



**PARAMETRIZACIÓN DE LA MARCHA HUMANA USANDO UN TRAJE EN  
EXTREMIDADES INFERIORES PARA GENERAR LA REFERENCIA DEL  
CONTROL CINEMÁTICO DE UNA PRÓTESIS**

JONATHAN SEBASTIAN BUITRAGO DURAN

Universidad de Pamplona  
Facultad de Ingenierías y arquitectura  
Departamento de Ingenierías Mecánica, Mecatrónica e Industrial  
Pamplona, Colombia  
2020

**PARAMETRIZACIÓN DE LA MARCHA HUMANA USANDO UN TRAJE EN  
EXTREMIDADES INFERIORES PARA GENERAR LA REFERENCIA DEL  
CONTROL CINEMÁTICO DE UNA PRÓTESIS**

JONATHAN SEBASTIAN BUITRAGO DURAN

Tesis Presentada como requisito de grado para optar al título de:  
Ingeniero en Mecatrónica

Director

Ph.D. CÉSAR AUGUSTO PEÑA CORTÉS  
Doctor en Automática y Robótica

Co-Director

M.Sc. LUIS ENRIQUE MENDOZA  
Magister en Ingeniería Biomédica

Universidad de Pamplona  
Facultad de Ingenierías y arquitectura  
Departamento de Ingenierías Mecánica, Mecatrónica e industrial  
Pamplona, Colombia  
2020

## *Dedicatoria*

*Esta tesis está dedicada principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.*

*A mi madre, quien ha tomado el rol de ser padre y madre, por su amor, trabajo y sacrificio en todos estos años, gracias a ti he logrado llegar hasta aquí y convertirme en lo que soy. Ha sido el orgullo y el privilegio de ser tú hijo, Eres la mejor madre.*

*A toda mi familia especialmente a mi abuela Romelia, mi abuelo Cornelio y mi a tía Diana, porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.*

## **Agradecimientos**

Al Ph.D. César Peña, quien como director de tesis presto su tiempo, conocimiento y motivación me oriento en la investigación.

Al M.Sc. Luis Enrique Mendoza, quien como codirector de tesis presto su tiempo, sabiduría y comprensión me oriento en la investigación.

A la Universidad de Pamplona, mi segunda casa, por haberme brindado tantas oportunidades y enriquecerme en conocimiento.

Agradezco a los todos docentes que, con su sabiduría, conocimiento y apoyo, motivaron a desarrollarme como persona y profesional.

Finalmente agradezco a todas mis amistades, especialmente a Angie Sánchez, Yerly Acosta, Yalena Arenas, Silvia Jiménez, Haiveth Suarez, Josué Barrera, Juan Calderón, Geovany Fuentes, por apoyarme cuando más los necesitaba, por la paciencia, por extender su mano en momentos difíciles y por el amor brindado cada día.

## Resumen

El análisis de la marcha humana, hace parte fundamental hoy en día en la correlación en donde se puede tener alteraciones en los diferentes sistemas corporales. A través del tiempo los investigadores les han despertado interés, en determinar los parámetros particulares del patrón de marcha en patologías específicas; una de ellas, está en diagnosticar en la pérdida de extremidades inferiores. Por ello, el objetivo principal de esta investigación es en obtener parámetros de la marcha humana usando un traje en extremidades inferiores para generar la referencia del control cinemático de una prótesis inferior. Se diseñó e implementó un traje sensorizado con el fin de analizar la marcha humana. Así pues, se relacionó los movimientos de las dos extremidades inferiores de una persona sana, con el fin de generar un control cinemático que permita reflejar un movimiento más natural en pacientes que usen prótesis; empleando sensores ubicados en el traje en las principales articulaciones y de esa manera permitió detectar los movimientos de las extremidades inferiores del paciente. Para las tomas de datos se realizó por medio de una tarjeta de desarrollo para garantizar la movilidad del paciente. Una vez obtenido los movimientos se procedió a emplear un sistema inteligente por medio de RNA utilizando Regresión Generalizada (GRNN); en tal sentido se obtuvo una eficiencia de aciertos mayor al 90 %, donde esta red logra ser muy eficiente y ligera ya que contiene una función de base radial suavizado. Por último, se consiguió así generar las referencias del control cinemático para una prótesis tomando como base las señales de la extremidad inferior opuesta de la pierna sana.

**Palabras claves:** marcha humana, sensorizado, tarjeta de desarrollo, control cinemático, sistema inteligente, regresión generalizada.

### Abstract

The analysis of human gait is a fundamental part today in the correlation where there can be alterations in the different body systems. Over time, researchers have aroused their interest in determining the particular parameters of the gait pattern in specific pathologies; one of them is to diagnose in the loss of lower extremities. Therefore, the main objective of this research is to obtain parameters of human gait using a suit on the lower extremities to generate the kinematic control reference for a lower prosthesis. A sensorized suit was designed and implemented in order to analyze human gait. Thus, the movements of the two lower extremities of a healthy person were related, in order to generate kinematic control that reflects a more natural movement in patients using prostheses; using sensors located in the suit in the main joints and in this way it was possible to detect the movements of the patient's lower extremities. For data collection, it was performed using a development card to guarantee patient mobility. Once the movements were

obtained, an intelligent system was used by means of RNA using Generalized Regression (GRNN); In this sense, a hit efficiency greater than 90% was obtained, where this network manages to be very efficient and light since it contains a smooth radial basis function. Finally, it was thus possible to generate the kinematic control references for a prosthesis based on the signals from the opposite lower extremity of the healthy leg.

**Keywords:** human gait, sensorized, development board, kinematic control, intelligent system, generalized regression.

# Contenido

	Pág.
<b>RESUMEN .....</b>	<b>1</b>
<b>LISTAS DE FIGURAS .....</b>	<b>1</b>
<b>LISTAS DE TABLAS .....</b>	<b>1</b>
<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1 PLANTEAMIENTO DEL PROBLEMA .....	3
1.2 JUSTIFICACIÓN .....	4
1.3 DELIMITACIÓN .....	5
1.3.1 <i>Objetivo general</i> .....	5
1.3.2 <i>Objetivos específicos</i> .....	5
1.3.3 <i>Acotaciones</i> .....	5
<b>2. MARCO TEÓRICO .....</b>	<b>6</b>
2.1 MARCHA HUMANA.....	6
2.2 PLANOS DE REFERENCIA.....	6
2.3 CICLO DE LA MARCHA HUMANA .....	7
2.4 FASES DE LA MARCHA HUMANA.....	8
2.4.1 <i>La fase de Aceptación de peso</i> .....	9
2.4.2 <i>La fase Soporte de una sola extremidad</i> .....	10
2.4.3 <i>Avance de miembros</i> .....	11
2.5 SENSOR FLEX .....	12
2.6 TARJETAS DE DESARROLLO .....	13
2.6.1 <i>El Arduino Nano</i> .....	14
2.6.2 <i>Módulo WiFi - ESP8266</i> .....	15
2.6.3 <i>NodeMcu ESP 32 WiFi + Bluetooth</i> .....	16
2.6.4 <i>WEMOS D1 ESP8266 WIFI</i> .....	17
2.6.5 <i>Nodemcu lua wifi basado en ESP8266</i> .....	19
2.6.6 <i>SparkFun ESP32 Thing</i> .....	20
2.7 TABLA COMPARATIVA .....	21
2.8 ANÁLISIS DE LA MARCHA HUMANA.....	22
2.9 INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL (IA).....	23
2.9.1 <i>Redes Neuronales</i> .....	23
2.9.2 <i>Características</i> .....	25
2.9.3 <i>Ventajas y Desventajas</i> .....	25
2.9.4 <i>Tipos de Redes Neuronales Artificiales</i> .....	26
2.9.5 <i>Formas de aprendizaje</i> .....	27
2.10 APLICACIONES EN DE DESARROLLO INTELIGENTE.....	28
2.11 ANÁLISIS CINEMÁTICO DE LA MARCHA HUMANA EN EL PLANO SAGITAL.....	29
2.11.1 <i>Intervalo I</i> .....	29
2.11.2 <i>Intervalo II</i> .....	32
2.11.3 <i>Intervalo III</i> .....	34
2.12 ESTUDIOS CINEMÁTICOS DE MIEMBROS INFERIORES .....	36
<b>3. METODOLOGÍA .....</b>	<b>37</b>
3.1 DISEÑO ELECTRÓNICO.....	37

3.1.2 Elección de componentes electrónicos.....	37
3.1.2 Caracterización del Sensor Flex.....	44
3.1.3 Diseño de la placa PCB.....	47
3.2 DISEÑO E IMPRESIÓN DEL CHASIS.....	52
3.3 DISEÑO DEL TRAJE.....	54
3.4 COMUNICACIÓN DE TARJETA NODEMCU ESP 32.....	56
3.4.1 Servidor UDP.....	60
3.4.2 Cliente UDP.....	61
3.5 CINEMÁTICA DIRECTA.....	67
3.6 DISEÑO DE ALGORITMO.....	70
3.5.2 Diagrama de flujo del algoritmo.....	73
3.6 ESQUEMA DE CONTROL DE LA RNA.....	74
<b>4. ANÁLISIS DE RESULTADOS.....</b>	<b>75</b>
4.1 RESPUESTA DE LA RNA1 CON LOS DATOS DE PRUEBA 2.....	75
4.2 RESPUESTA DE LA RNA1 CON LOS DATOS DE PRUEBA 3.....	77
4.3 RESPUESTA DE LA RNA1 CON LOS DATOS DE PRUEBA 4.....	79
4.4 COMPARACIÓN DE GRAFICAS.....	81
4.4.1 Graficas de movimiento de marcha de los valores de la prueba 2 del sensor 4 y 5 vs respuesta RNA1.....	82
4.4.2 Graficas de movimiento de marcha de los valores de la prueba 3 del sensor 4 y 5 vs respuesta RNA1.....	84
4.4.3 Graficas de movimiento de marcha de los valores de la prueba 4 del sensor 4 y 5 vs respuesta RNA1.....	86
4.5 PRUEBAS CON OTRA RED NEURONAL.....	89
4.6 ANÁLISIS ESTADÍSTICOS.....	91
<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>94</b>
5.1 CONCLUSIONES.....	94
5.2 RECOMENDACIONES.....	95
<b>6. REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>96</b>
<b>7. ANEXOS.....</b>	<b>101</b>
7.1 ANEXO I: CÓDIGO IDE DE ARDUINO PARA CARACTERIZAR EL SENSOR FLEX.....	101
7.2 ANEXO II: CÓDIGO IDE DE ARDUINO PARA SERVIDOR UDP.....	102
7.3 ANEXO III: CÓDIGO EN MATLAB PARA CLIENTE UDP.....	109
7.4 ANEXO IV: CÓDIGO SIMULACIÓN DE PRUEBA.....	110
7.4.1 función graficar.....	110
7.4.2 simulación.....	111
7.5 ANEXO V: RED NEURONAL DE REGRESIÓN GENERALIZADA.....	111

## Listas de Figuras

	Pág.
FIGURA 2.1: POSICIÓN ANATÓMICA CON PLANOS DE REFERENCIA.....	7
FIGURA 2.2: CICLO DE MARCHA Y SUS FASES .....	8
FIGURA 2.3: PERIODO DE “ <i>WEIGHT ACCEPTANCE</i> ” .....	9
FIGURA 2.4: PERIODO “ <i>SINGLE LIMB SUPPORT</i> ” .....	10
FIGURA 2.5: PERIODO “ <i>LIMB ADVANCEMENT</i> ” .....	11
FIGURA 2.6: SENSORES FLEXIBLES RESISTIVOS .....	13
FIGURA 2.7: ESQUEMA DE UN MICROCONTROLADOR .....	13
FIGURA 2.8: ARDUINO NANO V3.0 .....	15
FIGURA 2.9: MÓDULO WIFI - ESP8266 .....	16
FIGURA 2.10: TARJETA DE DESARROLLO ESP 32 WIFI + BLUETOOTH.....	17
FIGURA 2.11: WEMOS D1 ESP8266 WIFI.....	18
FIGURA 2.12: NODEMCU LUA WIFI BASADO EN ESP8266 .....	20
FIGURA 2.13: SPARKFUN ESP32 THING.....	21
FIGURA 2.14: ESTRUCTURA DE UNA RED NEURONAL ARTIFICIAL .....	24
FIGURA 2.15: ESTUDIO CINEMÁTICO DEL TOBILLO EN EL INTERVALO I DE LA MARCHA	30
FIGURA 2.16: ESTUDIO CINEMÁTICO DE LA RODILLA EN EL INTERVALO I DE LA MARCHA .....	31
FIGURA 2.17: ESTUDIO CINEMÁTICO DE LA CADERA EN EL INTERVALO I DE LA MARCHA .....	32
FIGURA 2.18: ESTUDIO CINEMÁTICO DEL TOBILLO EN EL INTERVALO II DE LA MARCHA	33
FIGURA 2.19: ESTUDIO CINEMÁTICO DE LA RODILLA EN EL INTERVALO II DE LA MARCHA .....	33
FIGURA 2.20: ESTUDIO CINEMÁTICO DE LA CADERA EN EL INTERVALO II DE LA MARCHA .....	34
FIGURA 2.21: ESTUDIO CINEMÁTICO DE LAS ARTICULACIONES EN EL INTERVALO III DE LA MARCHA .....	35
FIGURA 3.1: ESQUEMA DE TRABAJO DEL SENSOR FLEX.....	38
FIGURA 3.2: ADAPTACIÓN DE SENSOR FLEX HACIA TRAJE .....	38
FIGURA 3.3: DIAGRAMA DE PINES ESP-WROOM-32S.....	39
FIGURA 3.4: BATERÍA LIPO.....	40
FIGURA 3.5: REGULADOR DE VOLTAJE PARA LIPO. ....	40
FIGURA 3.6: .....	42
FIGURA 3.7: ESQUEMA DE CONEXIONES DE LA TARJETA ESP32 CON EL CONVERTIDOR	43

FIGURA 3.8: ESQUEMA PARA OBTENER LA SEÑAL DEL SENSOR [AUTOR]. .....	44
FIGURA 3.9: GRAFICA DE RESPUESTA DEL VALOR MÁXIMO [AUTOR]. .....	46
FIGURA 3.10: GRAFICA DE RESPUESTA DEL VALOR EFICAZ [AUTOR]. .....	46
FIGURA 3.11: GRAFICA DE RESPUESTA DEL VALOR MEDIO [AUTOR] .....	46
FIGURA 3.12: ESQUEMA DE IDENTIFICACIÓN DE PINES PARA LA TARJETA [AUTOR]. .....	47
FIGURA 3.13: ESQUEMA ELECTRÓNICO DE NODEMCU ESP 32 [AUTOR]. .....	48
FIGURA 3.14: ESQUEMA ELÉCTRICO EN ISIS [AUTOR]. .....	48
FIGURA 3.15: ESQUEMA DEL CIRCUITO EN ARES .....	49
FIGURA 3.16: ALINEACIÓN DEL ESQUEMA Y EL CAD DE LA TARJETA EN 3D. ....	50
FIGURA 3.17: ALINEACIÓN COMPLETADA CON EL CAD DE LA TARJETA [AUTOR]. .....	51
FIGURA 3.18: VISUALIZACIÓN 3D DE PCB TERMINADO [AUTOR]. .....	51
FIGURA 3.19: RESULTADO FINAL DE LA PLACA ELECTRÓNICA [AUTOR]. .....	52
FIGURA 3.20: DISEÑO CAD DEL CHASIS.....	52
FIGURA 3.21: DISEÑO CAD DE LA TAPA DEL CHASIS.....	53
FIGURA 3.22: DISEÑO CAD DEL CHASIS COMPLETO .....	53
FIGURA 3.23: CHASIS DEL PROTOTIPO IMPRESO EN 3D.....	54
FIGURA 3.24: RESULTADO FINAL DE LA PLACA Y EL CHASIS [AUTOR]. .....	54
FIGURA 3.25: VISTA FRONTAL DEL TRAJE CON LOS SENSORES FLEX .....	55
FIGURA 3.26: VISTA TRASERA DEL TRAJE CON EL CANGURO DE CONTROL.....	56
FIGURA 3.27: VISTA DE LOS ARCHIVOS PARA DESCARGAR FLEX .....	57
FIGURA 3.28: VISTA DEL IDE ARDUINO PARA SELECCIÓN DE LA PLACA.....	58
FIGURA 3.29: VISTA DEL IDE ARDUINO PARA SELECCIÓN DE HERRAMIENTAS.....	58
FIGURA 3.30: ESQUEMA DE SERVICIO UDP [45] .....	60
FIGURA 3.31: LECTURA DE LOS SENSORES USANDO IDE ARDUINO [AUTOR]. .....	61
FIGURA 3.32: ACTIVAR O DESACTIVAR FIREWALL DE WINDOWS DEFENDER [AUTOR]. ..	62
FIGURA 3.33: CAPTURA DE PAQUETES UDP GUARDADO EN MATLAB [AUTOR]. .....	63
FIGURA 3.34: TABLA DE LOS DATOS CAPTURADOS DEL SERVIDOR UDP [AUTOR]. .....	63
FIGURA 3.35: GRAFICA DEL MOVIMIENTO DE LA CADERA DERECHA [AUTOR]. .....	64
FIGURA 3.36: GRAFICA DEL MOVIMIENTO DE LA CADERA IZQUIERDA [AUTOR]. .....	64
FIGURA 3.37: GRAFICA DEL MOVIMIENTO DE LA RODILLA IZQUIERDA [AUTOR]. .....	65
FIGURA 3.38: GRAFICA DEL MOVIMIENTO DE LA RODILLA DERECHA [AUTOR]. .....	65
FIGURA 3.39: GRAFICA DEL MOVIMIENTO DEL TOBILLO DERECHO [AUTOR]. .....	66
FIGURA 3.40: GRAFICA DEL MOVIMIENTO DEL TOBILLO IZQUIERDA [AUTOR]. .....	66
FIGURA 3.41: GEOGRAFÍA DEL MECANISMO [AUTOR]. .....	67

FIGURA 3.42: GRAFICA DE PRUEBA 1[AUTOR].	68
FIGURA 3.43: GRAFICA DE PRUEBA 2[AUTOR].	69
FIGURA 3.44: GRAFICA DE PRUEBA 3[AUTOR].	69
FIGURA 3.45: GRAFICA DE PRUEBA 4[AUTOR].	70
FIGURA 3.46: ARQUITECTURA DEL GNN [46].	70
FIGURA 3.47: DATOS DE NUEVAS VARIABLES GUARDADOS EN WORKSPACE [AUTOR].	72
FIGURA 3.48: DATOS DE ENTRENAMIENTO (S1236) GUARDADOS EN WORKSPACE [AUTOR].	72
FIGURA 3.49: DATOS DE SALIDA DESEADA (S45) GUARDADOS EN WORKSPACE [AUTOR].	72
FIGURA 3.50: DIAGRAMA DE FLUJO DEL DISEÑO DE ALGORITMO [AUTOR].	73
FIGURA 3.51: BLOQUES DE SIMULINK DE LA RNA [AUTOR].	74
FIGURA 3.52: ESQUEMA DE LAZO CONTROL CERRADO [AUTOR].	74
FIGURA 4.1: GRÁFICA DE ERROR ABSOLUTO DE RNA1 [AUTOR].	75
FIGURA 4.2: GRÁFICA DEL ERROR ABS DE LA PRUEBA 2 DE VALIDACIÓN CON LA RNA1 [AUTOR].	76
FIGURA 4.3: GRÁFICA DE UN FRAGMENTO DEL ERROR ABS DE LA PRUEBA 2 DE VALIDACIÓN CON LA RNA1 [AUTOR].	77
FIGURA 4.4: GRÁFICA DEL ERROR ABS DE LA PRUEBA 3 DE VALIDACIÓN CON LA RNA1 [AUTOR].	78
FIGURA 4.5: GRÁFICA DE UN FRAGMENTO DEL ERROR ABS DE LA PRUEBA 3 DE VALIDACIÓN CON LA RNA1 [AUTOR].	79
FIGURA 4.6: GRÁFICA DEL ERROR ABS DE LA PRUEBA 4 DE VALIDACIÓN CON LA RNA1 [AUTOR].	80
FIGURA 4.7: GRÁFICA DE UN FRAGMENTO DEL ERROR ABS DE LA PRUEBA 4 DE VALIDACIÓN CON LA RNA1 [AUTOR].	81
FIGURA 4.8: MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 2 VS RNA1 [AUTOR].	82
FIGURA 4.9: FRAGMENTO DE MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 2 VS RNA1 [AUTOR].	82
FIGURA 4.10: MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 2 VS RNA1 [AUTOR].	83
FIGURA 4.11: FRAGMENTO DE MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 2 VS RNA1 [AUTOR].	83
FIGURA 4.12: MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 3 VS RNA1 [AUTOR].	84
FIGURA 4.13: FRAGMENTO DE MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 3 VS RNA1. [AUTOR].	84
FIGURA 4.14: MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 3 VS RNA1 [AUTOR].	85
FIGURA 4.15: FRAGMENTO DE MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 3 VS RNA1 [AUTOR].	85

<b>FIGURA 4.16: MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 4 VS RNA1 [AUTOR].</b>	<b>86</b>
<b>FIGURA 4.17: FRAGMENTO DE MOVIMIENTO DE LA RODILLA DURANTE LA PRUEBA 4 VS RNA1. [AUTOR].</b>	<b>86</b>
<b>FIGURA 4.18: MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 4 VS RNA1 [AUTOR].</b>	<b>87</b>
<b>FIGURA 4.19: FRAGMENTO DE MOVIMIENTO DEL TOBILLO DURANTE LA PRUEBA 4 VS RNA1 [AUTOR].</b>	<b>87</b>
<b>FIGURA 4.20: PUNTOS PRINCIPALES DEL MOVIMIENTO ANGULAR DE LA RODILLA [AUTOR].</b>	<b>88</b>
<b>FIGURA 4.21: PUNTOS PRINCIPALES DEL MOVIMIENTO ANGULAR DEL TOBILLO [AUTOR].</b>	<b>88</b>
<b>FIGURA 4.22: GUI DE ENTRENAMIENTO DE REDES NEURONALES K ITERACIONES.</b>	<b>89</b>
<b>FIGURA 4.23: GUI DE ENTRENAMIENTO DE REDES NEURONALES</b>	<b>90</b>
<b>FIGURA 4.24: DIAGRAMA DE BARRAS DEL PORCENTAJE DE ACIERTO PRUEBA 2 DE LA RNA1 VS LOS VALORES SENSORES 4 Y 5 [AUTOR].</b>	<b>91</b>
<b>FIGURA 4.25: DIAGRAMA DE BARRAS DEL PORCENTAJE DE ACIERTO PRUEBA 3 DE LA RNA1 VS LOS VALORES SENSORES 4 Y 5 [AUTOR].</b>	<b>91</b>
<b>FIGURA 4.26: DIAGRAMA DE BARRAS DEL PORCENTAJE DE ACIERTO PRUEBA 4 DE LA RNA1 VS LOS VALORES SENSORES 4 Y 5 [AUTOR].</b>	<b>92</b>

## Listas de Tablas

	Pág.
TABLA 2.1: COMPARACIÓN DE TARJETAS DE DESARROLLO.....	21
TABLA 2.2: CARACTERÍSTICAS DE LAS RNA .....	25
TABLA 2.3: MOVIMIENTO DEL TOBILLO INTERVALO I .....	29
TABLA 2.4: MOVIMIENTO DE LA RODILLA INTERVALO I.....	30
TABLA 2.5: MOVIMIENTO DE CADERA INTERVALO I .....	31
TABLA 2.6: MOVIMIENTO DEL TOBILLO INTERVALO II .....	32
TABLA 2.7: MOVIMIENTO DE LA RODILLA INTERVALO II.....	33
TABLA 2.8: MOVIMIENTO DE LA CADERA INTERVALO II .....	34
TABLA 2.9: MOVIMIENTO DEL TOBILLO INTERVALO III .....	34
TABLA 2.10: MOVIMIENTO DE LA RODILLA INTERVALO III.....	35
TABLA 2.11: MOVIMIENTO DE LA CADERA INTERVALO III .....	35
TABLA 3.1: CARACTERÍSTICAS DEL CONVERTIDOR .....	41
TABLA 3.2: CARACTERÍSTICAS DE LOS PINES DEL CONVERTIDOR ADS1115 16 BIT. ....	43
TABLA 3.3: VALORES RECOLECTADOS DE UNO DE LOS 6 SENSORES. ....	45
TABLA 3.4: DURACIÓN DE TIEMPO PARA TOMA DE DATOS.....	63
TABLA 4.1: ÍNDICES ESTADÍSTICOS DE LA PRUEBA 2 .....	92
TABLA 4.2: ÍNDICES ESTADÍSTICOS DE LA PRUEBA 3 .....	93
TABLA 4.3: ÍNDICES ESTADÍSTICOS DE LA PRUEBA 4 .....	93

### 1. Introducción

La presente investigación, se centra en la demanda de equipos de rehabilitación donde ha permitido captar el interés a investigadores al emplear nuevas tecnologías y estrategias de control para componer novedosas prótesis que permitan rehabilitar, reemplazar y aumentar las capacidades físicas de una persona que sufre alguna discapacidad.

Por otro lado, el estudio de la marcha se puede definir, como la forma de desplazamiento en posición bípeda propia del ser humano en la que se suceden apoyos bipodales y los monopodales. Además, aquella serie de movimientos alternantes y rítmicos de las extremidades y del tronco, la cual determinan un desplazamiento hacia delante del centro de gravedad con un mínimo gasto de energía [1]. Por lo tanto, el control cinemático está encargada de calcular la secuencia de posiciones y velocidades que deben seguir las articulaciones o grados de libertad de un robot u otro prototipo para seguir una trayectoria deseada [2].

El estudio de la marcha ha interesado desde tiempos remotos como: Aristóteles (384-322 a. C.), Hipócrates (460-377 a. C.) y Galeno (129-201) que dieron gran importancia al estudio del movimiento y en particular al estudio de la marcha [3]. Los métodos para su evaluación han avanzado mucho, la cual se han perfeccionado y simplificado las técnicas para su análisis. Al mismo tiempo han desarrollado nuevos métodos (videogrametría, electromiografía, dinamometría, mioeléctricos, etc.).

La característica principal de esta investigación, es que el ser humano necesita conseguir una postura estable en bipedestación antes de iniciar la marcha. En la posición bípeda la estabilidad mecánica se basa en el soporte musculoesquelético que se mantiene gracias a los reflejos posturales y a la integración de los estímulos aferentes visuales, vestibulares y propioceptivos [4]. Para este caso se centra al control y manejo de las futuras prótesis para tener una buena postura al momento de ejercer un desplazamiento más estable para el paciente.

Esta investigación, se enfoca al control de movimiento en nuevas prótesis en donde se ha convertido en una necesidad debido al crecimiento del número de amputaciones originadas por diversas causas como: accidentes, desastres naturales, guerras, enfermedades crónicas, etc. Al igual que la tecnología médica en general, el desarrollo tecnológico de las prótesis ortopédicas no ha parado y busca imitar lo más posible los miembros naturales del cuerpo [5].

## Introducción

---

Por último, para la parametrización de a marcha humana, es recurrente tener unos datos de estudio para ver el comportamiento de la persona al desplazarse. Al mismo modo, para poder hacer este estudio, es necesario tener unas tomas de datos. Así pues, para esta investigación se llevó acabo con la ayuda de una tarjeta de desarrollo, la cual facilita en recibir la lectura de los datos por medio del ordenador con la ayuda esencial de un sensor; que en este caso permite recibir datos de los movimientos de cada pierna, tanto una persona sana o con amputaciones.

Un sistema inteligente, nace con la intención de aplicar a control las técnicas de Inteligencia artificial, las cuales comenzó a tener una serie de resultados y se consolidó como disciplina. La "inteligencia artificial" en sí es un campo amplio, que abarca lógica, optimización, probabilidad, percepción, razonamiento, toma de decisiones y aprendizaje. El objetivo es integrar agentes inteligentes en los sistemas de control [6]. Tales como ayudar a generar las referencias del control cinemático para una prótesis en general.

### **1.1 Planteamiento del problema**

En el área de la salud en la parte biomédica existen múltiples métodos y tecnologías para el análisis de marcha humana en el cual resulta tener grandes aplicaciones; en donde se logra obtener parámetros durante el análisis de marcha fácilmente, de esa manera hay muchos factores que se deben considerar para estudios adecuados. Sin embargo, existe algunos de los métodos para la toma de datos que puede ser muy costosos o delicados, unas de las técnicas que se desarrollan son: videogrametría, electromiografía, dinamometría, mioeléctricos, etc.

Teniendo en cuenta las metodologías de las técnicas que se desarrollan en la actualidad se evidencian algunos puntos críticos, tales como; en posicionamiento de marcadores, fuentes de errores en los parámetros, calibración de cámaras; en la modificación de lugar, en la poca conexión estable de los electrodos de almohadilla; a los movimientos bruscos que genera el paciente, a la incorrecta detección de puntos anatómicos en la colocación de marcadores; en donde es difícil para los pacientes con condiciones patológicas. [7]

Dado a conocer las diferentes maneras para el desarrollo en la toma de datos y estudios de la marcha humana; se propone realizar una opción en manera de estudio con la facilidad en obtener parámetros de la marcha y de bajo presupuesto para determinar la viabilidad de trabajo para todo tipo de pacientes. Además, se logre desarrollar proyectos de investigación en la parte biomédica.

### **1.2 Justificación**

El estudio de análisis de marcha ha sido importante para el desarrollo de una diversidad de aplicaciones en algunos campos de salud. En la actualidad se han tomado consideración al sistema de análisis de movimiento humano como prioridad al momento de estudiar y diagnosticar el desempeño del paciente [8].

El desarrollo de esta investigación, tiene como objetivo apoyar a la comunidad de especialistas en biomédica que trabajen en el área de prótesis inferiores, para fortalecer el estudio a los pacientes. Ya que, de manera objetiva se logre parametrizar el estudio de movimiento, y que en un futuro se podrá apreciar los factores que pueden modificar, diagnosticar alteraciones del patrón de marcha en diversas patologías y lesiones traumáticas, y realizar un control y seguimiento de pacientes para observar la evolución de los mismas, valorar la efectividad del tratamiento, recuperación tras una intervención quirúrgica, necesidad de ayudas técnicas, etc. [9] De esta forma se desea implantar una manera de generar las referencias del control cinemático de una prótesis de miembro inferior.

Partiendo de este estudio, el aporte primordial es caracterizar la marcha humana basada en las señales obtenidas por los sensores, con el fin de relacionar los movimientos de las articulaciones de una extremidad con los movimientos de la otra extremidad; por medio de un traje sensorizado. Por último, en la actualidad ya se están trabajando este método con sistema inteligente, para el desarrollo de una prótesis [33],[34].

### **1.3 Delimitación**

#### **1.3.1 Objetivo general**

- Obtener parámetros de la marcha humana usando un traje en extremidades inferiores para generar la referencia del control cinemático de una prótesis.

#### **1.3.2 Objetivos específicos**

- Implementar un traje sensorizado para el registro de los movimientos de las extremidades inferiores de una persona.

- Caracterizar la marcha humana basada en las señales obtenidas por los sensores, con el fin de relacionar los movimientos de las articulaciones de una extremidad con los movimientos de la otra extremidad.

- Desarrollar un sistema de inteligencia artificial que permita generar las referencias del control cinemático de una prótesis.

#### **1.3.3 Acotaciones**

- En la evaluación del proyecto se trabajará con pacientes entre 18 a 25 años de edad aproximadamente; para recolectar cierta cantidad de datos del movimiento al caminar en línea recta sin obstáculos. Mediante la obtención de las señales articulares usando los sensores de flexión por medio del traje.

-En el desarrollo del estudio, se reitera que no se va hacer una prótesis. Si no un estudio de referencia para control cinemático para que en un futuro se desarrolle una prótesis.

-Se pretende trabajar con sensores flexibles, sin embargo, se puede trabajar con otros sensores.

-La selección de los elementos electrónicos, se podrá escoger algunos de bajos costos, dado a los recursos del autor para desarrollo del proyecto.

### 2. Marco teórico

Este capítulo contiene los fundamentos correspondientes tanto a la marcha humana, aplicaciones que se han trabajado respecto a la temática, métodos de sistema inteligente, sensor que se va trabajar, métodos que han desarrollado para el control de movimiento del sistema articulados, tipos de tarjeta de desarrollo.

Por otro lado, se aclara que la información teórica consultada no se limita a este capítulo, dado que se hace mención a otros contextos, métodos en los otros capítulos siguientes.

#### 2.1 Marcha Humana

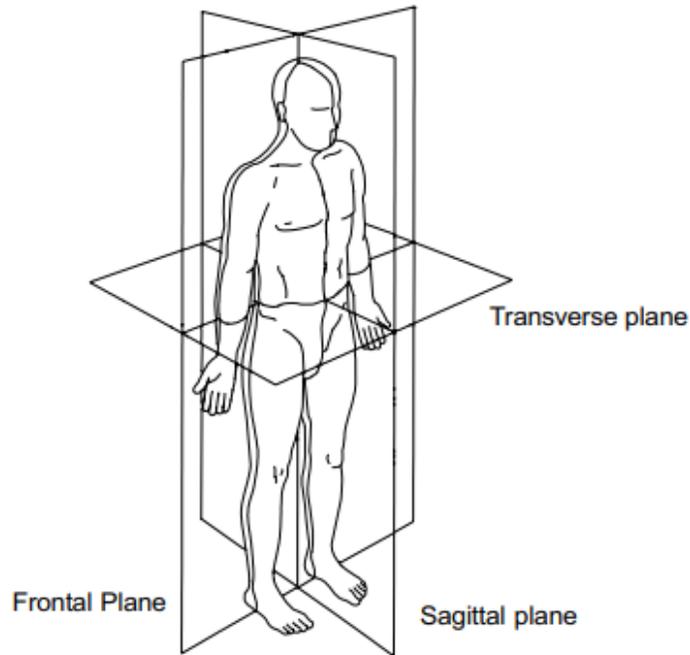
El análisis del movimiento humano y más concretamente de la marcha, ha dado curiosidad a muchos desde tiempos remotos, como Aristóteles, Leonardo da Vinci, Hipócrates entre otros [10],[11] llevando al desarrollo de diferentes métodos para su estudio. Como va avanzando el tiempo, la introducción de los sistemas computacionales y los avances en la informática, se ha logrado obtener los modernos y sofisticados sistemas para análisis del movimiento humano disponibles en la actualidad. Por ello, en un documento denominado “Laboratorio de análisis de marcha y movimiento” presenta como objetivo describir parte de la historia del desarrollo del análisis instrumentado de la marcha. Sus utilidades en el campo de la docencia, investigación y especialmente sus aplicaciones clínicas. Se describe el proceso de adquisición, requisitos para solicitud del estudio y sus aportes en el campo clínico en cuanto a definir perfil de comportamiento en el tiempo de diferentes enfermedades, tomar decisiones terapéuticas, controlar sus resultados en el tiempo y lograr una mejor optimización de los recursos [12].

#### 2.2 Planos de Referencia

Para centrar en el estudio de la marcha humana primordialmente ahí que tener en cuenta la planimetría del cuerpo humano. Por ello, se presta especial atención a los movimientos de la pierna, ya que es la parte más implicada en la marcha humana y en contacto directo con ortesis, prótesis, entre otros.

Los planos de referencia habituales son los que se muestran en la Figura 2.1. Se definen tres planos, estando el sujeto de pie y con la espalda recta: el plano sagital (secciona el cuerpo en su parte derecha e izquierda), plano frontal (secciona el cuerpo en parte anterior y posterior) y plano transversal (divide el cuerpo en parte superior e inferior) [13].

Para este concepto y teniendo en cuenta que el movimiento de la marcha se produce esencialmente en el plano sagital, y es donde se esperan las máximas fuerzas de contacto, será considerado el plano de movimiento principal en el presente estudio.

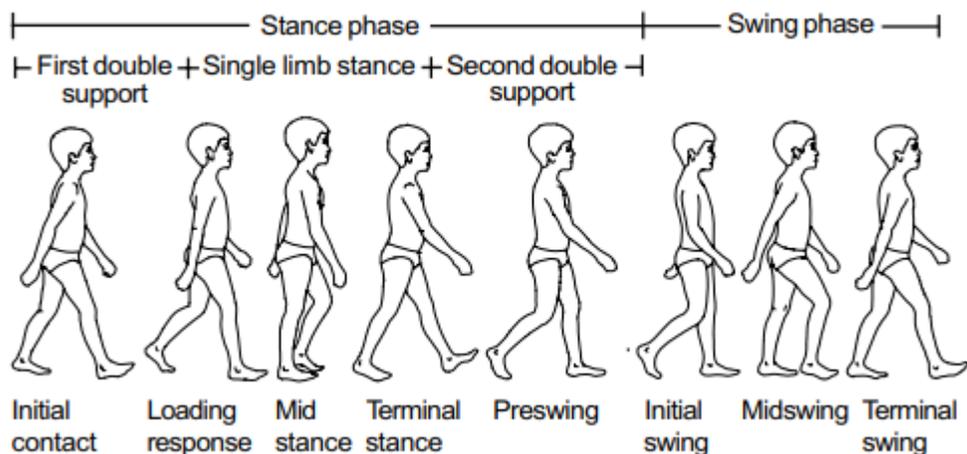


**Figura 2.1:** Posición anatómica con planos de referencia. [13]

### 2.3 Ciclo de la marcha humana

En el ciclo de marcha y fase, cada pierna pasa por una fase de apoyo durante la cual el pie se encuentra en contacto total o parcialmente con el suelo, seguido por una fase de oscilación, en la cual el pie se encuentra en el aire, al tiempo que avanza. La fase de apoyo comienza cuando el talón está en contacto con el suelo y finaliza cuando los dedos pierden el contacto con él; la fase de oscilación transcurre desde cuando el antepié se despega del suelo hasta cuando el pie se apoya en el talón nuevamente [13]. Como se ve en la figura 2.2, la fase de postura puede subdividirse en tres fases separadas:

1. Primer doble soporte, cuando ambos pies están en contacto con el suelo.
2. Posición de una sola extremidad, cuando el pie izquierdo se balancea y solo el pie derecho está en contacto con el suelo.
3. Segundo soporte doble, cuando ambos pies vuelven a estar en contacto con el suelo.



