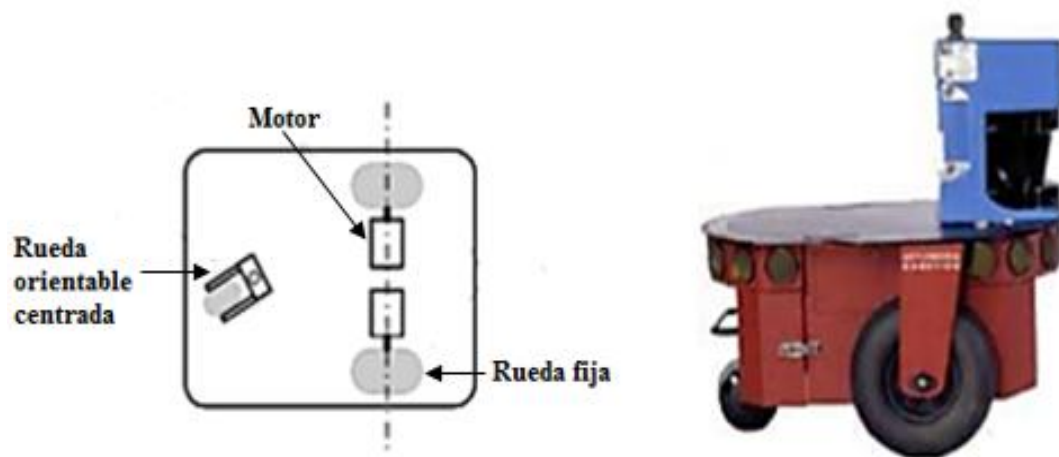


Figura 2.16: Robot tipo Uniciclo. [37]

Triciclo

El robot tipo triciclo está formado por dos ruedas convencionales fijas sobre un mismo eje y una rueda convencional centrada orientable que concentra las funciones de tracción-dirección (Figura 2.17).

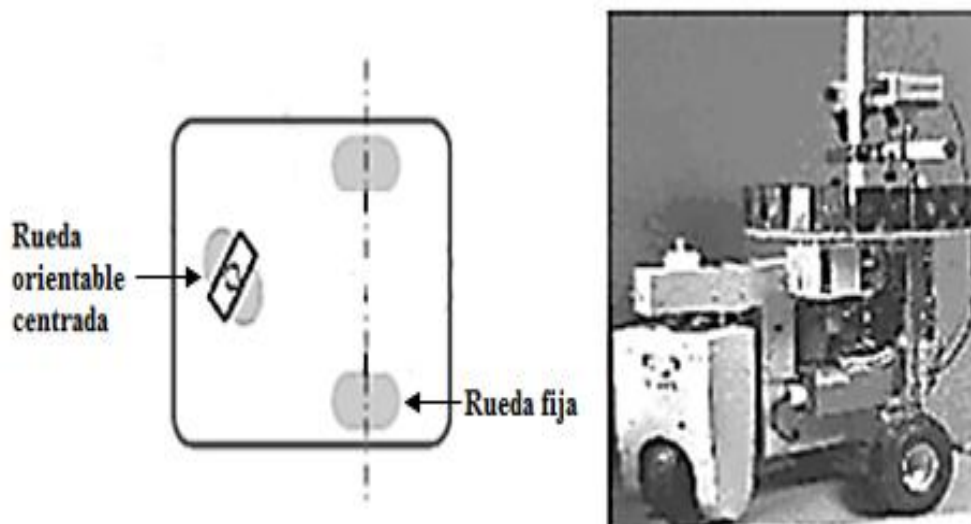


Figura 2.17: Robot móvil tipo Triciclo.[37]

Al igual que en el caso anterior, la estructura mecánica y la electrónica de control son sencillas, su tratamiento cinemático resulta de interés en áreas específicas del control de robots móviles. En cuanto a aplicaciones industriales esta configuración es apta para el transporte de cargas pesadas a baja velocidad[37].

Cuatriciclo

Un Problema asociado con la configuración tipo triciclo es que el centro de gravedad del vehículo se posiciona, en algunas ocasiones, en los límites de la superficie de equilibrio, definida por tres ruedas, cuando el vehículo está en movimiento.

Esto produce una pérdida de tracción en el vehículo y es fuente de error a la hora de estimar la posición del robot. Una solución a este problema lo presenta el sistema de dirección Ackerman.

Como se observa en la Figura 2.18 los ejes de las dos ruedas frontales se interceptan en un punto C que pertenecen al eje común de las ruedas traseras. El lugar de los puntos en el plano trazados por cada rueda, alrededor de este punto C, es un conjunto de arcos concéntricos donde todos los vectores velocidad instantánea son tangentes a estos arcos. Esta estructura, además de brindar mayor estabilidad, evita el deslizamiento en la rueda y por lo tanto reduce los errores de odometría.

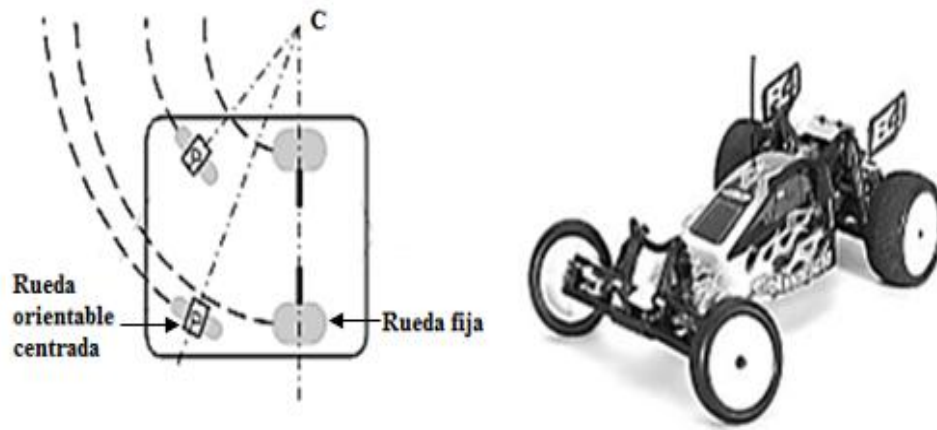


Figura 2.18: Sistema de dirección Ackerman. [37]

Robot Omnidireccional

En la robótica móvil, los robots omnidireccionales presentan una de las estructuras más eficientes para realizar movimientos frontales, laterales y rotacionales sobre una superficie plana, una de sus principales características es el tipo y la disposición de sus ruedas, tal como se presenta en la Figura 2.19[35].



Figura 2.19: Estructura de un robot omnidireccional. [35]

Un robot omnidireccional se define como un vehículo que cuenta con movilidad en cualquier dirección desde un punto arbitrario del sistema de coordenadas sin realizar rotaciones previas al desplazamiento, llegando a su destino con la orientación deseada (sistema holonomos). Esta condición se debe a que la

disposición de las ruedas brinda al robot tres grados de libertad en el plano xz, movimiento en el eje x, z y orientación o también llamado guiñar (yaw).

El principal componente de los robots omnidireccionales son los actuadores que posibilitan su desplazamiento; en este caso es su sistema de motorización, transmisión, sus ruedas omnidireccionales y la distribución de las mismas en el chasis del vehículo[35].

2.2.9 Estructuras de Robots Omnidireccionales

En el mundo académico e industrial se han estudiado varias estructuras de robots omnidireccionales, de distinta cantidad de ruedas y disposición, de los cuales los más relevantes se describen a continuación.

Estructura Omnidireccional de 3 ruedas

Esta estructura consiste en tres ruedas omnidireccionales dispuestas como se muestra en la Figura 2.20. Este tipo de estructuras posee la ventaja de que su control es simple, ya que para cada velocidad deseada del vehículo existe una única combinación en las velocidades de las ruedas (a diferencia de los vehículos omnidireccionales de más de 3 ruedas). A su vez, una determinada combinación de velocidades en las ruedas determina una única velocidad del vehículo[40].



Figura 2.20: Plataforma Omnidireccional de tres ruedas. [41]

Estructura Omnidireccional de 4 ruedas

Esta estructura consiste en una aproximación similar al omnidireccional de 3 ruedas; en la Figura 2.21 se puede ver la disposición de las ruedas. Este tipo de estructuras posee la ventaja de que tiene redundancia, o sea que para lograr cualquier movimiento en el plano, existen varias combinaciones de velocidades posibles. Otra de las ventajas que presenta esta estructura es que permite detectar si existe deslizamiento en una rueda de forma de tratar el problema[40].



Figura 2.21: Plataforma Omnidireccional de cuatro ruedas. [42]

2.2.10 Servidor Web

Un servidor Web es el corazón de Internet, entre otros importantes dispositivos. Todo ordenador tiene un servidor o máquina dedicada, la cual está conectada a Internet para dar disponibilidad a sitios Web, cuando son solicitadas por usuarios de la red. Este ordenador o máquina, es conocida como servidor Web[43].

Este equipo acepta las peticiones “http” del navegador Web del usuario, y entrega las páginas Web que podemos visualizar, las cuales suelen ser páginas HTML y objetos enlazados, imágenes, ficheros, etc (ver Figura 2.22).

En otras palabras, cuando alguien quiere ver una página Web en cuestión pinchando un enlace, el ordenador lanza una petición al servidor Web donde dicha página está alojada[43].

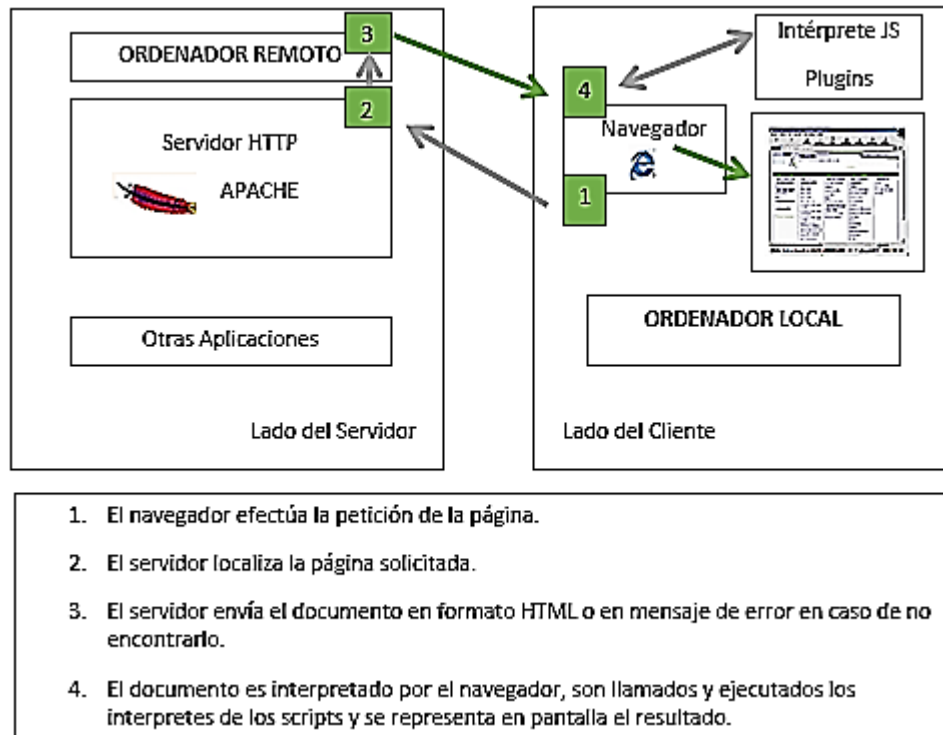


Figura 2.22: Páginas dinámicas usando aplicaciones del lado del cliente.[44]

Lenguajes de Programación del lado servidor o cliente

Lenguajes del lado servidor: son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él, por ejemplo[44]:

- **CGI**

Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente se encuentra un poco desfasado por diversas razones entre las que destaca la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta.

- **PHP**

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse.

- **PERL**

Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes.

Lenguajes del lado cliente: los lenguajes del lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

A continuación, daremos una introducción de los diferentes lenguajes de programación del lado cliente[44][45]:

- **Lenguaje HTML**

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

- **Lenguaje Javascript**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

- **FLASH**

Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Flash en realidad no es un lenguaje propiamente dicha; Sin embargo, si se

tuviese la necesidad de que catalogarlo en algún sitio quedaría dentro del ámbito de las páginas dinámicas de cliente.

- **CSS**

CSS, es una tecnología que permite la creación de páginas web de una manera más exacta. Mediante la inclusión de las CSS (por sus siglas en inglés, Cascading Style Sheets) es posible aplicar ciertos estilos que no son posibles hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc. Dando como resultado una mejor apariencia a la página web.

- **Lenguaje Python**

Python es comprado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar.

Permite la creación de todo tipo de programas incluyendo los sitios web. Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

2.3 Propuesta de solución

Mediante la construcción de un robot móvil con diseño esférico, se pretende estimular la ejecución de proyectos de investigación formativa y generativa. Además, con la presentación de un modelo robótico se intenta aportar nuevos criterios para la construcción de prototipos anteriormente no considerados en la Facultad de Ingeniería en Sistemas Electrónica e Industrial. Se prevee con ello la integración de ramas a fines con la construcción del mismo.

CAPITULO III

METODOLOGÍA

3.1. Modalidad de la investigación

El enfoque de esta investigación es aplicada, debido a que se emplean los conocimientos científicos y técnicos para brindar una solución al problema existente, utilizando la tecnología actual.

Las modalidades de investigación son las siguientes:

3.1.1 Investigación bibliográfica

Es bibliográfica puesto que se ha recurrido a diferentes medios de estudio para ampliar y profundizar conceptos, temas y criterios de los diversos autores y aspectos del estudio, se fortalecerá la información con artículos publicados en las páginas web del internet, además en tesis referenciales realizadas de las diferentes instituciones educativas del país y extranjeras.

3.1.2 Investigación experimental

Es de carácter experimental ya que se prevé realizar pruebas con diferentes equipos y dispositivos disponibles actualmente en el mercado, hasta determinar los tipos adecuados para el desarrollo óptimo de la propuesta.

3.2. Población y muestra

Para el presente proyecto de investigación no se contará con población, puesto que se realizará una investigación aplicada.

3.3. Recolección de información

La recolección de información se realizará revisando fuentes Online y libros en búsqueda de documentación necesaria que aporte al desarrollo de la investigación;

además el uso de las guías prácticas permitirá esclarecer la idea del diseño del proyecto, así como la selección de los componentes que formarán parte del mismo.

3.4. Procesamiento y análisis de datos

La información recolectada se someterá a una revisión crítica, para poder realizar una limpieza de información defectuosa, incompleta o no pertinente, así mismo descartar redundancias en el diseño y también la elección de las mejores alternativas en cuanto al diseño de cada uno de los elementos y accesorios presentes en el prototipo del robot móvil esférico.

3.5. Desarrollo del proyecto

Los aspectos básicos al momento del desarrollo del proyecto se detallan a continuación:

1. Analizar los antecedentes sobre los robots móviles esféricos, y clasificar la información más relevante acerca del diseño, construcción y aplicabilidad de los mismos.
2. Analizar los distintos tipos de mecanismos de desplazamiento para los robots tipo esfera.
3. Determinar los principales requerimientos para mejorar el desplazamiento del prototipo.
4. Determinar el material adecuado del cual estarán constituido, tanto las piezas internas del prototipo, como también el armazón esférico externo.
5. Construir las piezas que servirán de soporte para todos los componentes que se alojaran en el interior del prototipo.
6. Construir el armazón esférico una vez que se haya fijado las dimensiones del mecanismo interno.

7. Establecer una configuración inicial de la tarjeta controladora, sensores, actuadores y dispositivos afines al sistema de movilización y mando.
8. Acondicionar los componentes electrónicos y mecánicos que formarán parte del mecanismo de movilización.
9. Elegir un entorno de desarrollo gráfico de acuerdo a la tarjeta controladora.
10. Diseñar la interfaz gráfica que permita el control inalámbrico entre el prototipo y el usuario.
11. Probar el rendimiento tanto del prototipo como de la interfaz gráfica controladora y la comunicación entre sí.
12. Analizar los Resultados.

CAPITULO IV

DESARROLLO DE LA PROPUESTA

4.1 Introducción

Para realizar el control inalámbrico del prototipo esférico, se debe seguir un complejo desarrollo sistemático por lo cual en la propuesta se explica de manera detallada la construcción de un cuerpo capaz de desenvolverse en ciertos entornos sin perder su estabilidad, así como también, el estudio mecánico cuyos resultados son influyentes en el desplazamiento del robot, ya que está basado en fundamentos físicos como el centro de masa y movimiento de rodadura.

Se describe también el hardware y software utilizados, considerando y aprovechando la característica de cada una de estas herramientas como parte fundamental para conseguir la interacción entre el usuario y el prototipo.

Para la construcción de las piezas que albergarán todos los componentes de hardware, se proveerá distribuir el espacio interno de una manera óptima, haciendo uso de la tecnología de impresión 3D, ya que es eficiente, cómodo y económicamente moderado, así como también, de otras tecnologías actualmente disponibles.

4.2 Análisis de factibilidad

Los apartados siguientes del proyecto de investigación, representan las instancias de factibilidad para el desarrollo de la propuesta.

4.2.1 Factibilidad Técnica

El proyecto de investigación desde el punto de vista técnico es factible desarrollarlo debido a que el investigador cuenta con los materiales y equipo electrónico e informático ideal para la implementación del prototipo.

4.2.2 Factibilidad Económica

La propuesta de un robot móvil esférico, tiene apertura positiva en la factibilidad económica porque el investigador cuenta con los recursos económicos necesarios. El investigador en conjunto con la Facultad de Ingeniería en Sistemas, Electrónica e Industrial están conscientes que invertir en este proyecto, significa promover nuevos criterios para la construcción de prototipos anteriormente no considerados.

4.3 Análisis de requerimientos

4.3.1 Hardware

El siguiente proyecto de investigación se lo puede dividir de la siguiente forma, una que corresponde a todo lo concerniente al análisis mecánico lo cual involucra también la fabricación de las piezas que servirán de soporte al hardware del prototipo; una segunda que comprende la programación y las comunicaciones avanzadas, cuyos resultados deberán verse plasmados en una interfaz hombre-máquina para el control inalámbrico, ya sea desde un ordenador o un dispositivo móvil. Y finalmente, los resultados como la presentación del prototipo.

A continuación, realizaremos una selectividad de los componentes que conformarán el prototipo.

Plataforma de control SBC

Debido a las dimensiones del prototipo se necesita un mini Ordenador de placa SBC (Single-Board Computer) es decir un ordenador de placa única, en cual se pretende levantar el servidor web con el propósito de que el usuario pueda acceder y controlar el prototipo desde un navegador web, aprovechando cada uno de sus componentes físicos, así como también procesar ciertos algoritmos del sistema. En la Tabla 4.1 se presenta una comparativa completa de todos sus parámetros físicos de las plataformas actualmente existentes en el mercado.

Tabla 4.1: Comparación entre los distintos modelos SBC.

