NOMBRE DEL TRABAJO

TESIS_DIANA-SILVA-2.pdf

AUTOR

DIANA SILVA_DOMINGUEZ

RECUENTO DE PALABRAS

18769 Words

RECUENTO DE CARACTERES

106257 Characters

RECUENTO DE PÁGINAS

87 Pages

TAMAÑO DEL ARCHIVO

2.6MB

FECHA DE ENTREGA

Feb 28, 2024 7:23 AM GMT-5

FECHA DEL INFORME

Feb 28, 2024 7:26 AM GMT-5

17% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 16% Base de datos de Internet
- 5% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref

Excluir del Reporte de Similitud

- Base de datos de trabajos entregados
- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 8 palabras)

UNTELS | UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FORMULARIO DE AUTORIZACIÓN PARA LA PUBLICACIÓN DE TRABAJOS DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL DE LA UNTELS

(Art. 45º de la ley Nº 30220 – Ley)

Autorización de la propiedad intelectual del autor para la publicación de tesis en el Repositorio Institucional de la Universidad Nacional Tecnológica de Lima Sur (https://repositorio.untels.edu.pe), de conformidad con el Decreto Legislativo N° 822, sobre la Ley de los Derechos de Autor, Ley N° 30035 del Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, Art. 10º del Rgto. Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales en las universidades – RENATI Res. N° 084-2022-SUNEDU/CD, publicado en El Peruano el 16 de agosto de 2022; y la RCO N° 061-2023-UNTELS del 01 marzo 2023.

TIPO DE TRABAJO DE INVESTIGACIÓN

1). TESIS (X) 2). TRABAJO DE SUFICIENCIA PROFESIONAL ( )

DATOS PERSONALES

Apellidos y Nombres.	SILVA DOMIGUEZ DIANA ELIZABETH ALISON
D.N.I.:	73612463
Otro Documento:	
Nacionalidad:	PERUANA
Teléfono:	952762208
e-mail:	2015100667@UNTELS.EDU.PE

DATOS ACADÉMICOS

Pregrado

Facultad:	FACULTAD DE INGENIERÍA Y GESTIÓN
Programa Académico:	TESIS
Título Profesional otorgado:	INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

Postgrado

Universidad de Procedencia:	
País:	
Grado Académico otorgado:	

Datos de trabajo de investigación

Título:	"IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO PARA LA MEDICIÓN Y ANÁLISIS DE LA MARCHA MEDIANTE ACELERÓMETROS Y GIROSCOPIOS"
Fecha de Sustentación:	05 DE DICIEMBRE DEL 2023
Calificación:	APROBADO CON UNANIMIDAD
Año de Publicación:	2024

UNTELS | UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

AUTORIZACIÓN DE PUBLICACIÓN EN VERSIÓN ELECTRÓNICA

A través de la presente, autorizo la publicación del texto completo de la tesis, en el Repositorio Institucional de la UNTELS especificando los siguientes términos:

Marcar con una X su elección.

1) Usted otorga una licencia especial para publicación de obras en el REPOSITORIO INSTITUCIONAL DE LA UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR.

Si autorizo X No autorizo _____

2) Usted autoriza para que la obra sea puesta a disposición del público conservando los derechos de autor y para ello se elige el siguiente tipo de acceso.

Derechos de autor		
TIPO DE ACCESO	ATRIBUCIONES DE ACCESO	ELECCIÓN
ACCESO ABIERTO 12.1(*)	**info:eu-repo/semantics/openAccess** (Para documentos en acceso abierto)	(X)

3) Si usted dispone de una **PATENTE** puede elegir el tipo de **ACCESO RESTRINGIDO** como derecho de autor y en el marco de confiabilidad dispuesto por los numerales 5.2 y 6.7 de la directiva Nº 004-2016-CONCYTEC DEGC que regula el Repositorio Nacional Digital de CONCYTEC (Se colgará únicamente datos del autor y el resumen del trabajo de investigación).

Derechos de autor		
TIPO DE ACCESO	ATRIBUCIONES DE ACCESO	ELECCIÓN
ACCESO RESTRINGIDO	**info:eu-repo/semantics/restrictedAccess** (Para documentos restringidos)	( )
	info:eu-repo/semantics/embargoedAccess (Para documentos con períodos de embargo. Se debe especificar las fechas de embargo)	( )
	info:eu-repo/semantics/closedAccess (para documentos confidenciales)	( )

(*) http://renati.sunedu.gob.pe

UNTELS | UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

Rellene la siguiente información si su trabajo de investigación es de acceso restringido:

Atribuciones de acceso restringido:

Motivos de la elección del acceso restringido:

SILVA DOMINGUEZ, DIANA ELIZABETH ALISON

APELLIDOS Y NOMBRES

73612463

DNI

Firma y huella:

Lima, 01 de MARZO del 2024

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FACULTAD DE INGENIERÍA Y GESTIÓN

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

“IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO PARA LA MEDICIÓN Y ANÁLISIS DE LA MARCHA MEDIANTE ACELERÓMETROS Y GIROSCOPIOS”

TESIS

Para optar el Título Profesional de

INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

PRESENTADO POR EL BACHILLER

SILVA DOMINGUEZ, DIANA ELIZABETH ALISON
ORCID: 0009-0004-5415-8126

ASESOR

LEZAMA CALVO, JINMI GREGORY
ORCID: 0000-0003-2906-9741

Villa El Salvador
2023

UNIVERSIDAD NACIONAL
TECNOLÓGICA DE LIMA SUR

"Año de la unidad, la paz y el desarrollo"

DECANATO DE LA FACULTAD DE INGENIERÍA Y GESTIÓN

ACTA DE SUSTENTACIÓN DE TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

En Villa El Salvador, siendo las 10 horas del día 5 de diciembre de 2023,, en la Facultad de Ingeniería y Gestión, los miembros del Jurado Evaluador, integrado por:

PRESIDENTE: DR. MARK DONNY CLEMENTE ARENAS	DNI N° 41962207	C.I.P. N° 181400
SECRETARIO: MG. PABLO ANDRÉS VILLEGAS CHUNGA	DNI N° 09694556	C.I.P. N° 199274
VOCAL : MG. EDGARD OPORTO DÍAZ	DNI N° 09352077	C.I.P. N °106881

ASESOR : DR. JINMI GREGORY LEZAMA CALVO DNI N° 42294872 C.I.P N° 97712

Designados mediante Resolución de Decanato N° 337-2023-UNTELS-R-D de fecha 15 de agosto de 2023 quienes dan inicio a la Sesión Pública de Sustentación y Evaluación de Tesis.

Acto seguido, el (la) aspirante al : Grado de Bachiller ☐ Título Profesional [X]

Doña: DIANA ELIZABETH ALISON SILVA DOMINGUEZ identificada con D.N.I. N° 73612463 procedió a la Sustentación de:

Trabajo de investigación ☐ Tesis [X] Trabajo de suficiencia ☐ Artículo científico ☐

Titulada: "IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO PARA LA MEDICIÓN Y ANÁLISIS DE LA MARCHA MEDIANTE ACELERÓMETROS Y GIROSCOPIOS".

Aprobado mediante Resolución de Decanato N° 964-2023-UNTELS-R-D de fecha 27 de noviembre de 2023, de conformidad con las disposiciones del Reglamento General de Grados Académicos y Títulos Profesionales vigentes, sustentó y absolvió las interrogantes que le formularon los señores miembros del Jurado Evaluador.

Concluida la Sustentación se procedió a la evaluación y calificación correspondiente, resultando la aspirante APROBADA por unanimidad con la nota de: trece (letras) 13 (números), de acuerdo al Art. 65° del Reglamento General para optar el Título Profesional.

CALIFICACIÓN		CONDICIÓN	EQUIVALENCIA
NÚMERO	LETRAS		
13	trece	Aprobado por unanimidad	Bueno

Siendo las ..... horas del día 5 de diciembre de 2023, se dio por concluido el acto de sustentación, firmando el jurado evaluador el Acta de Sustentación, que obra en el Decanato de la Facultad de Ingeniería y Gestión.

DR. MARK DONNY CLEMENTE ARENAS
PRESIDENTE

MG. PABLO ANDRÉS VILLEGAS CHUNGA
SECRETARIO

MG. EDGARD OPORTO DÍAZ
VOCAL

DIANA ELIZABETH ALISON SILVA DOMINGUEZ
BACHILLER

DEDICATORIA

Este proyecto de investigación está dedicado a mis padres, quienes me ofrecieron su apoyo incondicional en todo momento.

A mi madre, Ofelia Elena, por transmitirme su valentía.

A mi padre, Jesús Ángel, que ha sido mi guía acertada en el camino.

AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a todas las personas que han contribuido de manera significativa a la culminación de este trabajo de tesis. En primer lugar, agradezco a mis padres por brindarme la oportunidad de estudiar y por la confianza depositada en mí, así como a mi hermano por su constante apoyo.

Al asesor, el Dr. Jinmi Gregory Lezama Calvo, le agradezco por su orientación, dedicación y paciencia, especialmente por el valioso tiempo que dedicó a lo largo de todo el proceso de investigación. Sus consejos y apoyo han sido fundamentales para el desarrollo de este trabajo.

Además, extiendo mi agradecimiento a los revisores, el Mg. Edgar Oporto Díaz y el Mg. Pablo Villegas Chunga, por sus detalladas revisiones y sugerencias que han mejorado considerablemente la calidad de este trabajo. Sus aportes han sido esenciales para culminar la investigación.

No puedo pasar por alto expresar mi gratitud a todos los docentes de la escuela profesional de Ingeniería Electrónica y Telecomunicaciones de la Universidad Nacional Tecnológica de Lima Sur por compartir sus conocimientos durante mi vida universitaria.

Finalmente, agradezco a mi familia por su constante apoyo, comprensión y aliento a lo largo de este arduo pero gratificante proceso. Su respaldo ha sido fundamental en cada paso de este camino académico.

Este trabajo no habría sido posible sin la colaboración y apoyo de todos ustedes. Gracias por ser parte fundamental de este logro académico.

RESUMEN

El trabajo de investigación titulado "Implementación de un Sistema Electrónico para la Medición y Análisis de la Marcha mediante Acelerómetros y Giroscopios" se centra en el desarrollo de un sistema para estimar los movimientos de la marcha en el cuerpo humano. El objetivo principal es abordar la escasez de sistemas electrónicos especializados para el monitoreo de la marcha en personas con problemas de movilidad. La investigación se enmarca en un enfoque cuantitativo.

Para alcanzar este objetivo, se seleccionaron cuidadosamente los materiales necesarios, incluyendo dos sensores inerciales (acelerómetros y giroscopios) y un microcontrolador. Estos componentes forman la base del sistema electrónico diseñado para medir y analizar la marcha. Las características de aceleración y velocidad angular obtenidos de los sensores son extraídas para un análisis posterior mediante una red neuronal.

La implementación de un modelo entrenado a lo largo de 120 épocas con una tasa de validación del 10%, los resultados revelan una pérdida de 0.0829 y una precisión del 97.14%, cercana al 100%. Estos valores destacan la eficacia del proceso de entrenamiento, indicando un rendimiento óptimo del modelo.

Palabras clave: sensor inercial, red neuronal, acelerómetro, giroscopio, marcha humana.

ABSTRACT

The research work titled "Implementation of an Electronic System for the Measurement and Analysis of Gait using Accelerometers and Gyroscopes" focuses on the development of a system to estimate gait movements in the human body. The main objective is to address the shortage of specialized electronic systems for monitoring gait in people with mobility issues. The research is framed within a quantitative approach.

To achieve this objective, the necessary materials were carefully selected, including two inertial sensors (accelerometers and gyroscopes) and a microcontroller. These components form the basis of the electronic system designed to measure and analyze gait. The acceleration and angular velocity characteristics obtained from the sensors are extracted for further analysis using a neural network.

The implementation of a model trained over 120 epochs with a validation rate of 10% reveals results showing a loss of 0.0829 and an accuracy of 97.14%, close to 100%. These values highlight the effectiveness of the training process, indicating optimal model performance.

Keywords: inertial sensor, neural network, accelerometer, gyroscope, human gait.

ÍNDICE

LISTADO DE FIGURAS

LISTADO DE TABLAS

INTRODUCCIÓN

El presente trabajo de investigación, titulado "Implementación de un Sistema Electrónico para la Medición y Análisis de la Marcha mediante Acelerómetros y Giroscopios", se centra en abordar la carencia de sistemas electrónicos especializados basados en sensores para la recopilación de datos específicamente en la región de la pierna. Los escasos dispositivos representan un obstáculo para la efectiva supervisión de la marcha en individuos con dificultades de movilidad, resaltando así la urgencia de desarrollar soluciones innovadoras. Estas soluciones buscan permitir una evaluación precisa y la detección de anomalías en el movimiento, con el propósito de mejorar la calidad de vida de aquellos que enfrentan limitaciones en su movilidad.