**Figura 2.2:** Ciclo de marcha y sus fases. [13]

El ciclo de la marcha se divide en dos fases, la fase de apoyo (Stance Phase) y la fase de oscilación o balanceo (Swing Phase). Durante la primera, el pie está en contacto con el suelo. Esta etapa ocupa alrededor del 62% del ciclo de la marcha normal, inicia con el contacto del talón en el suelo y finaliza al separar el antepié. En la segunda fase, el pie pasa a estar suspendido en el aire desplazándose hacia adelante hasta hacer contacto con el suelo, marcando el inicio de un nuevo ciclo. Esta etapa ocupa el 38% del ciclo de la marcha [13].

### 2.4 Fases de la marcha humana

La determinación por fases de la marcha humana permite una mejor interpretación de los diferentes movimientos que se producen durante el ciclo. Para el Autor Perry [14], da a conocer ocho fases distintas que son: Contacto inicial (*Initial Contact*), respuesta de carga (*Loading Response*), mitad de la fase de apoyo (*Mid-Stance*), final de la fase de apoyo (*Terminal Stance*), pre-fase de balanceo (*Pre-Swing*), inicio de balanceo (*Initial Swing*), mitad de la fase de balanceo (*Mid-Swing*) y final de la fase de balanceo (*Terminal Swing*).

La fase de *Weight Acceptance* inicia la fase de apoyo (*Stance Phase*) y utiliza las dos primeras fases, *Initial Contact* y *Loading Response*. Por último, tenemos la fase *Single Limb Support*, dentro de la cual tenemos el *Mid-Stance* y *Terminal Stance*. Por último, *Limb Advancement* comienza con el final de la fase de apoyo, con el *Pre-Swing*, continuando con las tres fases del swing: *Initial Swing*, *Mid-Swing* y *Terminal Swing* [14].

### 2.4.1 La fase de Aceptación de peso

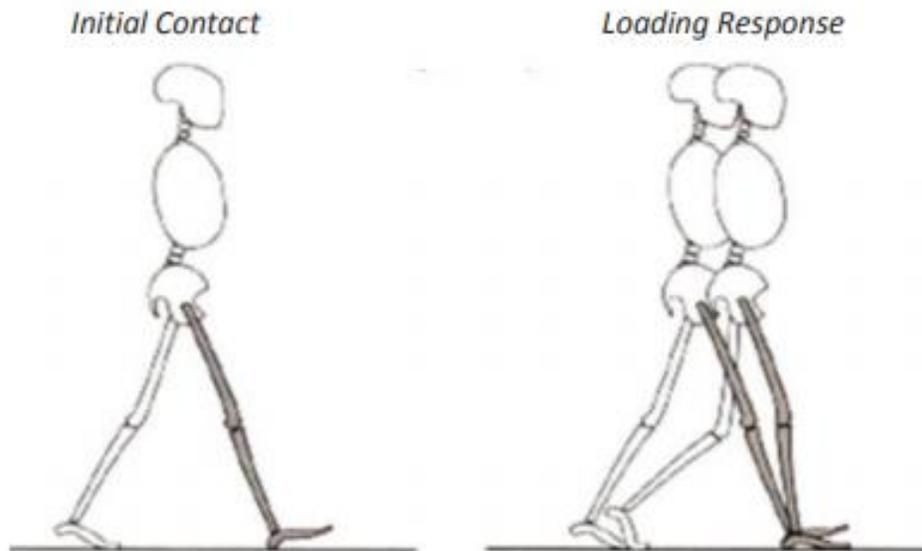
Para la aceptación del peso es la tarea más exigente, se requiere la transferencia del peso corporal sobre la pierna que acaba de terminar la fase de balanceo. Se inicia con la absorción del choque y sigue con la estabilidad y la conservación de la progresión. Por ello, estos tres patrones funcionales se dividen en dos fases: *Initial Contact* y *Loading Response*.

#### - Contacto inicial (Fase 1)

Esta fase comienza en el momento en el que el pie entra en contacto con el suelo, representando el 2% del ciclo de la marcha aproximadamente. El contacto lo realiza el talón, la cadera se flexiona y la rodilla está totalmente extendida. El tobillo está dorsiflexionado.

#### - Cargando respuesta (Fase 2)

Representa sobre el 10% del ciclo de la marcha y comienza con el contacto del pie con el suelo, continuando hasta que el otro pie se levanta entrando en la fase del balanceo. Esta fase se caracteriza por la absorción del choque debido al impacto del pie sobre el suelo y por la aceptación del peso del cuerpo. Se transfiere en esta etapa el peso corporal a la pierna de referencia y la rodilla pasa a flexionarse para la absorción del choque. La pierna opuesta está en su fase *Pre-Swing*. En la figura 2.3 se puede apreciar la configuración del cuerpo durante las dos fases anteriores.



**Figura 2.3:** Periodo de “*weight acceptance*”. [14]

### 2.4.2 La fase Soporte de una sola extremidad

En este periodo comienza cuando la pierna opuesta entra en la fase de balanceo, y continúa hasta que se llega al contacto con el suelo, por tanto, la pierna de referencia aquella que soporta el peso del cuerpo. Las siguientes fases son las implicadas en este periodo.

#### - Postura media (Fase 3)

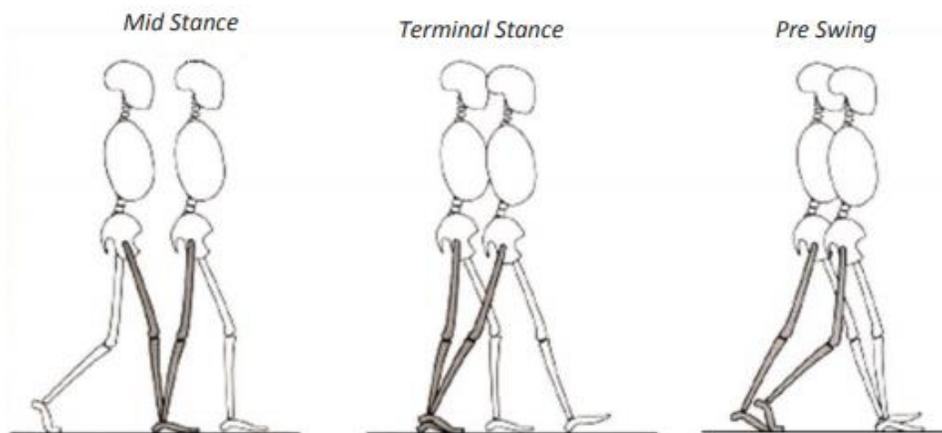
Representa la primera mitad del periodo cuando la pierna avanza sobre el pie estacionario gracias a la dorsiflexión del tobillo, con la rodilla y cadera extendidas. La pierna opuesta se encuentra avanzando en su fase de *Mid-Swing* con el tobillo moderadamente dorsiflexionado y la rodilla en extensión. Esta fase corresponde al intervalo aproximado entre el 10% y 30% del ciclo de la marcha.

#### - Postura terminal (Fase 4)

Es la segunda fase del periodo *Single Limb Support* y representa aproximadamente el intervalo entre el 30% y 50% del ciclo de la marcha. Durante esta fase, el talón se levanta, la rodilla se extiende más y luego comienza a flexionarse un poco y continúa hasta que el pie opuesto impacta sobre el suelo.

#### - Pre Oscilación (Fase 5)

Es la fase final de la fase de apoyo e inicio de la fase de balanceo. Comienza con el *Initial Contact* de la pierna opuesta y termina con la separación del pie de referencia (toe-off). Representa entre el 50% y 60% del ciclo y la pierna de referencia se descarga del peso corporal preparándose para la fase de balanceo. Ocurre un incremento de la flexión plantar del tobillo de referencia, aumenta la flexión de la rodilla y disminuye la extensión de la cadera. La pierna opuesta está en cambio en la fase *Loading Response* [14]. En la figura 2.4 las tres fases.



**Figura 2.4:** periodo “*single limb support*”. [14]

### 2.4.3 Avance de miembros

Por ultimo las fases de este periodo se muestran en la figura 2.5.

#### - Oscilación Inicial (Fase 6)

Esta fase comienza cuando el pie se levanta del suelo y termina cuando la pierna que oscila está justo frente a la pierna apoyada. Se caracteriza por un aumento de la flexión de la rodilla para evitar el roce con el suelo y por una flexión de la cadera para el avance. La pierna opuesta está en el inicio del *Mid-Stance*. Esta fase ocurre aproximadamente entre el 60% y 73% del ciclo.

#### - Oscilación media (Fase 7)

En el inicio, la pierna de oscilación está justo a la contra de la de apoyo y avanzará hasta que la pierna que balancea quede por delante y la postura de la cadera y rodilla sean iguales. Esta fase ocurre entre el 73% y 87% del ciclo y se caracteriza por un descenso de la flexión de la rodilla.

#### - Oscilación terminal (Fase 8)

Finalmente, esta fase corresponde al intervalo entre el 87% y 100% del ciclo de la marcha y se completa el avance de la pierna de referencia gracias a la extensión de la rodilla. La cadera se mantiene flexionada y el tobillo dorsiflexionado de manera neutral. La fase termina cuando el pie entra de nuevo en contacto con el suelo, comenzando un nuevo ciclo. La pierna opuesta está en su *Terminal Stance* [14].

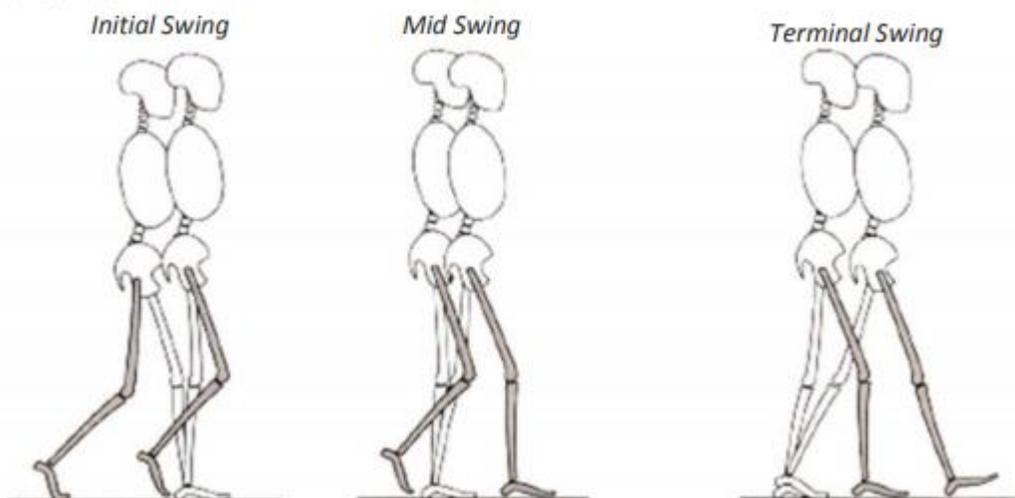


Figura 2.5: periodo "limb advancement".[14]

### 2.5 Sensor Flex

Para la parametrización de la marcha humana que genere las referencias del control cinemático de una prótesis, se necesita trabajar con sensores y uno de estos está el sensor flexible, que permite medir la cantidad de desviación o flexión. Por lo general, el sensor está pegado a la superficie y la resistencia del elemento sensor varía al doblar la superficie. Uno de los fabricantes de sensores flexibles más reconocidos esta Spectra Symbol, se enfoca en satisfacer a cada cliente al proporcionar una calidad sobresaliente y servicios excelentes. Pueden personalizar el diseño de los sensores flexibles resistivos para satisfacer a las necesidades específicas. Pueden cambiar la resistencia nominal real y producir sensores en diferentes sustratos para que sean adecuados para operaciones de alta temperatura.

- Alta resistencia a temperaturas extremas.
- Fiable y consistente
- Se puede montar en superficies estacionarias y flexibles.
- Innumerables relaciones de curvatura y posibilidades de resistencia
- Se puede diseñar a medida para que coincida con las especificaciones del cliente

1. Robótica: controlar el movimiento o la colocación de las articulaciones es fácil con sensores flexibles. También puede usar estos productos como interruptores de parachoques para una detección eficiente de la pared.

2. Bio-métrica: los fabricantes de equipos deportivos utilizan sensores flexibles para indicar la colocación o el movimiento.

3. Guantes de realidad virtual para juegos: los sensores son componentes importantes utilizados en la fabricación de guantes de realidad virtual. El Nintendo *Power Glove* es un buen ejemplo.

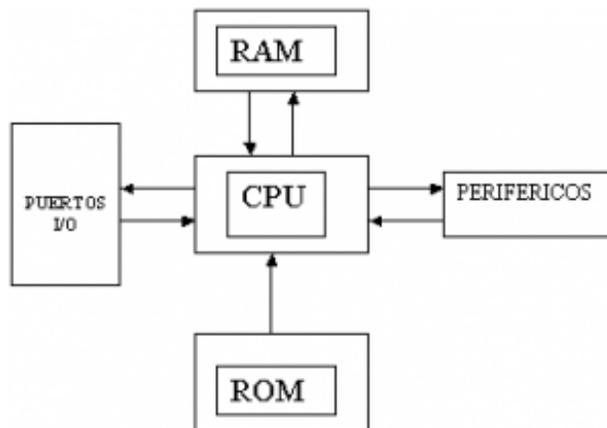
4. Otras aplicaciones: los sensores flexibles son comunes en diferentes productos tecnológicos, como controles automotrices e industriales, periféricos de computadora, joysticks y dispositivos de medición. También están presentes en productos de acondicionamiento físico, instrumentos musicales y sistemas de tecnología de asistencia de vehículos modernos [15].



**Figura 2.6:** Sensores flexibles resistivos. [15]

### 2.6 Tarjetas de desarrollo

Para el desarrollo en la toma de datos, se debe tener en cuenta que tipo de tarjeta de desarrollo se va a utilizar al momento de trabajar. Se conoce que una tarjeta está compuesta por un microcontrolador, que es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado [16].



**Figura 2.7:** Esquema de un microcontrolador.[16]

Existen varios fabricantes de microcontroladores tales como Texas Instruments, Motorola, Atmel, Intel, Microchip, Toshiba, Nacional, etc. Todos ellos ofrecen microcontroladores con características más o menos similares, sin embargo, en términos generales se puede decir que todos sirven para lo mismo: leer y ejecutar los programas del usuario. Evidentemente algunos modelos tienen más capacidad que otros, en cuanto a memoria, velocidad, periféricos, etc. En el diseño de un sistema con microcontrolador se pueden elegir dispositivos de cualquiera de los

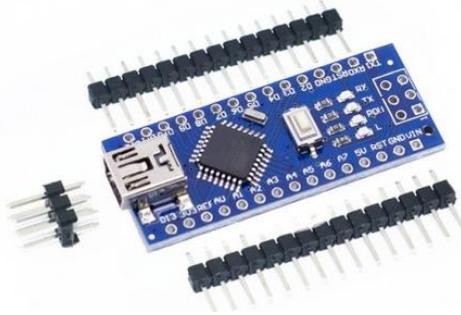
fabricantes y satisfacer la necesidad del sistema en cuestión. Se debe tomar en cuenta lo siguiente:

1. Disponibilidad de los microcontroladores en el mercado local y/o global. Algunos microcontroladores son más comúnmente encontrados en las casas de electrónica que otros. No es conveniente emprender un proyecto basado en cierto microcontrolador que escasea en el mercado local y/o global ya que podría no satisfacer nuestra demanda y detener el proyecto.
2. Disponibilidad de información y herramientas de desarrollo. Actualmente la mayoría de los fabricantes de microcontroladores ofrecen información suficiente para entender la operación y funcionamiento de sus dispositivos. El punto débil de algunos fabricantes es la pobre distribución de sus herramientas de desarrollo (programadores, emuladores, software, etc.) o bien su alto costo.
3. Precio. Hay gente y/o empresas que tienen los recursos para pagar lo que sea necesario cuando se trata de tecnología, sin embargo, la mayoría prefiere ahorrarse dinero, por lo que el precio resulta un punto a favor en la selección de un fabricante de microcontroladores.
4. El modelo de microcontrolador específico que se debe elegir depende de la aplicación, aunque uno puede hacer un juicio sobre la capacidad de un microcontrolador tomando en cuenta su capacidad de memoria, la cantidad de puertos de entrada y salida, los periféricos, la velocidad a la cual ejecuta las instrucciones, etc. [16]

Para el análisis de sistema de control, se debe trabajar con una tarjeta de desarrollo donde su funcionalidad debe ser inalámbrica que en este caso sería requerido a la toma de datos al momento de realizar el proyecto, a continuación, se dará a conocer algunas tarjetas de bajo costo que se da en el mercado con su funcionalidad y características para conocer cuál es la más recomendada para trabajar en el proyecto.

### 2.6.1 El Arduino Nano

El Arduino Nano, es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o el ATmega168 en sus versiones anteriores (Arduino Nano 2.x) que se usa conectándola a una protoboard. Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero con una presentación diferente. funciona con un cable USB Mini-B [17].



**Figura 2.8:** Arduino Nano V3.0. [17]

Especificación técnica:

- Referencia: Arduino Nano v3.0 ATMEGA328 CH340
- Microcontrolador: Atmel ATmega328 (ATmega168 versiones anteriores)
- Tensión de Operación (nivel lógico): 5 V
- Tensión de Entrada (recomendado): 7-12 V
- Tensión de Entrada (límites): 6-20 V
- Pines E/S Digitales: 14 (de los cuales 6 proveen de salida PWM)
- Entradas Analógicas: 8 Corriente máx por cada PIN de E/S: 40 mA
- Memoria Flash: 32 KB (ATmega328) de los cuales 2KB son usados por el bootloader (16 KB – ATmega168)
- SRAM: 2 KB (ATmega328) (1 KB ATmega168)
- EEPROM: 1 KB (ATmega328) (512 bytes – ATmega168)
- Frecuencia de reloj: 16 MHz

### 2.5.2 Módulo WiFi - ESP8266

El módulo inalámbrico ya viene preprogramado con un comando AT Firmware con él se podrá conectar el Arduino y hacer lo que se necesita. Uno de los muchos módulos y económicos está el **ESP8266**, que tiene un gran alcance, suficiente procesamiento y almacenamiento de datos que le permite integrarse con sensores y dispositivos a través de sus GPIOs, Su alto grado de integración en el chip permite la circuitería externa mínima, incluyendo el módulo de front-end, está diseñado para ocupar el área mínima de PCB. El ESP8266 apoya APSD para aplicaciones VoIP y las interfaces de co-Existence Bluetooth, contiene un RF auto-calibrado que permite que funcione en todas las condiciones de funcionamiento, y no requiere de partes de RF externas [18].



**Figura 2.9:** Módulo Wifi - ESP8266 [18].

Características:

- 802.11 b / g / n
- Pila de protocolos integrado TCP / IP
- Interruptor integrado TR, balun, LNA, amplificador de potencia y red de adaptación
- PLL integrados, reguladores, DCXO y unidades de administración de energía
- Velocidad de Transmisión paquetes en <2 ms
- Potencia de salida 19.5dBm en modo 802.11b
- Corriente de fuga de <10uA
- Bajo consumo de CPU de 32 bits integrado podría utilizarse como procesador de aplicaciones
- SDIO 1.1 / 2.0, SPI, UART
- El consumo de energía en espera de <1.0mW (DTIM3)
- STBC, 1 × 1 MIMO, 2 × 1 MIMO
- A-MPDU & A-MSDU agregación y 0,4 ms intervalo de guarda

### 2.6.3 NodeMcu ESP 32 WiFi + Bluetooth

El ESP32 es un SoC (*System on Chip*) que integra el procesador de 32 bits, Tensilica LX6, un doble núcleo que trabaja en modo normal a 160MHz (pudiendo alcanzar los 240MHz), es decir el doble de velocidad que el ESP8266. Este nuevo SoC se caracteriza por integrar, no solo la pila (*Stack*) de comunicación WiFi con una mejora en el aspecto de seguridad (encriptación criptográfica), sino que añade una pila (*Stack*) Bluetooth Low Energy (BLE, 4.1) que abre el abanico de posibilidades ya que permite implementar sobre ella diversos perfiles (*Profiles*) que

van más allá de la típica comunicación serie por bluetooth (perfil SPP), permitiendo utilizar perfiles de transmisión de audio, de pasarela (Gateway) Bluetooth-WiFi, o de implementación de dispositivos HID (interfaces hombre-máquina), ya sean periféricos existentes como ratones, teclados o gamepads, o cualquier tipo de dispositivo de control que se nos ocurra, entre otros [19].



**Figura 2.10:** Tarjeta de desarrollo ESP 32 WiFi + Bluetooth [19].

### Características

- Doble núcleo de 32 bits con velocidad de 160MHz
- Memoria ROM de 448 kBytes
- Memoria SRAM de 520kBytes
- Dispone de 48 Pines
- 18 ADC de 12 bits
- 2 DAC de 8 bits
- 10 pines sensores de contacto
- 16 PWM
- 20 entradas/salidas digitales

### 2.6.4 WEMOS D1 ESP8266 WIFI

La placa de desarrollo Wemos D1 está basada en el popular chip que revolucionó el Wifi en sistemas embebidos: Estamos hablando del ESP8266. Con este sencillo módulo podrás realizar el prototipo de cualquier sistema para el IoT (*internet of things*) en cuestión de horas. El concepto es excelente, pues no requiere de otros accesorios para tener conexión a internet como el accesorio *ethernet shield* o *wifi shield* de Arduino oficial. Con el Wemos D1 tienes todo: wifi y un potente procesador para potenciar tu aplicación. [20].

### Características de Wemos D1 Tarjeta compatible Arduino ESP8266

- Compatible con Redes 802.11 B/G/N
- Memoria Flash: 4MB
- Pines GPIO: 11
- Interfaz UART: 1
- Pines Analógicos: 1
- Interfaz SPI: 1
- Interfaz I2C por software
- Comunicación USB serial mediante el chip CH340
- Tres modos de operación: Cliente, Acces Point y Simultaneo.
- Seguridad: OPEN / WEP / WPA\_PSK / WPA2\_PSK / WPA\_WPA2\_PSK.
- Soporta comunicaciones TCP y UDP
- Soporta hasta 5 conexiones simultaneas como servidor o cliente TCP/UDP
- Compatible con infinidad de librerías en el IDE de Arduino
- API para Wifi compatible con el Shield Wifi Arduino
- Wemos D1 se programa con el IDE de Arduino



**Figura 2.11:** WEMOS D1 ESP8266 WIFI [20].

### 2.6.5 Nodemcu lua wifi basado en ESP8266

ESP8266 es un chip altamente integrado diseñado para las necesidades de un nuevo mundo conectado. Ofrece una solución de red Wi-Fi completa y autónoma, que permite alojar la aplicación o descargar todas las funciones de red Wi-Fi desde otro procesador de aplicaciones. ESP8266 cuenta con poderosas capacidades de procesamiento y almacenamiento integrados que permiten integrarlos con los sensores y otros dispositivos específicos de la aplicación a través de sus GPIO con un mínimo desarrollo inicial y una mínima carga durante el tiempo de ejecución. Su alto grado de integración en el chip permite un mínimo de circuitos externos, y toda la solución, incluyendo el módulo de *front-end*, está diseñada para ocupar un área mínima de PCB [21].

Características:

- ESP8266 ESP-13E, con todos los pines disponibles.
- Procesador compatible con Arduino y programable desde el IDE mediante el plugin ESP8266.
- -Incluye puerto micro USB para conectar fácilmente con nuestro PC.
- Pines a distancia protoboard.
- Puedes instalar el lenguaje LUA para programarlo.
- CPU RISC de 32-bit: Tensilica Xtensa LX106 a un reloj de 80 MHz.
- RAM de instrucción de 64 KB, RAM de datos de 96 KB.
- Capacidad de memoria externa flash QSPI - 512 KB a 4 MB\* (puede soportar hasta 16 MB)
- IEEE 802.11 b/g/n Wi-Fi.
- Tiene integrados: TR switch, balun, LNA, amplificador de potencia de RF y una red de adaptación de impedancias.
- Soporte de autenticación WEP y WPA/WPA2.
- 6 pines GPIO (Entradas/Salidas de propósito general).
- Interfaz I<sup>2</sup>S con DMA (comparte pines con GPIO).
- Pines dedicados a UART, más una UART únicamente para transmisión que puede habilitarse a través del pin GPIO2.
- 1 conversor ADC de 10-bit.



**Figura 2.12:** NODEMCU LUA WiFi basado en ESP8266 [21].

### 2.6.6 SparkFun ESP32 Thing

El SparkFun ESP32 Thing es una plataforma de desarrollo integral para el ESP32 de Espressif, su versión súper cargada del popular ESP8266. Al igual que el 8266, el ESP32 es un microcontrolador compatible con WiFi, pero agrega soporte para Bluetooth de baja energía (es decir, BLE, BT4.0, Bluetooth Smart) y casi 30 pines de E / S. La potencia y versatilidad del ESP32 lo convertirán en la base de IoT y proyectos conectados durante muchos años por venir [22].

#### Características

- Microprocesador Tensilica LX6 de doble núcleo
- Frecuencia de reloj de hasta 240MHz
- SRAM interno de 520kB
- Transceptor WiFi 802.11 BGN integrado
- Bluetooth de modo dual integrado (clásico y BLE)
- Rango operativo de 3.0 a 3.6V
- 2.5  $\mu$ A de corriente de sueño profundo
- 28 GPIO
- Soporte táctil capacitivo de 10 electrodos
- Cifrado acelerado por hardware (AES, SHA2, ECC, RSA-4096)
- 4 MB de memoria Flash
- Cargador de batería LiPo integrado

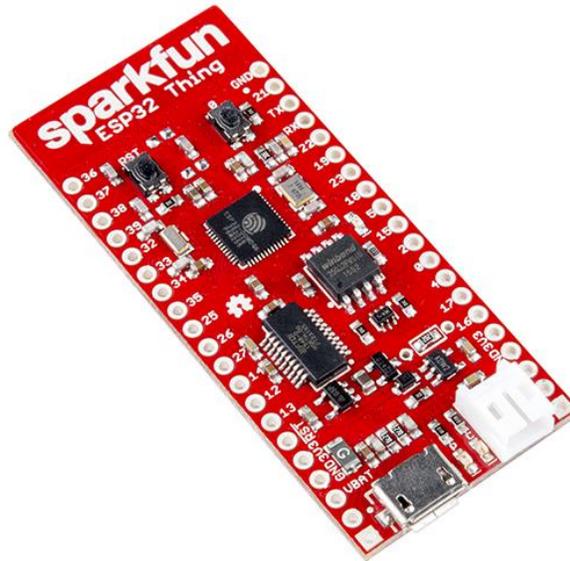


Figura 2.13: SparkFun ESP32 Thing [22].

## 2.7 Tabla comparativa

Tabla 2.1: Comparación de tarjetas de desarrollo

Nombre	Velocidad de Frecuencia	Pines digitales E/S	Pines analógicos	Comunicación inalámbrica
Arduino Nano	16MHz	14	8	ninguna
Nodemcu Esp32	160MHz/240MHz	20	18	WiFi+Bluetooth
Wemos D1 Esp8266	80MHz/160MHz	11	1	WiFi
Nodemcu Esp8266	80 MHz	17	1	WiFi
SparkFun ESP32 Thing	240 MHz	28	9	WiFi+Bluetooth

### 2.8 Análisis de la Marcha humana

La marcha humana es una etapa de la vida en que se va desarrollando desde pequeño dando unos primeros pasos. Sin embargo, según el informe mundial sobre la discapacidad, realizado por la Organización Mundial de la Salud, un poco más de “785 millones de personas (15.6%) de 15 años y más viven con una discapacidad, (...) del total estimado de personas con discapacidad, 110 millones (2.2%) tienen dificultades muy significativas de funcionamiento.” [23] esto se demuestra que cierta cantidad de la población mundial llega a presentar limitaciones motoras en el movimiento de marcha, de esta manera hace necesario el uso de dispositivos que logren ayudar al ser humano en su movilidad de caminar.

Para reducir estos efectos de discapacidad, hoy en día se utilizan dispositivos que llegan a aportar características al cuerpo como las prótesis, ya que es una extensión superficial que reemplaza una parte del cuerpo por diversas razones. Sin embargo, para el desarrollo de las prótesis implica varios estudios de la parte biomecánica del cuerpo humano. Hoy por hoy, ya se encuentra gran variedad de tipos de prótesis. Una de esas características se encuentra como el tipo material mecánicos, mioeléctricos, entre otros.

Ahora existe prótesis con mayor desarrollo una de ellas es una pierna protésica basada en EMG (Electromiograma), que mide la actividad eléctrica de los músculos cuando están en reposo y cuando están usando; para amputados que en este caso por encima de la rodilla. Las señales EMG de superficie recogidas de los músculos de la pantorrilla de una pierna sana durante la actividad muscular se interconectan con un microcontrolador que utiliza el sistema de adquisición EMG. Estas señales se procesan, analizan y utilizan para activar la articulación de la rodilla de la pierna protésica [24].

Existe otro tipo de ayuda está el exoesqueleto, es decir, un esqueleto externo que recubre, protege y soporta de una persona. En donde ya existe varios tipos de diseño, implementación y puesta en funcionamiento de un sistema de control de marcha humana, que en este caso está para un exoesqueleto de miembro inferior [25].

La Bio-Mecatrónica, es el campo que se ocupa del diseño de extremidades protésicas pasivas y activas; la cual ya están trabajando con un diseño del sistema de control para una pierna protésica utilizando un actuador de amortiguación en serie. La prótesis convencional pasiva tiene propiedades mecánicas constantes para que el movimiento de las articulaciones no sea similar al de los humanos, mientras que la prótesis de tipo activo representa de manera más realista el movimiento humano. Sin embargo, este último es más costoso que la prótesis pasiva y consume más energía [26].

### 2.9 Introducción a la inteligencia artificial (IA)

La IA se describe tanto en abstracto como en cuatro formas que hoy en día son evidentes no solo en laboratorios sino también en aplicaciones del mundo real. De esta manera se busca claridad sobre las preocupaciones del público articulando las amenazas que son inherentes a la IA. Se Ofrece que la IA debe ser reconcebida como 'inteligencia de artefactos complementarios', y que la noción robótica de 'máquinas que piensan' debe dar paso a la idea de 'intelectuales', con el foco en 'computadoras que lo hacen' [27].

La inteligencia artificial es la inteligencia que muestran las máquinas o el software. Por ello incluye razonamiento, lenguaje de procesamiento natural e incluso se utilizan varios algoritmos para poner la inteligencia en el sistema. Hoy por hoy, se investiga las motivaciones y expectativas para el desarrollo de la inteligencia artificial. Es importante resaltar que algunos artículos presentan el papel de la ética en el desarrollo de la inteligencia artificial. De lo anterior se ha comparado el nuevo alcance emergente de IA con las tecnologías antiguas en varios campos y sus ventajas para la sociedad [28].

El desarrollo inteligente ha sido un mayor impacto en la actualidad, donde ha tenido usos y aplicaciones. Básicamente, la inteligencia artificial es aquella que trata de explicar el funcionamiento mental basándose en el desarrollo de algoritmos para controlar diferentes cosas. La inteligencia artificial combina varios campos, como la robótica, los sistemas expertos y otros, los cuales tienen un mismo objetivo, que es tratar de crear máquinas que puedan pensar por sí solas, lo que origina que hasta la fecha existan varios estudios y aplicaciones, dentro de las que se encuentran las redes neuronales, el control de procesos o los algoritmos genéticos.

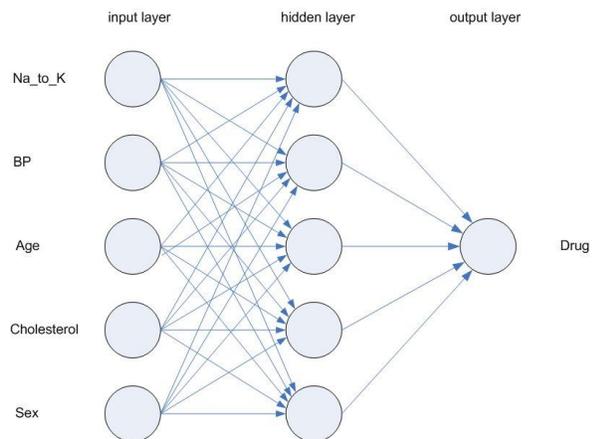
Una de sus características es que incluye varios campos de desarrollo, como la robótica, la comprensión y traducción de lenguajes, el reconocimiento y aprendizaje de palabras de máquinas o los variados sistemas computacionales expertos, que son los encargados de reproducir el comportamiento humano en una sección del conocimiento.

Tales tareas reducen costos y riesgos en la manipulación humana en áreas peligrosas, mejoran el desempeño del personal inexperto y el control de calidad en el área comercial [29].

#### 2.9.1 Redes Neuronales

Las redes neuronales es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida (ver figura 2.14), con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente. al final, se envía un resultado desde la capa de salida [30].



**Figura 2.14:** Estructura de una red neuronal artificial. [30]

La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada.

Al principio, todas las ponderaciones son aleatorias y las respuestas que resultan de la red son, posiblemente, disparatadas. La red aprende a través del entrenamiento. Continuamente se presentan a la red ejemplos para los que se conoce el resultado, y las respuestas que proporciona se comparan con los resultados conocidos. La información procedente de esta comparación se pasa hacia atrás a través de la red, cambiando las ponderaciones gradualmente. A medida que progresa el entrenamiento, la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado [30].

### 2.9.2 Características

Una de las principales características de las redes neuronales artificiales, es que se las entienden como un esquema computacional distribuido que básicamente se asimila a una estructura del sistema nervioso de los seres humanos [31]. En la tabla 2.2 se describen las tres características fundamentales que considera en una red neuronal:

**Tabla 2.2:** Características de las RNA

Topología de red	Permite determinar la capacidad representativa de como una cantidad de neuronas están distribuidos en capas y distribuidas entre sí. En donde la topología de red se basa en el problema a resolver.
Regla de aprendizaje	Una red neuronal es una integración de diversos sistemas de aprendizaje, es por ello que tienen la capacidad de aprender a través de un entrenamiento previo.
Tipo de entrenamiento	Una RNA presenta dos tipos de entrenamiento, una que durante la etapa de aprendizaje la red se entrena para que los pesos sinápticos se adecuen a la red. La segunda es una etapa de ejecución en donde la red ya pasa ser operativa cuya red toma un valor de funcionamiento real.

### 2.9.3 Ventajas y Desventajas

#### Ventajas

- Aprendizaje adaptativo en el cual aprende a realizar tareas a partir de un conjunto de datos siendo así que en el proceso de aprendizaje de estos datos sean representados como entradas y pesos.
- Auto-organización pueden crear su propia organización o representación de la información recibida, las redes neuronales autoorganizan su información que reciben durante el aprendizaje de la operación utilizando los métodos matemáticos Adeline, Madeline y Perceptrón entre otros [31].
- Tolerancia a fallos parciales sabiendo que la destrucción parcial de la red daña el funcionamiento a la misma, pero no la destruye completamente. Esto es debido a la redundancia de la información contenida, es decir, esto conlleva a que la información no se pierda ya que funciona como el cuerpo humano.

- Operación en tiempo real que pueden ser llevadas a cabo por computadoras o dispositivos de hardware especial para aprovechar así la capacidad de las RNA.

### Desventajas

- Menciona que las redes neuronales artificiales tienen complejidad para el aprendizaje de tareas grandes.
- Elevado tiempo de aprendizaje, esto dependerá de, el número de patrones a reconocer y la flexibilidad para reconocer patrones que sean bastante parecidos.
- No tiene la capacidad de interpretar los resultados que esta produce siendo necesario por tal razón la interpretación del desarrollador o de la aplicación para que le de significados a tales resultados.
- Tiempo de convergencia, es decir existen aplicaciones que requieren un enorme conjunto de ejemplos para el entrenamiento lo que se traduce en tiempo de cálculo [31].

### 2.9.4 Tipos de Redes Neuronales Artificiales

#### -Adaline y madaline

Son modelos adaptativos desarrollados por Bernard Widrows y Marcian Hoff en la Universidad de Stanford.