Modelos	CPU	GPU	RAM	USB	Video	Audio	Boot	Red	Consumo	Alimentación	Tamaño	SO	Precio
BeagleBone Black	AM335x 1GHz ARM® Cortex-A8	SGX530 aceleradora 3D	256 Mb	2	mini HDMI	HDMI	Interna de 2GB eMMC,m icro SD	Ethernet 10/100	460mA/5V	MicroUSB, jack	85,6x53,98 mm	AngstromL inux Android Ubuntu	\$ 45,00
Odriod-C1	1.5GHz quadcore ARM® Cortex®-A5	dual core Mali™-450MP 600MHz	1 Gb	5	micro HDMI	Micro HDMI	Micro SD	Ethernet 10/100	400mA/2W /5V	MicroUSB	85x56 mm	Ubuntu, Android	\$ 35,00
DragonBoard 401c	1.2 GHz quad- core ARM Cortex A53	Qualcomm Adr eno 306 400MHz GPU	1Gb	3	micro HDMI	HDMI,	Micro SD	WiFi 802 .11a/b/g/ n	500mA/2,5 W/5V	MicroUSB/GPIO	85x56 mm	Android 5.1, Ubuntu, Windows 10	\$ 75,00
Udoo Neo	NXP® i.MX 6SoloXARM Cortex-A9 core	Video Core IV	1 Gb	2	Micro HDMI, LVDS	HDMI, S/PIDF & I2S*	Micro SD	Ethernet 10/100, WiFi 802.11a/ b/g/n	800mA/4W /5V	Micro USB, DC Power Jack, RTC Battery Connector	89x59 mm	Android Lollipop, Linux UDOObunt u2	\$ 50,00
Raspberry Pi 2 Model B	1.2GHz QUAD ARM Cortex-A	Video Core IV	1 Gb	4	Jack, HDMI	Jack, HDMI	Micro SD	Ethernet 10/100	2,5A/12,5 W/5V	MicroUSB/GPIO	85x56 mm	Ubuntu, Raspbian, Winodows 10, entre otros	\$ 75,00

Elaborado por: Investigador

Después de haber comparado las diferentes características que tiene cada una de las plataformas Raspberry Pi, se optó por elegir el modelo Raspberry Pi 2 Model B principalmente por su capacidad de procesamiento, por la cantidad de pines de entrada y salida para poder interactuar con el medio exterior, de igual forma por la cantidad de puertos USB suficientes para el control de los servomotores marca Dynamixel y la comunicación inalámbrica mediante un adaptador Wifi, además cuenta con documentación, de costo moderado y siendo el único modelo disponible dentro de la zona geográfica al momento para el desarrollo del proyecto.

No obstante, su voltaje y corriente de operación obliga a la adquisición de una placa electrónica que sea capaz de proporcionar dichos valores de consumo. Internamente no cuenta con un Convertidor Análogo-Digital, por lo que si se desea procesar este tipo de señales analógicas se necesita de un circuito externo capaz de realizar ésta tarea.

De igual manera, perteneciente al sistema de control del robot esfera. Se analizará la inclusión de una plataforma Arduino Uno cuya tarea estará enfocada a la estabilidad del prototipo.

Plataforma de control basado en AVR

A continuación, en la tabla 4.2 se presenta de forma resumida los parámetros físicos de las plataformas actualmente existentes en el mercado.

Tabla 4.2: Especificaciones técnicas de distintas plataformas de control con AVR.[23]

Modelo	Arduino Uno	Almond V1.2	Tessel	Launchpad MSP430G2
Microcontrolador	ATMega328	ATMega 2560	Cortex-M3 LPC1830	MSP430G2553
Voltaje de operación	5V	5V	5V	1.8 – 3.6 V
Voltaje de entrada (limite)	6-20V	6-12V	5-15V	7-12V
Pines	14	9	6	-

entrada/salida digital				
Pines entrada analógica	6	2	6	8
Memoria Flash	32 Kb (0,5 Kb ocupados por el bootloader)	256Kb	32 Mb	16kB
SRAM	2 Kb	-	32 Mb	512B
EEPROM	1 Kb	4Kb	-	-
Frecuencia de reloj	16 MHz	16 MHz	180 MHz	16 MHz
Dimensiones	68.66 x 53.4 mm	45x60x15mm	65x55.5 mm	160x154,94x43,18 mm
Peso	25 gr	20 gr	33 gr	150,252
Precio	\$ 15,00	\$ 25,00	\$ 1200,00	\$167,00

Dentro de los requerimientos de instrumentación y control del presente proyecto , la plataforma Arduino Uno se adapta a las necesidades del mismo. En cuanto, a sus especificaciones técnicas como el consumo de voltaje y corriente es relativamente bajo, lo cual se supone se podrá compartir la alimentación con la plataforma de control raspberry Pi, posee también entradas/salidas digitales para la lectura de sensores como en el caso de la MPU-6050 también conocido como chip de movimiento inercial digital, entradas analógicas para otro tipo de sensores y la capacidad de ejecución de algoritmos de control de manera fluida, como el control PID que de igual manera se lo analizará más adelante.

Sensor acelerómetro-giroscopio

El circuito integrado MPU-6050 contiene un acelerómetro y giroscopio MEMS en un solo empaque. Cuenta con una resolución de 16-bits, lo cual significa que divide el rango dinámico en 65536 fracciones, estos aplican para cada eje X, Y y Z al igual que en la velocidad angular. El sensor es ideal para diseñar control de robótica, medición de vibración, sistemas de medición inercial (IMU), detector de caídas, sensor de distancia y velocidad, entre otras[46]. El MPU-6050 se comunica a través del protocolo I2C proporcionando unos valores conocidos como raw o “crudos” según el registro seleccionado. Opera con 3.3 voltios, aunque algunas versiones llevan un regulador que permite conectarla a 5V[47].La finalidad de incluir este chip integrado, es que sirva de variable controlada del

sistema para así aplicar correcciones a través de la variable manipulada y obtener el valor deseado. Además, se constituye como un dispositivo alcanzable dentro de la zona geográfica y cuenta también con gran soporte en la web sobre el tratamiento de los datos que emite, tal como se puede observar en el apartado 4.8 .

Dispositivo de comunicación inalámbrica

El dispositivo necesario para la comunicación entre el hardware localizado en el interior de la carcasa esférica y el dispositivo móvil personal u ordenador, deberá ser de largo alcance, fácil manejo y compatible con el hardware y software a emplearse; para ello se optó por utilizar el estándar de comunicación IEEE 802.11x. La plataforma de control seleccionada no está dotada internamente de una comunicación WiFi y mucho menos Bluetooth. No obstante, en el desarrollo del proyecto se verá la necesidad de incluir un dispositivo con mejores prestaciones en cuanto a velocidad y alcance, por lo que se hará una comparativa de los adaptadores Wi-Fi vía USB que permita dicha comunicación. Cabe recalcar que los dispositivos móviles tanto como los ordenadores portátiles actuales poseen una comunicación inalámbrica, por lo que no presentan ningún inconveniente.

En la tabla 4.3, se presenta unos cuantos adaptadores útiles para una comunicación wi-fi.

Tabla 4.3: Comparación de adaptadores Wi-Fi.

			
Adaptadores WiFi/Modelos	EDUP EP-N8508	Ralink RT5370	TP-LINK TL-WN725N
Banda de frecuencia:	2.4 ~ 2.4835GHz	2.4-2.4835GHz	2.400-2.4835GHz
Antena	Interna	Externa (150 mm)	Interna

Estándar Inalámbrico	IEEE 802.11n	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b	IEEE 802.11b, IEEE 802.11g, IEEE 802.11n
Velocidad de transmisión	11n: hasta 150Mbps	11n: hasta 150Mbps 11g: hasta 54Mbps 11b: hasta 11Mbps	11n: hasta 150Mbps 11g: hasta 54Mbps 11b: hasta 11Mbps
Interfaz	USB 2.0	USB 2.0	USB 2.0
Distancia de trabajo efectiva	Hasta 150 metros	Hasta 130 metros	Hasta 130 metros
Sistema de apoyo	Windows 8.1/8, Windows 7, Windows Vista, Windows XP, Mac, Linux	Windows 8.1/8, Windows 7, Windows Vista, Windows XP, Mac, Linux	Windows 8.1/8, Windows 7, Windows Vista, Windows XP, Mac, Linux
Dimensiones	18 x 15 x 6 mm	47 * 22 * 11mm (sin antena)	18.6x15x7.1mm
Peso	2 gr	16 gr	2.1 gr
Precio	\$7,99	\$ 11,99	\$ 9,99

Elaborado por: Investigador




Una vez analizado las especificaciones técnicas de los modelos propuestos en la Tabla 4.3, se descarta la utilización del Ralink RT5370 por sus dimensiones, a pesar de que sería una excelente elección por sus características. Sin embargo, el modelo TP-LINK TL-WN725N se ajusta a las necesidades del prototipo, en cuanto a alcance, velocidad de transmisión, dimensiones y sobre todo por la particularidad que es Plug n' Play.

Servomecanismos

Servomotor inteligente

Son actuadores modulares inteligente que incorpora un microcontrolador el cual es capaz de entender cierto numero de comandos dependiendo del modelo, la mayoría de los cuales fijan o leen parámetros que definen su comportamiento, como la capacidad de controlar su velocidad, temperatura, posición, tensión y carga soportada. A demás de tener un circuito de control especializado de interconexión entre sí. A continuación, se realizó una comparación de los modelos existentes (ver Tabla 4.4).

Tabla 4.4: Servomecanismos inteligentes.[48]

			
Modelo	XYZrobot A1-16	Dynamixel AX-12W	Feetech SCS115
Torque	350 oz-in	0,2N.m / 2,0 Kg-cm	17 Kg-cm
Velocidad (RPM)	70	470(a 12V, modo rueda) 54(a 12V, modo junta)	97
Voltaje de operación	9 ~ 12V (Rec. Voltaje 11.1V)	9 ~ 12V (Rec. Voltaje 11.1V)	6 ~ 8.4V
Corriente	0.5 A	1.4 A	1.5 A
Dimensiones	(50 x 32 x 40.5) mm	(32 x 50 x 40) mm	(40 x 20 x 39) mm
Peso	60 gr	52.9 gr	56 gr
Angulo de operación	300	300	200
Material del tren de engranajes	Engineering Plastic	Engineering Plastic	Engineering Metal
Sensor de posición	Potenciómetro	Potenciómetro	Potenciómetro
Protocolo de Comunicación	TTL	TTL	TTL
Velocidad de Comunicación	1Mbps	1Mbps	38400 bps~1Mbps
Compliance/PID	-	Si	-
Precio	\$ 44,90	\$ 44,90	\$ 15,00

En base a las características presentadas en la tabla 4.4, se optó por adquirir dos servomotores de la serie AX-12W (driving motors), ya que en comparación a los otros dos modelos, este ofrece mayor velocidad (RPM), que en este caso dependerá de esta característica para que el robot esfera se desplace de una manera adecuada hacia adelante o atrás sobre el plano XZ. En cuanto al resto de parámetros no difieren tanto entre sí, excepto por el precio que es moderadamente aceptable.

Sin embargo, es necesario indicar que para el control de este tipo de servomotores es precisola utilización de un dispositivo conversor de señales, de fácil manejo y multifuncional, para ello la misma empresa Robotis ha desarrollado el USB2Dynamixel, tal como se describe a continuación.

USB2Dynamixel

USB2Dynamixel es un dispositivo que se utiliza para operar los servomotores Dynamixel directamente desde el Ordenador o en nuestro caso desde la plataforma de control Raspberry Pi 3 Model B, de manera que al conectar el USB2Dynamixel al puerto USB del ordenador, podamos escoger entre los conectores de 3P o 4P incluidos, para controlar varios Dynamixels conectados en cadena. Además, USB2Dynamixel se puede utilizar para cambiar de puerto USB a puerto Serial, esta función es muy útil en caso de que los controladores exclusivos Dynamixel como CM-2, CM-2 +, CM-5, y la CM-510 se conecten a un puerto USB, o cuando ZIG2Serial está conectado al puerto USB para controlar los robots de forma inalámbrica (ver Figura 4.1). Cuenta con librerías de soporte para programar en los entornos Matlab, Microsoft Visual Studio, Java, C/C++, C#,

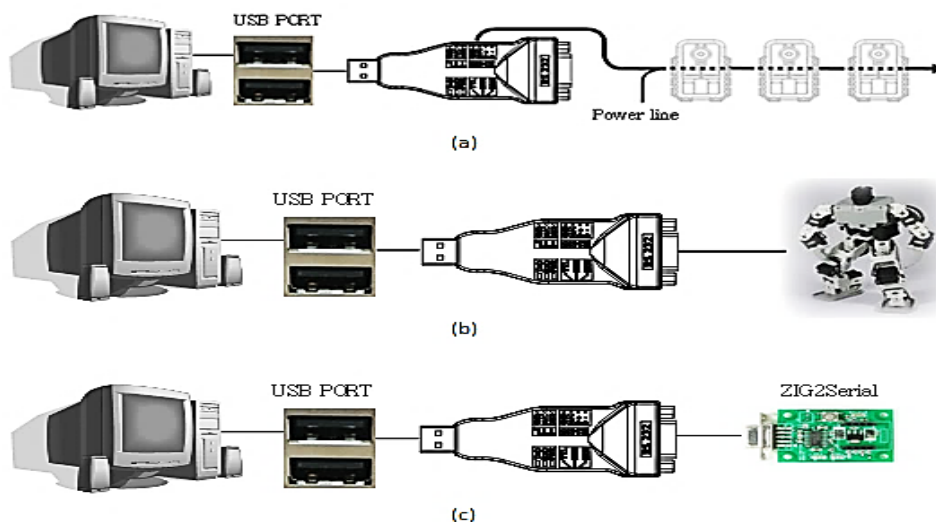


Figura 4.1: (a) Control Dynamixel a través del ordenador (b) Conversión a serial (c) Comunicación inalámbrica. [49]

Servomotor MG996R (High Torque)

La finalidad de incluir este tipo de servomotor, es la de proporcionar dirección (steering motor) al prototipo esférico, es decir, de acuerdo con la unidad interna de locomoción (UIL) basado en péndulo, éste se encargará de sacar del estado de reposo al mismo y como consecuencia alterar el centro de masa del prototipo en sí, generando movimientos laterales sobre el plano XZ, a diferencia de los motores de conducción o driving motors. Sin embargo, es preciso indicar las especificaciones de este servomotor (ver Tabla 4.5).

Tabla 4.5: Especificaciones técnicas servomotor MG996R.

Modelo	MG996R (High Torque)
Voltaje de Operación	4,8 V / 7,2 V
Corriente de Operación	500mA / 600mA
Velocidad de Operación	0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
Torque	9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
Dimensiones	40.7 x 19.7 x 42.9 mm approx.
Peso	55 gr
Engranajes	Metalicos
Cable / Conector	180mm 26AWG/JR
Precio	\$ 20,00

Elaborado por: Investigador

Alimentación

Las fuentes de alimentación o baterías desempeñan un papel importante, debido a que son las encargadas de proveer la energía necesaria a los dispositivos electrónicos distribuidos internamente dentro de la estructura esférica durante un lapso de tiempo considerable. Para lo cual se decidió emplear una batería recargable Li-Po de 11.1V y 1300 mAh (ver figura 4.2) la cual está destinada a alimentar los servomotores Dynamixel con el fin de obtener la máxima velocidad en RPM. Otra batería de Li-Po de 7.1 V y 1000 mAh cuya finalidad es energizar dos componentes electrónicos que son: la tarjeta Arduino Uno y el servo motor de dirección MG996R.

En cuanto a la alimentación de la Tarjeta Controladora RaspberryPi 2 Model B y todos sus periféricos conectados en ella, se empleó un banco de baterías conocido

como cargador portátil USB, cuya circuitería resultó ser lo suficientemente robusta para dicho propósito, está constituida por un par de baterías de Ion-Litio recargables con una capacidad de 4.7V y 5600 mAh, garantizando una autonomía de trabajo aceptable (ver Figura 4.3).



Figura 4.2: Bateria de Li-Po de 11.1 V a 1300 mAh.[55]

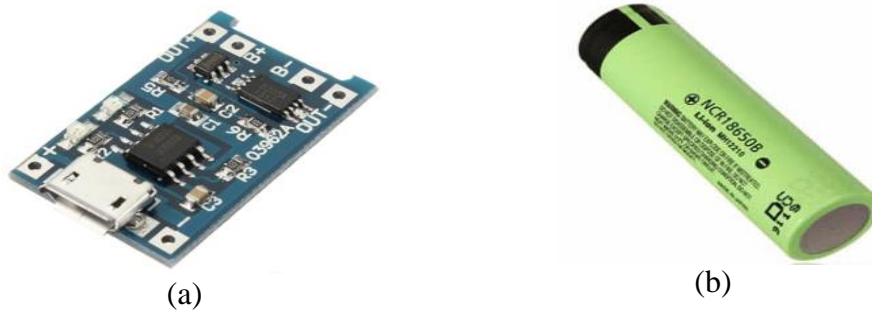


Figura 4.3: (a) Circuito fijador de voltaje de Carga y Descarga (b) Baterías Ion-Litio 4.7V a 5600mAh.[56]

4.3.2 Software

Distribuciones para Raspberry Pi

La **SBC (Single Board Computer) Raspberry Pi** ha cosechado grandes éxitos por su bajo precio y por sus grandes posibilidades, tanto para los desarrolladores de software, estudiantes de informática, como para los aficionados que desean emplearla para sus propios proyectos caseros[50]. Para ello es necesario inicializar una distribución que, de acuerdo con la página oficial Raspberry Pi Foundation, se puede encontrar una serie de Sistemas Operativos tanto oficiales como desarrollados por terceros a disposición.

Para el presente proyecto de investigación se hizo uso de la imagen Ubuntu MATE, más que todo por su estabilidad al correr en la tarjeta de control, por poseer una interfaz amigable con el usuario y sus actualizaciones más recientes permiten trabajar de forma fluida e intuitiva. Claramente desarrollada para propósitos específicos.

IDLE (Python)

Python es un lenguaje de programación con una sintaxis muy limpia y con un código legible. Se trata de un lenguaje de programación dinámico, orientado a objetos, es gratuito, de código abierto y muy fácil de usar[51].

Para el desarrollo de la investigación se utilizó Python 2, en principio porque es un lenguaje de programación libre y por defecto viene instalado en la gran mayoría de las versiones actuales de Linux. Además, por la existencia de paquetes y librerías compatibles con esta versión, útiles para establecer una comunicación entre la plataforma de control y el hardware correspondiente.

DreamWeaver

DreamWeaver como una de las herramientas bien consolidada y avanzada para el desarrollo de aplicaciones y sitios Web, la misma que a través de sus características y funcionalidades, fue seleccionada para el testeado de códigos de programación y el desarrollo de la interfaz, mediante la cual el Usuario podrá interactuar con el Robot esfera por medio de un ordenador o un móvil personal. Debido a los lenguajes de programación que incluye Dreamweaver como son: ASP, CCS, PHP, SQL, JSP, HTML y XML, se constituyen como el verdadero potencial del software en cuanto a la capacidad de combinar estos lenguajes citados, permitiéndonos la creación de dicha interfaz.

