



**UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E  
INDUSTRIAS**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**DESARROLLO DE LA INTERFAZ ELECTRÓNICA Y  
ELÉCTRICA PARA EL CONTROL DE UN ROBOT HUMANOIDE**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN MECATRÓNICA**

**JOFRE MARCELO SUQUI PACHECO**

**DIRECTOR: FAUSTO FREIRE PhD.**

**Quito, noviembre 2016**

© Universidad Tecnológica Equinoccial. 2016  
Reservados todos los derechos de reproducción

## **FORMULARIO DE REGISTRO BIBLIOGRÁFICO**

### **PROYECTO DE TITULACIÓN**

<b>DATOS DE CONTACTO</b>	
<b>CÉDULA DE IDENTIDAD:</b>	<b>1722611694</b>
<b>APELLIDOS Y NOMBRES:</b>	<b>SUQUI PACHECHO JOFRE MARCELO</b>
<b>DIRECCIÓN:</b>	<b>Av. Díaz de la Madrid y Politt Lazo</b>
<b>EMAIL:</b>	<b>marce_pasu@hotmail.com</b>
<b>TELÉFONO FIJO:</b>	<b>022770373</b>
<b>TELÉFONO MOVIL:</b>	<b>0990082098</b>

<b>DATOS DE LA OBRA</b>	
<b>TITULO:</b>	<b>DESARROLLO DE LA INTERFAZ ELECTRÓNICA Y ELÉCTRICA PARA EL CONTROL DE UN ROBOT HUMANOIDE</b>
<b>AUTOR O AUTORES:</b>	<b>JOFRE MARCELO SUQUI PACHECO</b>
<b>FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:</b>	<b>13 DE OCTUBRE DE 2016</b>
<b>DIRECTOR DEL PROYECTO DE TITULACIÓN:</b>	<b>FAUSTO FREIRE PhD.</b>
<b>PROGRAMA</b>	<b>PREGRADO <input checked="" type="checkbox"/> POSGRADO <input type="checkbox"/></b>
<b>TITULO POR EL QUE OPTA:</b>	<b>INGENIERO EN MECATRÓNICA</b>
<b>RESUMEN:</b>	<p>El presente trabajo trata sobre el desarrollo y diseño electrónico y eléctrico de un robot humanoide. Para el avance de este proyecto, se basa en la mecatrónica como una ingeniería concurrente. Los principales referentes teóricos están vinculados con la propia mecatrónica con una perspectiva histórica, definición y alcance de la misma; la robótica y elementos de computación libre como Arduino o Raspberry Pi. La metodología utilizada fue la mecatrónica y particularmente del Modelo en V. El diseño electrónico propiamente, implicó la selección</p>

	<p>y adaptación de la fuente de energía del robot, para lo que se utilizó baterías tipo LiPo, la implementación de placa Arduino Nano y shield Adafruit de 16 canales de Arduino junto con servos que permitieran la adecuada movilización del humanoide, el uso de sensores de proximidad y giroscopio, así como la capacidad de comunicación entre el controlador remoto Raspberry Pi y la conexión de Arduino con el uso de software libre. Este proyecto forma una de tres partes del proyecto final, los cuales tratan sobre la creación del humanoide, la parte mecánica elaborado totalmente en impresora 3D y la integración a Raspberry para poder ser manipulado desde una aplicación WEB.</p>
<p><b>PALABRAS CLAVES:</b></p>	<p>Humanoide, Arduino, Marcha Humana, Baterías LiPo, Raspberry Pi, Software libre.</p>
<p><b>ABSTRACT:</b></p>	<p>This paper deals with the development and electronic and electrical design of a humanoid robot. For the advancement of this project, it relies on mechatronics as a concurrent engineering. The main theoretical references are linked to the mechatronics itself with a historical perspective, definition and scope of the same; Robotics and free-computing elements like Arduino or Raspberry Pi. The methodology used was mechatronics and particularly the V-model. The electronic design itself involved the selection and adaptation of the robot's energy source, using LiPo batteries, the implementation of Arduino Nano plate and Adafruit shield of 16 channels of Arduino along with servos that allowed the proper mobilization of the humanoid, the use of proximity sensors and gyroscope, as well as</p>

	the ability to communicate between the Raspberry Pi remote controller and the Arduino connection with the use of free software. This project forms one of three parts of the final project, which deals with the creation of the humanoid, the mechanical part totally elaborated in 3D printer and the integration to Raspberry to be manipulated from a WEB application.
KEYWORDS	Humanoid, Arduino, Human gait, LiPo batteries, Raspberry Pi, Open source.

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.

f:   
SUQUI PACHECO JOFRE MARCELO  
1722611694

## DECLARACIÓN Y AUTORIZACIÓN

Yo, **SUQUI PACHECO JOFRE MARCELO**, CI: 1722611694 autor del proyecto: **Desarrollo de la interfaz electrónica y eléctrica para el control de un robot humanoide** previo a la obtención del título de **INGENIERO EN MECATRÓNICA** en la Universidad Tecnológica Equinoccial.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad Tecnológica Equinoccial a tener una copia del referido trabajo de graduación con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 12 de octubre del 2016.

f. \_\_\_\_\_

SUQUI PACHECO JOFRE MARCELO

1722611694

## DECLARACIÓN

Yo **SUQUI PACHECO JOFRE MARCELO**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

f: 

---

**SUQUI PACHECO JOFRE MARCELO**  
1722611694

## CERTIFICACIÓN

Certifico que el presente trabajo que lleva por título “**Desarrollo de la interfaz electrónica y eléctrica para el control de un robot humanoide**”, que, para aspirar al título de Ingeniero en Mecatrónica fue desarrollado por **Jofre Marcelo Suqui Pacheco**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 19, 27 y 28.



Fausto Freire PhD.

**DIRECTOR DEL TRABAJO**

C.I. 1802424737



# DEDICATORIA

## **A Jehová.**

*El creador del Universo.*

## **A mis padres.**

*Jaime Suqui y Teresa Pacheco, quienes siempre me apoyaron en todo lo necesario para cumplir este importante logro, sin su ayuda todo me hubiese sido muy difícil, a ellos con quien siempre estaré en deuda por que jamás podré pagar todo el amor que me mostraron.*

## **A mi familia.**

*Mis queridos abuelitos que con sus consejos me permitieron darme cuenta que la sabiduría viene de la mano con la edad.*

*Mi hermano Aldair Suqui que me hizo compañía muchas veces y lo sigue haciendo.*

*Mis tíos y primos, quienes me han demostrado siempre un gran afecto los quiero mucho.*

*A Karen Macías quien me ayudo para que este trabajo sea una realidad.*

*Y a mis queridas mascotas Osito, Coco y Oso.*

## **AGRADECIMIENTO**

Agradezco este trabajo de Tesis al Ing. Luis Hidalgo, aunque no formo parte de este trabajo de Tesis, fue un ejemplo de docente en mis años de estudiante universitario, por sus actitudes y conocimientos, los que marcaron los cimientos de responsabilidad y superación en mi vida profesional.

# ÍNDICE DE CONTENIDOS

	PÁGINA
RESUMEN .....	VIII
ABSTRACT .....	IX
1 INTRODUCCIÓN .....	
2 MARCO TEÓRICO.....	
2.1 ANTECEDENTES DE LA INVESTIGACIÓN.....	5
2.1.1 INVESTIGACIONES INTERNACIONALES .....	5
2.1.2 INVESTIGACIONES NACIONALES.....	6
2.2 BASES TEÓRICAS.....	6
2.2.1 ROBÓTICA.....	6
2.2.2 ARDUINO.....	10
2.2.3 LENGUAJE DE PROGRAMACIÓN.....	11
2.2.3.1 Elementos básicos en la Programación en C++.....	12
2.2.4 FUENTE DE ENERGÍA .....	13
2.2.5 BUS I2C .....	15
2.2.6 GIROSCOPIO .....	17
2.2.7 MARCHA HUMANA .....	17
2.2.7.1 Descripción de la Marcha.....	18
2.2.7.2 Fases de la Marcha.....	18
2.2.7.3 Parámetros de la Marcha .....	20
3 METODOLOGÍA.....	
3.1 METODOLOGÍA MECATRÓNICA .....	24
3.1.1 REQUERIMIENTOS TÉCNICOS DEL PROYECTO .....	25
3.1.2 SELECCIÓN DE ALTERNATIVAS DE desarrollo.....	26
3.1.3 MÉTODO DE CRITERIOS PONDERADOS.....	27
3.1.4 PROTOCOLO DE PRUEBAS.....	29
4 DISEÑO .....	
4.1 ANÁLISIS DE DISEÑO ELECTRÓNICO .....	30
4.1.1 SELECCIÓN DE FUENTE DE ENERGÍA.....	30
4.1.2 DESCRIPCIÓN DE LAS PLACAS USADAS.....	35

4.1.2.1	Microcontrolador .....	35
4.1.2.2	Modelo de Placa Arduino .....	35
4.1.2.3	Comunicación de Placas (Shield) con Arduino.....	37
4.1.2.4	Comunicación entre Raspberry Pi y Arduino .....	38
4.1.3	<i>FUNCIONAMIENTO DE ADICIONALES COMO GIROSCOPIO..</i>	41
4.1.4	<i>LIBRERÍAS UTILIZADAS .....</i>	44
4.1.5	<i>DIAGRAMA DE CONEXIÓN .....</i>	44
4.1.6	<i>ORGANIZACIÓN DE CABLEADO PARA LOS ACTUADORES, PLACAS Y SENSORES.....</i>	44
4.1.7	<i>ANÁLISIS DE MARCHA DEL HUMANOIDE.....</i>	46
4.1.7.1	Lógica de movimientos del humanoide .....	46
4.1.7.2	Estudio de movimientos: Marcha normal hacia adelante.....	49
4.1.7.3	Estudio de movimientos: Marcha normal hacia atrás. ....	59
4.1.7.4	Estudio de movimientos: Giro hacia la derecha.....	61
4.1.7.5	Estudio de movimientos: Giro hacia izquierda.....	70
4.1.8	<i>DIAGRAMA DE FLUJO .....</i>	78
5	ANÁLISIS DE RESULTADOS .....	
5.1	PRUEBA DE FUENTE DE ENERGÍA .....	79
5.2	PRUEBA DE PLACA, SHIELDS Y SERVOS.....	79
5.3	PRUEBA DE SENSOR DE PROXIMIDAD Y GIROSCOPIO .....	80
5.4	PRUEBA DE SOFTWARE CENTRAL .....	81
5.5	COMUNICACIÓN RASPBERRY PI Y ARDUINO.....	81
5.6	PRUEBA DE VELOCIDAD Y DISTANCIA DE LA MARCHA.....	82
5.7	PRUEBAS DE ERROR EN SERVOS MOTORES.....	84
5.8	ANÁLISIS DE RESULTADOS.....	84
6	CONCLUSIONES Y RECOMENDACIONES.....	
	CONCLUSIONES.....	86
	RECOMENDACIONES .....	87
	BIBLIOGRAFÍA .....	89

## ÍNDICE DE TABLAS

	PÁGINA
<b>Tabla 1.</b> Requerimientos de diseño del proyecto .....	26
<b>Tabla 2.</b> Alternativas de solución del proyecto.....	26
<b>Tabla 3.</b> Valoración de las alternativas del proyecto.....	27
<b>Tabla 4.</b> Relación entre criterios con requerimientos de ingeniería.....	28
<b>Tabla 5.</b> Sumatoria de las alternativas en función de los criterios.....	28
<b>Tabla 6.</b> Consumo de corriente de cada servo .....	34
<b>Tabla 7.</b> Tiempos de duración de acuerdo al consumo de corriente .....	34
<b>Tabla 8.</b> Lista de definición de partes .....	49
<b>Tabla 9.</b> Posición de los ángulos .....	50
<b>Tabla 10.</b> Posición de los ángulos para primer paso .....	51
<b>Tabla 11.</b> Posición de los ángulos para segundo paso .....	52
<b>Tabla 12.</b> Posición de los ángulos para tercer paso .....	53
<b>Tabla 13.</b> Posición de los ángulos para cuarto paso.....	54
<b>Tabla 14.</b> Posición de los ángulos para quinto paso.....	55
<b>Tabla 15.</b> Posición de los ángulos para sexto paso .....	56
<b>Tabla 16.</b> Posición de los ángulos para séptimo paso .....	57
<b>Tabla 17.</b> Posición de los ángulos para inicio de giro .....	62
<b>Tabla 18.</b> Posición de los ángulos para segundo paso de giro .....	63
<b>Tabla 19.</b> Posición de los ángulos para tercer paso de giro.....	64
<b>Tabla 20.</b> Posición de los ángulos para cuarto paso de giro.....	65
<b>Tabla 21.</b> Posición de los ángulos para quinto paso de giro .....	66
<b>Tabla 22.</b> Posición de los ángulos para sexto paso de giro .....	67
<b>Tabla 23.</b> Posición de los ángulos para séptimo paso de giro .....	68
<b>Tabla 24.</b> Posición de los ángulos para finalización de giro.....	69
<b>Tabla 25.</b> Posición de los ángulos para inicio de giro .....	70
<b>Tabla 26.</b> Posición de los ángulos para segundo paso de giro .....	71
<b>Tabla 27.</b> Posición de los ángulos para tercer paso de giro.....	72
<b>Tabla 28.</b> Posición de los ángulos para cuarto paso de giro.....	73
<b>Tabla 29.</b> Posición de los ángulos para quinto paso de giro .....	74
<b>Tabla 30.</b> Posición de los ángulos para sexto paso de giro .....	75

<b>Tabla 31.</b> Posición de los ángulos para séptimo paso de giro .....	76
<b>Tabla 32.</b> Posición de los ángulos para finalización de giro .....	77
<b>Tabla 33.</b> Prueba de duración de funcionamiento .....	79
<b>Tabla 34.</b> Comparación de voltajes entre Datasheet y Prueba .....	81
<b>Tabla 35.</b> Evaluación del Tiempo de respuesta de la comunicación .....	82
<b>Tabla 36.</b> Ciclos de configuración del humanoide .....	83
<b>Tabla 37.</b> Errores de posición en los servos del humanoide .....	84

## ÍNDICE DE FIGURAS

	PÁGINA
<b>Figura 1.</b> Fases de la marcha .....	20
<b>Figura 2.</b> Modelo en V.....	24
<b>Figura 3.</b> Servomotor .....	30
<b>Figura 4.</b> Configuración de Baterías LiPo.....	32
<b>Figura 5.</b> Baterías del proyecto .....	33
<b>Figura 6.</b> Arduino Nano V3.....	36
<b>Figura 7.</b> Shield Adafruit 16 Channel .....	36
<b>Figura 8.</b> PCA9685 .....	37
<b>Figura 9.</b> Ejemplo de trama Bus I2C .....	37
<b>Figura 10.</b> Pines de Arduino Nano .....	39
<b>Figura 11.</b> Comunicación Arduino - Raspberry Pi.....	40
<b>Figura 12.</b> Monitor de Puerto Serie .....	40
<b>Figura 13.</b> InvenSense MPU2650 .....	41
<b>Figura 14.</b> Conexión MPU - Arduino .....	43
<b>Figura 15.</b> Diagrama de Conexión .....	44
<b>Figura 16.</b> Organización de Cableado.....	45
<b>Figura 17.</b> Diseño de PCB .....	45
<b>Figura 18.</b> Definición de valores.....	46
<b>Figura 19.</b> Declaración de pines por servo.....	47
<b>Figura 20.</b> Código para control de ángulos de lado izquierdo.....	47
<b>Figura 21.</b> Posición inicial .....	48
<b>Figura 22.</b> Señalización de partes del humanoide.....	49
<b>Figura 23.</b> Modelo y posición inicial .....	50
<b>Figura 24.</b> Modelo y primer paso.....	51
<b>Figura 25.</b> Modelo y segundo paso .....	52
<b>Figura 26.</b> Modelo y tercer paso.....	53
<b>Figura 27.</b> Modelo y cuarto paso.....	54
<b>Figura 28.</b> Modelo y quinto paso .....	55
<b>Figura 29.</b> Modelo y sexto paso .....	56
<b>Figura 30.</b> Modelo y séptimo paso .....	57

<b>Figura 31.</b> Secuencia inicial de marcha.....	58
<b>Figura 32.</b> Secuencia intermedia de marcha .....	59
<b>Figura 33.</b> Secuencia final de marcha .....	59
<b>Figura 34.</b> Secuencia inicial de marcha hacia atrás .....	60
<b>Figura 35.</b> Secuencia intermedia de marcha hacia atrás.....	60
<b>Figura 36.</b> Secuencia final de marcha hacia atrás.....	61
<b>Figura 37.</b> Inicio de giro.....	62
<b>Figura 38.</b> Segundo paso para girar.....	63
<b>Figura 39.</b> Tercer paso para girar.....	64
<b>Figura 40.</b> Cuarto paso para girar .....	65
<b>Figura 41.</b> Quinto paso para girar .....	66
<b>Figura 42.</b> Sexto paso para girar .....	67
<b>Figura 43.</b> Séptimo paso para girar .....	68
<b>Figura 44.</b> Séptimo paso para girar .....	69
<b>Figura 45.</b> Primer paso para girar a la izquierda .....	70
<b>Figura 46.</b> Segundo paso para girar a la izquierda.....	71
<b>Figura 47.</b> Tercer paso para girar a la izquierda.....	72
<b>Figura 48.</b> Cuarto paso para girar a la izquierda .....	73
<b>Figura 49.</b> Quinto paso para girar a la izquierda .....	74
<b>Figura 50.</b> Sexto paso para girar a la izquierda.....	75
<b>Figura 51.</b> Séptimo paso para girar a la izquierda .....	76
<b>Figura 52.</b> Finalización de giro a la izquierda .....	77
<b>Figura 53.</b> Diagrama de Flujo.....	78
<b>Figura 54.</b> Conexión de servos .....	80
<b>Figura 55.</b> Datasheet de sensor de proximidad.....	80
<b>Figura 56.</b> Código de la marcha.....	83



## ÍNDICE DE ANEXOS

	PÁGINA
<b>Anexo 1.</b> Impresoras Prusa i3, en funcionamiento .....	93
<b>Anexo 2.</b> Impresión de partes del humanoide .....	93
<b>Anexo 3.</b> Ensamble de torso y piernas.....	94
<b>Anexo 4.</b> Impresión de abdomen del humanoide .....	94
<b>Anexo 5.</b> Colocación de baterías en humanoide .....	95
<b>Anexo 6.</b> Ensamble total del humanoide con sensores y electrónica .....	95
<b>Anexo 7.</b> Código de Programación en arduino (SETUP).....	96
<b>Anexo 8.</b> Código de Programación en arduino (PRINCIPAL).....	98
<b>Anexo 9.</b> Código de Programación en arduino (LOOP).....	100

## RESUMEN

El presente trabajo trata sobre el desarrollo y diseño electrónico y eléctrico de un robot humanoide. Para el avance de este proyecto, se basa en la mecatrónica como una ingeniería concurrente. Los principales referentes teóricos están vinculados con la propia mecatrónica con una perspectiva histórica, definición y alcance de la misma; la robótica y elementos de computación libre como Arduino o Raspberry Pi. La metodología utilizada fue la mecatrónica y particularmente del Modelo en V. El diseño electrónico propiamente, implicó la selección y adaptación de la fuente de energía del robot, para lo que se utilizó baterías tipo LiPo, la implementación de placa Arduino Nano y shield Adafruit de 16 canales de Arduino junto con servos que permitieran la adecuada movilización del humanoide, el uso de sensores de proximidad y giroscopio, así como la capacidad de comunicación entre el controlador remoto Raspberry Pi y la conexión de Arduino con el uso de software libre. Este proyecto forma una de tres partes del proyecto final, los cuales tratan sobre la creación del humanoide, la parte mecánica elaborado totalmente en impresora 3D y la integración a Raspberry para poder ser manipulado desde una aplicación WEB.

## **ABSTRACT**

This paper deals with the development and electronic and electrical design of a humanoid robot. For the advancement of this project, it relies on mechatronics as a concurrent engineering. The main theoretical references are linked to the mechatronics itself with a historical perspective, definition and scope of the same; Robotics and free-computing elements like Arduino or Raspberry Pi. The methodology used was mechatronics and particularly the V-model. The electronic design itself involved the selection and adaptation of the robot's energy source, using LiPo batteries, the implementation of Arduino Nano plate and Adafruit shield of 16 channels of Arduino along with servos that allowed the proper mobilization of the humanoid, the use of proximity sensors and gyroscope, as well as the ability to communicate between the Raspberry Pi remote controller and the Arduino connection with the use of free software. This project forms one of three parts of the final project, which deals with the creation of the humanoid, the mechanical part totally elaborated in 3D printer and the integration to Raspberry to be manipulated from a WEB application.

# **1 INTRODUCCIÓN**

Parafraseando a Craig, J. (2006), generalmente los científicos presentan la sensación de que por medio de su trabajo aprenden sobre algún aspecto de ellos mismos; los físicos, psicólogos o químicos ven esta conexión en su labor. En la robótica, dicha conexión es inusualmente obvia; y a diferencia de una ciencia que solo analiza, el objetivo actual de la robótica requiere que la ingeniería se concentre en la síntesis. En sí, la robótica se relaciona con el deseo de sintetizar algunos aspectos de la función humana mediante el uso de mecanismos, sensores, actuadores y computadores. Es importante la comprensión de esto, porque el corazón de este proyecto es la locomoción de un robot humanoide.

Así pues, como indica Kurfess, T. (2005) los robots son sistemas complejos y útiles que se usan en la industria desde hace décadas; y en la medida en que avanza la tecnología, la capacidad y utilidad de los mismos aumenta drásticamente. Actualmente tenemos robots que ensamblan automóviles, hacen soldaduras, vuelan en entornos hostiles o exploran los entornos más difíciles (desde las profundidades del océano hasta las entrañas de volcanes activos); así los robots realizan tareas muy aburridas y monótonas, o muy peligrosas, o simplemente donde mejoran el desempeño que un humano tendría. En este sentido, el robot humanoide es uno de los tipos de robot en los que mayor inversión se está realizando a nivel global como la línea «REEM» de la empresa española PAL-Robotics (desde 2006), la línea «Surena» de la Universidad de Teherán (desde 2008) o el «Manav» diseñado por investigadores de Instituto de Investigación y Entrenamiento A-SET de la India en 2014; este último fue el primer robot humanoide expansible e impreso en 3D de ese país.

En este sentido, se insiste en el hecho de que existen sectores en la industria inaccesibles para los seres humanos por diversas condiciones (ya sea climatológicas, de riesgos, u otras), en los que el empleo de esta y otras clases de robots capaces de llevar a cabo estas tareas contribuye y contribuirá a nuestro desarrollo. Además, la implementación de robots controlados por Web

contribuye a la disminución de los costos de adquisición, ya que constituyen una plataforma accesible para cualquier estudiante de ingeniería.

En este orden de ideas, es importante resaltar lo que comenta Massimo Banzi en una conferencia de TED (2012):

*“Unas semanas atrás, un amigo mío le dio un carrito de juguete a su hijo de ocho años; pero en vez de ir a una tienda y comprar uno como hacemos normalmente, él se dirigió a un sitio web y descargó un archivo que luego imprimió en su impresora 3D. Así que esta idea en la que puedes manufacturar objetos digitalmente usando estas máquinas es algo que la revista El Economista define como la «Tercera Revolución Industrial».*

*Realmente, creo que hay otra revolución en marcha, y es la que tiene que ver con el hardware de código abierto y el movimiento de los «creadores», porque la impresora que mi amigo usó para crear el juguete es, de hecho, de código abierto; así que cualquiera puede ir al mismo sitio web, descargar todos los archivos que se necesitan para hacer la impresora: los archivos de construcción, el hardware, el software; todas las instrucciones están allí. Y además, esto es parte de una amplia comunidad, donde hay miles de personas alrededor del mundo que están haciendo esta clase de impresoras; y hay mucha innovación porque todo es código abierto. No necesitas el permiso de nadie para hacer crear algo genial”.*

En el caso de Ecuador, existen compañías dedicadas a la automatización industrial, robótica, diseño industrial, mercadeo y educación. De esta manera, la robótica en el país ha venido desarrollándose de manera relativamente rápida, ahora se diseñan y construyen robots autónomos y semiautónomos. Este desarrollo ha venido acompañado de concursos de robótica que realizan las universidades y empresas dedicadas a esta rama en el país, como el Concurso Ecuatoriano de Robótica de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil (UCSG, 2015).

Asimismo, existe una iniciativa llamada Teebot; que basa su tecnología en software libre y pretende iniciar a niños desde los cuatro años de edad en actividades relacionadas con programación y robótica. En este proyecto las empresas Clear Minds y EGM robotics se fusionaron para desarrollar el proyecto; la primera se encargó del software que fue desarrollado en Java, y la segunda del hardware. Con respecto al hardware, EGM demoró dos años en madurar la placa hasta que permita realizar actividades adicionales atractivas para los infantes como la instalación de sensores, dispositivos bluetooth, luces y dispositivos de audio. En función del software, indican que lo que hace es ir poniendo sentencias en el computador, a través de las cuales se decide qué hace el robot; por ejemplo, que vaya hacia adelante, que pite o encienda las luces; estas órdenes las da el niño con el simple arrastre de unos íconos de colores, que luego se transfieren al robot y serán ejecutadas por este. (Mora, A. 2015).

En función de esto, es pertinente lo que señala Bolton, W. (2010) que muchos productos de esta tecnología requieren que los ingenieros y técnicos adopten un enfoque interdisciplinario e integrado. Ya que se necesitan habilidades y conocimientos que no se limitan a una sola área en un tema. Se necesita tener la capacidad de operar y comunicarse a través de un rango de disciplinas de ingeniería y unir las con aquellas con más habilidades especializadas.

Igualmente, este proyecto; tiene una gran utilidad porque permitirá conocer el comportamiento de los diferentes componentes de hardware y software útiles para el control y manipulación de un robot humanoide mediante herramientas Web actuales.

Para el presente trabajo se planteó el siguiente objetivo:

- Diseñar el circuito electrónico y eléctrico de interfaz entre el humanoide y la PC.

Además, se plantearon los siguientes objetivos específicos:

- Aplicar diferentes métodos de programación para el movimiento del humanoide.
- Diseñar el sistema eléctrico y electrónico para el control del humanoide con métodos funcionales y viables.
- Realizar pruebas de funcionamiento necesarias para corregir y mejorar el funcionamiento del sistema.
- Implementar el uso de sensores en la programación para ampliar las funcionalidades del humanoide.

El alcance de la investigación implica la estructuración de un diseño final que se consolida en etapas previas de análisis y procesos de selección, acorde a las especificaciones iniciales establecidas que forman un punto de referencia para la generación de alternativas, y que conjuntamente brinden una solución final al problema planteado. La especificación de los requerimientos del sistema como punto inicial permite que en las etapas siguientes de diseño mecánico, electrónico y de control se establezcan soluciones óptimas. La dinámica conjunta entre subprocesos ayuda a que se realice un análisis profundo de la eficiencia que va a presentar el prototipo final del proyecto para que evalúe las alternativas posibles y se acerque a la más óptima de las soluciones.



## **2 MARCO TEÓRICO**

## 2.1 ANTECEDENTES DE LA INVESTIGACIÓN

### 2.1.1 INVESTIGACIONES INTERNACIONALES

Bordallo, F. (2012), realizó un Trabajo de grado llamado *Diseño e Implementación de brazo robot SCARA de 3gdl*, presentado para el cumplimiento parcial de los requisitos para la obtención del título de Master Universitario en Diseño y Fabricación Integrada Asistidos por Computador de la Escuela Técnica Superior de Ingeniería del Diseño de la Universidad Politécnica de Valencia (España). El SCARA, es un acrónimo de Selective Compliant Articulated Robot Arm (Brazo Robot Articulado Selectivo Obediente), es un brazo robot de tres grados de libertad. Permite trabajar de manera rápida dentro del plano X-Y utilizando sus dos primeras articulaciones. Su última articulación, una de tipo prismático, otorga control sobre el eje Z, seleccionando de esta manera la altura del plano X-Y de trabajo. Este proyecto ha servido principalmente como sustento teórico en lo referente a la robótica y además como modelo interdisciplinario.

Méndez, V. (2012) llevó a cabo una investigación titulada *Diseño y Construcción de un Sistema Mecatrónico Articulado de Dos Dedos, Actuado con Músculos Neumáticos* como requisito para la obtención del grado de Maestría en Ciencias en Ingeniería Mecatrónica del Centro Nacional de Investigación y Desarrollo Tecnológico Departamento de Mecatrónica de México. En este documento se presenta el diseño y desarrollo de un sistema mecatrónico de dos dedos conformado con músculos neumáticos; donde cada dedo contó con 4 grados de libertad, logrando producir movimientos de flexión, extensión, abducción y aducción. El prototipo estuvo constituido por una parte mecánica, electrónica, de instrumentación y computación; la mecánica integró elementos que dan soporte y rigidez al sistema, así como los mecanismos y elementos de transmisión de potencia; en tanto que la instrumentación la conforman sensores y actuadores; la parte electrónica la representó un sistema embebido destinado a la adquisición, comunicación y

control del prototipo; y el elemento computacional correspondió a la interface y al algoritmo de control implementado en el sistema embebido. La interface tuvo la función de monitorear y configurar el sistema, logrando calibrar y ajustar los sensores vía software.