Adaline tiene las siglas *Adaptive Linear Element* fue desarrollado con el propósito de que pueda reconocer patrones binarios. Su funcionamiento se basa en la suma de los pesos de las entradas produciendo una salida con 0 y 1 esto lo hace dependiendo si pasa o no un umbral. Es un elemento muy importante ya que de él se derivan redes más complejas.

Varias Adaline son organizadas en capas de tal manera que se obtengan grandes arquitecturas formando una red Madaline por lo que la cual produce funciones más complicadas.

Madaline tiene las siglas *Múltiple Adaptive Linear Element* fue la primera red neuronal multicapa que incluía elementos adaptivos.

Está constituida por dos o más unidades del Adaline, esta estructura fue usada como un filtro para eliminar el eco en las líneas telefónicas [31]. Esta fue diseñada después del Adaline con el fin de realizar soluciones más complejas que la solución de problemas linealmente separables.

### **-Red backpropagation**

Fue Creada en 1986, este tipo de red se basa en el aprendizaje de la asociación de patrones existentes.

El backpropagation tiene la capacidad de generalizar lo que es un comportamiento lo hace mediante un conjunto de muestras escogidas como patrones de entrenamiento. Es como el cerebro humano, por lo que esta red necesita de una etapa de aprendizaje.

Por lo que se quiere decir en otras palabras este tipo red requiere una búsqueda de una función de comportamientos que se acople a los valores muestreados y al comportamiento del sistema con un valor mínimo de error [31].

### **- Perceptrón multicapa**

Está conformado por un conjunto de neuronas que permite la comunicación mediante él envió de señales entre sí, con un conjunto de buenas conexiones, además no puede reconocer caracteres complejos. El perceptrón multicapa tiene algunos aspectos que son importantes y se los describe a continuación:

- Tiene un conjunto de unidades de procesamiento que son las neuronas, células.
- Tiene lo que es un estado de activación que determina la salida de la neurona.
- Hay conexiones entre las neuronas, pero cada conexión se determina por un peso, en la cual ayuda a fijar el efecto de la señal de la neurona.
- Además, tiene una regla que se le llama propagación esta determina lo que es la entrada efectiva de una neurona a partir de las entradas externas.
- Tiene un entorno que le proporciona las señales de entrada, con el cual debe operar el sistema de la neurona.
- Tiene un método para trata la información lo hace mediante una regla de aprendizaje.

### **2.9.5 Formas de aprendizaje**

#### **- Aprendizaje supervisado**

El aprendizaje supervisado, es una técnica de aprendizaje de las redes neuronales artificiales que se basa en deducir una función a partir de datos o información de entrenamiento, dichos datos están compuestos por vectores u objetos dados en par, un par de la información es la entrada mientras que el otro son los resultados deseados. La finalidad de este sistema de aprendizaje es predecir de manera correspondiente de cualquier entrada valida que un supervisor ingrese [31].

### - Aprendizaje no supervisado

El aprendizaje no supervisado o auto supervisado es cuando el proceso de aprendizaje de las redes neuronales no es inspeccionado, ni controlado por ningún agente externo.

A diferencia del supervisado, este no requiere de un vector de salida, partiendo únicamente de un vector de entrada. La red distribuye patrones utilizando la densidad probabilística, extrayendo rasgos según la similitud a las redes de entrada.

Se necesita gran cantidad de vectores de entrada, de tal manera que la red dispondrá de números suficiente de patrones y se obtendrá resultados de calidad [31].

### 2.10 Aplicaciones en de desarrollo inteligente

Una de esas aplicaciones se presenta en el estudio de la cuantificación de la simetría de la marcha y su aplicación al desarrollo inteligente de prótesis de pierna. Al emplear el análisis de componentes principales (PCA), las señales cinemáticas de un lado se reconstruyen usando la del lado opuesto. Las señales reconstruidas coinciden muy bien con las adquiridas, lo que verifica la existencia de fuertes correlaciones lineales en la marcha humana sana.

Para evaluar la simetría de la marcha cuantitativamente, se seleccionan varias características clásicas espacio-temporales para la marcha calculadas por algoritmos que implican la función de autocorrelación para la construcción de criterios cuantitativos mediante la combinación con índices de simetría. Para la validación del esquema de cuantificación de simetría de la marcha, se llevan a cabo experimentos de caminata nivelada que incluyen sujetos humanos sanos y un sistema de acoplamiento protésico de pierna inteligente humano-inteligente. Los resultados indican la exactitud y efectividad del esquema propuesto y el análisis de simetría de la marcha muestra un gran potencial en el campo de la rehabilitación biomédica [32].

La marcha humana está siendo sometida hoy en día a un extenso análisis. En el trabajo “Predicción del movimiento de las extremidades inferiores utilizando características de diagramas de ángulo-ángulo e inteligencia artificial” se centra en predecir la marcha humana con el uso de diagramas de ángulo-ángulo, también llamados ciclogramas. Junto con la inteligencia artificial, los ciclogramas ofrecen una amplia área de aplicaciones médicas. Las predicciones de los movimientos de las piernas se pueden utilizar para la programación de ortesis y prótesis, y también para la rehabilitación. Hemos identificado características del ciclograma como la pendiente y el área del ciclograma para un algoritmo de aprendizaje de redes neuronales. Las redes neuronales aprendidas por las características del

ciclograma predijeron la curva del ciclograma y ofrecen amplias aplicaciones en sistemas de control de prótesis [33].

Uno de los problemas más importantes en robótica es el cálculo de la cinemática inversa (IK). Esta tarea aparentemente simple es necesaria para determinar cómo mover cada articulación para alcanzar la pose deseada del efector final en el espacio cartesiano. Sin embargo, la cinemática directa asociada puede ser una función altamente no lineal, no biyectiva y multidimensional para la cual puede ser difícil o incluso imposible encontrar soluciones de forma cerrada para su inverso, especialmente a medida que aumenta el número de Grados de libertad (DoF). Se han tomado varios enfoques utilizando aproximadores no lineales para resolver problemas IK. Dado a eso, Ya se presenta estudios para ese problema, uno de ellos esta “Un estudio sobre la resolución de la cinemática inversa de los robots en serie utilizando la red neuronal artificial y la red neuronal difusa”. presentamos un estudio sobre la resolución de la cinemática inversa de múltiples brazos robóticos utilizando Redes Neuronales Artificiales (ANN) y Sistemas Adaptativos de Inferencia *Neuro-Fuzzy* (ANFIS). Para este estudio, experimentamos con 4, 5, 6 y 7 robots seriales DoF, con combinaciones de articulaciones prismáticas y revolutivas [34].

### 2.11 Análisis Cinemático de la Marcha Humana en el Plano Sagital

El análisis cinemático describe los movimientos del cuerpo en conjunto y los movimientos relativos de las partes del cuerpo durante las diferentes fases de la marcha. Donde el análisis está dividido en tres intervalos, en los cuales se describe cómo actúan el tobillo, rodilla y cadera en el plano sagital para cada una de las fases [35].

#### 2.11.1 Intervalo I

Movimiento de las articulaciones en el plano sagital entre el contacto del talón con el suelo y el punto de apoyo medio.

##### A. El Tobillo

**Tabla 2.3:** Movimiento del Tobillo intervalo I

Momento de contacto del talón con el suelo.	La articulación del tobillo está en posición neutra (0°). Justo entre la dorsiflexión y la flexión plantar.
Simultáneamente con el contacto del talón.	La articulación del tobillo empieza a moverse en dirección de la flexión plantar.
Momento en que la planta del pie hace	La articulación del tobillo se mueve 15° de la posición neutra a la flexión plantar.

contacto con el suelo.	
En la fase media.	La articulación del tobillo pasa rápidamente a aproximadamente 5° de dorsiflexión.

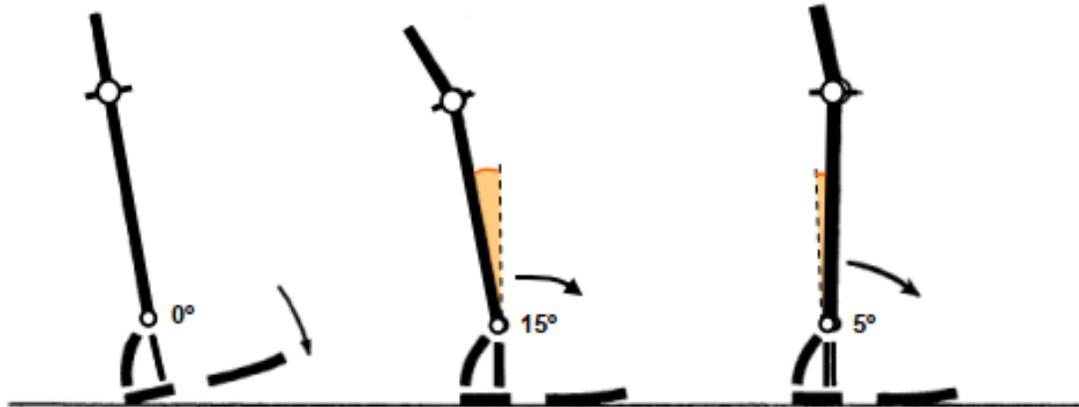


Figura 2.15: Estudio cinemático del tobillo en el Intervalo I de la marcha [35].

B. La Rodilla

Tabla 2.4: Movimiento de la Rodilla intervalo I

Inmediatamente antes del contacto del talón con el suelo.	La articulación de la rodilla se encuentra en completa extensión.
Simultáneamente con el contacto del talón con el suelo.	La articulación de la rodilla comienza a flexionarse y continúa hasta que la planta del pie esté plana en el suelo.
Inmediatamente después de haber alcanzado la posición plana del pie.	La rodilla tiene aproximadamente un ángulo de 20° de flexión y comienza a extenderse.
En el apoyo medio.	La rodilla tiene aproximadamente un ángulo de 10° de flexión y continúa extendiéndose.

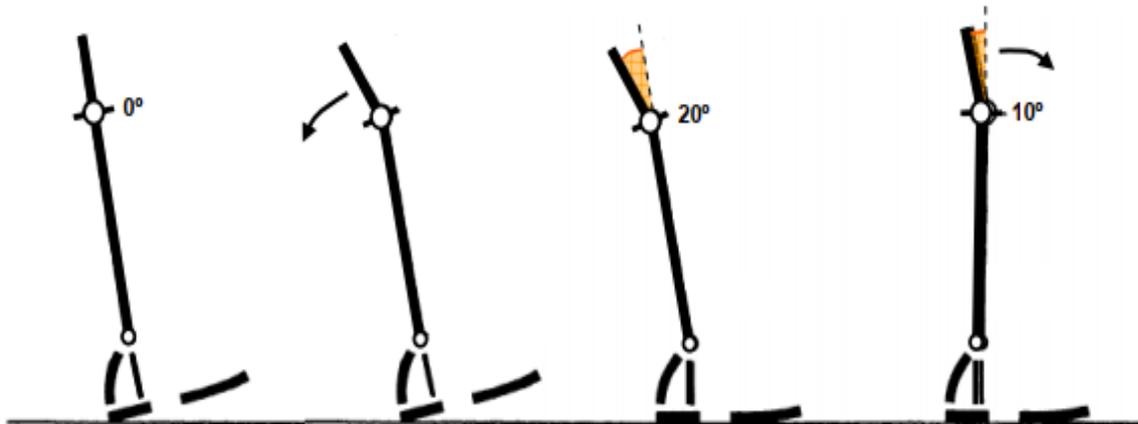
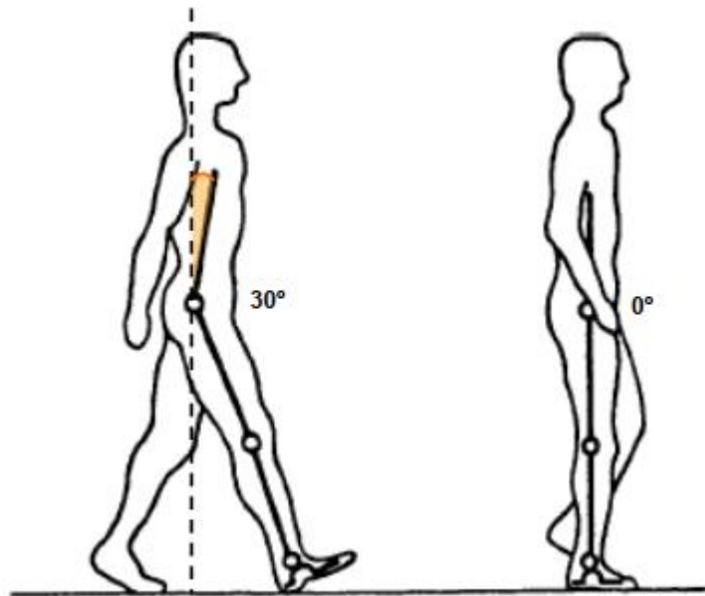


Figura 2.16: Estudio cinemático de la rodilla en el Intervalo I de la marcha [35].

### C. La Cadera

Tabla 2.5: Movimiento de Cadera intervalo I

Simultáneamente con el contacto del talón con el suelo.	La cadera está aproximadamente a 30° de flexión.
Inmediatamente después del contacto del talón con el suelo.	La articulación de la cadera comienza a extenderse.
En la posición del pie plano en el suelo.	El ángulo de flexión disminuye alrededor de 20°.
Entre el pie plano y el apoyo medio.	La articulación de la cadera se mueve a su posición neutral (0°).



**Figura 2.17:** Estudio cinemático de la cadera en el Intervalo I de la marcha [35].

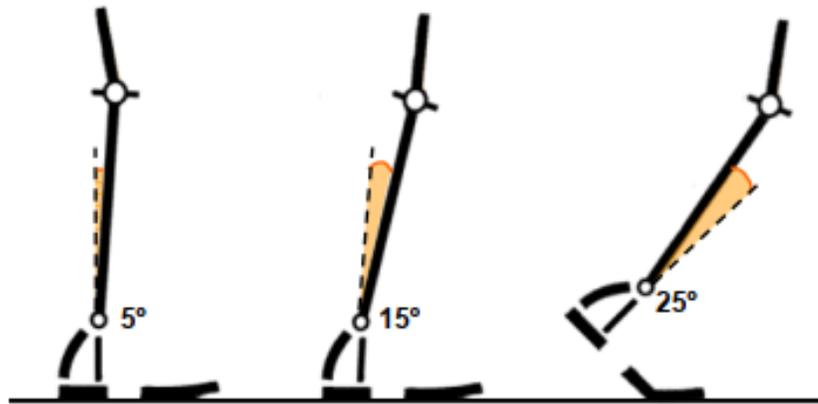
### 2.11.2 Intervalo II

Movimiento de las articulaciones en el plano sagital entre el apoyo medio y despegue del pie del suelo.

#### A. El Tobillo

**Tabla 2.6:** Movimiento del Tobillo intervalo II

En el apoyo medio.	La articulación del tobillo pasa rápidamente a aproximadamente 5° de dorsiflexión.
En el momento que el talón se despegaba del suelo.	La articulación del tobillo está aproximadamente a 15° de dorsiflexión.
En el intervalo de elevación del talón y el despegue del pie.	El tobillo se mueve rápidamente 35°, con lo que al despegar el pie del suelo la articulación está aproximadamente en 20° de flexión plantar.



**Figura 2.18:** Estudio cinemático del tobillo en el Intervalo II de la marcha [35].

B. La Rodilla

**Tabla 2.7:** Movimiento de la Rodilla intervalo II

En el apoyo medio.	La rodilla tiene aproximadamente un ángulo de 10° de flexión y continúa extendiéndose.
Inmediatamente antes de que el talón pierda contacto con el suelo.	La rodilla está a 4° de la extensión completa.
Entre el despegue del talón y el de los dedos.	La articulación de la rodilla se mueve de una extensión casi completa a 40° de flexión.

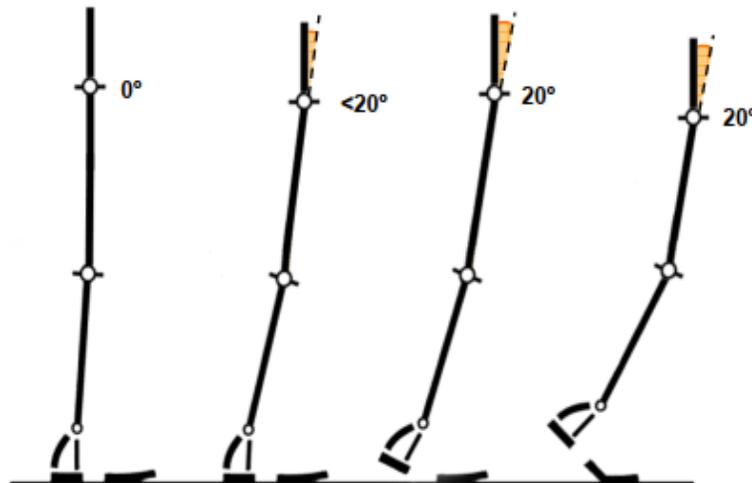


**Figura 2.19:** Estudio cinemático de la rodilla en el Intervalo II de la marcha [35].

C. La Cadera

**Tabla 2.8:** Movimiento de la Cadera intervalo II

En el apoyo medio.	La articulación de la cadera se encuentra en posición neutra ( $0^\circ$ ) y comienza a moverse hacia la extensión.
Inmediatamente después del despegue del talón.	La cadera alcanza un máximo de hiperextensión de $20^\circ$ .
En el momento de despegue de los dedos del suelo.	La cadera está cerca de una posición neutral y se mueve en dirección de la flexión.



**Figura 2.20:** Estudio cinemático de la cadera en el Intervalo II de la marcha [35].

**2.11.3 Intervalo III**

Describe el movimiento de las articulaciones en el plano sagital en la etapa de balanceo.

A. El Tobillo

**Tabla 2.9:** Movimiento del Tobillo intervalo III

Durante la etapa de balanceo.	El pie se mueve de su posición inicial de flexión plantar al desprenderse del suelo a una posición esencialmente neutra ( $0^\circ$ ) que se mantiene durante toda la etapa de balanceo.
-------------------------------	--

B. La Rodilla

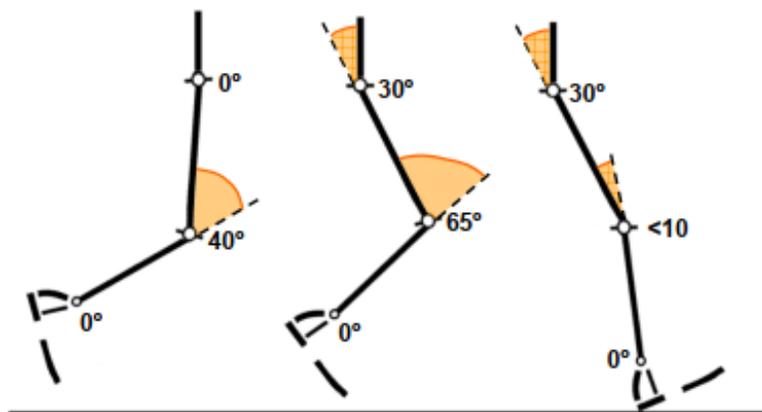
**Tabla 2.10:** Movimiento de la Rodilla intervalo III

Entre el despegue del pie y la parte media de la etapa de balanceo.	La rodilla se flexiona de una posición inicial de aproximadamente $40^\circ$ a un ángulo de máxima flexión de aproximadamente $65^\circ$ .
Entre la parte media de la etapa de balanceo y el contacto del talón.	La rodilla se extiende casi completamente hasta el último instante de la etapa de balanceo.

C. La Cadera

**Tabla 2.11:** Movimiento de la Cadera intervalo III

Durante la etapa de balanceo.	Partiendo de una posición neutral, la articulación de la cadera se flexiona aproximadamente $30^\circ$ y se mantiene en esa posición.
-------------------------------	---



**Figura 2.21:** Estudio cinemático de las articulaciones en el Intervalo III de la marcha [35].

### 2.12 Estudios cinemáticos de miembros inferiores

En la análisis y control cinemático, se da a conocer cuáles son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos fijados por el usuario (punto de destino, trayectoria cartesiana del efector final del robot, tiempo invertido por el usuario, etc.). Estas trayectorias se seleccionarán atendiendo a las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad o precisión de la misma [36]. Por ello, hay muchos artículos donde trabajan en el tema cinemático y unos ellos está en “el análisis cinemático y dinámico de las prótesis transfemorales, implicaciones clínicas”, en que la asimetría cinética en la marcha son unas de las principales causas de las patologías degenerativas de la cadera. La incorporación de variables cinemáticas y cinéticas permite establecer con los ensayos clínicos de radiografía y densitometría, importantes conclusiones relacionadas con el desequilibrio dinámico principal causa de las patologías degenerativas de la cadera observadas en este paciente, con lo cual son un importante instrumento para la evaluación de la factibilidad del diseño del artefacto protésico [37].

Otro estudio está, en “el uso de señales cinemáticas y mioeléctricas de miembros inferiores bilaterales para predecir las actividades locomotoras: un estudio piloto”. Hoy en día los exoesqueletos activos de las extremidades inferiores pueden proporcionar asistencia a las extremidades inferiores y pueden mejorar drásticamente la capacidad de caminar de millones de personas con problemas de la marcha. Sin embargo, la mayoría de los sistemas de control disponibles actualmente para estos dispositivos no pueden predecir los movimientos previstos del usuario y aún no permiten caminar con transiciones sin interrupciones. Los desarrollos recientes en el reconocimiento de la intención de las prótesis activas de miembros inferiores han demostrado que el uso de señales cinemáticas y cinéticas del dispositivo y las señales mioeléctricas del usuario pueden proporcionar una interfaz de control intuitiva para una transición fluida entre diferentes actividades locomotoras [38].

Por último, en otro artículo se centra en “el análisis cinemático y generación de trayectoria de seis dispositivos ortopédicos para miembros inferiores”. Donde el análisis cinemático de seis articulaciones del exoesqueleto de la extremidad inferior se realiza aquí para el estudio de control de movimiento del exoesqueleto. Los resultados simulados muestran que los parámetros espaciales están en límites aceptables de acuerdo con la variación de los ángulos articulares. También se puede usar para decidir la posición de golpe del talón junto con el sensor de presión del pie. Al considerar la discapacidad de la marcha, este trabajo fortalece

nuestro discernimiento de la interacción compleja de las articulaciones de las extremidades inferiores y la importancia de su mecánica [39].

### 3. Metodología

Para el desarrollo del traje, primero hay que tener en cuenta que la tela debe ser elástica para que sea acorde y ajustable al cuerpo de la persona. Donde se puede encontrar la licra fácilmente en almacenes deportivos. Por lo tanto, se precede a las directrices del diseño y algunas restricciones durante todo el proceso.

#### 3.1 Diseño Electrónico

La electrónica seleccionada para el prototipo, se debe determinar en la facilidad de poder desarrollar y en la parte de bajo costo con el objetivo para las tomas de datos. Teniendo en cuenta para la persona al momento de desplazarse. Dado a eso se dará a conocer la elección de los componentes electrónicos.

##### 3.1.2 Elección de componentes electrónicos

A continuación, se seleccionaron los componentes para al desarrollo del prototipo, dando a conocer ciertas características importantes. Y teniendo en cuenta para futuros trabajos en donde posiblemente se haga avances.

##### -Sensor

La Empresa Spectra Symbol, ha hecho sensores resistivos de alta calidad donde ha hecho con los mejores materiales, este producto es esencial en cualquier instalación de fabricación y robótica. Los sensores flexibles Spectra Symbol están disponibles en las siguientes longitudes activas: 2,18 pulgadas y 3,75 pulgadas. A continuación, se mostrará algunas características.

##### Especificaciones mecánicas

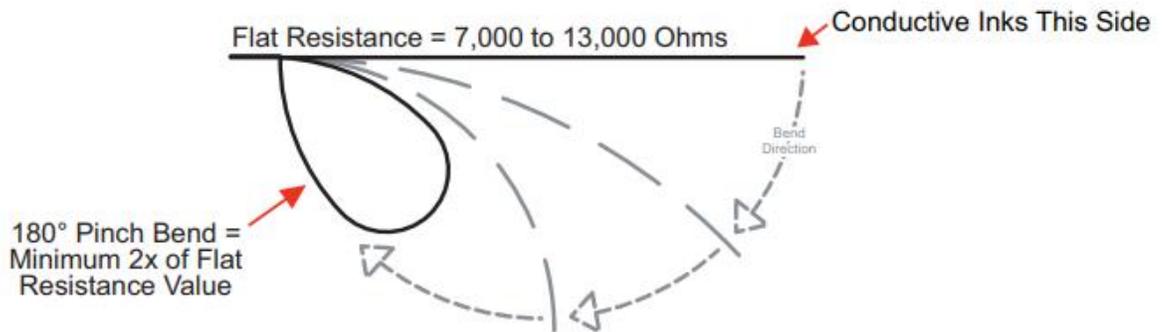
- Ciclo de vida:> 1 millón
- Altura: 0.43 mm (0.017 ")
- Rango de temperatura: -35 ° C a + 80 ° C

##### Especificaciones eléctricas

- Resistencia plana: 10K ohmios  $\pm$  30%
- Resistencia a la flexión: mínimo 2 veces mayor que la resistencia plana en una curva de pellizco de 180 ° (consulte "Cómo funciona" a continuación)
- Capacidad de potencia: 0.5 vatios continua; 1 vatio pico.

### Cómo funciona

En la figura 3.1, se puede apreciar el esquema del sensor Flex al momento de ejercer una movilidad. En donde el valor neutral del sensor es de 13k ohms sin doblar y el valor mínimo es de 7k ohms al momento de doblar al máximo tope, permitiendo así un movimiento de trabajo de 180 °.



**Figura 3.1:** Esquema de trabajo del Sensor Flex [15].

Dado a conocer lo anterior, este sensor permite tomar datos en movimiento en donde estará puesto en el traje. El cual se utilizó 6 sensores, cada uno de ellas va estar colocados en las articulaciones en la parte inferior del cuerpo que sería cadera, rodilla y tobillo, que se apreciara más adelante se apreciara más adelante.



**Figura 3.2:** adaptación de Sensor Flex hacia traje [Autor].

En la figura 3.2, se puede apreciar como resultado una manera de factible en utilizar los sensores hacia el traje en que se puede despegar y pegar los sensores. Además, se coloca dos cables de diferente color para identificar la polaridad con el fin de facilitar al conectar hacia la tarjeta. Por último, esto ayudara en de poder tener agilidad.

#### -Tarjeta de Control

Para realizar las conexiones de los sensores que en este caso son analógicos, es recurrente el uso de una tarjeta que llegue cumplir las características necesarias. Que tenga pines analógicos y una comunicación inalámbrica rápida. Consiguente, se llegó a utilizar la Tarjeta NodeMcu Esp 32, en la cual cumple los parámetros necesarios en donde se dará a conocer a continuación:

Los pines nombrados con la letra ADC son los pines analógicos, de las cuales se escogieron los pines ADC3, ADC6, ADC7, ADC4, ADC5, ADC18, para programar y para hacer pruebas con los sensores y calibrar los valores con un mapeo. Luego para trabajar con la parte de protocolo UDP con la tarjeta, se toma los pines ADC0, ADC3, ADC6, ADC7, ADC4, ADC5, (ver figura 3.3) ya que en el programa Arduino solo permite trabajar con eso pines. Visto que tenga alguna interferencia en la tarjeta [40].

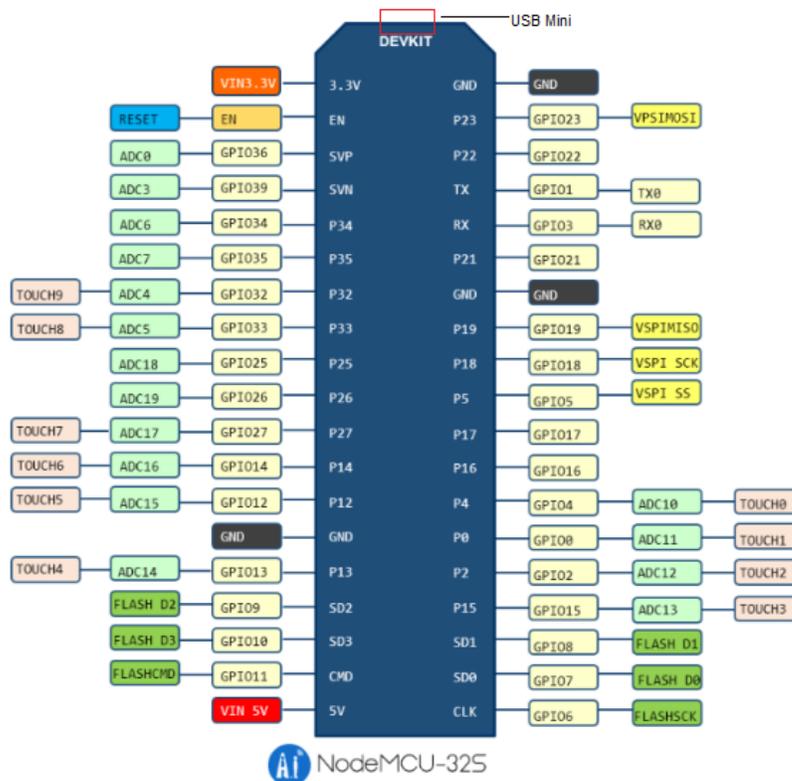


Figura 3.3: Diagrama de pines ESP-WROOM-32s [40].

### -Alimentación del prototipo

El módulo tiene una manera factible en alimentar los sensores que es en el puerto USB Mini (ver figura 3.3) que es la fuente de alimentación. Donde se puede trabajar con una batería de Polímero de Litio (LiPo). En efecto, será viable trabajar en campo abierto.

### Batería LiPo

La tarjeta se requiere una alimentación no superior de 5 V que una característica del fabricante, y de esa forma trabaja con un Voltaje de fuente de alimentación 3.0V ~ 3.6V, 3.3V típico, y Corriente > 500mA. Dado a eso, se hace elección de una batería LiPo de 7.4 V a 1000 mAh de dos celdas, ver en la figura 3.4.



**Figura 3.4:** Batería LiPo.

Las características de esta batería, satisface las necesidades en toda la parte electrónica y logra tener un tiempo de autonomía aceptable, más adelante se muestra sus cálculos de consumo y rendimiento. No obstante, el módulo debe estar alimentado no mayor a 5 V, de esta forma es obligatorio utilizar un convertidor de cargador de LiPo a USB (ver figura 3.5) para regular la entrada y salida de la batería ya puede arrojar Voltaje y corriente de salida: 5V-2A. para la tarjeta.



**Figura 3.5:** Regulador De Voltaje Para LiPo.

Posteriormente, los sensores Flex se alimentará aun voltaje de 3.3 V y con un consumo de corriente de 0.0003 A. De este modo, el consumo de total de los 6 sensores es:

$$\text{Consumo de corriente a máximo par} = N^{\circ} \text{ sensores} \times \text{Consumo de corriente} \quad (1.1)$$

Tomando de la ecuación 1.1, se analiza la máxima corriente requerida por lo sensores empleado:

$$\text{Consumo de corriente a máximo par} = 6 \times 0.0003$$

$$\text{Consumo de corriente a máximo par} = 1.8 \text{mA}$$

Teniendo en cuenta lo anterior, hay que conocer el tiempo de funcionamiento del prototipo al ser alimentado por la batería LiPo, a las condiciones par y están dadas por:

$$TD(h) = CB \text{ (mAh)} / CI \text{ (mA)} \quad (1.2)$$

Donde,

$TD$ = Tiempo de descarga

$CB$ = Capacidad de batería

$CI$ = consumo de corriente a máximo par.

Dando como resultado de los 6 sensores, el cálculo es:

$$TD(h) = 1000 \text{mAh} / 1.8 \text{mA} = 555 \text{ min} = 9,25 \text{ horas.}$$

#### - ADC Conversor Analógico Digital ADS1115 16 Bit

EL módulo ADS1115 es ideal para microcontroladores sin convertidor analógico a digital o cuando se desea un ADC de mayor precisión, el ADS1115 proporciona precisión de 16 bits a 860 muestras/s con protocolo de comunicación I2C. El chip se puede configurar como 4 canales analógicos de entrada independientes o dos canales diferenciales [41].

Tabla 3.1: Características del convertidor

CARACTERISTICAS	VALOR
Voltaje de operación	De 2V a 5.5V
Consumo de Corriente	150 $\mu$ A (modo continuo)
Velocidad de muestreo	de 8 Hz a 860 Hz
Resolución	16-bit
Canales	4 canales de entrada o 2 diferenciales
Interfaz de comunicación	I2C (4 direcciones)

La resolución de un convertidor analógico digital (ADC) se expresa en número de bits. Establece el número de niveles en los que se puede dividir un rango de entrada analógico.

Para calcular la resolución de un convertidor analógico digital (ADC) de n-bit, se divide 1 entre 2 elevado a n.

$$resolucion = \frac{1}{2^n}$$

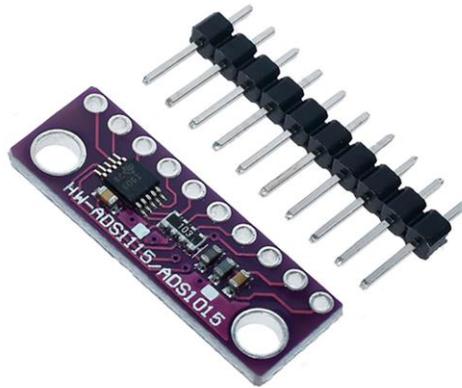
Para el caso que del convertidor analógico digital ADS1115 de 16 Bit tiene una resolución de

$$resolucion = \frac{1}{2^{16}} = \frac{1}{65536} = 0,0000152$$

Ahora si el rango de entrada de este convertidor es de 3.3 V la resolución es

$$resolucion = \frac{3.3V}{2^{16}} = \frac{3.3}{65536} = 0,00005035V = 0,05035mV$$

Lo que viene a decir que se puede medir variaciones de 0,05035mV en la señal analógica de entrada.



**Figura 3.6:** ADC Conversor Analógico Digital ADS1115 16 Bit.

Este convertidor de alta resolución fue necesario utilizar ya que la tarjeta Esp32 no puede diferenciar entre señales de 1mv y 2mv, lo que significa que ofrece una resolución muy baja [42]. Dado a eso se implementó las conexiones de este convertidor hacia la tarjeta Esp32 (ver figura 3.7). Para las conexiones se debe tener en cuenta el tipo el de dirección I2C, ya que hay 4 posibles direcciones que se pueden utilizar con el pin ADDR. Dependiendo de a donde se conecte utilizará una dirección u otra.

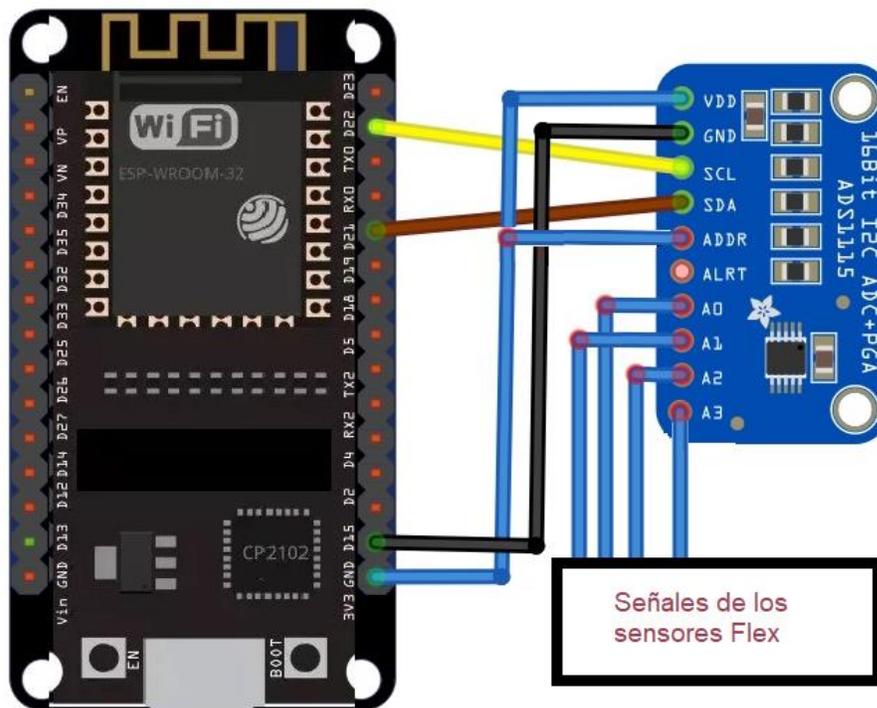
seleccionar una dirección entre las 4 posibles se utiliza el pin ADDR. Dependiendo de a donde se conecte utilizará una dirección u otra. Las posibles direcciones I2C son 0x48, 0x49, 0x4A y 0x4B [43].

En la siguiente tabla se muestra las posibles conexiones con sus direcciones correspondientes. En el caso del proyecto se tomó la dirección 0x49

**Tabla 3.2:** Características de los pines del convertidor ADS1115 16 Bit.

Conexión ADDR	Dirección
GND	0x48
5V	0x49
SDA	0x4A
SCL	0x4B

Se logro conectar 4 sensores al convertidor, ya que estos sensores son puestos en posición de la cadera y tobillos en el traje, en donde se presento dificultades en reconocer cierto nivel de ángulo.