Los objetivos específicos delinean un plan de acción para abordar la problemática. Se propone analizar y definir los parámetros fundamentales para medir el movimiento en personas adultas, profundizando en aspectos clave relacionados con el desplazamiento y la postura durante la marcha. Además, se pretende desarrollar un sistema electrónico embebido basado en sensores inerciales y microcontroladores, que garantice la adquisición precisa de datos de movimiento durante la marcha en adultos, ofreciendo una solución eficiente. Asimismo, se busca crear un algoritmo basado en redes neuronales diseñado para evaluar la marcha y detectar posibles anomalías, aprovechando la capacidad de aprendizaje de estas redes para mejorar la precisión en la identificación de patrones anómalos en el movimiento. Por último, se plantea la creación de una aplicación que permita la representación visual de los datos recopilados, destacando las anomalías detectadas durante la marcha y brindando apoyo a los profesionales de la salud en la toma de decisiones.

La estructura del trabajo comprende cuatro capítulos esenciales. En el planteamiento del problema se contextualiza la necesidad de abordar la escasez de sistemas electrónicos para el monitoreo de la marcha. El marco teórico revisa las bases conceptuales y tecnológicas relevantes, proporcionando un fundamento sólido para el desarrollo de la investigación. La metodología detalla los pasos seguidos para lograr los objetivos propuestos, y la discusión de resultados analiza los hallazgos obtenidos, arrojando luz sobre la eficacia del sistema desarrollado.

En las conclusiones, se destaca el desarrollo de un sistema electrónico para estimar movimientos de la marcha se logró mediante la selección cuidadosa de materiales, incluyendo sensores inerciales y un microcontrolador. El sistema proporciona valores de

características de la aceleración y características de la velocidad angular que se utilizan para un análisis posterior en una red neuronal.

I. PLANTEAMIENTO DEL PROBLEMA

1.1 Motivación

La necesidad de sistemas electrónicos inalámbricos para monitorear el movimiento humano es un desafío continuo. El seguimiento de la marcha en individuos con movilidad limitada es crucial, pero la falta de sistemas específicos para recopilar datos en áreas clave, como la pierna, obstaculiza diagnósticos y tratamientos eficaces. El presente trabajo propone abordar este vacío mediante un sistema innovador basado en acelerómetros y giroscopios, aprovechando tecnología de sensores y análisis avanzado para ofrecer soluciones concretas en el seguimiento exhaustivo de la marcha. La inclusión de redes neuronales artificiales para analizar datos de la marcha, recogidos por sensores inerciales, añade relevancia. Su importancia radica en influir en el diagnóstico del proceso de rehabilitación de la marcha y cambiar la perspectiva en la que se afrontan los retos de movilidad. Más que atender una necesidad urgente, este estudio impulsa progreso e innovación en la convergencia de tecnología y salud, ofreciendo una perspectiva con el potencial de transformar vidas y reformular el abordaje de problemas de movilidad en la sociedad.

1.2 Estado del arte

Varios estudios recientes han explorado el uso de sensores IMU para el análisis de la marcha y la rehabilitación. Guajarathi (2019) desarrolló un sistema de análisis de la marcha que extrajo parámetros como tiempos de apoyo, balanceo y paso, longitudes de paso y velocidad. Rodríguez (2017) evaluó un sistema IMU comparándolo con el sistema BTS, demostrando mayor precisión en la medición de parámetros de la marcha. Qiu (2018) diseñó sistemas ambulatorios de análisis de la marcha en China y Estados Unidos, resaltando su utilidad en la evaluación y rehabilitación de pacientes con problemas neurológicos y articulares. Además, Gomez (2021) creó un dispositivo de asistencia para la rehabilitación pasiva en pacientes con lumbalgia aguda, mientras que Luna (2019) desarrolló un sistema para medir el balance corporal en tiempo real. Tribeño (2021) diseñó un dispositivo portátil para la rehabilitación pasiva de muñeca.

Los trabajos mencionados en el párrafo anterior contribuyen a una mejor comprensión del tema de investigación, estos estudios demuestran el potencial de los sensores IMU y tecnologías afines en la mejora de la marcha y la rehabilitación en diversas condiciones médicas.

1.3 Descripción del problema

Según la Organización Mundial de la Salud (OMS), alrededor de 1710 millones de personas en todo el mundo sufren de trastornos musculoesqueléticos, que son la principal causa de discapacidad motriz. Estos trastornos pueden ser repentinos y de corta duración o enfermedades crónicas que limitan las capacidades funcionales e incluso causan discapacidad permanente. Afectan a articulaciones, huesos, músculos, columna vertebral y diferentes sistemas o regiones del cuerpo, y son el factor principal que contribuye a la necesidad de rehabilitación a nivel global, siendo los niños responsables de aproximadamente dos tercios de las necesidades de rehabilitación en comparación con los adultos (Organización Mundial de la Salud, 2023).

La dificultad de la marcha es notoria en los trastornos musculoesqueléticos ya que se aprecia la disminución de la velocidad de la marcha, así como la alteración en las características del paso, inestabilidad o modificación en la sincronía de ambas extremidades inferiores, generando ineficacia para el desplazamiento y alterando las actividades de vida diaria (Cerda, 2018).

Existen diversas evaluaciones para analizar la evolución de la marcha, pero suelen ser incómodas para el paciente tanto en términos de traslado como a nivel económico. Actualmente, se utilizan métodos y dispositivos como plataformas de fuerza en el suelo, videografía y análisis de video, electromiografía (EMG) y sensores inerciales. Las plataformas de fuerza miden la fuerza de reacción durante la marcha, pero su acceso está limitado a entornos clínicos e investigaciones específicas (Acevedo, 2020). La videografía y análisis de video son una alternativa, pero los resultados pueden no ser tan precisos debido a la interpretación subjetiva. La electromiografía implica la colocación de electrodos en los músculos para medir la actividad eléctrica durante la marcha, aunque puede resultar incómoda y dolorosa para el paciente debido a los adhesivos utilizados en los electrodos.

Destacando que los sensores inerciales, como los acelerómetros y los giroscopios, son una opción preferida para la evaluación de la marcha. Estos sensores se colocan en diferentes partes del cuerpo, como los pies, las piernas o el tronco, para capturar el movimiento tridimensional durante la marcha. Proporcionan información precisa sobre la cinemática de la marcha, incluyendo la velocidad, longitud del paso, ángulo de las

articulaciones y simetría de la marcha. Además, los sensores inerciales ofrecen una solución versátil y no invasiva para evaluar la marcha de manera efectiva.

1.4 Formulación del problema

1.4.1 Problema General

La escasez de sistemas electrónicos adecuados basados en sensores que adquieran información de la zona en estudio (pierna), para el monitoreo de la marcha en personas con problemas de movilidad.

1.4.2 Problemas específicos

- ¿Cómo implementar el circuito electrónico para la adquisición del movimiento de la marcha en personas adultas mayores?
- ¿Cómo se puede desarrollar un algoritmo basado en redes neuronales para evaluar la marcha y detectar anomalías?
- ¿Cómo se puede desarrollar una aplicación que represente de manera gráfica los datos recolectados en relación a las anomalías de la marcha?

1.5 Objetivos

1.5.1 Objetivo general

Implementar un sistema electrónico basado en acelerómetros y giróscopos para el monitoreo de la marcha en personas con problemas de movilidad.

1.5.2 Objetivos específicos

- Implementar un sistema electrónico embebido inalámbrico para la adquisición del movimiento de la marcha en personas adultas.
- Desarrollar un algoritmo basado en redes neuronales para evaluar la marcha y detectar anomalías.
- Desarrollar una aplicación que permita la representación gráfica de los datos obtenidos de la marcha.

1.6 Justificación

El monitoreo constante y adecuado de la marcha es crucial para analizar su evolución, especialmente en pacientes que requieren rehabilitación, ya que una marcha adecuada permite un movimiento corporal correcto. Los terapeutas buscan mejorar la marcha de los pacientes, ya que es un factor importante para la salud en general, para superar

estas limitaciones, se utilizan sistemas instrumentados de medición de la marcha, que permiten evaluaciones objetivas y reducen la variabilidad en los diagnósticos. Estos sistemas pueden variar en complejidad y costo, pero avances tecnológicos recientes han permitido desarrollar herramientas más accesibles y precisas. La implementación del sistema electrónico para la medición y análisis de la marcha se situará en la pierna o pie para la captura del movimiento, esto proporciona una visión detallada de la cinemática de la marcha, incluyendo la posición, velocidad y aceleración en los tres ejes del espacio (x, y, z). Esta información es esencial para comprender el patrón de marcha y detectar alteraciones o disfunciones en el movimiento.

Este trabajo podría beneficiar la salud al desarrollar un sistema básico que proporcione información para el diagnóstico de la marcha. Esto permitirá detectar posibles problemas a tiempo y prevenir el avance de enfermedades, como desbalances musculares, patrones de marcha anormales, problemas de alineación de pies, dolor o molestias al caminar, coordinación y equilibrio deficiente, problemas posturales entre otros. Evitar estas consecuencias irreversibles será fundamental para mejorar la calidad de vida de las personas.

II. MARCO TEÓRICO

2.1 Antecedentes

2.1.1 Antecedentes internacionales

Guajarathi (2019) en su trabajo de investigación, India, tuvieron como objetivo el desarrollo de un sistema de análisis de la marcha basado en sensores IMU. Utilizaron un algoritmo para extraer parámetros de la marcha, como tiempos de apoyo, balanceo y paso, longitudes de paso, zancadas y velocidad. Recopilaron datos de voluntarios sanos que caminaron 40 metros en línea recta a velocidad normal. Se seleccionó el eje Y como el más sensible, detectaron los puntos de golpe de talón y punta utilizando señales de los ciclos de marcha, y calcularon los parámetros de la marcha. El sistema demostró ser efectivo para el análisis de la marcha, extrayendo con éxito los parámetros de los datos del acelerómetro y el giroscopio. El sistema mejoró la precisión y ofreció una técnica simple y de bajo costo. Este análisis puede ser útil en la rehabilitación de pacientes con problemas de marcha.

Rodríguez (2017), en su investigación la cual se llevó a cabo en Colombia, evaluaron un sistema de medición de parámetros temporales de la marcha humana basado en tecnología IMU. Compararon el sistema IMU con el sistema de captura de movimiento mediante BTS (Biomechanical Tracking System), sistema de seguimiento biomecánico, y analizaron el porcentaje de error. Se obtuvo como resultado que el error promedio fue inferior al 15% en todos los casos. El estudio incluyó un protocolo de evaluación que involucró la preparación de los participantes, la sincronización de los sistemas y la adquisición de datos durante la marcha en una cinta rodante. Se concluyó que en la comparación el sistema IMU demostró mayor precisión en la medición de los parámetros de la marcha en comparación con el sistema BTS.

Qiu (2018) desarrollaron un sistema de análisis de la marcha ambulatorio corporales basado en sensores IMU, en China. El sistema permitió mediciones precisas y ambulatorias en entornos del mundo real. Se analizaron datos de adultos sanos, pacientes con accidente cerebrovascular y enfermedades articulares. Los parámetros de la marcha fueron diferentes entre los sujetos sanos y los pacientes, proporcionando información valiosa para diagnóstico y tratamiento. El sistema fue efectivo para estimar los parámetros de la marcha y controlar los síntomas en personas sanas y pacientes con afecciones neurológicas o articulares. Su objetivo fue construir una plataforma portátil de análisis de la marcha de bajo costo,

inteligente y liviana basado en las redes de sensores corporales emergentes, que se pueden utilizar para la evaluación de la rehabilitación de pacientes con alteraciones de la marcha. El propósito de su trabajo fue investigar la efectividad de los datos de marcha obtenidos para diferenciar sujetos sanos y pacientes con marcha impedimentos Los resultados preliminares de los experimentos clínicos de marcha mostraron que el sistema propuesto puede ser efectivo en el diagnóstico auxiliar y la formulación del plan de rehabilitación en comparación con los existentes lo que indicó que el método propuesto tiene un gran potencial como auxiliar para la evaluación de la rehabilitación.