Herramienta de diseño y análisis SolidWorks

Para el diseño de la estructura interna del robot esfera, se realizó con la ayuda del software SolidWorks, ya que es una solución de diseño tridimensional con un

entorno gráfico basado en Microsoft Windows, fácil de manejar y desde el punto de vista de diseño; intuitivo. De lo contrario la misma página oficial de SolidWorks pone a disposición un sin número de videos guía online y foros, así como también, una vasta información por parte de terceros para el aprendizaje del mismo.

Integra un gran número de funciones avanzadas para facilitar el modelado de piezas, crear grandes ensamblajes, generar planos y otras funcionalidades que le permiten validar, gestionar y comunicar proyectos de forma rápida precisa y fiable.

Otras de las razones, por la que se empleó esta herramienta de diseño y análisis, es por su capacidad de verificar de inmediato el rendimiento gracias a los componentes 3D realistas que se comportan como lo harían en el mundo real.

4.4 Análisis del Mecanismo Interno

4.4.1 Análisis del centro de masa del prototipo

Se conoce que, el centro de masa es un punto o posición promedio en donde parece estar concentrada toda la masa de las partículas que forman un sistema o un objeto en particular.

El centro de gravedad coincide con el centro de masa debido a que en nuestro planeta el campo gravitatorio es uniforme, por tal razón es necesario realizar un análisis del centro de gravedad de forma general.

Centro de Gravedad

Consideremos un sistema de n partículas fijas contenidas en una región del espacio como se muestra en la Figura 4.4. Los pesos de las partículas forman un sistema de fuerzas paralelas que pueden ser reemplazado por un peso único resultante (equivalente) y un determinado punto de aplicación. A este punto se le llama centro de gravedad[52].

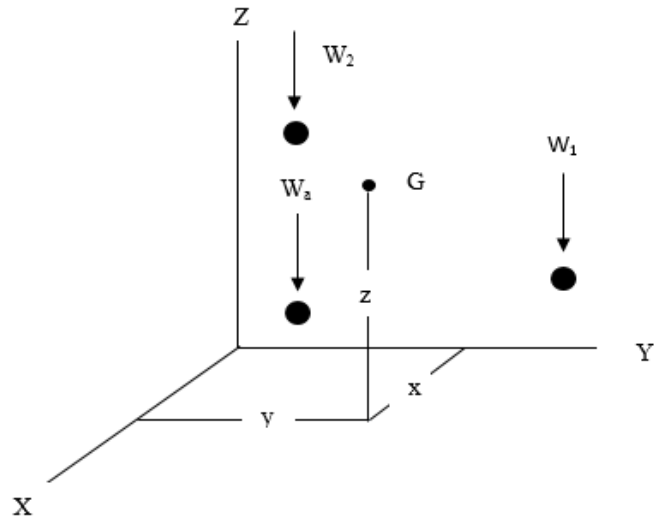


Figura 4.4: Sistema de n partículas. [52]

Para esto se requiere que el peso resultante sea igual al peso total de las n partículas; esto es,

$$W_R = \sum W \quad (1)$$

La suma de los momentos de los pesos de las partículas con respecto a los ejes X, Y y Z es, entonces, igual al momento del peso resultante con respecto a estos ejes.

Así, para determinar la coordenada de x de G, se puede sumar los momentos con respecto al eje Y. Esto nos da,

$$xW_R = x_1W_1 + x_2W_2 + \dots + x_nW_n \quad (2)$$

De manera similar, sumando momentos con respecto al eje x, es posible obtener la coordenada Y, es decir,

$$yW_R = y_1W_1 + y_2W_2 + \dots + y_nW_n \quad (3)$$

Aunque los pesos no generan un momento con respecto al eje z, se puede obtener la coordenada z de G imaginando el sistema coordenado, con las partículas fijas en él, como si hubiera girado 90 grados con respecto al eje X(o Y); (ver Figura 4.5). Sumando momentos con respecto al eje x[52]:

$$zW_R = z_1W_1 + z_2W_2 + \dots + z_nW_n \quad (4)$$

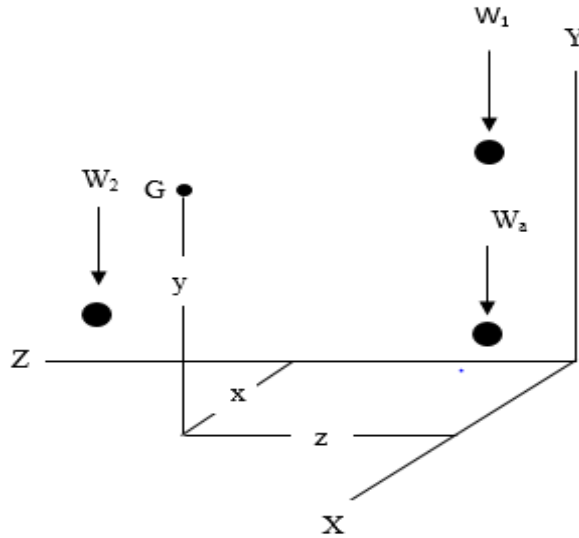


Figura 4.5: Sistema de n partículas con giro de 90 grados anti-horario con respecto al eje X. [52]

Se puede generalizar estas fórmulas y escribirlas simbólicamente en la forma siguiente,

$$x = \frac{\sum x'W}{\sum W} \quad (5)$$

$$y = \frac{\sum y'W}{\sum W} \quad (6)$$

$$z = \frac{\sum z'W}{\sum W} \quad (7)$$

Aquí x , y , z representan las coordenadas del centro de gravedad G del sistema de partículas, sin considerar la orientación de los ejes X, Y, Z. Para representar las coordenadas de cada partícula en el mismo sistema se emplea la siguiente simbología x' , y' , z' , lo cual difiere del sistema de partículas total. Y por último, $W_R = \sum W$, representa la suma total de los pesos de todas las partículas del sistema.

Centro de masa

Para estudiar los problemas referentes al movimiento de materia bajo la influencia de alguna fuerza, es decir, de la dinámica, es necesario ubicar un punto llamado centro de masa. Mientras la aceleración de la gravedad g para cada partícula sea constante, entonces $W = mg$. Si en (5), (6) y (7) se cancela g , tanto del numerador como del denominador, se tiene[52]:

$$x = \frac{\sum x' m}{\sum m} \quad (8)$$

$$y = \frac{\sum y' m}{\sum m} \quad (9)$$

$$z = \frac{\sum z' m}{\sum m} \quad (10)$$

Donde, $\sum m$ representa a la suma total de las masas de todas las partículas del sistema. Por comparación, se observa entonces que la ubicación del centro de gravedad coincide con la del centro de masa. Sin embargo, las partículas tienen un “peso” solo cuando están bajo la influencia ejercida por la atracción gravitacional, mientras que el centro de masa es independiente de la gravedad.

Por ejemplo, carece de sentido localizar el centro de gravedad de un sistema de partículas representando los planetas de nuestro sistema solar; sin embargo, es importante precisar su centro de masa.

Cálculo del centro de masa del prototipo

Se considera que la estructura del robot esférico está compuesto por un sistema de partículas fijas respecto al centro de la esfera como punto de referencia. Con ayuda del software de diseño y análisis SolidWorks, es posible localizar dicho punto de referencia y adicionalmente los centros de masa de los componentes que involucra el mecanismo interno, tal como se muestra en la Figura 4.6, con lo cual se pretende garantizar la estabilidad y el equilibrio del mismo.

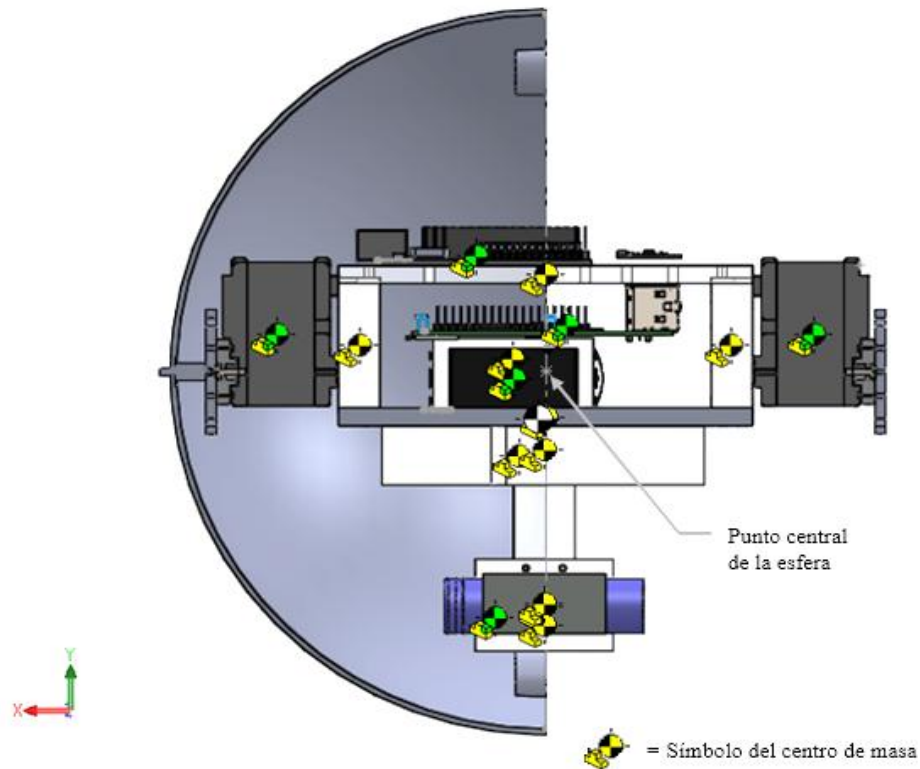


Figura 4.6: Punto medio del robot esférico y centros de masa de los elementos que lo conforman.

Elaborado por: Investigador

En la Tabla 4.6 se describe la masa y posición de los elementos respecto al punto central de la esfera.

Tabla 4.6: Elementos internos con su respectiva masa y coordenadas.

Elemento	Masa [gr]	Coordenadas de elementos (mm)		
		x	y	z
Tarjeta Raspberry Pi	45	6.67	15.99	6.67
Tarjeta Arduino Uno	28	22.93	39.38	6.60
Servomotor Dynamixel 1	52.9	87.71	12.61	1.74
Servomotor Dynamixel 2	52.9	87.31	12.61	1.74
Servo MG996R	55	10.59	2.19	25.60
USB2Dynamixel	40	8.80	28.19	1.74
Batería Li-Po	107	0.58	81.03	42.31
Batería Ion-Li	94	0.57	79.03	42.31
Plataforma General	65.64	0.03	15.24	1.73
Plataforma Secundaria	66.33	0.21	32.85	1.73
Columna Izquierda	38.62	60.74	8.77	1.74
Columna Derecha	30.07	60.40	8.76	5.61
Base Pendulo	44.97	0.63	85.62	1.74
Contrapeso Plomo	176	0.64	78.23	0.74

Elaborado por: Investigador

Para obtener el Centro de masa del prototipopropuesto, se considera(8), (9) y (10), indicadas anteriormente.

Centro de masa para la coordenada x:

$$x(\mathbf{mm}) = \frac{\sum xm}{\sum m} =$$

$$= \frac{6.67+22.93+87.71+87.31+10.59+8.80+0.58+0.57+0.03+0.21+60.74+60.40+0.63+0.64}{45+28+52.9+52.9+55+40+107+94+65.64+66.33+38.62+30.07+44.97+176}$$

$$= \frac{347.81}{896.43}$$

$$= 0,3879\text{mm}$$

Centro de masa para la coordenada y:

$$y(\mathbf{mm}) = \frac{\sum ym}{\sum m} =$$

$$= \frac{15.99 + 39.38 + 12.61 + 12.61 + 2.19 + 28.19 + 81.03 + 79.03 + 15.24 + 32.85 + 8.77 + 8.76 + 85.62 + 78.23}{45 + 28 + 52.9 + 52.9 + 55 + 40 + 107 + 94 + 65.64 + 66.33 + 38.62 + 30.07 + 44.97 + 176}$$

$$= \frac{500.5}{896.43}$$

$$= 0.5583 \text{ mm}$$

Centro de masa para la coordenada z:

$$z(\mathbf{mm}) = \frac{\sum zm}{\sum m} =$$

$$= \frac{6.67+6.60+1.74+1.74+25.60+1.74+42.31+42.31+1.73+1.73+1.74+5.61+1.74+0.74}{45+28+52.9+52.9+55+40+107+94+65.64+66.33+38.62+30.07+44.97+176}$$

$$= \frac{142}{896.43}$$

$$= 0,1584\text{mm}$$

Por lo tanto, el centro de masa del robot esférico se localiza en la posición CM= [0.3879, 0.5583, 0.1584]tomando como referencia el centro de la carcasa esférica.

Lo cual indica que el CM esta por debajo del punto de apoyo garantizando equilibrio y estabilidad.

4.4.2 Análisis de la unidad de manejo interno

Para ello, es necesario emplear ciertos conceptos básico sobre física que ayudarán en el análisis del comportamiento de la unidad interna del prototipo esfera.

Movimiento Circular

Cuando un cuerpo gira alrededor de un eje, sus puntos (partículas) describen trayectorias circulares en planos perpendiculares al eje. El movimiento realizado por cada una de estas partículas se denomina movimiento circular.

El análisis del movimiento circular se facilita si se hace coincidir el origen del sistema de referencia con el centro de la trayectoria, tal como se observa en la Figura 4.7.[53]

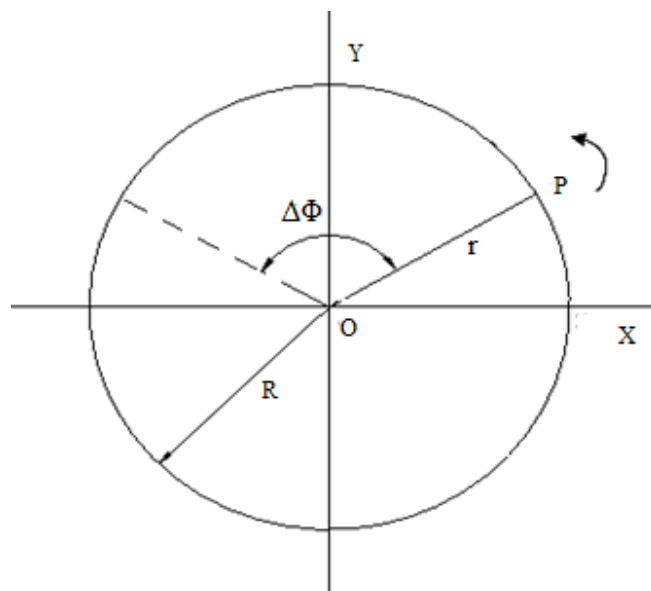


Figura 4.7: Movimiento circular de partículas.[53]

Donde, R corresponde al radio de la trayectoria, mientras la partícula P se desplaza por la trayectoria circular, su vector posición r barre ángulos centrales ($\Delta\theta$). Por

esto es conveniente definir variables de tipo angular que permitan analizar el movimiento.

Posición Angular

Es el ángulo θ que existe entre el vector posición de la partícula y un eje de referencia, que generalmente es X (ver Figura 4.8).

El ángulo θ , comúnmente se expresa en radianes, recordando que $180^\circ = \pi \text{ rad}$ [53].

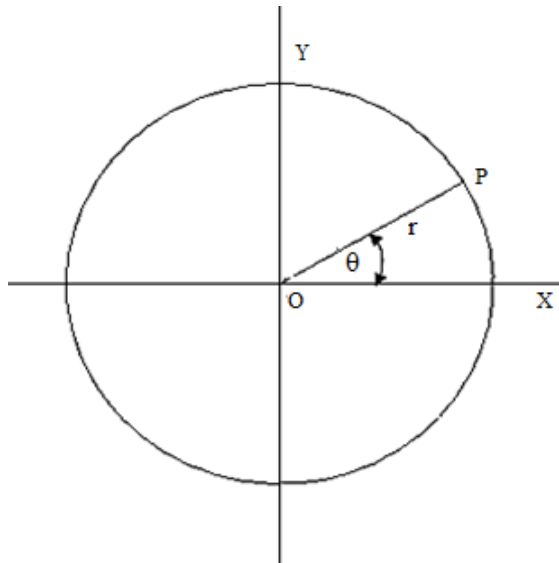


Figura 4.8: Angulo entre eje de referencia y vector posición. [53]

Todo ángulo medido en grados se puede convertir en radianes multiplicando el número de grados por $\frac{\pi}{180}$.

Todo ángulo medido en radianes se puede convertir en grados multiplicando el número de radianes por $\frac{180}{\pi}$.

Desplazamiento Angular

Es la variación neta de la posición angular de una partícula $\Delta\theta$, respecto de un sistema de referencia[53]. (ver Figura 4.9)

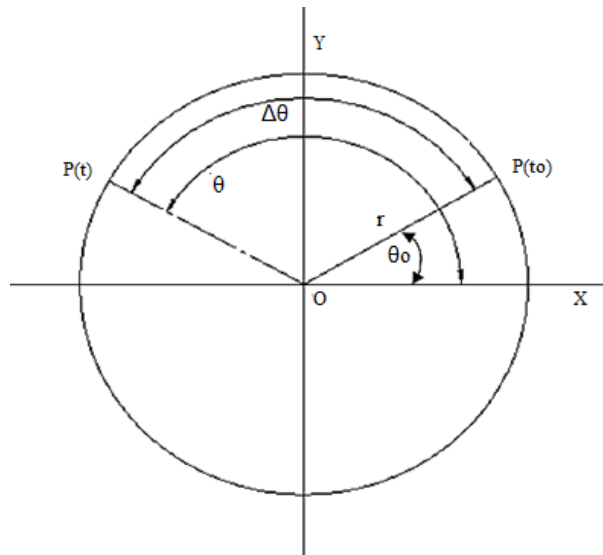


Figura 4.9: Variación angular de una partícula. [53]

$$\Delta\theta = \theta - \theta_0 \quad (11)$$

Donde, θ representa el ángulo de desplazamiento final de la partícula con respecto a un eje de referencia; θ_0 representa el ángulo de desplazamiento inicial de la partícula con respecto a un eje de referencia.

También se lo puede expresar como la relación entre el arco ΔS y el radio R con que ha sido trazado.

$$\Delta\theta = \frac{\Delta S}{R} \quad (12)$$

El desplazamiento angular $\Delta\theta$ se expresa en radianes.

Velocidad Angular Media

Es la razón entre el desplazamiento angular efectuado por la partícula y el tiempo invertido en dicho desplazamiento[53]:

$$\omega_m = \frac{\Delta\theta}{\Delta t} = \frac{\theta - \theta_0}{t - t_0} \quad (13)$$

Cuando la velocidad angular varía uniformemente, la ω_m es igual a la semisuma de las velocidades angulares inicial y final:

$$\omega_m = \frac{\omega_o - \omega}{2} \quad (14)$$

La velocidad angular ω se expresa en rad/s, en algunos casos es más cómodo utilizar RPM = $\frac{rev}{min}$, teniendo en cuenta que una revolución equivale a $2\pi rad$.