**Figura 3.7:** Esquema de conexiones de la tarjeta Esp32 con el convertidor ADS1115 16 Bit.

### 3.1.2 Caracterización del Sensor Flex

Para la toma de la señal del sensor Flex, se utilizó una resistencia (R1) adicional de 10k ohm para reducir el error debido a la impedancia de la fuente del sensor flexible aplicando como divisor de voltaje. De esta forma, se obtendrá la salida de la señal del sensor que luego será tomada como pin de entrada a la tarjeta de desarrollo (ver figura 3.8).

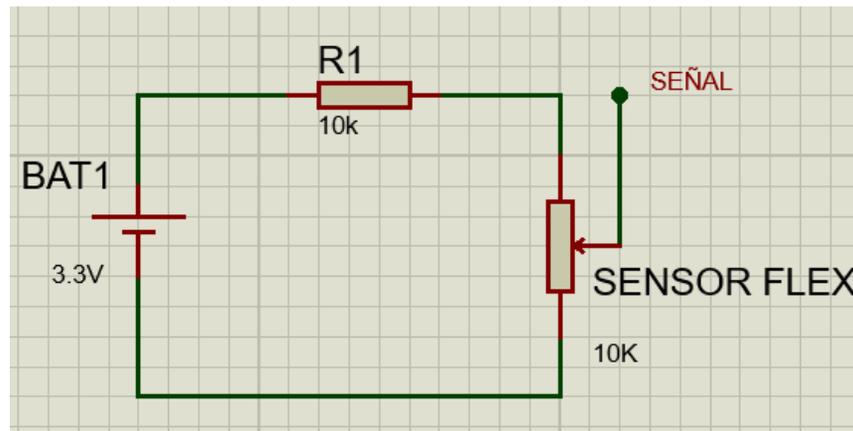


Figura 3.8: Esquema para obtener la señal del sensor [Autor].

$$V_{señal} = \frac{R_{Sensor}}{R1 + R_{Sensor}} * 3.3v$$

Por otra parte, para la caracterización del Sensor, se hizo midiendo el valor del voltaje con la ayuda de un multímetro y en un código se hacía la lectura del valor de ADC y el voltaje en el monitor serial. Luego va tomando los datos del sensor al momento de cierto ángulo (**código en ANEXO I**). Para la toma de datos se hizo la variación del sensor, a ángulos en intervalos de 10 ° hasta 90 °, permitiendo así almacenar cierta cantidad de datos en una tabla a cada ángulo. Por último, En una tabla general, es representado el valor de la media, el valor eficaz, el valor máximo y el valor de voltaje que representa cada ángulo, como se puede apreciar en la **tabla 3.3**, que es un ejemplo de cómo se obtuvo los valores y de esta forma se hizo para los demás sensores. Todo esto se realizó con la ayuda del programa Microsoft Excel.

**Tabla 3.3:** valores recolectados de uno de los 6 sensores.

Sensor 1 (10k ohm)				
Ángulos (grados)	Media Lec. Analógico	Voltaje	Vef	Vmax
0	9270,35	1,74	9269,7	9361,6
10	9456,1	1,78	9453,5	9555,8
20	9660,6	1,82	9658,1	9751,4
30	9846,4	1,85	9844,7	9891,65
40	9938,75	1,87	9938,6	10018,8
50	10100,85	1,9	10099	10146,5
60	10196,45	1,92	10194	10268
70	10340,9	1,94	10342	10391,5
80	10442,95	1,96	10441	10495,5
90	10551,2	1,98	10550	10550

Para poder representar en una gráfica de dispersión del comportamiento del sensor, se tuvo primero que seleccionar los datos correspondientes que van a efectuarse en el plano X & Y, donde en el eje X van a ser los valores de resistencia y en el eje Y que van a hacer los intervalos de los ángulos. De esta manera se logra sacar las gráficas correspondientes con los métodos estadísticos, como se pueden observar en las figuras 3.9, 3.10 y 3.11. Estas gráficas logran comportarse de forma lineal ascendente y su margen de error es de 0.3 %.

Por último, se utilizó la herramienta de línea de tendencia para obtener la ecuación representativa de cada gráfica; estas ecuaciones pasan a ser utilizadas para la programación de Arduino, en donde se determina cuál de esas ecuaciones llega a enviar una mejor respuesta al momento de doblar el sensor, y esto se logra visualizar en el monitor serial de Arduino. Para este caso la mejor ecuación representativa fue con los datos de la media, en donde se puede observar la gráfica en la figura 3.11.

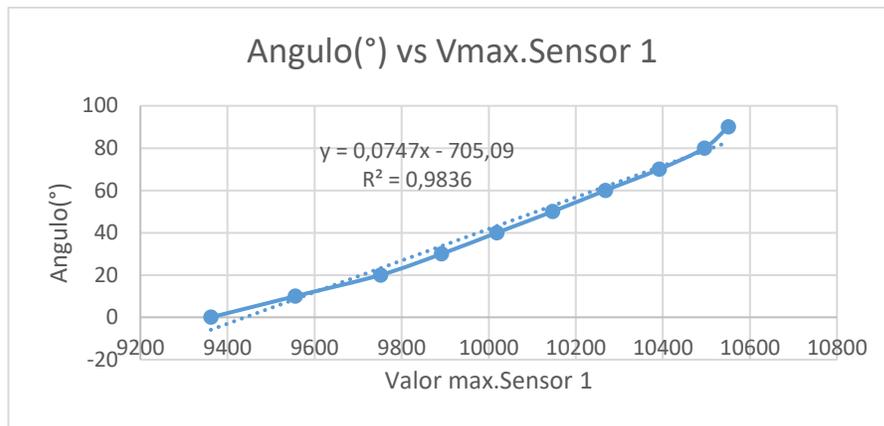


Figura 3.9: Grafica de resposta do valor máximo [Autor].

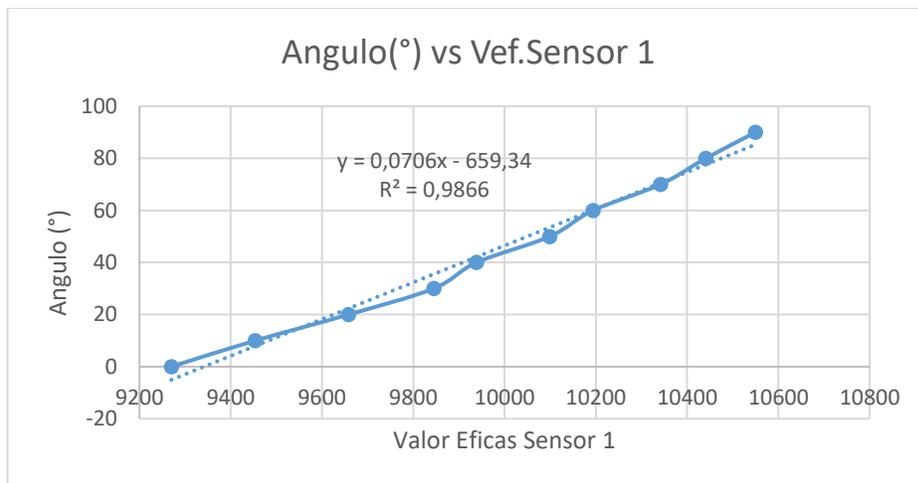


Figura 3.10: Grafica de resposta do valor eficaz [Autor].

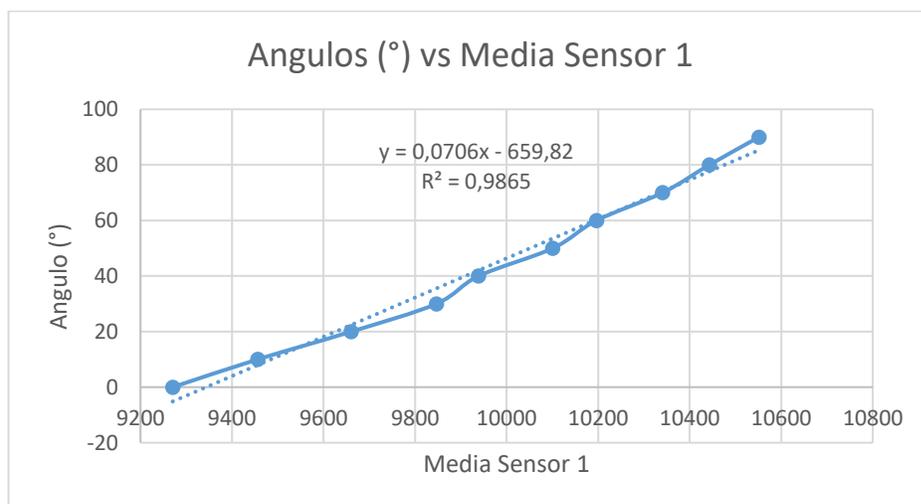


Figura 3.11: Grafica de resposta do valor medio [Autor]

### 3.1.3 Diseño de la placa PCB

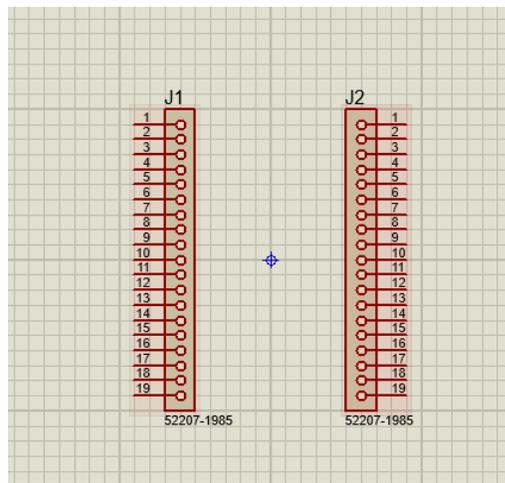
Para hacer el diseño del PCB (Printed Circuit Board) se empezó utilizando la herramienta PROTEUS, que es un software que permite la simulación de circuitos electrónicos, diseño del esquema electrónico, construcción de placas de circuitos impresos, entre otros.

Por lo tanto, se procede una explicación breve de varios pasos de cómo se desarrolló el diseño de la placa:

#### -Diseño del circuito electrónico

Para comenzar, el programa tiene una herramienta que se llama ISIS que sirve para la elaboración avanzada de esquemas eléctricos. Esta incorpora más de 6.000 de modelos electrónicos digitales y analógicos; pero esta vez no cuenta con el esquema del circuito de la tarjeta ESP32. En este sentido, se procede a crear desde cero el esquema de la tarjeta, ya que este programa nos facilita en hacerlo.

Proteus, tiene una opción que se llama ISIS, que es una herramienta para la elaboración avanzada de esquema eléctricos. En la figura 3.12, se observa la manera que se empieza hacer el esquema para la tarjeta en donde hay que tener en cuenta la cantidad de pines que tiene ella.



**Figura 3.12:** Esquema de identificación de pines para la tarjeta [Autor].

De este modo, se coloca el nombre del esquema y los pines con sus respectivos números. En donde se podrá identificar las conexiones necesarias para conectar a lo que se está llevando, como se puede ver en la figura 3.13.

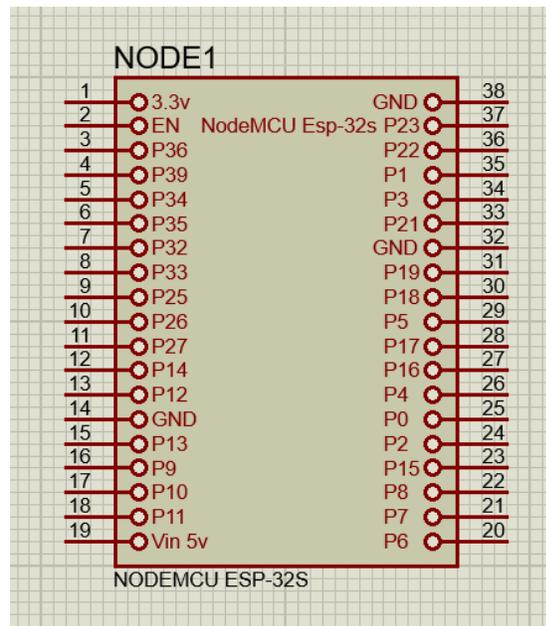


Figura 3.13: Esquema electrónico de NodeMcu Esp 32 [Autor].

Por último, se procede realizar las conexiones de entrada en cada sensor en los pines del esquema de la tarjeta. También se debe colocar una resistencia de 10k ohms en cada pin de entrada analógico como una división de voltaje; en donde pueda permitir una diferencia de voltaje tomando como una señal de datos para la tarjeta. Luego se hace las conexiones de puntos a tierra y la alimentación de voltaje, como se puede ver en la figura 3.14.

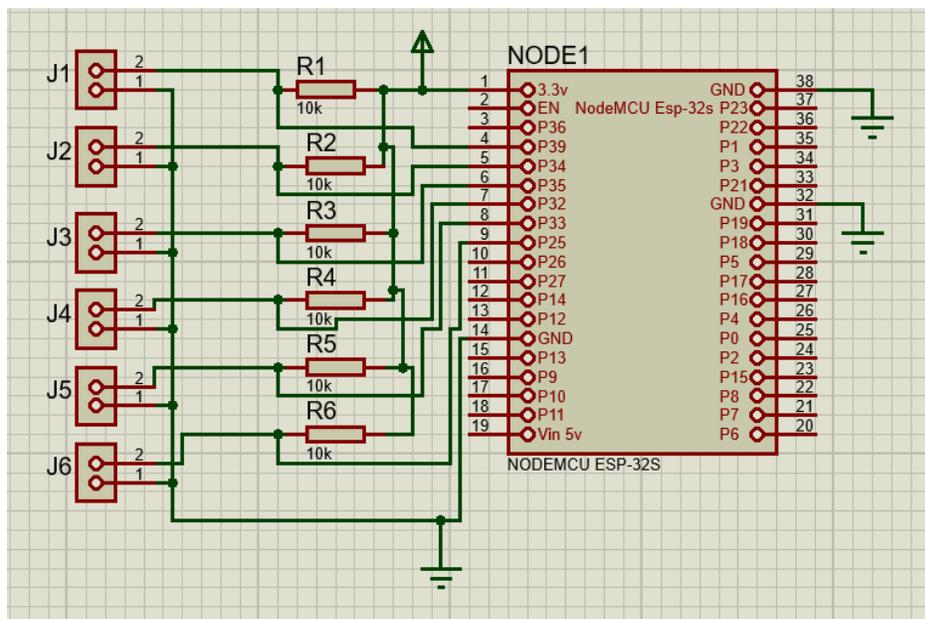


Figura 3.14: Esquema eléctrico en ISIS [Autor].

#### -Diseño del circuito impreso

La Herramienta ARES en Proteus, permite hacer en rotulado, ubicación y edición de componentes, entre otras. Teniendo ya el esquema del circuito, se procede a posicionar los componentes y hacer las pistas; que en este caso no genere tanto espacio y para que sea más compacto en llevar a todas partes.

Cabe señalar, que una vez realizado el ruteado habrá ocasiones en donde se tenga que cambiar algunos detalles como: cambiar el grosor de la línea de pista, en que capa se encuentra. A igual tenga que modificar el tipo de pad en el que se coloca los componentes y de esta forma no tener inconvenientes al momento de hacer los orificios para colocar los componentes y poder soldar.

Finalmente, al momento de tener todo listo y verificado que no haya ningún problema en la placa. Se procede la exportación de la pista a PDF, con el fin poder imprimir y aplicar el método de planchado para obtener la placa. Como se ve en a figura 3.15.

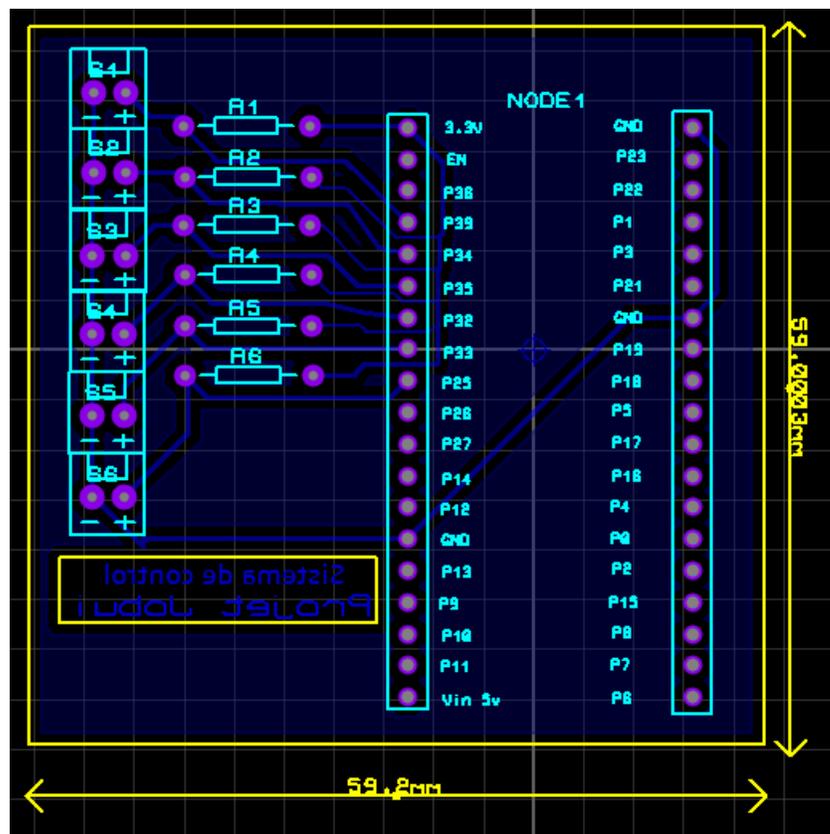
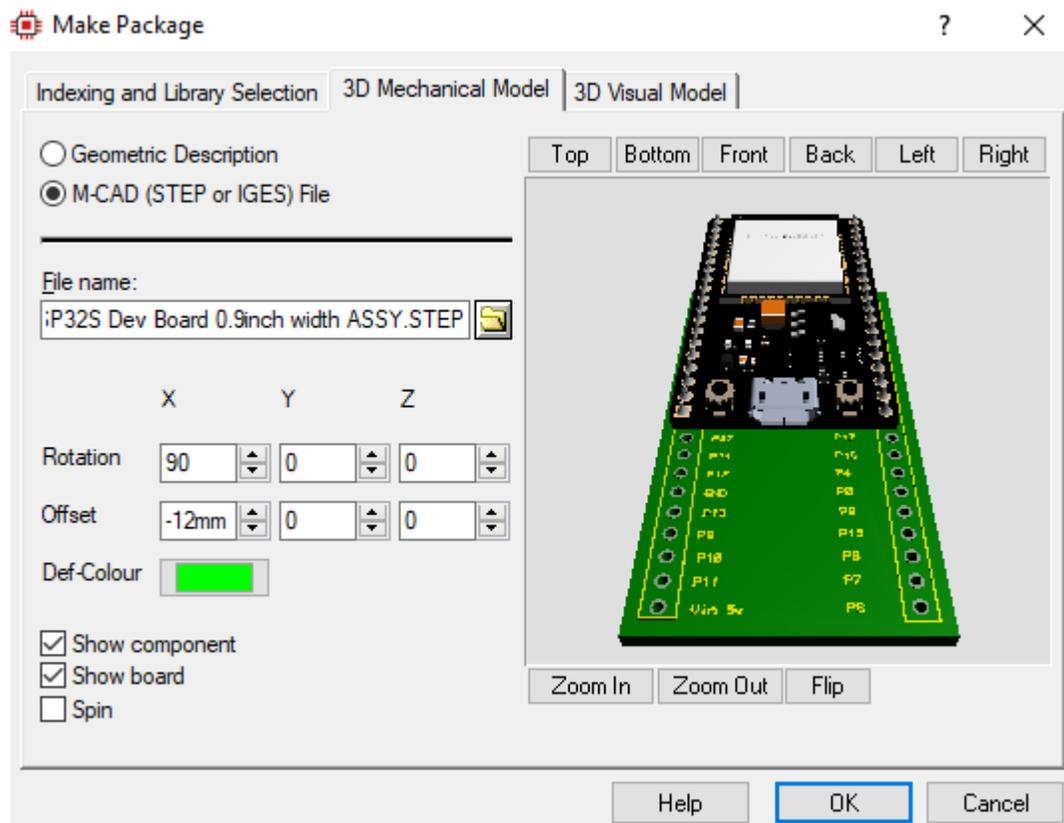


Figura 3.15: Esquema del circuito en ARES [Autor].

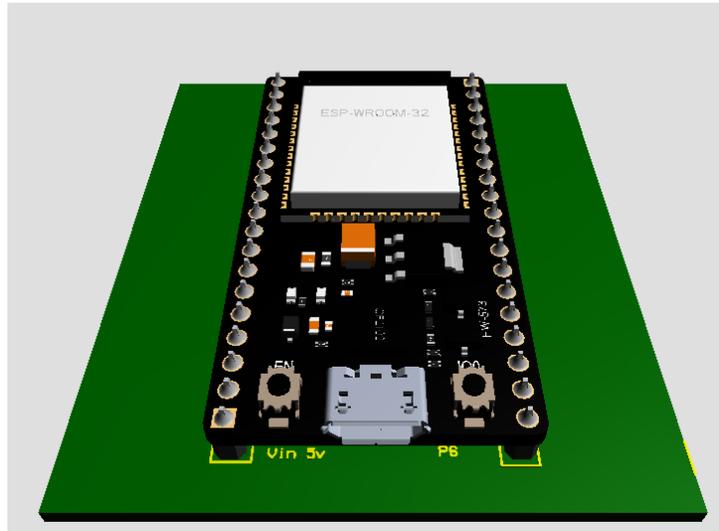
### -Construcción del prototipo físico

Para la visualización en 3D del PCB, ARES tiene una opción que se llama “3D Visualizer”; la cual permite tener una pista simulada de como quedaría armada la placa. Por lo tanto, Proteus tiene muchos esquemas de electrónicos en 3D, pero en este caso con la tarjeta no lo tiene. Dado a lo anterior, se procede a descargar el CAD de la tarjeta para luego hacer las configuraciones en la herramienta “Make Package “. Con el objetivo de poder subir el archivo CAD y hacer coordenadas de posición, con el fin que el diseño de la tarjeta y el esquema del circuito que esta nombrado los pines logren estar alineados, como se puede apreciar en la figura 3.16.



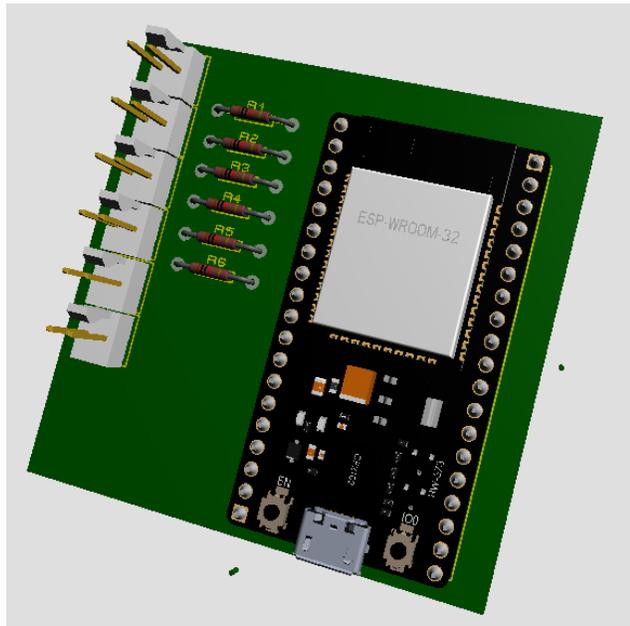
**Figura 3.16:** Alineación del Esquema y el CAD de la tarjeta en 3D.

En la figura 3.17, se aprecia la alineación completada con la base del esquema del circuito con el CAD de la tarjeta NodeMcu Esp 32, en donde se podrá visualizar una parte de cómo va quedar el circuito real.



**Figura 3.17:** Alineación completada con el CAD de la tarjeta [Autor].

En la figura 3.18, se observa la construcción final en 3D del diseño del PCB como va quedar la estructura real. Con el objetivo de poder visualizar una mejor manera de cómo va ser estructura la parte electrónica del prototipo.



**Figura 3.18:** Visualización 3D de PCB terminado [Autor].

Finalmente, en la figura 3.19, se aprecia el resultado final de la placa electrónica en donde está las resistencias, las borneras en donde va ir conectado los sensores, y por último el ensamble de la tarjeta NodeMcu Esp32.

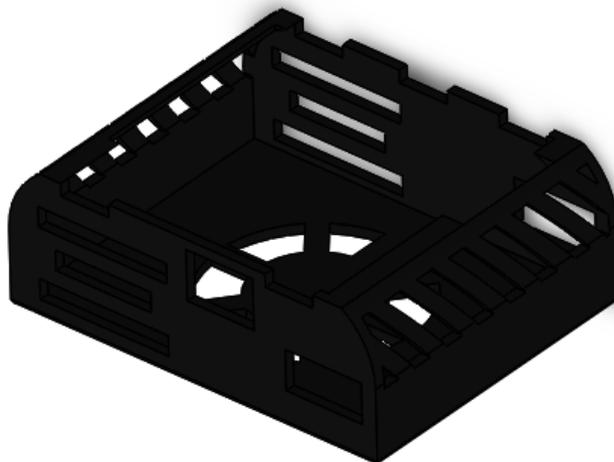


**Figura 3.19:** Resultado final de la Placa electrónica [Autor].

### 3.2 Diseño e impresión del chasis

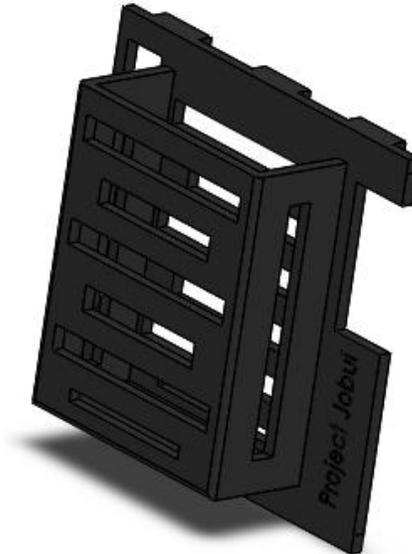
Para el diseño del chasis donde va ir la parte de electrónica, se hizo en SolidWorks 2017 que es un software de diseño 3D para modelar piezas y ensambles en 3D y planos 2D. De igual forma permite tener la extensión de guardar en varios archivos y permite en uno de ellos ser compatible con software de impresora 3D.

Para comodidad del traje, se debe tener en cuenta las dimensiones donde va ir la parte electrónica como, la tarjeta de desarrollo que permitirá la toma de datos en cada articulación. A continuación, se mostrará el modelo CAD diseñado para incorporar la parte electrónica del prototipo.



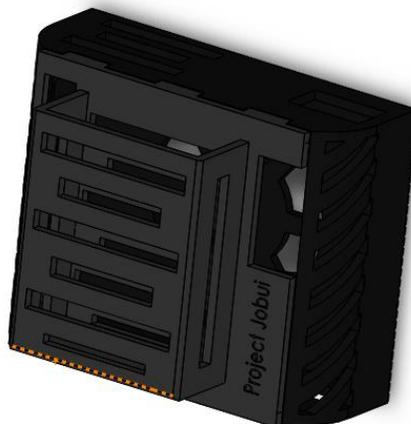
**Figura 3.20:** Diseño CAD del chasis [Autor].

En la figura 3.20, se aprecia el chasis donde va estar todos los componentes electrónicos, como: la tarjeta de desarrollo, la baqueta PBC, el convertidor de voltaje; se puede detallar orificios de los puertos y otros orificios para que le trasmite aire a ciertos puntos, de esa forma llegaría evitar en calentar la tarjeta y una ventaja en reducir gasto de material al momento de imprimir.



**Figura 3.21:** Diseño CAD de la tapa del chasis [Autor].

En la figura 3.21, se observa la tapa superior en donde va ser ensamblada en la parte superior del chasis. Tiene un espacio para colocar la batería, un orificio que permite ver la carga de la batería, otro es orificios es para las conexiones de los sensores y por ultimo los demás orificios es para lograr una ventilación a la tarjeta. Esta tapa permite tener una seguridad a los componentes electrónicos.



**Figura 3.22:** Diseño CAD del chasis completo [Autor].

En a la figura 3.22, se puede detallar el chasis completo, en que logra tener un peso aproximadamente de 56 gramos y está hecho del material Ácido Poliláctico (PLA) que es un termoplástico biodegradable, hecho con base de recurso renovable. Mas adelante será anexado una correa de seguridad en la parte trasera del chasis, eso será para sostenerse en la cintura de la persona y permitir una mejor movilidad al momento de desplazarse.



**Figura 3.23:** Chasis del prototipo impreso en 3D [Autor].

En la figura 3.23, se aprecia el resultado final del diseño con la ayuda de la impresa 3D. Con esto se procede a colocar la parte electrónica en su interior.

### 3.3 Diseño del traje

En la parte del diseño se ha dicho antes que está conformada por 6 sensores Flex, cables UTP, una batería LiPo 7.5 v, un convertidor de voltaje, una tarjeta de desarrollo NodeMcu Esp 32, borneras, un chasis, correa de seguridad, etc. Dicho lo anterior se puede decir que está estructurado de forma de canguro, Como se puede en ver en la figura 3.24.



**Figura 3.24:** Resultado final de la Placa y el chasis [Autor].



**Figura 3.25:** Vista frontal del traje con los sensores Flex [Autor].

En la figura 3.25, se puede apreciar el traje con unos de sus componentes principales que son los sensores Flex; la cual están colocadas en donde se presentan el mayor movimiento de las articulaciones inferiores que en este caso son: la cadera, rodilla y tobillo.

En último lugar en la figura 3.26, se puede observar el canguro puesto en el traje con sus respectivos componentes electrónicos y en otra parte, el cableado que van conectados los sensores hacia la tarjeta donde está el chasis. De esta manera, se logra tener una comunicación por medio del puerto USB o inalámbrica, portabilidad, versatilidad y simplicidad con el de la captura de datos en movimiento libre en campo abierto.

Dicho lo anterior, el prototipo tiene un rendimiento que permite la captura de datos con una velocidad de frecuencia ajustable de 80 MHz a 240 MHz. Ya que es un módulo de uso general Wifi + BT + BLE MCU.



**Figura 3.26:** Vista trasera del traje con el canguro de control [Autor].

### 3.4 Comunicación de tarjeta NodeMcu Esp 32

Para el desarrollo de la comunicación y programación de la tarjeta, primero que todo hay que descargar unos archivos en el repositorio "GitHub" en la página Espressif Systems. La cual se llama **arduino-esp32** (Ver figura), en la misma Plataforma dan una serie de instrucciones para instalar y trabajar en la IDE (Entorno de Desarrollo Integrado) de Arduino; obviamente hay que instalar el programa Arduino, dado a eso tiene un entorno muy sencillo de usar y donde se puede escribir toda promoción que necesite ejecutar [44].

github.com/espressif/arduino-esp32

Tarjetas Para Desar... Home Feed | Puerta... (1) (PDF) Fundamen... tesis fire duty fire duty – Firebase... Ver Venom (2018... Ver Ant-Man y la...

Arduino core para el ESP32

arduino esp-idf plataforma esp32

1,071 confirmaciones 7 sucursales 0 paquetes 20 lanzamientos 281 contribuyentes LGPL-2.1

Rama: Maestro Nueva solicitud de extracción Encontrar archivo Clonar o descargar

Commit	Descripción	Fecha
Gitshaoxiang actualiza los pines M5Camera (# 4021)	Actualizar install-platformio-esp32.sh	el mes pasado
núcleos / esp32	Hardware Corrección de errores serie y mejora (# 3713)	hace 18 días
docs	Docs para lib builder (# 3415)	hace 6 meses
bibliotecas	actualizar los pines M5Camera (# 4021)	ayer
paquete	Agregar cadena de herramientas ARM	hace 8 meses
herramientas	Permitir el uso de scripts de enlazado personalizados (# 3735)	hace 19 días
variantes	Agregar placa M5Stack-ATOM (# 3883)	el mes pasado
.gitignore	Expanda gitignore para cubrir archivos creados por Visual Micro (# 2934)	hace 11 meses
.gitmodules	Establezca ESP-IDF en 3.2 (# 2662)	hace 14 meses

**Figura 3.27:** Vista de los archivos para descargar Flex [44].

Dicho lo anterior, hay que verificar si la IDE de Arduino ya tiene los archivos instalados. Para identificar, el primer paso es ir al programa, dar la opción herramientas; seleccionar la placa que en este caso es “ESP32 Dev Module” (ver figura 3.27), el segundo paso es la selección de herramientas necesarias a poder empezar a programar una de ellas son la velocidad de carga (upload speed), Frecuencia de CPU (CPU frequency), la selección del puerto, entre otros. Como se puede ver en la figura 3.28.

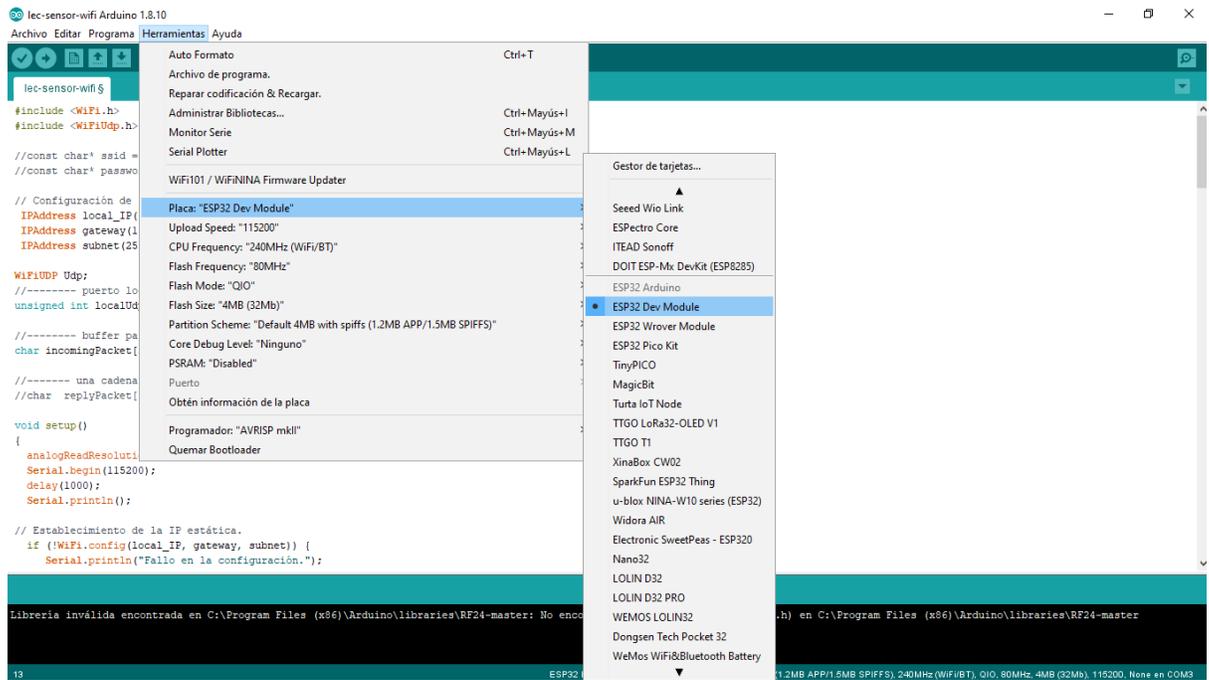


Figura 3.28: Vista del IDE Arduino para selección de la placa [Autor].

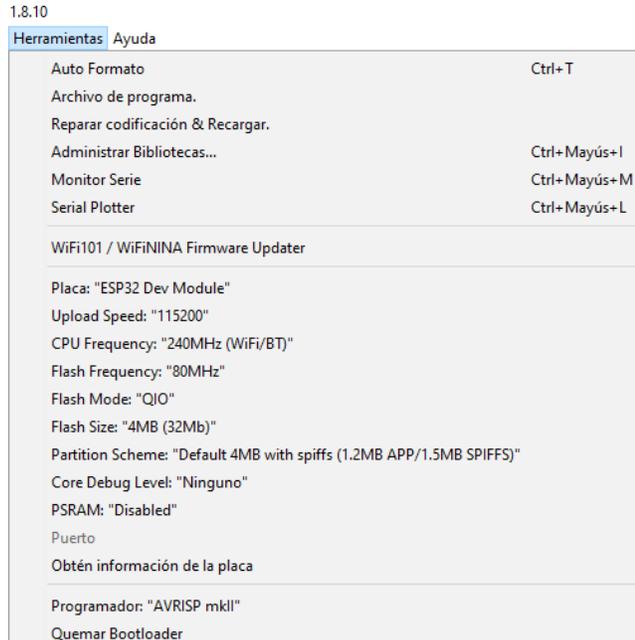


Figura 3.29: Vista del IDE Arduino para selección de herramientas [Autor].