Qiu (2018) desarrolla un artículo, en E.E.U.U, donde presenta un sistema de análisis de la marcha basado en sensores corporales, que ha demostrado ser preciso y eficiente en comparación con métodos convencionales. Esta investigación tiene como objetivo construir un sistema de análisis de la marcha ambulatorio basado en el cuerpo emergente redes de sensores (c). El sistema se probó en sujetos sanos y pacientes con marcha anormal, y se concluye que tiene un gran potencial para la rehabilitación médica. El sistema puede ser una herramienta auxiliar para rehabilitación médica. Dos características clave son la adquisición de datos e interconexión. BSN (Body Sensor Network), red de sensores corporales, desarrolló completamente la superioridad de red inalámbrica e informatización, así como efectivamente realizó el monitoreo continuo y ambulatorio de la salud humana.

2.1.2 Antecedentes nacionales

Gómez, K. (2021) en su tesis de grado, “Diseño de un dispositivo para la rehabilitación pasiva en pacientes con diagnóstico de lumbalgia aguda”, Perú, dicho trabajo de investigación tiene como objetivo el diseño de un dispositivo de asistencia para la rehabilitación pasiva en pacientes con diagnóstico de lumbalgia aguda. La metodología usada para el desarrollo del dispositivo fue la tecnología soft, con actuadores flexibles que permiten la movilidad del usuario, con respecto al diseño incorporó una batería para proporcionar asistencia mediante actuadores piezoeléctricos, los cuales generaron fuerza de compresión al ser excitados eléctricamente. Como resultado del trabajo realizado el dispositivo contó con una autonomía de dos horas de batería, pero también dispuso de una barra como actuador mecánico adicional, lo que permite al usuario continuar experimentando asistencia de movimiento en la zona lumbar incluso sin la activación de los actuadores. Se concluyó que el dispositivo diseñado portable cumplió el objetivo de ayudar en la rehabilitación pasiva en pacientes con diagnóstico de lumbalgia aguda.

Luna, J. (2019) en su tesis de grado, “Desarrollo de un sistema de adquisición para la evaluación del balance corporal en base a la medida del movimiento del tronco”, Perú, el objetivo fue desarrollar un sistema electrónico para medir en tiempo real el ángulo del movimiento del tronco de una persona mientras realiza movimientos laterales (izquierda o derecha) que puedan afectar su equilibrio corporal, donde se evalúa la velocidad ante cambios predecibles e impredecibles en su posición de equilibrio. Los pasos de la metodología usada fueron el diseño del sistema donde se establece el uso del sensor inercial MPU6050 para la adquisición de la señal del movimiento, la selección de filtros donde usa el filtro complementario y el filtro Kalman, el desarrollo de interfaz gráfica desarrollada utilizando un microcontrolador Rasberry Pi 3B y la obtención de resultados donde obtuvieron respuesta aproximada de 30.2ms con el filtro complementario y 25.2ms aplicando el filtro de Kalman; estos resultados demostraron que el sistema responde de forma óptima en ambos casos. Como conclusión, el sistema de adquisición cumple con la medición correcta del ángulo del movimiento del tronco mientras la persona está en movimiento.

Tribeño, E. (2021) en su tesis de grado, “Diseño de un dispositivo portátil para la rehabilitación pasiva de muñeca con tres grados de libertad”, Perú, tuvo como objetivo el desarrollo de un dispositivo portátil de rehabilitación pasiva de muñeca que permita al usuario realizar movimientos específicos de flexión/extensión, desviación radial/cubital y pronación/supinación. El diseño del dispositivo fue adaptado a la antropometría del antebrazo y muñeca de la mujer peruana. La metodología utilizada en este proyecto consiste en definir una lista de exigencias, elaborar una estructura de funciones, plantear una matriz morfológica y establecer conceptos de solución que serán evaluados según criterios técnicos y económicos. El diseño consta de un hardware electrónico que a su vez se compone de sensores inerciales (IMU), un controlador que envía y recibe información a la interfaz de usuario a través del módulo bluetooth. Se concluye que el prototipo realizado permite analizar los movimientos asistidos de la muñeca respecto a cada movimiento especifico, así mismo se realizó una interfaz de usuario mediante un aplicativo móvil.

2.2 Bases teóricas

2.2.1 La marcha humana

La marcha humana es un patrón coordinado de movimiento al caminar. Implica fases como el balanceo de brazos, oscilación de piernas y transferencia de peso. Variaciones en velocidad y estilo son comunes. El análisis de la marcha es importante para medicina, rehabilitación y biomecánica, permitiendo comprender su función, detectar anomalías y evaluar terapias.

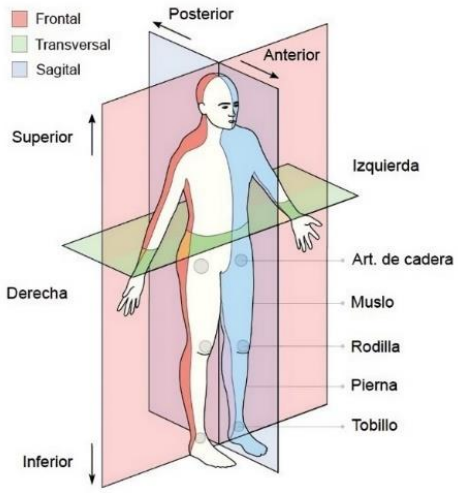
2.2.1.1 Características y convenciones anatómicas

Para iniciar en al análisis de la marcha es necesario tener en cuenta normas que facilitan su descripción y caracterización. La marcha es evaluada desde una posición erguida de la persona, con los pies juntos y las palmas de las manos hacia adelante. El cuerpo humano se analiza desde tres planos denominados sagital, transversal y frontal, los cuales se definen en relación con partes del cuerpo como la cabeza y el torso, y se ajustan según la rotación o el desplazamiento de la persona. El presente trabajo se desarrollará en el plano sagital. (Luna., 2017).

Se observa en la Figura 1 los planos anatómicos y segmentos de los miembros inferiores.

Figura 1

Planos anatómicos y segmentos

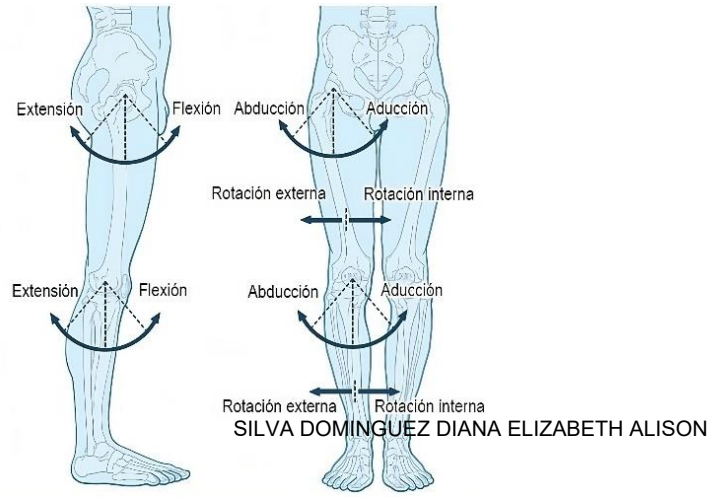


Nota. Planos anatómicos y segmentos de los miembros inferiores. La intersección de los planos frontal y sagital se identifica como la línea media del cuerpo. Tomado de Luna. (2017).

Las extremidades inferiores, piernas, son las estructuras esenciales para la movilidad. Para el estudio de la marcha se dividen estas extremidades inferiores en segmentos que están determinados por las articulaciones. Los segmentos mencionados son muslo, pierna y pie, los cuales están conectados por articulaciones denominadas cadera, rodilla y tobillo, los que son responsables de modificar la orientación de los segmentos mencionados con anterioridad. En la Figura 2, se puede apreciar las nomenclaturas mencionadas.

Figura 2

Movimientos articulares de los miembros inferiores



Nota. Movimientos de articulaciones de los miembros inferiores. Tomada de Luna, J. (2017).

2.2.1.2 Ciclo de la marcha

El ciclo de marcha es una secuencia repetitiva de eventos que ocurren al caminar. Comienza cuando un pie hace contacto con el suelo y termina cuando ese mismo pie vuelve a hacer contacto. (Rodríguez, 2017).

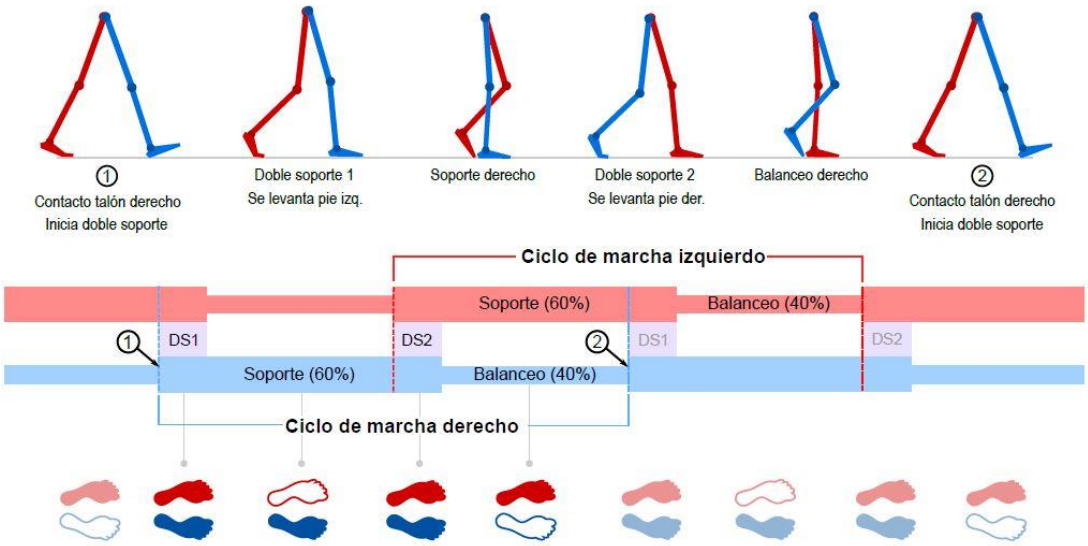
El ciclo de marcha se divide en dos períodos: el período de soporte y el período de balanceo. En el período de soporte, el pie en contacto con el suelo soporta la mayor parte del peso corporal mientras el otro pie avanza hacia adelante. Este período representa aproximadamente el 60% de la duración del ciclo de marcha. Por otro lado, el período de balanceo comienza cuando el pie se levanta del suelo. Durante este período, el peso del cuerpo se apoya en el otro pie, permitiendo que el pie en movimiento avance hacia su próximo punto de apoyo. El período de balanceo constituye el 40% restante del ciclo de marcha.

Hay momentos en el ciclo de marcha en los que ambos pies están en contacto con el suelo, llamados períodos de doble soporte. Durante estos períodos, se transfiere la mayor parte del peso corporal de un pie al otro. El primer período de doble soporte comienza cuando el pie que realiza el ciclo de marcha hace contacto con el suelo y dura hasta que el pie opuesto se levanta. El segundo período de doble soporte ocurre hacia el final del período de soporte, justo antes de levantar el pie de apoyo del suelo, cuando el pie opuesto hace contacto con el suelo. (Luna, 2017).

En la Figura 3 se muestra el ciclo de la marcha humana.

Figura 3

Ciclo de la marcha



Nota. Ciclo de la marcha normal. Proceso de marcha típica. El tono azul indica el miembro inferior derecho, mientras que el tono rojo representa el izquierdo. La porción superior ilustra diferentes etapas del ciclo de marcha en el miembro derecho, indicando su inicio y conclusión mediante los marcadores uno y dos, respectivamente. En la parte inferior, las huellas indican qué pie está en contacto con el suelo en cada fase. Tomado de Luna J. (2017).

2.2.1.3 Parámetros de la marcha

En el estudio de la marcha, se utilizan mediciones conocidas como parámetros de la marcha para cuantificar y analizar el proceso.