### **2.1.2 INVESTIGACIONES NACIONALES**

Parra, M. (2013), elaboró un Trabajo de grado titulado *Diseño e implementación de un Sistema de control automático para un robot imitador en una plataforma humanoide mediante adquisición y procesamiento digital de imagen y video*, presentado para obtener el título de Ingeniero en Electrónica y control de la Escuela Politécnica Nacional. Sus resultados experimentales con un humanoide, sistemas de control y servomotores constituyen una referencia importante para el desarrollo de la presente indagación.

Molina, M. (2015), en investigación presentada para optar al título de Ingeniero en Mecatrónica, de la Universidad Tecnológica Equinoccial titulada *Sistema de control inteligente para robot manipulador*, orientada a ampliar aplicaciones de la inteligencia artificial mediante uso de imágenes digitales para el control de un manipulador industrial a través del desarrollo de un algoritmo de reconocimiento y asociación de formas, encontró un sistema evaluación de la cinemática del manipulador y del posicionamiento de los objetos reconocidos. Este diseño también sirve de guía e inspiración metodológica para algunos de los elementos de la presente investigación.

## **2.2 BASES TEÓRICAS**

### **2.2.1 ROBÓTICA**

El diccionario de la Real Academia Española (2015) define Robot con dos acepciones: la primera dice que es una “máquina o ingenio electrónico

programable, capaz de manipular objetos y realizar operaciones ates reservadas solo a las personas” y la segunda como “programa que explora automáticamente la red para encontrar información”. En este sentido, la palabra Robot fue usada por primera vez por Karel Capek en 1921 en la obra de teatro *Robots Universales Rossum*, derivada de la palabra checa «robota» que significa «trabajo forzado».

En relación con esto, Wise, E. (2005) indica que los robots de la obra de Capek eran creados para reemplazar a los hombres y, de manera simplificada, como mano de obra barata. Estos robots tenían memoria perfecta pero eran incapaces de tener nuevos pensamientos; a su vez, eran un reflejo de las leyendas hebreas de los «Golem» (seres de barro o arcilla que recibían la vida por medios místicos), e incluso se relacionan con el monstruo de Dr. Frankenstein. Así, algo que estas historias tienen en común es que la creación termina implicando la caída de sus creadores; son una ilustración de lo que ocurre cuando se quiere ir demasiado lejos y las consecuencias inesperadas terminan afectando. Así pues, como dice Bordallo, F. (2012), a lo largo de la historia, la robótica ha sufrido grandes cambios. En sus inicios, y hasta los años 50, los «robots» eran servomecanismos teleoperados. Su programación estaba basada en sistemas de palancas y levas, y su uso se daba en ámbitos recreativos. Por ejemplo, marionetas capaces de seguir un programa de movimiento de forma cíclica. Esto se debía principalmente a la falta de capacidad computacional; capacidad que actualmente aumenta y mejora de manera exponencial.

En este sentido, dentro de esta investigación un robot es un “agente que lleva a cabo sus tareas automáticamente con un mínimo de impulso externo más que una recreación de la vida en sí misma” (Wise, E. 2005). En función de esto se resalta lo que señalan Khare, R, et al (2007) que ya para para 1995 había aproximadamente 700,000 robots en el mundo industrial, duplicándose el número para la siguiente década y con un crecimiento sostenido, en donde se evidencia un mayor incremento en las líneas de ensamblaje de diversas

compañías por la mayor precisión y menor costo. Los robots se utilizan en muchos casos para actividades que son peligrosas o desagradables para los humanos o incluso aquellas en extremo repetitivas.

Como señala Bordallo, F. (2012) en un mundo cambiante, en el que la tecnología forma parte de nuestra vida cotidiana y de nuestro trabajo, la robótica se ha desarrollado a pasos agigantados. A través de los años ha pasado de entretenernos y asustarnos a servirnos en múltiples situaciones de riesgo. En el campo de la industria moderna, cada vez observamos un mayor número de sistemas mecánicos de asistencia. Aparecen en forma de transportines, elevadores, cintas, etc. Desde la aparición del UniMate, el primer brazo robot construido, hasta la actualidad, se ha extendido el uso de los robots en la realización de tareas de fabricación, ensamblado, pintado, embalado, entre otros; este progreso demanda de una gran cantidad de indagación y desarrollo de la fabricación y control de estos dispositivos. En este orden de ideas, es útil la clasificación que realiza Sandler, B. (1999) derivada del Buró Nacional de Estándares Norteamericano, algunos tipos de robot son:

- Robot: Dispositivo mecánico que puede ser programado para realizar una tarea de manipulación o locomoción bajo control automático. La programación puede variar ampliamente en su naturaleza.
- Robot industrial: Manipulador multifuncional programable diseñado para mover materiales, partes, herramientas o dispositivos especiales a través de diversos movimientos programados.
- Robot “pick and place”: Robot simple, generalmente con dos o tres grados de libertad, que transfieren ítems de un lugar a otro a través de movimientos de punto-a-punto.
- Manipulador: Mecanismo que usualmente se compone de una serie de segmentos articulados con varios grados de libertad cuyo propósito es agarrar y mover objetos.
- Robot inteligente: Robot programado para tomar decisiones contingentes con entradas sensoriales.

De manera que, la robótica es la rama de la ingeniería mecánica, eléctrica y ciencias de la computación que lidia con el diseño, construcción, operación y aplicación de robots; así como de los sistemas usados para su control, retroalimentación sensorial y procesamiento de la información. De esta manera está íntimamente ligado con la mecatrónica; por lo que para los efectos de esta investigación es un área de vital interés. Igualmente, resumiendo a Craig, J. (2006), actualmente expertos de diversos campos trabajan en la investigación de distintos temas de la robótica; no es común que un solo individuo maneje todos los campos relacionados, por lo que es normal esperar una subdivisión. Realizando una abstracción importante se puede dividir la robótica en cuatro grandes áreas: 1) manipulación mecánica, 2) locomoción, 3) visión computacional, y 4) inteligencia artificial. Para los efectos de esta investigación se trabaja dentro de la segunda área principal; la locomoción.

Asimismo, según Cook, D. (2009) la robótica involucra al menos cuatro grandes ramas de aprendizaje: la ingeniería eléctrica (por sensores y circuitos), la ingeniería mecánica (engranajes, motores y cuerpo), computación (por comportamiento pseudointeligente) y el arte (expresión, estilo y diversión). Sin embargo, no es necesario ser un experto en cada campo para construir un robot decente; aunque tener antecedentes en algún dominio implica generalmente que las creaciones giren en torno a esa destreza. En el camino, la robótica provee una excelente oportunidad para aprender nuevas habilidades y encontrar talentos ocultos.

Además, según Knudsen, J. (1999) un robot es una máquina cuyo comportamiento puede ser programado; una definición tan amplia que incluye cosas como hornos microondas o reproductores de DVD, muy alejado de androides hablantes. Los robots según él tienen cinco componentes fundamentales:

- Un cerebro que controla las acciones y responde a entradas sensoriales; generalmente algún tipo de computador.
- Un cuerpo que es el chasis físico que mantiene a las otras piezas juntas.
- Actuadores que permiten el movimiento del robot, generalmente motores, pero hay muchas posibilidades como los pistones hidráulicos.
- Sensores que le proporcionan información sobre el ambiente al robot.
- Una fuente de energía que proporciona lo necesario para el funcionamiento de las partes.

### **2.2.2 ARDUINO**

Según Enríquez, R. (2009) es una plataforma electrónica de prototipos de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Una plataforma Arduino puede «sentir» el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el «Arduino Programming Language» (basado en Wiring<sup>1</sup>) y el «Arduino Development Environment» (basado en Processing<sup>2</sup>). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.). Las placas se pueden ensamblar a mano o encargarse pre-ensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia de código abierto, por lo que existe libertad para adaptarlas de acuerdo necesidades específicas.

Banzi, M. (2016) señala que gracias a su simple y accesible experiencia de usuario, Arduino es usado en miles de proyectos y aplicaciones diferentes. Su

software es fácil de usar para principiantes, pero lo suficientemente flexible para usuarios avanzados; además funciona en Mac, Windows y Linux. Docentes y estudiantes lo usan para construir instrumentos científicos económicos, para probar principios químicos y físicos, o para iniciarse en programación y robótica. Diseñadores y arquitectos desarrollan prototipos interactivos, músicos y artistas lo usan para experimentar con nuevos instrumentos musicales por ejemplo. De esta forma, Arduino es una herramienta clave para aprender nuevas cosas; cualquiera, desde niños, jóvenes, entusiastas, hasta programadores o artistas pueden empezar a componer simplemente siguiendo las instrucciones paso a paso de un Kit, o compartir y/o tomar ideas en línea con otros miembros de la comunidad de Arduino.

### **2.2.3 LENGUAJE DE PROGRAMACIÓN**

Según Crespo, E. (2015) el lenguaje de programación de Arduino está basado en C++ y aunque la referencia para el lenguaje de programación del primero está disponible en la página principal del sitio en línea de Arduino, también es posible usar comandos estándar de C++ en la programación de Arduino.

Algunas características de «C» son: a) que es el lenguaje de programación de propósito general asociado al sistema operativo UNIX; b) es de medio nivel. Trata con objetos básicos como caracteres, números, etc, y también con bits y direcciones de memoria; c) es altamente portable; y d) se usa para la programación de sistemas: compiladores, editores de texto, construcción de intérpretes, entre otros.

El lenguaje C++ fue diseñado a mediados de los años 1980 por Bjarne Stroustrup; su intención fue la de extender al exitoso lenguaje de programación «C» con mecanismos que permitieran la manipulación de objetos. En virtud de esto, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Asimismo, luego se añadieron



facilidades de programación genérica, que se sumaron a los otros paradigmas admitidos (programación estructurada y la programación orientada a objetos). Por esta razón, se suele decir que el C++ es un lenguaje de programación multiparadigma; actualmente existe un estándar, denominado ISO C++.

### 2.2.3.1 Elementos básicos en la Programación en C++

- **Entre llaves «{ }»:** Definen el inicio y el final de un bloque de instrucciones; se usan para los bloques de programación `setup()`, `loop()`, `if..`, etc. Una llave de apertura «{» siempre debe ir seguida de una llave de cierre «}», si no es así el compilador dará errores. El entorno de programación de Arduino incluye una herramienta de gran utilidad para comprobar el total de llaves. Sólo tienes que hacer click en el punto de inserción de una llave abierta e inmediatamente se marca el correspondiente cierre de ese bloque (llave cerrada).
- **Punto y coma «;»:** Se utiliza para separar instrucciones en el lenguaje de programación de Arduino; así como para separar elementos en una instrucción de tipo “bucle for”. Si no se pone fin a una línea con un punto y coma se producirá en un error de compilación.
- **Bloque de comentarios «/\*... \*/»:** También conocidos como comentarios multilínea son áreas de texto que el programa ignora, cuyo uso es para las descripciones del código o comentarios que ayudan a comprender el programa. Comienzan con «/\*», terminan con «\*/» y pueden abarcar varias líneas. Como los comentarios son ignorados por el compilador y no ocupan espacio en la memoria de Arduino se pueden usar generosamente.
- **Línea de comentarios «//»:** Una línea de comentario empieza con «//» y terminan con la siguiente línea de código. Así como ocurre con los comentarios de bloque, los de línea son ignorados por el programa y no ocupan espacio en la memoria. Por lo general, después de una instrucción se usa una línea de comentario para proporcionar más información acerca de lo que hace ésta o para recordarla más adelante.

Cuando se programa en Arduino, es fundamental usar la referencia que se dispone en línea o en la ayuda del IDE de Arduino, cualquier duda sobre un comando o función; se debe consultar en la referencia de Arduino. Igualmente, existe una guía de estilo para escribir código claro de Arduino y que sea fácil de entender; aunque no es obligatorio, se recomienda: documentar al máximo, usar esquemas, predominar la facilidad de lectura sobre la eficiencia del código, poner el «setup()» y «loop()» al principio del programa, usar variables descriptivas, explicar el código al principio y usar indentación.

## **2.2.4 FUENTE DE ENERGÍA**

Según el Blog de TodoSAI (2015), que es una empresa en línea especializada en Sistemas de Alimentación Ininterrumpida, las baterías o acumuladores son dispositivos capaces de acumular energía eléctrica a través de procesos de carácter electroquímicos. En donde se llama batería eléctrica al dispositivo que, formado por una o varias celdas electroquímicas, puede convertir la energía química que almacena en electricidad; y que a diferencia de las pilas, la reacción química que permite su funcionamiento es reversible, por lo que se puede volver a cargar la batería por medios eléctricos.

Así, las baterías se caracterizan fundamentalmente por los siguientes aspectos: a) el tipo o tecnología según sea su naturaleza interna; b) la tensión o voltaje nominal que suministran, que se mide en voltios (V); c) la capacidad de carga, que determina la intensidad que la batería puede suministrar a lo largo del tiempo a su tensión nominal, que se mide en amperios/hora (Ah); d) la energía, resultado de suministrar una intensidad a una determinada tensión a lo largo del tiempo. Es el resultado de conocer la tensión y la capacidad. Se mide en vatios/hora (Wh); e) el factor de autodescarga, ya que las baterías no mantienen la carga de forma indefinida, sino que van perdiéndola poco a poco, así, este factor indica el porcentaje de carga que la batería pierde en un tiempo determinado; y f) el efecto memoria, es una consecuencia no deseada que afecta a las baterías y

que reduce su capacidad para almacenar energía, generalmente cuando una batería se carga aunque no se ha producido una descarga completa, o los ciclos de carga/descarga que ha realizado, o si ha trabajado a una temperatura elevada, o bien si ha soportado grandes demandas de intensidad.

En este sentido, según la enciclopedia de clasificaciones (2016) los tipos son:

- **Baterías de Plomo-Ácido:** están compuestas por dos electrodos de plomo, y suelen usarse en automóviles. En cuanto a su funcionamiento, en el proceso de carga el sulfato de plomo se convierte en plomo metal en el cátodo o polo negativo; mientras que, en el ánodo o polo positivo se produce la formación de óxido de plomo. En el proceso de descarga, las fases se llevan a cabo de forma invertida; así, el óxido de plomo se reduce a sulfato de plomo, y en el ánodo el plomo comienza a oxidarse, convirtiéndose en sulfato de plomo. No obstante, esta clase de transformación no puede ser repetida de manera indefinida porque luego de un tiempo el sulfato de plomo forma cristales, y no es posible realizar el proceso de manera reversible. Es en ese momento cuando la batería se ha sulfatado y ya no es posible volver a emplearla.
- **Pila alcalina:** esta clase de acumuladores están formados por cloruro de sodio o de potasio; cuentan con potencia y corriente de gran estabilidad debido a la amplia movilidad de los iones a través del electrolito. Estas pilas son blindadas con el propósito de que no se viertan los componentes de la misma.
- **Baterías de Níquel-Hierro:** estas baterías fueron desarrolladas por Edison a principios del siglo XX; se conforman por filas de tubos compuestos por acero niquelado, los cuales contienen hidróxido de níquel. Son de fabricación simple, bajo costo y pueden sobrecargarse o descargarse reiteradas veces sin por ello perder su capacidad.
- **Baterías alcalinas de Manganeso:** son similares a las pilas alcalinas, con la excepción de que están conformadas por hidróxido de potasio; su envoltura es de acero y el zinc es polvo ubicado en el centro. Tienen

un valor elevado y se emplean en máquinas de mayor consumo de energía como juguetes con motor.

- **Baterías de Níquel-Cadmio:** funcionan a partir de un ánodo de cadmio y un cátodo compuesto por hidróxido de níquel. Mientras que, el electrolito se conforma de hidróxido de potasio. Pueden ser recargadas una vez gastadas, aunque disponen de poca capacidad.
- **Baterías de Níquel-Hidruro Metálico:** emplean un ánodo de hidróxido de níquel y un cátodo compuesto por una aleación de hidruro metálico. Son reacias al contacto con las bajas temperaturas, disminuyendo en gran parte su eficacia.
- **Baterías de Iones de Litio:** disponen de un ánodo de grafito, mientras que el cátodo funciona a partir de óxido de cobalto, óxido de manganeso o trifilina. No permiten la descarga y son capaces de alcanzar potencias elevadas; aunque se ven afectadas por los cambios de temperatura.
- **Baterías de Polímero de Litio (LIPO):** cuentan con características análogas a las baterías de iones de litio, aunque su densidad es mayor. Son de tamaño reducido por lo que suelen utilizarse en pequeños equipos.

### 2.2.5 BUS I2C

Según Carletti, E. (2015), el bus I2C, fue diseñado por Philips y es un estándar que permite la comunicación entre memorias, microcontroladores y otros dispositivos con cierto nivel de «inteligencia», que sólo requiere de dos líneas de señal y un común (o masa). El bus, facilita el intercambio de información a una velocidad de unos 100 Kbits por segundo. Igualmente, la forma de comunicación de datos es en serie y sincrónica; de manera que una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

Las señales del bus son las que se presentan a continuación:

- **SCL** (System Clock): es la línea de los pulsos de reloj que sincronizan el sistema.
- **SDA** (System Data): es la línea por la que se mueven los datos entre los dispositivos.
- **GND** (Masa): común de la interconexión entre todos los dispositivos *enganchados* al bus.

Los dispositivos conectados al bus I2C tienen una dirección única. Se debe indicar que existen dispositivos maestros y dispositivos esclavos; y sólo los dispositivos maestros pueden iniciar una comunicación.

Una trama I2C está comprendida por los siguientes campos:

1. **Bit de start:** Este es un bit especial. La línea SDA no puede cambiar a menos que SCL este a 0; este bit rompe dicha norma y provoca un cambio de 1 a 0 cuando SCL está a nivel alto.
2. **Address:** El primer byte enviado empieza con 7 bits de dirección, el cual indica a quién se envía o se solicita el dato.
3. **R/W (Read/Write):** El siguiente bit indica si se va a realizar una operación de lectura o escritura.
4. **ACK:** Este bit está presente al final de cada byte que se envía y permite asegurar que el byte ha llegado a su destino. De este modo el que envía deja el bit a 1 y si alguien ha recibido el mensaje fuerza ese bit a 0. Así, confirma que le ha llegado el byte y la transmisión puede continuar.
5. **1º Byte de datos:** Este es el primer byte de datos propiamente dicho ya que lo anterior no se elige y viene impuesto por el protocolo. Aquí se puede poner el dato que se desee; en caso de comunicación con sensores remotos, un uso habitual es poner el número de registro al que se quiere escribir o leer. Después del byte de datos se espera otro ACK del receptor.

6. Se repite el paso 5 tantas veces como sea necesario.

7. **Bit de Stop.** En este caso ocurre lo contrario al bit de Start, se pasa de 0 a 1 cuando la línea SCL se encuentra en alto. Esto termina la transmisión y deja el bus libre para que otro puede empezar a transmitir.

### **2.2.6 GIROSCOPIO**

La página Robologs (2014), señala que un giroscopio como el que se usa en el presente proyecto es una unidad de medición de inercia (IMU por sus siglas en inglés). Es decir, se trata de un dispositivo capaz de medir la fuerza (aceleración) y la velocidad. Genéricamente, esta unidad no mide ángulos directamente, ya que requiere algunos cálculos. Asimismo, el MPU-6050 posee Seis grados de libertad; esto significa que el acelerómetro y el giroscopio son ambos de 3 ejes ( $3+3 = 6$  grados de libertad). El MPU-6050 opera con 3.3 voltios.

### **2.2.7 MARCHA HUMANA**

Para entender el Análisis de la marcha del robot, es fundamental comprender primero cómo funciona la marcha humana, ya que los movimientos tienen su fundamento en este tipo de locomoción bípeda. Al respecto, Agudelo, A. et al (2013) expresan que el movimiento humano se compone de diversos patrones y elementos motrices, que son la consecuencia de la interacción de diferentes sistemas corporales, los cuales permiten al hombre un alto grado de funcionalidad e independencia para la realización de actividades cotidianas. Una parte básica del movimiento, es la marcha, la cual significa un patrón altamente complejo relacionado con el desplazamiento en el espacio, por lo que representa un eje fundamental de la capacidad de interacción del hombre con su entorno.

La investigación de la marcha dentro de la valoración del movimiento corporal, es una herramienta sumamente útil que descubre las posibilidades motoras,

la capacidad para realizar diversas actividades cotidianas y el grado de interacción dentro del contexto de cada sujeto. Así, el análisis cualitativo y cuantitativo de los rasgos y medidas de la marcha es de gran interés entre investigadores y clínicos.

#### **2.2.7.1 Descripción de la Marcha**

Daza, J. (2007) define la marcha como el paso bípedo que usan los humanos para movilizarse de un sitio a otro, realizando bajo esfuerzo y mínimo consumo de energía. Asimismo, Agudelo, A. et al (2013) señalan que es la capacidad de locomoción en bípedo que nos diferencia de las demás especies animales, y que con una sucesión de movimientos alternos y cadenciosos de las extremidades y del tronco determinan el traslado del centro de gravedad. La marcha se define por el contacto constante del sujeto con el suelo, con al menos una de sus extremidades, a través de la integración de varios sistemas y segmentos del cuerpo.

El patrón de marcha está relacionado con muchos factores, que pueden ser extrínsecos como el terreno, tipo de calzado o transporte de carga; intrínsecos como edad o sexo; físicos como tamaño y peso (servos y otros materiales en el caso del robot); psicológicos vinculados con la personalidad y tono afectivo del sujeto; o incluso patológicos como lesiones o enfermedades neurológicas, psiquiátricas o músculo esqueléticas.

Para realizar el análisis de la marcha se pueden usar técnicas cualitativas como el examen del patrón de marcha en pasarela, o cuantitativas a través de diversas herramientas de recolección de datos.

#### **2.2.7.2 Fases de la Marcha**

La marcha se compone de dos fases, la primera es la de soporte (o apoyo) y la segunda es la de balanceo. A continuación se describen las cinco partes de la Fase de soporte:

- **Contacto inicial:** Se encuentra dentro del 0 al 2% del ciclo, es de apoyo bipodal; implica el instante en el que el pie contacta con el suelo. Generalmente el contacto tiene lugar en la zona del talón, motivo por el cual esta fase se considera para registrar el inicio y culminación del ciclo de marcha.
- **Respuesta a la carga:** Se encuentra dentro del 10% del ciclo de marcha, es de apoyo bipodal; el pie hace contacto completo con el piso y el peso se transfiere al miembro adelantado.
- **Soporte medio:** Ocurre entre el 10% y el 30% del ciclo; el miembro contralateral (del lado opuesto) deja de tocar el piso y el peso del cuerpo se transmite a lo largo del pie hasta que se alinea con la cabeza de los metatarsianos. El traslado del peso se da debido a la rotación de la tibia sobre el pie estático.
- **Soporte Terminal o final:** Sucede entre el 30% y el 50% de la cadena de marcha; el talón se alza para mover el peso hacia los dedos y desplazar la carga al pie contralateral, el cual, pasa a tocar el piso.
- **Pre-balanceo o fase previa a la oscilación:** Acontece entre el 50% y el 60% del ciclo; es la etapa de transición entre la fase de soporte y la de balanceo. Se inicia cuando el pie contralateral toca el piso y termina cuando el pie ipsilateral (del mismo lado) se levanta del piso. Aquí, el peso se pasa totalmente de un miembro al otro.

En cambio, la Fase de balanceo consta de tres partes que se describen a continuación:

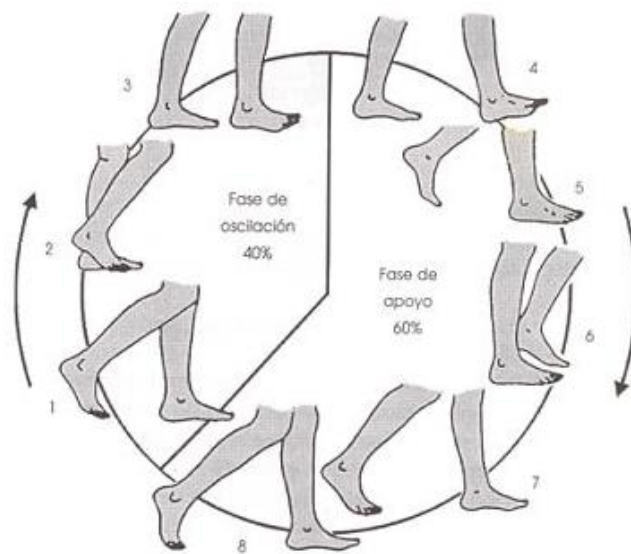
- **Balanceo Inicial:** Ocupa alrededor del 50% al 73% del ciclo; comienza cuando los dedos del pie se elevan sobre el suelo y finaliza cuando la rodilla alcanza la flexión máxima durante la marcha (de unos 60°), el muslo está debajo del cuerpo y paralelo a la extremidad inferior contralateral que, en este momento, aguanta el peso del cuerpo.
- **Balanceo Medio:** Ocurre aproximadamente entre el 73% y el 87% del ciclo; el muslo sigue en su avance y la rodilla, que ya llega a su flexión



máxima, ahora se extiende, de forma que el pie permanece levantado del piso y culmina cuando la tibia se coloca perpendicular al suelo.

- **Balanceo Terminal:** Comienza con la posición vertical de la tibia, sigue a medida que la rodilla se extiende totalmente y el miembro inferior se dispone para aceptar la carga durante el contacto inicial, reiniciando la fase de soporte.

En la figura 1 se puede observar gráficamente un ejemplo de las fases de la marcha:



**Figura 1.** Fases de la marcha  
(Igal, C., Muñoz, E., Aramburu, C., 1996)

### 2.2.7.3 Parámetros de la Marcha

La marcha también se detalla a través de varios parámetros, que por ejemplo pueden ser: espaciales, temporales o espacio-temporales; éstos son diferentes entre personas e incluso entre el mismo sujeto. Además, son característicos de un individuo cuando las circunstancias y elementos que repercuten en la marcha son constantes. Los resultados del análisis de estos parámetros permiten vincular los datos obtenidos durante el proceso de evaluación del movimiento y la tipificación de insuficiencias corporales que afectan la marcha y/o de limitaciones en la actividad.

## **Parámetros Espaciales**

- Longitud de zancada: distancia lineal entre dos toques de talón seguidos del mismo miembro.
- Longitud de paso: distancia lineal entre el primer toque del talón de una extremidad y el del miembro contralateral (depende del tamaño del sujeto).
- Ancho de paso (Amplitud de base): la distancia entre los dos pies, usualmente entre talones, significa la extensión de la base de soporte. Se vincula con el equilibrio y la estabilidad.
- Altura del paso: el desplazamiento de los miembros inferiores concede una elevación al paso, que evita el arrastre de los pies.
- Ángulo de la marcha: Implica la orientación del pie durante el apoyo, en este sentido, el eje longitudinal de cada pie genera un ángulo con la línea de dirección de la marcha.

## **Parámetros Temporales:**

- Apoyo: Se refiere al porcentaje del ciclo completo de la marcha durante el cual el cuerpo está asentado sobre una sola pierna.
- Balanceo: Implica el porcentaje del ciclo durante el cual el miembro inferior se encuentra en el aire y se mueve hacia adelante.
- Doble apoyo: Es la proporción del ciclo de la marcha en el cual los dos pies están tocando el piso.
- Periodo de zancada: Período en el que transcurren dos eventos idénticos sucesivos del mismo pie, por lo general entre dos contactos iniciales del mismo miembro inferior.
- Periodo de soporte: Se refiere al lapso de tiempo que ocurre desde que el pie toca el suelo, hasta el instante que se levantan los dedos del mismo pie.
- Periodo de balanceo: Implica el tiempo que pasa entre el instante de despegue de los dedos hasta el contacto inicial del mismo pie.

- Cadencia: Se describe como el número de pasos por unidad temporal, usualmente se calcula en un minuto; la frecuencia establece el ritmo y velocidad de la marcha.

### **Parámetros espaciotemporales**

- Velocidad: Implica la proporción de la distancia transitada en dirección de la marcha por unidad temporal ( $\text{Velocidad} = \text{Distancia} / \text{Tiempo}$ ).
- Velocidad de Balanceo: Periodo que demora una extremidad inferior desde la primera aceleración hasta el siguiente paso.
- Velocidad media: Se refiere al producto de la zancada por la cadencia (se expresa en m/seg).
- Cadencia (ritmo del paso): Se vincula con la longitud del paso y representa generalmente el ritmo más eficiente para ahorro de energía en el sujeto particular de acuerdo a su estructura corporal. Los organismos más pequeños tienden a dar pasos más rápidos, mientras que los altos suelen dar pasos a una cadencia más lenta (se expresa en pasos/min).