Para realizar la programación, primero hay que indagar un poco sobre sobre la funcionalidad de Wifi ESP32; en donde se puede encontrar ejemplos, debates, opiniones en páginas de block, forum.arduino.c, entre otros sobre el tema. Dado como resultado unas de las opciones de comunicación esta UDP/IP y TCP/IP; hay que resalta que esta es una manera más factible para poder trabajar con otros programas como en el caso de Matlab.

Dicho lo anterior, UDP es el protocolo escogido para la comunicación entre el desarrollo del prototipo. Es importante descargar la librería que se va trabajar en IDE de Arduino, y así poder trabajar con el protocolo de comunicación UPD; pero primero, es importante conocer un poco del tema, a continuación, se dará a conocer.

### **-Comunicación UPD**

Dentro este marco, UDP se conoce como (User Datagram Protocol, protocolo de datagrama de usuario); la cual proporciona puertos de protocolos utilizados para distinguir entre muchos programas que se ejecutan en la misma máquina. Esto es, además de los datos, cada mensaje UDP contiene tanto el número de puerto de destino, como el número de puerto de origen, haciendo posible que el software UDP en el destino entregue el mensaje al receptor correcto y que este envíe una respuesta.

UDP utiliza el Protocolo de Internet (IP) como protocolo subyacente para transportar un mensaje de una máquina a otra y al igual que este brinda servicio de entrega:

No orientado a la conexión: No emplea acuses de recibo para asegurarse de que llegan mensajes, no ordena los mensajes entrantes, ni proporciona retroalimentación para controlar la velocidad a la que fluye la información entre las máquinas. Por lo tanto, los mensajes UDP se pueden, duplicar o llegar sin orden.

No fiable: Un programa de aplicación que utiliza el UDP acepta toda la responsabilidad por el manejo de problemas de confiabilidad, incluyendo la pérdida, duplicación y retraso de los mensajes, entrega fuera de orden y la pérdida de conectividad [45].

Por lo tanto, se menciona algunas de las principales características del protocolo UPD:

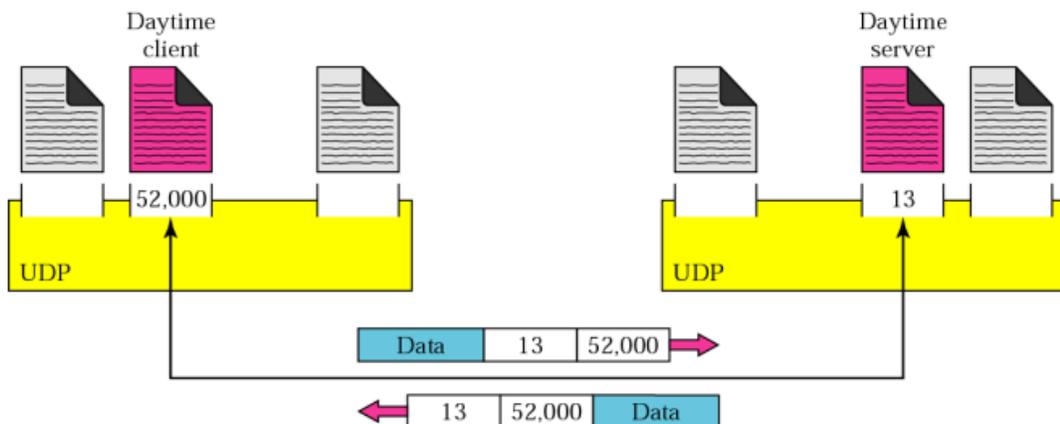
1. El protocolo UDP funciona sin conexión: el protocolo UDP se caracteriza porque permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión entre el emisor y el receptor.
2. UDP utiliza puertos: al igual que el TCP, el protocolo UDP utiliza puertos para permitir que los datagramas se transfieran a los protocolos correctos, es decir, a

las aplicaciones elegidas del sistema de destino.

3. El protocolo UDP permite una comunicación rápida y sin retardos: el protocolo de transporte es el adecuado para una transmisión de datos rápida debido a que no hay que llevar a cabo una configuración de la conexión.

4. El protocolo UDP no ofrece ninguna garantía de seguridad e integridad de los datos: la ausencia de acuse de recibo mutuo entre el emisor y el receptor garantiza que la velocidad de transmisión en el protocolo UDP sea excelente; no obstante, el protocolo no puede garantizar la seguridad ni la integridad de los datagramas [46].

LA figura 3.30, se muestra un ejemplo de la funcionalidad de la comunicación UDP, donde el cliente diurno envía los datagramas con la ayuda de IP hacia el servidor diurno; de esta forma logra devolver nuevamente la información llegada hacia al cliente.



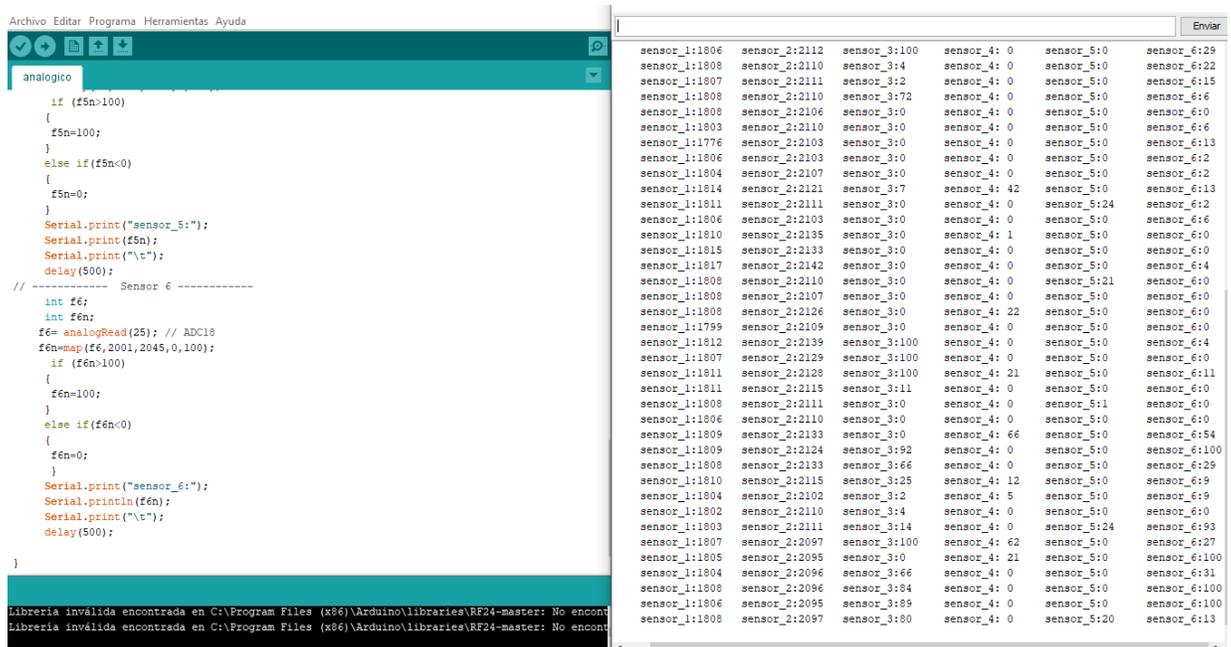
**Figura 3.30:** Esquema de Servicio UDP [45].

### 3.4.1 Servidor UDP

Ahora, teniendo claridad de la temática de UDP, se procede a realizar la programación en IDE de Arduino; para comenzar se debe identificar una dirección IP con su contraseña y puerto local, en donde será la herramienta principal para la comunicación y poder enviar los datos que en este caso al cliente. La programación se puede apreciar en **Anexos II**. Por otra parte, para la lectura de datos de los sensores Flex, primero se deben calibrar cada uno, ya que esto va dependiendo la compostura del cuerpo de la persona.

**Nota:** Antes de subir o copilar el código hacia la tarjeta, se debe mantener presionado el pulsado "IO0" unos segundos para que cargue el código y para resetear con el pulsador "EN".

En la figura 3.31, se observa la primera prueba en donde se hace lectura de todos los sensores Flex. Para ello, se debe colocar el traje completo con sus respectivas conexiones de cada sensor en cada pin de la tarjeta. Mencionado lo anterior, se debe calibrar los sensores para obtener los valores más convenientes al momento de trabajar. En ese mismo modo, el código se encuentra en la espera del que el cliente le envíe un mensaje por el protocolo UDP, para que el servidor le envíe una respuesta que en este caso son los datos de los sensores.



```
Archivo Editar Programa Herramientas Ayuda
analogico
if (f5n>100)
{
  f5n=100;
}
else if(f5n<0)
{
  f5n=0;
}
Serial.print("sensor_5:");
Serial.println(f5n);
Serial.print("\t");
delay(500);
// ----- Sensor 6 -----
int f6;
int f6n;
f6= analogRead(25); // ADC18
f6n=map(f6,2001,2045,0,100);
if (f6n>100)
{
  f6n=100;
}
else if(f6n<0)
{
  f6n=0;
}
Serial.print("sensor_6:");
Serial.println(f6n);
Serial.print("\t");
delay(500);
}
Librería inválida encontrada en C:\Program Files (x86)\Arduino\libraries\RF24-master: No encontrada
Librería inválida encontrada en C:\Program Files (x86)\Arduino\libraries\RF24-master: No encontrada
```

```
sensor_1:1806 sensor_2:2112 sensor_3:100 sensor_4: 0 sensor_5:0 sensor_6:29
sensor_1:1808 sensor_2:2110 sensor_3:4 sensor_4: 0 sensor_5:0 sensor_6:22
sensor_1:1807 sensor_2:2111 sensor_3:2 sensor_4: 0 sensor_5:0 sensor_6:15
sensor_1:1808 sensor_2:2110 sensor_3:72 sensor_4: 0 sensor_5:0 sensor_6:6
sensor_1:1808 sensor_2:2106 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1803 sensor_2:2110 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:6
sensor_1:1776 sensor_2:2103 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:13
sensor_1:1806 sensor_2:2103 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:2
sensor_1:1804 sensor_2:2107 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:2
sensor_1:1814 sensor_2:2121 sensor_3:7 sensor_4: 42 sensor_5:0 sensor_6:13
sensor_1:1811 sensor_2:2111 sensor_3:0 sensor_4: 0 sensor_5:24 sensor_6:2
sensor_1:1806 sensor_2:2103 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:6
sensor_1:1810 sensor_2:2135 sensor_3:0 sensor_4: 1 sensor_5:0 sensor_6:0
sensor_1:1815 sensor_2:2133 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1817 sensor_2:2142 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:4
sensor_1:1808 sensor_2:2110 sensor_3:0 sensor_4: 0 sensor_5:21 sensor_6:0
sensor_1:1808 sensor_2:2107 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1808 sensor_2:2126 sensor_3:0 sensor_4: 22 sensor_5:0 sensor_6:0
sensor_1:1799 sensor_2:2109 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1812 sensor_2:2139 sensor_3:100 sensor_4: 0 sensor_5:0 sensor_6:4
sensor_1:1807 sensor_2:2129 sensor_3:100 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1811 sensor_2:2128 sensor_3:100 sensor_4: 21 sensor_5:0 sensor_6:11
sensor_1:1811 sensor_2:2115 sensor_3:11 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1808 sensor_2:2111 sensor_3:0 sensor_4: 0 sensor_5:1 sensor_6:0
sensor_1:1806 sensor_2:2110 sensor_3:0 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1809 sensor_2:2133 sensor_3:0 sensor_4: 66 sensor_5:0 sensor_6:54
sensor_1:1809 sensor_2:2124 sensor_3:92 sensor_4: 0 sensor_5:0 sensor_6:100
sensor_1:1808 sensor_2:2133 sensor_3:66 sensor_4: 0 sensor_5:0 sensor_6:29
sensor_1:1810 sensor_2:2115 sensor_3:25 sensor_4: 12 sensor_5:0 sensor_6:9
sensor_1:1804 sensor_2:2102 sensor_3:2 sensor_4: 5 sensor_5:0 sensor_6:9
sensor_1:1802 sensor_2:2110 sensor_3:4 sensor_4: 0 sensor_5:0 sensor_6:0
sensor_1:1803 sensor_2:2111 sensor_3:14 sensor_4: 0 sensor_5:24 sensor_6:93
sensor_1:1807 sensor_2:2097 sensor_3:100 sensor_4: 62 sensor_5:0 sensor_6:27
sensor_1:1805 sensor_2:2095 sensor_3:0 sensor_4: 21 sensor_5:0 sensor_6:100
sensor_1:1804 sensor_2:2096 sensor_3:66 sensor_4: 0 sensor_5:0 sensor_6:31
sensor_1:1808 sensor_2:2096 sensor_3:84 sensor_4: 0 sensor_5:0 sensor_6:100
sensor_1:1806 sensor_2:2095 sensor_3:89 sensor_4: 0 sensor_5:0 sensor_6:100
sensor_1:1808 sensor_2:2097 sensor_3:80 sensor_4: 0 sensor_5:20 sensor_6:13
```

Figura 3.31: Lectura de los sensores usando IDE Arduino [Autor].

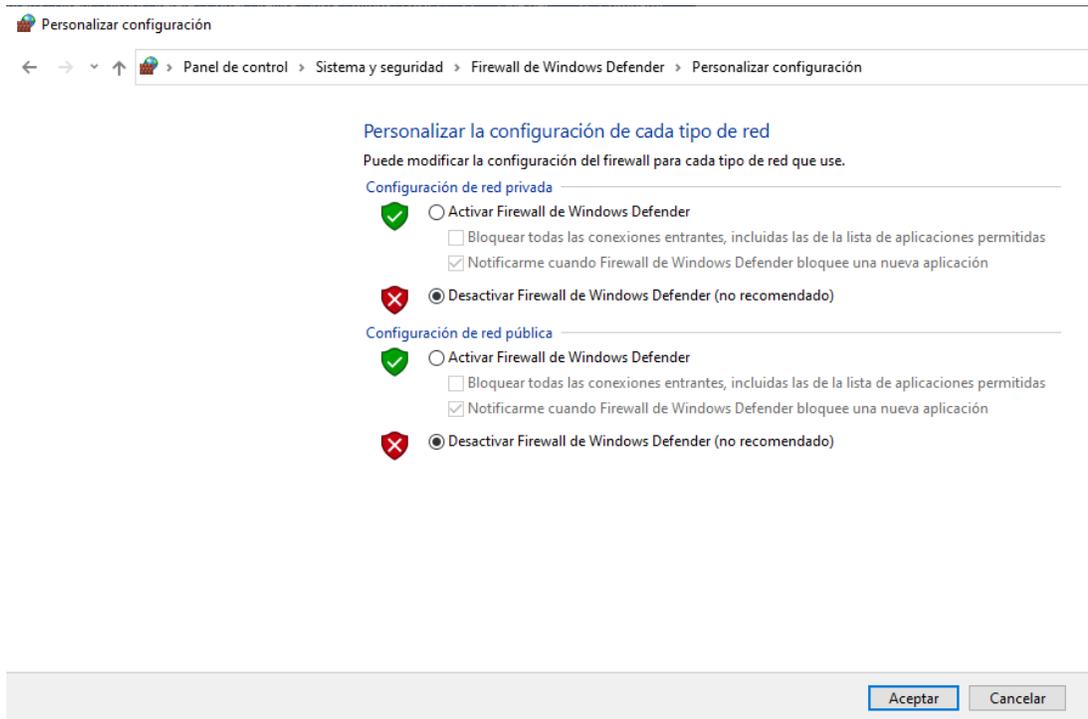
### 3.4.2 Cliente UDP

La comunicación del cliente que en este caso va ser Software Matlab (abreviatura MATrix LABratory, laboratorio de matrices), como bien sabe, es un programa computacional que ejecuta una gran variedad de operaciones y ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Para empezar, se debe tener el programa instalado, una de sus herramientas principales es que tiene múltiples funciones; la cual se encuentra la función judp.m: donde utiliza la interfaz Java de Matlab para manejar las comunicaciones del Protocolo de datagramas de usuario (UDP) con otra aplicación, ya sea en la misma computadora o en una remota.

Un ejemplo de la función judp.m esta la siguiente:

JUDP ('ENVIAR', PUERTO, HOST, MSSG) envía un mensaje al puerto y host especificados. HOST puede ser un nombre de host (por ejemplo, 'www.example.com') o una representación de cadena de una dirección IP (por

ejemplo, '192.0.34.166'). El puerto es un número de puerto entero entre 1025 y 65535. El puerto especificado debe estar abierto en el firewall de la máquina receptora, es decir, que la PC debe tener desactivado el Firewall de Windows Defender, ya que su función es impedir que hackers o software malintencionado obtengan acceso a la PC a través de internet o una red. La desactivación se puede ver en la figura 3.32.

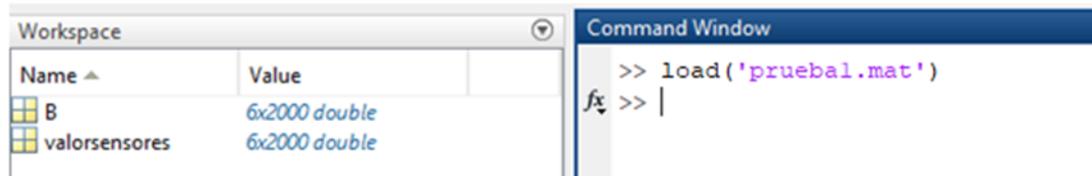


**Figura 3.32:** Activar o desactivar Firewall de Windows Defender [Autor].

Lo anterior dicho, se procede a realizar la programación en donde se debe tener en detalle la misma dirección IP y número de puerto que se hizo en la programación IDE de Arduino. Los mensajes enviados por judp.m están en formato int8. La función int8.m se puede usar para convertir enteros o caracteres al formato correcto (use double.m o char.m para volver a convertir después de que se reciba el datagrama). Los valores no enteros se pueden convertir a formato int8 usando typecast.m (use typecast.m para volver a convertir). El script de cliente será guardado con el nombre "estado\_sensores.m". Por último, los datos serán guardados en una variable B formando una matriz nxm.

Una vez terminado lo anterior, se realiza el funcionamiento del traje sensorizado, donde ya copiado el código IDE de Arduino, comienza a cargar los datos de los sensores y de ahí, es donde la persona comienza a desplazarse de un punto a otro punto, cumpliendo con los ciclos y fases. En esa secuencia se ejecuta el código de "estado\_sensores.m" en Matlab; en este caso, comienza a recibir los

datos secuenciales a lo que la persona se va desplazando. Dicho eso, se va almacenando en una matriz de 6x2000 en una variable B y se procede a guarda como “prueba1” recolectando en si 2000 datos por cada sensor, en un tiempo aproximado de 234 segundos que eso equivale a 190 periodos, (ver figura 3.33). el código se encuentra en **Anexo III**. A continuación, en la figura 3.34 se mostrará la respuesta obtenida en la ventana de Workspace de Matlab. Este procedimiento se realizó 4 veces para tener varios tipos de pruebas, en que luego será utilizados más adelanté.



**Figura 3.33:** Captura de paquetes UDP guardado en Matlab [Autor].

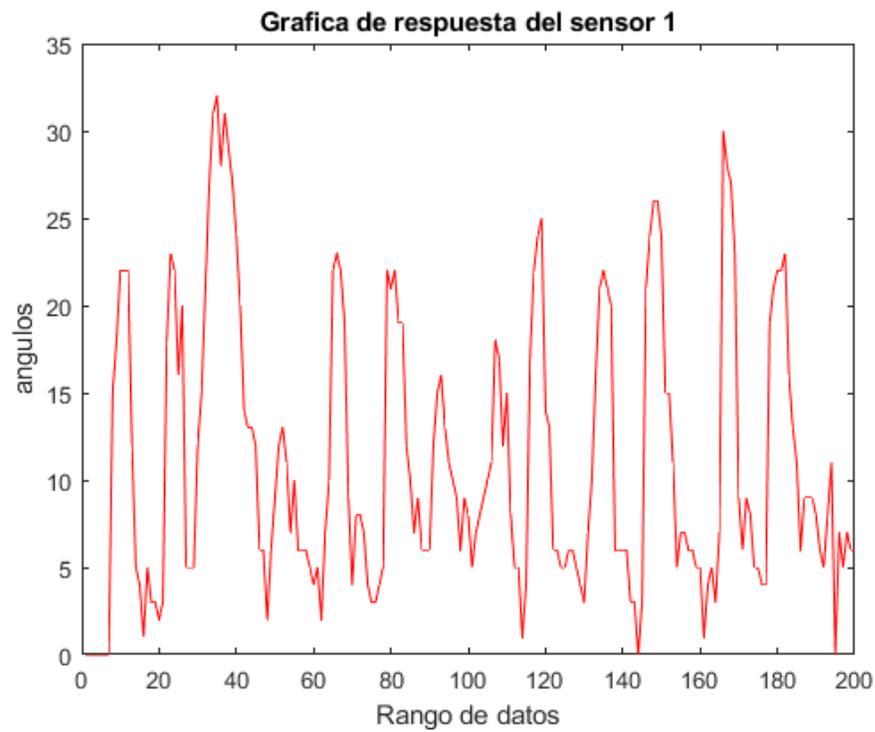
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	3	9	20	21	18
2	9	9	10	10	12	14	14	14	13	13	13	13
3	14	13	14	13	13	14	13	13	14	14	14	13
4	24	24	23	24	29	37	45	51	55	40	35	33
5	45	45	44	43	32	27	28	27	26	45	31	31
6	59	59	60	61	65	71	72	72	72	74	75	76

**Figura 3.34:** tabla de los datos capturados del servidor UDP [Autor].

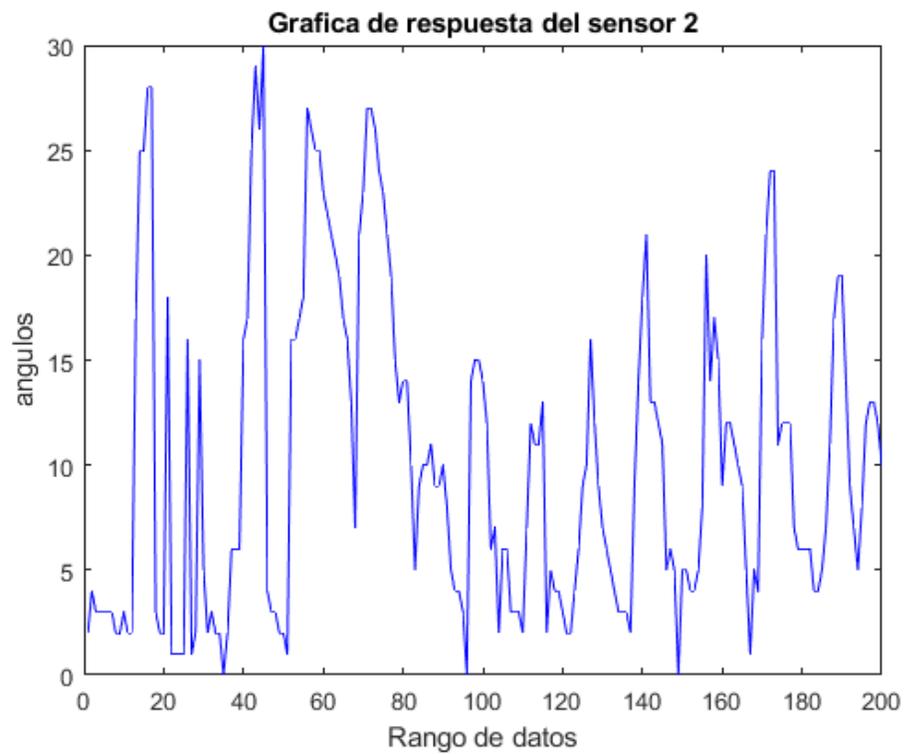
**Tabla 3.4:** Duración de tiempo para toma de datos

Prueba	Tiempo(segundos)
1	234.4053
2	226.8011
3	230.4222
4	229.1687
Promedio	230.1993

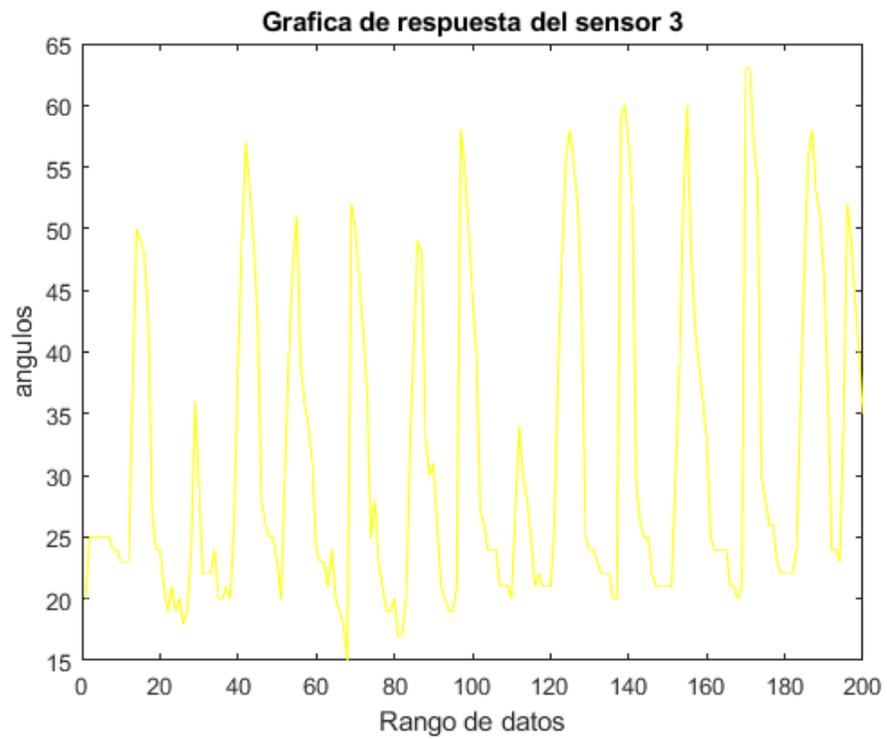
Teniendo en cuenta los datos anteriores, se procese a mostrar el comportamiento de cada sensor gráficamente en la posición de las articulaciones de la primera prueba. Esto permite ver la trayectoria y los factores de cada articulación que son cadera, rodilla y tobillo; al momento de desplazarse en línea recta. Como se puede observar en las siguientes figuras.



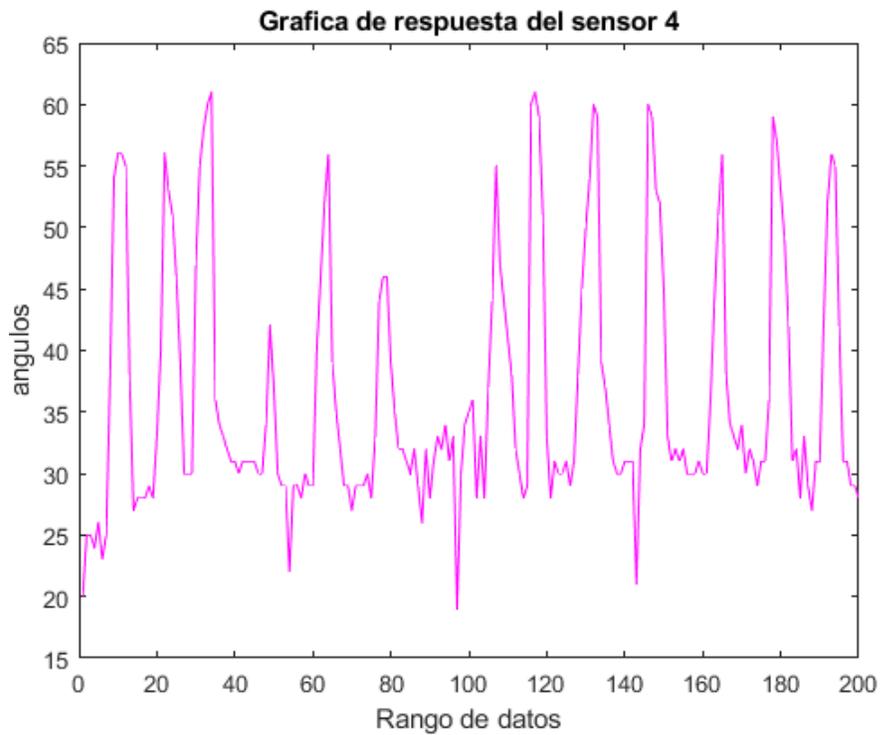
**Figura 3.35:** Grafica del movimiento de la cadera derecha [Autor].



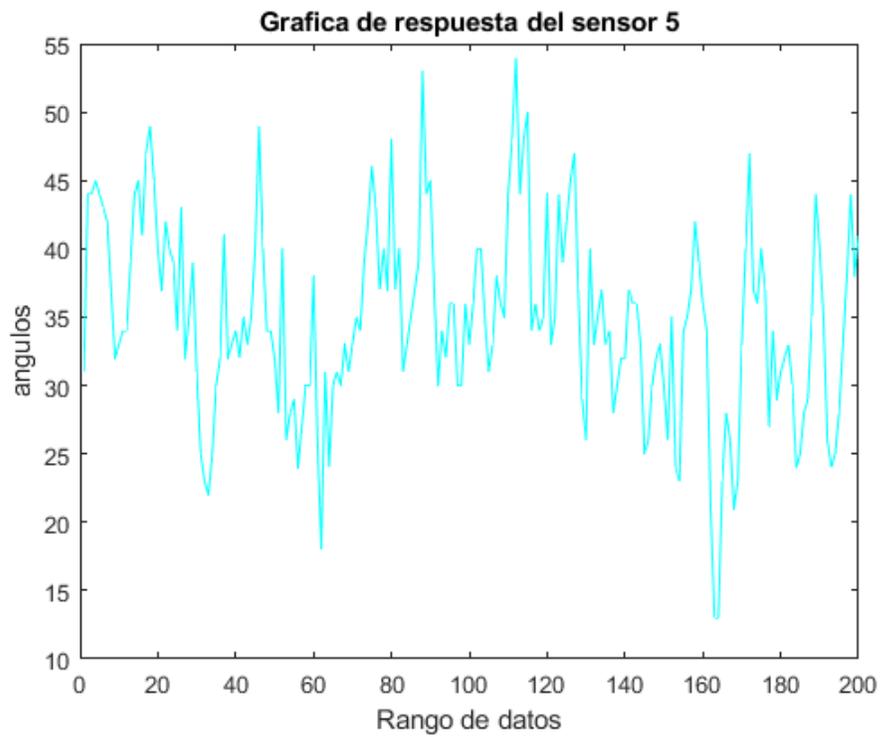
**Figura 3.36:** Grafica del movimiento de la cadera izquierda [Autor].



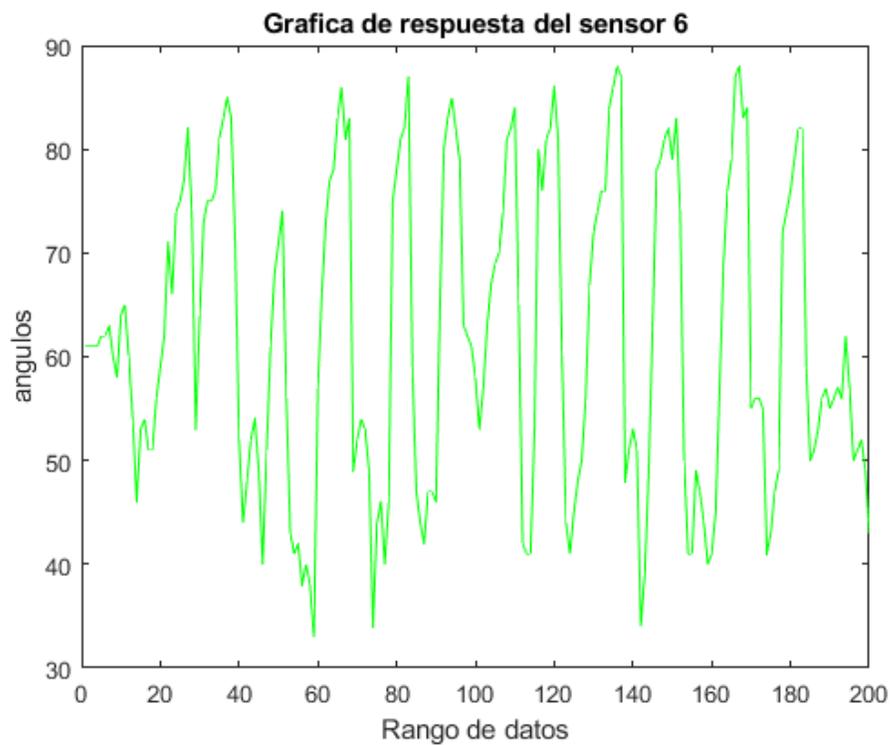
**Figura 3.37:** Grafica del movimiento de la rodilla izquierda [Autor].



**Figura 3.38:** Grafica del movimiento de la rodilla derecha [Autor].



**Figura 3.39:** Grafica del movimiento del tobillo derecho [Autor].

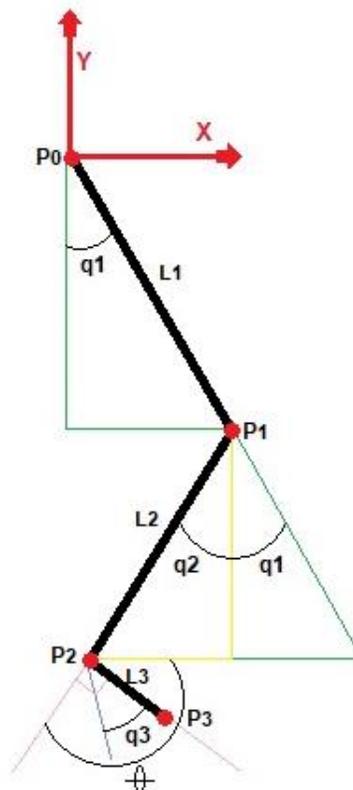


**Figura 3.40:** Grafica del movimiento del tobillo izquierda [Autor].

### 3.5 Cinemática Directa

Para obtener una comprobación de la simulación de los datos de prueba, se debe contar previamente con un completo análisis cinemático del futuro mecanismo, para ello es necesario definir su configuración geométrica, la cual está constituida por los eslabones, unidos entre sí mediante junta rotacionales. La cadera en este caso se toma como movimiento de flexión-extensión (eje de giro normal al plano sagital). Para la rodilla, también se toma como movimiento de flexión-extensión, tomando como una junta rotacional. Para el movimiento del tobillo se toma e igualmente como movimiento de flexión-extensión. Teniendo en cuenta lo anterior, se debe tener mínimo 12 GDL para lograr una marcha cercana a la locomoción humana. Cosa que en este caso se aplica porque en cada pierna se tienen 6 GDL.

Para hallar la cinemática directa hay varios métodos conocidos, para este proyecto se utilizó el método geométrico. En el cual es necesario establecer un marco de referencia general que es un punto fijo (0,0). Para este el marco de referencia se ubicó en la parte superior como P0, asimilando la posición de la cadera, y así determinando el eje z colineal al eje de giro de esa articulación, el eje x horizontal y el eje y vertical. Teniendo en cuenta en la figura 3.41, que para el movimiento angular de cada articulación se toma el sentido de giro antihorario.