El movimiento puede ser estudiado por medio de sensores que permitan medir estas condiciones, por lo cual se ha planteado el uso de medición de la aceleración lineal y velocidad angular ya que estas variables ofrecen información para analizar y comprender los patrones de movimiento. La aceleración lineal proporciona datos sobre la tasa de cambio de velocidad en dirección lineal, mientras que la velocidad angular revela la velocidad de rotación alrededor de un eje específico. Estas mediciones combinadas permiten una evaluación más completa y precisa de los comportamientos y características del movimiento estudiado.

a. Aceleración Lineal:

Se refiere a la magnitud y la dirección de la tasa de cambio de velocidad de un objeto en movimiento. Este concepto se utiliza comúnmente en el contexto de sensores inerciales, como acelerómetros, que miden la aceleración experimentada por un objeto en las tres dimensiones del espacio: x, y, y z. Se mide en unidades de aceleración, como metros por segundo al cuadrado (m/s²) en el sistema internacional. La aceleración lineal $〚(a〛\_t)$ es el cambio de la velocidad (Δv) entre la variación del tiempo (Δt), se representa en la ecuación (1). (Martín, 2003)

$$a_t = \Delta v/\Delta t \qquad (1)$$

b. Velocidad Angular:

Se refiere a la magnitud y dirección de la tasa de cambio angular de un objeto en movimiento. Este concepto es fundamental en la medición del movimiento rotacional y se utiliza comúnmente en el contexto de sensores inerciales, como giroscopios. La velocidad angular describe cómo cambia la orientación de un objeto en torno a sus ejes de rotación, y se expresa típicamente en unidades angulares, como radianes por segundo (rad/s). La velocidad angular (w) es el cambio del ángulo rotado ($\Delta\theta$) entre el cambio del tiempo (Δt), se representa en la ecuación (2). (Martín, 2003)

$$w = \Delta\theta/\Delta t \qquad (2)$$

2.2.2 Dispositivos electrónicos

2.2.2.1 Sensores

Los sensores detectan estímulos y los convierten en señales eléctricas. Se usan en campos como electrónica, ingeniería y biomedicina.

2.2.2.2 Sensores inerciales

En física, "inercial" se refiere a la capacidad de un cuerpo para mantener su estado de reposo o movimiento uniforme a menos que actúe sobre él una fuerza externa. El "movimiento inercial" implica que las fuerzas externas afectan el desplazamiento de un objeto, ya sea cambiando su estado o modificando su trayectoria. En el ámbito de la Unidad de Medición Inercial (IMU), se analizan y miden las características cinemáticas de un objeto en movimiento, vinculadas a sus propiedades inerciales. Los IMU, constituye un dispositivo electrónico diseñado para registrar la aceleración lineal y la velocidad angular de un objeto. Sus principales componentes incluyen acelerómetros, giróscopos y, en ocasiones, magnetómetros. Los acelerómetros registran la aceleración en tres direcciones, los giróscopos miden la velocidad angular, y los magnetómetros contribuyen a mejorar la precisión de la orientación.

2.2.2.3 Acelerómetro

Dentro de los sensores inerciales se encuentran los acelerómetros, se emplean para la medición de la aceleración, es decir, variaciones de velocidad lineal. La medición se basa en la fuerza de inercia que se genera cuando se produce un cambio de velocidad, siguiendo la Ley de Newton. (Valcarce, 2014)

2.2.2.4 Giroscopio

El giroscopio, además de ser un sensor inercial, mide velocidades angulares o de rotación, presenta dos tipos de movimiento principales; precisión y nutación, derivadas de las ecuaciones de Euler Los giroscopios son usados en sensores de cambio de dirección. (Finn, 1970)

2.2.2.5 Sensor inercial MPU6050

El MPU6050 es un sensor inercial compacto de seis ejes. Tiene un acelerómetro de tres ejes y un giróscopo de tres ejes en un solo chip. Utiliza tecnología MEMS para medir la aceleración lineal y la velocidad angular. Los acelerómetros MEMS detectan la aceleración midiendo el desplazamiento de una masa microscópica dentro del dispositivo y convierten

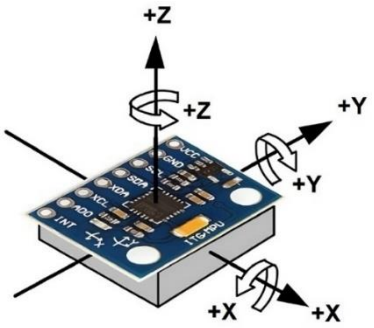
esta información en una señal eléctrica que representa la magnitud y dirección de la aceleración.

Algunas características clave del sensor MPU6050, iniciando por el acelerómetro de tres ejes el cual mide la aceleración lineal en las direcciones ‘x’,‘y’,‘z’, proporciona información sobre los cambios de velocidad y dirección del movimiento. El rango de medición se puede configurar según las necesidades, generalmente desde ±2 g hasta ±16 g. El giróscopo de tres ejes mide la velocidad angular en los ejes ‘x’,‘y’,‘z’, da información sobre la velocidad de rotación y la orientación, su rango de medición se puede configurar generalmente desde ±250°/s hasta ±2000°/s.

Al combinar el acelerómetro y giróscopo, el MPU6050 proporciona mediciones precisas de movimiento, orientación y cambios de velocidad angular. Es conocido por su facilidad de uso y disponibilidad en el mercado. En la Figura 4 se muestra el módulo MPU 6050 donde se distinguen los ejes ‘x’,‘y’,‘z’.

Figura 4

Modulo acelerómetro y giroscopio MPU 6050



Nota. El módulo Acelerómetro MPU tiene un giroscopio de tres ejes con el que podemos medir velocidad angular y un acelerómetro también de 3 ejes con el que medimos los componentes X, Y y Z de la aceleración.

2.2.2.6 Microcontrolador

Un microcontrolador es un dispositivo electrónico integrado en un solo chip que contiene un procesador central, memoria, periféricos de entrada/salida y otros componentes necesarios para controlar y ejecutar tareas específicas en sistemas embebidos. Es una forma compacta y eficiente de implementar funciones de control en diversos dispositivos electrónicos.

En la Tabla 1 se muestran características clave de los microcontroladores.

Tabla 1

Características generales de microcontroladores

Característica	Descripción
Procesador central	Ejecuta instrucciones y realiza cálculos, su arquitectura varía según el fabricante y el modelo del microcontrolador.
Memoria	Almacena el programa o firmware, así como datos temporales durante la ejecución. Esto incluye memoria de programa (flash o ROM) para almacenar el código y memoria RAM para almacenar datos.
Periféricos de entrada/salida (I/O):	Estos periféricos pueden incluir puertos GPIO (Entrada/Salida de Propósito General) para la conexión de sensores, actuadores, pantallas, comunicación serial (UART, SPI, I2C), conversiones analógico-digital (ADC) y digital-analógico (DAC), entre otros.
Comunicación	Interfaces de comunicación integradas, como UART, SPI, I2C, CAN, Ethernet, USB, entre otras, que permiten la comunicación con otros dispositivos.

Nota. Descripción de algunas de las características de los microcontroladores.

Los microcontroladores ofrecen una plataforma flexible y programable para implementar lógica de control y funcionalidades en sistemas electrónicos. Interactúan con el entorno, el procesamiento de datos y la toma de decisiones en tiempo real. Al programar el microcontrolador con el lenguaje adecuado, se pueden desarrollar aplicaciones personalizadas y adaptadas a las necesidades específicas de cada proyecto.

2.2.2.7 Microcontrolador NODE MCU ESP8266

El NodeMCU ESP8266 incluye el chip ESP8266 y una antena Wi-Fi integrada, lo que lo hace ideal para aplicaciones que requieren conectividad inalámbrica. Es programable a través del entorno de desarrollo de Arduino, lo que facilita la escritura y la carga de código. La Tabla 2 muestra características del microcontrolador MCU ESP8266.

Tabla 2

Características del microcontrolador Node MCU ESP8266

Característica	Descripción
Procesador y memoria:	Procesador central de 32 bits de Tensilica L106.
Frecuencia típica	80 MHz
Conectividad Wi-Fi	Capacidad de conectividad Wi-Fi integrada. Con protocolo 802.1 1 n hasta 150 Mbit/s Frecuencia Wifi de 2.4 GH.
Interfaces de comunicación	I2C, UART, SPI, y ADC.

Nota. Descripción de las características principales del microcontrolador Node MCU ESP8266.

En la Figura 5 se muestra la tarjeta de desarrollo NODE MCU ESP8266.

Figura 5

NODE MCU ESP8266

Nota. Imagen de un microcontrolador Node MCU ESP8266

2.2.3 Procesamiento de señales

El procesamiento de señales mejora la calidad y extrae información útil. Incluye adquisición, preprocesamiento, análisis de características y aplicación específica. Se utiliza

para tomar decisiones basadas en la señal, como clasificación, reconocimiento de patrones o generación de alarmas.

Iniciando por la adquisición de señales donde implica la captura y digitalización de las señales analógicas usando sensores o dispositivos de medición, para su posterior procesamiento por donde se aplican técnicas de limpieza, filtrado y normalización de las señales. e incluso la eliminación de ruido. Luego de culminado el proceso anterior se procede al análisis y extracción de características lo cual implica el cálculo de medidas estadísticas, transformaciones matemáticas, detección de eventos o cambios significativos, o la identificación de patrones específicos, las señales pasan al procesamiento y análisis específico de la aplicación que usa algoritmos y técnicas específicas según el dominio de aplicación, por ultimo esta la etapa de interpretación y toma de decisiones en la que se utiliza la información extraída de las señales para tomar decisiones o realizar acciones específicas. Esto puede incluir la clasificación de señales, el reconocimiento de patrones, la detección de eventos. Oppenheim, A. (2011).

2.2.3.1 Python

Python es un lenguaje de alto nivel, creado por Guido van Rossum lanzado a finales de los años 80, cuenta con una sintaxis clara y legible, lo cual facilita su escritura y comprensión de código.

Cada vez es más común el uso de Python para el procesamiento de señales debido a su versatilidad y las bibliotecas disponibles, algunas de las bibliotecas más usadas en este ámbito son NumPy, SciPy, Matplotlib y Pandas. (Python.org, 2023).

2.2.3.2 Google Colab

Es una plataforma en línea proporcionada por Google que permite escribir y ejecutar código en el entorno de Jupyter Notebooks de forma gratuita. Los cuadernos de Colab se almacenan en Google Drive, lo que facilita el acceso y la colaboración en tiempo real. Brinda soporte para Python.

2.2.3.3 Django

Django es un framework que permite desarrollar aplicaciones web, basado en Python. Además, Django es una plataforma gratuita y de código abierto.

2.2.3.4 Redes neuronales artificiales

Las redes neuronales artificiales (RNA) son modelos computacionales inspirados en el funcionamiento del sistema nervioso humano. Estas redes están diseñadas para aprender y realizar tareas complejas mediante el procesamiento de información en paralelo, como reconocimiento de patrones, clasificación, regresión, procesamiento de lenguaje natural, entre otras.

Las RNA están compuestas por capas de neuronas artificiales, también conocidas como nodos o unidades, organizadas en diferentes capas. Cada neurona recibe una o varias entradas, aplica una función de activación para combinar las entradas ponderadas con los pesos de las conexiones y produce una salida. Estas conexiones, también llamadas sinapsis, tienen asociados pesos que determinan la importancia de cada entrada en el cálculo de la salida. (Zurada, 2005).

En una red neuronal con una sola neurona, la salida (y) se puede calcular mediante una ecuación lineal, donde las entradas (x) multiplicadas por sus respectivos pesos (b), y luego se suma un sesgo (a), tal como se muestra en la ecuación (3).

$$y = bx + a \qquad (3)$$

En redes neuronales más avanzadas y complejas, que cuentan con múltiples capas y emplean funciones de activación no lineales, las relaciones entre las entradas y salidas se vuelven más intrincadas y no pueden ser simplemente expresadas mediante ecuaciones lineales. A pesar de esta complejidad, en capas individuales de la red, especialmente en aquellas completamente conectadas, cada conexión entre una neurona y las neuronas de la capa anterior sigue una relación lineal, similar a la ecuación lineal mencionada anteriormente.

La inclusión de funciones de activación no lineales, tales como la función sigmoide, tangente hiperbólica o ReLU, es lo que habilita a las redes neuronales para aproximar funciones no lineales más sofisticadas. Estas funciones permiten a las redes neuronales llevar a cabo tareas avanzadas, como la clasificación y regresión en conjuntos de datos más complejos.