Es importante resaltar que la evaluación de la marcha es un instrumento diagnóstico significativo sobre todo en la valoración de patologías habitualmente vinculadas con el sistema musculoesquelético. Históricamente, su estudio ha generado interés y los mecanismos para su evaluación se han desarrollado con el tiempo; paulatinamente se han perfeccionado y facilitado las técnicas para su análisis y se han generado nuevos métodos para apreciar los diferentes parámetros de la marcha de objetiva y eficazmente. En este sentido, las técnicas existentes para el análisis de la marcha son múltiples, y facilitan la obtención de los parámetros cuantitativos característicos de una manera objetiva. Para evaluar los diversos parámetros, se han puesto en funcionamiento desde la tinta y papel, a interruptores de pie, placas de fuerza, calzado con sensores e incluso pasillos instrumentados con sistemas complejos de análisis como videogrametría (donde se colocan marcadores

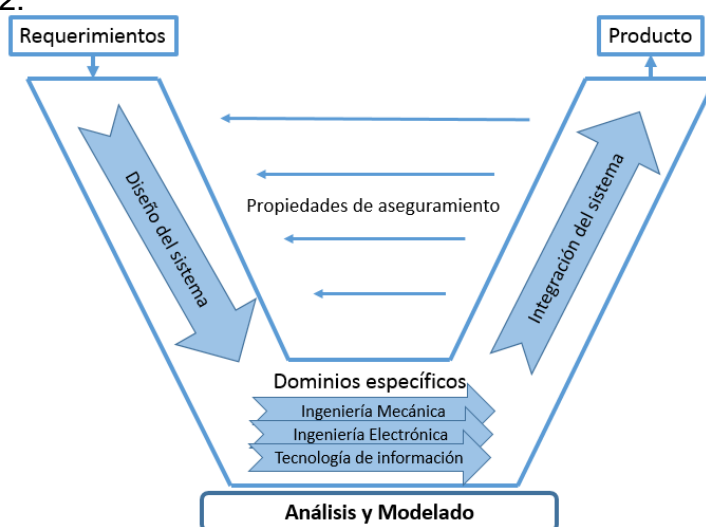
corporales, que luego detectan sistemas optoelectrónicos); de hecho, algunos de los métodos implican el uso de acelerómetros y goniómetros.

### **3 METODOLOGÍA**

### 3.1 METODOLOGÍA MECATRÓNICA

Parafraseando a Hernández Sampieri, R., et al (2014) con el propósito de cumplir con los objetivos del estudio y responder a los retos y preguntas que plantea la investigación, se debe desarrollar un diseño específico. El diseño se refiere al plan o estrategia concebida para obtener la información que se desea con el fin de responder al planteamiento del problema; una vez que se precisa el problema y se define el alcance inicial de la investigación, hay que visualizar la manera práctica y concreta de cumplir los objetivos fijados. Además, se destaca que los módulos electrónicos y eléctricos son el corazón del proyecto, ya que son el medio que permite la función e integración del humanoide; son el soporte de la interacción entre software y robot, comandos y ejecución, fuente de poder y acciones. En una analogía con biología humana: el apartado mecánico sería el sistema óseo, el control y software sería el sistema nervioso, y lo electrónico serían los órganos internos.

Igualmente, se resalta que este proyecto se inscribe dentro del movimiento de hardware y software libre, por lo que la mayoría de su operatividad está vinculada con esta corriente de diseño y aplicaciones. Con el objeto de reforzar la aproximación metodológica de este proyecto, la base del desarrollo del diseño es el proceso que se conoce como Modelo en V, como se aprecia en la figura 2:



**Figura 2.** Modelo en V  
(Padilla, 2015)

De esta manera, el presente proyecto usó el análisis y modelado en V, incorporando los aspectos que han sido señalados previamente, con la meta de conseguir una estructura lógica de diseño que encamine el desarrollo hacia un producto óptimo.

### **3.1.1 REQUERIMIENTOS TÉCNICOS DEL PROYECTO**

En el presente proyecto; se puede observar el elemento mecánico principalmente en lo que se refiere a movimiento del humanoide a través de los diferentes servos y actuadores, el elemento electrónico por medio del diseño del circuito electrónico y eléctrico que permita la interacción entre la interfaz del humanoide y la PC, y el elemento de control y software que se relaciona principalmente con la programación y comandos desde la PC.

Sin embargo, el proyecto completo implica la labor conjunta de tres estudiantes, donde cada uno se encargó de un elemento específico que en el caso de este trabajo se refiere al apartado electrónico. De esta forma, con el reconocimiento de la importancia de todos los elementos involucrados y habiendo participado en general de todo el proceso, se hará énfasis justamente en el análisis del diseño electrónico, tal como se precisa a continuación:

- Selección de fuente de energía para el robot.
- Selección de placas.
- Comunicación entre placas.
- Comunicación placas/software.
- Organización de cableado.
- Capacidad de conexión a internet.
- Autonomía inalámbrica
- Adaptabilidad a funciones adicionales (como giroscopio).
- Soporte para diversos tipos de movimiento.

La tabla 1 presenta un resumen de la función, restricciones y objetivo del trabajo de titulación.

**Tabla 1.** Requerimientos de diseño del proyecto

<b>Función</b>	Estructurar el funcionamiento interno y movimiento del robot humanoide.
<b>Restricciones</b>	<ul style="list-style-type: none"> <li>• Uso de herramientas y código libre.</li> <li>• Fuente de energía (Robot autónomo)</li> </ul>
<b>Objetivos</b>	Desarrollar una interfaz electrónica y eléctrica óptima para el funcionamiento del humanoide.

### 3.1.2 SELECCIÓN DE ALTERNATIVAS DE DESARROLLO

Con el objeto de seleccionar las mejores alternativas para el desarrollo de la estructura interna de funcionamiento del sistema eléctrico y electrónico, y la implementación de adicionales del humanoide se deben analizar los elementos tecnológicos a disposición y las metodologías actuales vinculadas con la construcción del proyecto, así mismo, se deben tomar en cuenta los aspectos financieros y operativos. Alcanzar una elección óptima implica la necesidad de proponer varias soluciones que puedan cumplir con las funciones del proyecto. En la tabla 2 a continuación se presentan las diversas alternativas contempladas.

**Tabla 2.** Alternativas de solución del proyecto

<b>Solución</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>
<b>Fuente de Energía</b>	LiPo	Alcalinas	Iones de litio
<b>Regulador de Voltaje</b>	LM2577	LM7805	LM317
<b>Disipador de Calor</b>	Sin disipador	Aluminio	Ventilado
<b>Dispositivo de Control</b>	Arduino	Microcontrolador (PIC)	Raspberry
<b>Placas de control de Servos</b>	Adafruit 16 Channel (PWM) // I2C	PWM (Microcontrolador)	PWM (Raspberry)
<b>Sensor de proximidad</b>	IR	Ultrasonico	Fotoceldas
<b>Giroscopio</b>	MPU2650	MPU2650	MPU2650
<b>Software de central</b>	Arduino (C++)	C++	Python

Para determinar la alternativa óptima de solución se realizó un análisis mediante el método de criterios ponderados.



### 3.1.3 MÉTODO DE CRITERIOS PONDERADOS

Consiste en la asignación de factores numéricos a los elementos que se consideran relevantes para la elección de la mejor alternativa de solución; lo que conduce a una comparación cuantitativa entre los mismos, a través de la ponderación de factores de preferencia para el investigador en la toma de decisiones. La principal ventaja de este método consiste en ser rápido y sencillo; aunque en ocasiones, por depender de las preferencias del investigador no sea reproducible

Para sopesar los factores relevantes en la elección de la mejor alternativa de este proyecto, se tomó en cuenta principalmente: costo, peso, funcionalidad, instalación, mantenimiento, control y vida útil. Así, de acuerdo con estos criterios se realiza una calificación con cada opción en una escala de uno (1) al cinco (5); en donde 1 es la escala más baja y 5 más alta.

En la tabla 3 se presentan los datos con los criterios ponderados para la selección de la mejor alternativa del proyecto.

**Tabla 3.** Valoración de las alternativas del proyecto

<b>ALTERNATIVAS</b>  5 ES EXCELENTE 1 ES MENOR	<b>COSTO</b>	<b>PESO</b>	<b>INSTALACIÓN</b>	<b>MANTENIMIENTO</b>	<b>CONTROL</b>	<b>VIDA ÚTIL</b>	<b>TOTAL</b>
<b>ALTERNATIVA 1</b>	5	5	5	4	4	4	27
<b>ALTERNATIVA 2</b>	4	3	4	5	5	4	25
<b>ALTERNATIVA 3</b>	5	5	2	3	3	3	21

Una vez asignados los valores de las alternativas de desarrollo, se realiza otra tabla donde se comparan los criterios de ingeniería entre sí para obtener la valoración de cada criterio. En consecuencia, se obtiene la pertinencia de cada una en el proyecto. Los valores corresponden a la relación que tenga cada criterio con el otro donde 10 es alta 5 es media y 0 es ninguna, en la

tabla 4 se observa los criterios con mayor relación entre sí. Luego, para determinar la alternativa ganadora se multiplican los valores de relación entre criterios por los valores de la tabla 5 El resultado del análisis tiene como ganador a la primera alternativa; esta solución es adecuada para las necesidades de diseño del proyecto, ya que presenta los componentes más idóneos en la solución como: Placa Arduino, Baterías LiPo, Shields Adafruit de 16 canales y el lenguaje propio de Arduino (basado en C++).

**Tabla 4.** Relación entre criterios con requerimientos de ingeniería

CRITERIOS	COSTO	PESO	INSTALACIÓN	MANTENIMIENTO	CONTROL	VIDA ÚTIL	TOTAL	PODNERACIÓN	%
<b>COSTO</b>		0	10	10	0	10	30	0,18	17,65%
<b>PESO</b>	0		10	0	5	5	20	0,12	11,76%
<b>INSTALACIÓN</b>	5	10		5	0	0	20	0,12	11,76%
<b>MANTENIMIENTO</b>	10	0	10		5	10	35	0,21	20,59%
<b>CONTROL</b>	0	0	5	10		10	25	0,15	14,71%
<b>VIDA ÚTIL</b>	10	0	10	10	10		40	0,24	23,53%
						<b>SUMA</b>	170	1,00	100%

**Tabla 5.** Sumatoria de las alternativas en función de los criterios

CRITERIOS	COSTO	PESO	INSTALACIÓN	MANTENIMIENTO	CONTROL	VIDA ÚTIL	TOTAL	PONDERACIÓN
<b>ALTERNATIVA 1</b>	0,90	0,60	0,60	0,84	0,60	0,96	4,50	1
<b>ALTERNATIVA 2</b>	0,72	0,36	0,48	1,05	0,75	0,96	4,32	2
<b>ALTERNATIVA 3</b>	0,90	0,60	0,24	0,63	0,45	0,72	3,54	3

### 3.1.4 PROTOCOLO DE PRUEBAS

Para la verificación del alcance de los objetivos de este proyecto se desarrollaron siete pruebas que evalúan el funcionamiento de los elementos clave del diseño. Las pruebas fueron las siguientes:

- Fuente de energía: Que se refiere a la duración de la alimentación del humanoide.
- Placa, shields y servos: Esta prueba implica la ponderación de la adecuación de los controladores.
- Sensor de proximidad y giroscopio: Para verificar el funcionamiento de los sensores adicionales del robot.
- Software central: Implica la evaluación de la ejecución del código de programación del humanoide.
- Comunicación Raspberry Pi y Arduino: Para medir el tiempo de respuesta de la conexión entre ambos sistemas.
- Velocidad y distancia de la marcha: Para alcanzar la velocidad adecuada de desplazamiento del humanoide.
- Errores de posición de los servos: Para evidenciar la precisión del posicionamiento de los servos.

## **4 DISEÑO**

## 4.1 ANÁLISIS DE DISEÑO ELECTRÓNICO

### 4.1.1 SELECCIÓN DE FUENTE DE ENERGÍA

Los servos seleccionados, para este proyecto se muestran en la Figura 3 y en las especificaciones siguientes:



**Figura 3.** Servomotor  
(Hobby King, 2016)

- Voltaje: 6.0v - 7.4v
- Velocidad: 0.17sec/60deg (6.0v) - 0.15sec/60deg (7.4v)
- Torque: 27.00kg.cm (6.0v) - 31.00kg.cm (7.4v)
- Dimensiones: 40mm x 20mm x 39.2mm
- Peso: 60g

Para calcular la corriente consumida por cada servo se toma en cuenta lo siguiente:

- Voltaje: 6V - 7.4V
- Velocidad: 0.15sec/60deg a 7.4V
- Torque de salida: 31Kg.cm a 7.4V

Con estos datos, el método de cálculo para el consumo de la fuente de energía fue el siguiente:

$$\text{Momento} = 31\text{Kg} \cdot \text{cm} = 31\text{Kg} \cdot \text{cm} * 10 \frac{\text{N}}{\text{Kg}} * \frac{1\text{m}}{100\text{cm}} = 3\text{N} \cdot \text{m} \quad [1]$$

$$\text{Ángulo en recorrido} = 60^\circ * \frac{\text{Pi}}{180} = 1,047 \text{ Rad} \quad [2]$$

$$\text{Velocidad Angular} = \text{Ángulo recorrido} / \text{Tiempo} = 1,047 \text{ Rad} / 0,15\text{s} = 6,98 \text{ Rad/S} \quad [3]$$

$$\text{Potencia} = \text{momento} * \text{velocidad angular} = 3\text{N} \cdot \text{m} * 6,98 \frac{\text{Rad}}{\text{s}} = 20,941 \text{ W} \quad [4]$$

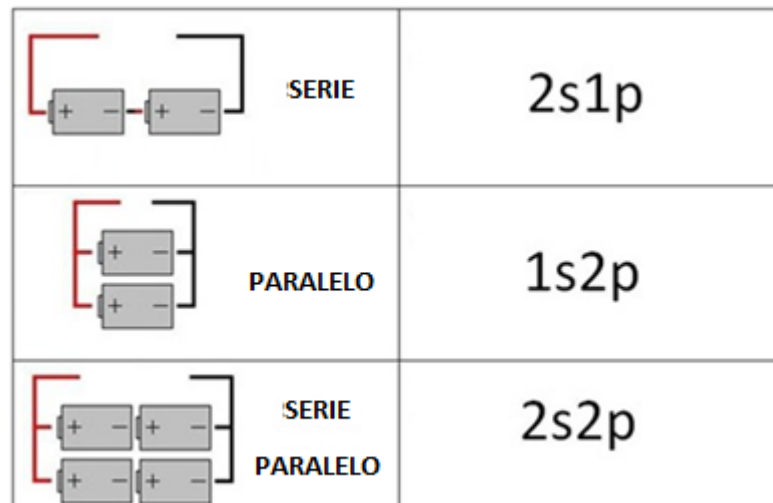
$$\text{Potencia} = \text{Voltaje} * \text{Intensidad} \implies I = \frac{P}{V} = 20,941\text{W} / 7,4\text{V} = 2,830\text{A} \quad [5]$$

La comprobación se realizó mediante un multímetro en serie con un servo; los valores de corriente obtenidos fueron entre 2A y 3A cuando estaban sometidos a un esfuerzo máximo. Lo que implica que, al momento de usarse todos los servos al mismo tiempo, la corriente supera con facilidad los 10 Amperios cuando su torque está al máximo.

Elegir la alimentación de energía adecuada para esta clase de proyectos es un reto, ya que se necesitan considerar varios aspectos técnicos que van desde su tamaño y posibilidad de integración dentro del cuerpo del robot (dadas las dimensiones particulares del mismo) hasta la capacidad y energía de la misma.

En el desarrollo de este proyecto se seleccionaron las baterías de Polímero de Litio (LiPo) con las siguientes características:

- LiPo Batería.
- Voltaje nominal de 3,7 voltios.
- Capacidad de carga de 1250 mAh o 1.25 Ah.
- Factor de autodescarga: 20C
- Efecto memoria: no se ven afectadas por este efecto.



**Figura 4.** Configuración de Baterías LiPo

La nomenclatura que se usa al trabajar con baterías LiPo, es con un dígito que indica el número de celdas que tiene y una letra que indica el tipo de conexión de dichos elementos; se utiliza la letra «S» para conexión en serie y la letra «P» para conexión en paralelo como se observa en la Figura 4. Por ejemplo: a) 2S1P o 2S: es un conjunto de 2 celdas conectadas en serie; o b) 2S2P: conjunto de 2 celdas en serie conectadas en paralelo a otro conjunto de 2 celdas en serie.

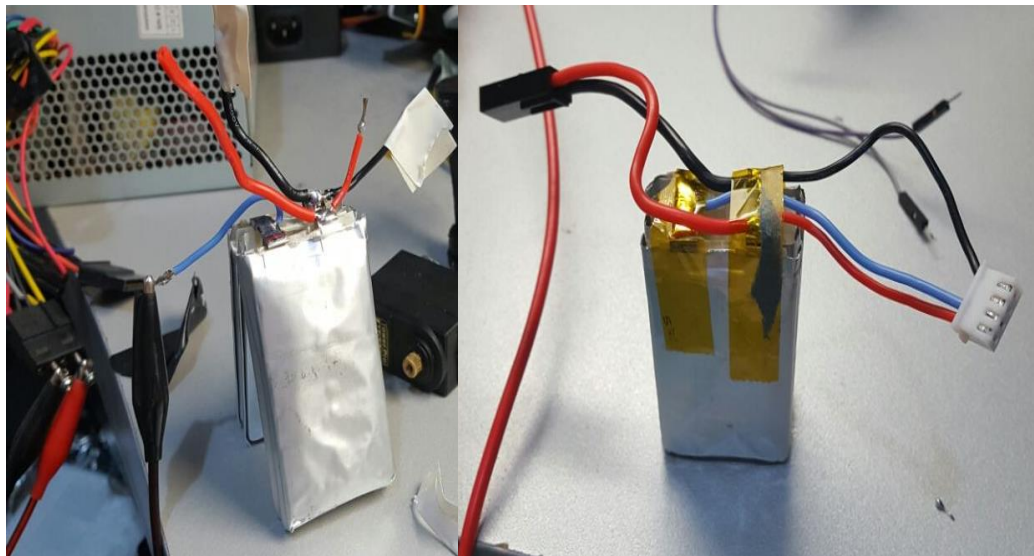
Cada elemento o celda tiene un voltaje de 3.7V de valor nominal, por lo que el valor de la batería conjunta al final va a variar dependiendo del tipo de conexión que se realice

Para el diseño, se necesitaba regular el voltaje de las baterías a 7,4 voltios; que son los necesarios para que funcionen correctamente los servos del humanoide.

Se configuraron dos baterías 2S2P (conjunto de 2 celdas en serie conectadas en paralelo a otro conjunto de 2 celdas en serie), con lo cual se obtuvo una batería LiPo con la siguiente especificación:

- Voltaje nominal de 7.4 voltios.
- Capacidad de carga de 2500 mAh o 2.5 Ah.
- Factor de auto descarga: 20C
- Efecto memoria: no se ven afectadas por este efecto.

En la figura 5 se observan las baterías armadas del proyecto:



**Figura 5.** Baterías del proyecto

Las LiPo proporcionan 2.5 Amperios por cada hora, lo previsto era tener una autonomía de alrededor de 20 a 25 minutos, para obtener el tiempo estimado de duración de las baterías, primero se toma en cuenta el tiempo mínimo de descarga de la batería.

$$Tiempo(min) = Capacidad\ Batería\ (Amp * h) / Velocidad\ de\ descarga(Amp) \quad [6]$$

La velocidad de descarga, se puede definir como la cantidad de amperios que la batería nos puede suministrar durante una hora de forma continuada.

A máxima capacidad nos proporciona lo siguiente:

$$Tiempo = 2,5Ah / (20C * 2.5A) = 2,5Ah / 50A = 0,05h \equiv 3min \quad [7]$$



Para evaluar el consumo de cada servo, en funcionamiento se presenta los valores en la tabla 6, donde se detalla el consumo de cada servo, cuando los motores se encuentran incorporados en el humanoide los motores que sufren mayor esfuerzo son los de las extremidades inferiores, la prueba de se realizó con un sensor de corriente el cual nos generó la siguiente información.

**Tabla 6.** Consumo de corriente de cada servo

SERVO	Derecha (mA)	Izquierda (mA)
pie1	321	324
pie2	284	283
rodilla	211	209
muslo1	205	201
muslo2	203	199
cadera	90	90
hombro1	90	90
hombro2	90	90
codo	90	90
<b>Promedio</b>	<b>176</b>	<b>175,111</b>

Ahora, el tiempo estimado real se muestra a continuación:

$$Tiempo = 2,5Ah / ((176 + 175.77mA) * \text{Número de servos}) \approx 2,5Ah / (350mA * 18) = 2,5Ah / 6A = 0,417h \equiv 25 \text{ min aproximadamente} \quad [8]$$

**Tabla 7.** Tiempos de duración de acuerdo al consumo de corriente

CAPACIDAD DE CARGA (A/h)	CONSUMO DE CORRIENTE (A)	TIEMPO ESTIMADO (DURACIÓN (T))
2.5 A/h	50 A	3 min
2.5 A/h	40 A	3.75 min
2.5 A/h	30 A	5 min
2.5 A/h	20 A	7.5 min
2.5 A/h	10 A	15 min
2.5 A/h	6 A	25 min
2.5 A/h	3 A	50 min
2.5 A/h	1 A	150 min

La tabla 7, permite observar un estimado del tiempo de duración de las baterías configuradas para diferentes consumos de corriente en este proyecto.

## **4.1.2 DESCRIPCIÓN DE LAS PLACAS USADAS**

### **4.1.2.1 Microcontrolador**

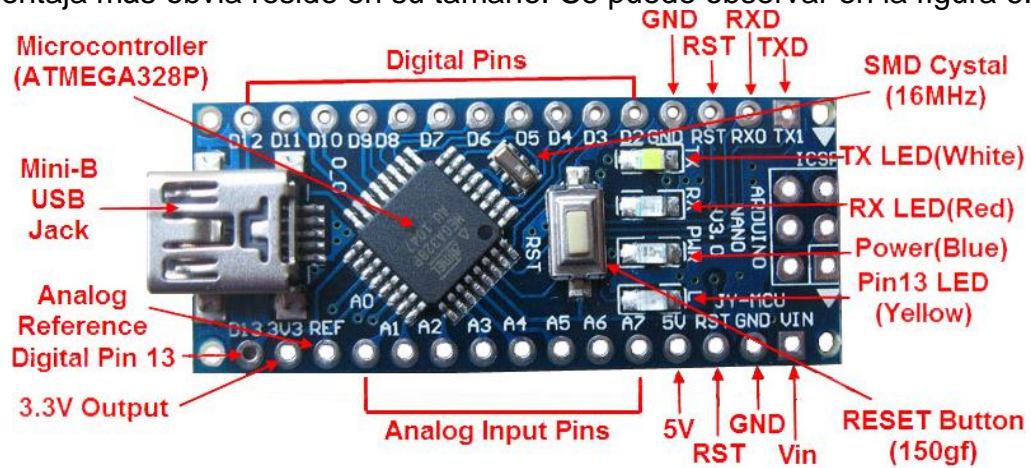
En el presente proyecto, se seleccionó la placa Arduino, dada la potencia de desarrollo de sus placas, que se puede explicar en tres características fundamentales:

- El Hardware pre-armado alrededor del Microcontrolador provee al usuario de una interface de puertos y periféricos que pueden estar ligados o no directamente al microcontrolador Atmega, sin embargo la simbología que el usuario observa es diferente de la simbología del microcontrolador.
- La potencia de las librerías de Arduino el desarrollo de múltiples aplicaciones de manera simple y rápida.
- Los Shields (módulos de hardware especial) de Arduino simplifican aún más la tarea del desarrollo de las aplicaciones puesto que existe un Shield a la medida de cada aplicación; además, estos Shields preparan la adaptabilidad de señales para acondicionamiento de los sensores, drivers de corriente, regulación de alimentación, entre otros.

El CPU del humanoide es un Raspberry Pi, es decir, un ordenador de placa reducida que tiene una integración relativamente fácil con las placas Arduino, lo que se convierte en otro elemento de peso para la selección de estas placas.

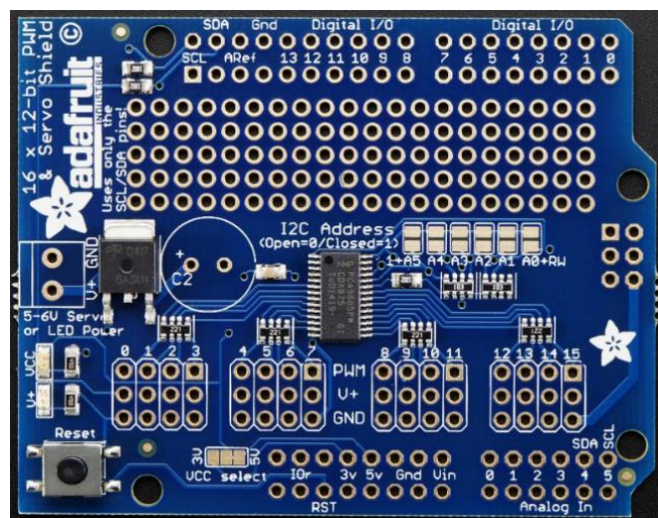
### **4.1.2.2 Modelo de Placa Arduino**

Para este proyecto se eligió la placa «Nano v3». Este Arduino es la versión más pequeña del Arduino uno mide 1,7cm de largo por 0,7cm de alto, está basada en el Atmega328 SMD; la disposición de sus pines facilitan la conexión de los componentes sin necesidad de muchos cables. Además, la ventaja más obvia reside en su tamaño. Se puede observar en la figura 6.



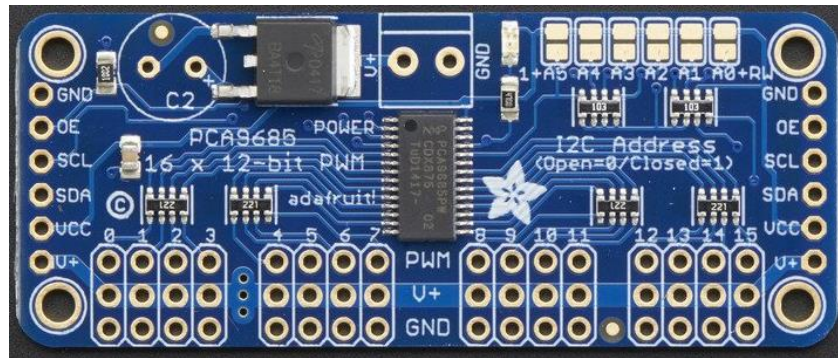
**Figura 6.** Arduino Nano V3  
(Patagoniatec, 2016)

Los 10 puertos digitales de la placa «Nano V3», resultaban insuficientes para el correcto funcionamiento de todos los servos del humanoide. Esta diferencia fue resuelta por medio de un módulo «Shield» denominado «Adafruit 16 Channel 12-bit PWM / Servo Shield - I2C interface», que permitió usar hasta dieciséis canales; como se observa en la figura 7.



**Figura 7.** Shield Adafruit 16 Channel  
(Adafruit, 2016)

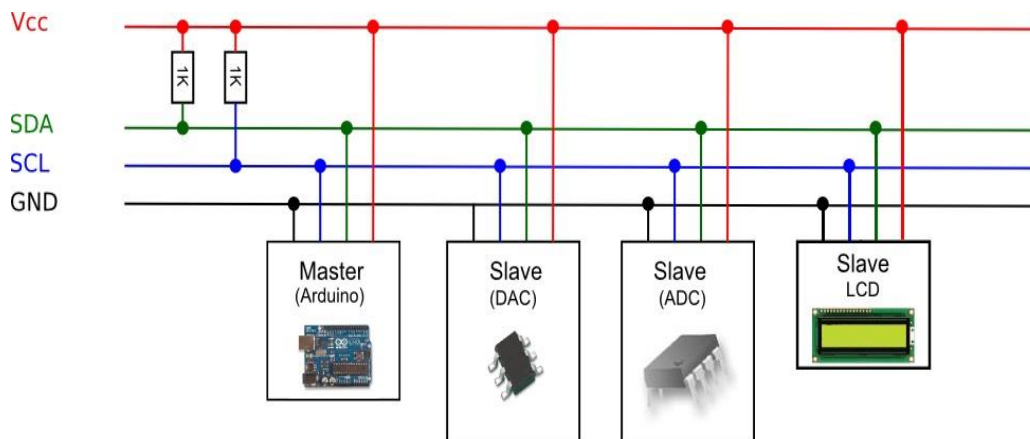
No obstante, al final se cambió este módulo por un modelo más actual, que posee dimensiones aún más reducidas y mantiene las mismas características, este modelo se llama « Adafruit 16 Channel 12-bit PWM / Servo Shield - I2C mini », se observa en la figura 8.



**Figura 8.** PCA9685  
(Adafruit, 2016)

#### 4.1.2.3 Comunicación de Placas (Shield) con Arduino

La comunicación se basa en una trama de BUS I2C, en la figura 9 se ve un ejemplo de una trama:



**Figura 9.** Ejemplo de trama Bus I2C  
(Prometec, 2016)

En el desarrollo del proyecto, con la mayoría de los elementos listos para la integración, el compañero de investigación encargado de la parte mecánica manifiesta que el diseño final del humanoide constará de 18 servos y no de

16, previamente el diseño fue pensado para un máximo de 16 servos, por lo que se replanteó el funcionamiento para cubrir este requerimiento. De forma que se hizo indispensable el uso de la comunicación I2C con la ventaja de que la placa que controla los servos presenta este tipo de enlace, se adaptó con facilidad otra placa que trabaja en distinto canal, pero de forma simultánea con la placa arduino gracias a esta comunicación. De hecho, se puede fácilmente adaptar hasta un número máximo de 16 placas; esto quiere decir que se podrían controlar hasta 256 motores funcionando todos a la vez. Para cubrir la necesidad se necesitaron solo 2 placas que permiten conectar hasta 32 servomotores, de los cuales solo se ocuparon 18.