**Figura 3.41:** Geografía del mecanismo [Autor].

De acuerdo con la figura 3.41, se obtienen las ecuaciones posición de cada eslabón, definiendo la dirección del movimiento en que se expresa como:

$$P_0 = [0; 0]$$

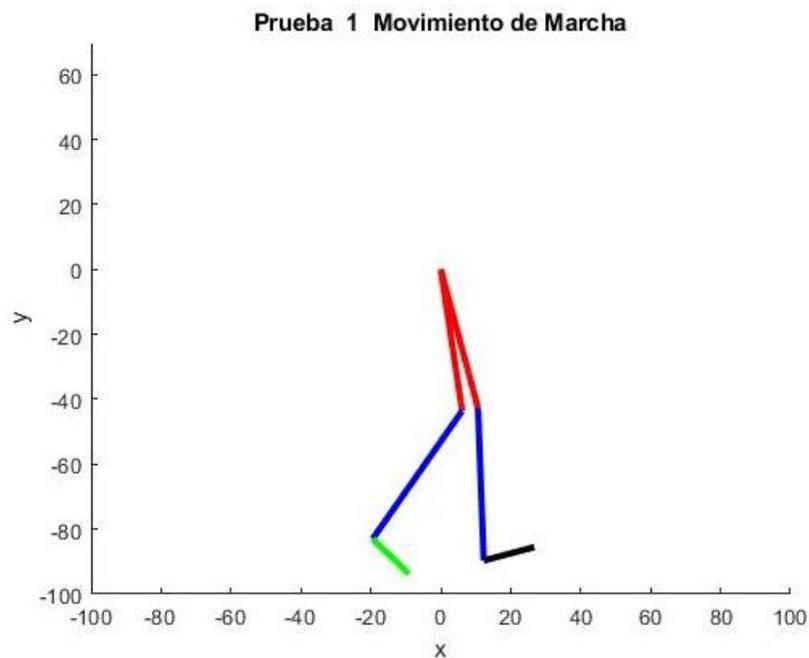
$$P_1 = [L1 * \text{sen}(q1); -L1 * \text{cos}(q1)]$$

$$P_2 = P_1 + [L2 * \text{sen}(q1 + q2); -L2 * \text{cos}(q1 + q2)]$$

$$\theta = q1 + q2 + 90 + q3$$

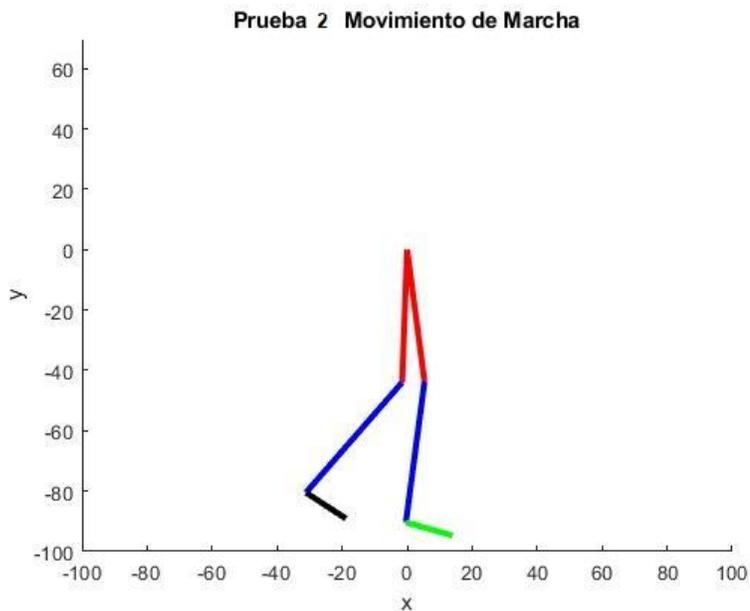
$$P_3 = P_3 + [L3 * \text{sen}(\theta); -L3 * \text{cos}(\theta)]$$

Tenido las ecuaciones se procede a realizar las simulaciones, para ver el comportamiento de la marcha de cada una de las pruebas realizadas en que se puede apreciar a continuación:

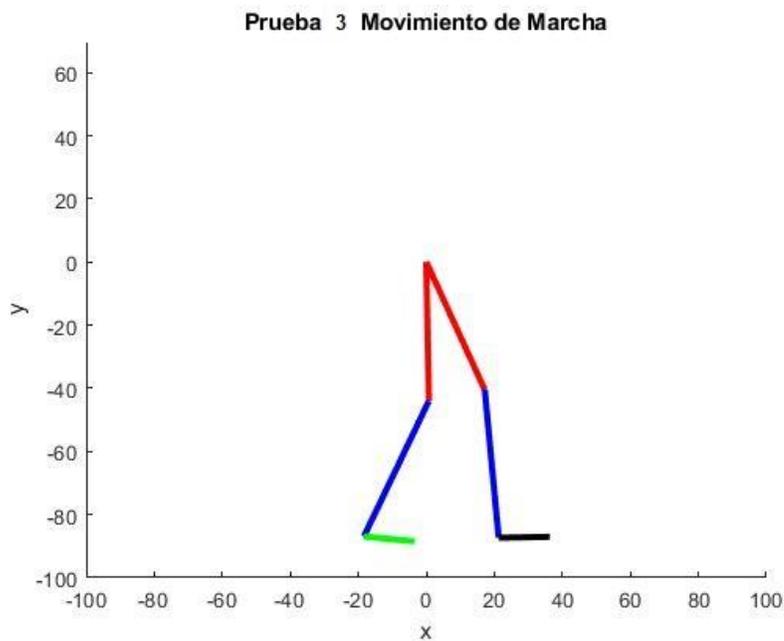


**Figura 3.42:** Grafica de Prueba 1[Autor].

En la figura 3.42, el vector de la línea negra define el pie derecho y la línea verde define el pie izquierdo de igual forma se aplica para las demás figuras siguientes. El código se puede apreciar en **Anexos IV**



**Figura 3.43:** Grafica de Prueba 2[Autor].



**Figura 3.44:** Grafica de Prueba 3[Autor].

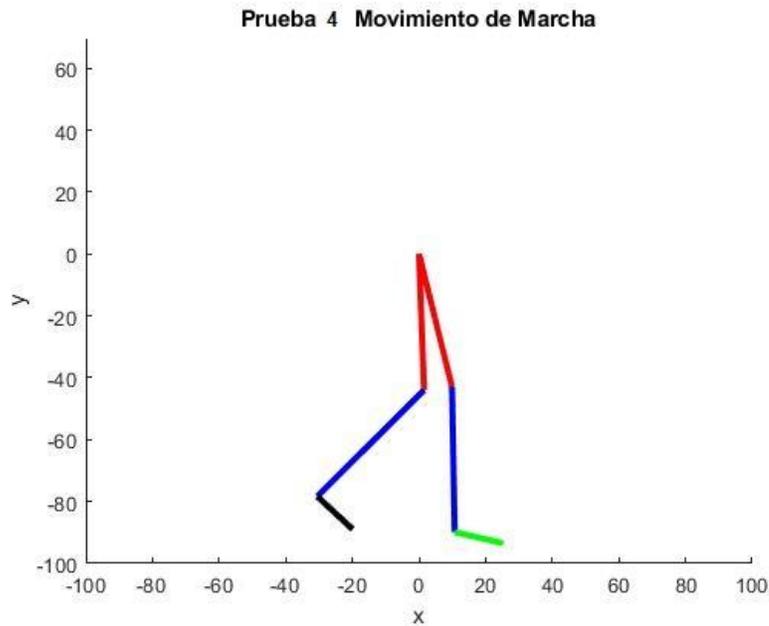


Figura 3.45: Grafica de Prueba 4[Autor].

### 3.6 Diseño de Algoritmo

El diseño de la red neuronal, en la cual se decidió a trabajar en desarrollo del prototipo es la **Regresión Generalizada (GRNN)**. Esta red tiene una capacidad de modelar relaciones desconocida y de naturaleza no lineal. Su arquitectura de red, está conformada por dos capas escondidas: la capa básica radial y la capa lineal especial; se usa a menudo para llegar a funciones cercanas y es similar a la red de base radial, pero tiene una segunda capa ligeramente diferente. En la figura 3.46, podrá apreciar la estructura de GRNN [46].

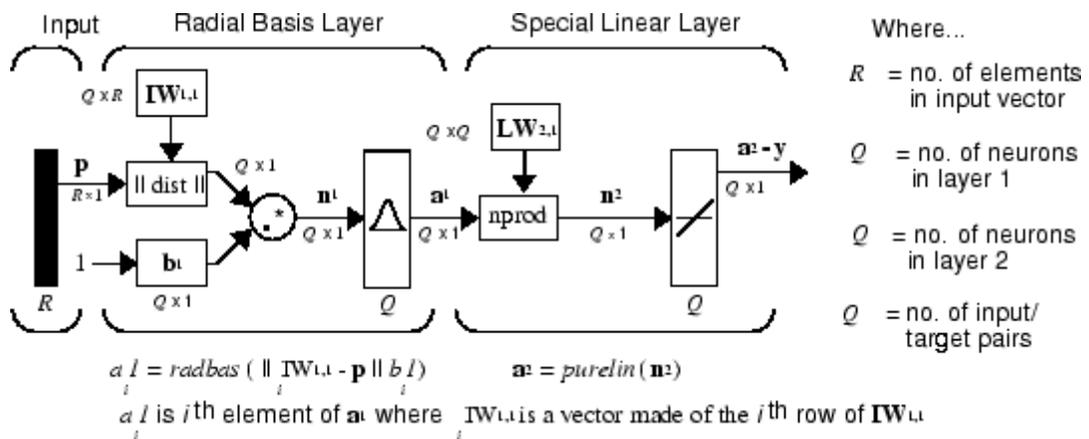


Figura 3.46: Arquitectura del GRNN [46].

Su estructura está conformada por,

R = N°. de elementos en el vector puesto.

Q = N°. de neuronas en la capa 1.

Q = N°. de neuronas en la capa 2.

Q = N°. de pares de entrada / destino.

En el cuadro `nprod` que se muestra arriba (función de código `normprod`) produce elementos  $S^2$  en el vector  $n^2$ . Cada elemento es el producto escalar de una fila de  $LW_{2,1}$  y el vector de entrada  $a^1$ , todo normalizado por la suma de los elementos de  $a^1$ . La primera capa es así para las redes `newrbe`. Tiene tantas neuronas como vectores de entrada / objetivo en P. Específicamente, los pesos de la primera capa se establecen en  $P'$ . El bias  $b^1$  se establece en un vector de columna de  $0.8326 / \text{SPREAD}$  (Difusión de funciones de base radial). El usuario elige `SPREAD`, la distancia que debe ser un vector de entrada del vector de peso de una neurona para ser menor a 1 [46].

### 3.5.1 Diseño de red neuronal de regresión generalizada

Las redes neuronales de regresión generalizada (`grnns`) son un tipo de red de base radial que a menudo se usa para la aproximación de funciones. Las `grnns` se pueden diseñar muy rápidamente. De esta forma su función está dada por la siguiente manera:

$$net = newgrnn(P, T, spread)$$

P = matriz R-por-Q de vectores de entrada Q.

T = matriz S-por-Q de vectores de clase objetivo Q.

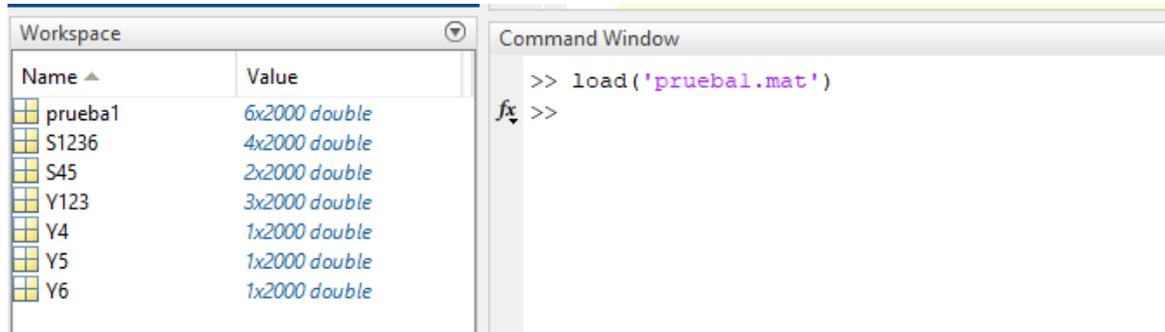
spread = Spread de funciones de base radial (por defecto = 1.0).

#### Propiedades

La `newgrnn` crea una red de dos capas. La primera capa tiene neuronas `radbas` y calcula las entradas ponderadas con `dist` y la entrada neta con `netprod` funciones de Matlab. La segunda capa tiene neuronas de `purelin`, calcula la entrada ponderada con `Normprod` y las entradas netas con `netsum`. Solo la primera capa tiene biases, como se puede detallar la arquitectura en la figura 3.46.

Por último, el `newgrnn` establece los pesos de la primera capa en  $P'$ , y el bias de la primera capa se establecen en  $0.8326 / \text{spread}$ , lo que da como resultado funciones de base radial que cruzan 0.5 en entradas ponderadas de  $\pm \text{spread}$ . Los pesos de la segunda capa  $W_2$  se establecen en T [47].

Conociendo la funcionalidad de la red, se procede hacer la programación. Primero se debe sacar las variables de entrenamiento y las variables de salida deseada; en este caso teniendo los datos de la matriz “prueba1” de 6x2000 ya mencionado anteriormente. Se localiza los valores correspondientes de entrenamiento, que sería los sensores 1, 2, 3 y 6, para concatenar en una nueva matriz de 4x2000 y guardando con un nuevo nombre de variable “S1236.mat”, y igualmente para los valores de salida deseada sería los sensores 4 y 5 formando un matriz de 2x2000, guardando con nuevo nombre S45, ver figura 3.47.



**Figura 3.47:** Datos de nuevas variables guardados en Workspace [Autor].

Así mismo, se puede apreciar los valores de los sensores 1, 2, 3 y 6 como S1236; se puede observar en la figura 3.48.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	15	18	22	22	22
2	2	4	9	9	10	10	10	10	10	10	10	10
3	10	15	15	15	15	15	15	14	14	13	13	14
4	61	61	61	61	62	62	63	60	58	64	65	60

**Figura 3.48:** Datos de entrenamiento (S1236) guardados en Workspace [Autor].

Por otro lado, se determina el sensor 4 y 5 como S45. En la figura 3.49, se muestra una parte de los datos de S45.

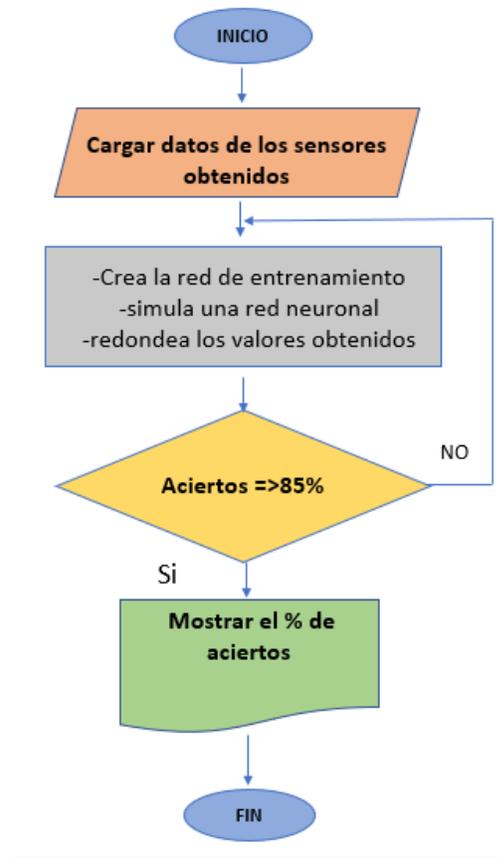
	1	2	3	4	5	6	7	8	9	10	11	12
1	20	25	25	24	26	23	25	38	54	56	56	55
2	31	44	44	45	44	43	42	37	32	33	34	34

**Figura 3.49:** Datos de salida deseada (S45) guardados en Workspace [Autor].

Ahora, todos los datos nuevamente pasan a otra variable para S1236 se transporta estos datos a una ventana de tiempo, en donde se toma los valores del presente, luego los valores del anterior; presumiendo así un después y un antes a los valores de entrenamiento y quedan guardado como S1236iim1; de igual forma

se hace con los valores de salida deseada de S45 en donde quedan guardados como S45i. Esta manera da más eficiencia a los valores de entrenamiento, permitiendo así que la RNA tenga un mejor pronóstico al momento de entrenar. Por lo tanto, se guarda con el nombre de "GRNN1.m"; y se procede a correr el código. luego, se calcula el porcentaje de acierto de la red neuronal con respecto a los valores de salida deseada. Finalmente, se hace pruebas de esta red con los valores de validación de las pruebas 2,3,4 (S1236iim1v) que se hicieron anteriormente, después se guarda red neuronal en un diagrama de bloque con el fin de hacer pruebas futuras para el control de una prótesis. El código se puede observar en **Anexo V**.

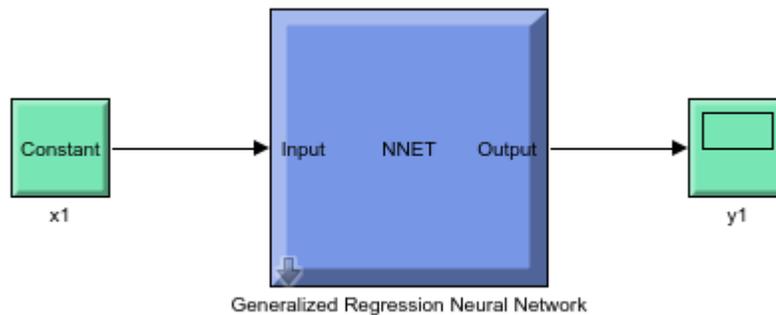
#### 3.5.2 Diagrama de flujo del algoritmo



**Figura 3.50:** Diagrama de flujo del Diseño de Algoritmo [Autor].

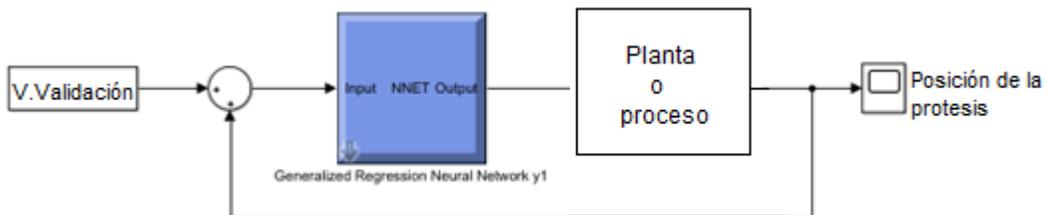
#### 3.6 Esquema de control de la RNA

A continuación, se precede a utilizar el comando *gensim (net, -1)* en el mismo código de “GNN1.m” de la red neuronal ya entrenada (net1). Su función es crear un bloque de la red en Simulink, que es un entorno de programación visual, es una toolbox especial de Matlab que sirve para simular el comportamiento de los sistemas dinámicos; como sistemas lineales, no lineales, y múltiples sistemas o modelos. Para tal efecto se logra obtener el esquema de la salida de validación de la RNA1, como se puede ver la figura 3.46. Donde la variable x1 sería los valores de validación y y1 los valores deseados.



**Figura 3.51:** Bloques de Simulink de la RNA [Autor].

En la figura 3.47, se puede apreciar un lazo de control de la red neuronal, donde se puede apreciar una idea de cómo se haría el control de una prótesis para desarrollos futuros.

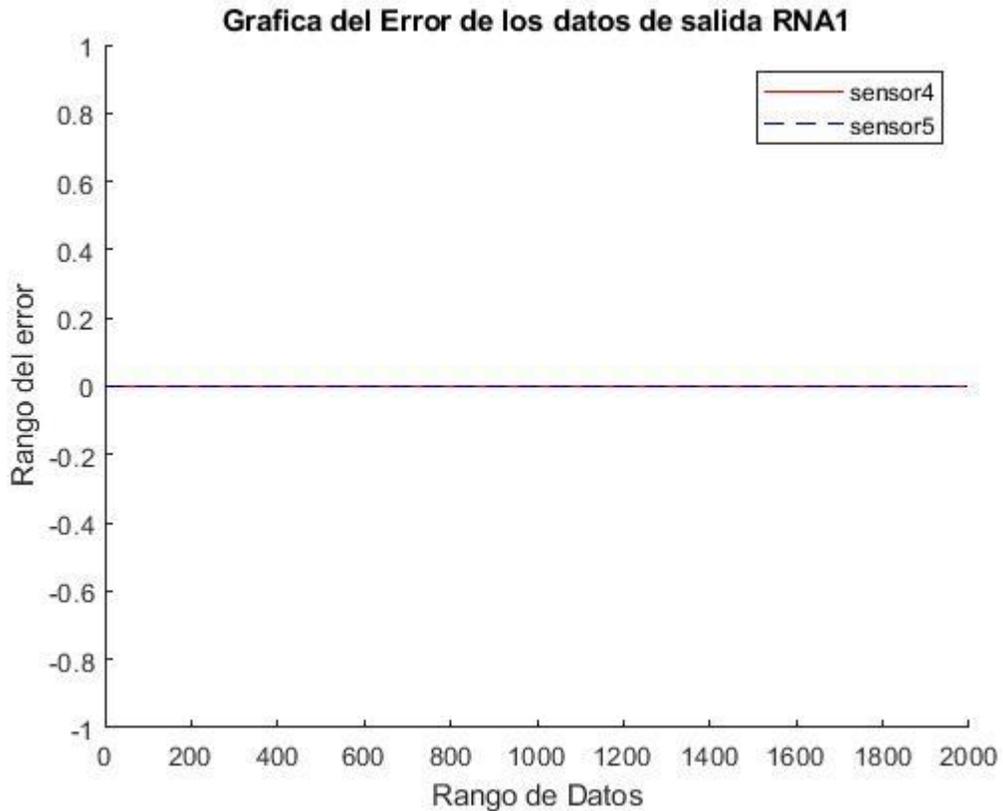


**Figura 3.52:** Esquema de lazo control Cerrado [Autor].

## 4. Análisis de Resultados

Después de implementar el modelo GRNN, se logra tener los resultados del error absoluto y porcentaje de aciertos entre los valores de salida deseada y los valores de RNA1 de las 3 pruebas realizadas, respecto a los sensores 4 y 5.

### 4.1 Respuesta de la RNA1 con los datos de validación de la prueba 1

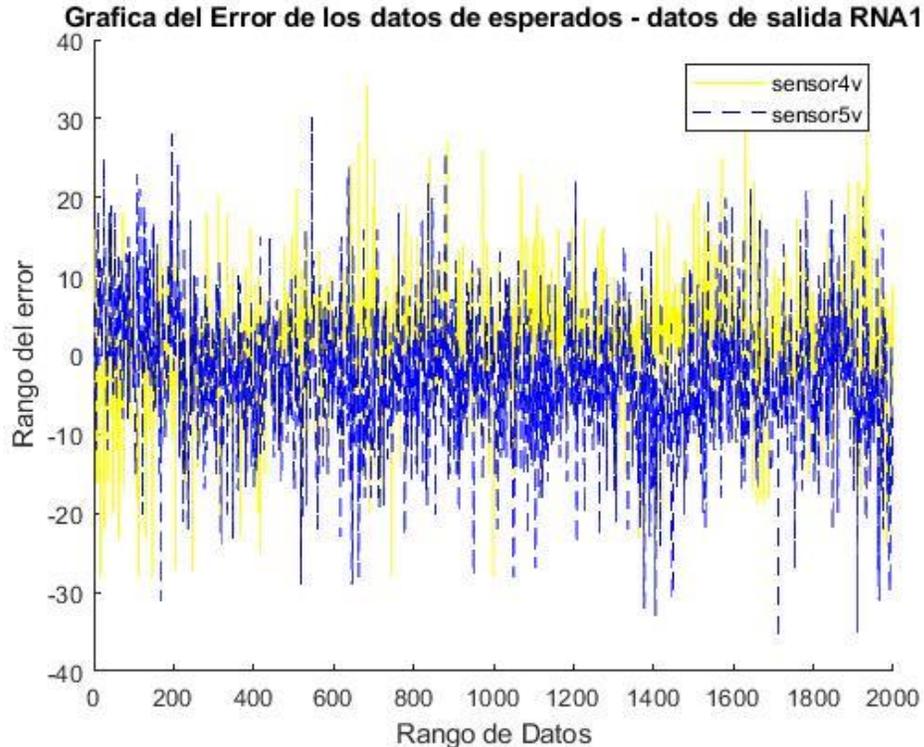


**Figura 4.1:** Gráfica de error absoluto de RNA1 [Autor].

Para la figura 4.1, se puede observar que no presenta ninguna perturbación, dado eso el porcentaje de acierto es 100% para los valores del sensor 4 y 5. Esto se obtuvo con los valores de salida deseada de la prueba 1 menos los valores de la RNA1 que fue entrenada con los valores de entrenamiento de la prueba 1, luego con el valor del error se halla el porcentaje de acierto teniendo en cuenta con el error máximo permitido que se determinó a 2 grados, de esta manera se logra obtener la gráfica.

Para el segundo paso, en la verificación del funcionamiento de la RNA1, se procede calcular con los valores de validación de la prueba 2 (S1236iim1v), los

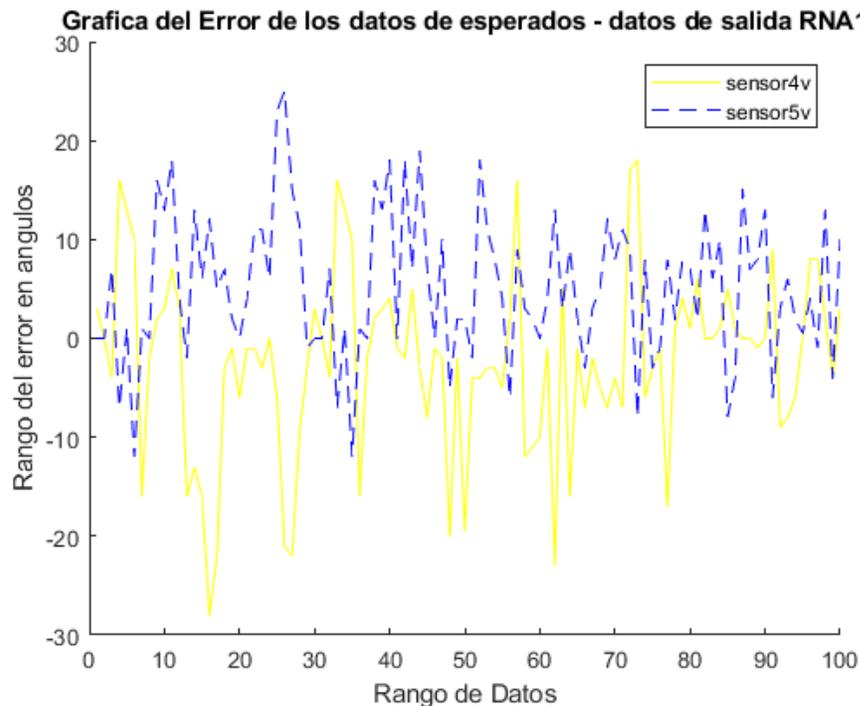
nuevos valores de respuesta de la RNA1, en que luego se halla el error absoluto de los valores deseados de la prueba 2 menos los nuevos valores de respuesta de la RNA1, y de esta forma se calcula el porcentaje de acierto y el comportamiento grafico del error absoluto (ver en la figura 4.2).



**Figura 4.2:** Gráfica del error abs de la prueba 2 de Validación con la RNA1 [Autor].

En la figura 4.2, se puede apreciar el margen de error de los grados que se obtiene entre los valores de validación con respecto a los nuevos valores de la RNA1. El nuevo porcentaje de acierto para este caso se aplica a un error máximo permitido de 15 grados, permitiendo así obtener un acierto para el sensor 4 (rodilla) de 94.84% y para el sensor 5 (tobillo) de 91.39 %. Se puede detallar que la mayoría de picos no se sobre salen a los 15 grados, esto es porque se da un límite de error permitido.

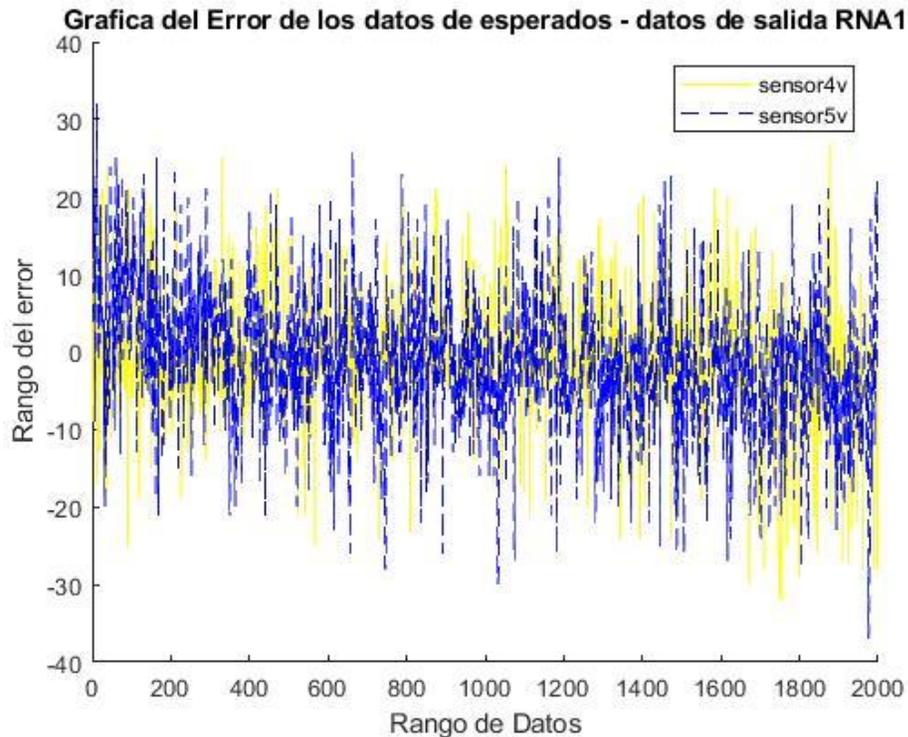
En la figura 4.3, puede observar un fragmento con más detalles de la gráfica de la figura 4.2, permitiendo así ver el comportamiento del desfase que hay entre las articulaciones de la prueba 2 y la RNA1. El error absoluto obtenido entre el movimiento de marcha de la prueba 2 y la RNA1 es de  $1,677^\circ$  para el sensor 4 (rodilla) y  $2,624^\circ$  para sensor 5 (tobillo).



**Figura 4.3:** Gráfica de un fragmento del error abs de la prueba 2 de Validación con la RNA1 [Autor].

### 4.2 Respuesta de la RNA1 con los datos de validación de la prueba 3

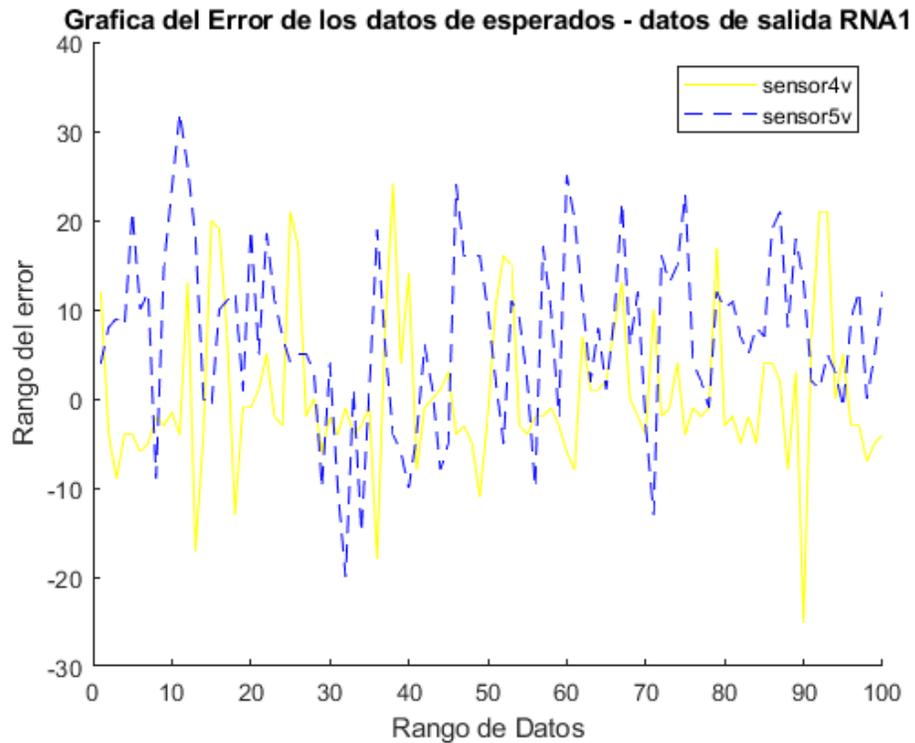
Para la prueba 3, se toma los valores de validación (S1236iim1v) el cual se procede calcular con la RNA1, en donde se obtiene los nuevos valores de salida de la RNA1, en que luego se hace la prueba de error absoluto de los valores deseados de la prueba 3 menos los nuevos valores de salida de la RNA1, y de esta forma se calcula el porcentaje de acierto y el comportamiento grafico del error absoluto (ver en la figura 4.4).



**Figura 4.4:** Gráfica del error abs de la prueba 3 de Validación con la RNA1 [Autor].

En la figura 4.4, se puede apreciar el margen de error de los grados que se obtiene entre los valores de validación de la prueba 3 con respecto a los nuevos valores de la RNA1. El nuevo porcentaje de acierto para este caso se aplica a un error máximo permitido de 15 grados, de esta forma se obtiene un acierto para el sensor 4 de 94.29% y para el sensor 5 de 90.99 %. Se puede detallar que la mayoría de picos no se sobre salen a los 15 grados.

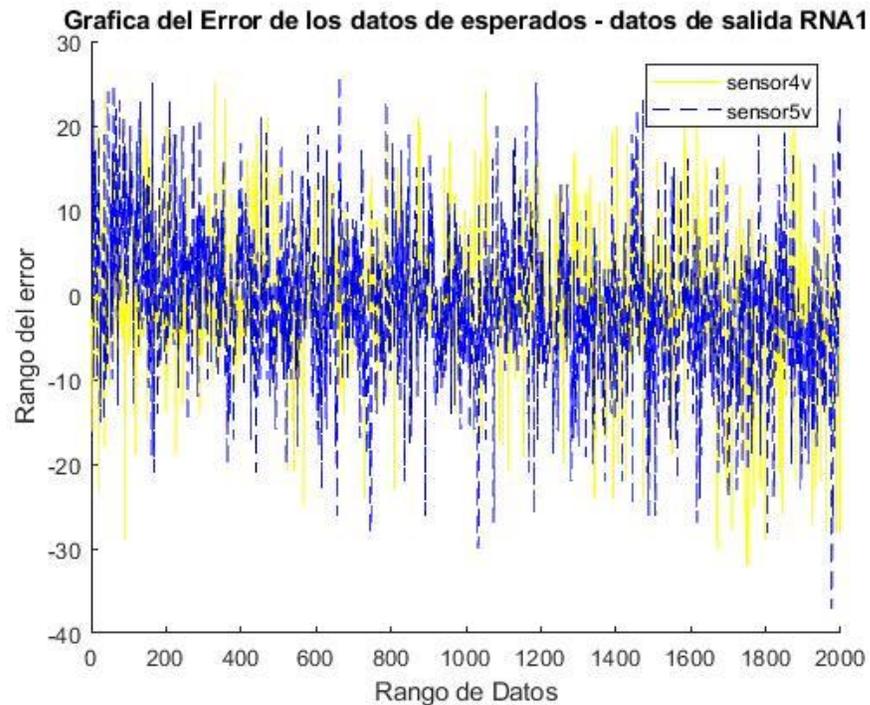
En la figura 4.5, puede observar un fragmento con más detalles de la gráfica de la figura 4.4, permitiendo así ver el comportamiento del desfase que hay entre las articulaciones de la prueba 3 y la RNA1. El error absoluto obtenido por el movimiento de marcha de la prueba 3 es de  $0,833^\circ$  para el sensor 4 (rodilla) y  $1,121^\circ$  para sensor 5 (tobillo).



**Figura 4.5:** Gráfica de un fragmento del error abs de la prueba 3 de Validación con la RNA1 [Autor].

### 4.3 Respuesta de la RNA1 con los datos de validación de la prueba 4

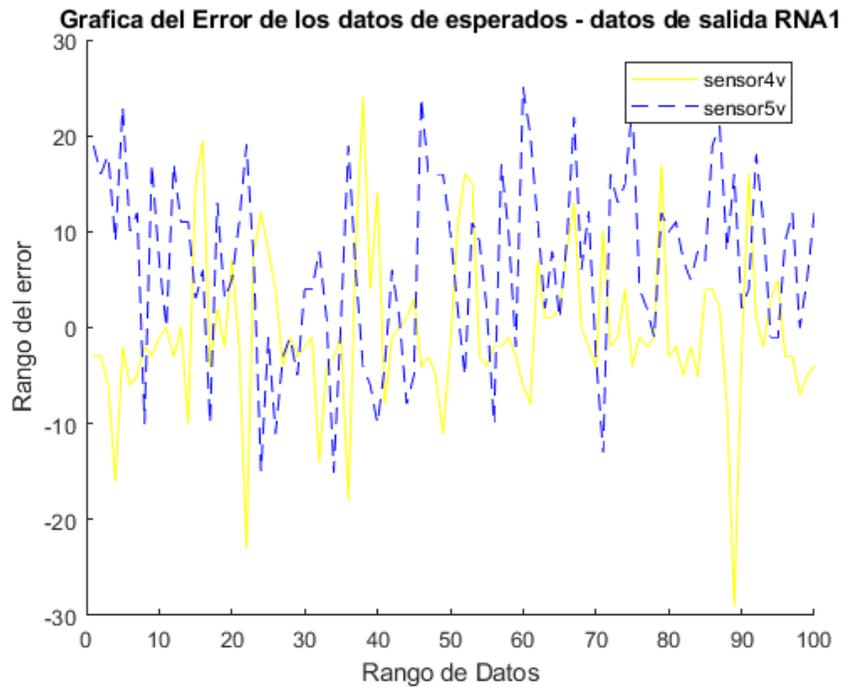
Por último, para la prueba 4 se determina los valores de validación (S1236iim1v) el cual se procede calcular con la RNA1, en donde se obtiene los nuevos valores de salida de la RNA1. Se halla la prueba del error absoluto de los valores deseados de la prueba 4 menos los nuevos valores de salida de la RNA1, y de esta forma se calcula el porcentaje de acierto y el comportamiento grafico del error absoluto (ver en la figura 4.6).