El proceso de entrenamiento de una red neuronal implica los siguientes pasos:

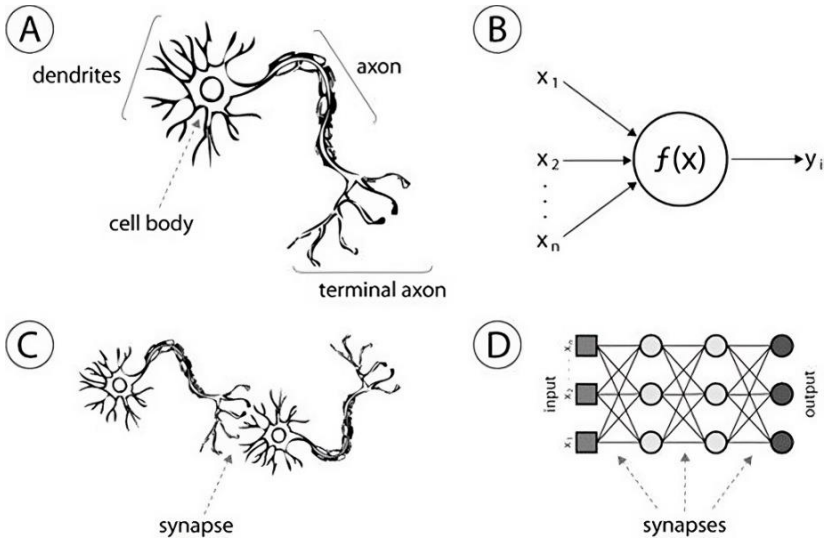
a) **Inicialización de pesos:** Al comienzo del entrenamiento, los pesos y los parámetros de la red se inicializan aleatoriamente o con algún valor predefinido.

b) **Paso hacia adelante (Forward Propagation):** Durante esta fase, los datos de entrenamiento se introducen en la red y se propagan a través de las capas. Las neuronas calculan sumas ponderadas de las entradas y aplican funciones de activación para generar las salidas.
c) **Cálculo de la pérdida o error:** Se calcula comparando las salidas de la red con las salidas deseadas para los datos de entrenamiento. Esta pérdida mide cuán cerca o lejos está la red de producir los resultados correctos.
d) **Paso hacia atrás (Backpropagation):** En esta etapa, el algoritmo de backpropagation calcula las gradientes de la pérdida con respecto a los pesos y parámetros de la red. Estos gradientes indican cómo los cambios en los pesos afectarían la pérdida.
e) **Actualización de pesos:** Los algoritmos de optimización utilizan los gradientes calculados en el paso anterior para ajustar los pesos y parámetros de la red. El objetivo es reducir la pérdida con cada época o iteración.
f) **Época o iteración:** La red se expone repetidamente a los datos de entrenamiento y ajusta sus pesos para mejorar su rendimiento.
g) **Validación y ajuste:** A medida que la red se entrena, es importante monitorear su rendimiento en un conjunto de datos de validación que no se usa para el entrenamiento. Si el rendimiento en los datos de validación no mejora o empeora, se pueden tomar medidas para ajustar la arquitectura de la red, la tasa de aprendizaje u otros parámetros.
h) **Finalización del entrenamiento:** El entrenamiento se detiene cuando el número de épocas o cuando la pérdida se ha reducido a un valor aceptable.

Para comprender la representación gráfica de la red neuronal, se ilustrará en la Figura 8, donde se realizará una comparación entre la neurona biológica y la artificial. Asimismo, se establecerá una relación entre la sinapsis biológica y la conexión en una red neuronal artificial. La Figura 6 presenta una comparación entre la red neuronal artificial y una neurona real.

Figura 6

Red neuronal artificial



Nota. En la sección A se representa una neurona humana, en la parte B se visualiza una neurona artificial, en la imagen C se ilustran las conexiones biológicas entre neuronas (sinapsis biológicas) y en el segmento D se muestra la conexión entre neuronas en una red neuronal artificial (sinapsis de una red neuronal artificial). (Matarollo, 2013)

2.2.3.5 Inteligencia artificial

La inteligencia artificial (IA) es la simulación de procesos de pensamiento humano por parte de sistemas computacionales para realizar tareas como aprendizaje, razonamiento, percepción y toma de decisiones.

a) **Inteligencia artificial supervisada:** Es un enfoque de aprendizaje automático en el que se entrena a un sistema utilizando datos etiquetados, lo que le permite hacer predicciones o clasificaciones basadas en patrones identificados durante el entrenamiento.

b) **Inteligencia artificial no supervisada:** Implica el aprendizaje automático a partir de datos no etiquetados, con el objetivo de encontrar patrones y estructuras ocultas en los datos sin orientación previa.

En el presente trabajo se desarrollará la red neuronal artificial la cual usará la inteligencia artificial supervisada.

2.2.3.6 Métrica de precisión

La precisión es una métrica fundamental utilizada en problemas de clasificación en el aprendizaje automático. Se define como la proporción de predicciones correctas realizadas por el modelo con respecto al total de predicciones. Se calcula dividiendo los verdaderos positivos (instancias positivas clasificadas correctamente) entre la suma de los verdaderos positivos y los falsos positivos (instancias negativas clasificadas incorrectamente como positivas).

En términos simples, la precisión en el contexto de evaluación de modelos de clasificación mide la exactitud de las predicciones positivas realizadas por el modelo.

La precisión proporciona una visión general de la calidad de las predicciones positivas, indicando cuántas de ellas son correctas. La precisión es valiosa, pero su interpretación debe contextualizarse y complementarse con otras métricas relevantes según las características específicas del problema y el conjunto de datos. La precisión debe ser lo más cercano a 100% y el error deber lo más cercano a 0. (Nur-Masruriyah et al., 2019)

2.2.3.7 Funciones de activación

Las funciones de activación son componentes fundamentales en las redes neuronales y se utilizan para introducir no linealidades en los modelos. Estas funciones determinan la salida de cada neurona en función de su entrada ponderada y proporcionan la capacidad de aprender patrones complejos y no lineales en los datos. Entre las funciones de activación más usadas se encuentran la función sigmoide, función ReLU, función softmaw, entre otras.

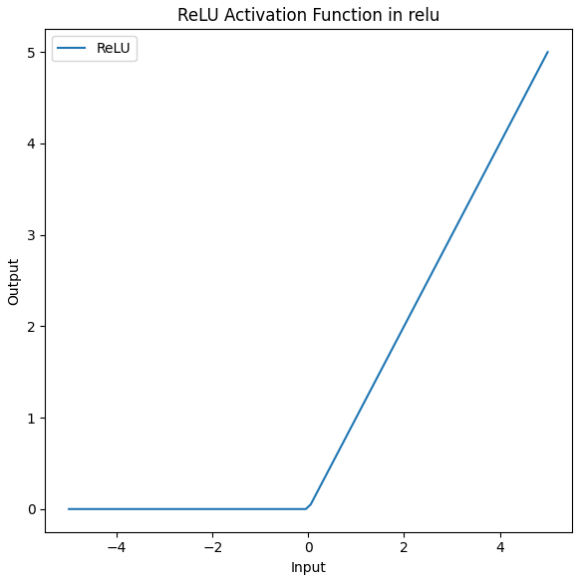
2.2.3.8 Función de activación ReLU

La función de activación usada fue ReLU, que significa "Rectified Linear Unit", en español unidad lineal rectificada, esta función de activación se usa en las capas de entrada de la red. Esta función devuelve cero para todos los valores negativos de x y para los valores no negativos, devuelve el mismo valor de x. En términos gráficos, ReLU traza una línea recta para los valores positivos de x y es completamente plana (devuelve cero) para los valores negativos. Su expresión matemática se puede ver en la ecuación (1) según (SuperDataScience Team, 2018). La ecuación (4) muestra la función de activavión ReLU, mintras que la Figura 7 nos muestra la gráfica de la misma.

$$ReLU(x) = \max(0, x) \qquad (4)$$

Figura 7

Gráfica de la función de activación ReLU



2.2.3.9 Función de activación Softmax

La función de activación softmax, se usa para la capa de salida de la red neuronal. Transforma valores originales en probabilidades, utilizando la exponenciación para suavizar las diferencias y el denominador para normalizar las probabilidades. Matemáticamente, para un conjunto de n valores $z\_1, z\_2, \dots, z\_n$, la función softmax se en la ecuación (5) según (SuperDataScience Team, 2018).

$$\text{softmax}(z_i) = \frac{e^{z}i}{\sum_{j=1}^{n} e^{z_j}} \quad (5)$$

Donde:

- z_i valor en la posición i del vector z.
- e número de Euler
- $\sum_{j=1}^{n} e^{z_j}$ sumatoria de las probabilidades sea igual a 1.

III. METODOLOGÍA.

3.1 Descripción de la metodología

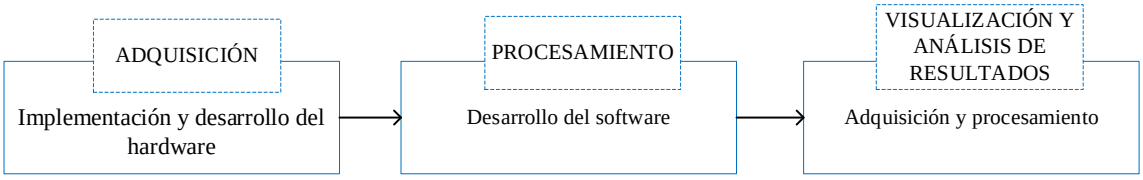
La presente investigación es de tipo cuantitativa, el nivel de investigación es experimental, descriptiva y aplicada ya que se desarrollará un dispositivo electrónico para el análisis de la marcha humana, de dicho dispositivo se obtendrán datos para su posterior análisis y aplicación respectiva.

3.2 Implementación de la investigación

Con el propósito de abordar de manera efectiva cada uno de los objetivos establecidos, el proyecto se desglosa en diversas fases que comprenden la implementación y desarrollo del hardware, obtención de datos, el procesamiento de información y la realización de pruebas para garantizar el funcionamiento adecuado. Las etapas se muestran en la Figura 8.

Figura 8

Etapas de la investigación



(i) (ii) (iii)

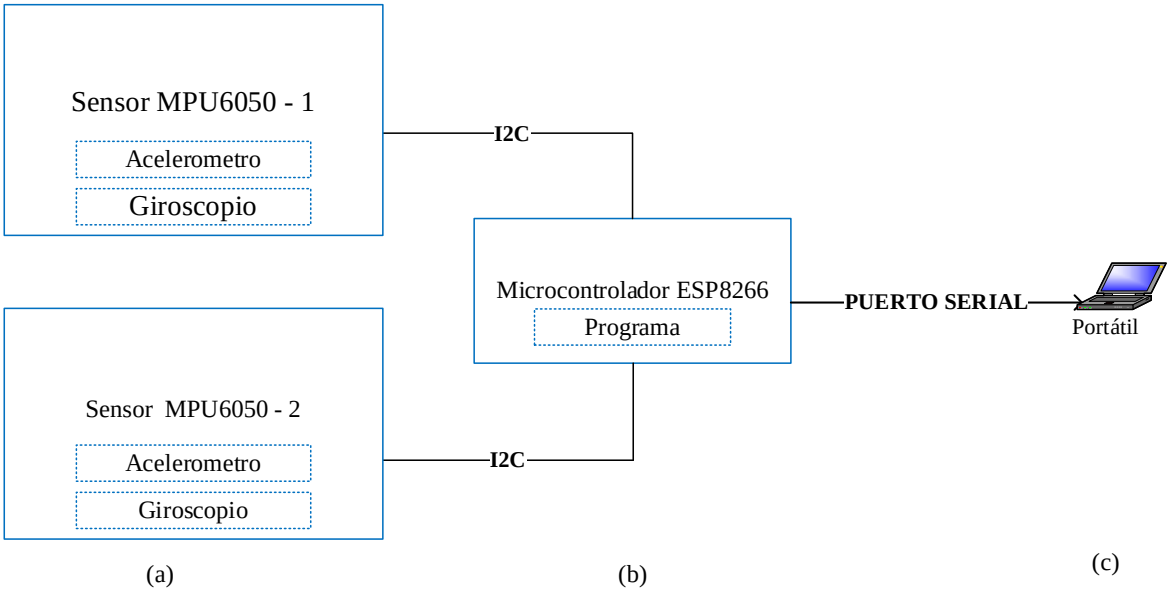
Nota. (i) La adquisición es la etapa en la que se implementa el circuito electrónico para así poder extraer los datos de cada sensor inercial, para luego transmitir mediante el microcontrolador. (ii) En la etapa de procesamiento se utilizan diferentes lenguajes de programación, cada uno propios para el tratamiento de los datos adquiridos. (iii) La etapa de visualización y análisis de resultados se podrán visualizar los datos según lo requerido, para su posterior análisis.

3.2.1 Implementación de circuito electrónico

Iniciando con la etapa de adquisición, se implementa el circuito electrónico en el que se usan dispositivos de sensado y un microcontrolador. Los dispositivos usados son los sensores inerciales MPU6050 y el microcontrolador ESP8266, para el análisis de la marcha. El la Figura 9 se describe la implementación del sistema.