Aceleración Angular

Es la razón entre la variación de la velocidad angular que experimenta una partícula y el intervalo de tiempo en que se produjo[53]:

$$\alpha = \frac{\Delta\omega}{\Delta t} = \frac{\omega - \omega_o}{t - t_o} \quad (15)$$

La aceleración angular α se expresa en $\frac{rad/s}{s} = \frac{rad}{s^2}$

Relación entre velocidad lineal y velocidad angular

Cuando un disco gira con cierta rapidez, la velocidad lineal definida sobre la trayectoria y la velocidad angular definida sobre el ángulo barrido en un tiempo dado se producen de forma simultánea. Por lo tanto, es posible establecer una relación entre la velocidad lineal y la angular. La velocidad angular y el desplazamiento angular son respectivamente[54]:

$$\omega = \frac{\Delta\theta}{\Delta t} \quad (16)$$

$$\Delta\theta = \frac{\Delta S}{R} \quad (17)$$

Despejando $\Delta\theta$ de (16) se tiene la siguiente expresión:

$$\Delta\theta = \omega\Delta t \quad (18)$$

Igualando

$$\frac{\Delta S}{R} = \omega \Delta t \quad (19)$$

Reordenando

$$\frac{\Delta S}{\Delta t} = \omega R \quad (20)$$

Como

$$v = \frac{\text{arco}}{\text{tiempo}} = \frac{\Delta S}{\Delta t} \quad (21)$$

Entonces

$$v = \omega R \text{ ó} \quad (22)$$

$$v = \omega \frac{D}{2} \quad (23)$$

Por lo tanto, se puede definir que la velocidad lineal es directamente proporcional a la velocidad angular, siendo la constante de proporcionalidad el radio de giro. Con base en las ecuaciones (22) y (23), es posible calcular la velocidad aproximada con que el robot esfera se podrá desplazar.

De la Tabla 4.4, se indica la velocidad angular máxima en RPM de los servomotores Dynamixel.

Conversión de RPM a $\frac{rad}{s}$,

$$\omega_{ax12w} = 470 \text{ RPM} \left(\frac{2\pi \text{ rad}}{1 \text{ rev}} \right) \left(\frac{1 \text{ min}}{60 \text{ seg}} \right) \quad (24)$$

$$\omega_{ax12w} = 49,2183 \frac{rad}{s} \quad (25)$$

A partir de (25), se puede determinar la velocidad lineal que alcanzarán los motores de conducción (AX-12W), conociendo que el diámetro del eje central es de 50 mm, tal y como se indicó en apartados anteriores de este mismo capítulo.

$$V_1 = \omega_{ax12w} \frac{D_1}{2} \quad (26)$$

$$V_1 = 49,2183 \frac{rad}{s} \left(\frac{50 mm}{2} \right) \quad (27)$$

$$V_1 = 1,23 \frac{m}{s} \quad (28)$$

A partir de este resultado es posible calcular la velocidad con que se desplazaría el robot esfera empleando dicho mecanismo. Para ello es necesario partir de la siguiente relación matemática, que se ve a continuación

$$\frac{\omega_1}{\omega_2} = \frac{D_2}{D_1} \quad (29)$$

Donde, ω_1 es la velocidad angular de los servomotores AX-12W. ω_2 representa la velocidad angular de la carcasa esférica (que se pretende calcular). Por último, D_1 y D_2 representa el diámetro del eje interno y el diámetro externo de la carcasa esférica, respectivamente.

Despejando ω_2 de (29),

$$\omega_2 = \omega_1 \frac{D_1}{D_2} \quad (30)$$

Reemplazando en (30),

$$\omega_2 = 470 \text{ RPM} \left(\frac{50 mm}{245 mm} \right) \quad (31)$$

$$\omega_2 = 95,9183_{\text{RPM}} \quad (32)$$

Conversión de RPM a $\frac{rad}{s}$,

$$\omega_2 = 95,9183 \left(\frac{2\pi rad}{1 rev} \right) \left(\frac{1 min}{60 s} \right) \quad (33)$$

$$\omega_2 = 10,0445 \frac{\text{rad}}{\text{s}} \quad (34)$$

A partir de (32) y (22), se puede determinar la velocidad de desplazamiento del robot esfera.

$$V_2 = \omega_2 \frac{D_2}{2} \quad (35)$$

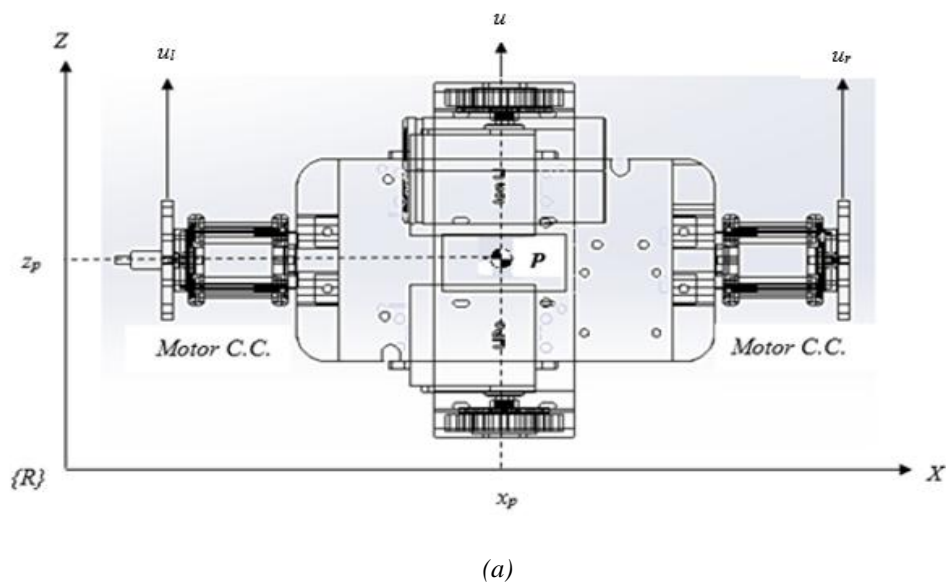
$$V_2 = 10,0445 \frac{\text{rad}}{\text{s}} \left(\frac{245 \text{ mm}}{2} \right) \quad (36)$$

$$V_2 = 1,23 \frac{\text{m}}{\text{s}} \quad (37)$$

4.4.3 Modelo Cinemático

En este tipo de modelos se considera una masa puntual, por lo que ésta no ejerce efecto alguno sobre la estructura, anulando las perturbaciones a ella asociadas (momentos de inercia y rozamientos).

La Figura 4.10a muestra la geometría de un robot móvil tipo unicycle, donde la posición del robot móvil la define el punto P, ubicado en el centro del eje que une los motores de conducción.



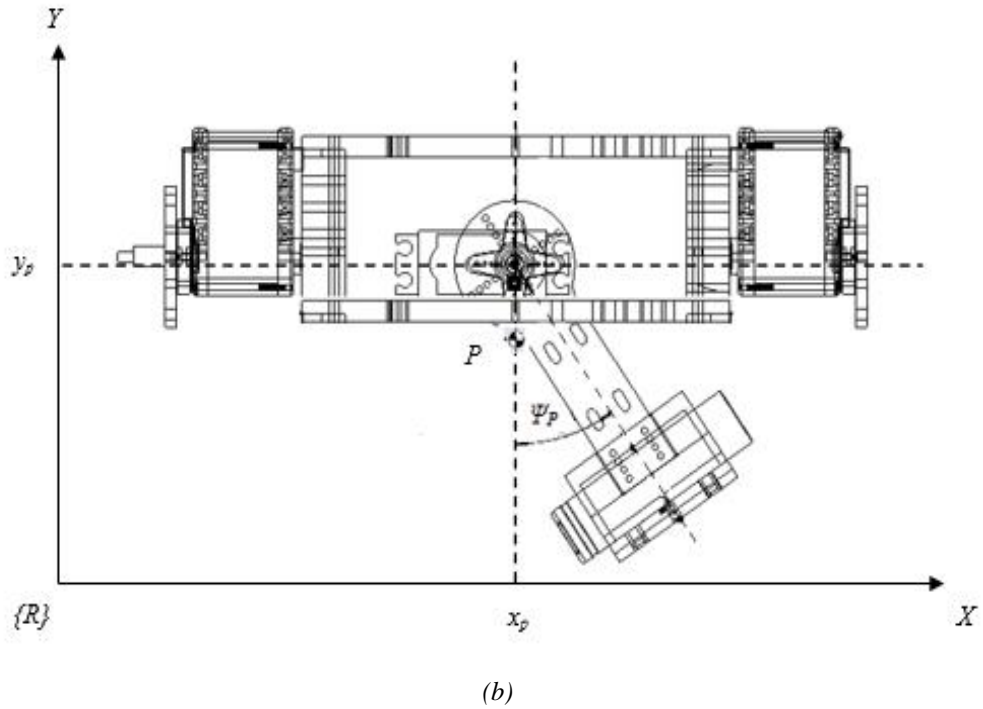


Figura 4.10: (a) Modelo de la estructura mecánica del robot móvil. (b) Orientación del robot móvil.

Elaborado por: investigador

Donde, $\{R\}$ representa el sistema de referencia de coordenadas globales, (x_p, y_p, z_p) que definen la posición del robot respecto a $\{R\}$ y ψ_p determina la orientación del robot respecto a $\{R\}$, tal como se muestra en la Figura 4.10b.

La velocidad lineal del robot móvil se determina promediando la velocidad del motor derecho y la velocidad del motor izquierdo de la siguiente manera:

$$u = \frac{u_r + u_l}{2} \quad (38)$$

Donde, u representa la velocidad lineal del robot móvil, u_r y u_l son las velocidades lineales de desplazamiento de la rueda derecha e izquierda, respectivamente.

Para este caso, la velocidad del motor izquierdo y derecho son iguales ($u_r = u_l$). Por lo tanto la velocidad lineal del robot se puede definir como:

$$u = u_r \quad (39)$$

Por otro lado, la velocidad angular del mismo se define de la siguiente manera:

$$\omega = \frac{\Delta\theta}{\Delta t} \quad (40)$$

Donde, ω corresponde a la velocidad angular del robot móvil, $\Delta\theta$ y Δt son el desplazamiento angular y el tiempo desde un punto inicial hasta un punto final, respectivamente.

La cinemática de un robot móvil (con el punto de control ubicado en el centro del eje que une los motores de conducción) puede ser modelada como sigue:

$$\begin{cases} \dot{x} = u \cos \psi \\ \dot{y} = u \sin \psi \\ \dot{\psi} = \omega \end{cases} \quad (41)$$

4.5 Consideraciones del Diseño Mecánico

4.5.2 Diseño de las piezas internas

En principio, para el diseño y fabricación de las piezas internas del prototipo, se consideró para este trabajo el basado en Masa Pendular como UID (por sus siglas en inglés, Unit Internal Driving), proporcionando dos grados de libertad al robot esfera. El primer grado de libertad se ve reflejado en la tracción de la esfera el cual es un movimiento de rotación de 360° con respecto al eje central fijo, es decir, involucra la creación de piezas soportes para los actuadores de tracción.

El segundo grado de libertad corresponde a la inclinación de la esfera cuyo propósito es darle dirección a la misma en torno a un diámetro perpendicular a dicho eje el cual varía desde $\pm 90^\circ$ en condiciones ideales, sin embargo en la práctica esta variación de ángulo se ve bastante reducida por ciertas limitaciones mecánicas. No obstante, con el bosquejo preliminar en 2D de una base general (ver Figura 4.11) se pudo prever los detalles pertinentes en cuanto a la distribución de los elementos, de tal forma que encajen de manera precisa y se mantengan fijas en todo momento mediante la sujeción de tuercas y pernos.

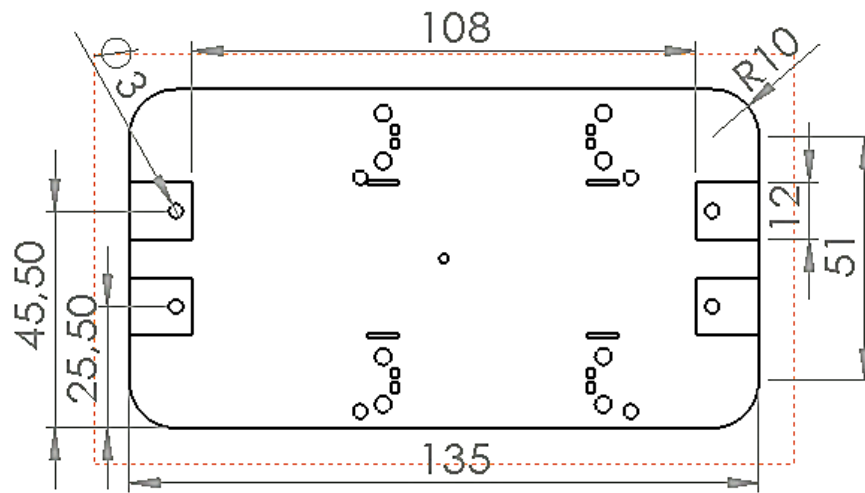


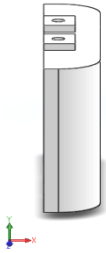
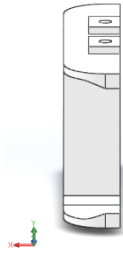
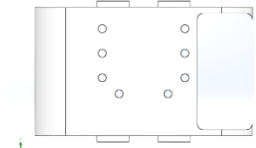
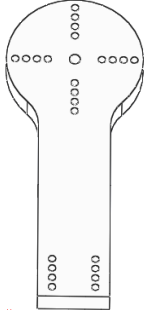
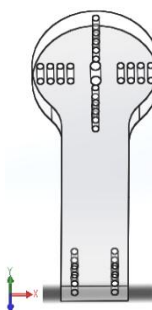

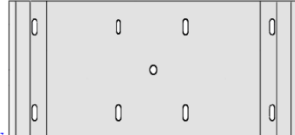
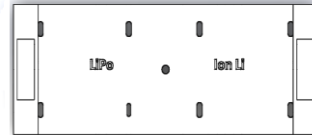
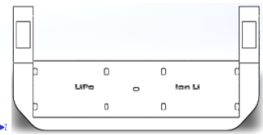
Figura 4.11: Vista en 2D de la base general con sus resectivas dimensiones.

Elaborado por: investigador

A continuación, se presenta (ver Tabla 4.7) el resto de las piezas que forman parte del mecanismo interno desde distintos puntos de vista. Claramente se puede apreciar ciertos detalles como: ranuras, espacios dedicados para las tarjetas de control, agujeros de distintas medidas para pernos, entre otras cosas.

Tabla 4.7: Vistas en 3D de las piezas que conforman el mecanismo interno.

Pieza	Vistas		
	Inferior	Superior	Lateral
Plataforma General			
Plataforma Secundaria			

Columna Derecha			
Brazo pendular			
Base Péndulo			

Elaborado por: Investigador

En la Figura 4.12, se muestra la estructura interna ensamblada

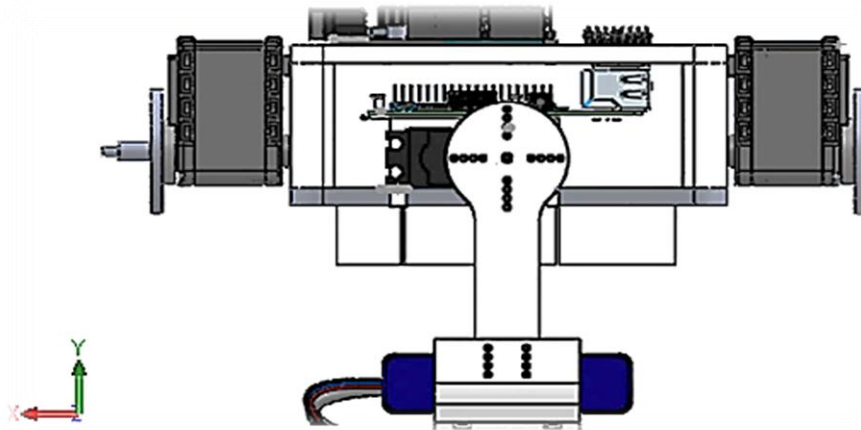


Figura 4.12: Estructura interna ensamblada.

Elaborado por: Investigador.

En cuanto a la carcasa esférica que albergará a todos los componentes electrónicos y demás elementos en su interior, se utiliza como cuerpo esférico una bola de color azul semi-transparente conocida como bola de entrenamiento para mascotas (Hamster Ball), cuyo diámetro exterior corresponde a 245 mm y diámetro interior a 241 mm dejando el espacio necesario para armar todo lo necesario mencionado anteriormente (ver Figura 4.13).



Figura 4.13: Bola de ejercicio para mascotas

Elaborado por: Investigador.

Se caracteriza también por estar fabricada de plástico flexible capaz de soportar choques siendo más duradero que el material acrílico, garantizando así el contacto de rodadura a la intemperie, pero con ciertas limitaciones debido a su diseño original de fábrica.

Uno de los materiales de construcción de las piezas internas, se le conoce como PLA (por sus siglas en inglés PolyLactic Acid o Ácido Poliláctico), es un polímero que se utiliza altamente en la industria de la impresión 3D siendo ésta una de sus aplicaciones, es un material biodegradable ya que es fabricado a partir de materias primas renovables (almidón de maíz), gracias a su carácter no tóxico este material es usado generalmente para la fabricación de objetos que entran en contacto con alimentos, pero se debe asegurar que la extrusora sea de acero inoxidable. Dentro de sus propiedades físicas se describen las siguientes:

- Tiene una densidad aproximada de 1.27 g/cc, lo que lo convierte en un material más pesado que el ABS (por sus siglas en inglés, Acrylonitrile Butadiene Styrene o Acrilonitrilo Butadieno Estireno).
- Tiene un módulo de elasticidad de 3.63 GPa, lo cual dificulta su deformación ante cargas y lo hace un material menos elástico.
- Temperatura de transición vítrea: 60-65°C.

Por otro lado, el plástico acrílico y la técnica de corte mediante láser facilitaron la labor de la creación del juego de piezas restantes. Dentro de las características técnicas del Acrílico podemos encontrar las siguientes:

- La luz transmitida a través de las láminas de Acrílico conserva el 92% de su intensidad inicial (en el rango de luz visible), por lo que se aprecia transparente.
- Dada la composición plástica del Acrílico, no se muestran cambios significativos en sus propiedades físicas o químicas después de exposiciones prolongadas a las condiciones climáticas.
- Dureza similar a la de metales no ferrosos como el cobre.
- Posee una resistencia de 0.2 a 0.5 lb*ft/in. En comparación, es 6 veces más resistente que el vidrio.

El tiempo de elaboración por pieza está ligado a sus dimensiones es decir, que cuanto mayor sea éste parámetro mayor será el tiempo fabricación y por ende su costo. No obstante, para la elaboración de este trabajo práctico resultó ser moderado y asequible, lo cual se puede observar más adelante en el análisis económico del proyecto.