#### **4.1.2.4 Comunicación entre Raspberry Pi y Arduino**

Para alcanzar los objetivos de autonomía y comunicación inalámbrica, se realiza una comunicación entre Raspberry y Arduino en la cual se empleó el siguiente hardware:

- Raspberry Pi 2 con Raspbian como sistema operativo.
- Arduino nano V3.
- Cable serial.

Cabe señalar como indica Llamas, L. (2014) que prácticamente todas las placas Arduino disponen al menos de una unidad «UART» (universally asynchronous receiver/transmitter), éste es un componente que incorporan ciertos procesadores, que tiene la misión de realizar la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada. Las placas Arduino UNO y Mini Pro disponen de una unidad UART que operan a nivel TTL 0V / 5V, por lo que son directamente compatibles con la conexión USB.

Los puertos serie están físicamente unidos a distintos pines de la placa Arduino; por lo que, mientras se usan los puertos de serie no se pueden utilizar

como entradas o salidas digitales los pines asociados con el puerto de serie en uso. En el presente proyecto se utilizan los siguientes pines:

- 30 GND
- GND
- 5V
- SLC
- SDA
- 3.3V
- Pin 6

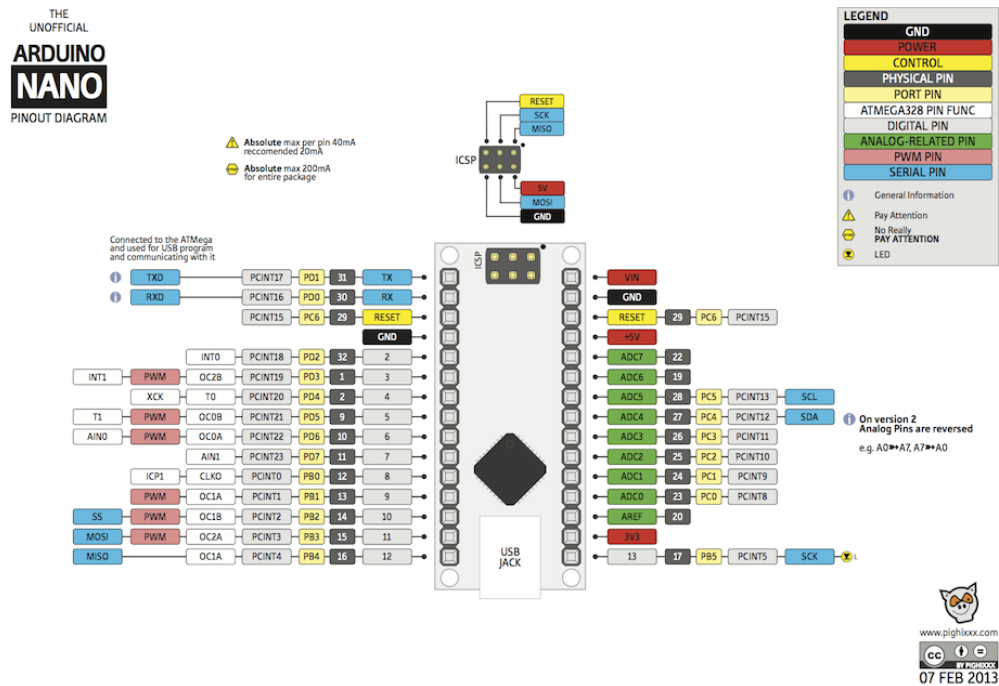
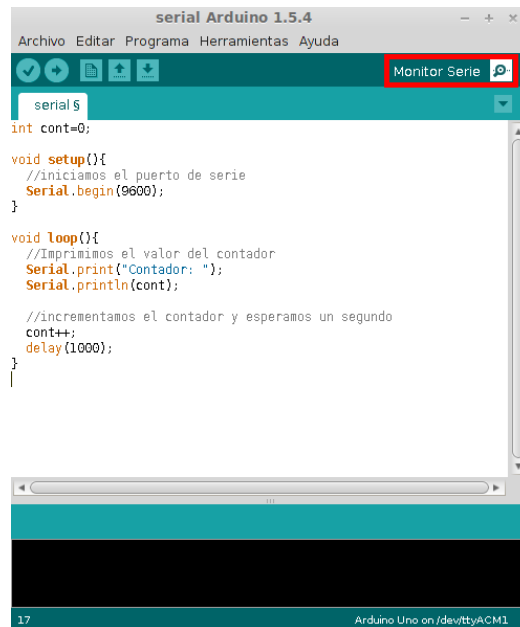


Figura 10. Pines de Arduino Nano (Patagoniatec, 2016)

Igualmente, muchos modelos de placas Arduino disponen de un conector USB o Micro USB conectado a uno de los puertos de serie, lo que simplifica el proceso de conexión con un ordenador. En el caso particular de la placa usada en este diseño tiene una conexión Micro USB que se observa en la Figura 10.

Para realizar la conexión mediante puerto serie, solo es necesario conectar la placa Arduino empleando el mismo puerto que empleamos para programarlo. Posteriormente se abre el IDE Standard de Arduino y se pulsa donde indica «Monitor Serial» como se muestra en la figura 11:



**Figura 11.** Comunicación Arduino - Raspberry Pi  
(Llamas, 2014)

El monitor de puerto serie es una herramienta integrada dentro de IDE Standard que permite enviar y recibir información a través del puerto serie; su uso es muy simple, y dispone de dos zonas, una que muestra los datos enviados, y otra para los recibidos; estas zonas se muestran en la figura 12:



**Figura 12.** Monitor de Puerto Serie  
(Llamas, 2014)

#### 4.1.3 FUNCIONAMIENTO DE ADICIONALES COMO GIROSCOPIO

La facilidad que brinda trabajar con el bus de datos I2C, que soporta hasta 256 distintos tipos de dispositivos y que pueden ser transferidos desde otra placa Arduino, así como cualquier componente que trabaje con esta clase de comunicación. Se procedió a integrar la función del giroscopio por medio del módulo sensor InvenSense MPU2650, el cual puede ser apreciado en la figura 18 y que según la web de Arduino (2016) tiene las siguientes características: a) contiene tanto un acelerómetro MEMS como un giroscopio MEMS en un solo chip; b) es muy preciso ya que contiene hardware de 16 bits de conversión análoga a digital para cada canal, por lo que captura los canales X, Y y Z al mismo tiempo; c) el sensor usa el bus de datos I2C para interactuar con Arduino; y d) es económico, tomando en cuenta que combina giroscopio y acelerómetro. La figura 13 muestra el sensor usado:



**Figura 13.** InvenSense MPU2650  
(Robologs, 2014)

El giroscopio mide la velocidad angular; la cual es el número de grados que se gira en un segundo. Si se conoce el ángulo inicial de la unidad, se puede sumar el valor que marca el giroscopio para saber el nuevo ángulo a cada momento. Por ejemplo, si se inicia la unidad a  $0^{\circ}$ ; y si el giroscopio realiza una



medida cada segundo y marca 3 en el eje X, tendremos el ángulo con esta sencilla fórmula:

$$\text{Ángulo}Y = \text{Ángulo}Y \text{ Anterior} + \text{Giroscopio}Y * \Delta t \quad [9]$$

Donde  $\Delta t$  es el tiempo que transcurre cuando se realiza esta operación, *ÁnguloYAnterior* es el ángulo calculado la última vez que se llamó esta fórmula y *GiroscopioY* es la lectura del ángulo Y del giroscopio. Lo mismo pasa con los ejes X y Z; aunque se suele ignorar el eje Z, puesto que al no poder calcular un ángulo Z con el Acelerómetro, no se puede aplicar un Filtro Complementario para este eje.

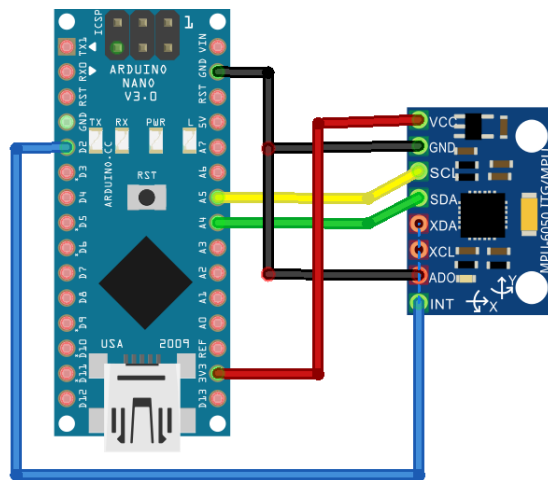
Sin embargo, existen dos problemas muy importantes: el ruido y los errores; el primero consiste en todas aquellas interferencias que afectan a los dispositivos electrónicos, en tanto que el segundo se refiere a que cuando el giroscopio realiza los cálculos del ángulo es inevitable que se produzca un pequeño error, que con el tiempo va acumulándose; esto se llama *drift*. Para eliminar el ruido y el drift hay distintos algoritmos llamados filtros que hacen esta tarea. Un ejemplo es el conocido como Filtro Complementario (*Complementary Filter*); que es ideal para implementar con Arduino, porque es fácil de utilizar, tiene bajo coste de procesamiento y presenta una precisión muy buena.

Este filtro, es en realidad una unión de dos filtros diferentes: un *High-pass Filter* para el giroscopio y un *Low-pass Filter* para el Acelerómetro. El primero deja pasar únicamente los valores por encima de un cierto límite, al contrario que el Low-pass filter, que sólo permite a los que están por debajo. La fórmula resultante de combinar (o complementar) ambos filtros es:

$$\text{Ángulo} = 0,98 * (\text{Ángulo} + \text{Ángulo}Gyro * \Delta t) + 0,02 * \text{Ángulo}Accel \quad [10]$$

Donde  $\text{ÁnguloGyro}$  es el ángulo del Giroscopio que se ha calculado previamente, y  $\text{ÁnguloAccel}$  con el ángulo del Acelerómetro calculado con la fórmula de la tangente.  $\Delta t$  es el tiempo que ha pasado (en segundos) desde la última vez que se ha calculado el filtro. Esta fórmula es la misma para el eje X, Y. En caso necesario, se puede cambiar el valor de 0.98 y 0.02 por un valor personalizado; eso sí, ambos tienen que sumar 1.

La conexión de la MPU-6050 con el Arduino es como se ve en la figura 14:



**Figura 14.** Conexión MPU - Arduino  
(Mianbao, 2016)

En donde:

- MPU Vcc -> Arduino 3.3v (o 3v3 en algunos modelos)
- MPU Gnd -> Arduino Gnd
- MPU SCL -> Arduino A5
- MPU SDA -> Arduino A4

#### 4.1.4 LIBRERÍAS UTILIZADAS

Para el correcto funcionamiento del diseño, tanto de las placas como de adicionales (giroscopio) se deben instalar las siguientes librerías:

- <Adafruit\_PWMServoDriver.h>
- <Servo.h>
- <Wire.h>
- <I2Cdev.h>
- <MPU6050.h>

#### 4.1.5 DIAGRAMA DE CONEXIÓN

La figura 15 muestra el Diagrama de conexión del diseño electrónico, el cual fue realizado en Fritzing:

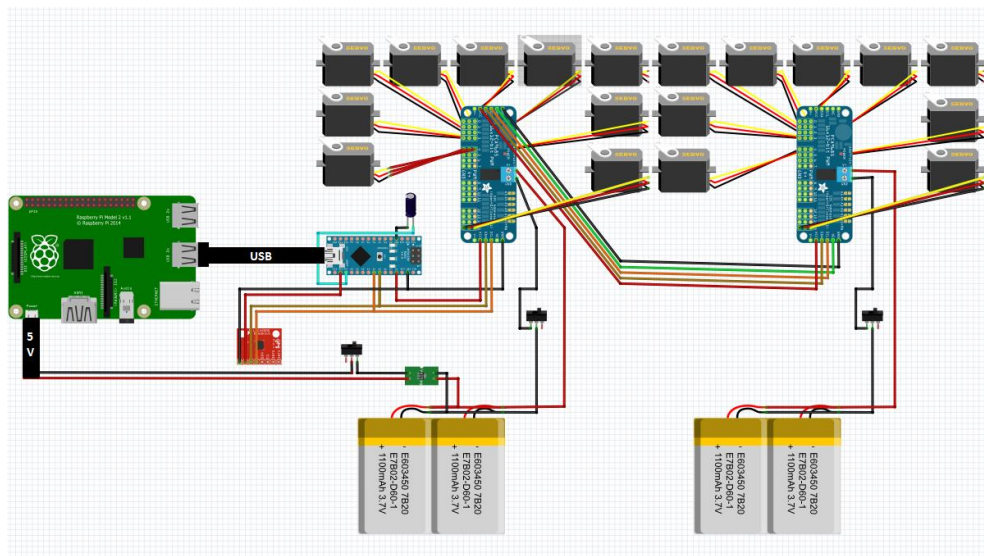


Figura 15. Diagrama de Conexión

#### 4.1.6 ORGANIZACIÓN DE CABLEADO PARA LOS ACTUADORES, PLACAS Y SENSORES

En la figura 16 se observa el Diseño en Proteus, organizador de cableado para baterías, sensores y actuadores del Humanoide.

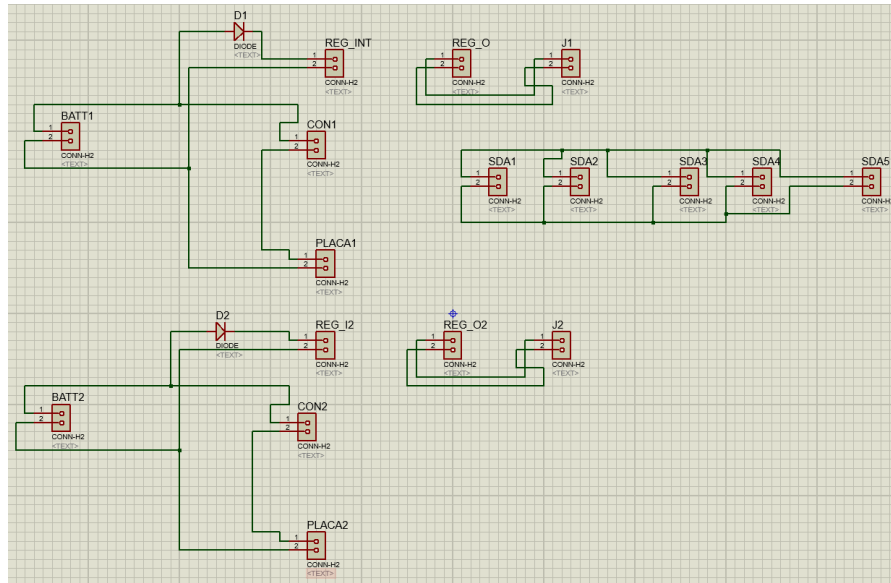


Figura 16. Organización de Cableado

Adicionalmente, en la figura 17 se observa el Diseño de PCB realizado en ARES (Proteus)

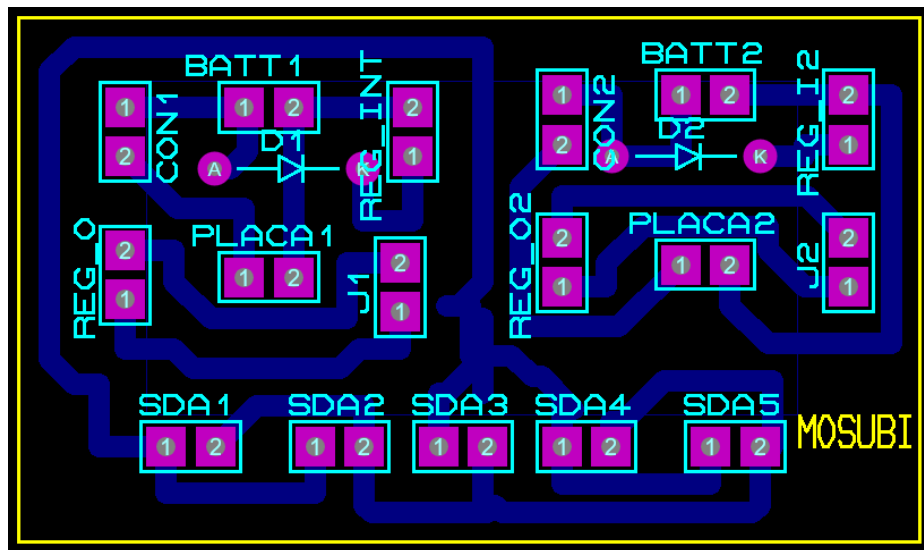


Figura 17. Diseño de PCB

La leyenda de la figura 17 es la siguiente:

- BATT1: Batería LiPo 1
- BATT2: Batería LiPo 2
- REG\_INT: Regulador de voltaje 1
- REG\_I2: Regulador de voltaje 2
- J1 y J2: Tomas directas de cada regulador

- CON1: Interruptor 1
- CON2: Interruptor 2
- SDA: Conexiones para distintos sensores o placas que trabajan con I2C

#### 4.1.7 ANÁLISIS DE MARCHA DEL HUMANOIDE

La marcha normal humana implica la interacción de múltiples sistemas y segmentos corporales; uno de los más importantes es el musculoesquelético: En este proyecto se busca emular este movimiento, pero se tienen limitaciones evidentes derivadas del hecho de que el robot no cuenta con un sistema idéntico al humano, por ejemplo, no se tienen músculos ni otros elementos que intervienen. De manera que la estabilización, el equilibrio y la marcha se logran a través del ensayo y diferentes pruebas, hasta que se logra que los servos y el cuerpo del robot cumplan la función análoga al sistema musculoesquelético y el humanoide pueda permanecer de pie y desplazarse. En este sentido, una característica resaltante es que para cada paso el robot soporta su centro de gravedad en la pierna de apoyo, de esta manera puede alzar la pierna contraria manteniendo el balance y generar la zancada.

##### 4.1.7.1 Lógica de movimientos del humanoide

Para realizar el desplazamiento del humanoide se analizaron los diferentes movimientos de una persona real en conjunto con las restricciones que presenta el diseño mecánico. Los cuatro movimientos que se estudiaron fueron: caminar adelante, caminar atrás, derecha, izquierda, correr. En las siguientes figuras se muestra la definición y declaración de los pines para cada servo, se nombró con partes del cuerpo humano para su distinción y se separó en eje izquierdo y derecho, como se muestra en la Figura 18 y 19.

```
#define SERVOMIN 150 //valor mínimo para ancho de pulso (4096)
#define SERVOMED 350
#define SERVOMAX 550 //valor máximo para ancho de pulso (4096)
```

**Figura 18.** Definición de valores

```

//declaracion de pines para cada servo

uint8_t ihombro1 = 12;
uint8_t icadera = 13;
uint8_t ihombro2 = 14; // servos pierna derecha 7.4V
uint8_t icodo = 15;
uint8_t imuslo2 = 4;
uint8_t imuslo1 = 5;
uint8_t irodilla = 6;
uint8_t ipie2 = 7;
uint8_t ipie1 = 8;

uint8_t dhombro1 = 12;
uint8_t dcadera = 13;
uint8_t dhombro2 = 14; // servos pierna derecha 7.4V
uint8_t dcodo = 15;
uint8_t dmuslo2 = 4;
uint8_t dmuslo1 = 5;
uint8_t drodilla = 6;
uint8_t dpi2 = 7;
uint8_t dpi1 = 8;

```

**Figura 19.** Declaración de pines por servo

Para controlar el movimiento de los servos se estableció como función principal en de la programación en Arduino un rango en los ángulos de giro desde 0 a 90 grados; en la figura 20 se observa el código usado:

```

angle1 = map(iiihombro1, 90, -90, SERVOMIN, SERVOMAX);
ihombro1 = (angle1);
pwml.setPWM(ihombro1, 0, ihombro1);

angle2 = map(iiicadera, -90, 90, SERVOMIN, SERVOMAX);
icadera = (angle2);
pwml.setPWM(icadera, 0, icadera);

angle3 = map(iiihombro2, 90, -90, SERVOMIN, SERVOMAX);
ihombro2 = (angle3);
pwml.setPWM(ihombro2, 0, ihombro2);

angle4 = map(iiicodo, 90, -90, SERVOMIN, SERVOMAX);
icodo = (angle4);
pwml.setPWM(icodo, 0, icodo);

angle5 = map(iiimuslo2, -90, 90, SERVOMIN, SERVOMAX);
imuslo2 = (angle5);
pwml.setPWM(imuslo2, 0, imuslo2);

angle6 = map(iiimuslo1, -90, 90, SERVOMIN, SERVOMAX);
imuslo1 = (angle6);
pwml.setPWM(imuslo1, 0, imuslo1);

angle7 = map(iiiodilla, 90, -90, SERVOMIN, SERVOMAX);
irodilla = (angle7);
pwml.setPWM(irodilla, 0, irodilla);

angle8 = map(iiipie2, -90, 90, SERVOMIN, SERVOMAX);
ipie2 = (angle8);
pwml.setPWM(ipie2, 0, ipie2);

```

**Figura 20.** Código para control de ángulos de lado izquierdo

Para realizar los movimientos del humanoide se comenzó con la definición de una posición inicial y la respectiva configuración de los servos, de forma que en esta posición inicial sus valores sean cero o cercanos a cero. En la figura 30 se observa el código y la posición del robot.

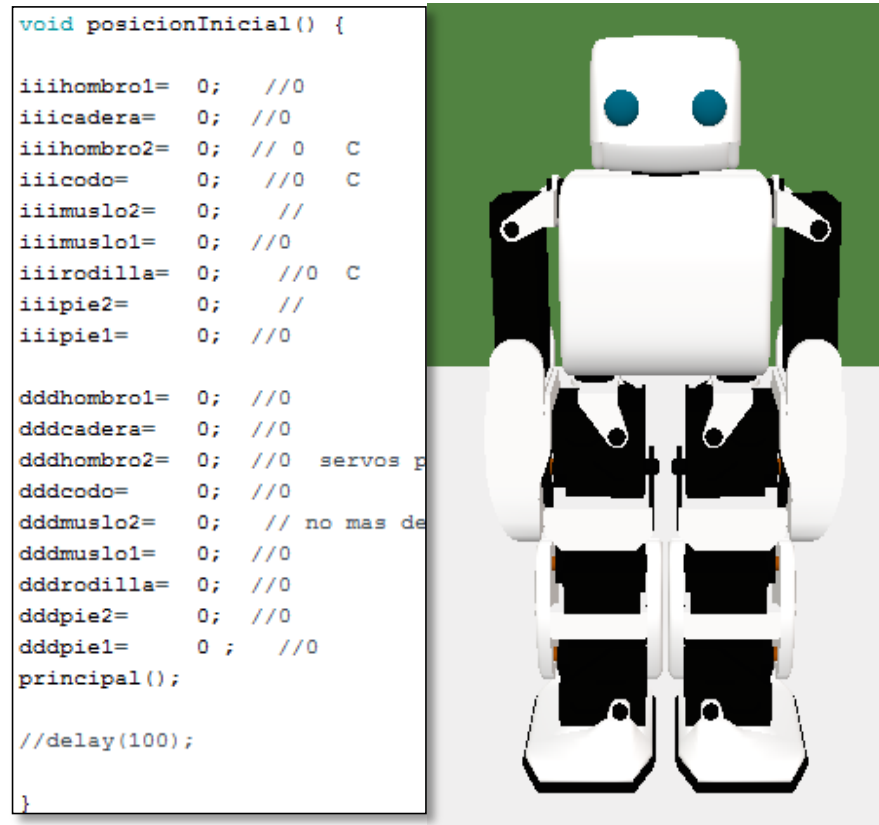


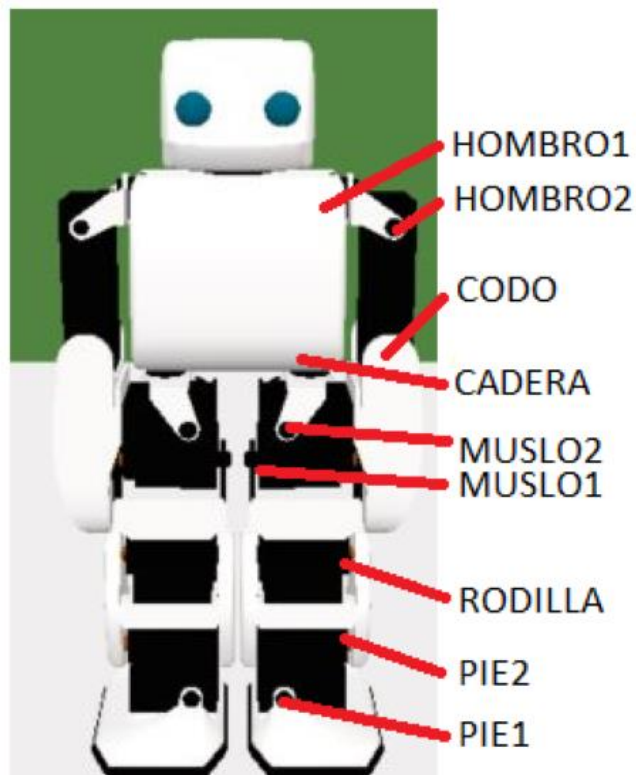
Figura 21. Posición inicial

Luego, se modifican los valores de los ángulos de cada servo para generar el movimiento; aquí, se toma en cuenta el análisis de marcha de una persona real y se busca imitar los movimientos con el humanoide.

La especificación del movimiento que realiza cada servo para generar los ángulos es como se evidencia en tabla 8, se comienza con el lado izquierdo del humanoide y después con el lado derecho. Asimismo, en la figura 22 se puede observar la correspondencia de estos elementos con las partes físicas del humanoide.

**Tabla 8.** Lista de definición de partes

Lado izquierdo	Lado derecho
iiihombro1 = hombro1	dddhombro1 = hombro1
iiicadera = cadera	dddcadera = cadera
iiihombro2 = hombro2	dddhombro2 = hombro2
iiicodo = codo	dddcodo = codo
iiimuslo2 = muslo2	dddmuslo2 = muslo2
iiimuslo1 = muslo1	dddmuslo1 = muslo1
iiirodilla = rodilla	dddrodilla = rodilla
iiipie2 = pie2	dddpie2 = pie2
iiipie1 = pie1	dddpie1 = pie1



**Figura 22.** Señalización de partes del humanoide

#### 4.1.7.2 Estudio de movimientos: Marcha normal hacia adelante.

Dentro del proyecto, este desplazamiento consta principalmente de ocho movimientos, de los cuales seis están configurados en una secuencia, como se describe a continuación.



## 1. Posición inicial

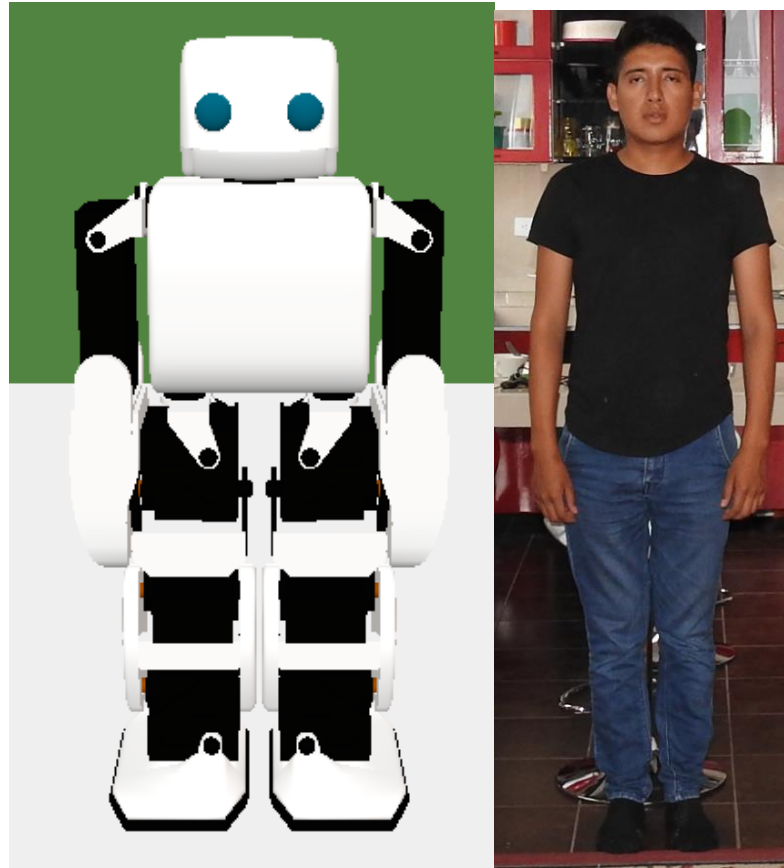


Figura 23. Modelo y posición inicial

Tabla 9. Posición de los ángulos

Lado izquierdo del humanoide		Lado derecho del humanoide	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= 0	hombro2	= 0
codo	= 0	codo	= 0
muslo2	= 0	muslo2	= 0
muslo1	= 0	muslo1	= 0
rodilla	= 0	rodilla	= 0
pie2	= 0	pie2	= 0
pie1	= 0	pie1	= 0

Por medio de la figura 23 y de la tabla 9 se evidencia el estado de la posición inicial del robot. Se puede observar en la imagen que ambos se encuentran en una posición firme y lista para dar comienzo a la marcha.