Figura 9

Diagrama de bloques del sistema



Nota. (a) Sensor MPU6050 recopila datos de aceleración y giroscopio relacionados con la marcha. (b) Microcontrolador ESP8266 procesa los datos del sensor, los almacena en variables y transmite a la computadora. (c) La computadora recibe los datos para su posterior análisis.

a. Selección de sensores inerciales

Existen diferentes opciones de sensores inerciales que contienen acelerómetros y giroscopios, por ellos se debe hacer una respectiva comparación entre ellos para elegir el idóneo para el circuito. En la Tabla 3 representa la comparación realizada de los sensores inerciales.

Tabla 3

Comparativa de los sensores inerciales.

Parámetros	MPU9250	MPU6050
Alimentación	3.3V	3.3V – 5V
Rango de acelerómetro	+/- 2g, 4g, 8g y 16g	+/- 2g, 4g, 8g y 16g
Rango de giroscopio	+/- 250 Grad/Seg +/- 2000 Grad/Seg	+/- 250 Grad/Seg +/- 2000 Grad/Seg +/- 1000 Grad/Seg +/- 2000 Grad/Seg
Grados de libertad	9	6
Interfaz de comunicación	SPI e I2C	I2C

Las dos opciones presentan similitudes en cuanto a características técnicas para llevar a cabo la medición. No obstante, la distinción principal se encuentra en los grados de libertad (DOF, que indican la cantidad de sensores en cada eje de medición) ofrecidos por cada unidad inercial, así como en el costo asociado. En este contexto, se decidió emplear el sensor MPU6050, principalmente debido a su asequibilidad y a que cuenta con los grados de libertad necesarios para realizar la medición.

El MPU 6050 contiene tres acelerómetros y tres giroscopios, uno por cada eje (X, Y, Z). Cuenta con pines para comunicación I2C, esencial para el envió de datos. Su tamaño es pequeño el cual facilita su colocación en el área determinada.

b. Selección de microcontrolador

Dentro de los microcontroladores existen enumerarles contienen acelerómetros y giroscopios, por ellos se debe hacer su respectiva comparación entre ellos para elegir el idóneo para el circuito. En la Tabla 4 representa la comparación realizada de los microcontroladores.

Tabla 4

Comparativa de microcontroladores

Parámetros	ATMEGA2560	ESP8266
Alimentación	7V – 12V	3V – 3.3V
Frecuencia de operación	16 MHz	80 MHz
Número de líneas digitales	54	17 pines GPIO
Número de entradas analógicas	16	1
Temperatura de operación	-40°C a 85°C	Máxima de 125 °C
Memoria interna	256.8 KB RAM	4MB de memoria FLASH (32 Mbit)
Interfaz de comunicación	SERIAL, UART, I2C2, SPI.	I2C, SPI, WIFI 802.11 b/g/n

El microcontrolador ESP8266 fue elegido por su compatibilidad con el entorno de desarrollo Arduino, también por la velocidad en el procesamiento de datos y por su tamaño, lo que lo hace adecuado para aplicaciones donde el espacio es limitado.

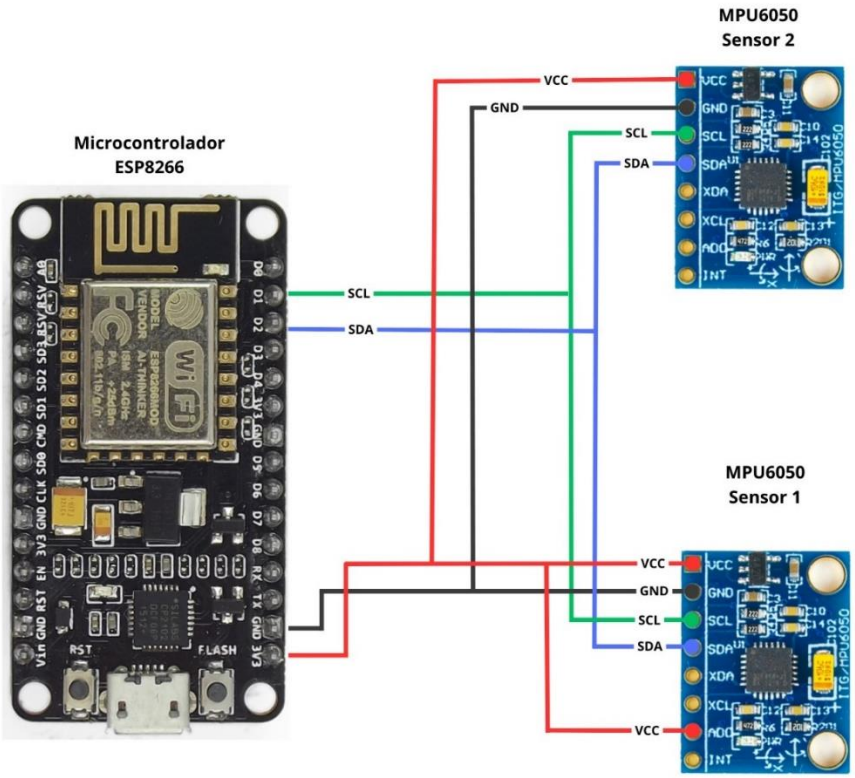
3.2.1.1 Hardware para la adquisición de datos

c. Esquema del circuito

El circuito incluye dos módulos MPU6050 y el microcontrolador ESP8266 el cual se utiliza para la obtención y la transmisión de datos de los sensores de movimiento a través de una red Wi-Fi. El esquema del circuito se muestra en la Figura 10.

Figura 10

Esquema del circuito



Nota. El circuito muestra las conexiones entre los dos sensores MPU6050 y el microcontrolador ESP8266, usando los pines correspondientes.

La adquisición de datos de ambos sensores inerciales MPU6050 mediante el microcontrolador ESP8266 se realizó a través del protocolo de comunicación I2C. Este protocolo emplea direcciones hexadecimales para la identificación de dispositivos en el bus.

El sensor inercial MPU6050, de manera predeterminada, tiene asignada la dirección 0x68, que corresponde a una ubicación específica en el bus destinada para la interacción con el microcontrolador. Dado que se utilizaron dos sensores inerciales, se asigna a cada uno una dirección de bus única. Para lograr esto, se conectó el pin AD0 de uno de los sensores a VCC, configurándolo en 1, lo que establece su dirección I2C en 0x69, para el segundo sensor.

Las conexiones se establecieron entre el pin D2 a la línea SDA (Serial Data Line), que es la línea de datos en el bus I2C. El pin D1 se utiliza para la línea SCL (Serial Clock Line), que es la línea de reloj en el bus I2C.

La conexión de los pines VCC (alimentación) y GND (tierra) de ambos sensores al pin de 3.3V y GND del ESP8266, respectivamente, se realiza para proporcionar la alimentación eléctrica necesaria a los sensores y asegurar una referencia común de tierra, la alimentación es proporcionada por la laptop.

En resumen, las conexiones de la Figura 13 son las siguientes:

- Los pines VCC y GND de los sensores MPU6050 a la fuente de alimentación de 3.3V del microcontrolador ESP8266.
- El pin SDA (Serial Data) del MPU6050 hacia el pin D1 del microcontrolador ESP8266.
- El pin SCL (Serial Clock) del MPU6050 hacia el pin D2 del microcontrolador ESP8266.
- El pin AD0 de uno de los sensores MPU6050 al VCC de la fuente de alimentación del microcontrolador, para activar el bus 0x69.

d. Calibración de sensores inerciales

El proceso de calibración del sensor inercial MPU6050 resulta fundamental debido a la posibilidad de que el sensor no se encuentre alineado en una posición horizontal de manera precisa. Para resolver esta situación, se pueden efectuar ajustes en los offsets del MPU6050 con el propósito de corregir posibles errores y lograr mediciones de mayor precisión. Los offsets se refiere a valores de corrección aplicados a las lecturas crudas (sin procesar) del sensor, estos ajustes compensan pequeñas desviaciones o errores sistemáticos en las mediciones, con el propósito de eliminar posibles sesgos o imprecisiones en los datos capturados por el sensor.

En el código para la calibración del sensor, se modifican contantemente los offsets intentando eliminar el error, es decir que las aceleraciones y velocidades angulares logren ser 0, es decir ax=0,ay=0,az=1g y gx=0,gy=0,gz=0.

Al iniciar la calibración ubicar el sensor en posición horizontal y evitar mover el sensor durante la calibración, ya que esta posición será el nivel de referencia para futuras mediciones.

Los valores deseados de la aceleración deben aproximarse a p_ax=0, p_ay=0, p_az=+16384 y la velocidad angular: p_gx=0 , p_gy=0 , p_gz=0.

Para este caso se considera la fuerza de gravedad igual 9.8 m/s^2. Se trabajó con el rango de acelerómetro de ±2g, el cual tiene una sensibilidad de 16384, por ello en la posición horizontal el valor en z debe ser cercano a 16384. Tal como se indica en la Tabla 5.

Tabla 5

Valores del sensor MPU6050

AFS_SEL	Rango de acelerómetro	Factor de sensibilidad	m/s^2
0	± 2g	16384	19.6

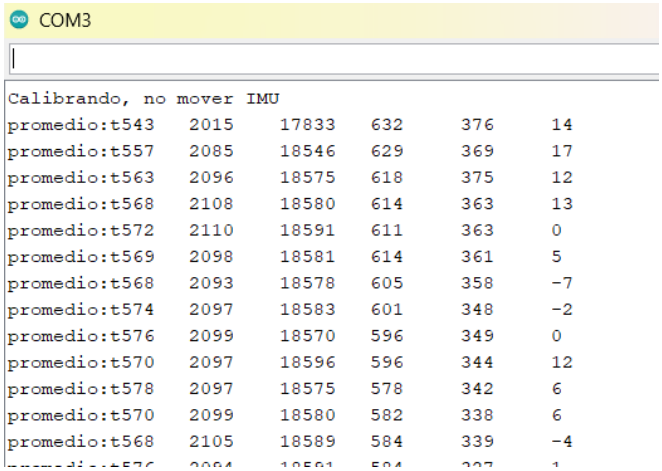
El indicador AFS_SEL se refiere a la configuración del rango completo de escala del acelerómetro, el rango del acelerómetro será de 2g, es decir dos veces la gravedad 2(9.8 m/s^2) dando un valor de 19.6 m/s^2 como valor máximo.

El factor de sensibilidad, para un acelerómetro configurado en 2g debe ser 16384, todos estos valores se encuentran en la hoja de datos del sensor.

El código para la calibración comienza con la inicialización del sensor MPU6050, muestra los offsets actuales, espera un carácter para iniciar la calibración y luego en el bucle principal lee las lecturas del sensor, aplica un filtro pasa bajos y realiza la calibración cada 100 lecturas. La calibración ajusta los offsets para mantener el sensor correctamente calibrado. La Figura 11 muestra el inicio de la calibración del sensor inercial donde se muestra un mensaje que no se debe mover el sensor inercial IMU al momento de iniciar la calibración.

Figura 11

Inicio de la calibración del sensor inercial



e. Posicionamiento de sensores

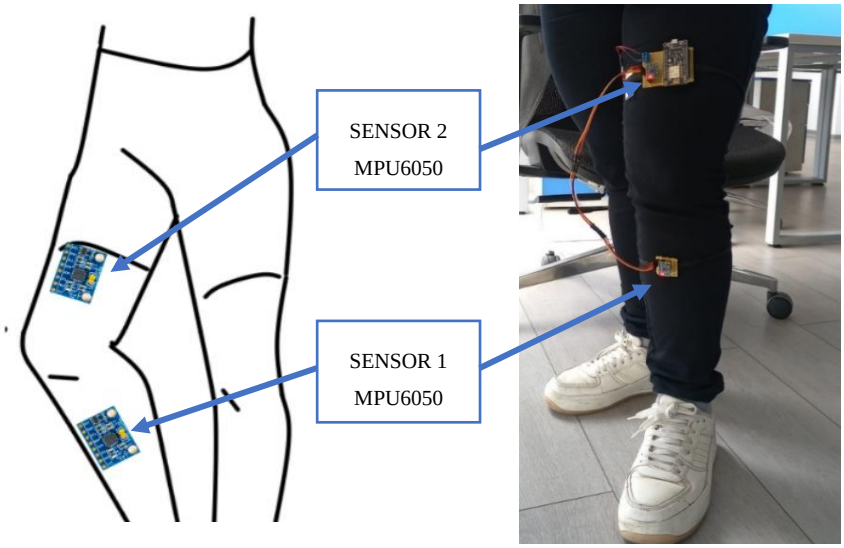
Los sensores inerciales, MPU6050, son colocados en áreas específicas del cuerpo o en el área determinada para la captura de datos de la marcha. Para este proyecto serán colocados en la pierna, específicamente en las siguientes zonas detalladas:

- Zona de media del muslo, encima de la rodilla.
- Zona de media de la pantorrilla, debajo de la rodilla.