4.6 Diseño Electrónico

El diseño electrónico hace referencia a los componentes electrónicos seleccionados (*detallado en el apartado 4.3.1*) y su forma de estar interconectados entre sí. En la Figura 4.14, se establece un esquema de interconexión en el cual es posible entender cómo interactúan dichos componentes electrónicos y su

respectiva forma de comunicarse cumpliendo de esta manera parte de los objetivos planteados.

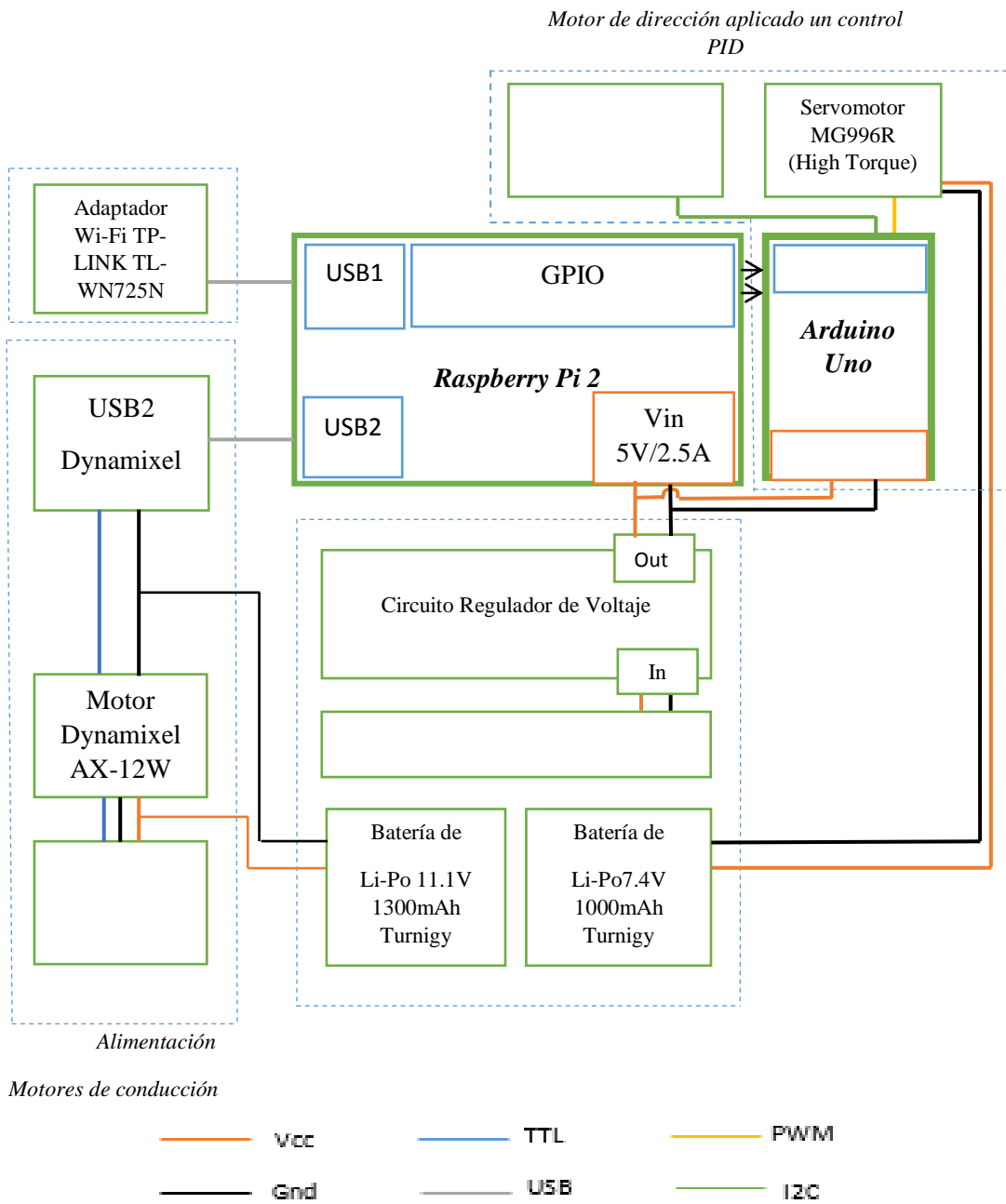


Figura 4.14: Esquema de interconexión eléctrica de los componentes.

Elaborado por: Investigador

4.7 Interfaz de operación hombre-máquina

La interfaz de usuario se desarrolló de tal manera, que el mismo pueda tener control sobre el prototipo con la mayor fluidez posible, ya sea desde un ordenador personal (joystick) o un dispositivo móvil (tecnología táctil) con la cual el usuario podrá tener una alternativa de mando a distancia y mejorar su experiencia al controlar el robot.

La interfaz de control, independientemente del enfoque que tenga. Tiene como finalidad servir de intermediario para la ejecución de los movimientos sencillos del robot, como serían la marcha hacia adelante y hacia atrás, así como los giros derecha-izquierda mediante la combinación de botones.

No obstante, es necesario cumplir con algunos requisitos previos, como son:

- Configurar el servidor Apache para Ubuntu MATE 16.04 o LAMP dicho de otra manera Linux, Apache, MySQL, PHP (siendo estas dos últimas opcionales).
- De igual manera, es necesario levantar el servidor FTP, el cual facilita el intercambio de archivos entre el lado del servidor y el lado del cliente.
- Instalar la plataforma Node.js.
- Y, por último, incluir el módulo Socket.io, así como paquetes extras.

Todos los detalles de instalación de cada uno de los requisitos se pueden observar más explícitamente en los Anexos B, C y D respectivamente.

4.7.1 Interfaz de mando

El desarrollo de esta primera interfaz constituirá en cierta parte la base para el desarrollo de la segunda interfaz debido a que, parte de una investigación para la inclusión de nuevas herramientas las cuales permitirán al usuario a cargo poder realizar dichos movimientos mencionados anteriormente.

Control de mando

La interfaz táctil se compone de tres archivos principales, los mismos que están alojados en un directorio propio denominado Robot_touch, cuya ruta corresponde a /home/ftpuser/Robot_touch, dentro del servidor.

Los archivos pertenecientes al directorio o carpeta Robot_touch se enlistan a continuación, y cuyo código de programación se puede observar en detalle en los anexos E, F y G respectivamente:

Robot_touch:

- **InterfazTactil.html**, en este archivo se puede encontrar todo lo concerniente a la representación gráfica en 2D de los botones a través de la inclusión del elemento Canvas y mediante los cuales el usuario podrá comandar. Además, integra todos los eventos enfocados con la parte táctil. Requiere de un complemento o librería extra denominada *jquery-1.11.1.js* para su correcto funcionamiento.
- **index.js**, este archivo es el encargado de establecer una comunicación entre la InterfazTactil.html con el servody.py cada vez que se ejecuta. De igual manera requiere de módulos extras, tales como:

node_modules

- ✓ express
 - ✓ socket.io
 - ✓ string
- **servody.py**, este archivo tiene como propósito la ejecución de los servomotores Dynamixel mediante la inclusión de la librería *lib_robotis.py*.

Cada uno de estos archivos se complementan entre sí, de tal forma que el usuario a cargo del manejo del robot esfera pueda hacerlo de manera intuitiva.

Para empezar a utilizar la interfaz táctil, es necesario correr el archivo *index.js*, de la siguiente manera: `$ sudo node index.js`.

De esta forma, se genera un puerto de comunicación mediante el cual el servidor *http* empieza a escuchar (Ej. <http://192.168.0.10:3000>). En la Figura 4.15, se presenta la interfaz táctil. Siendo 3000 el puerto por el cual se establece la comunicación bidireccional entre cliente y el servidor.



(a)



(b)

Figura 4.15: Captura de pantalla de la Interfaz táctil (a) Modo vertical (b) Modo horizontal.

Elaborado por: Investigador

Dicha interfaz se caracteriza por tener la capacidad de detectar el sentido de orientación del dispositivo móvil. Esto quiere decir, que las áreas de mando o botones adaptan su funcionalidad ya sea de forma vertical u horizontal.

Otras de las singularidades de esta interfaz, es que se ajusta a las dimensiones de pantalla de forma automática, así como también, se previno el zoom por defecto al realizar doble toque sobre la misma.

Distribución y funcionalidad de los botones en modo horizontal:

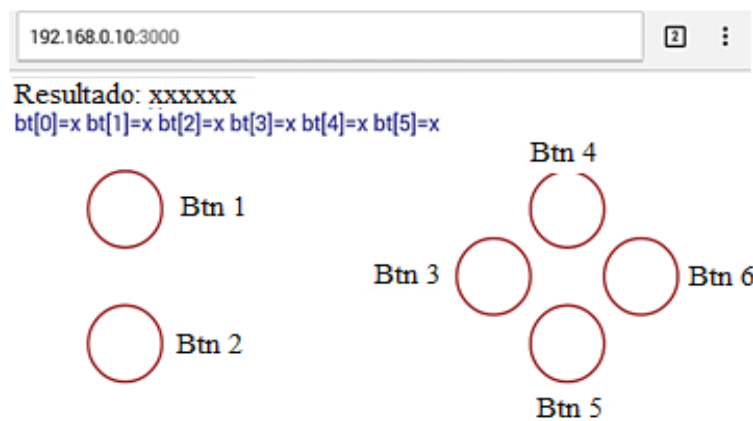


Tabla 4.8: Descripción de la funcionalidad de los botones del mando táctil.

	Funcionalidad
Btn 1	Permite el avance del robot esfera (dependiendo del punto de orientación)
Btn 2	Permite el regreso (reversa) del robot esfera (dependiendo del punto de orientación)
Btn 3	Desequilibra el centro de masa hacia la izquierda
Btn 4	Aumenta la velocidad de forma gradual de los motores de conducción en una escala de 1 a 10
Btn 5	Disminuye la velocidad de forma gradual de los motores de conducción en una escala de 1 a 10
Btn 6	Desequilibra el centro de masa hacia la derecha







Elaborado por: Investigador.

Adicionalmente a la interfaz de mando táctil, se consideró pertinente generar otro archivo.js con el nombre *start.js* en el directorio */home/ftpuser/Robot_Teclado2*, el cual permite hacer uso del teclado constituyéndose como una segunda opción de

manejo del prototipo con el cual el usuario al mando podrá mejorar su experiencia al tener control sobre el mismo.

Está desarrollada en base a la creación de la anterior interfaz. Es decir, que se aplicaron ciertas modificaciones en el código de programación para alcanzar este fin. En la Tabla 4.9, se muestra la funcionalidad de los botones que forman parte de dicho mando por teclado.

Tabla 4.9: Descripción de la funcionalidad de los botones del mando por teclado.

	Funcionalidad
	Permite el avance del robot esfera (dependiendo del punto de orientación)
	Permite el regreso (reversa) del robot esfera (dependiendo del punto de orientación)
	Desequilibra el centro de masa hacia la izquierda
	Aumenta la velocidad de forma gradual de los motores de conducción en una escala de 1 a 10
	Disminuye la velocidad de forma gradual de los motores de conducción en una escala de 1 a 10
	Desequilibra el centro de masa hacia la derecha

Elaborado por: Investigador.

4.8 Controlador PID

El control Proporcional Integral Derivativo (PID) es un mecanismo de control mediante realimentación negativa, el cual aplica una acción correctora al sistema para obtener el valor de consigna (Setpoint).

La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controles PID resultan más útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo[55].

Debido a que la Unidad de Manejo Interno del prototipo se presenta como un sistema simple, se plantea como una opción relevante la implementación de un controlador PID basado en las reglas de sintonía de Ziegler y Nichols, las cuales se analizan más adelante en este mismo apartado.

4.8.1 Estructura del controlador PID

Las tres componentes de un controlador PID son: la acción proporcional, acción Integral y la acción derivativa. A continuación, se presenta el diagrama de bloques con el que se representa este controlador (ver Figura 4.16).

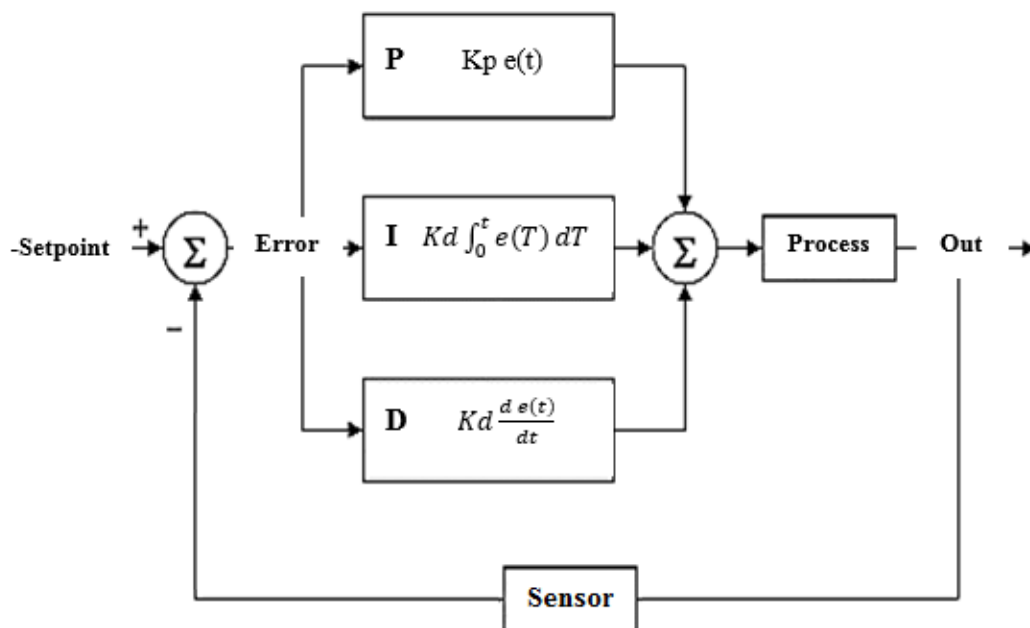


Figura 4.16: Esquema de control PID .

Elaborado por: Investigador

Al combinar estas tres acciones se tienen como resultado los siguientes tipos de control[56]:

- **P: acción de control proporcional**, da una salida del controlador que es proporcional al error, que descrita desde su función transferencia queda:

$$u(t) = Kp \cdot e(t) \quad (42)$$

$$C_p(s) = Kp \quad (43)$$

Donde, Kp es una ganancia proporcional ajustable.

- **I: acción de control integral**, da una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento.

$$u(t) = K_i \int_0^t e(t) dt \quad (44)$$

$$C_i(s) = \frac{K_i}{s} \quad (45)$$

La señal de control $u(t)$ tiene un valor diferente de cero cuando la señal de error $e(t)$ es cero. Por lo que se determina que dada una referencia constante, o perturbaciones, el error en régimen permanente es cero.

- **PI: acción de control proporcional-integral**, se define mediante [62]:

$$u(t) = Kp \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (46)$$

Donde, T_i se denomina tiempo integral y es quien ajusta la acción integral.

La función de transferencia resulta:

$$C_{PI}(s) = Kp \left(1 + \frac{1}{T_i s} \right) \quad (47)$$

- **PID: acción de control proporcional-integral-derivativa**, esta acción combinada reúne las ventajas de cada una de las tres acciones de control individuales. La ecuación de un controlador con esta acción combinada se obtiene mediante:

$$u(t) = Kp \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (48)$$

Cuya su función transferencia correspondiente es:

$$C_{PID}(s) = Kp \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (49)$$

4.8.2 Sintonización del controlador PID mediante el método de curva de reacción de Ziegler-Nichols

El proceso de seleccionar los parámetros del controlador que cumplan con las especificaciones de comportamiento dadas se conoce como sintonía del controlador [55]. Para el presente proyecto, se aplicarán las reglas sugeridas por

Ziegler y Nichols para la sintonización del mismo. Lo que significa, que se dará valores a K_p , T_i y T_d , en basea las respuestas escalón experimentales constituyéndose como el método a utilizar.

Método Basado en la Curva Reacción

Éste se constituye como el primer método sugerido por Ziegler y Nichols, el mismo que consiste en: Obtenerla respuesta de la planta a una entrada escalón unitario de manera experimental en lazo abierto cuya respuesta a su salida en sí, tendrá una forma de “S”. Los parámetros que se deben determinar son: el tiempo de retardo L y la constante de tiempo T . Y eventualmente, las constantes K_p , K_i y K_d de acuerdo con los parámetros de ajuste propuestos (ver Tabla 4.10).

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Tabla 4.10: Reglas de sintonización Ziegler-Nichols primer método.[55]

Cálculo de las constantes K_p , K_i y K_d

Para el presente proyecto se determinó que el tiempo de retardo L y la constante de tiempo T corresponden a :

$$L = 13 \text{ y } T = 24$$

Calculo de la constante K_p

$$K_p = 1.2 \frac{T}{L} \tag{50}$$

$$K_p = 1.2 \frac{24}{13} \tag{51}$$

$$K_p = 2,2223 \tag{52}$$

Calculo de la constante Ki

$$T_i = 2L \quad (53)$$

$$T_i = 26 \quad (54)$$

$$k_i = \frac{K_p}{T_i} \quad (55)$$

$$k_i = 0,0855 \quad (56)$$

Calculo de la constante Kd

$$T_d = 0,5L \quad (57)$$

$$T_d = 6,5 \quad (58)$$

$$k_d = K_p T_d \quad (59)$$

$$k_d = 14,445 \quad (60)$$

A partir de éste punto cabe señalar que las reglas de sintonía de Ziegler-Nichols han aportado un punto de partida para la sintonización fina. Por lo que, las constantes del controlador PID en este caso adquieren los siguientes valores de sintonía.

$$K_p = 2 \quad K_i = 1,2 \quad K_d = 18$$

Implementación del control PID

Para poder realizar dicha implementación, es necesario partir desde la toma y tratamiento de datos emitidos por el sensor inercial o MPU-6050. Para ello se tomó la librería *MPU6050.h* que se encuentra a disposición en la página web oficial de Arduino la misma que facilitó esta tarea. Sin embargo, los datos emitidos por dicho sensor se les conoce como datos en bruto los cuales pueden verse afectados por señales no deseadas presentes en el ambiente generando

errores en la lectura. Es por ello que, se vió la necesidad de implementar un filtro complementario (Filtro pasa altos y Filtro pasa bajos). Siendo éste el más fácil, preciso y aplicable mediante programación.

A continuación, se puede observar las líneas de código referente a la lectura del acelerómetro y giroscopio.

```
//Lee los valores emitidos por el Acelerómetro
Wire.beginTransmission(MPU);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU,6, true);
AcX=Wire.read() <<8|Wire.read();
AcY=Wire.read() <<8|Wire.read();
AcZ=Wire.read() <<8|Wire.read();

// Lee los valores emitidos por el Giroscopio
Wire.beginTransmission(MPU);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU,4, true);
GyX=Wire.read() <<8|Wire.read();
GyY=Wire.read() <<8|Wire.read();
```

Aplicación del filtro complementario después de la lectura de datos

```
Angle [0] = 0.98 *(Angle [0] +Gy [0] *0.010) + 0.02*Acc[0];
Angle [1] = 0.98 *(Angle [1] +Gy [1] *0.010) + 0.02*Acc[1];
t = Angle [0] + 90;
```

En la última línea de este extracto de código, los datos se presentan desde 0 a 180 grados para un mejor tratamiento en la implementación del control PID.