## 2. Primer paso

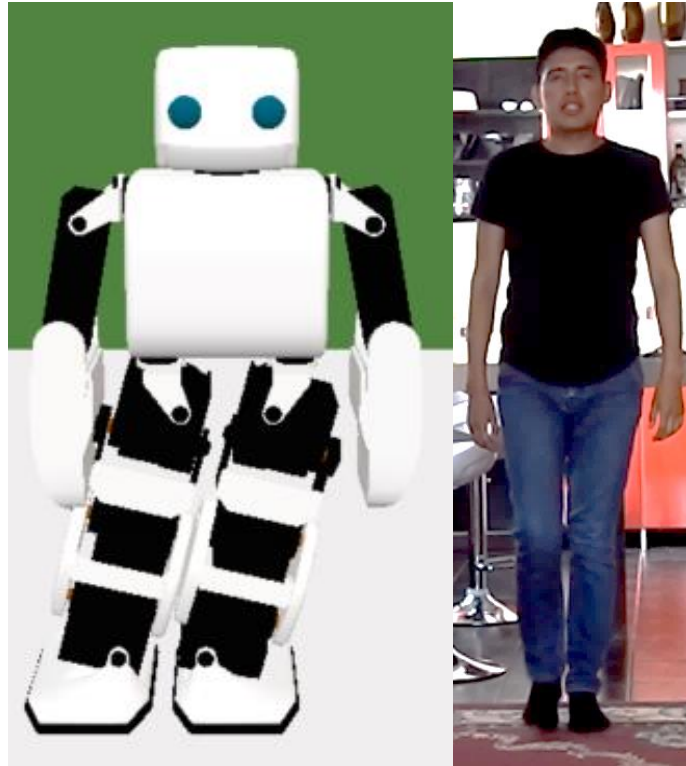


Figura 24. Modelo y primer paso

Tabla 10. Posición de los ángulos para primer paso

Lado izquierdo		Lado derecho	
hombro1	= -1	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 14	muslo2	= 14
muslo1	= 0	muslo1	= 0
rodilla	= 0	rodilla	= 0
pie2	= 0	pie2	= 0
pie1	= 14	pie1	= 14

Observando la figura 24 se puede evidenciar como el ser humano por instinto y cualidades biomecánicas no necesita soportar totalmente el cuerpo en la pierna que hará de apoyo, ya que los músculos y sistemas involucrados hacen el trabajo para estabilizar el cuerpo, mientras que en el caso del humanoide es necesario hacer el soporte completo del centro de gravedad en la pierna de apoyo. En la tabla 10 se describe la angulación de los servos para generar este movimiento.

### 3. Segundo paso

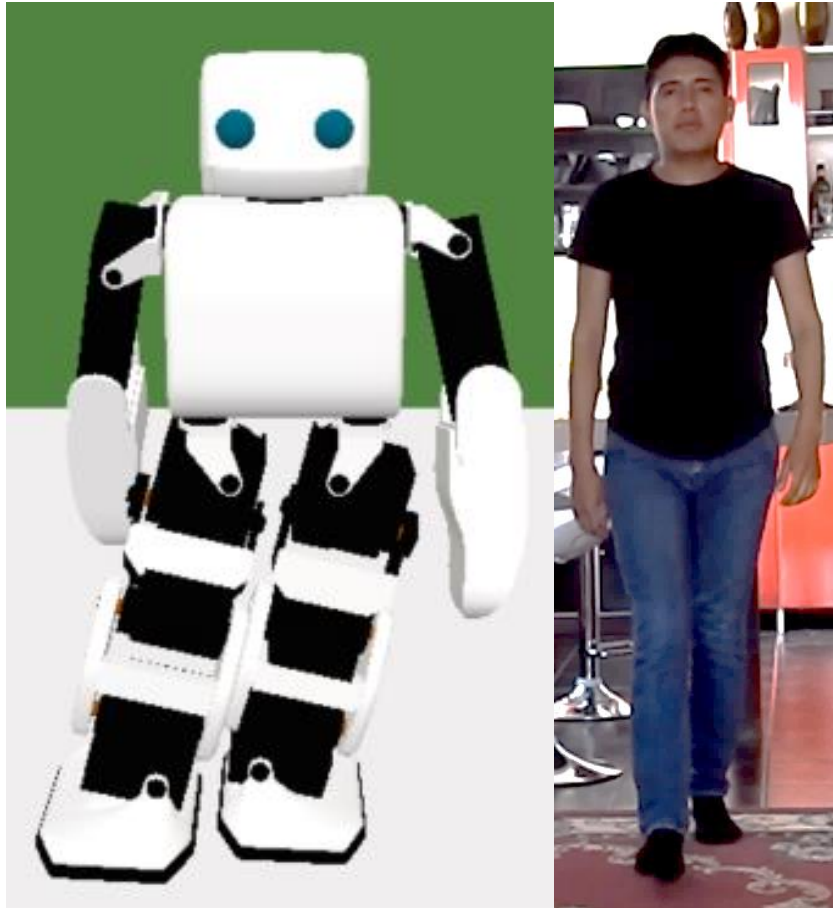


Figura 25. Modelo y segundo paso

Tabla 11. Posición de los ángulos para segundo paso

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.90
muslo2	= 7	muslo2	= 15
muslo1	= 0	muslo1	= -26.8
rodilla	= 0	rodilla	= 60.5
pie2	= 0	pie2	= 29.4
pie1	= 20	pie1	= 14

En la figura 25 se puede evidenciar la forma en que la pierna derecha tanto del modelo como del humanoide adelantan a la pierna izquierda para realizar el desplazamiento. Asimismo, en la tabla 11 se describe la posición de los ángulos para ejecutar este movimiento.

#### 4. Tercer paso

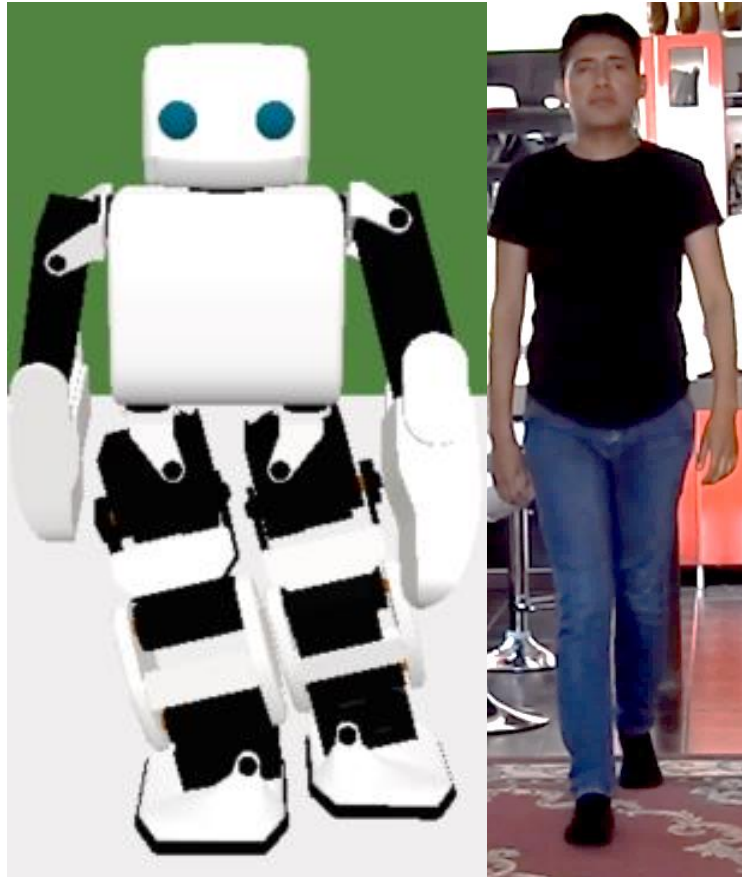


Figura 26. Modelo y tercer paso

Tabla 12. Posición de los ángulos para tercer paso

Lado izquierdo		Lado derecho	
hombro1	= 14.2	hombro1	= 18.5
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 7	muslo2	= 14
muslo1	= 0	muslo1	= -38.5
rodilla	= 0	rodilla	= 40.2
pie2	= 0	pie2	= 2.7
pie1	= 20	pie1	= 10.2

En la figura 26 se puede evidenciar como la pierna derecha se prepara para hacer de apoyo, igualmente se resalta que las posiciones de los brazos del robot emulan la oscilación que realiza el modelo. Igualmente, en la tabla 12 se describe la posición de los ángulos para ejecutar este movimiento.

## 5. Cuarto paso

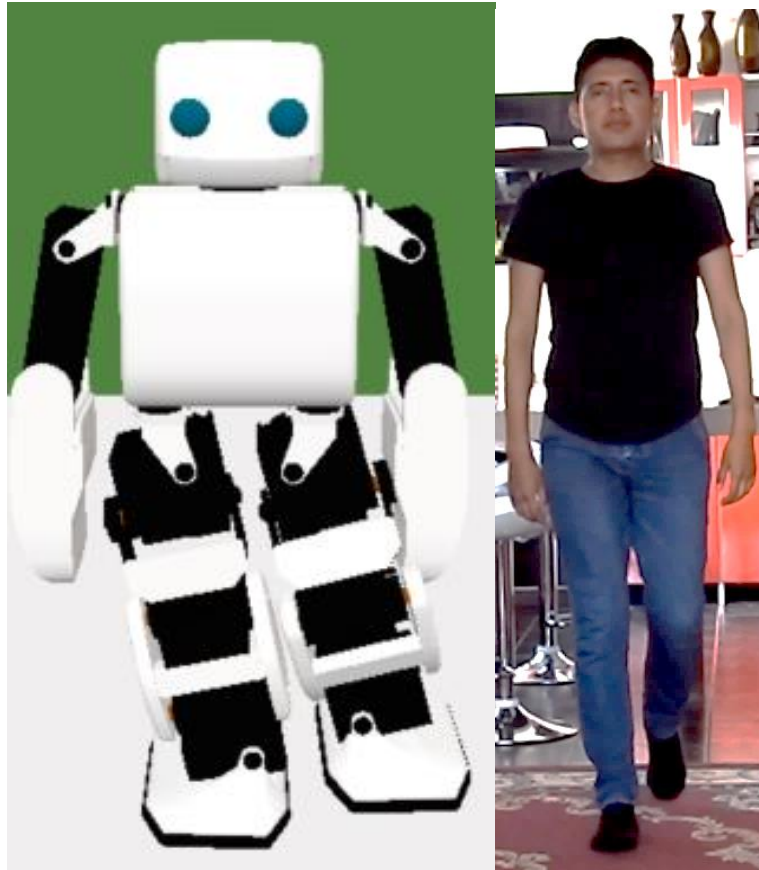


Figura 27. Modelo y cuarto paso

Tabla 13. Posición de los ángulos para cuarto paso

Lado izquierdo		Lado derecho	
hombro1	= 28.1	hombro1	= 14.7
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= -14.0	muslo2	= -14
muslo1	= 1.5	muslo1	= -25.5
rodilla	= -16.5	rodilla	= 29.8
pie2	= -22.7	pie2	= 9.6
pie1	= -14	pie1	= -14

En la Figura 27 se observa que la pierna izquierda se levanta para continuar con el desplazamiento. Además, en la tabla 13 se describe la posición de los ángulos para ejecutar este movimiento.

## 6. Quinto paso

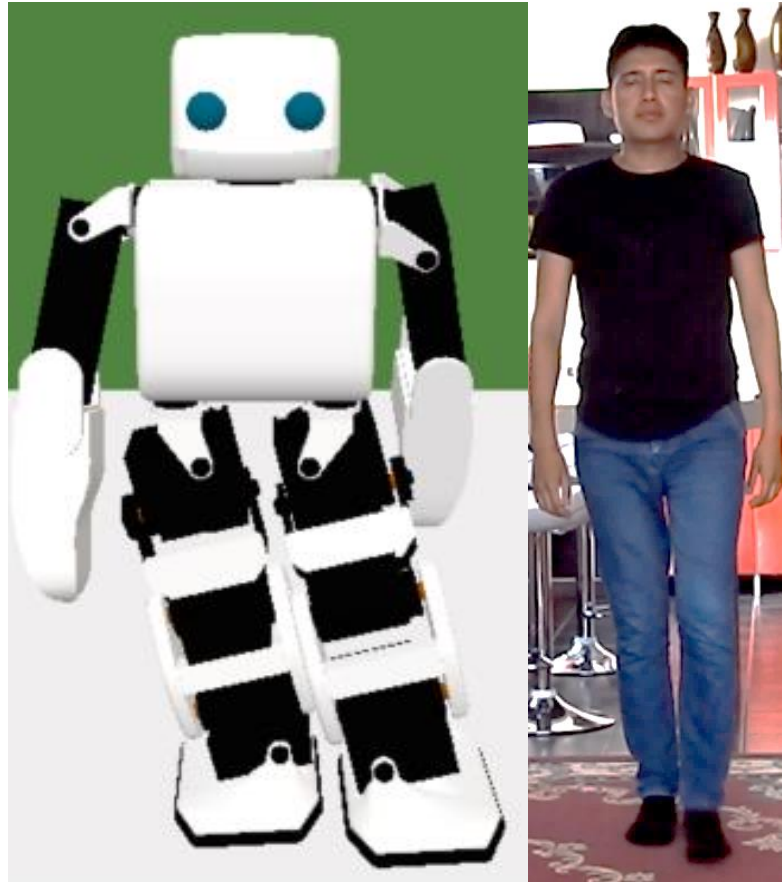


Figura 28. Modelo y quinto paso

Tabla 14. Posición de los ángulos para quinto paso

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= -15	muslo2	= -7
muslo1	= 26.8	muslo1	= 0
rodilla	= -60.5	rodilla	= 0
pie2	= -29.4	pie2	= 0
pie1	= -14	pie1	= -20

En la Figura 28 se puede evidenciar como la pierna izquierda se asienta para hacer de apoyo, asimismo las posiciones de los brazos del robot mantienen la emulación la oscilación que realiza el modelo. Igualmente, en la tabla 14 se describe la posición de los ángulos para ejecutar este movimiento.

## 7. Sexto paso

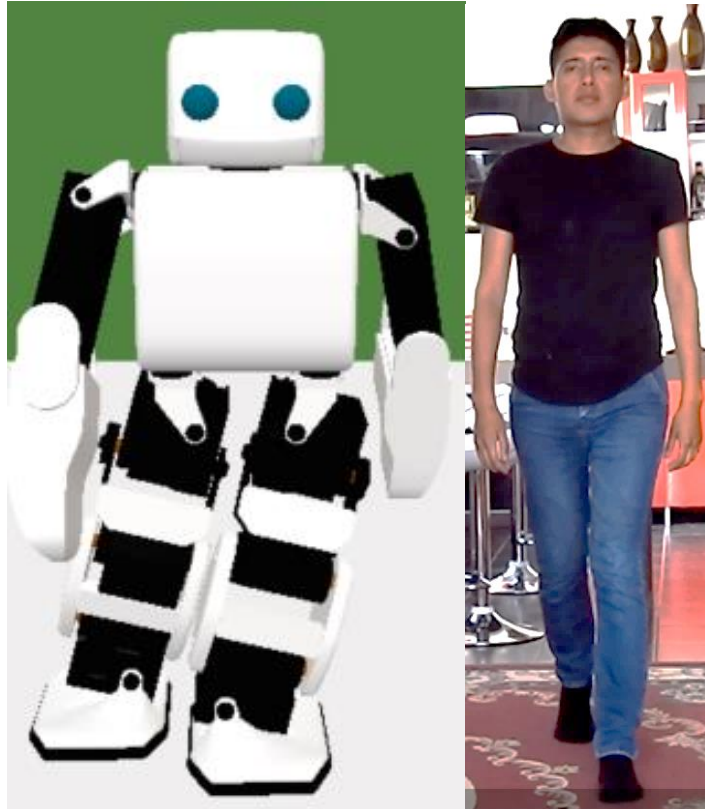


Figura 29. Modelo y sexto paso

Tabla 15. Posición de los ángulos para sexto paso

Lado izquierdo		Lado derecho	
hombro1	= -18.5	hombro1	= -14.2
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= -14	muslo2	= -7
muslo1	= 38.5	muslo1	= 0
rodilla	= -40.2	rodilla	= 0
pie2	= -2.7	pie2	= 0
pie1	= -10.2	pie1	= -20

En la Figura 29 se evidencia que la pierna izquierda sirve de apoyo, mientras que la pierna derecha se levanta para hacer el desplazamiento. También, en la tabla 15 se describe la posición de los ángulos para ejecutar este movimiento.

## 8. Séptimo paso

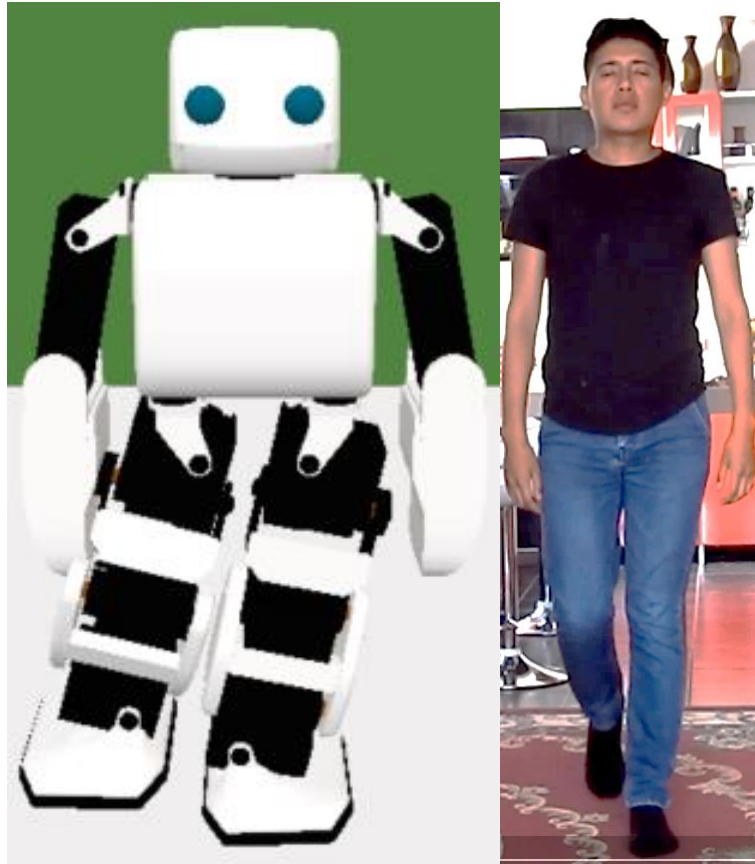


Figura 30. Modelo y séptimo paso

Tabla 16. Posición de los ángulos para séptimo paso

Lado izquierdo		Lado derecho	
hombro1	= -14.7	hombro1	= -28.1
cadera	= 0	cadera	= -0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 14	muslo2	= 14
muslo1	= 25.5	muslo1	= -1.5
rodilla	= -29.8	rodilla	= 16.5
pie2	= -9.6	pie2	= 22.7
pie1	= 14	pie1	= 14

En la Figura 30 se puede ver que la pierna izquierda continúa como apoyo, en tanto que la pierna derecha sigue hacia adelante para mantener el movimiento. Asimismo, en la tabla 16 se describe la posición de los ángulos para ejecutar este desplazamiento.



### Secuencia de la marcha

A continuación se presentan una serie de figuras dónde se muestra el modelo de la secuencia completa de la marcha del humanoide. En ella al principio el robot se encuentra estático y erguido en la posición inicial; luego apoya el centro de gravedad en la pierna izquierda, lo que permite que la pierna derecha se alce y pueda desplazarse; después el pie derecho se adelanta con respecto al izquierdo manteniendo el equilibrio; posteriormente el centro de gravedad se apoya en el pie derecho y el izquierdo queda en libertad para desplazarse (el movimiento es especular, es decir, que en los turnos lo que hace un eje luego lo realiza el otro); luego la pierna derecha vuelve a avanzar con respecto a la izquierda. De aquí en adelante se entra en bucle que pasará de la cadena intermedia a la cadena final hasta que alguna acción finalice este bucle y devuelva al robot a la posición inicial.

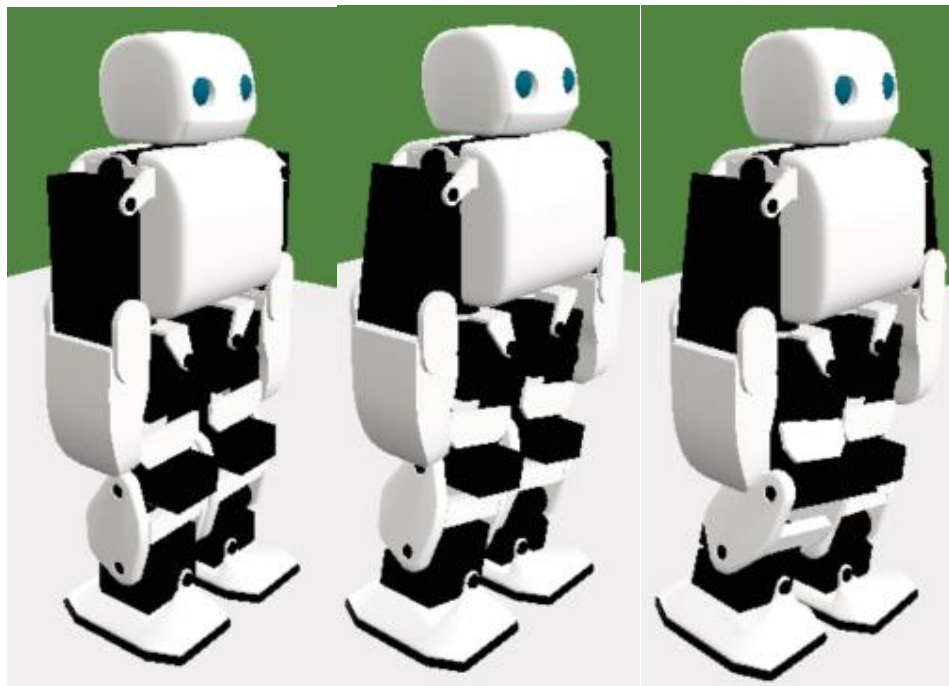
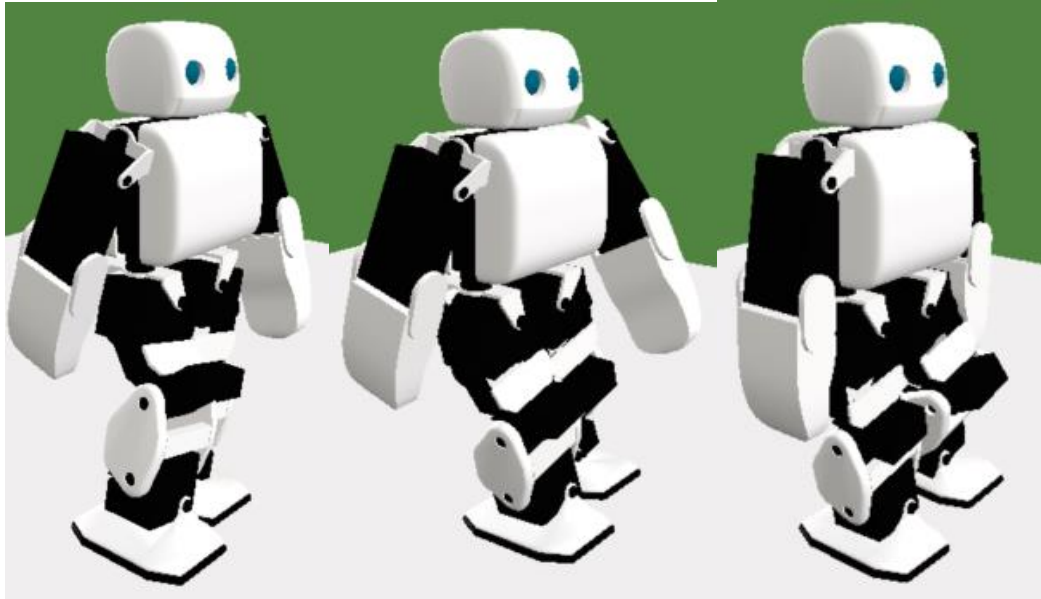
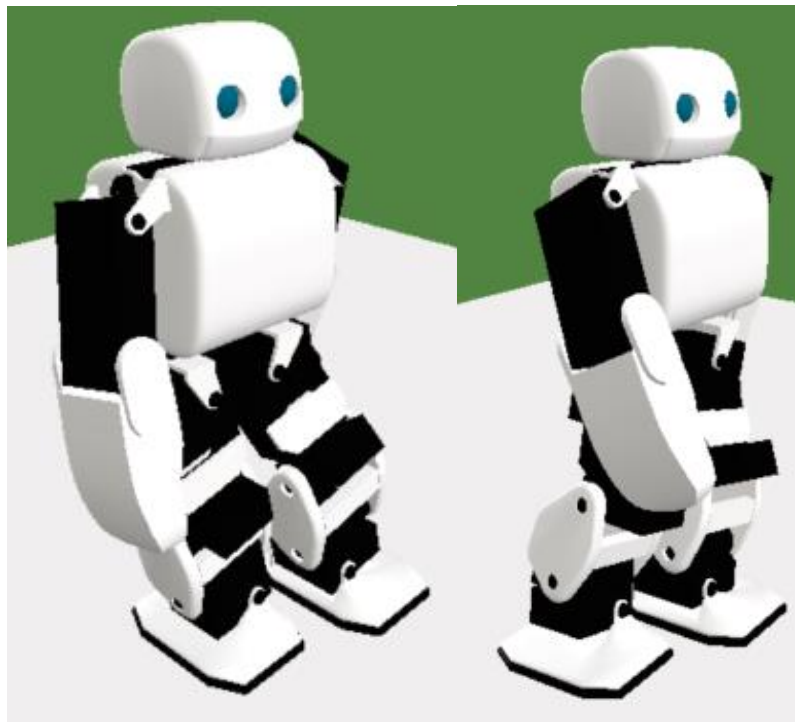


Figura 31. Secuencia inicial de marcha



**Figura 32.** Secuencia intermedia de marcha

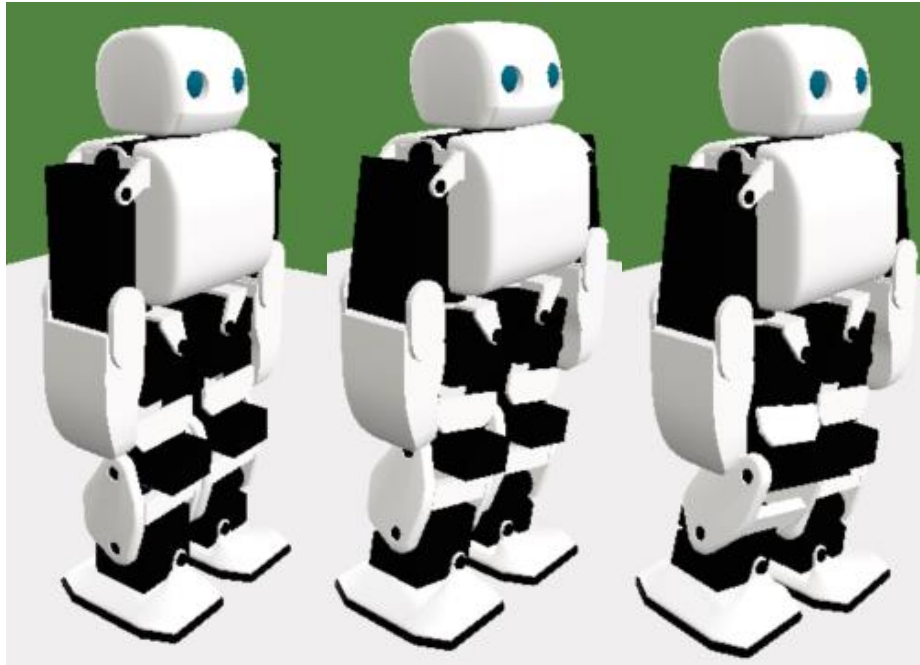


**Figura 33.** Secuencia final de marcha

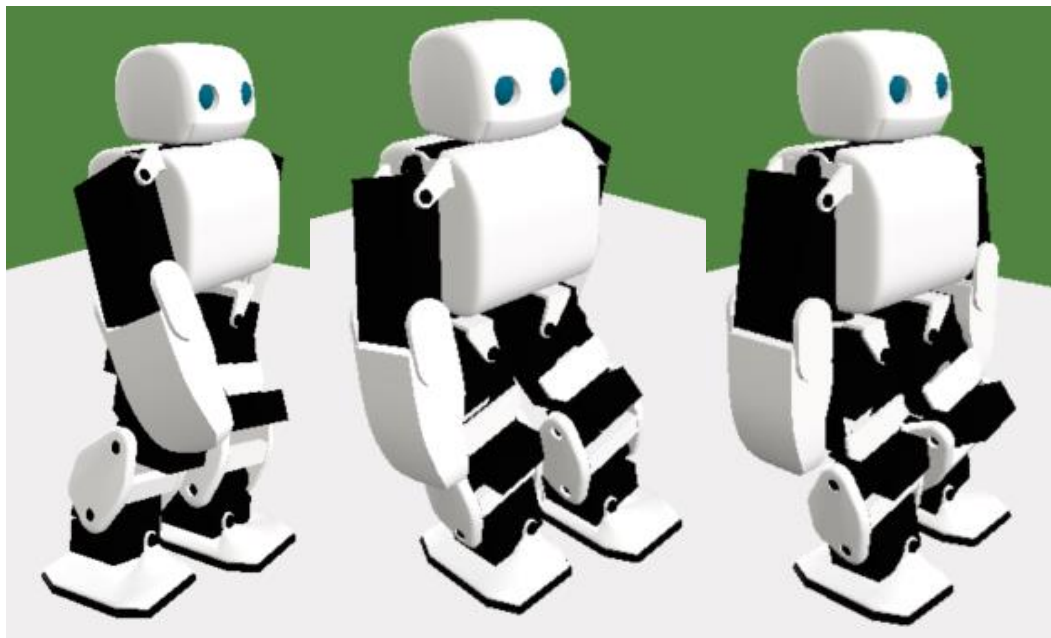
#### **4.1.7.3 Estudio de movimientos: Marcha normal hacia atrás.**

A continuación se presentan una serie de figuras donde se muestra el modelo de la secuencia completa de la marcha hacia atrás del humanoide. En esta

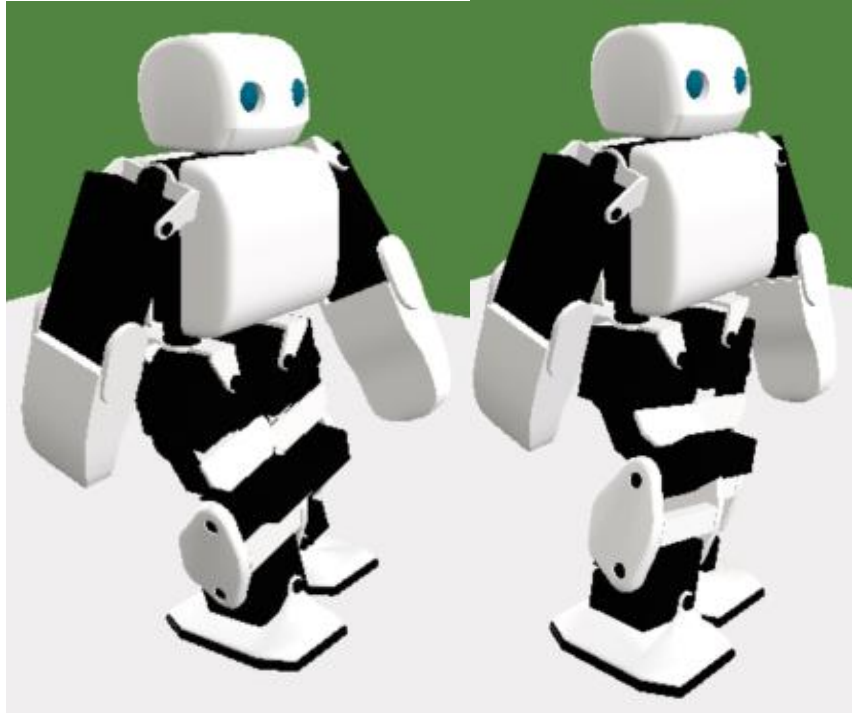
cadena se resalta que son los mismos movimientos que en la marcha hacia adelante con el orden invertido.