La Figura 12 muestra el posicionamiento de los sensores, se observa que uno de ellos se encuentra la parte media del muslo, por encima de la rodilla y el otro se encuentra debajo de la rodilla.

Figura 12

Ubicación de sensores



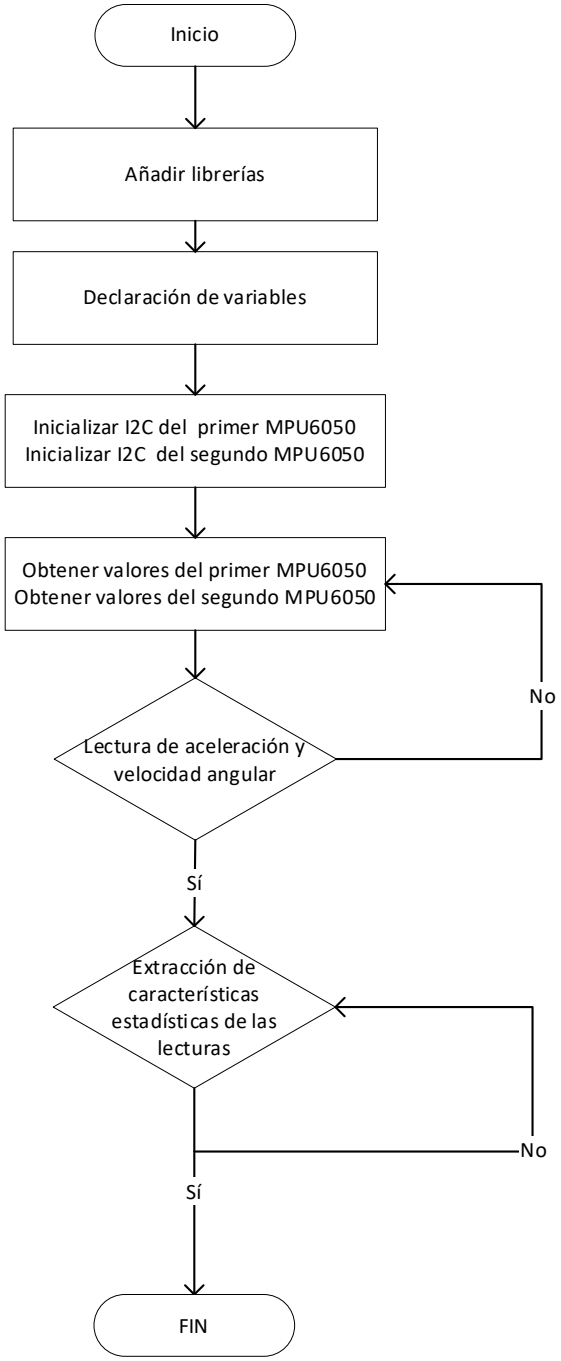
f. Adquisición de la información

El código de programación para el microcontrolador (ESP8266) que se encargará de leer los datos de los sensores. Esto implica configurar el protocolo de comunicación (I2C en el caso del MPU6050) y programar la obtención de datos de aceleración y giroscopio.

En la Figura 13 se observa el diagrama de flujo del código para la obtención de datos.

Figura 13

Diagrama de flujo del código para recolectar datos



Nota. Diagrama de flujo del código escrito en Arduino.

El código en Arduino se encarga de leer información de dos sensores MPU6050. Inicia por importar las bibliotecas necesarias, I2Cdev, MPU6050 y Wire. Estas bibliotecas ofrecen funciones y utilidades específicas para la comunicación I2C, simplificando la configuración y lectura de datos del MPU6050. I2Cdev maneja operaciones básicas de I2C, mientras que MPU6050 está diseñada para el MPU6050, proporcionando funciones específicas como inicialización y lectura de datos. Wire, una biblioteca estándar de Arduino, simplifica la comunicación I2C, facilitando la conexión y la transferencia de datos entre dispositivos.

Se crean variables para almacenar las lecturas crudas de los sensores MPU6050 en ambos ejes del acelerómetro (x, y, z) y giroscopio (x, y, z) para dos sensores distintos, sensor1 y sensor2. Estas variables se utilizarán para procesar y analizar las lecturas de los sensores en el programa, las variables son de tipo entero de 16 bits.

La comunicación serial fue a una velocidad de 57600 baudios. Esto estableció una conexión serial entre el sistema y la computadora, permitiendo la comunicación de datos entre ambos.

Una vez establecido lo mencionado en párrafos anteriores, se procede a verificar las conexiones de los sensores para luego dar inicio de la lectura y almacenarlas en variables.

La declaración de variables para ambos sensores para almacenar las lecturas crudas de los sensores. La lectura del sensor nos otorga datos crudos, las variables asignadas se muestran en la Figura 14.

Figura 14

Lectura de datos de los dos sensores

```
void loop() {
  // Leer las aceleraciones y velocidades angulares del primer sensor
  sensor1.getAcceleration(&ax1, &ay1, &az1);
  sensor1.getRotation(&gx1, &gy1, &gz1);

  // Leer las aceleraciones y velocidades angulares del segundo sensor
  sensor2.getAcceleration(&ax2, &ay2, &az2);
  sensor2.getRotation(&gx2, &gy2, &gz2);
```

Las lecturas crudas de los sensores MPU6050 se deben convertir a unidades físicas específicas, para obtener valores en unidades físicas como metros por segundo cuadrado (m/s^2) para el acelerómetro y grados por segundo (°/s) para el giroscopio, se utilizan factores de escala.

La aceleración en m/s^2 se calcula multiplicando el valor crudo por el factor de escala. Este factor de escala está definido como la relación entre la gravedad y factor de sensibilidad del acelerómetro, en este caso, el factor de escala es (9.81/16384). El cálculo se debe realizar para la aceleración en todos los ejes de ambos sensores. Se emplean los valores del eje x del primer sensor a modo de ejemplo, por ende, el cálculo de la aceleración en el eje x se realiza de la siguiente forma:

$$ax1\_m\_s2 = ax1 \times (9.81/16384) \quad (5)$$

Las velocidades angulares son expresadas en grados por segundo (°/s) para todos los ejes de ambos sensores. La conversión de las lecturas crudas del giroscopio a unidades físicas se realiza mediante el factor de escala (250 / 32768), donde 250 representa la velocidad angular máxima del giroscopio y 32768 es el rango completo de escala del giroscopio para ±250°/s. Este procedimiento garantiza que las velocidades angulares medidas estén adecuadamente representadas en grados por segundo, ofreciendo información precisa sobre la rotación en los tres ejes del sensor. A modo de ejemplo se usan los valores del eje x del primer sensor a modo de ejemplo, por ende, el cálculo de la velocidad angular se realiza de la ecuación (6)

$$gx1\_deg\_s = gx1 \times (250/32768) \quad (6)$$

En la Figura 15 se muestran líneas de código para convertir las lecturas crudas de los sensores MPU6050 en unidades físicas comprensibles, permitiendo que las lecturas se interpreten correctamente en términos de aceleración y velocidad angular.

Figura 15

Código de conversión de aceleraciones y velocidades angulares.

```
float ax1_m_s2 = ax1 * (9.81 / 16384.0);
float ay1_m_s2 = ay1 * (9.81 / 16384.0);
float az1_m_s2 = az1 * (9.81 / 16384.0);
float gx1_deg_s = gx1 * (250.0 / 32768.0);
float gy1_deg_s = gy1 * (250.0 / 32768.0);
float gz1_deg_s = gz1 * (250.0 / 32768.0);

float ax2_m_s2 = ax2 * (9.81 / 16384.0);
float ay2_m_s2 = ay2 * (9.81 / 16384.0);
float az2_m_s2 = az2 * (9.81 / 16384.0);
float gx2_deg_s = gx2 * (250.0 / 32768.0);
float gy2_deg_s = gy2 * (250.0 / 32768.0);
float gz2_deg_s = gz2 * (250.0 / 32768.0);
```

Finalizando el código se imprimen las lecturas de los sensores MPU6050 en el monitor serial, separando las lecturas de los diferentes ejes y sensores mediante etiquetas y caracteres de tabulación.

3.2.2 Entrenamiento de la red neuronal

Para el diseño de la red se consideraron las etapas de descripción de los datos adquiridos, el diseño de la red neuronal y el entrenamiento de la red neuronal.

3.2.2.1 Datos adquiridos

Para el diseño de la red neuronal se utilizarán los datos adquiridos ya categorizados en valores de entrada y valores de salida para así poder continuar con el entrenamiento En la Tabla 6, se encuentran registrados los nombres de las variables, que representan los valores de aceleración provenientes de ambos sensores.

Tabla 6

Variables que representan las aceleraciones de ambos sensores

Sensor 1			Sensor 2		
“acx”	“acy”	“acz”	“acx2”	“acy2”	“acz2”

En la Tabla 7, se presenta el listado de las variables, las cuales representan los valores del giroscopio de ambos sensores.

Tabla 7

Variables que representan las velocidades angulares de ambos sensores

Sensor 1			Sensor 2		
“gx”	“gy”	“gz”	“gx2”	“gy2”	“gz2”

La Tabla 8 presenta el conjunto de parámetros que facilitan la categorización de los datos recopilados.

Tabla 8

Parámetros de salidas

Parámetros de salidas				
parado	caminando	sentado	mov.derecha	mov.izquierda

Al termino de nombran las variables y los parámetros de salida, se procede a categorizar cada uno de los datos. Es decir, por ejemplo, para los datos adquiridos cuando la persona está caminando se marcan con un ‘1’, mientras que los otros parámetros se marcan con ‘0’. Los datos adquiridos se organizan en un archivo, como se muestra en la Figura 16. Este proceso de categorización es fundamental para analizar y comprender los datos recopilados y extraer información relevante. El número de filas que contenían los datos fue de 126,130.

Figura 16

Datos adquiridos y parametrizados

acx	acy	acz	gx	gy	gz	acx2	acy2	acz2	gx2	gy2	gz2	parado	caminando	sentado	mov.derecha	mov.izquierda
2.4	9.41	-0.03	3.41	-21.16	-6.12	2.4	9.44	0.02	4.91	-18.64	-5.95	0	1	0	0	0
0.93	9.7	-2.12	19.21	13.66	0.78	0.86	9.68	-2.1	18.51	12.68	0.7	0	1	0	0	0
2.11	9.68	-0.74	16.26	1.64	0.08	2.16	9.64	-0.73	16.39	3.22	0.3	0	1	0	0	0
0.9	9.35	-1.99	29.79	38.63	3.23	0.91	9.39	-2	30.46	37.54	3.29	0	1	0	0	0
0.75	10.22	-0.98	38.73	11.62	12.59	0.81	10.29	-0.95	38.77	12.26	13.09	0	1	0	0	0
1.18	11.58	-3.18	12.82	25.67	24.95	1.15	11.66	-3.2	10.89	24.01	26.15	0	1	0	0	0
4.48	10	0.04	-48.56	15.27	26.73	4.65	9.79	0.08	-51.77	15.67	27.51	0	1	0	0	0
2.61	11.57	-2.74	-138.61	47.86	18.22	2.45	11.58	-2.77	-140.35	43.86	17.29	0	1	0	0	0
2.47	9.63	-0.81	-183.58	-38.12	-24.95	2.21	9.55	-0.86	-184.72	-39.99	-22.8	0	1	0	0	0
3.43	7.43	1.07	-95.75	-19.05	-29.09	3.53	7.31	1.14	-93.22	-16.94	-30.53	0	1	0	0	0
2.27	5.64	0.62	5.9	-21.09	-28.39	2.15	5.65	0.89	6.87	-23.09	-27.7	0	1	0	0	0
6.04	9.32	7.61	49.88	66.37	-31.21	3.87	10.27	8.15	45.68	68.63	-30.35	0	1	0	0	0
1.52	13.64	3.79	-8.95	-38.65	-10.77	1.64	13.79	3.63	-6.86	-39.34	-11.38	0	1	0	0	0

3.2.2.2 Diseño de la red neuronal

g. Extracción de características

Los datos recopilados de los dos sensores comprenden un conjunto total de 12 valores (acx, acy, acz, acx2, acy2, acz2, gx, gy, gz, gx2, gy2, gz2), de las cuales solo se consideran 8 valores (acx, acy, acz, acx2, acy2, acz2, gx, gy), solo se empleará el giroscopio del primer sensor ubicado en la parte baja de la rodilla y, a su vez, únicamente se extrajo la característica del valor máximo a las dos señales de dicho giroscopio (gx, gy). La elección de utilizar el sensor uno se basa en su mayor capacidad de movimiento de giro. Se omite la extracción de características de 'gz' debido a su limitada participación en este eje.