De igual manera, se encuentra a disposición la librería *PID_v1.h* desarrollada para la plataforma arduino lo cual se constituye como una ventaja para la implementación de dicho control.

La sintonización tiene como parámetros iniciales las tres acciones anteriormente calculadas, las cuales se declaran de la siguiente manera:

```
float Kp=2;
float Ki=1.3;
float Kd=28;
double Setpoint, Input, Output;
PID myPID (&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

Los parámetros Input, Output y Setpoint, corresponden a la variable controlada, variable manipulada y valor de consigna o valor a alcanzar, respectivamente.

A más de ello, es necesario definir el tiempo de muestro cuyo valor por defecto es 200 ms. Para aplicaciones de robótica éste valor puede variar.

```
myPID.SetSampleTime(sampleRate);
```

Por último, a una frecuencia especificada por SetSampleTime se calculará una nueva salida mediante la función compute(), la cual contiene todo el algoritmo PID.

```
myPID.Compute();
```

En el anexo H se puede observar en detalle el código de programación con respecto a la implementación del controlador PID.

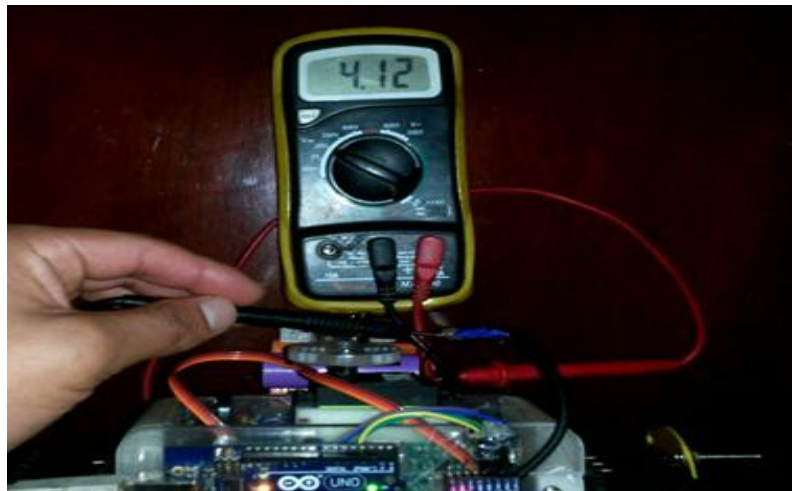
4.9 Pruebas de Funcionamiento

A continuación, se detallan las pruebas realizadas del prototipo esférico una vez ensamblada la estructura interna del mismo, así como sus componentes electrónicos relacionados mediante algoritmos de control y su interacción con la interfaz web.

Para comprobar el correcto funcionamiento del sistema de control se debe realizar pruebas que permitan asegurar las condiciones necesarias para el desplazamiento del mismo.

Pruebas de alimentación

En la Figura 4.17 se muestra la etapa de alimentación, funcionando y suministrando voltaje directo de 4.7VCD para la tarjeta de control, así como también, se muestra la medición del voltaje de entrada correspondiente a 11.1 VDC en los servos motores Dynamixel.



(a)



(b)

Figura 4.17: Medición del voltaje de entrada (a)Raspberry Pi(b)Servomotor MG996R.

Elaborado por: Investigador.

Pruebas de comunicación entre los servomotores Dynamixel y el ordenador mediante el uso de RoboPlus-Wizard

Al trabajar con este tipo de servomotores se puede establecer el tipo de rotación que mejor convenga (todo dependerá de la aplicabilidad), pues permite escoger entre una rotación o giro continuo y una rotación controlada por grados dentro de un rango que va desde los 0° hasta los 300°.

Para cumplir con el propósito de esta investigación es necesario que los servomotores se comporten como un motor DC o lo que es igual una rotación continua en ambos sentidos. Con la ayuda de una herramienta de configuración y testeo denominado RoboPlus se puede realizar esta tarea, tal como se puede apreciar en la Figura 4.18.

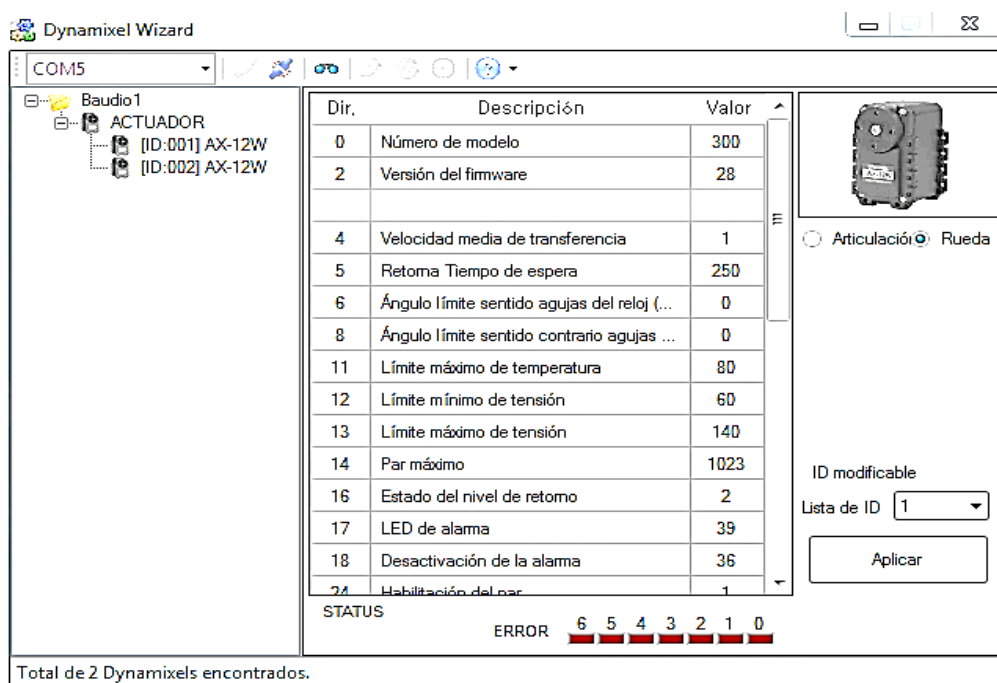


Figura 4.18: Reconocimiento y manejo de los dos servomotores conectados en cadena.

Elaborado por: Investigador.

Adicionalmente, es posible corroborar que el voltaje de entrada que alimentan a estos servomotores es el indicado, tal como se presenta en la Figura 4.19.

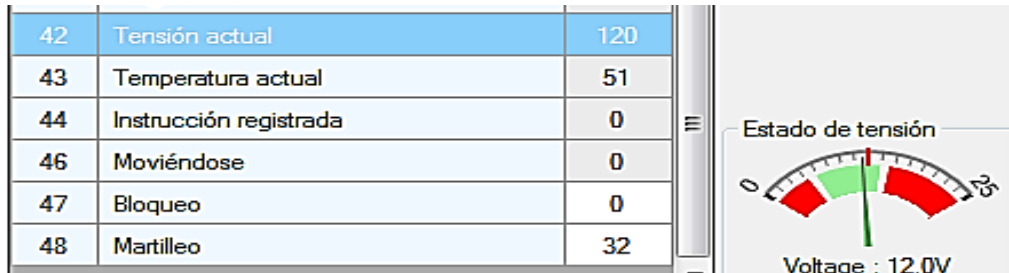


Figura 4.19: Medición de voltaje de entrada a través de Roboplus.

Elaborado por: Investigador.

Pruebas de funcionamiento de la comunicación inalámbrica a través de WiFi

Para esta prueba es necesario que todos los dispositivos estén conectados previamente a un mismo punto de acceso inalámbrico.

Dentro de las características mencionadas en apartados anteriores acerca de la tarjeta de control empleada en este proyecto, indica que no posee un chip de comunicación inalámbrica integrado como WiFi. Sin embargo, al realizar pruebas con el adaptador WiFi TP-LINK TL-WN725N se garantiza que la comunicación sea fluida al entablar un acceso remoto como es el caso de SSH o Telnet.

Pruebas de ejecución de los servomotores de la interfaz de mando

Para poner en marcha este tipo de prueba, inicialmente, se debe correr el servidor web o lo que es lo mismo ejecutar el archivo *index.js* o *start.js* tal como se indicó en el apartado 4.7 de este mismo capítulo.

Las pruebas demostraron tal y como se intuía, que al correr los dos archivos a la vez entran en conflicto de reconocimiento de datos enviados desde la terminal de mando imposibilitando el manejo del prototipo. Sin embargo, los resultados son satisfactorios al realizar pruebas de manera individual con ligero retardo de activación de aproximadamente 1000 mseg, debido a que se toma éste tiempo en ejecutarse el archivo *servosdy.py* encargado de controlar los servomotores

Dynamixel. En la Figura 4.20, se presenta la ejecución del servidor (index.js) y simultáneamente la respuesta en consola del archivo generado en Python.

```
user@user-desktop:/home/ftpuser/Robot_touch$ sudo node index.js
[sudo] password for user:
Escuchando en *:3000
datos: xxxxxx
js: Apagado
datos: xxxxex
js: Reversa
py: Reversa