**Figura 4.6:** Gráfica del error abs de la prueba 4 de Validación con la RNA1 [Autor].

Finalmente, en la figura 4.6, se puede apreciar el comportamiento de margen de error de los grados que se obtiene entre los valores de validación de la prueba 4 con respecto a los nuevos valores de la RNA1. El nuevo porcentaje de acierto para este caso se aplica a un error máximo permitido de 15 grados, de esta forma se obtiene un acierto para el sensor 4 de 94.29% y para el sensor 5 de 90.89 %. Esta grafica de la prueba 4 tiene mismas similitudes de la prueba 3 llegando casi el mismo porcentaje de acierto.

En la figura 4.7, puede observar un fragmento con más detalles de la gráfica de la figura 4.6, permitiendo así ver el comportamiento del desfase que hay entre las articulaciones de la prueba 4 y la RNA1. El error absoluto obtenido por el movimiento de marcha de la prueba 4 es de  $0,852^\circ$  para el sensor 4 (rodilla) y  $1,086^\circ$  para sensor 5 (tobillo).



**Figura 4.7:** Gráfica de un fragmento del error abs de la prueba 4 de Validación con la RNA1 [Autor].

Dicho lo anterior, respecto al porcentaje de acierto y el valor del error absoluto. Se logra mencionar el error máximo permitido que es de  $15^\circ$ , este error se tiene en cuenta para tener un mejor porcentaje de acierto. De igual forma, no se necesita una precisión alta, debido a que ya se tiene los patrones de comportamiento o de evolución del proceso de marcha. Es decir, saber cuál va ser el sentido, la posición aproximada, para la futura prótesis.

#### 4.4 Comparación de graficas

A partir de la respuesta obtenida de la RNA1 para cada una de las pruebas, se procede a comparar gráficamente los valores deseados de las articulaciones de la cadera, rodilla y tobillo con respecto al valor de la RNA1. Es necesario destacar que se ha definido la línea punteada para los valores de RNA1, y la línea continua para los valores deseados de cada una de las pruebas.

A continuación, se podrán observar las siguientes gráficas:

4.4.1 Graficas de movimiento de marcha de los valores de la prueba 2 del sensor 4 y 5 vs respuesta RNA1

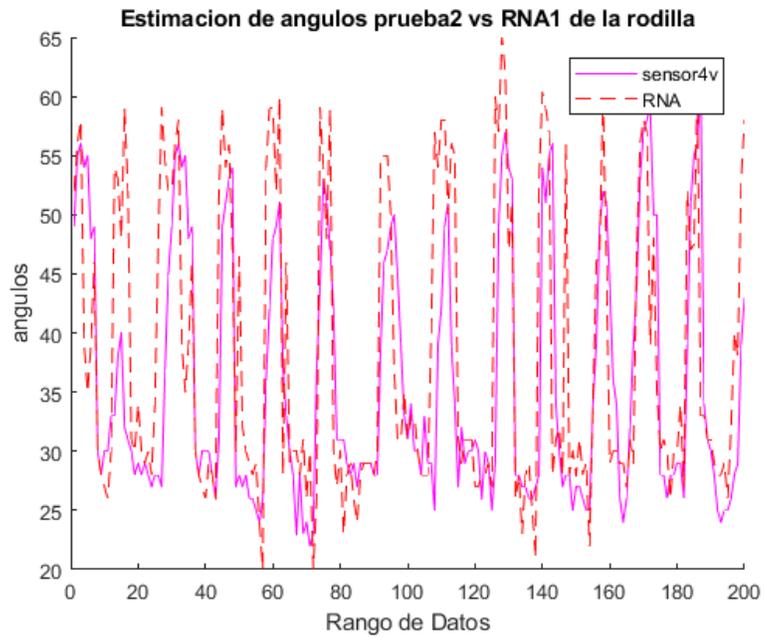


Figura 4.8: Movimiento de la rodilla durante la prueba 2 vs RNA1 [Autor].

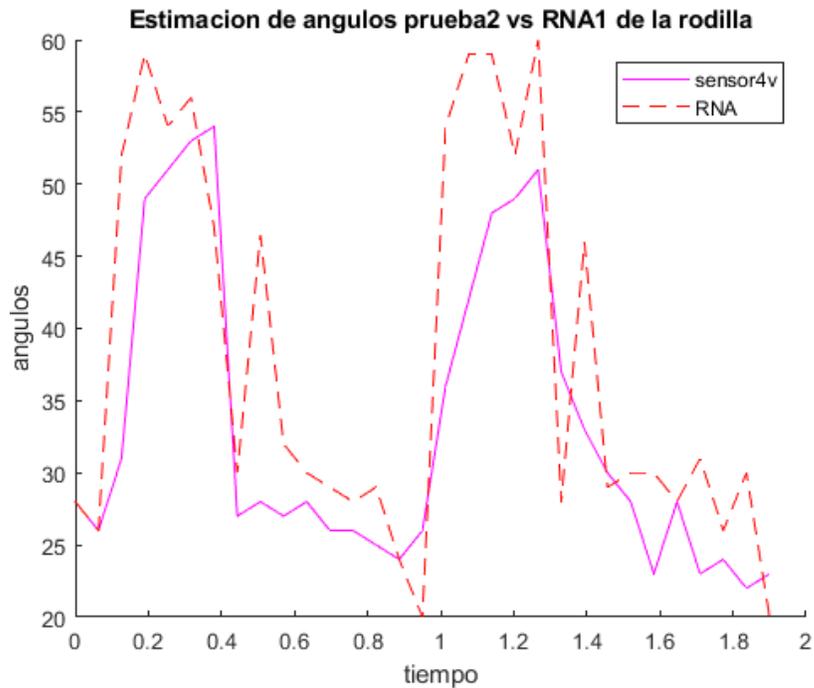


Figura 4.9: Fragmento de movimiento de la rodilla durante la prueba 2 vs RNA1 [Autor].

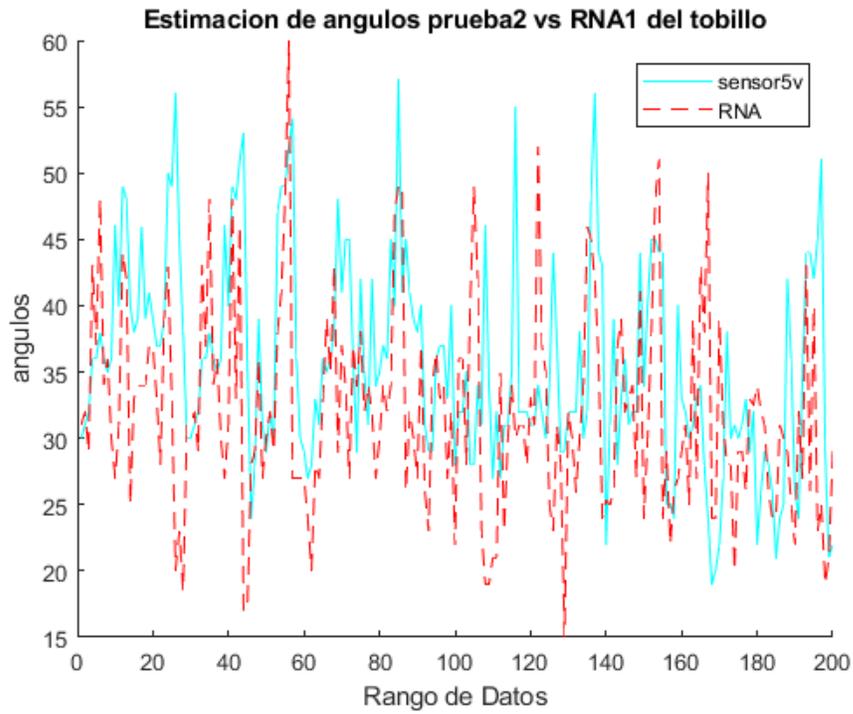


Figura 4.10: Movimiento del tobillo durante la prueba 2 vs RNA1 [Autor].

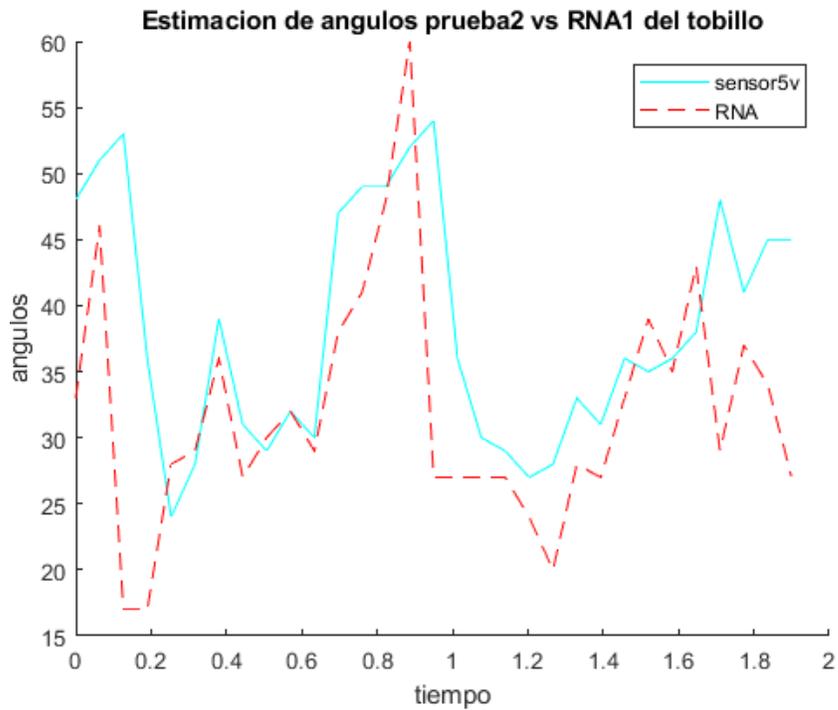


Figura 4.11: Fragmento de movimiento del tobillo durante la prueba 2 vs RNA1 [Autor].

4.4.2 Graficas de movimiento de marcha de los valores de la prueba 3 del sensor 4 y 5 vs respuesta RNA1

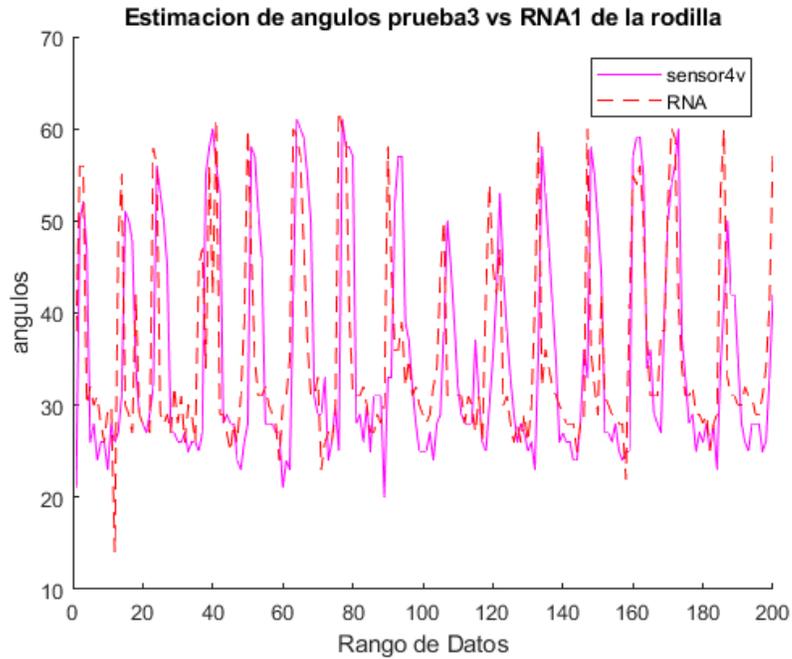


Figura 4.12: Movimiento de la rodilla durante la prueba 3 vs RNA1 [Autor].

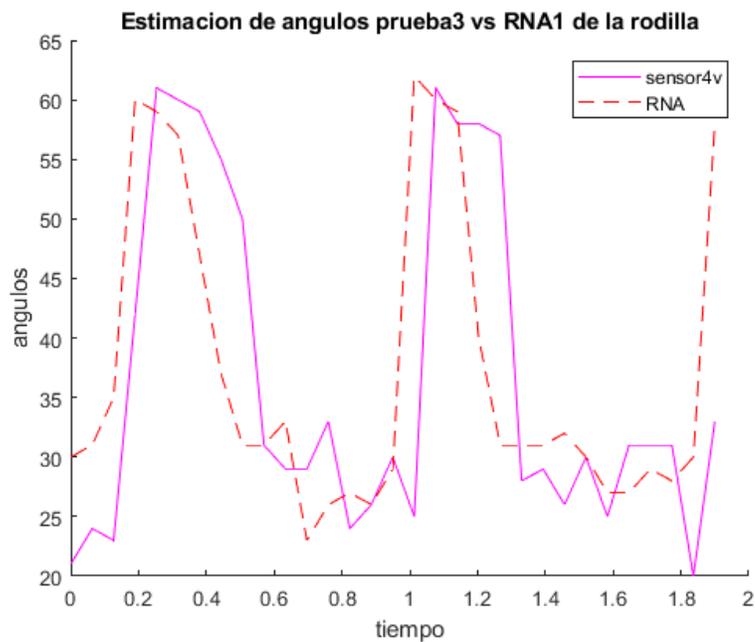


Figura 4.13: Fragmento de movimiento de la rodilla durante la prueba 3 vs RNA1. [Autor].

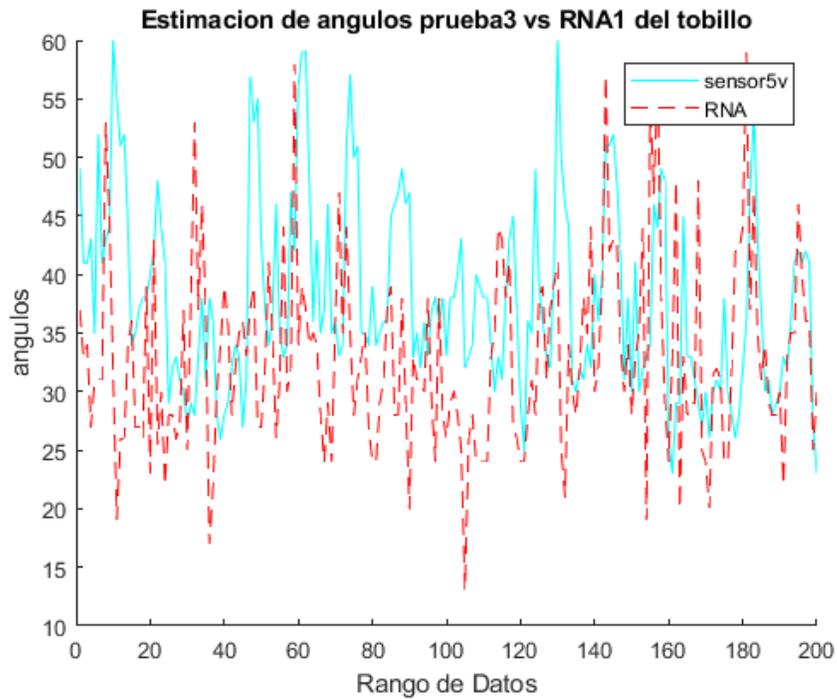


Figura 4.14: Movimiento del tobillo durante la prueba 3 vs RNA1 [Autor].

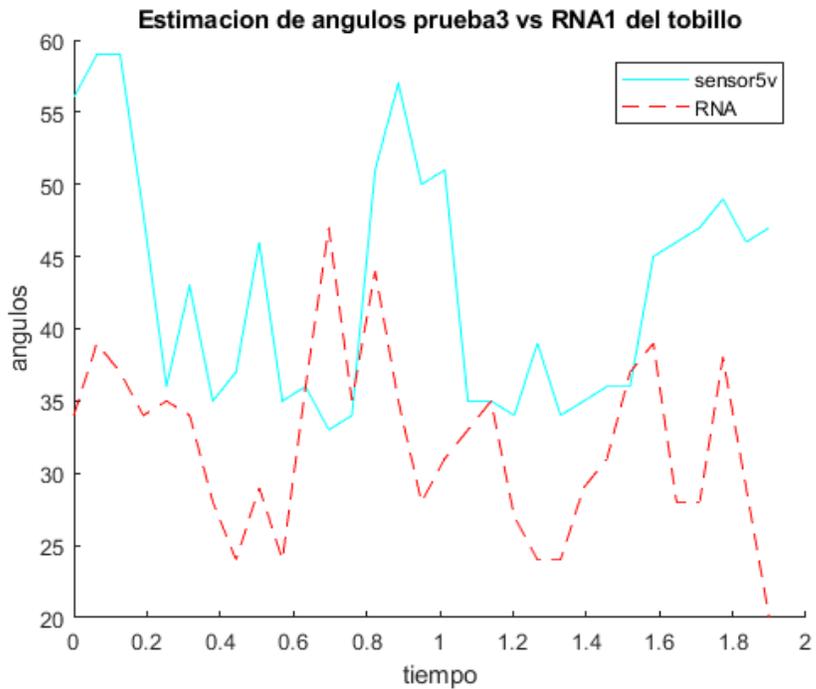


Figura 4.15: Fragmento de movimiento del tobillo durante la prueba 3 vs RNA1 [Autor].

4.4.3 Graficas de movimiento de marcha de los valores de la prueba 4 del sensor 4 y 5 vs respuesta RNA1

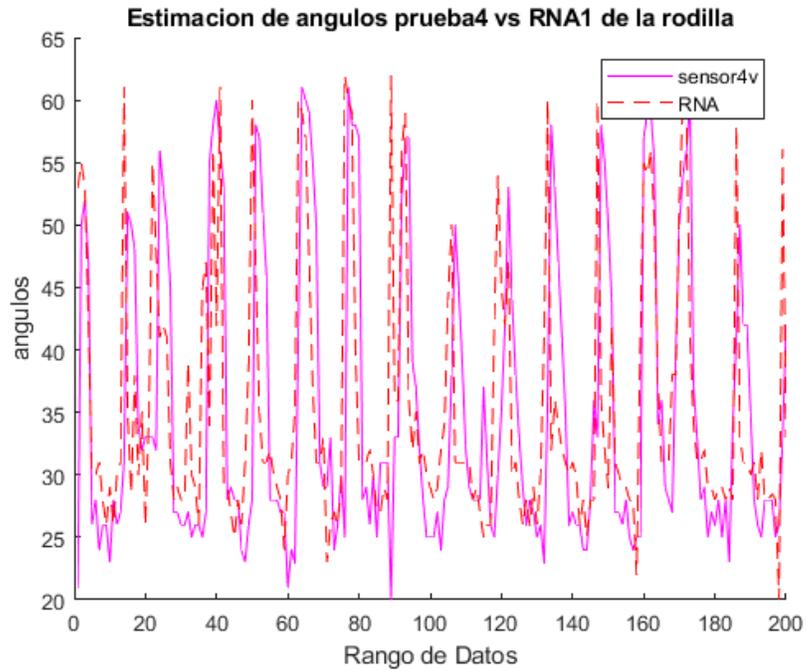


Figura 4.16: Movimiento de la rodilla durante la prueba 4 vs RNA1 [Autor].

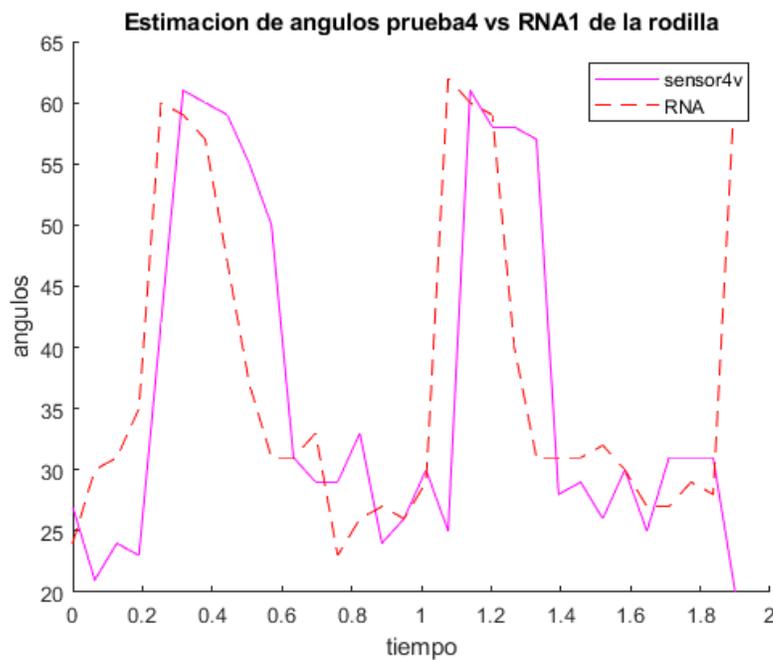


Figura 4.17: Fragmento de movimiento de la rodilla durante la prueba 4 vs RNA1. [Autor].

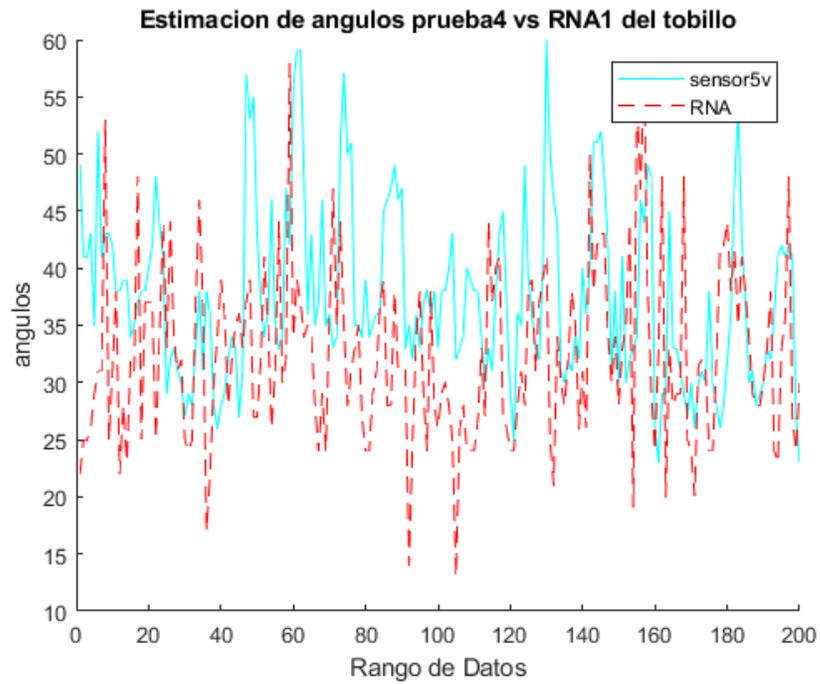


Figura 4.18: Movimiento del tobillo durante la prueba 4 vs RNA1 [Autor].

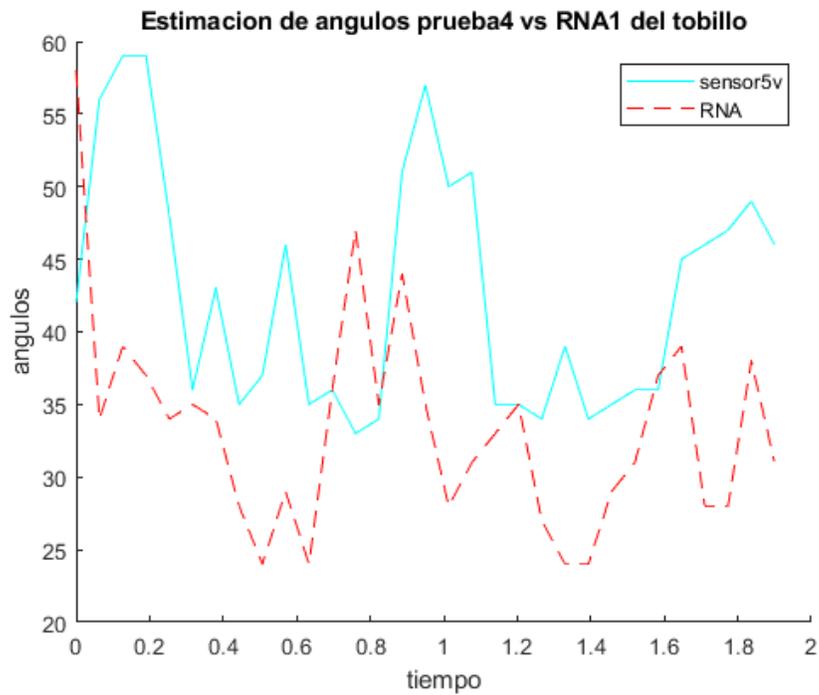
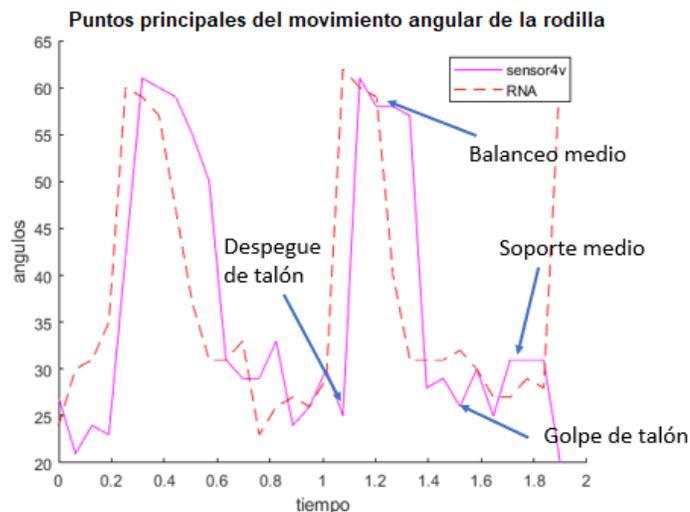
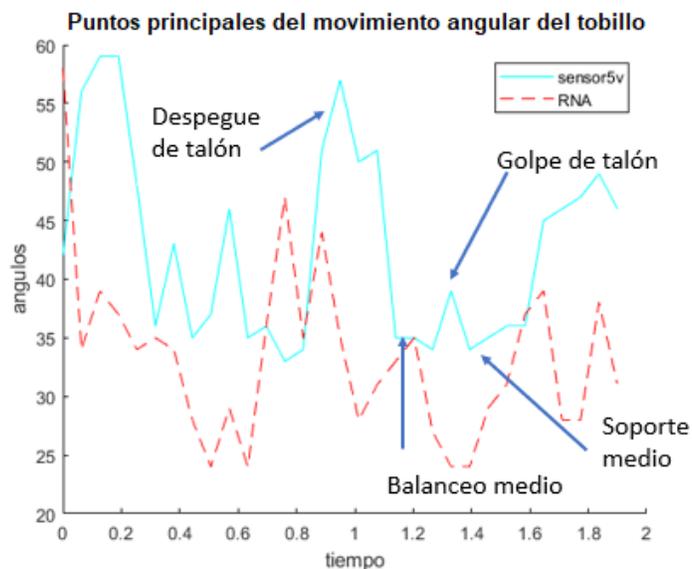


Figura 4.19: Fragmento de movimiento del tobillo durante la prueba 4 vs RNA1 [Autor].

En estas graficas anteriores de comparación, se observa que las variaciones angulares son visibles cerca a los picos y valles de la curva, el cual se debe a los lapsos que los sensores están sometidos a movimientos ligeros y a la fricción en las articulaciones, de esta forma aumentan la viabilidad de los datos. En el caso de la rodilla que es el sensor 4, las variaciones angulares son más apreciables que el sensor 5; ya que genera una mayor movilidad que en las demás articulaciones dado que hace que el sensor tenga mayor flexión al momento de desplazarse; permitiendo así identificar cuando inician y finalizan los periodos.



**Figura 4.20:** Puntos principales del movimiento angular de la rodilla [Autor].



**Figura 4.21:** Puntos principales del movimiento angular del tobillo [Autor].

En la figura 4.20 y 4.21, se puede observar los puntos principales de la fase del ciclo de marcha. Por lo tanto, se identifica que un balanceo medio, habrá un punto de despegue de talón y después un golpe de talón y por último el soporte medio. Cuando se determina estos puntos en la pierna, es posible identificar parámetros como: cadencia, balanceo y de paso. Esto facilita identificar cuando inician y finalizan los periodos de soporte y de balanceo en un ciclo [48].

### 4.5 Pruebas con otra Red Neuronal

Se realizó ensayos con los datos de las pruebas obtenidas con unos de los métodos de Validación cruzada. El método K iteraciones, toma los datos de muestra en que luego se dividen en k subconjuntos. Unos de los subconjuntos se utilizan como datos de prueba y el resto como dato de entrenamiento (ver figura 4.17).

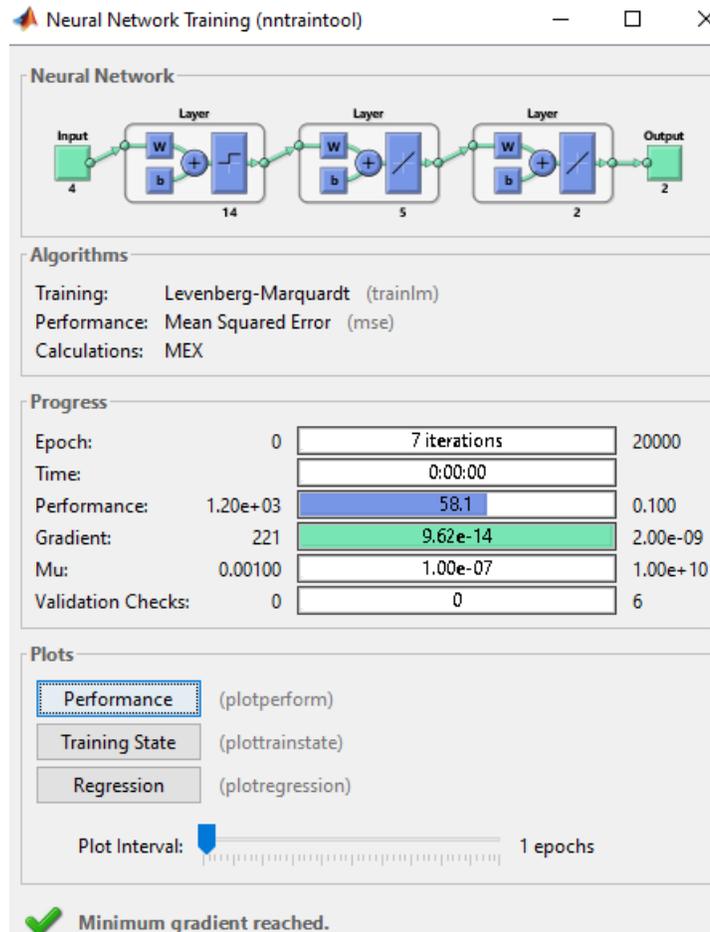
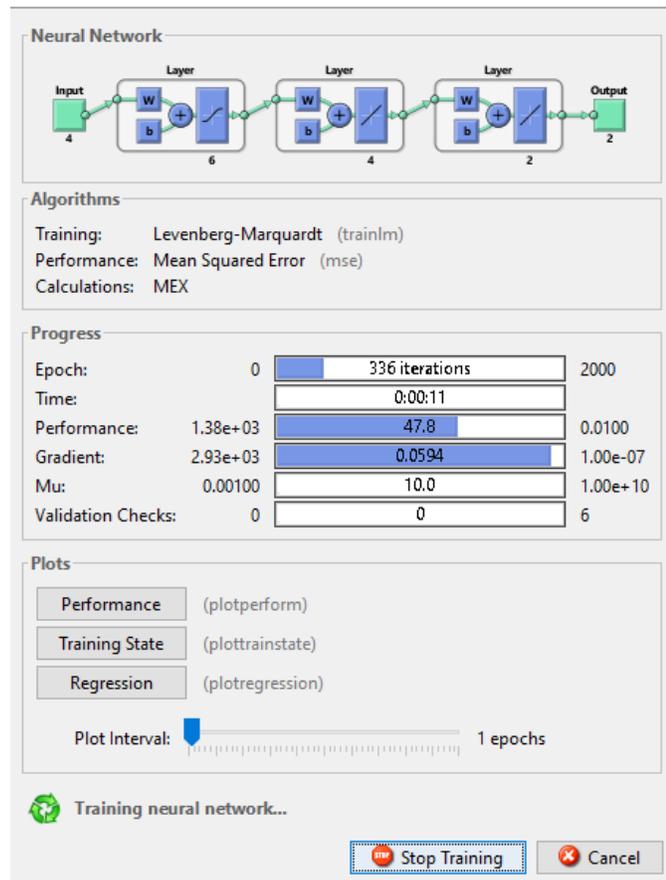


Figura 4.22: GUI de entrenamiento de redes neuronales k iteraciones.

Este método puede ser preciso puesto que se evalúa a partir de  $k$  combinaciones de los datos de entrenamiento y de prueba, pero aun así tiene una desventaja, es lento desde el punto de vista computacional. Dato que al momento se demora en copilar y lo otro hay que hacer infinidad de pruebas en terminar que tipo de función, las capas que se debe efectuar la red neuronal y entre otras cosas.

Otro método está el de validación cruzada dejando uno afuera -Leave-one-out cross-validation (LOOCV). Su funcionalidad implica separar los datos de forma que para cada iteración se obtenga una sola muestra para los datos de prueba y el resto se conforma como datos de entrenamiento. (ver figura 4.18)



**Figura 4.23:** GUI de entrenamiento de redes neuronales LOOCV.

Este método a nivel computacional se toma demasiado tiempo, dado que tiene que realizar un elevado número de iteraciones, tanto números de muestra y para cada una analizar los datos tanto entrenamiento como prueba.

Dado a lo anterior, estos dos tipos de métodos puede ser servir para cierto tipos datos, ya que al momento de realizar la pruebas se tomó demasiado infinidad de tiempo para efectuar la red de entrenamiento llegando así en descartarlos.

4.6 Análisis Estadísticos

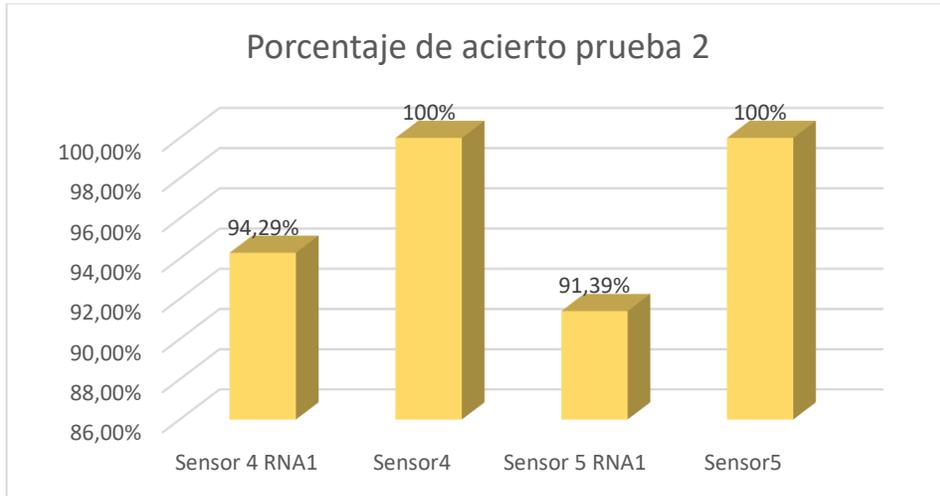


Figura 4.24: Diagrama de barras del porcentaje de acierto prueba 2 de la RNA1 vs los valores sensores 4 y 5 [Autor].

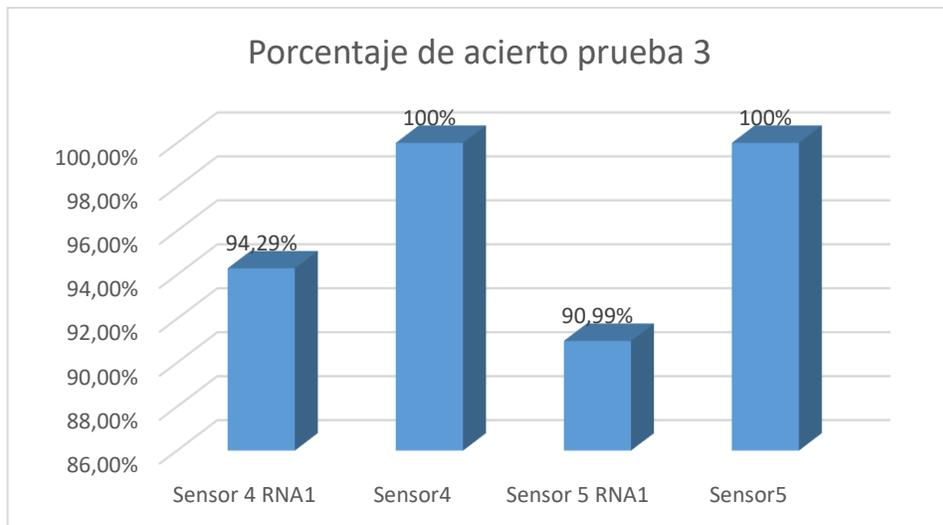
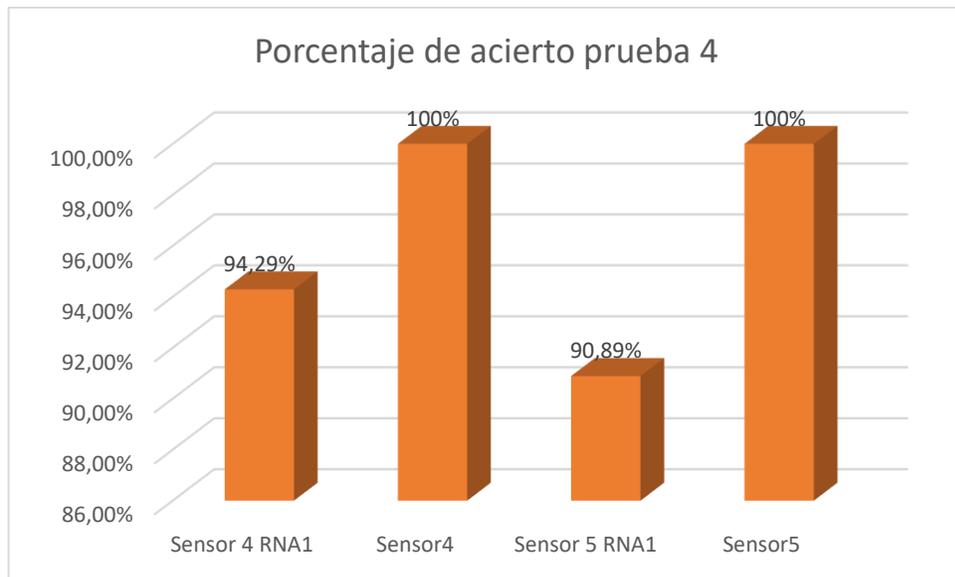


Figura 4.25: Diagrama de barras del porcentaje de acierto prueba 3 de la RNA1 vs los valores sensores 4 y 5 [Autor].