**Figura 34.** Secuencia inicial de marcha hacia atrás



**Figura 35.** Secuencia intermedia de marcha hacia atrás



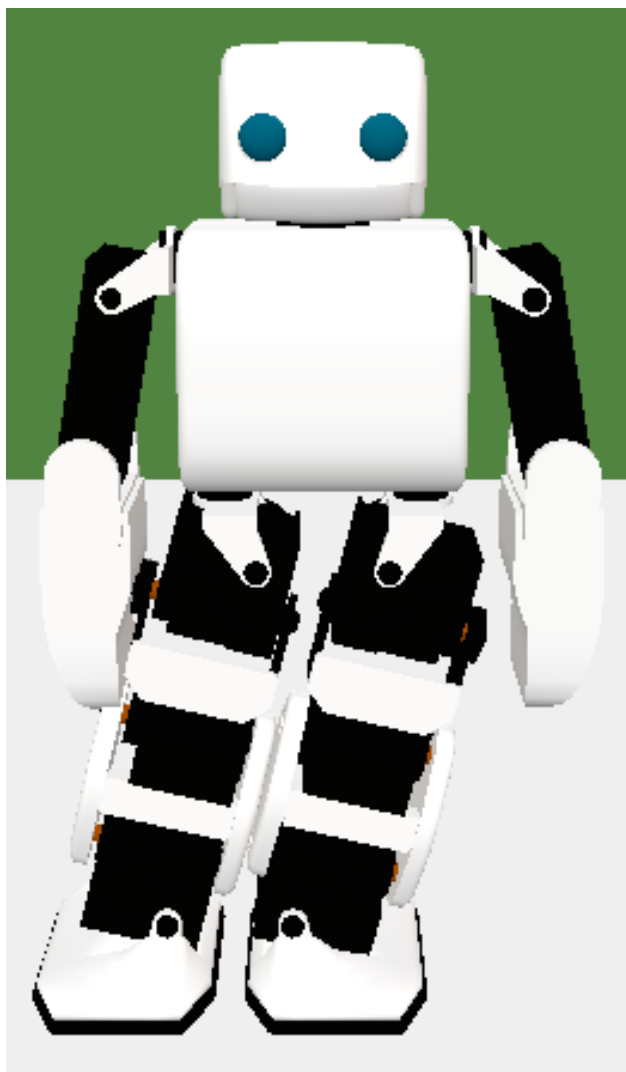
**Figura 36.** Secuencia final de marcha hacia atrás

Igualmente, el desplazamiento entra en un bucle entre la secuencia intermedia y la final hasta que alguna acción termine la ejecución y devuelva al humanoide a la posición inicial.

#### **4.1.7.4 Estudio de movimientos: Giro hacia la derecha**

Para la descripción de este desplazamiento se presenta a continuación la figura que representa a cada paso, como muestra del modelo del movimiento, mientras que la tabla de cada paso señala los valores de la posición de los ángulos para ejecutarlo (los ángulos son en grados sexagesimales).

## 1. Primer paso

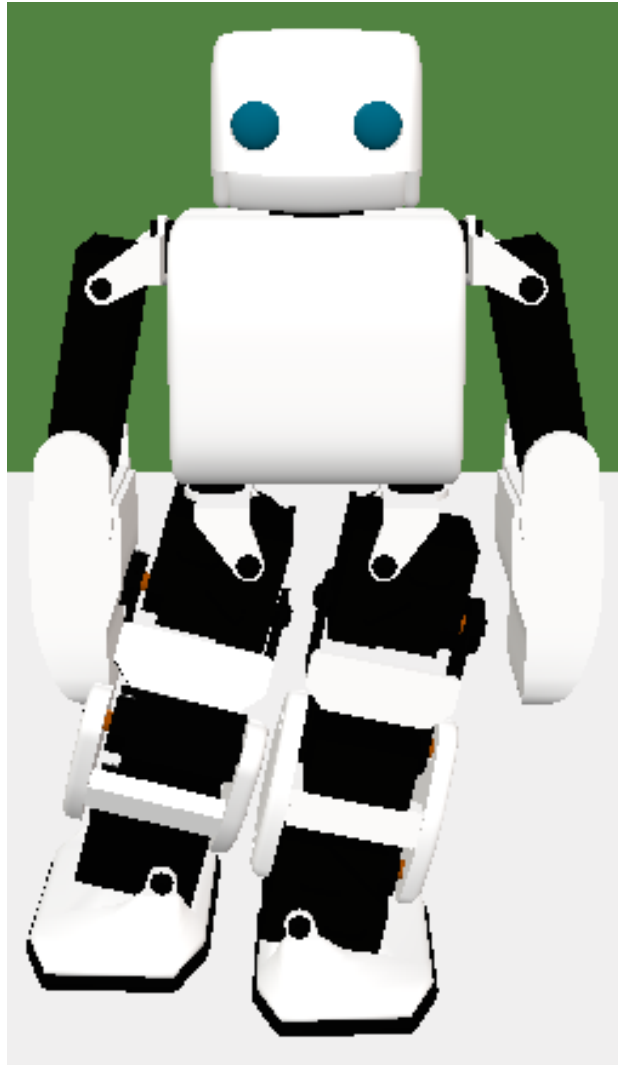


**Figura 37.** Inicio de giro

**Tabla 17.** Posición de los ángulos para inicio de giro

Lado izquierdo		Lado derecho	
hombro1	= -1.0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 14.0	muslo2	= 14.0
muslo1	= 0	muslo1	= 0
rodilla	= 0	rodilla	= 0
pie2	= 0	pie2	= 0
pie1	= 14.0	pie1	= 14.0

## 2. Segundo paso

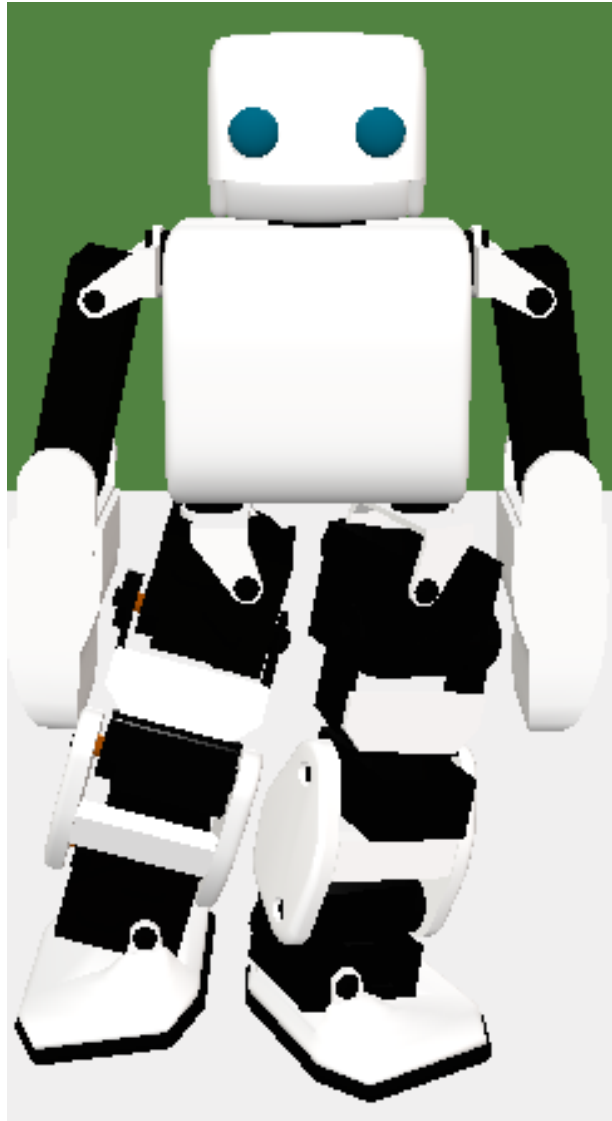


**Figura 38.** Segundo paso para girar

**Tabla 18.** Posición de los ángulos para segundo paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 12.0	muslo2	= 15.0
muslo1	= 0	muslo1	= -26.8
rodilla	= 0	rodilla	= 60.5
pie2	= 0	pie2	= 29.4
pie1	= 14.0	pie1	= 14.0

### 3. Tercer paso



**Figura 39.** Tercer paso para girar

**Tabla 19.** Posición de los ángulos para tercer paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 26.7	cadera	= 0
hombro2	= -8.6	hombro2	= 13.1
codo	= -8.0	codo	= 9.4
muslo2	= 12.0	muslo2	= 16.0
muslo1	= 0	muslo1	= -22.4
rodilla	= 0	rodilla	= 44.4
pie2	= -4.0	pie2	= 21.5
pie1	= 13.0	pie1	= 24.5

#### 4. Cuarto paso

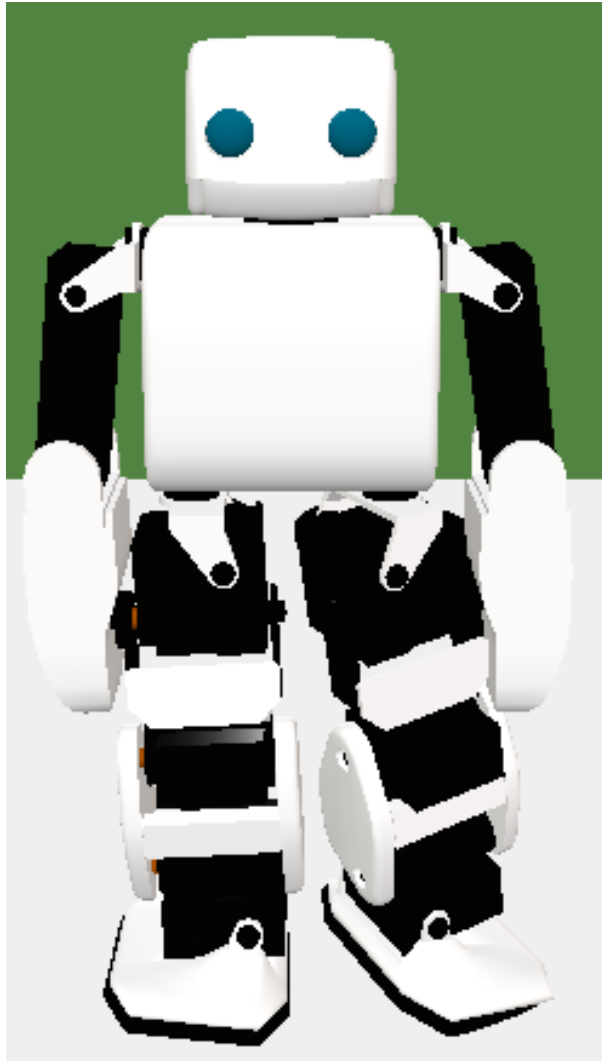


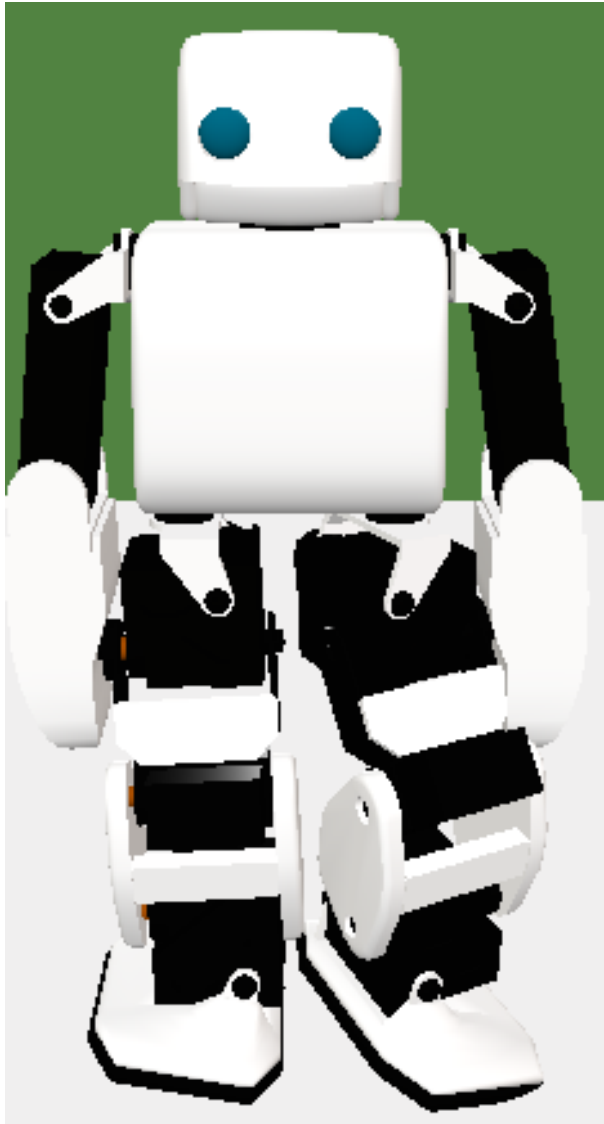
Figura 40. Cuarto paso para girar

Tabla 20. Posición de los ángulos para cuarto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 26.7	cadera	= 0
hombro2	= -8.6	hombro2	= 6.1
codo	= -8.0	codo	= 9.4
muslo2	= -8.7	muslo2	= -3.2
muslo1	= 0	muslo1	= -12.4
rodilla	= 0	rodilla	= 23.6
pie2	= 4.2	pie2	= 15.7
pie1	= -16.3	pie1	= -9.7



## 5. Quinto paso



**Figura 41.** Quinto paso para girar

**Tabla 21.** Posición de los ángulos para quinto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 26.7	cadera	= 0
hombro2	= -8.6	hombro2	= 6.1
codo	= -8.0	codo	= 9.4
muslo2	= -2.7	muslo2	= -3.2
muslo1	= 32.6	muslo1	= -12.4
rodilla	= -46.4	rodilla	= 23.6
pie2	= -4.9	pie2	= 13.7
pie1	= -16.3	pie1	= -13.7

## 6. Sexto paso

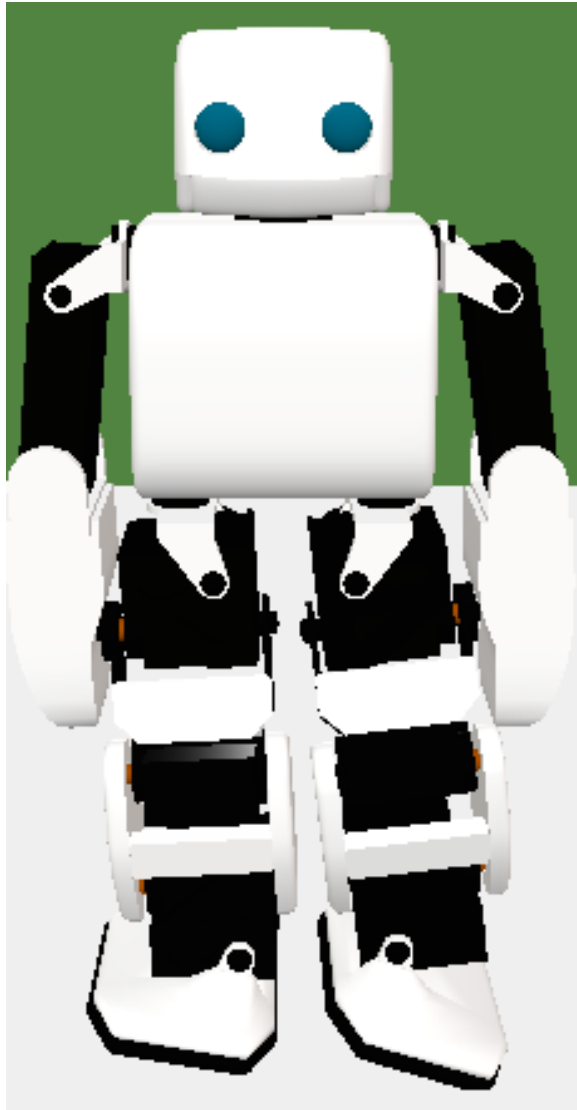


Figura 42. Sexto paso para girar

Tabla 22. Posición de los ángulos para sexto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 1.8	cadera	= 0
hombro2	= -8.6	hombro2	= 6.1
codo	= -8.0	codo	= 9.4
muslo2	= -6.6	muslo2	= -3.2
muslo1	= 27.5	muslo1	= -12.4
rodilla	= -46.4	rodilla	= 23.6
pie2	= -17.9	pie2	= 12.7
pie1	= -26.4	pie1	= -13.7

## 7. Séptimo paso

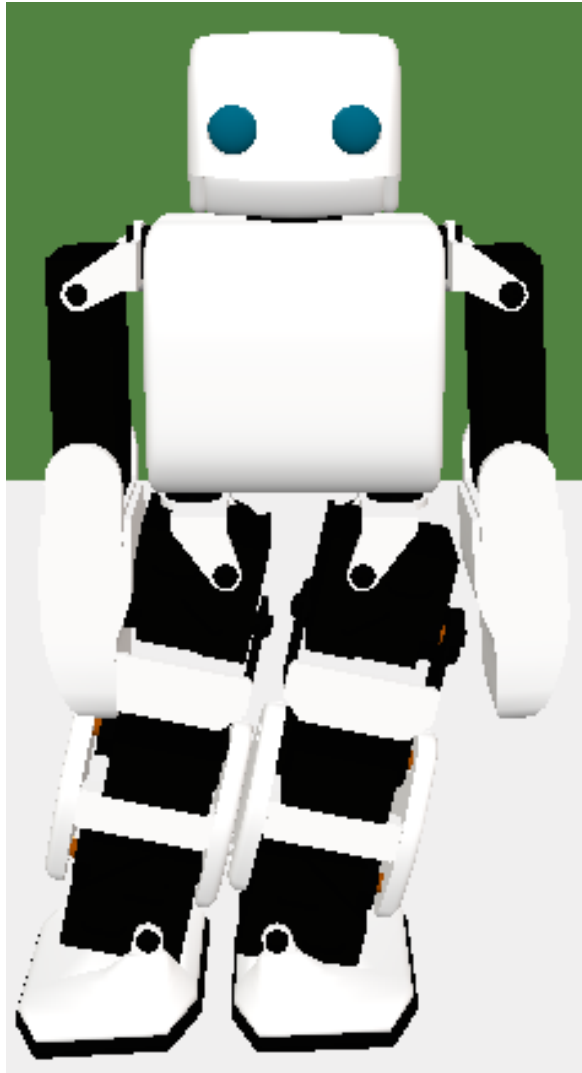


Figura 43. Séptimo paso para girar

Tabla 23. Posición de los ángulos para séptimo paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -3.6	hombro2	= 0
codo	= 0	codo	= 0
muslo2	= 12.0	muslo2	= 12.0
muslo1	= 0	muslo1	= 0
rodilla	= 0	rodilla	= 0
pie2	= 0	pie2	= 0
pie1	= 15.0	pie1	= 16.0

## 8. Finalización de giro

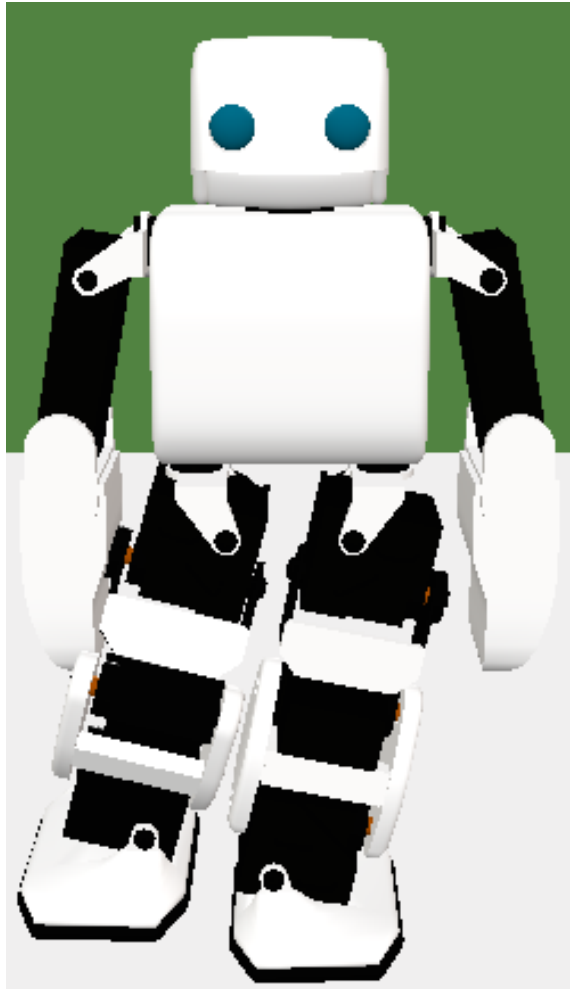


Figura 44. Séptimo paso para girar

Tabla 24. Posición de los ángulos para finalización de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 12.0	muslo2	= 15.0
muslo1	= 0	muslo1	= -26.8
rodilla	= 0	rodilla	= 60.5
pie2	= 0	pie2	= 29.4
pie1	= 14.0	pie1	= 14.0

#### 4.1.7.5 Estudio de movimientos: Giro hacia izquierda.

Para describir también a este desplazamiento se presenta a continuación la figura que representa a cada paso como muestra del patrón del movimiento, mientras que la tabla de cada paso indica los valores de la posición de los ángulos para ejecutarlo (los ángulos son en grados sexagesimales).

##### 1. Primer paso

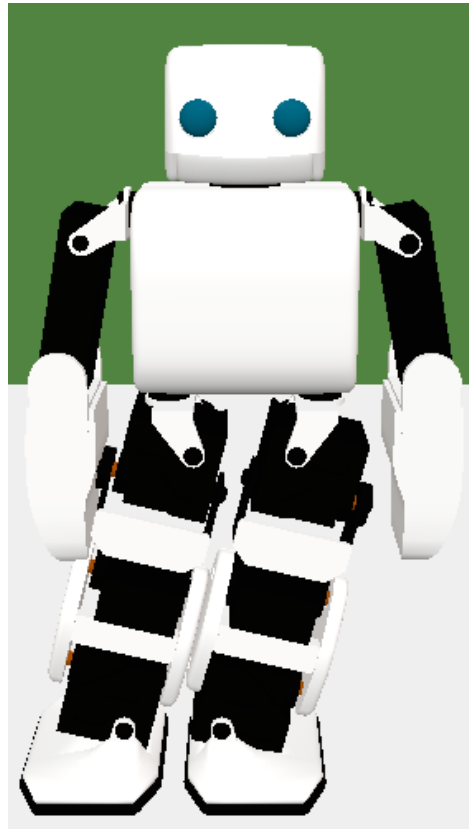


Figura 45. Primer paso para girar a la izquierda

Tabla 25. Posición de los ángulos para inicio de giro

Lado izquierdo		Lado derecho	
hombro1	= -1.0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 14.0	muslo2	= 14.0;
muslo1	= 0	muslo1	= 0
rodilla	= 0	rodilla	= 0
pie2	= 0	pie2	= 0
pie1	= 14.0	pie1	= 14.0

## 2. Segundo paso

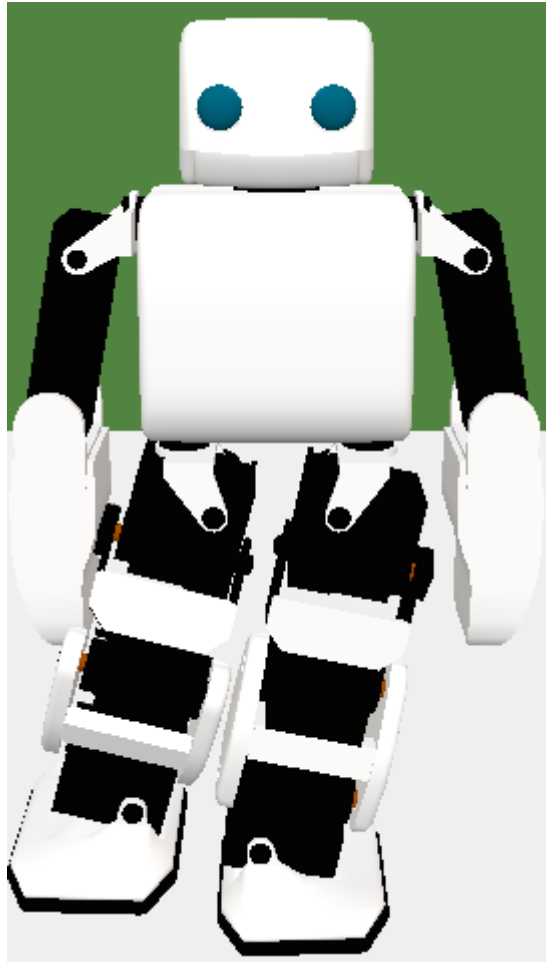


Figura 46. Segundo paso para girar a la izquierda

Tabla 26. Posición de los ángulos para segundo paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 12.0	muslo2	= 15.0
muslo1	= 0	muslo1	= -26.8
rodilla	= 0	rodilla	= 60.5
pie2	= 0	pie2	= 29.4
pie1	= 14.0	pie1	= 14.0