datos: xxxxxx
js: Apagado
py: Apagado

datos: xxxdxx
js: Avanza
py: Avanza
```

Figura 4.20: Muestra en consola de la ejecución de los servomotores Dynamixel.

Elaborado por: Investigador.

Pruebas de funcionamiento del controlador PID implementado

En base a la obtención de las constantes de sintonía mencionadas en el paratado 4.8, dichas acciones que involucra este de controlador se ajustan de tal manera que alcanzan el valor de consigna o SetPoint el cual esta establecido en un valor fijo, teniendo como resultado la estabilidad del prototipo como reacción a una fuerza externa que trate de sacarlo de su estado de reposo. Esta acción de corrección se la puede observar en la siguiente grafica (ver Figura 4.21).

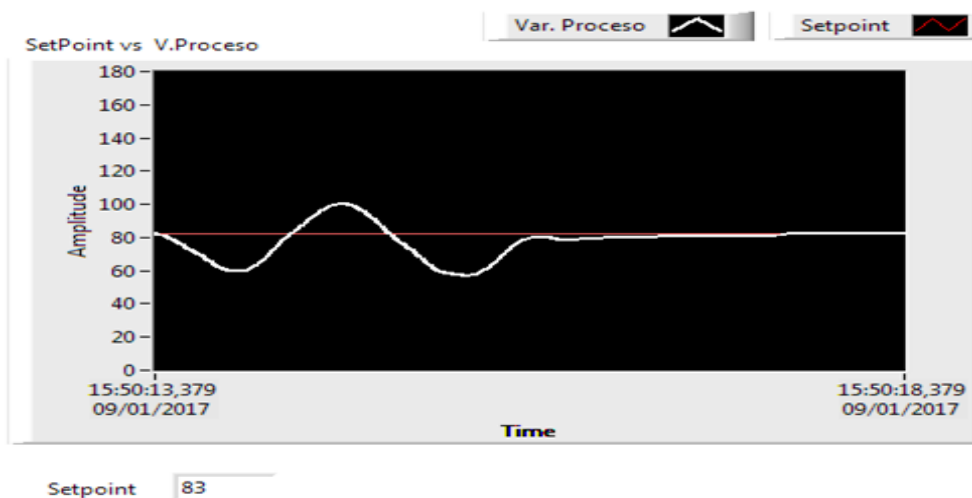


Figura 4.21: Acción de corrección del controlador PID con SetPoint fijo.

Elaborado por: investigador

A continuación, se presenta la gráfica correspondiente a la señal de proceso escala de 0 a 100 %. (ver Figura 4.22)

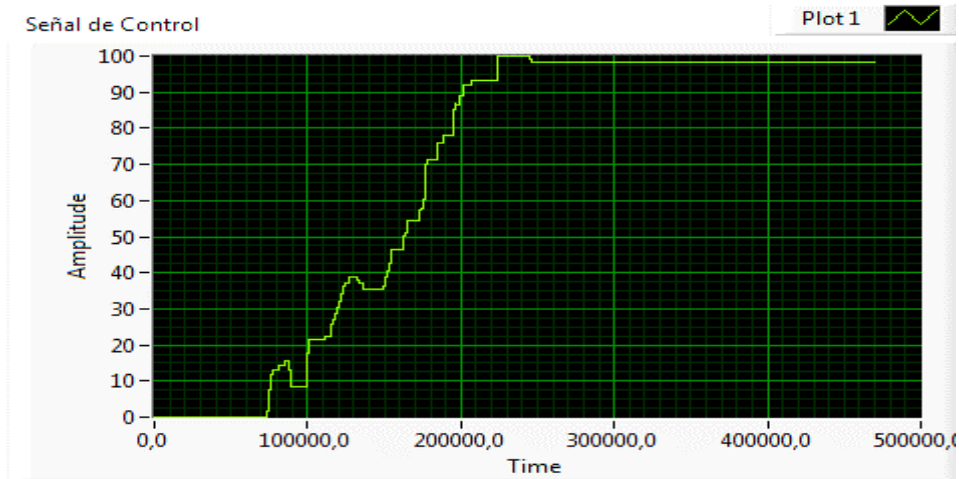


Figura 4.22: Señal de control o variable controlada.

Elaborado por: investigador

4.9 Análisis económico del proyecto

El costo del proyecto abarca la implementación total del prototipo robot esfera, es decir el diseño, construcción y finalmente las pruebas de funcionamiento, ha sido financiado por el investigador como se muestra en la tabla 4.11.

Tabla 4.11: Costo del proyecto

Ítem	Descripción	Unidad	Cantidad	Valor Unitario	Valor Total (Dólares)
Costo del Hardware					
1	Tarjeta de control Raspberry Pi 2 model B	c/u	1	75,00	\$75,00
2	Tarjeta de control arduino uno	c/u	1	15,00	\$15,00
3	Servomotor Dynamixel AX 12W	c/u	3	45,00	\$135,00
4	Paquete de cables marca Dynamixel	c/u	1	14,00	\$14,00
5	Convertor USB2Dynamixel	c/u	1	50,00	\$50,00

6	Paquete de tuercas y pernos Dynamixel	c/u	1	24,00	\$24,00
7	Llantas omnidireccionales	c/u	3	8,00	\$24,00
8	Adaptador WiFi marca TP Link	c/u	1	13,25	\$13,25
9	Baterias LiPo 11,1V a 1300 mAh marca Turnigy	c/u	2	22,50	\$45,00
10	Bateria LiPo 7,4V a 1000mAh	c/u	1	8,50	\$8,50
11	Cargador portátil incluido baterías Ion-Litio	c/u	1	12,00	\$12,00
12	Cable HDMI 1metro	c/u	1	16,25	\$16,25
13	Convertor HDMI a VGA	c/u	1	14,30	\$14,30
14	Mini cables USB para alimentación	c/u	2	2,25	\$4,50
15	Bola de entrenamiento para mascotas	c/u	1	16,00	\$16,00
Costo de fabricación por piezas					
16	Plataforma general acrilico	c/u	1	10,00	\$10,00
17	Plataforma secundaria acrilico	c/u	1	10,00	\$10,00
18	Brazo pendular acrilico	c/u	2	5,00	\$10,00
19	Columna izquierda	c/u	1	3,00	\$25,00
20	Columna derecha	c/u	2	3,00	\$10,00
21	Base pendular	c/u	1	5,50	\$5,50
				TOTAL	\$537.30

Elaborado por: Investigador

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

A continuación, se presentan las conclusiones que se obtuvieron en el diseño y construcción del prototipo robot esfera.

- El diseño esférico del prototipo resultó ser capaz de moverse por ambientes que presenten cierto grado de complejidad, demostrando su capacidad de recuperación gracias a su principio de locomoción basado en péndulo siempre y cuando no se encuentre atrapado. En cuanto a la estrategia de control empleada, hasta el punto en el que se ha desarrollado, ha demostrado ser útil, mejorando el comportamiento del mismo en caso de que experimente algún tipo de movimiento brusco externo.
- El control de operación inalámbrico del prototipo esfera, es más sensible debido a que la aplicación es en tiempo real por la inclusión del WebSocket. Sin embargo, a partir de este punto los actuadores toman un considerable tiempo de activación el mismo que se determinó de manera experimental y que bordeda los 70 a 80 mseg, generando un cierto grado de complejidad en cuanto a maniobrabilidad. Es por ello que la incorporación del control de velocidad que ayuda relativamente al manejo del mismo.
- La tecnología de comunicación inalámbrica está dictaminado por el estándar IEEE 802.11, que es un sistema ampliamente utilizado en la actualidad. Por lo tanto, es fácil de implementar en sistemas digitales con una computadora o dispositivos móviles. Permitiendo una conectividad a larga distancia mejorando en espacios abiertos, constituyéndose como el más adecuado para el propósito de esta investigación.

5.2 Recomendaciones

Se recopilan algunos datos como recomendaciones de los conocimientos adquiridos para que en trabajos futuros se las tenga en cuenta.

- Al tener una comunicación con niveles TTL, se recomienda que la longitud máxima del cable que interconecta los servomotores Dynamixel, no exceda los 50 cm de longitud entre motor y motor, debido a que se pierde la capacidad de control de los mismos a pesar que de forma teórica se estima que se puede establecer una comunicación hasta los 5 metros de distancia. Por tal razón, es aconsejable hacer uso de los cables cuyas longitudes vienen dadas de forma estándar por la misma marca.
- Es recomendable que el voltaje y corriente de operación de la Tarjeta de Control Raspberry pi, no deben estar por debajo de los valores mínimos correspondientes a 4,7V y una corriente de 1.5A, debido a que presenta problemas como: reinicio del sistema de manera inesperada, fallas en las comunicaciones inalámbricas, desconocimiento de periféricos conectados en los puertos USB, entre otros. Por este motivo se aconseja vigilar la actividad de los leds de estado propios de la tarjeta.
Por otro lado, sería pertinente la búsqueda y la utilización de tarjetas de control con enfoque a la robótica que integren sensores y protocolos de comunicación inalámbrica.
- Seguir en las investigaciones de robots con estructura esférica ya que es posible explotar aun más su estructura creando interacciones particulares de dinámica y control más amigables y robustos que ayudarán a establecer una relación más cercana entre los robots y los usuarios. Tales son sus aplicaciones también, que pueden variar desde objetivos muchos más ambiciosos con fines científicos hasta muchos más simples como domésticos y el entretenimiento.

BIBLIOGRAFIA

- [1] T. Ylikorpi and J. Suomela, "Ball-shaped robots," *Climbing Walk. Robot. Towar. New Appl. Vienna, Austria Itech*, pp. 235–256, Oct. 2007.
- [2] J. Maturana, "ROSPHERE, robot español de pura cepa: agricultor con forma de esfera." [Online]. Available: <https://www.xataka.com/robotica-e-ia/rosphere-robot-espanol-de-pura-cepa-agricultor-con-forma-de-esfera>. [Accessed: 16-Feb-2017].
- [3] Metro World News, "FOTOS: 8 geniales juguetes tecnológicos para niños | Publimetro Chile." [Online]. Available: <https://www.publimetro.cl/cl/teknik/2015/07/27/fotos-8-geniales-juguetes-tecnologicos-ninos.html>. [Accessed: 17-May-2016].
- [4] P. J. Cañada Betancourt, Ramiro Sebastian; Nazate Burgos, "Diseño y construcción de un robot móvil esférico con visión en tiempo real," p. 157, 2012.
- [5] V. Crossley, "A literature review on the design of spherical rolling robots," *Pittsburgh, PA*, pp. 1–6, 2006.
- [6] X. Niu, A. P. Suherlan, G. S. Soh, S. Foong, K. Wood, and K. Otto, "Mechanical development and control of a miniature nonholonomic spherical rolling robot," *2014 13th Int. Conf. Control Autom. Robot. Vision, ICARCV 2014*, pp. 1923–1928, 1997.
- [7] J. D. Hernández, "ROSPHERE: Diseño, Construcción y Aplicación de una Esfera Robótica," p. 103, 2012.
- [8] M. Pereira, "La fundanetación teórica de la educación para la carrera," vol. 507, pp. 309–338, 1995.
- [9] P. Edmundo, "Introducción a la Robótica," pp. 1–8, 2014.
- [10] K. Vélez, "Universidad de guayaquil," *Tesis*, no. ESTUDIO E IMPLEMENTACIÓN DE ESTRATEGIAS DE DESTREZAS Y VELOCIDAD APLICADO A LA ROBÓTICA, p. 121, 2014.
- [11] J. R. Otero, P. Paparel, D. Atreya, and K. Touijer, "Antecedentes ,desarrollo y situación actual de la robótica en la cirugía," *Antecedentes, Desarro. Y Situación Actual La Robótica En Cirugía.*, vol. 4, pp. 335–341, 2007.
- [12] V. Kaznov and M. Seeman, "Outdoor navigation with a spherical amphibious robot," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 5113–5118, 2010.
- [13] S. Bhattacharya and S. K. Agrawal, "Design, experiments and motion planning of a spherical rolling robot," *Robot. Autom. 2000. Proceedings. ICRA'00. IEEE Int. Conf.*, vol. 2, no. April, pp. 1207–1212, 2000.
- [14] S. Sang, J. Zhao, H. Wu, S. Chen, and Q. An, "Modeling and simulation of a spherical mobile robot," *Comput. Sci. Inf. Syst.*, vol. 7, no. 1, pp. 51–62, 2010.

- [15] A. Abadía, "Introducción a las redes WiFi," pp. 1–36, 2010.
- [16] V. K. Varma, "Wireless Fidelity—WiFi," *IEEE Emerg. Technol.*, pp. 1–2, 2012.
- [17] R. Gonzalez Carvajal, V. Perez Mira, D. Daza Rebollo, and J. Carrillo Castrillo, "Tecnologías inalámbricas," *Segur. y Medio Ambient.*, pp. 10–21, 2012.
- [18] P. Belmonte, "Sobre el sistema WIMAX | Ecologistas en Acción." [Online]. Available: <http://www.ecologistasenaccion.org/article11321.html>. [Accessed: 10-Jun-2016].
- [19] E. De Ingeniería *et al.*, "Estudio del estándar IEEE 802.15.4 'Zigbee' para comunicaciones inalámbricas de área personal de bajo consumo de energía y su comparación con el estándar IEEE 802.15.1 'Bluetooth,'" 2006.
- [20] S. A. Gonzalez Vergara and J. E. Pérez Carmona, "Tecnología Bluetooth," p. 86, 2008.
- [21] J. Dignani, "Análisis del protocolo zigbee," p. 42, 2011.
- [22] Ministerio de Educación Cultura y Deporte, "Recursos - educaLAB." [Online]. Available: http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena12/4quincena12_contenidos_1b.htm. [Accessed: 11-Jul-2016].
- [23] Arduino, "Arduino UNO & Genuino UNO." [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Accessed: 21-Oct-2016].
- [24] E. Crespo, "Qué es Arduino y el Hardware Libre | Aprendiendo Arduino." [Online]. Available: <https://aprendiendoarduino.wordpress.com/2015/03/22/que-es-el-hardware-libre/>. [Accessed: 22-Oct-2016].
- [25] Isaac, "Comparativa de todas las placas Arduino," 2014. [Online]. Available: <http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>. [Accessed: 22-Oct-2016].
- [26] Terasic Technologies, *DEO-Nano User Manual*. 2013.
- [27] M. Bejarano, "¿Que es el Rasperry Pi? - Frambuesa Pi." [Online]. Available: <http://www.frambuesapi.co/que-es-el-rasperry-pi/#comment-1851>. [Accessed: 12-Nov-2016].
- [28] Raspberry Pi Foundation, "Raspberry Pi 2 Model B." [Online]. Available: <https://www.raspberrypi.org/products/rasperry-pi-2-model-b/>. [Accessed: 12-Nov-2016].
- [29] C. García and J. López, "Integración de Redes Telemáticas," *UOC Univ. Oberta Catalunya*, p. 84, 2014.
- [30] F. J. Jimenez Montero, "Sensores y Actuadores," *Publ. En Internet*, pp. 1–20, 2007.
- [31] I. Quintanilla, "Construcción de un prototipo de robot con un microcontrolador,

- sensores de luz y servomotores que realice movimientos autónomos,” pp. 1–167, 2012.
- [32] J. A. CORRALES RAMÓN and F. A. CANDELAS HERÍAS, “Servomotores,” 2007.
- [33] Mechatronics, “Tutorial uso de servomotores con arduino.” [Online]. Available: http://www.naylampmechatronics.com/blog/33_Tutorial-uso-de-servomotores-con-arduino-.html. [Accessed: 02-Nov-2016].
- [34] Á. Araya del Río, “Una Introducción a los Robots Móviles. Il Bambino - PDF.” [Online]. Available: <http://docplayer.es/7826797-Una-introduccion-a-los-robots-moviles-il-bambino-piu-avanti-hs-gmail-com.html>. [Accessed: 02-Nov-2016].
- [35] V. D. Zambrano Pérez, “Implementación de algoritmos de determinación de rutas para el robotino® de festo,” 2015.
- [36] R. Salavarría, Q. L. Imbaquingo, and Q. I. Ronald, “Robot Omnidireccional Controlado con NIOS II,” no. 1.
- [37] G. Andaluz, “Modelación, identificación y control de robots móviles,” *Vasa*, p. 209, 2011.
- [38] Superdroid, “Omni Wheeled Robots | Robots with Omni Wheels | Mecanum Wheels.” [Online]. Available: <http://www.superdroidrobots.com/shop/custom.aspx/vectoring-robots/44/>. [Accessed: 12-Nov-2016].
- [39] M. Guanoluisa and D. Pazmiño, “Cargador de baterías de plomo-ácido de 48 voltios,” pp. 1–126, 2008.
- [40] S. Martínez and R. Sisto, “Control y Comportamiento de Robots Omnidireccionales Posicionamiento y Sensor Data Fusion,” pp. 1–21, 2009.
- [41] Ward, “Robot Platform - STL,STEP / IGES,SOLIDWORKS - 3D CAD model - GrabCAD.” [Online]. Available: <https://grabcad.com/library/robot-platform>. [Accessed: 21-Nov-2016].
- [42] S. Sadiq, “4WD Robot Omni Wheel at 45 Degree - STEP / IGES - 3D CAD model - GrabCAD.” [Online]. Available: <https://grabcad.com/library/4wd-robot-omni-wheel-at-45-degree-1>. [Accessed: 21-Nov-2016].
- [43] F. Silva, “Implementación de un servidor web y dns para la empresa gigowireless para proveer el servicio de internet a la ciudad de Quito,” pp. 1–167, 2008.
- [44] A. De la Torre, “Lenguajes del lado servidor o cliente.” [Online]. Available: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html. [Accessed: 20-Nov-2016].
- [45] D. Pérez, “Los diferentes lenguajes de programación para la web.” [Online]. Available: <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/>. [Accessed: 20-Nov-2016].
- [46] HetPro, “HetPro | Acelerómetro Giroscopio MPU-6050 6DOF.” [Online].

Available: <https://hetpro-store.com/acelerometro-giroscopio-mpu-6050-6dof/>. [Accessed: 21-Nov-2016].

- [47] J. García and A. Zurita, "Diseño de un robot móvil interactivo y modular," p. 212, 2016.
- [48] Robotics, "Trossen Robotics Dynamixel Guide." [Online]. Available: <http://learn.trossenrobotics.com/projects/159-trossen-robotics-dynamixel-guide.html>. [Accessed: 23-Nov-2016].
- [49] Robotis, "USB2Dynamixel." [Online]. Available: http://support.robotis.com/en/product/auxdevice/interface/usb2dxl_manual.htm. [Accessed: 22-Nov-2016].
- [50] Isaac, "Las mejores distribuciones para Raspberry Pi," 2015. [Online]. Available: <http://comohacer.eu/distribuciones-raspberry-pi/>. [Accessed: 22-Oct-2017].
- [51] R. Cirino, R. Murillo, and M. Yapur, "Implementación de un electrocardiógrafo mostrando la señal cardiaca en una computadora gobernada por software libre (Python)," pp. 1–7, 2011.
- [52] R. Hibbeler, *Ingeniería Mecánica: Estática*. México: Pearson Educación, 2010.
- [53] P. Vallejo and J. Zambrano, *Física Vectorial 1*. Rodin, 2010.
- [54] Cidead, "Movimiento circular uniforme," *Articulo*, pp. 1–12, 2012.
- [55] K. Ogata, *Ingeniería de Control Moderna*, vol. 53, no. 9. 2013.
- [56] V. Mazzone, "Controladores PID," p. 12, 2012.
- [57] Raspbeery Pi, "GPIO: Models A+, B+, Raspberry Pi 2 B and Raspberry Pi 3 B - Raspberry Pi Documentation." [Online]. Available: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>. [Accessed: 02-Oct-2016].

ANEXOS

Anexo A

Guía de instalación del sistema operativo en Raspberry Pi

Paso 1

Descargar la imagen .ISO disponible de forma gratuita en la página oficial de la Fundación Raspberry Pi, en el siguiente enlace <https://www.raspberrypi.org/downloads/raspbian/>, del cual se aconseja obtenerlo en archivo .ZIP

Paso 2

Preparar la tarjeta SD (8 Gb ó 16 Gb), con ayuda de un adaptador y un ordenador, procedemos a formatear la misma (se asume que se está trabajando en Windows).

Paso 3

Localizar el directorio del archivo.ZIP y extraerlo directamente en la tarjeta SD, volviéndola booteable de forma automática y lista para ser instalada en la Raspberry Pi.

Paso 4

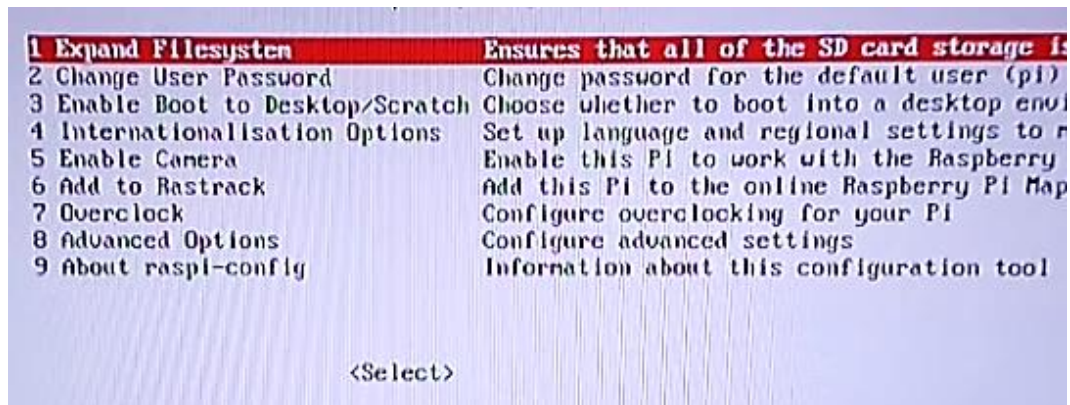
Insertar la tarjeta SD en la ranura de específica de la Raspberry Pi, y automáticamente iniciará la instalación del Sistema Operativo.

Es importante conocer que antes de conectar la alimentación respectiva de la tarjeta controladora, se deben conectar todos los periféricos tales como: mouse, teclado, pantalla, entre otros.

Paso 5

Realizar una configuración previa.

En una Pantalla Terminal, escribir el siguiente comando `$ sudo raspi-config`



A-1: Captura de pantalla de configuración inicial

Interpretación de las opciones desplegadas:

[1]Expand Filesystem (Expansión de los archivos del Sistema):esta opción permite la expansión de nuestra instalación para completar el resto de la tarjeta SD, proporcionando más espacio para el uso de archivos. Esta opción requiere de un reinicio para su habilitación.

[2]Enable Boot to Desktop/Scratch (Habilitar el arranque al Escritorio/Scratch):esta opción cambia el arranque de preferencia es decir a líneas comandos, al escritorio o directamente a Scratch.

[3]Change User Password (Cambiar Contraseña de Usuario):el usuario por defecto en raspbian es *pi* y cuya contraseña es *raspberrypi*. que por seguridad es recomendable cambiar estos valores ingresando en esta opción.

[4] Internationalisation Options (Opciones Internacionales): al ingresar en esta opción se despliega un submenú con las siguientes opciones

- Change locale (cambiar localidad)
- Change Timezone (cambiar zona horaria)
- Change keyboard layout (cambiar formato de teclado)

[5] **Enable camera (habilitar cámara):** para utilizar el módulo de cámara raspberry pi, se debe habilitar esta opción.

[6] **Add to Rastrack (Añadir Rastrack):** Rastrack es un usuario aportado por Google Map a los usuarios Pique hayan ingresado su localidad; muestra un mapa del lugar de los usuarios Pi, para así ser conocidos alrededor del mundo.

[7] **Overclock:** es posible realizar un overclock al CPU de la raspberry pi. Por defecto está configurada a 700 MHz, pero puede ser configurada hasta los 1000MHz. Sin embargo, esta opción puede causar inestabilidad, así como reducir el tiempo de vida de la raspberry.

[8] **Advanced Options (Opciones Avanzadas):** de igual manera se despliega un submenú con las siguientes opciones, las mismas que deberán ser activadas según sea necesario:

- **Overscan:** ajusta la imagen a la pantalla de trabajo.
- **Hostname:** asigna un nombre visible a la raspberry en una red.
- **Memory Split:** cambia el montaje de la memoria haciéndola disponible para la GPU.
- **SSH:** Habilita o deshabilita el acceso remoto a nuestra raspberry pi a través de SSH.
- **Device Tree:** Habilita o deshabilita el uso de Device Tree.
Para más información revisar la respectiva documentación en el siguiente [enlace web](https://www.raspberrypi.org/documentation/configuration/device-tree.md)
<https://www.raspberrypi.org/documentation/configuration/device-tree.md>
- **SPI-I2C-Serial-Audio**
- **Update**

[9] **About Raspi-Config**

Anexo B

Instalación del servidor LAMP en la Raspberry Pi 2

Para poder levantar el siguiente grupo de softwares de código abierto denominado LAMP, se toma como información de soporte, la que se muestra en el siguiente enlace web <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-debian>

Es recomendable actualizar los repositorios con el siguiente comando en consola:

```
$ sudo apt-get update
```

B-1 Instalación del Servidor Apache

Apache es un software de código abierto el cual puede correr sobre el 50% de los servidores del mundo

Instalar apache:

```
$ sudo apt-get install apache2
```

Comprobar la instalación ingresando la dirección ip (estática o dinámica) (Ej. <http://192.168.0.15>). La página desplegará el siguiente mensaje “It Works”.

Para conocer la dirección ip de nuestro dispositivo, se puede escribir el siguiente comando en consola `$ ip add`.

B-2 Instalación MySQL

MySQL es un sistema de gestión de bases de datos ampliamente utilizado para organizar y recuperar los datos.

Para instalar MySQL, abrir una terminal y teclear el siguiente comando:

```
$ sudo apt-get install mysql-server
```

Es necesario aplicar cierta configuración

```
$ mysql_secure_installation
```

Ingresar la contraseña de usuario Root

```
Enter current password for root (enter for none):
```

En las siguientes opciones de configuración se recomienda aceptar todas ellas, tecleando “Yes”.

```
Remove anonymous users? [Y/n] y
    ... Success!
Disallow root login remotely? [Y/n] y
    ... Success!
Remove test database and access to it? [Y/n] y
- Dropping test database...
    ... Success!
- Removing privileges on test database...
    ... Success!
Reload privilege tables now? [Y/n] y
    ... Success!

Cleaning up...
```

B-3 Instalación de PHP

PHP es un lenguaje web de código abierto que es ampliamente usado para la creación de páginas web dinámicas

De igual manera que los anteriores procesos de instalación, abrir una ventana terminal y escribir la siguiente línea de comando:

```
$ sudo apt-get php5 php-pear php5-mysql
```

Aceptar la instalación de PHP, y esperar hasta su finalización.

Es necesario reiniciar el servidor Apache:

```
$ sudo service apache2 restart
```

Para revisar que PHP se ha instalado correctamente, se recomienda crear el siguiente archivo de prueba denominado info.php en el directorio /www.

```
$ sudo nano /var/www/info.php
```

Salir y guardar los cambios con CTRL+X, Y, Enter.

Abrir el navegador de preferencia y escribir lo siguiente (*Ej. http://192.168.0.10/info.php*), si todo está correctamente instalado, se desplegará la información por defecto de la versión PHP instalada.

Anexo C

Instalación del servidor FTP

El servidor FTP (por sus siglas en inglés File Transfer Protocol), es una forma de transferir archivos entre ordenadores en internet. Es útil y cómodo en cuanto a la carga o descarga de archivos desde o hacia un sitio de forma rápida.

Instalar FTP:

```
# sudo apt-get install vsftpd
```

Parámetros de configuración:

Abrir el archivo de configuración localizado en la siguiente ruta:

```
# sudo nano /etc/vsftpd.conf
```

Y, aplicar las modificaciones pertinentes:

```
anonymous_enable=NO
```

```
local_enable=YES
```

```
write_enable=YES
```

```
chroot_local_user=YES
```

Salir y guardar los cambios con CTRL+X, Y, Enter

Crear un usuario FTP:

```
$ sudo useradd <-USERNAME->
```

```
$ sudo passwd <-USERPASS->
```

```
$ sudo groupadd <-GROUPNAME->
```

```
$ sudo gpasswd -a <-USERNAME-><-GROUPNAME>
```

```
$ sudo useradd <-USERNAME-> -g <-GROUPNAME-> -s /sbin/nologin -  
d /dev/null
```

Agregar el usuario en el siguiente directorio:

```
$sudo nano /etc/vsftpd.chroot_list
```

```
<-USERNAME->
```

Salir y guardar los cambios con CTRL+X, Y, Enter.

Reiniciar el servidor FTP:

```
$ sudo systemctl restart vdftpd
```

Anexo D

Instalación de la plataforma Node.js y la inclusión del módulo Socket.io

Node.js realmente es sólo otro contexto que permite correr código JavaScript en el backend, fuera del browser. Es decir, el código JavaScript que se pretende correr en el backend, este necesita ser interpretado y, eventualmente ejecutado. Esto es lo que Node.js realiza, haciendo uso de la Máquina Virtual V8 de Google, el mismo entorno de ejecución para JavaScript que Google Chrome utiliza.

Además, Node.js viene con muchos módulos útiles, de manera que no es necesario escribir todo desde cero, como, por ejemplo, algo que ponga un string a la consola.

Entonces, Node.js en realidad se define como dos cosas: un entorno de ejecución y una librería. Para hacer uso de éstas (la librería y el entorno), es primordial instalar Node.js como primer paso.

Comandos de instalación:

Ejecutar el siguiente comando por consola

```
$ sudo apt-get install curl
```

```
$ curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
```

```
$ sudo apt-get install -y nodejs
```

Alternativa para Node.js v6:

```
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

```
$ sudo apt-get install -y nodejs
```

Para comprobar la instalación:

```
$ node -v // Proporciona la versión del Node.js instalado
```

```
$ npm -v // Proporciona la versión del NPM instalado
```

Socket.io es un módulo que nos permite controlar eventos en tiempo real a través de conexiones TCP proporcionando una comunicación bidireccional evitando problemas de compatibilidad entre equipos. En cierta manera es un buen sustituto de AJAX como tecnología para obtener datos del servidor, ya que no tenemos que pedirlos, el servidor nos los enviará cuando haya nuevos.