**Figura 4.26:** Diagrama de barras del porcentaje de acierto Prueba 4 de la RNA1 vs los valores sensores 4 y 5 [Autor].

A partir de la información de obtenida de los datos las articulaciones de los sensores 4, 5 y de la RNA1, se presenta los indicadores estadísticos principales como la media, la mediana, la moda, el error medio absoluto y error medio relativo porcentual. Esto demuestra que tiene un mejor desempeño los índices es la prueba 3. Por último, en los datos de las tablas se observa que la moda tiene los mismos índices para las pruebas 3 y 4, dado que es el número que más se repite en que los movimientos de las articulaciones por lo que siempre pasan en una posición constantemente.

**Tabla 4.1:** Índices estadísticos de la prueba 2

Índice estadístico	Sensor 4	Sensor 5	RNA1(Sensor 4)	RNA1(Sensor 5)
Media	38.2205	29.4835	36.5428	32.1078
Mediana	34	28	31	31
Moda	31	25	28	26

## Capítulo 4

---

Articulación	Error medio absoluto (°)	Error medio Relativo
Rodilla (Sensor 4)	1.6777	4.3895
Tobillo (Sensor 5)	2.6243	8.90

**Tabla 4.2:** Índices estadísticos de la prueba 3

Índice estadístico	Sensor 4	Sensor 5	RNA1(Sensor 4)	RNA1(Sensor 5)
Media	35.2430	31.1960	36.0762	32.3178
Mediana	30	30	31	31
Moda	28	26	28	31

Articulación	Error medio absoluto (°)	Error medio Relativo
Rodilla (Sensor 4)	0.8332	2.3641
Tobillo (Sensor 5)	1.1218	3.4711

**Tabla 4.3:** Índices estadísticos de la prueba 4

Índice estadístico	Sensor 4	Sensor 5	RNA1(Sensor 4)	RNA1(Sensor 5)
Media	35.2440	31.1665	36.0969	32.2530
Mediana	30	30	31	31
Moda	28	26	28	31

Articulación	Error medio absoluto (°)	Error medio Relativo
Rodilla (Sensor 4)	0.8529	2.4199
Tobillo (Sensor 5)	1.0865	3.4861

# 5. Conclusiones y recomendaciones

## 5.1 Conclusiones

- En resumen, la tarjeta NodeMcu Esp32, es ampliamente compatible, no solo por la fabricación del modelo, sino también dentro los círculos profesionales y aficionados. Puede usarle en varios programas IDE que en este caso el Arduino, ya que logra facilitar una mejor interacción en desarrollar múltiples proyectos.
- Se determinó que el módulo tiene una gran ventaja en su procesador de doble núcleo de 240MHz, Dado que posee un segundo núcleo, donde se dedica exclusivamente para trabajar eventos de Wifi que viene por defecto. Otra ventaja es que tiene la posibilidad de utilizar más sensores de lecturas analógicas sin la necesidad de utilizar una compuerta multiplexor.
- Al utilizar el protocolo UDP es más ligero y su flujo es unidireccional que TCP/IP, ya que es indispensable en ese protocolo es en la rapidez con la que puede viajar no tanto por la seguridad. Además, no utiliza tantas capas como el TCP/IP, en lo convierte más lento por lo que utiliza más proceso, pero aun así gestiona bastante seguridad. También el TCP/IP llega a limitar en tal punto que solo puede enviar cierta cantidad de paquetes.
- Frente a evidencias recaudadas, las redes neuronales de regresión generalizada representa una gran herramienta de ayuda para modelados de relaciones complejas, como funciones no lineales. Se puede usarse para regresión, predicción y clasificación. Por otra parte, tiene una alta precisión en estimación, ya que utiliza las funciones de Gauss.
- Para terminar, la verdadera fortaleza de las redes neuronales es su capacidad para hacer frente a complicadas funciones, que realmente no tenemos un método para describir cosas como objetos detección de visión por computadora o juegos, que requieren pensamiento sofisticado más allá de lo que podemos poner fácilmente en una sola ecuación.
- De este modo, La función newgmn, tiene dos capas que son la capa básica radial y la capa lineal especial; donde unas de las ventajas es realizar un aprendizaje de un solo paso, para que no se requiera propagación hacia atrás. Puede manejar ruidos de las entradas P.
- Se analizo que, si el spread es pequeño, la función de la base radial va ser muy pronunciada; dicha manera la neurona del vector del peso más cercano a la entrada obtendrá una salida mucho mayor que otras neuronas. De otra manera, si spread se hace más grande, la pendiente de la función de base radial se suaviza, da el paso de que varias neuronas puedan responder a un vector de entrada.

## **Conclusiones**

---

- Para concluir el spread de la función de base radial, al momento utilizar valores mayores que 1, el porcentaje de acierto al momento de entrenar la red son bajas. Dado que se llegó a tomar valores de menores que 1, pero sin llegar a valores negativos, tal motivo la base radial trabaja siempre con valores positivos.

- En los resultados obtenidos de la RNA1 se llegó a tener un porcentaje de acierto mayor a 90%. Teniendo en cuenta a un margen de error permitido de 15 grados, esto nos demuestra que la función de esta red es eficiente y ligera ya que contiene una función de base radial suavizado.

-la validación cruzada es principalmente adecuada para fines exploratorios, donde su objetivo principal es la predicción y se requiere estimar la precisión de un modelo que se llevara a cabo a la práctica. Pero su principal desventaja de este método es su costo computacional dato que toma mucho tiempo en efectuar el obtener los resultados.

### **5.2 Recomendaciones**

- Para el siguiente paso de esta investigación puede ser el estudio y desarrollo de un traje con más sensores como de fuerza, que permitirá identificar rápidamente los numero de pasos y que fuerza de fricción está ejerciendo; otro sería el sensor de inercia, para medir la velocidad angular del eje de cada extremidad, donde se podría identificar los periodos de soporte y balanceo de la marcha.

- Para posteriores trabajos que incluyan un diseño y construcción de un circuito con componentes integrados, para los sensores Flex, esto serviría para reducir la amplificación de la señal del sensor que envía a la tarjeta de desarrollo, el cual ayudaría obtener tal vez una mejor toma de datos.

### 6. REFERENCIAS BIBLIOGRÁFICAS

- [1]. Collado-Vázquez S. Análisis de la marcha con plataformas dinamométricas. Influencia del transporte de carga. [Tesis Doctoral]. Madrid: Facultad de Medicina de la Universidad Complutense; 2002.
- [2]. Real Academia de Ingeniería, Control Cinemático, Fecha de Acceso:19/03/2020, <http://diccionario.raing.es/es/lema/control-cinem%C3%A1tico>.
- [3]. López Chicharro J, Fernández Vaquero A. Fisiología del ejercicio. Editorial Médica Panamericana.
- [4]. Lasserson Daniel; Gabriel Carolyn; Sharrack Basil. Lo esencial en sistema nervioso y sentidos especiales. Harcourt Brace, Madrid 1998.
- [5]. protésica, ¿Que es una prótesis ortopédica?, Fecha de Acceso:25/03/2020, <http://protesica.com.co/ajuste-de-protesis-ortopedica/>.
- [6]. ceautomatica, Sistemas de Control Inteligente, Fecha de Acceso:25/03/2020, <https://intranet.ceautomatica.es/og/control-inteligente/es/sistemas-de-control-inteligente>.
- [7]. A. Villa, E. Gutiérrez, J. Pérez, “consideraciones para el análisis de la marcha humana. técnicas de videogrametría, electromiografía y dinamometría”, Rev. ing. biomed, vol. 2 no.3 Medellín, abril, 2008, pp.1909-9762.
- [8]. J. Salgado, D. Perdomo “Diseño e Implementación de un Sistema de Videogrametría Digital para Análisis de Saltabilidad en Deportistas”, Revista Ingeniería y Región No. 7 año 2010.
- [9]. S. Collado, “Análisis De La Marcha Humana Con Plataformas Dinamométricas. Influencia Del Transporte De Carga.”, Tesis de Doctorado, Departamento de Medicina Física y de Rehabilitación. Hidrología Médica, Universidad Complutense De Madrid, Madrid, España, 2002.
- [10]. J.P. Paul, History and fundamentals of gait analysis, Biomed Mater Eng, 8 (1998), pp. 123-135.
- [11]. S. Collado-Vásquez, La marcha: historia de los procedimientos de análisis. Biociencias (Revista de la Facultad de Ciencias de la Salud),2 (2004), pp. 1-15.
- [12]. D. Mariana Haro, “Laboratorio de análisis de marcha y movimiento”, Revista Médica Clínica Las Condes, Volumen 25, Issue 2, March 2014, Pages 237-247.
- [13]. C. L. Vaughan, B. L. Davis, L. Christopher, and J. C. O. Connor, Dynamics of

## Referencias Bibliografía

---

human gait, vol. 26, no. 3. Human Kinetics Publishers Champaign, Illinois, 2005.

[14]. J. Perry, "Gait Analysis: Normal and Pathological Function," J. Pediatr. Orthop., vol. 12, p. 815, 1992.

[15]. Spectra symbol, Sensores flexibles resistivos, Fecha de Acceso: 19/09/2019, <https://www.spectrasymbol.com/product/flex-sensors/>

[16]. electronicaestudio, ¿qué es un microcontrolador? Fecha de Acceso:19/09/2019, <https://www.electronicaestudio.com/que-es-un-microcontrolador/>.

[17]. tdrobotica, Nano 3, Fecha de Acceso:19/09/2019, [http://tdrobotica.co/nano-v30/707.html?search\\_query=arduino+nano&results=6](http://tdrobotica.co/nano-v30/707.html?search_query=arduino+nano&results=6).

[18]. tdrobotica, Módulo Wifi - ESP8266, Fecha de Acceso:19/09/2019, <http://tdrobotica.co/modulo-wifi-esp8266/350.html>.

[19]. demosspro, Tarjeta de desarrollo ESP 32 Wifi + Bluetooth, Fecha de Acceso:20/10/2019, <https://www.demosspro.com/iot-internet-of-things/esp32-wifi-bluetooth>.

[20]. geekfactory.mx, Modelo Wemos D1 ESP8266, Fecha de Acceso:20/10/2019, <https://www.geekfactory.mx/tienda/compatibles-con-arduino/wemos-d1-esp8266/>.

[21]. tdrobotica, NodeMcu Lua WIFI, Fecha de Acceso:20/10/2019, [http://tdrobotica.co/nodemculuawifi/647.html?search\\_query=nodemcu+esp8266&results=2html](http://tdrobotica.co/nodemculuawifi/647.html?search_query=nodemcu+esp8266&results=2html).

[22]. Sparkfun, SparkFun ESP32 Thing, Fecha de Acceso:20/10/2019, <https://www.sparkfun.com/products/13907>.

[23]. Organización Mundial de la Salud, «Resumen: Informe mundial sobre la discapacidad,» Malta, 2011.

[24]. N. Sawake, S. Gupta, A. Ghatge, A. Khatri and K. M. Bhurchandi, "EMG-based Prosthetic Leg for Above-Knee Amputee," 2014 Texas Instruments India Educators' Conference (TIIEC), Bangalore, 2014, pp. 69-72.

[25]. C. Alvis, N. Aranguren "diseño, implementación y puesta en funcionamiento de un sistema de control de marcha humana para un exoesqueleto de miembro inferior", Tesis de Ingeniería, Departamento de Ingeniería, Universidad Militar Nueva Granada, Bogotá, Colombia, 2015.

[26]. M. J. Khan, M. R. Afzal, N. Naseer and Z. U. Koreshi, "Control system design for a prosthetic leg using series damping actuator," 2012 International Conference of Robotics and Artificial Intelligence, Rawalpindi, 2012, pp. 1-6.

## Referencias Bibliografía

---

- [27]. Roger Clarke, "Why the world wants controls over Artificial Intelligence", *Computer Law & Security Review*, Volume 35, Issue 4, 2019, Pages 423-433, ISSN 0267-3649, <https://doi.org/10.1016/j.clsr.2019.04.006>.
- [28]. N. Kumar, N. Kharkwal, R. Kohli and S. Choudhary, "Ethical aspects and future of artificial intelligence," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Noida, 2016, pp. 111-114, doi: 10.1109/ICICCS.2016.7542339.
- [29]. Luis Alberto García Fernández, Usos y aplicaciones de la inteligencia artificial, Fecha de Acceso: 19/09/2019, <https://www.uv.mx/cienciahombre/revistae/vol17num3/articulos/inteligencia/index.htm>.
- [30]. IBM, El modelo de redes neuronales, Fecha de Acceso: 19/09/2019, [https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html).
- [31]. Rivas-Asanza, Wilmer & Mazón-Olivo, Bertha & Mejía, Fernando, Capítulo 1: Generalidades de las redes neuronales artificiales", en *Redes neuronales artificiales aplicadas al reconocimiento de patrones*, 1° Edición, Machala - Ecuador: Editorial UTMACH, 2018, pp. 13-31.
- [32]. F. Wang, K. Kim, S. Wen, Y. Wang and C. Wu, "Study of gait symmetry quantification and its application to intelligent prosthetic leg development," 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Phuket, 2011, pp. 1361-1366.
- [33]. P. Kutilek and J. Hozman, "Prediction of lower extremities movement using characteristics of angle-angle diagrams and artificial intelligence," 2011 E-Health and Bioengineering Conference (EHB), Iasi, 2011, pp. 1-4.
- [34]. J. Demby's, Y. Gao and G. N. DeSouza, "A Study on Solving the Inverse Kinematics of Serial Robots using Artificial Neural Network and Fuzzy Neural Network," 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2019.8858872.
- [35]. catarina.udlap.mx, CAPÍTULO TRES: BIOMECÁNICA DE LA MARCHA HUMANA, Fecha de Acceso: 09/05/2020, [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lep/hernandez\\_s\\_f/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_s_f/capitulo3.pdf)
- [36]. A. Barrientos, C. Balaguer, L. Peñin y R. Aracil, "Fundamentos de Robótica", McGraw-Hill ,2 edición, 2007, pp. 279-280.
- [37]. L. Broche, R. Sagaro, C. Ochoa, A. Pandiha, F. Martínez, "Análisis cinemático y dinámico de las prótesis transfemorales. Implicaciones clínicas",

## Referencias Bibliografía

---

Scientific Electronic Library Online (ScieLO), Ingeniería Mecánica vol.19 no.3 La Habana sept.-dic. 2016.

[38]. B. H. Hu, E. J. Rouse and L. J. Hargrove, "Using bilateral lower limb kinematic and myoelectric signals to predict locomotor activities: A pilot study," 2017 8th International IEEE / EMBS Conference on Neural Engineering (NER), Shanghai, 2017, pp. 98-101.

[39]. M. K. Meena, K. I. K. Sherwani and N. Kumar, "Kinematic analysis and trajectory generation of six joint lower limb orthotic device," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-4.

[40]. wiki.ai-thinker.com, Nodemcu-32s Datasheet, Fecha de Acceso: 16/05/2020, [http://wiki.ai-thinker.com/\\_media/esp32/docs/nodemcu32s\\_product\\_specification.pdf](http://wiki.ai-thinker.com/_media/esp32/docs/nodemcu32s_product_specification.pdf).

[41]. circuitarte.com, ADC Conversor Analógico Digital ADS1115 16 Bits, Fecha de Acceso: 25/11/2020, <https://www.circuitarte.com/producto/adc-conversor-analogo-digital-ads1115-16-bits-arduino/>.

[42]. microcontrollerslab.com, ADC externo ADS1115 I2C con ESP32 en Arduino IDE, Fecha de Acceso: 25/11/2020, [https://microcontrollerslab.com/ads1115-external-adc-with-esp32/?fbclid=IwAR0RAEp\\_Qg4- knLf5eU0ynb7fQxNqUkA6JMkBnyxBAz0DCurwECltMJUA\\_Q](https://microcontrollerslab.com/ads1115-external-adc-with-esp32/?fbclid=IwAR0RAEp_Qg4- knLf5eU0ynb7fQxNqUkA6JMkBnyxBAz0DCurwECltMJUA_Q).

[43]. programarfacil.com, ADS1115 convertidor analógico digital ADC para Arduino y ESP8266, Fecha de Acceso: 25/11/2020, <https://programarfacil.com/blog/arduino-blog/ads1115-convertidor-analogico-digital-adc-arduino-esp8266/>.

[44]. Espressif Systems.com, Espressif//arduino-esp32, Fecha de Acceso: 23/05/2020, <https://github.com/espressif/arduino-esp32>.

[45]. E.V.A., UCI. Conferencia#5 Protocolos de la capa de transporte UDP. Teleinformática II.D. E. Comer. "Internetworking with TCP/IP. Volume I: Principle Protocols and Architecture".

[46]. www.ionos.es, UDP: ¿qué es el protocolo UDP?, Fecha de Acceso: 23/05/2020, <https://www.ionos.es/digitalguide/servidores/know-how/udp-user-datagram-protocol/>.

[45]. www.tlm.unavarra.es, Transporte UDP y TCP, Fecha de Acceso: 23/05/2020, [https://www.tlm.unavarra.es/pluginfile.php/25584/mod\\_resource/content/1/cap4-1-UDP.pdf](https://www.tlm.unavarra.es/pluginfile.php/25584/mod_resource/content/1/cap4-1-UDP.pdf).

[46]. la.mathworks.com, Generalized Regression Neural Networks, Fecha de

## Referencias Bibliografía

---

Acceso: 24/05/2020, <https://la.mathworks.com/help/deeplearning/ug/generalized-regression-neural-networks.html>.

[47]. Wasserman, P.D., *Advanced Methods in Neural Computing*, New York, Van Nostrand Reinhold, 1993, pp. 155–61.

[48]. ALVES, Mario. "Feasibility of Wearable Sensors to Determine Gait Parameters ". Tesis de Maestria.Florida. University of South Florida. Departamento de Ingeniería Mecánica, 2011.

## 7. Anexos

### 7.1 Anexo I: Código IDE de Arduino para caracterizar el sensor Flex

```
#include <Wire.h>
#include <Adafruit_ADS1015.h>
Adafruit_ADS1115 ads(0x49);
float Voltage = 0.0;
void setup()
{
  Serial.begin(115200);
  ads.begin();
}
void loop()
{
  int16_t adc0;
  adc0 = ads.readADC_SingleEnded(3);
  Voltage = (adc0 * 0.1875)/1000;
  Serial.print("AIN0: ");
  Serial.print(adc0);
  Serial.print("\tVoltage: ");
  Serial.println(Voltage, 7);
  Serial.println();
  delay(1000);
}
```

### 7.2 Anexo II: Código IDE de Arduino para Servidor UDP

```
//---- Comunicación UPD Arduino Matlab -----  
// ----- Toma de datos de los 6 Sensores Flex-----  
// Link de guías sobre el tema  
//https://forum.arduino.cc/index.php?topic=546271.0  
// https://forum.arduino.cc/index.php?topic=652513.0  
// https://forum.arduino.cc/index.php?topic=218319.0  
// librerias para comunicacion  
#include <WiFi.h>  
#include <WiFiUdp.h>  
//#include <Wire.h>  
#include <Adafruit_ADS1015.h>  
Adafruit_ADS1115 ads(0x49);  
const char* ssid = "JUANCHOS";  
const char* password = "0513LJJB";  
// Configuración de la IP estática.  
// IPAddress local_IP(192, 168, 1, 115);  
// IPAddress gateway(192, 168, 1, 1);  
// IPAddress subnet(255, 255, 255, 0);  
WiFiUDP Udp;  
//----- puerto local para escuchar  
// local port to listen on  
unsigned int localUdpPort = 4212;  
unsigned int clientUdpPort = 4210;  
//----- buffer para paquetes entrantes  
// buffer for incoming packets  
char incomingPacket[255];
```

```
//----- una cadena de respuesta para enviar de vuelta
// a reply string to send back
//char replyPacket[] = "Hi there! Got the message :-)";

void setup()
{
  analogReadResolution(12); //--- Resolucion al sensor
  Serial.begin(115200);
  ads.begin();
  delay(1000);
  Serial.println();
  // Establecimiento de la IP estática.
  // if (!WiFi.config(local_IP, gateway, subnet)) {
  //   Serial.println("Fallo en la configuración.");
  // }
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("connected IP ");
  Udp.begin(localUdpPort);
  Serial.printf("Now listening at IP %s, UDP port %d\n",
WiFi.localIP().toString().c_str(), localUdpPort);
}
```

```
void loop()
{

//- Lee el tamaño del paquete
int packetSize = Udp.parsePacket();
if (packetSize)
{
// receive incoming UDP packets
// recibir paquetes UDP entrantes
Serial.printf("Received      %d      bytes      from      %s,      port
%d\n",packetSize,Udp.remoteIP().toString().c_str(),clientUdpPort);
int len = Udp.read(incomingPacket, 255);
if (len > 0)
{
incomingPacket[len] = 0;
}
Serial.printf("UDP packet contents: %s\n", incomingPacket);
}

//--- Declaracion de pines analogicos
int16_t adc0, adc1, adc2, adc3;
adc0 = ads.readADC_SingleEnded(0);
adc1 = ads.readADC_SingleEnded(1);
adc2 = ads.readADC_SingleEnded(2);
adc3 = ads.readADC_SingleEnded(3);

//----- declaracion sel Sensores analogicos -----
// ----- Sensor 1 Cadera -----
int f1n;
```

```
//f1n= 0.0747*adc0-705.09; //Vmax - acceptable 90%
//f1n= 0.0706*adc0-659,34; //Vef - acceptable 95%
f1n= 0.0706*adc0-659.82; //Promedio - acceptable 96%
f1n=f1n-4;
if (f1n>90)
{
  f1n=90;
}
else if(f1n<0)
{
  f1n=0;
}
Serial.print("sensor_1:");
Serial.print(f1n);
Serial.print("\t");
//delay(400);
// ----- Sensor 2 Cadera -----
int f2n;
//f2n= 0.1105*adc1-1164.8; //Vmax - acceptable 93%
//f2n= 0.1046*adc1-1095.4; //Vef - acceptable 95%
f2n= 0.0998*adc1-1044.3; //Promedio - acceptable 96 %
f2n=f2n-3;
if (f2n>90)
{
  f2n=90;
}
else if(f2n<0)
```

```
{
  f2n=0;
}
Serial.print("sensor_2:");
Serial.print(f2n);
Serial.print("\t");
//delay(400);
// ----- Sensor 3 Rodilla -----
int f3;
int f3n;
  f3= analogRead(35); // sensor 3
  //f3n= 0.1769*f3-352.55; // Vmax- Aceptacion 95 %
  //f3n= 0.1718*f3-336.7; // Vef - Aceptacion 96 %
  f3n= 0.1714*f3-336.05; //Media- Aceptacion 97%
  f3n=f3n-4;
  if (f3n>90)
  {
    f3n=90;
  }
  else if(f3n<0)
  {
    f3n=0;
  }
  Serial.print("sensor_3:");
  Serial.print(f3n);
  Serial.print("\t");
  //delay(400);
```

```
// ----- Sensor 4 Rodilla-----  
  
int f4;  
int f4n;  
  
    f4= analogRead(32); // ADC4 // sensor 4  
  
    //f4n= 0.1766*f4-285.68; // Vmax- Aceptacion 60 %  
  
    //f4n= 0.1722*f4-273.04; // Vef - Aceptacion 90 % en movimiento  
  
    f4n= 0.1743*f4-277.03; //Media- Aceptacion 95% en movimiento  
  
    f4n=f4n-4;  
  
    if (f4n>90)  
    {  
        f4n=90;  
    }  
  
    else if(f4n<0)  
    {  
        f4n=0;  
    }  
  
    Serial.print("sensor_4: ");  
  
    Serial.print(f4n);  
  
    Serial.print("\t");  
  
    //delay(400);  
  
// ----- Sensor 5 Tobillo -----  
  
int f5n;  
  
    //f5n= -0.1099*adc2+1818.3; //Vmax - aceptable 90% movimiento  
  
    //f5n= -0.1092*adc2+1811.5; //Vef - aceptable 93%  
  
    f5n= -0.1093*adc2+1813.7+20; //Promedio - aceptable 95%  
  
    //f5n=f5n;  
  
    if (f5n>90)
```

```
{
  f5n=90;
}
else if(f5n<0)
{
  f5n=0;
}
Serial.print("sensor_5:");
Serial.print(f5n);
Serial.print("\t");
//delay(400);
// ----- Sensor 6 Tobillo -----
int f6n;
//f6n= 0.0997*adc3-1036,1; //Vmax - acceptable 93%
//f6n= 0.0979*adc3-1012,8; //Vef - acceptable 95%
f6n= 0.0979*adc3-1012.2; //Promedio - acceptable 96%
f6n=f6n-15;
if (f6n>90)
{
  f6n=90;
}
else if(f6n<0)
{
  f6n=0;
}
Serial.print("sensor_6:");
Serial.println(f6n);
```

```
Serial.print("\t");
delay(10);

//-----

//---- enviar una respuesta a la dirección IP y al puerto de donde obtuvimos el
paquete
// send back a reply, to the IP address and port we got the packet from
    Udp.beginPacket(Udp.remoteIP().toString().c_str(),clientUdpPort);

// Salidas de la lecturas hacia el servidor Matlab
//for(int i=0; i<400; i++){
    Udp.write(f1n);
    Udp.write(f2n);
    Udp.write(f3n);
    Udp.write(f4n);
    Udp.write(f5n);
    Udp.write(f6n);
//}
    Udp.endPacket();
}
```

### 7.3 Anexo III: Código en Matlab para Cliente UDP

```
% Direccion IP 192.168.0.104, Puerto UDP 4212
%function [m] = estado_sensores( )
% envia un mensaje a la plataforma de la tarjeta
% JUDP ('ENVIAR', PUERTO, ANFITRIÓN, MENSAJE)
judp('SEND',4212,'192.168.0.105',int8('.'));
% vacia la variable de recepcion
msg = [];
disp('Esperando datos de los sensores')
disp(' s1 s2 s3 s4 s5 s6 ')
% activa ciclo para la recepción de mensajes
```

```
while (isempty(msg))
    B = zeros(6,2000);
    D = [0 0 0 0 0 0];
    tic;
    for j=1:2000
        %activa ciclo para la recepción de mensajes
        % MSSG = JUDP ('RECIBIR', PUERTO, LONGITUD DE PAQUETE, TIEMPO DE ESPERA)
        [msg,sourceHost]=judp('RECEIVE',4210,100,500);
        %m = double(msg); % convierte el datagrama en el formato correcto
        %Guarda la similitud de los datos de los sensores
        D=msg';
        [xx yy] = size(D);
        if (yy<6)
            [msg,sourceHost]=judp('RECEIVE',4210,100,500);
            %B(:,j)=D;
            D=msg';
        else
            B(:,j)=D;
        end
        disp(msg')
        T=toc;
    end
end
```

### 7.4 Anexo IV: Código simulación de prueba

#### 7.4.1 función graficar

```
%q1=cadera derecha sensor 1
%q2=cadera derecha sensor 4
%q3=rodilla derecha sensor 5
%q4=rodilla izquierda sensor 2
%q5=rodilla izquierda s3
%q6=rodilla izquierda s5
function graficar(prueba1)
%load('prueba1.mat');

q1=(-prueba1(1)+19);
q2=(-prueba1(4)+16);
q3=(-prueba1(5)+13);
q4=(-prueba1(2)+15);
q5=(-prueba1(3)+13);
q6=(-prueba1(6)+64);

% close all; % cierra las ventanas abiertas
L1=44;
L2=47;
L3=15;
axis([-100,100,-100,70]);
f1=figure(1); % Crea una ventana para hacer nueva figura
```

```

set(f1, 'Color', [1, 1, 1]);
title('Prueba 1a Movimiento de Marcha'); % Crea el titulo de la figura
xlabel('x');
ylabel('y');
%----- pierna Derecha -----
p0 = [0 , 0];
p1 = [L1*sind(q1), -L1*cosd(q1)];
p2 = p1+[L2*sind(q1+q2), -L2*cosd(q1+q2)];
alfa1=q1+q2+90+q3;
p3 = p2+[L3*sind(alfa1), -L3*cosd(alfa1)];
line([p0(1) ,p1(1)], [p0(2) , p1(2)], 'Color', 'red', 'LineWidth', 3);
line([p1(1) ,p2(1)], [p1(2) , p2(2)], 'Color', 'blue', 'LineWidth', 3);
line([p2(1) ,p3(1)], [p2(2) , p3(2)], 'Color', 'black', 'LineWidth', 3);

%----- pierna Izquierda -----
p0 = [0 , 0];
p4 = [L1*sind(q4), -L1*cosd(q4)];
p5 = p4+[L2*sind(q4+q5), -L2*cosd(q4+q5)];
alfa2=q4+q5+90+q6;
p6 = p5+[L3*sind(alfa2), -L3*cosd(alfa2)];
line([p0(1) ,p4(1)], [p0(2) , p4(2)], 'Color', 'red', 'LineWidth', 3);
line([p4(1) ,p5(1)], [p4(2) , p5(2)], 'Color', 'blue', 'LineWidth', 3);
line([p5(1) ,p6(1)], [p5(2) , p6(2)], 'Color', 'green', 'LineWidth', 3);
end

```

### 7.4.2 simulación

```

load('prueba1.mat');
for j=1:1000 % dimension de cantidad de datos
    cla; %clear axes -->limpia la grafica
    %el:indica "toma los 6 datos",
    %el j, va a ser el numero de columna que va a ir incrementando
    A=prueba1(:,j);
    graficar(A);
    pause(0.2);
end

```

### 7.5 Anexo V: Red Neuronal de Regresión Generalizada

```

%---- *Diseño de red neuronal de regresión generalizada* ----

% Las redes neuronales de regresión generalizada (grnns) son un tipo de red
% de base radial que a menudo se usa para la aproximación de funciones.
% Las grnns se pueden diseñar muy rápidamente.
% y devuelve una nueva red neuronal de regresión generalizada.

% net = newgrnn (P, T, spread) toma tres entradas,
%P = Matriz R-por-Q de vectores de entrada Q.
%T = Matriz S-por-Q de vectores de clase objetivo Q.
%spread = Extensión de funciones de base radial (por defecto = 1.0).

```

```

% Para ajustar los datos muy de cerca, se usa una extensión más pequeña que
% la distancia típica entre los vectores de entrada.

% clear; % Borra las variables del workspace
% close all; % cierra las ventanas abiertas
clc; % borra la pantalla
%---- Cargar datos de los sensores obtenidos ----
load('prueba1.mat')
% -- Guardar los datos de Validacion ----
Y4=prueba1(4:4,:); % datos del sensor 4
Y5=prueba1(5:5,:); % datos del sensor 5
% -- Guardas los datos de entrenamiento -----
Y123=prueba1(1:3,:); % datos de los sensores 1,2,3
Y6=prueba1(6,:); % datos del sensor 6
%--toma los datos del sensor 1,2,3 y 6 para CONCATENAR como dato de entrada
% ----- Datos de entrenamiento -----
S1236=[Y123;Y6]; % CONCATENAR como dato de entrada
S1236iim1=[S1236(:,2:length(Y4)); S1236(:,1:length(Y4)-1)];%concatenar

% ----- Datos de Saalida -----
%-- toma los datos del sensor 4 Y 5 para CONCATENAR como dato de salida
S45=[Y4;Y5];
S45i=S45(:,2:length(Y4)); % reduciendo la informacion
% ----- *RNA entrenamiento para sensor 4 y 5* -----
net1 = newgrnn(S1236iim1,S45i,0.01); %crea la RNA
S45ie = sim(net1,S1236iim1); % ejecuta red neuronal
%----- calculo del error
error=S45i-S45ie; %vector del error absoluto
EmaxP=2; %Error Maximo permitido
acierto= sum(abs(error')<=EmaxP)*100/length(error);
% fprintf(1,'%s\n','porcentaje de acierto del sensor 4:',acierto(1));
% fprintf(1,'%s\n','porcentaje de acierto del sensor 5:',acierto(2));
hold on
%----- grafica de entrenamiento
f1=figure(1);
set(f1,'Color',[1,1,1]);
plot(error(1,:), 'r-'); % grafica del error del sensor 4
plot(error(2,:), 'b--'); % grafica del error del sensor 5
legend('sensor4','sensor5');
xlabel('Rango de Datos');
ylabel('Rango del error');
title('Grafica del Error de los datos de salida RNA1')
%----- validacion -----
% -----Guardas los datos de Validacion -----
load('prueba4.mat')
Z123=prueba4(1:3,:); % datos de los sensores 1,2,3
Z6=prueba4(6,:); % datos del sensor 6
Z4=prueba4(4:4,:); % datos del sensor 4
Z5=prueba4(5:5,:); % datos del sensor 5
S45v=[Z4;Z5];
S45iv=S45v(:,2:length(Z4)); % reduciendo la informacion
% --- toma los datos del sensor 1,2,3 y 6 para CONCATENAR
S1236v=[Z123;Z6];%

```

```

S1236iim1v=[S1236v(:,2:length(Z4)); S1236v(:,1:length(Z4)-1)];%concatenar

%----- prueba de la RNA
S45ive = sim(net1,S1236iim1v); %ejecuta red neuronal
%---- caclulos del error
errorv=S45iv-S45ive; %vector del error absoluto
EmaxP= 15; %Error Maxio permitido
aciertov= sum(abs(errorv')<=EmaxP)*100/length(errorv);
%----- grafica de la validacion
f2=figure(2);
set(f2,'Color',[1,1,1]);
hold on
title('Grafica del Error de los datos de esperados - datos de salida RNA1')
plot(errorv(1,:), 'y-'); % grafica del error del sensor 4
plot(errorv(2,:), 'b--'); % grafica del error del sensor 5
legend('sensor4v','sensor5v');
xlabel('Rango de Datos');
ylabel('Rango del error');
%gensim (net1, -1)
%% ---- grafica de valor deseado de la prueba 2
load('prueba4.mat')
f3=figure(3);
set(f3,'Color',[1,1,1]);
hold on
plot(prueba4(4,1:200), 'm-'); % grafica del sensor 4
xlabel('Rango de datos');
ylabel('angulos');
title('Grafica de respuesta del sensor 4')
f4=figure(4);
set(f4,'Color',[1,1,1]);
hold on
plot(prueba4(5,1:200), 'c-'); % grafica del sensor 5
xlabel('Rango de datos');
ylabel('angulos');
title('Grafica de respuesta del sensor 5')

%% ---- grafica de respuesta RNA1 de la prueba 2

f5=figure(5);
set(f5,'Color',[1,1,1]);
hold on
plot(S45ive(1,1:200), 'm-'); % grafica del sensor 4
xlabel('Rango de datos');
ylabel('angulos');
title('Grafica de respuesta de RNA1 del sensor 4')
f6=figure(6);
set(f6,'Color',[1,1,1]);
hold on
plot(S45ive(2,1:200), 'c-'); % grafica del sensor 5
xlabel('Rango de datos');
ylabel('angulos');
title('Grafica de respuesta de RNA1 del sensor 5')
%% ---- valores estadisticos prueba 2 ---

```

```
promedio4=mean(prueba4(4,:));
promedio5=mean(prueba4(5,:));
Mediana4=median(prueba4(4,:));
Mediana5=median(prueba4(5,:));
moda4=mode(prueba4(4,:));
moda5=mode(prueba4(5,:));
%% ---- valores estadísticos de valores obtenidos de RNA1
promedio4R=mean(S45ive(1,:));
promedio5R=mean(S45ive(2,:));
Mediana4R=median(S45ive(1,:));
Mediana5R=median(S45ive(2,:));
moda4R=mode(S45ive(1,:));
moda5R=mode(S45ive(2,:));
```