### 3. Tercer paso

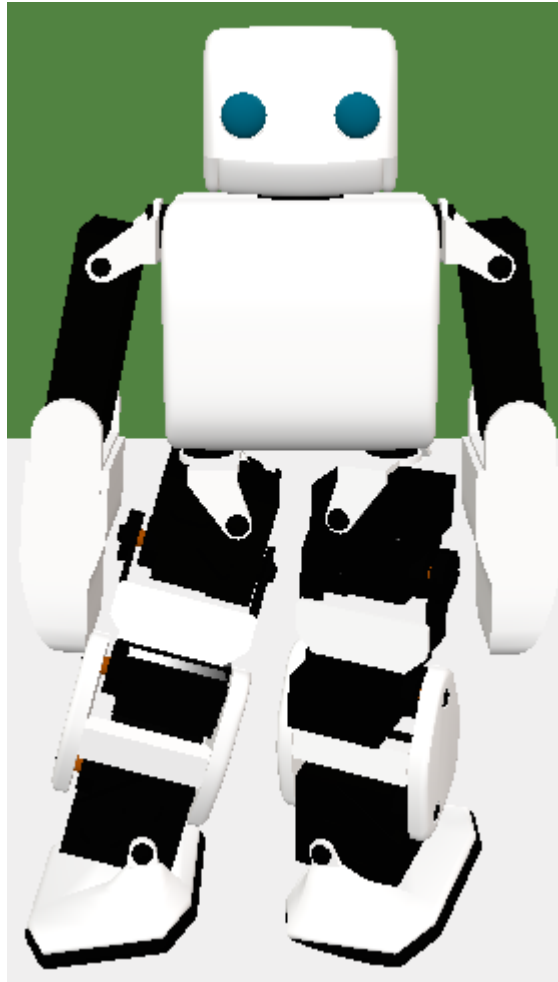


Figura 47. Tercer paso para girar a la izquierda

Tabla 27. Posición de los ángulos para tercer paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= -23.3	cadera	= 0
hombro2	= -8.6	hombro2	= 13.1
codo	= -8.0	codo	= 9.4
muslo2	= 3.0	muslo2	= 16.0
muslo1	= 0	muslo1	= -16.4
rodilla	= 0	rodilla	= 28.4
pie2	= 7.0	pie2	= 13.5
pie1	= 11.0	pie1	= 24.5

#### 4. Cuarto paso



**Figura 48.** Cuarto paso para girar a la izquierda

**Tabla 28.** Posición de los ángulos para cuarto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= -23.3	cadera	= 0
hombro2	= -8.6	hombro2	= 13.1
codo	= -8.0	codo	= 9.4
muslo2	= -7.0	muslo2	= -1.0
muslo1	= 0	muslo1	= -6.4
rodilla	= 0	rodilla	= 6.4
pie2	= 3.0	pie2	= 3.5
pie1	= -15.0;	pie1	= -3.5



## 5. Quinto paso

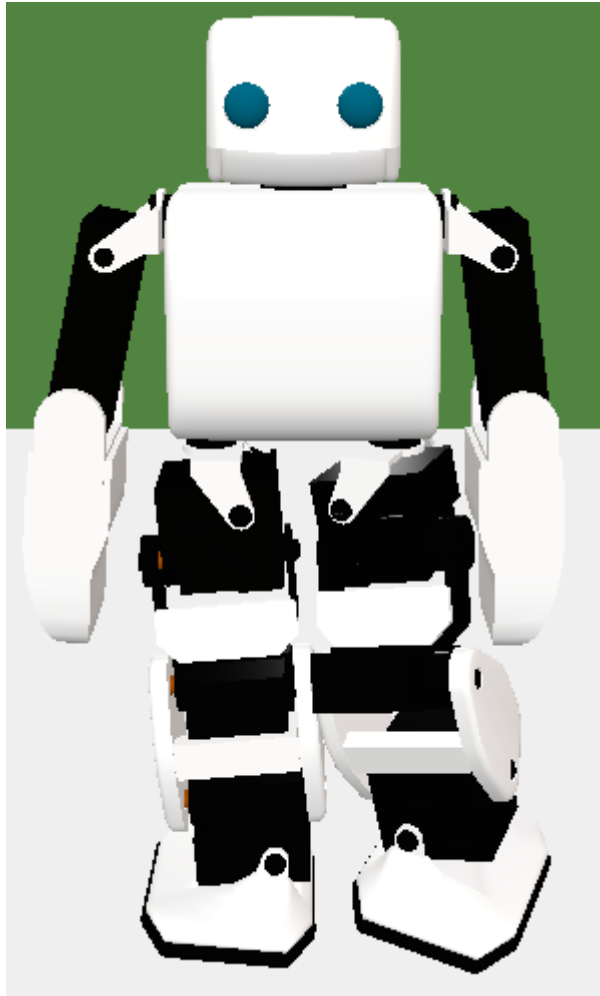


Figura 49. Quinto paso para girar a la izquierda

Tabla 29. Posición de los ángulos para quinto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= -23.3	cadera	= 0
hombro2	= -8.6	hombro2	= 13.1
codo	= -8.0	codo	= 9.4
muslo2	= -12.0	muslo2	= -5.0
muslo1	= 22.3	muslo1	= -6.4
rodilla	= -45.3	rodilla	= 6.4
pie2	= -20.1	pie2	= 3.5
pie1	= -19.0	pie1	= -10.5

## 6. Sexto paso

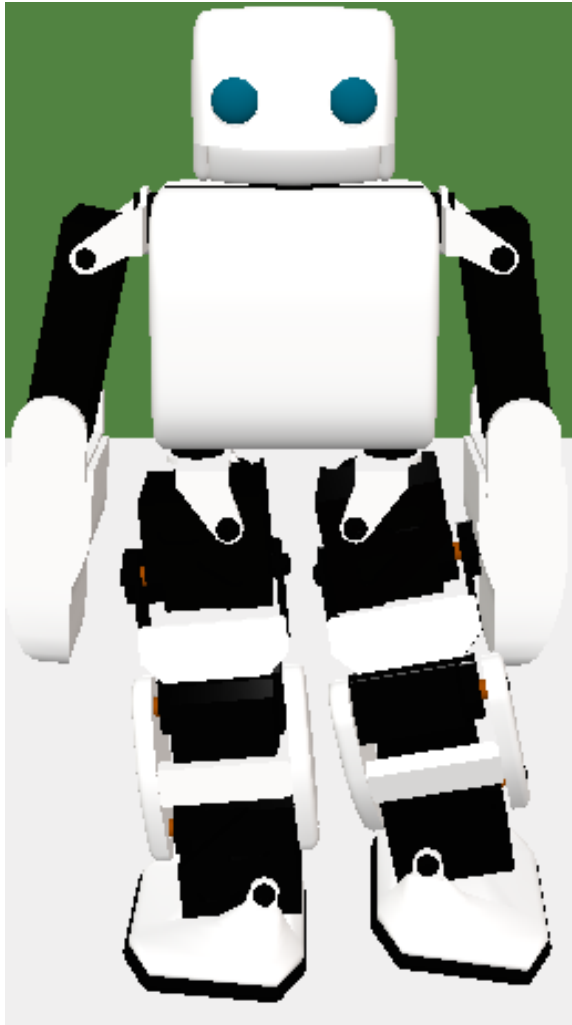
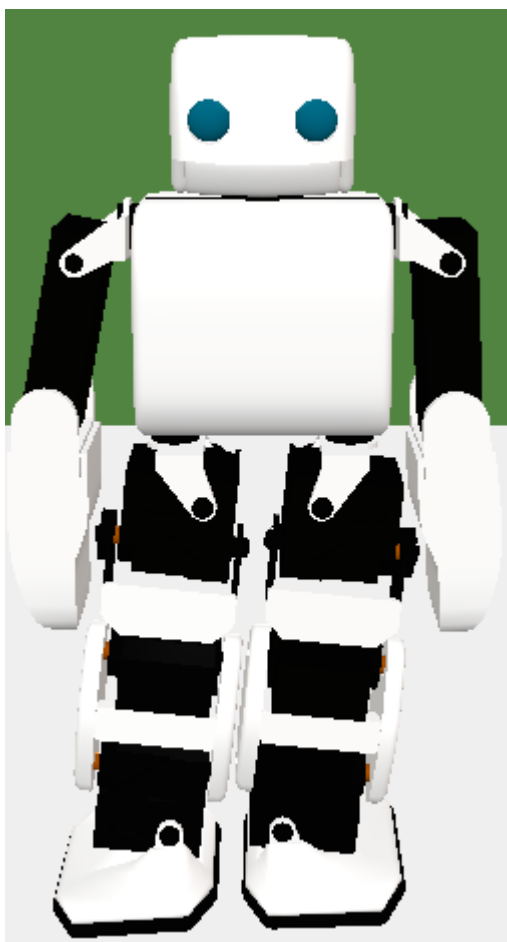


Figura 50. Sexto paso para girar a la izquierda

Tabla 30. Posición de los ángulos para sexto paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= -3.7	cadera	= 0
hombro2	= -8.6	hombro2	= 13.1
codo	= -8.0	codo	= 9.4
muslo2	= -12.0	muslo2	= -5.0
muslo1	= 22.3	muslo1	= -6.4
rodilla	= -45.3	rodilla	= 6.4
pie2	= -20.1	pie2	= 3.5
pie1	= -19.0;	pie1	= -10.5

## 7. Séptimo paso



**Figura 51.** Séptimo paso para girar a la izquierda

**Tabla 31.** Posición de los ángulos para séptimo paso de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -3.6	hombro2	= 12.0
codo	= 0	codo	= 12.5
muslo2	= 6.0	muslo2	= 6.0
muslo1	= 0	muslo1	= -2.0
rodilla	= 0	rodilla	= 0
pie2	= 2.0	pie2	= 1.0
pie1	= 9.0	pie1	= 10.0

## 8. Finalización de giro

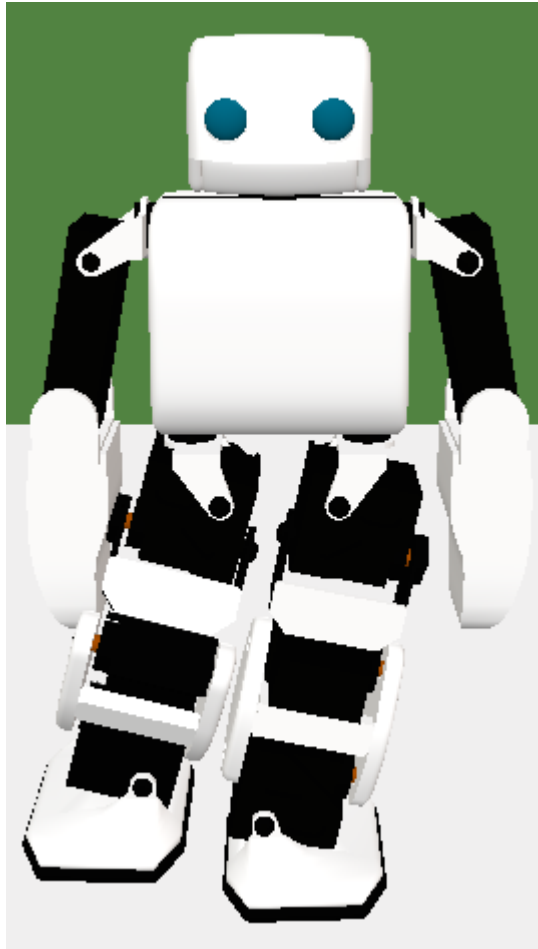


Figura 52. Finalización de giro a la izquierda

Tabla 32. Posición de los ángulos para finalización de giro

Lado izquierdo		Lado derecho	
hombro1	= 0	hombro1	= 0
cadera	= 0	cadera	= 0
hombro2	= -11.1	hombro2	= 11.1
codo	= -11.9	codo	= 11.9
muslo2	= 12.0	muslo2	= 15.0
muslo1	= 0	muslo1	= -26.8
rodilla	= 0	rodilla	= 60.5
pie2	= 0	pie2	= 29.4
pie1	= 14.0	pie1	= 14.0

Para finalizar el análisis de la marcha se destaca que el movimiento principal y de donde nacen el resto de desplazamientos es la marcha normal hacia

adelante; igualmente cuando termina una marcha o giro, el humanoide debe regresar a la posición inicial.

#### 4.1.8 DIAGRAMA DE FLUJO

A continuación, en la figura 53 se presenta el diagrama de flujo del proyecto:

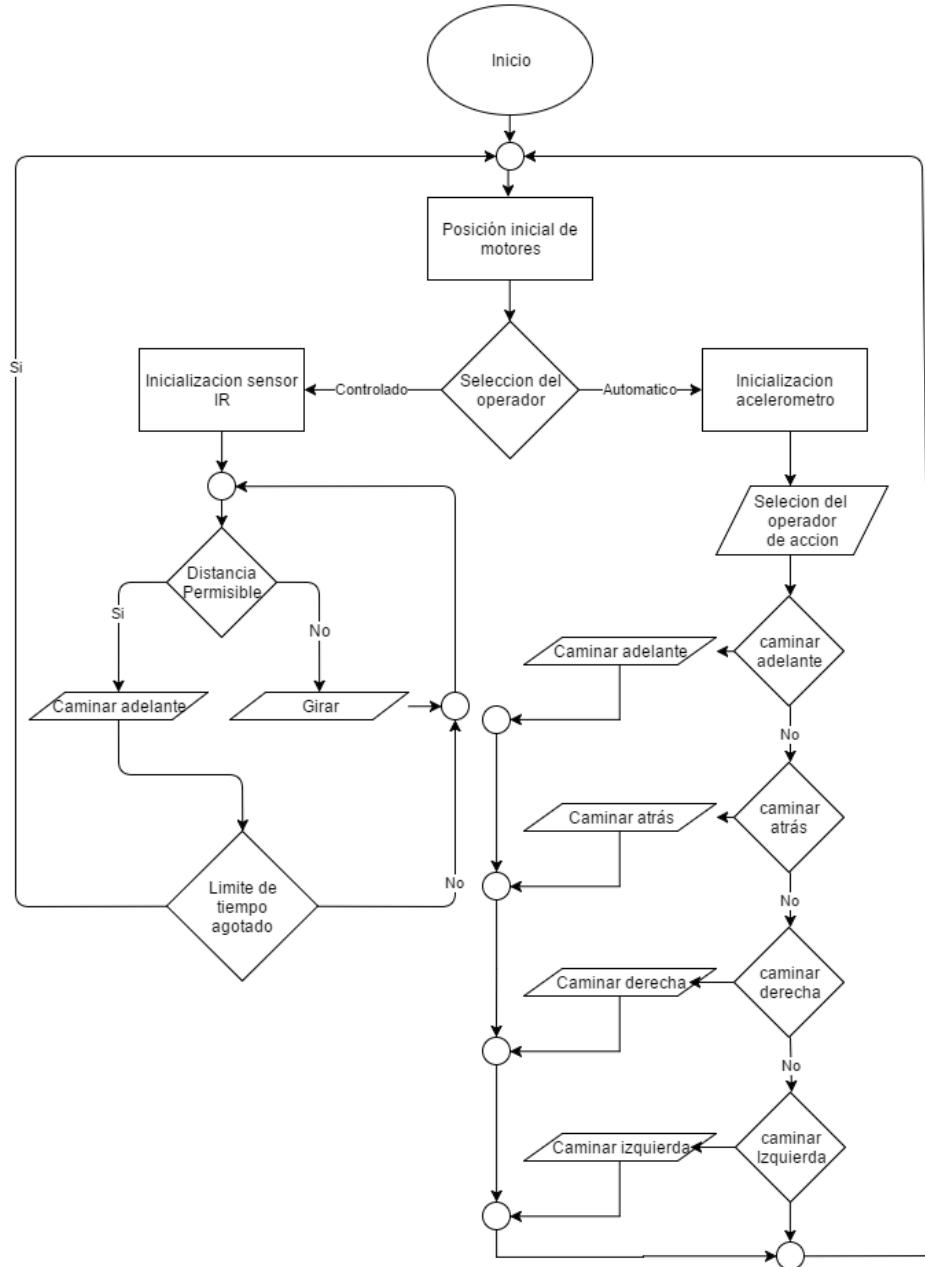


Figura 53. Diagrama de Flujo

## **5 ANÁLISIS DE RESULTADOS**

## 5.1 PRUEBA DE FUENTE DE ENERGÍA

Para comprobar el tiempo de funcionamiento del robot, se probó con diversos movimientos hasta que las baterías se descargaran, las pruebas se basaron en programar el humanoide con un solo tipo de movimiento a la vez, en este sentido, los tipos de movimiento fue: a) caminar adelante, b) caminar atrás, c) correr adelante, d) marcha sin desplazamiento. En la tabla 33 se observan los resultados obtenidos de las pruebas.

**Tabla 33.** Prueba de duración de funcionamiento

<b>Tipo de Movimiento</b>	<b>Tiempo de actividad</b>
Caminar adelante	26 minutos
Caminar atrás	24 minutos
Correr adelante	19 minutos
Marcha sin desplazamiento	27 minutos
<b>PROMEDIO DE TIEMPO</b>	24 minutos

De estos resultados se desprende que; el tiempo promedio de duración con uso continuo de la energía fue de 24 minutos. En donde, la marcha estacionaria tiene el mayor período con 27 minutos, y el movimiento de correr hacia adelante tiene el menor lapso con 19 minutos.

## 5.2 PRUEBA DE PLACA, SHIELDS Y SERVOS

En el proyecto, se implementó el Arduino nano como placa de control, la placa Adafruit 16 Channel como Shield para servos, con ellos, se controlaron un máximo de 32 servos a la vez como prueba (el humanoide solo usa 18 servos), de esta forma, el uso de servos es inversamente proporcional al de la corriente. En la prueba con 32 servos funcionando simultáneamente, la duración fue de aproximadamente 16 minutos, sin verse afectado el correcto funcionamiento de cada servo en este tiempo. En la figura 54 se observa la conexión de los servos para la realización de la prueba.

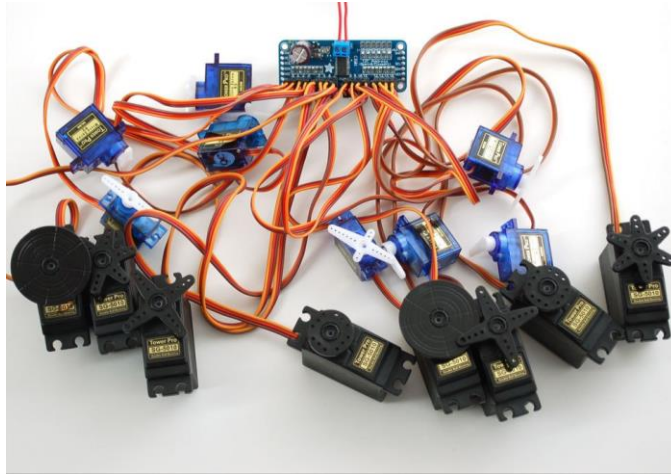


Figura 54. Conexión de servos

### 5.3 PRUEBA DE SENSOR DE PROXIMIDAD Y GIROSCOPIO

Para la realización de las pruebas del sensor de proximidad SHARP GP2Y0A41SK0F, se tomaron como referencia los valores establecidos por el fabricante como se observan en la figura 55:

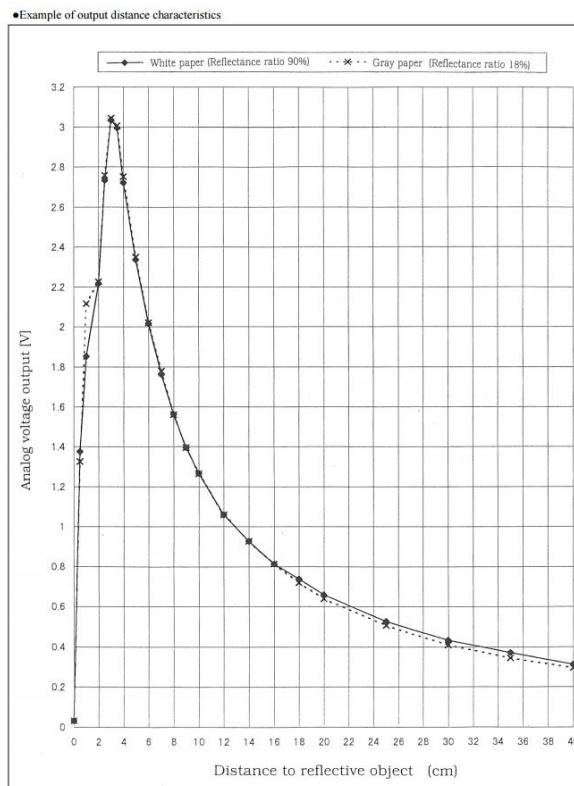


Figura 55. Datasheet de sensor de proximidad (Sharp, 2016)



En la tabla 34 se pueden ver los voltajes generados por este sensor en comparación al Datasheet que proporciona el fabricante:

**Tabla 34.** Comparación de voltajes entre Datasheet y Prueba

<b>Distancia</b>	<b>Valores SHARP según Datasheet (V)</b>	<b>Valores SHARP de la prueba (V)</b>
4cm	2.8	2.742
8cm	1.6	1.626
12cm	1.1	1.092
16cm	0.8	0.983
20cm	0.6	0.561

En función de lo que se expresa en la tabla, se puede verificar que los valores de las pruebas son muy cercanos a los que indica el Datasheet de la compañía; presentando diferencias mínimas. Lo que implica que el funcionamiento del sensor es muy próximo a lo que ofrece el fabricante, y por tanto adecuado para el proyecto en cuestión.

## **5.4 PRUEBA DE SOFTWARE CENTRAL**

La programación de este proyecto tiene a Arduino, como software central; y para la comprobación de su desempeño se realizaron varias pruebas para cada uno de los movimientos que el humanoide debía ejecutar: Luego de varias pruebas a través de ensayo y error, se seleccionó para cada uno de los movimientos el código con mayor funcionalidad; para esto, se tomaron en cuenta aspectos como movilidad, equilibrio y estética al ejecutar las acciones correspondientes. Por efectos de estilo y espacio, los códigos seleccionados de programación se presentarán como anexos del proyecto.

## **5.5 COMUNICACIÓN RASPBERRY PI Y ARDUINO**

La comunicación entre Raspberry Pi y Arduino representó uno de los retos más significativos del proyecto; que implicó la conjunción de los elementos del diseño electrónico con los elementos del diseño de software y control. Así

pues, para comprobar la comunicación entre las placas de control y del sistema de comunicación de la aplicación con el robot humanoide, se realizaron pruebas desde diferentes ubicaciones del operador. Esta evaluación se llevó a cabo desde diferentes bloques de la universidad, como los Laboratorios de Mecatrónica, Áreas verdes de la universidad, el Bar o el ITIC. En la tabla 35 se pueden observar los resultados obtenidos en las pruebas de control realizadas al robot humanoide desde las distintas ubicaciones.

**Tabla 35.** Evaluación del Tiempo de respuesta de la comunicación

<b>Ubicación</b>	<b>Tiempo de respuesta (en segundos)</b>
CIMETICS	0.8
Laboratorios Mecatrónica	0.9
ITIC	1
Bar de la Universidad	1
Áreas verdes de la Universidad	1.2

Estos resultados indican que el tiempo promedio de respuesta del humanoide al control del operador desde distintas áreas de la Universidad es de 0,96 segundos; este tiempo es adecuado para los efectos del desarrollo del proyecto, ya que no es un proyecto donde se involucre directamente riesgos por el tiempo de respuesta como es el caso de aplicaciones industriales y ahora de telemedicina, donde los tiempos de respuesta deben ser extremadamente rápido.

## **5.6 PRUEBA DE VELOCIDAD Y DISTANCIA DE LA MARCHA**

Para lograr la velocidad apropiada del humanoide se realizaron varias pruebas con distintas configuraciones en el tiempo de retraso de la programación, de esta manera se pudo seleccionar y establecer los tiempos de ejecución de las acciones que controlan la velocidad de marcha. En este sentido, para poder

realizar la acción de la marcha se ejecuta la línea de código que se muestra en la figura 56.

```
void caminar_adelante() {
    posicionInicial();
    delay(200);
    paso1_adelante();
    delay(200);
    paso2_adelante();
    delay(200);
    paso3_adelante();
    delay(200);
    paso3a_adelante();
    delay(200);
    paso4_adelante();
    delay(200);
    paso5_adelante();
    delay(200);
    paso6_adelante();
    delay(200);
    paso6a_adelante();
    delay(200);
    paso7_adelante();
    delay(200);
}
```

**Figura 56.** Código de la marcha

Por medio del código mostrado, se puede observar que para la ejecución completa de un ciclo de la acción de la marcha debe cumplirse hasta el paso 7, y entre cada paso se evidencia la presencia de un retardo; la sumatoria de estos retardos proporcionarán el tiempo total que transcurrió para ejecutarse la acción. Este tiempo, al sumarse los 200 milisegundos de cada paso da como resultado 2 segundos de retardo. Además, en la siguiente tabla se muestran los tiempos y los diferentes ciclos en los cuales se configuró al humanoide. En función de estos datos, para configurarlo finalmente se seleccionó la opción 3 de la tabla 36, ya que presentó mayor estabilidad.

**Tabla 36.** Ciclos de configuración del humanoide

Opción	Tiempo de cada ciclo (s)	Ciclos	Tiempo al final del ciclo (s)	Distancia recorrida por el humanoide (cm)
1	0.5	40	20	71
2	1	20	20	63
3	2	10	20	54
4	3	6	18	41

5	4	5	20	32
---	---	---	----	----

## 5.7 PRUEBAS DE ERROR EN SERVOS MOTORES

Para evidenciar si existe errores al momento de posicionar los servos en sus distintos movimientos se procedió a seleccionar aleatoriamente uno de los movimientos que el humanoide ejecutara, en este caso se seleccionó el paso 4 de giro a la derecha. Se evidencia que existe un margen de error que no supera los 0.5 grados. Se debe tener presente que este error es directamente proporcional a un mayor ángulo a ejecutar, pero al analizar la tabla 37, podemos observar que no existirá un error mayor a 1 grado el peor de los casos.

**Tabla 37.** Errores de posición en los servos del humanoide

Nombre del servo motor	Movimiento de servo (Grados)	Movimiento de servo en humanoide (Grados)	Error absoluto (Grados)	Error porcentual (%)
Hombro1	0	0	0	0
Cadera	-23.3	-22.8	0.5	-2.1
Hombro2	-8.6	-8.4	0.2	-2.3
Codo	-8	-7.8	0.2	-2.5
Muslo1	-7	-6.8	0.2	-2.9
Muslo2	0	0	0	0
Rodilla	0	0	0	0
Pie2	3	3.05	0.05	1.6
Pie1	-15	-14.6	0.4	-2.7

## 5.8 ANÁLISIS DE RESULTADOS

Al haber realizado las pruebas de cada bloque que compone el diseño electrónico y eléctrico de este proyecto, cuyo objeto es generar la movilidad del humanoide a través de la comunicación con una placa Raspberry Pi (encargada de enviar las órdenes), y mediante la programación en Arduino; de forma que en función de todos estos elementos permitir el correcto funcionamiento del robot.

Para hacer un resumen e integración de los resultados de las pruebas se puede señalar que, en lo concerniente a la duración de la fuente de energía, ésta fue de 24 minutos en promedio; asimismo, se pudo tener a 32 servos funcionando simultánea y correctamente durante aproximadamente 16 minutos, lo cual es completamente funcional porque el humanoide usa 18 servos únicamente. También, las diferencias entre el funcionamiento del sensor SHARP en las pruebas y lo que indica el manual fueron mínimas. Igualmente, se obtuvieron los códigos óptimos de funcionamiento del robot luego de varias pruebas de ensayo y error. Además, el tiempo de respuesta del humanoide al control del operador desde distintas áreas de la Universidad fue adecuado, con un promedio de 0,96 segundos. Asimismo, la elección de la velocidad de marcha fue de 10 ciclos a 2 segundos por ciclo con un recorrido de 54cm.

De esta manera, en la medida en que se desarrolló del proyecto y la realización de las pruebas correspondientes de cada elemento del diseño se puede evidenciar la aplicación de diversos métodos de programación para el movimiento del humanoide; se diseñó el sistema eléctrico y electrónico para el control del humanoide de manera funcional; se realizaron las pruebas de desempeño necesarias para corregir y mejorar el funcionamiento del sistema; y se implementaron sensores para ampliar las funcionalidades del humanoide. Así, todos los objetivos específicos de la investigación fueron alcanzados; por lo que a su vez se logra el objetivo general de la investigación, que fue: Diseñar el circuito electrónico y eléctrico de interfaz entre el humanoide y la PC

## **6 CONCLUSIONES Y RECOMENDACIONES**

## CONCLUSIONES

El sistema de LiPo baterías usado en este proyecto se muestra conveniente, debido a que proporciona una autonomía alta en comparación con otros tipos de baterías y ocupa poco espacio, por lo que es considerablemente adaptable. Su capacidad brinda alrededor de 25 minutos de operación; tiempo suficiente para demostrar las funciones del Humanoide.

Asimismo, el uso de Arduino como placa de control central, es una selección acertada para cumplir con los objetivos del proyecto, ya que con su enorme variedad de shields, periféricos, librerías, diferentes modelos de placas, así como la facilidad de integración con otras placas como la Raspberry Pi usada para el control del operador. También, es importante destacar que es una plataforma de código y hardware abierto como Arduino, Adafruit, Raspberry; incluso para el diseño mecánico se usó una impresora de código libre basada en la Prusa I3. De esta forma se puede acceder a todo aspecto del funcionamiento circuital y algorítmico de las placas y herramientas; lo que colabora a que los investigadores y desarrolladores puedan tener un acceso completo de las herramientas usadas y así, un mayor entendimiento de todos los procesos y funcionamiento.

Igualmente, el uso del bus de comunicaciones I2C como principal enlace entre los periféricos y la placa de arduino, brindó la posibilidad de conectar dos placas controladoras de servos y un sensor que cumple funciones como giroscopio y acelerómetro, cada dispositivo conectado tiene una dirección única y se puede seleccionar maestro o esclavo a cualquier de los dispositivos conectados en este bus. Con una capacidad para este mismo bus de conectar hasta 128 dispositivos simultáneamente.