Está desarrollado completamente en Javascript y, su objetivo es hacer que las aplicaciones en tiempo real tengan posibilidad de ejecutarse en cualquier navegador, incluidos los dispositivos móviles.

Comandos de instalación:

```
$ npm install --save express
```

```
$ npm install --save socket.io
```

Anexo E

InterfazTactil.html y su respectivo código fuente en JavaScript y Canvas

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Interfaz Tactil</title>

</head>

<body>

<script src="/socket.io/socket.io.js"></script>

<script src="http://192.168.0.50/Archivos/TouchSocket/jquery-1.11.1.js"></script>

<p>Resultado0: <span id="resultado0"></span></p>

</body>

</html>

<script>

/* -----
Inicialización de variables y llamado de eventos para la utilización de la interfaz
touch
----- */

var socket = io();

var canvas = document.createElement( 'canvas' );

var c = canvas.getContext( '2d' );

var container = document.createElement( 'div' );

container.className = "container";

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

document.body.appendChild( container );
```



```

container.appendChild(canvas);

var botones = ['x', 'x', 'x', 'x', 'x', 'x'], mouseX, mouseY, touchable = 'createTouch' in
document, touches = [];

setInterval(draw, 1000/35);

function onTouchStart(e) {
    e.preventDefault();
    touches = e.touches;
}

function onTouchMove(e) {
    e.preventDefault();
    touches = e.touches;
}

function onTouchEnd(e) {
    e.preventDefault();
    touches = e.touches;
}

function onMouseMove(event) {
    mouseX = event.offsetX;
    mouseY = event.offsetY;
}

function resetCanvas (e) {
    canvas.width = window.innerWidth;
    canvas.height = window.innerHeight;
    window.scrollTo(0,0);
}

if(touchable) {
    canvas.addEventListener( 'touchstart', onTouchStart, false );
    canvas.addEventListener( 'touchmove', onTouchMove, false );
    canvas.addEventListener( 'touchend', onTouchEnd, false );
    window.onorientationchange = resetCanvas;
    window.onresize = resetCanvas;
}

```

```

else {
    canvas.addEventListener( 'mousemove', onMouseMove, false );
    window.ondirectionchange = resetCanvas;
    window.onresize = resetCanvas;
}

```

```

/* *****

```

Asignación de valores para cada botón de acuerdo a su respectiva posición en la pantalla

```

***** */

```

```

function boton(posX, posY, ancho, alto){
    if(ancho > alto){
        if(posX >= (ancho*.1) & posX < (ancho*.2) & posY >= (alto*.3 -
            ancho*.05) & posY < (alto*.3 + ancho*.05)){
            botones[0] = 'a';
        }
        if(posX >= (ancho*.1) & posX < (ancho*.2) & posY >= (alto*.7 -
            ancho*.05) & posY < (alto*.7 + ancho*.05)){
            botones[1] = 'b';
        }
        if(posX >= (ancho*.6) & posX < (ancho*.7) & posY >= (alto*.5 -
            ancho*.05) & posY < (alto*.5 + ancho*.05)){
            botones[2] = 'c';
        }
        if(posX >= (ancho*.7) & posX < (ancho*.8) & posY >= (alto*.3 -
            ancho*.05) & posY < (alto*.3 + ancho*.05)){
            botones[3] = 'd';
        }
        if(posX >= (ancho*.7) & posX < (ancho*.8) & posY >= (alto*.7 -
            ancho*.05) & posY < (alto*.7 + ancho*.05)){
            botones[4] = 'e';
        }
        if(posX >= (ancho*.8) & posX < (ancho*.9)
            & posY >= (alto*.5 - ancho*.05) & posY < (alto*.5 + ancho*.05)){

```

```

        botones[5] = 'f';
    }
}
else{
    if(posX >= (ancho*.7 - alto*.05) & posX < (ancho*.7 + alto*.05) &
    posY >= (alto*.1) & posY < (alto*.2)){
        botones[0] = 'a';
    }

    if(posX >= (ancho*.3 - alto*.05) & posX < (ancho*.3 + alto*.05) & posY >=
    (alto*.1) & posY < (alto*.2)){
        botones[1] = 'b';
    }

    if(posX >= (ancho*.5 - alto*.05) & posX < (ancho*.5 + alto*.05) & posY >=
    (alto*.6) & posY < (alto*.7)){
        botones[2] = 'c';
    }

    if(posX >= (ancho*.7 - alto*.05) & posX < (ancho*.7 + alto*.05) & posY >=
    (alto*.7) & posY < (alto*.8)){
        botones[3] = 'd';
    }

    if(posX >= (ancho*.3 - alto*.05) & posX < (ancho*.3 + alto*.05) & posY >=
    (alto*.7) & posY < (alto*.8)){
        botones[4] = 'e';
    }

    if(posX >= (ancho*.5 - alto*.05) & posX < (ancho*.5 + alto*.05) & posY >=
    (alto*.8) & posY < (alto*.9)){
        botones[5] = 'f';
    }
}
}

```

/* **** */

Devolución de datos y presentados en el HTML a través del Socket

**** */

```

var mensaje;

socket.on('chat1', function(msg){

    document.getElementById("resultado0").innerHTML = msg;

});

m = 0;

/* *****
Envío de datos a través del Socket y diseño de los botones con el uso del Canvas
***** */

function draw() {
datos = botones[0] + botones[1] + botones[2] + botones[3] + botones[4] + botones[5];

    socket.emit('chat', datos);

    c.clearRect(0,0,canvas.width, canvas.height);

    c.font="30px Arial";

    if(canvas.width > canvas.height){

        gridX = canvas.width * .1;
        gridY = canvas.height * .2;

        c.beginPath();

        c.strokeStyle = "red";

        c.lineWidth = "3";

        c.arc(gridX * 1.5, gridY * 1.5, gridX / 2, 0, Math.PI*2, true);

        c.stroke();

        c.beginPath();

        c.arc(gridX * 1.5, gridY * 3.5, gridX / 2, 0, Math.PI*2, true);

        c.stroke();

        c.beginPath();

        c.arc(gridX * 6.5, gridY * 2.5, gridX / 2, 0, Math.PI*2, true);

        c.stroke();

        c.beginPath();

        c.arc(gridX * 7.5, gridY * 1.5, gridX / 2, 0, Math.PI*2, true);

        c.stroke();

```

```

    c.beginPath();
    c.arc(gridX * 7.5, gridY * 3.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
    c.beginPath();
    c.arc(gridX * 8.5, gridY * 2.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
}
else{
    gridX = canvas.height * .1;
    gridY = canvas.width * .2;
    c.beginPath();
    c.strokeStyle = "green";
    c.lineWidth = "3";
    c.arc(gridY * 1.5, gridX * 1.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
    c.beginPath();
    c.arc(gridY * 3.5, gridX * 1.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
    c.beginPath();
    c.arc(gridY * 2.5, gridX * 6.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();

    c.beginPath();
    c.arc(gridY * 1.5, gridX * 7.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
    c.beginPath();
    c.arc(gridY * 3.5, gridX * 7.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
    c.beginPath();
    c.arc(gridY * 2.5, gridX * 8.5, gridX / 2, 0, Math.PI*2, true);
    c.stroke();
}

```

```

    }
    if(touchable) {
        botones = ['x', 'x', 'x', 'x', 'x', 'x'];
        for(var i=0; i<touches.length; i++)
        {
            var touch = touches[i];
            boton(touch.clientX, touch.clientY, canvas.width, canvas.height);

                c.beginPath();
                c.strokeStyle = "cyan";
                c.lineWidth = "6";
                c.arc(touch.clientX, touch.clientY, 40, 0, Math.PI*2, true);
                c.stroke();

            }
            c.beginPath();
            c.fillStyle = "blue";
            c.fillText("bt[0]="+botones[0]+" bt[1]="+botones[1]+" bt[2]="+botones[2]+"
                bt[3]="+botones[3]+" bt[4]="+botones[4]+" bt[5]="+botones[5], 0, 30);
            c.stroke();
        }
        else {
            c.fillStyle = "blue";
            c.fillText("mouse : "+mouseX+", "+mouseY, mouseX, mouseY);
        }
    }
}
</script>

```

Anexo F

Código fuente del archivo Index.js para entablar la comunicación entre .html y .py

```
var spawn = require('child_process').spawn;
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);
var S = require('string');
ant = 0;

/* *****
Recepción datos del archivo HTML a cuál apunta
***** */

app.get('/', function(req, res){
  res.sendFile(__dirname + '/InterfazTactil.html');
});

/* *****
Envío de datos y ejecución del archivo .py
***** */

io.on('connection', function(socket){
  socket.on('chat', function(msg){
    if(ant != msg){
      ant = msg;
      console.log('datos: ', msg);
      io.emit('chat1', msg);
      est = 0;
      if(msg == 'xxxxxx'){
        console.log('js: Apagado');
        est = 0;
      }
      if (S(msg).contains('c')){
```

```

        console.log('js: Izquierda');
        est = 4;
    }
    if(S(msg).contains('d')){
        console.log('js: Avanza');
        est = 1;
    }
    if(S(msg).contains('e')){
        console.log('js: Reversa');
        est = 2;
    }
    if (S(msg).contains('f')){
        console.log('js: Derecha');
        est = 3;
    }
    var py = spawn('python', ['servosdy.py']);
    py.stdout.on('data', function(dat){
        console.log(dat.toString());
    });
    py.stdin.write(JSON.stringify(est));
    py.stdin.end();
}
});
});
/* *****
Ingreso del puerto por el cual escucha el servidor HTML
***** */
http.listen(3000, function(){
    console.log('Escuchando en *:3000');
});

```


Anexo G
servody.py y su respectivo código fuente para la activación de los
servomotores Dynamixel

```
# servody.py.py
import sys, json, numpy as np
import lib_robotis as lib
#Read data from stdin

/* *****
Recepción de datos enviados desde el archivo Index.js
***** */

def read_in():
    lines = sys.stdin.readlines()
    #Since our input would only be having one line, parse our JSON data from that
    return json.loads(lines[0])

/* *****
Activación de los servomotores de acuerdo al dato recibido
***** */

def main():
    #get our data as an array from read_in()
    lin = read_in()
    servo= lib.USB2Dynamixel_Device('/dev/ttyUSB0',1000000)

    if lin == 0:
        robot=lib.Robotis_Servo(servo,1)
        robot.set_angvel(0)
        robot=lib.Robotis_Servo(servo,2)
        robot.set_angvel(0)
        print 'py: Apagado'
    if lin == 1:
        robot=lib.Robotis_Servo(servo,1)
```

```
robot.set_angvel(10)
robot=lib.Robotis_Servo(servo,2)
robot.set_angvel(-10)
print 'py: Avanza'
if lin == 2:
    robot=lib.Robotis_Servo(servo,1)
    robot.set_angvel(-10)
    robot=lib.Robotis_Servo(servo,2)
    robot.set_angvel(10)
    print 'py: Reversa'
#start process
if __name__ == '__main__':
    main()
```

Anexo H

Código de programación del controlador PID

```
/******
```

Inclusión de las librerías para la lectura, escritura y aplicación del PID

```
*****/
```

```
#include <PID_v1.h>
```

```
#include <Servo.h>
```

```
# include <Wire.h>
```

```
#define MPU 0x68
```

```
#define A_R 16384.0
```

```
#define G_R 131.0
```

```
#define RAD_A_DEG = 57.295779
```

```
/******
```

Declaración de variables para el tratamiento posterior de los datos adquiridos

```
*****/
```

```
Servo myservo;
```

```
int buttonPin = 7;
```

```
int buttonPin1 = 6;
```

```
int ledPin = 13;
```

```
int buttonState = 0;
```

```
int buttonState1 = 0;
```

```
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;
```

```
float Acc[2];
```

```
float Gy[2];
```

```
float Angle[2];
```

```
float t, s;
```

```
/******
```

Aplicación de las contantes de sintonización del control PID

```
*****/
```

```
float Kp=2;
```

```
float Ki=1.2;
```

```
float Kd=18;
```

```

double Setpoint, Input, Output;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
const int sampleRate = 1;
const long serialPing = 500;
unsigned long now = 0;
unsigned long lastMessage = 0;

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin1, INPUT);
  myservo.attach(9);
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true);
  AcX=Wire.read()<<8|Wire.read();
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();
  Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
  Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,4,true);
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();
  Gy[0] = GyX/G_R;
  Gy[1] = GyY/G_R;
  Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
  Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];
  t = Angle[0] + 90;

```

```

Input = map(t, 0, 180, 0, 255);
Setpoint = 113;
Serial.begin(9600);
myPID.SetMode(AUTOMATIC);
myPID.SetSampleTime(sampleRate);
Serial.println("INICIO");
lastMessage = millis();
}

void loop()
{
/*****

Control de dirección del robot esfera (derecha-izquierda)

*****/

buttonState = digitalRead(buttonPin);
buttonState1 = digitalRead(buttonPin1);
if (buttonState == HIGH) {
myservo.write(120);
}
else if(buttonState1 == HIGH){
myservo.write(70);
}
else{
/*****

Algoritmo de ejecución del control PID

*****/

Setpoint = 113;
Wire.beginTransmission(MPU);
Wire.write(0x3B);

```

```

Wire.endTransmission(false);

Wire.requestFrom(MPU,6,true);

AcX=Wire.read()<<8|Wire.read();
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();

Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;

Wire.beginTransmission(MPU);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU,4,true);
GyX=Wire.read()<<8|Wire.read();
GyY=Wire.read()<<8|Wire.read();
Gy[0] = GyX/G_R;
Gy[1] = GyY/G_R;
Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];
t = Angle[0] + 90;
Input = map(t, 0, 180, 0, 255);
myPID.Compute();
s = map(Output, 0, 255, 100,20);
myservo.write(s);
String stringOne="I";
String stringThree=stringOne+t+'a'+s+'F';
Serial.println(stringThree);
delay(10);
}
}

```

Anexo I

Distribución de los GPIO de la tarjeta de control Raspberry Pi 2

Al programar los pines GPIO hay dos maneras diferentes para referirse a ellos: GPIO de numeración ó BCM y la numeración física ó BOARD

NUMERACIÓN GPIO O BCM

Se refiere a los pines por el número de canal Braodcom SoC.

NUMERACIÓN FÍSICA

Se refiere a los pines por el número de la clavija, es decir, los números impresos en la tarjeta.

Se trata de numeración física, como se aprecia en la Figura H-1.

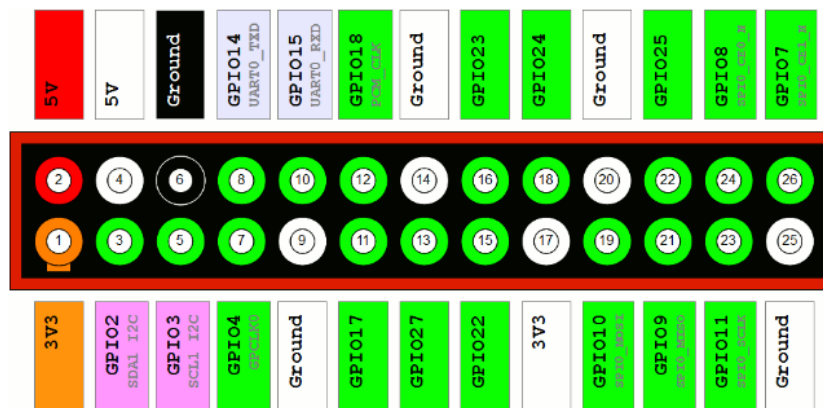


Figura H-1: Distribución GPIO modelo B.[57]

Tabla H-8: Tabla de numeración física y numeración GPIO.[57]

cuña de la seda	Python (BCM)	WiringPi GPIO	Nombre	P1 Número de patilla		Nombre	WiringPi GPIO	Python (BCM)	cuña de la seda
			3.3v de alimentación de CC	1	2	5v de alimentación de CC			
SDA		8	GPIO02 (SDA1, I2C)	3	4	5v de alimentación de CC			
SCL		9	GPIO03 (SCL1, I2C)	5	6	Suelo			
G4	4	7	GPIO04 (GPIO_GCLK)	7	8	GPIO14 (TXD0)	15		TXO
			Suelo	9	10	GPIO15 (RXD0)	dieciséis		RXI
G17	17	0	GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)	1	18	G18
G27	27	2	GPIO27 (GPIO_GEN2)	13	14	Suelo			
G22	22	3	GPIO22 (GPIO_GEN3)	15	dieciséis	GPIO23 (GPIO_GEN4)	4	23	G23
			3.3v de alimentación de CC	17	18	GPIO24 (GPIO_GEN5)	5	24	G24
MOSI		12	GPIO10 (SPI_MOSI)	19	20	Suelo			
MISO		13	GPIO09 (SPI_MISO)	21	22	GPIO25 (GPIO_GEN6)	6	25	G25
		(Sin worky 14)	GPIO11 (SPI_CLK)	23	24	GPIO08 (SPI_CEO_N)	10		CD0
			Suelo	25	26	GPIO07 (SPI_CE1_N)	11		CE1
IDSD		30	ID_SD (I2C EEPROM ID)	27	28	ID_SC (I2C EEPROM ID)	31		IDSC
G05	5	21	GPIO05	29	30	Suelo			
G6	6	22	GPIO06	31	32	GPIO12	26	12	G12
G13	13	23	GPIO13	33	34	Suelo			
G19	19	24	GPIO19	35	36	GPIO16	27	dieciséis	G16
G26	26	25	GPIO26	37	38	GPIO20	28	20	G-20
			Suelo	39	40	GPIO21	29	21	G21