Además, con el MPU6050 y el sensor de distancia (el primero usado como giroscopio), se aprovechó el cambio del ángulo que se genera con respecto a los tres ejes, y se pudo establecer rangos a los ángulos para saber conjuntamente con el sensor de distancia, cuándo el humanoide se

encontraba en alguna posición incorrecta con el objeto de luego llevarlo a la posición inicial.

Es relevante señalar que, en general un proyecto de este tipo tiene un costo elevado, debido al desarrollo de la parte mecánica, implementación de la electrónica, adquisición de servomotores y todo lo que conlleva la investigación. En este sentido, la adquisición de los materiales tuvo complejidad debido a su elevado costo y al hecho de no encontrarse en el país, por lo que siempre se debió recurrir a la importación.

Para finalizar, se resalta que este proyecto ha permitido aplicar la mecatrónica en su modo concurrente más puro, ya que ha implicado la integración de varias destrezas y conocimientos técnicos de variadas disciplinas como la electrónica, mecánica, física, matemática, entre otras; todo con el objetivo de alcanzar satisfactoriamente las metas propuestas. De esta manera, los requerimientos de proyectos académicos como éste, se transforman en un pilar fundamental que impulsa significativamente en el desarrollo técnico de las personas involucradas.

## **RECOMENDACIONES**

En primer lugar, se señala que como en todo proyecto científico hay cabida para las observaciones, optimizaciones y mejoras en general. En el caso particular de este trabajo, uno de los aspectos susceptibles de mejora se encuentra en los algoritmos de programación de la movilidad. Existe una elevada probabilidad que a través de la investigación y la optimización del código, o por medio de nuevas (o distintas) funciones que permitan interactuar al humanoide de una manera más eficiente, tanto estudiantes, como investigadores y desarrolladores de mecatrónica podrán realizar mejores prácticas.

Igualmente, para proyectos similares donde el uso de actuadores como servomotores es elevado; se sugiere que todos sean del mismo tipo para



evitar problemas al momento de generar los movimientos, ya que algunos fabricantes de servos tienen invertido el ángulo de giro con respecto a otros fabricantes y eso genera diversos conflictos al momento de programar los movimientos.

Además, se evidenció que una de las debilidades más grandes del proyecto estaba referida a la hora de integrar con la parte mecánica, debido a que el diseño presentaba inconvenientes; por lo que se requirió la modificación de este diseño en varias oportunidades para que se adaptara a las características electrónicas seleccionadas.

Finalmente, se resalta que el proyecto está pensado para seguir implementando más funcionalidades relacionadas con nuevos movimientos, así como nuevos sensores que permitirán facilitar el estudio e investigación de humanoides.

## BIBLIOGRAFÍA

- Agudelo, A. et al (2013). *Marcha: Descripción, métodos, herramientas de evaluación y parámetros de normalidad reportados en la literatura*. Revista CES Movimiento y Salud Volumen 1 N° 1. Colombia.
- Arduino (2016). *Sensor Giroscopio*. Imagen obtenida de: <http://playground.arduino.cc/Main/MPU-6050>
- Arduino Nano V3. Imagen obtenida de: <http://saber.patagoniatec.com/arduino-nano-328-arduino-atmega-clon-compatible-arduino-argentina-ptec/>
- Banzi, M. (2012). *Cómo Arduino está abriendo la imaginación*. Charla de TED. Obtenida de: [https://www.ted.com/talks/massimo\\_banzi\\_how\\_arduino\\_is\\_open\\_sourcing\\_imagination/transcript](https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination/transcript)
- Bishop, R. (2006). *Mecatrónica: Una introducción*. Editorial Taylor and Francis Group. Estados Unidos.
- Blog de TodoSAI. (2015). *Baterías – Tipos y usos*. Obtenido de: <http://todosai.com/blog/baterias-tipos-y-usos-b129.html>
- Bolton, W. (2010) *Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica*. 4ta edición. Editorial Alfaomega. México.
- Bordallo, F. (2012) *Diseño e Implementación de brazo robot SCARA de 3gdl*. (Tesis inédita de Maestría). Universidad Politécnica de Valencia. España.
- Carletti, E. (2015). *Comunicación – Bus I2C*. Obtenido de: [http://robots-argentina.com.ar/Comunicacion\\_busI2C.htm](http://robots-argentina.com.ar/Comunicacion_busI2C.htm)
- Chip Adafruit 16 canales. Imagen obtenida de: <https://www.adafruit.com/product/1411>
- Chip PCA9685. Imagen obtenida de: <https://www.adafruit.com/product/815>
- Circelli, G. (2015). *Microcontroladores o Arduino, ¿no es lo mismo?*. Obtenido de: <http://panamahitek.com/microcontroladores-o-arduino-no-es-lo-mismo/>

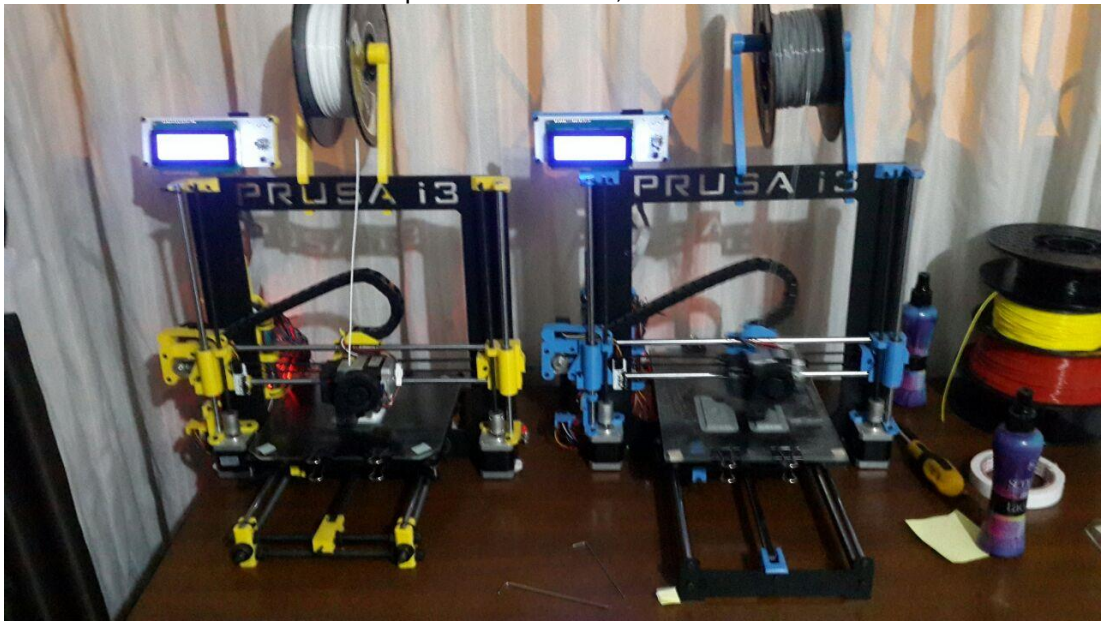
- Cook, D. (2009). *Construcción de robots para principiantes*. 2da edición. Editorial Apress. Estados Unidos
- Craig, J. (2006). *Robótica*. 3era edición. Editorial Pearson. México.
- Craig, K. y Stolfi, F. (2006) *Introducción a las computadoras y sistemas lógicos*. Editorial Taylor and Francis Group. Estados Unidos.
- Crespo, E. (2015). *Aprendiendo Arduino*. Obtenido de: <https://aprendiendo-arduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>
- Daza, J (2007). *Examen de la marcha humana. Evaluación clínico-funcional del movimiento corporal humano*. Editorial Panamericana. Colombia.
- Diccionario de la Real academia española (2016) Robot. Obtenido de : <http://dle.rae.es/?id=WYRIhzm>
- Enciclopedia de Clasificaciones. (2016). *Tipos de baterías*. Obtenido de: <http://www.tiposde.org/cotidianos/420-tipos-de-baterias/>
- e-bay (2016). *Regulador de Voltaje*. Imagen obtenida de: <http://www.ebay.com/itm/LM2577-DC-DC-Power-Converter-Adjustable-Step-up-Power-Converter-Module-/110953869492>
- Igual, C., Muñoz, E., Aramburu, C., (1996). *Fisioterapia general – Cinesiterapia*. Editorial Síntesis. España.
- Hernández Sampieri, et al (2014). *Metodología de la Investigación*. 6ta edición. Editorial McGraw Hill. México.
- HobbyKing (2016). Servomotor. Imagen obtenida de: [http://www.hobbyking.com/hobbyking/store/\\_\\_27749\\_\\_HobbyKing\\_8482\\_Mi\\_Digital\\_High\\_Torque\\_Servo\\_HV\\_MG\\_31kg\\_0\\_15sec\\_60g.html](http://www.hobbyking.com/hobbyking/store/__27749__HobbyKing_8482_Mi_Digital_High_Torque_Servo_HV_MG_31kg_0_15sec_60g.html)
- Khare, R. et al (2007) *Robótica*. Editorial Infinity Science Press. Estados Unidos – India.
- Kurfess, T. (2005) *Manual de Robótica y automatización*. Editorial CRC Press. Estados Unidos.
- Knudsen, J. (1999). *La guía no oficial para LEGO Mindstorms*. Editorial O'Reilly. Estados Unidos.

- Llamas, L. (2014) Arduino Puertos Serie. Obtenido de: <http://www.luisllamas.es/2014/04/arduino-puerto-serie/>
- Martínez, F. (2015) Instalación de Linux. Obtenido de: <https://openwebinars.net/tutorial-arduino-ide-arduino/>
- Méndez, V. (2012). *Diseño y Construcción de un Sistema Mecatrónico Articulado de Dos Dedos, Actuado con Músculos Neumáticos*. (Tesis inédita de Maestría). Centro Nacional de Investigación y Desarrollo Tecnológico. México
- Mianbao (2016) Conexión Mpu Arduino. Imagen obtenida de: <http://mianbao.space/article/7>
- Molina, M. (2015). *Sistema de control inteligente para robot manipulador*. (Tesis inédita de Pregrado). Ecuador.
- Moon, F. (2006) *Modelando Sistemas electromecánicos*. Editorial Taylor and Francis Group. Estados Unidos.
- Mora, A. (2015). *Teebot, el robot que enseña programación y robótica a niños*. Obtenido de <http://www.propiedadintelectual.gob.ec/teebot-el-robot-que-ensena-programacion-y-robotica-a-ninos/>
- Motor de inducción de Tesla. Imagen obtenida de: [https://commons.wikimedia.org/wiki/File:Tesla%27s\\_induction\\_motor.jpg](https://commons.wikimedia.org/wiki/File:Tesla%27s_induction_motor.jpg)
- Motor de Vapor Boulton y Watts. Imagen obtenida de: [https://commons.wikimedia.org/wiki/File:Boulton\\_and\\_Watt\\_centrifugal\\_governor-MJ.jpg](https://commons.wikimedia.org/wiki/File:Boulton_and_Watt_centrifugal_governor-MJ.jpg)
- Parra, N. (2013). *Diseño e Implementación de un Sistema de Control Automático para un Robot Imitador en una Plataforma Humanoide mediante adquisición y procesamiento digital de imagen y video*. (Tesis inédita de Pregrado). Ecuador
- Patagoniatec (2016). Pines de Arduino nano. Imagen obtenida de: <http://saber.patagoniatec.com/arduino-nano-328-arduino-atmega-clon-compatible-arduino-argentina-ptec/>
- Placa Arduino Mega ADK. Imagen obtenida de: <https://www.openhacks.com/page/productos/id/23/titl/Arduino-Mega-ADK#.V6FqmfI97IU>

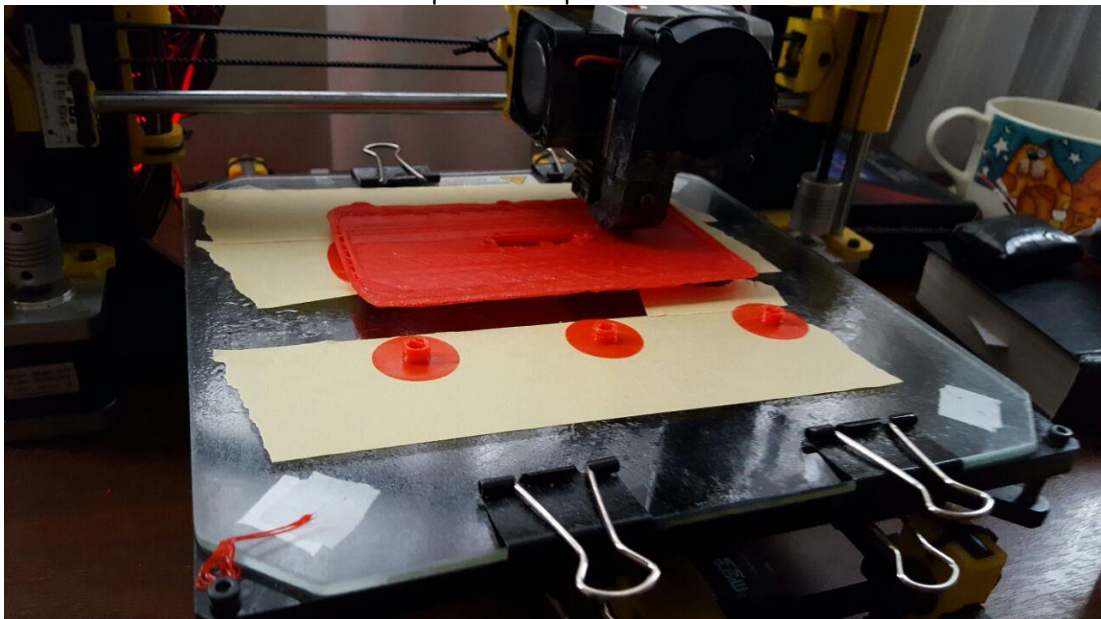
- Redes Zone (2014). *Controla tu Raspberry Pi de forma remota a través del protocolo RDP*. Obtenido de: <http://www.redeszone.net/raspberry-pi/controla-tu-raspberry-pi-de-forma-remota-traves-del-protocolo-rdp/>
- Reloj de agua Ctesibio. Imagen obtenida de [https://commons.wikimedia.org/wiki/File:ARAGO\\_Francois\\_Astronomie\\_Populaire\\_T1\\_page\\_0067\\_Fig16-17.jpg](https://commons.wikimedia.org/wiki/File:ARAGO_Francois_Astronomie_Populaire_T1_page_0067_Fig16-17.jpg)
- Robologs (2014). *Tutorial de Arduino y MPU-6050*. Imagen obtenida de: <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>
- Sandier, B. (1999). *Robótica: Diseñando los mecanismos para máquinas automáticas*. Editorial Academic Press. California
- Universidad Católica de Santiago de Guayaquil (2015). *Concurso Ecuatoriano de Robótica*. Obtenido de: <http://www2.ucsg.edu.ec/bi-servicios/45-noticias/detalle/1071-xi-concurso-ecuatoriano-de-robotica-2015.html>
- Web Prometec (2016). *Bus I2C*. Imagen obtenida de: <http://www.prometec.net/bus-i2c/>
- Wise, E (2005) *Robótica desmitificada*. Editorial McGraw Hill. Estados Unidos.

## ANEXOS

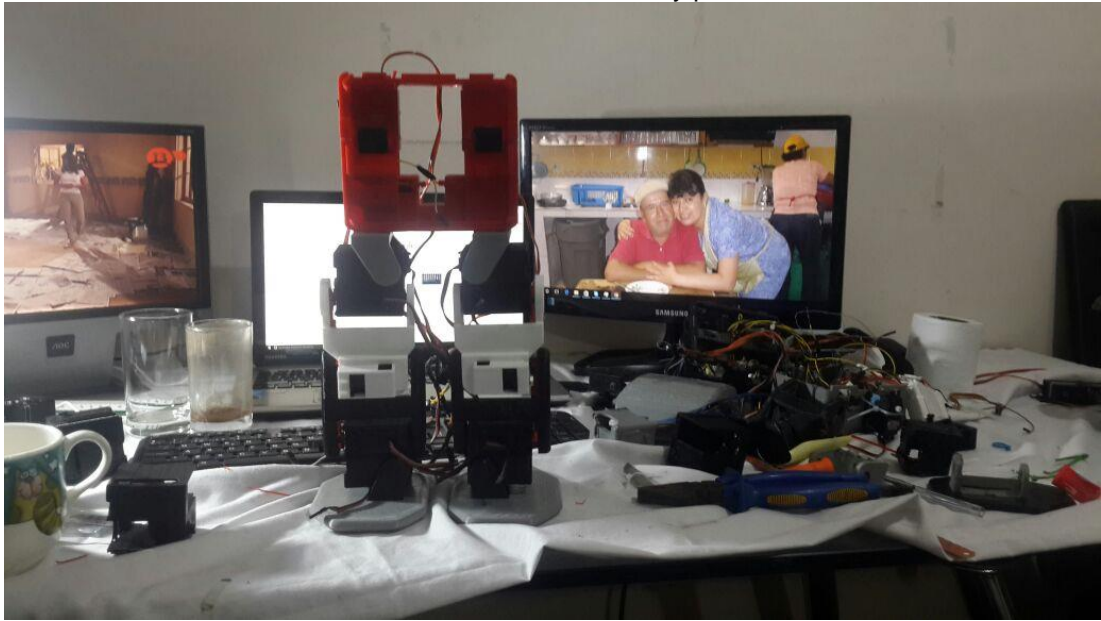
Anexo 1. Impresoras Prusa i3, en funcionamiento



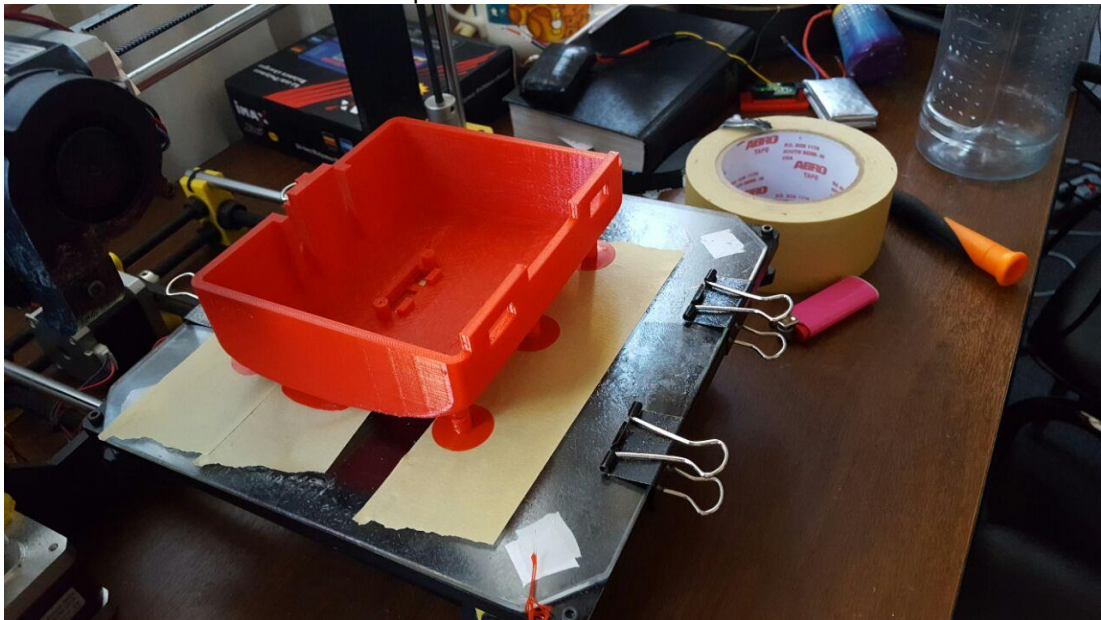
Anexo 2. Impresión de partes del humanoide



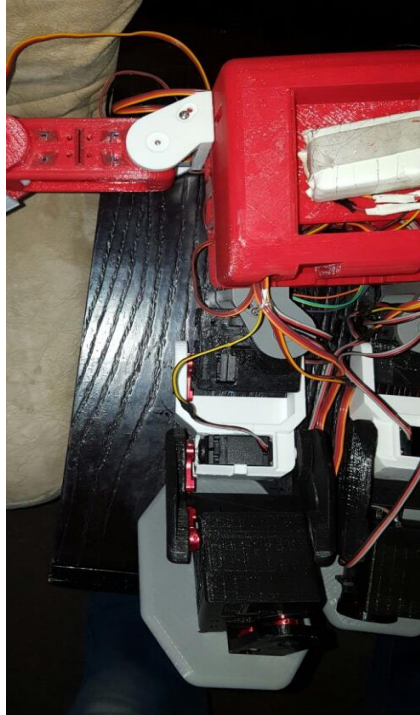
**Anexo 3. Ensamble de torso y piernas**



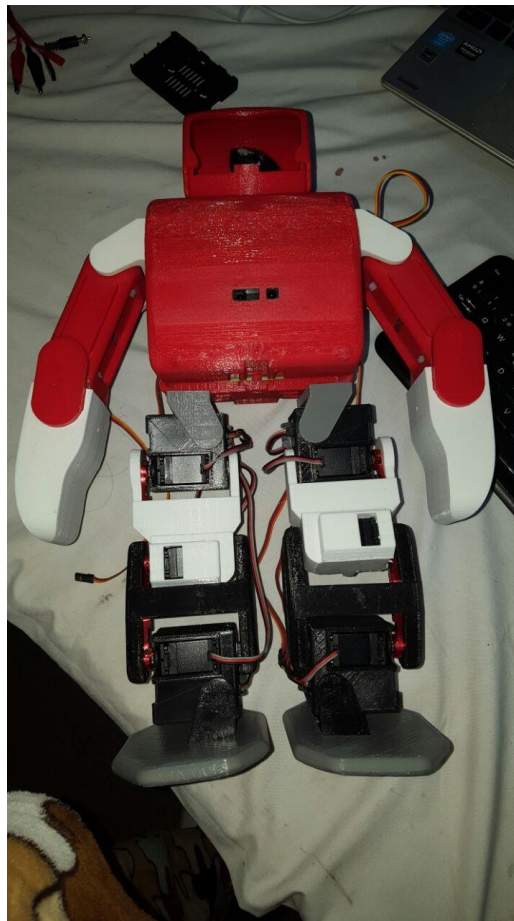
**Anexo 4. Impresión de abdomen del humanoide**



**Anexo 5.** Colocación de baterías en humanoide



**Anexo 6.** Ensamble total del humanoide con sensores y electrónica







```

uint8_t ihombro2 = 14; // servos pierna derecha 7.4V
uint8_t icodo = 15;
uint8_t imuslo2 = 4;
uint8_t imuslo1 = 5;
uint8_t irodilla = 6;
uint8_t ipie2 = 7;
uint8_t ipie1 = 8;

uint8_t dhombro1 = 12;
uint8_t dcadera = 13;
uint8_t dhombro2 = 14; // servos pierna derecha 6.0V
uint8_t dcodo = 15;
uint8_t dmuslo2 = 4;
uint8_t dmuslo1 = 5;
uint8_t drodilla = 6;
uint8_t dpi2 = 7;
uint8_t dpi1 = 8;

int iihombro1;
int iicadera;
int iihombro2; // servos pierna iz
int iicodo;
int iimuslo2;
int iimuslo1;
int iirodilla;
int iipie2;
int iipie1;
int ddhombro1;
int ddcadera;
int ddhombro2; // servos pierna derecha
int ddcodo;
int ddmuslo2;
int ddmuslo1;
int ddrodilla;
int ddpie2;
int angle1;
int angle2;
int angle3;
int angle4;
int angle5;
int angle6;
int angle7;
int angle8;
int angle9;
int angle10;
int angle11;
int angle12;
int angle13;
int angle14;
int angle15;
int angle16;
int angle17;
int angle18;

int iiihombro1;
int iiicadera;
int iiihombro2; // servos pierna derecha
int iiicodo;
int iiimuslo2;
int iiimuslo1;
int iirodilla;
int iipie2;
int iipie1;

int dddhombro1;
int dddcadera;
int dddhombro2; // servos pierna derecha
int dddcodo;
int dddmuslo2;

```

```

void setup()
{
  Serial.begin(9600);
  //Serial.println("32 channel PWM test!");

  pwm1.begin();
  pwm1.setPWMPfreq(60); // This is the maximum PWM frequency

  pwm2.begin();
  pwm2.setPWMPfreq(60); // This is the maximum PWM frequency

  // configure Arduino LED for
  pinMode(LED_PIN, OUTPUT);
}

```

## Anexo 8. Código de Programación en arduino (PRINCIPAL).

MOSUBIvale	caminar_adelante	caminar_atras	derecha	funciones_ordenes	izquierda	levantar_cara_abajo	levantar_cara_arriba	ordenes	principal
------------	------------------	---------------	---------	-------------------	-----------	---------------------	----------------------	---------	-----------

```

void principal() {

  angle1 = map(iihombro1, 75, -105, SERVOMIN, SERVOMAX);
  iihombro1 = (angle1);
  pwm1.setPWM(iihombro1, 0, iihombro1);

  angle2 = map(iiicadera, 80, -100, SERVOMIN, SERVOMAX);
  iiicadera = (angle2);
  pwm1.setPWM(iiicadera, 0, iiicadera);

  angle3 = map(iihombro2, 90, -90, SERVOMIN, SERVOMAX);
  iihombro2 = (angle3);
  pwm1.setPWM(iihombro2, 0, iihombro2);

  angle4 = map(iiicodo, 105, -75, SERVOMIN, SERVOMAX);
  iiicodo = (angle4);
  pwm1.setPWM(iiicodo, 0, iiicodo);

  angle5 = map(iiimuslo2, -90, 90, SERVOMIN, SERVOMAX);
  iiimuslo2 = (angle5);
  pwm1.setPWM(iiimuslo2, 0, iiimuslo2);

  angle6 = map(iiimuslo1, -100, 80, SERVOMIN, SERVOMAX);
  iiimuslo1 = (angle6);
  pwm1.setPWM(iiimuslo1, 0, iiimuslo1);

  angle7 = map(iiiodilla, -80, 100, SERVOMIN, SERVOMAX);
  iiiodilla = (angle7);
  pwm1.setPWM(iiiodilla, 0, iiiodilla);

  angle8 = map(iiipie2, -90, 90, SERVOMIN, SERVOMAX);
  iiipie2 = (angle8);
  pwm1.setPWM(iiipie2, 0, iiipie2);

  angle9 = map(iiipie1, -105, 75, SERVOMIN, SERVOMAX);
  iiipie1 = (angle9);
  pwm1.setPWM(iiipie1, 0, iiipie1);

  angle10 = map(dddhombro1, 90, -90, SERVOMIN, SERVOMAX);
  ddhombro1 = (angle10);
  pwm2.setPWM(dddhombro1, 0, ddhombro1);

  angle11 = map(dddcadera, 90, -90, SERVOMIN, SERVOMAX);
  ddcadera = (angle11);
  pwm2.setPWM(dddcadera, 0, ddcadera);

  angle12 = map(dddhombro2, 90, -90, SERVOMIN, SERVOMAX);
  ddhombro2 = (angle12);
  pwm2.setPWM(dddhombro2, 0, ddhombro2);
}

```

```
angle13 = map(dddcodo, 70, -110, SERVOMIN, SERVOMAX);
ddcodo = (angle13);
pwm2.setPWM(dccodo, 0, ddcodo);

angle14 = map(dddmuslo2, -100, 80, SERVOMIN, SERVOMAX);
ddmuslo2 = (angle14);
pwm2.setPWM(dmuslo2, 0, ddmuslo2);

angle15 = map(dddmuslo1, -95, 85, SERVOMIN, SERVOMAX);
ddmuslo1 = (angle15);
pwm2.setPWM(dmuslo1, 0, ddmuslo1);

angle16 = map(dddrodilla, -90, 90, SERVOMIN, SERVOMAX);
ddrodilla = (angle16);
pwm2.setPWM(drodilla, 0, ddrodilla);

angle17 = map(dddpie2, -87,93, SERVOMIN, SERVOMAX);
ddpie2 = (angle17);
pwm2.setPWM(dpie2, 0, ddpie2);

angle18 = map(dddpie1, -90, 90, SERVOMIN, SERVOMAX);
ddpie1 = (angle18);
pwm2.setPWM(dpie1, 0, ddpie1);

}
```



```

// read raw accel/gyro measurements from device
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

// these methods (and a few others) are also available
//accelgyro.getAcceleration(&sax, &say, & saz);
//accelgyro.getRotation(&gx, &gy, &gz);

#ifdef OUTPUT_READABLE_ACCELYGYRO
// display tab-separated accel/gyro x/y/z values
Serial.print("a/g:\t");

Serial.print(ax); Serial.print("\t");
Serial.print(ay); Serial.print("\t");
Serial.print(az); Serial.print("\t");
Serial.println();
#endif

#ifdef OUTPUT_BINARY_ACCELYGYRO

Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)(ay & 0xFF));
Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)(ay & 0xFF));
Serial.write((uint8_t)(az >> 8)); Serial.write((uint8_t)(az & 0xFF));

#endif

// blink LED to indicate activity
//blinkState = !blinkState;
//digitalWrite(LED_PIN, blinkState);
}

```