También establecieron cinco condiciones de medición (parado, caminando, sentado, movimiento a la derecha y movimiento a la izquierda), que constituirán las variables de salida.

Con el propósito de realizar un procesamiento eficiente mediante la red neuronal, se llevará a cabo la extracción de características de las señales obtenidas. Estas características se emplearán como entradas para el diseño de la red neuronal. Las características son la media de la señal, la desviación estándar, el valor máximo, el sesgo, la curtosis y la energía.

La media de la señal representa el valor promedio de todos los datos en la señal. Se calcula sumando todos los valores individuales y dividiendo el resultado por el número total de datos en la señal. La fórmula para calcular la media en el caso de una señal discreta se expresa como la suma de todos los valores dividida por el número total de esos valores, donde x_i representa los valores individuales y N el número total de valores.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{7}$$

La Desviación estándar evalúa la consistencia o la amplitud de los datos en una señal. Un valor más alto de desviación estándar implica una mayor dispersión de los datos, mientras que un valor más bajo indica una concentración más cercana a la media. Se expresa como la raíz cuadrada de la suma de los cuadrados de las diferencias entre cada valor individual x_i y la media $\bar{x}$ al cuadrado, dividida por el número total de datos (N).

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}} \quad (8)$$

El valor máximo proporciona una indicación clara del punto más alto alcanzado por la señal, siendo un aspecto fundamental en el análisis y la interpretación de las características extremas de los datos.

El sesgo es una medida estadística que evalúa la asimetría de la distribución de datos en una señal. Indica si la distribución de los datos está sesgada hacia la izquierda, hacia la derecha o si es simétrica.

$$S = \frac{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{N}}{\left(\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}\right)^{3/2}} \quad (9)$$

La curtosis es una medida estadística que evalúa la forma de la distribución de datos en una señal. Se centra en cómo las colas de la distribución se comparan con las colas de una distribución normal.

$$K = \frac{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{N}}{\left(\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}\right)^{2}} - 3 \quad (10)$$

La energía de la señal es una medida que evalúa la cantidad total de energía presente en dicha señal.

$$E = \sum_{i=1}^{N} x_i^2 \quad (11)$$

Las seis características mencionadas se aplicarán a las seis aceleraciones extraídas de ambos sensores. Sin embargo, únicamente la característica del valor máximo se aplicará a las señales del giroscopio del primer sensor, específicamente en las coordenadas "x" e "y". En total, se obtendrán treinta y ocho entradas. Para calcular dichas características de las

señales mediante Python, se emplean distintos métodos. Para el cálculo del valor medio, se utiliza el método ‘. mean()', para la desviación estándar se recurre a '.std()', para el valor máximo se emplea '.max()', para el sesgo se utiliza la función 'skew', para la curtosis se emplea 'kurtosis' y para la energía se recurre al cálculo mediante la función 'energy'.

Las Figuras 17 y 18 muestran las características mencionadas mediante un formato de código de programación.

Figura 17

Extracción de características de las aceleraciones en 'x'

```
nuevacaract[j][0]=ventana['acx'].mean()
nuevacaract[j][3]=ventana['acx'].std()
nuevacaract[j][6]=ventana['acx'].max()
nuevacaract[j][9]=skew(ventana['acx'])
nuevacaract[j][12]=kurtosis(ventana['acx'])
nuevacaract[j][15]=energy(ventana['acx'])
```

Figura 18

Extracción de características de las velocidades angulares del primer sensor

```
nuevacaract[j][36]=ventana['gx'].max()
nuevacaract[j][37]=ventana['gy'].max()
```

h. Descripción de la red neuronal

La estructura de la red neuronal se determina en función de las condiciones de las características, que actúan como las entradas del modelo. El diseño consiste en una capa inicial con 38 características de entrada, seguida por una segunda capa con 36 neuronas. Posteriormente, se incorpora una tercera capa con 18 neuronas y, finalmente, se establecen 5 capas de salida. La información mencionada es evidente tanto en la Tabla 9 como en la Figura 19. La tabla presenta las cantidades de las capas del diseño, mientras que la figura ilustra el diagrama de la red neuronal diseñada.

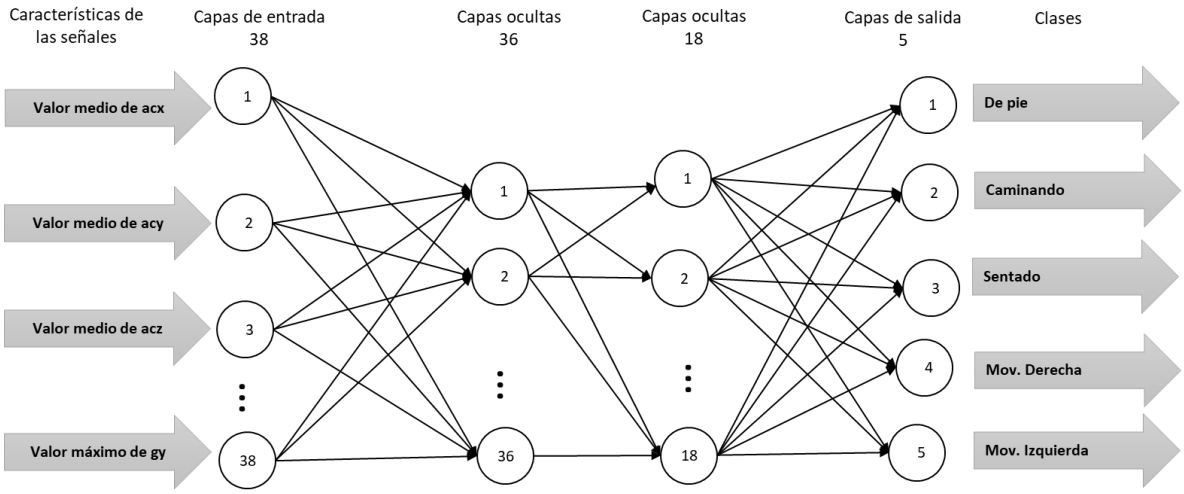
Tabla 9

Cantidad de capas de la red neuronal

Capas	Cantidad
Capa visible	38 entradas
Primera capa oculta	36 neuronas
Segunda capa oculta	18 entradas
Capa de salida	5 salidas

Figura 19

Esquema de la red neuronal diseñada



Para comenzar la creación de la red neuronal en código, se importan las bibliotecas esenciales, como NumPy para operaciones numéricas y Keras para la construcción de modelos de aprendizaje profundo.

La función de activación usada fue ReLU fue utilizada en las capas de entrada de la red ya que es computacionalmente eficiente y fácil de calcular, lo que acelera el proceso de entrenamiento, para la capa de salida se usó la función de activación softmax por que la capa de salida requiere una clasificación multiclase, es decir transforma las salidas en una

distribución de probabilidad, facilitando la interpretación de las predicciones como probabilidades

Una vez establecidas la cantidad de características y las funciones de activación a usar se procede a crear la red neuronal en Python, tal como se muestra en la Figura 20.

Figura 20

Código de creación del modelo de la red neuronal

```
model = Sequential()
model.add(Dense(38, input_dim=38, activation='relu'))
model.add(Dense(36, activation='relu'))
model.add(Dense(18, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

3.2.2.3 Entrenamiento de la red neuronal

Para el entrenamiento de la red se configuró el modelo de red creado. La configuración usa la entropía cruzada categórica como función de pérdida (loss function), el optimizador (optimizer) Adam para la optimización de los pesos, y la precisión (accuracy) como la métrica de evaluación del rendimiento del modelo.

La función de pérdida especificada fue la entropía cruzada categórica (categorical crossentropy). Se usó esta función de pérdida porque las etiquetas se han codificado en formato one-hot, es decir cada etiqueta se representa como un vector binario donde solo un elemento (la posición correspondiente a la clase real) es 1 y los demás son 0. En este caso, se espera que las salidas del modelo y las etiquetas sean distribuciones de probabilidad sobre las clases, y la entropía cruzada categórica mide la discrepancia entre estas distribuciones.

El optimizador 'Adam' fue elegido para ajustar los pesos de la red durante el proceso de entrenamiento ya que es un optimizador más utilizado que adapta la tasa de aprendizaje de manera dinámica para cada parámetro, lo que lo hace efectivo en una variedad de problemas de aprendizaje profundo.

La métrica usada para medir la precisión nos permite evaluara el rendimiento del modelo durante el proceso de entrenamiento y evaluación. Durante el entrenamiento, el modelo buscará mejorar esta métrica, ajustando los pesos de la red para maximizar la proporción de predicciones correctas. Además, al evaluar el modelo en datos de prueba o validación, la precisión se calculará para proporcionar una medida cuantitativa del

rendimiento del modelo en términos de la capacidad para clasificar correctamente nuevas instancias. La Figura 21 muestra la configuración del modelo de la red neuronal.

Figura 21

Configuración del modelo de la red neuronal

```
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

Entrenar a la red neuronal del conjunto de datos que se tiene, se optó por usa el método 'fit' el cual es un modelo de Keras. A este método se le asignaron cuatro parámetros, los cuales son la data de entrenamiento, el conjunto de datos de entrenamiento, el número de épocas y la proporción del conjunto de datos.

El código llevó a cabo el entrenamiento de un modelo utilizando el conjunto de datos '*training_data'* y sus respectivas etiquetas 'target_data'. El proceso de entrenamiento se extiende a lo largo de 120 épocas, y se destina el 10% de los datos como un conjunto de validación. Este conjunto de validación se utiliza para evaluar el rendimiento del modelo durante el entrenamiento, esto se llevó a cabo para ajustar los pesos del modelo a lo largo de estas épocas para mejorar su capacidad para realizar predicciones precisas en datos no vistos, contribuyendo así a la mejora del rendimiento del modelo en tareas futuras.

La Figura 22 muestra el código usado para entrenar la red.

Figura 22

Código de entrenamiento de la red neuronal

```
model.fit(training_data,
          target_data,
          epochs=120,
          validation_split=0.1)
```

Después de completar la creación y entrenamiento de la red, se procede a almacenar el modelo. Este proceso implica guardar la arquitectura de la red en un archivo JSON y los pesos en un archivo H5. El proceso consiste en guardar la arquitectura de una red neuronal en un archivo JSON, que describe sus características estructurales, y los pesos de la red en un archivo H5, un formato binario que almacena datos numéricos. Estos pesos, ajustables durante el entrenamiento, son esenciales para el funcionamiento de la red. Al preservar

ambos elementos, se facilita la replicación y utilización futura de la red sin necesidad de volver a entrenarla, lo que simplifica su compartición, implementación y reproducción. Esto permite determinar los movimientos, es decir, si los datos indican si la persona está sentada, de pie, en movimiento hacia la izquierda, o en movimiento hacia la derecha. Se muestra el código en la Figura 23.

Figura 23

Código para guardar una red neuronal creada

```
# Serializar el modelo a JSON
model_json=model.to_json()
with open("model.json", "w") as json_file:
  json_file.write(model_json)

# Serializar los pesos a HDF5
model.save_weights("model.h5")
print("Modelo Guardado!")
```

1.1.4. Representación gráfica

La representación gráfica fue llevada a cabo empleando el framework de Python llamado Django. El propósito fue ejecutar los archivos en formato JSON y H5 que albergan el modelo, para posteriormente compararlos con nuevos datos y exhibir las predicciones de acuerdo con el modelo previamente entrenado.

Para llevar a cabo la gráfica se procede a crear y activar el entorno virtual para tener un trabajo más optimo, este entorno albergará todos los procesos necesarios para visualizar las predicciones.

Se establece el proyecto llamado 'redneuronal', el cual engloba las configuraciones para una instancia de Django, abarcando aspectos como la configuración de la base de datos, opciones específicas de Django y configuraciones particulares de la aplicación. En la Figura 24 se observa el comando para crear el proyecto, mientras que en la Figura 25 muestra lo que se crea al iniciar el proyecto.

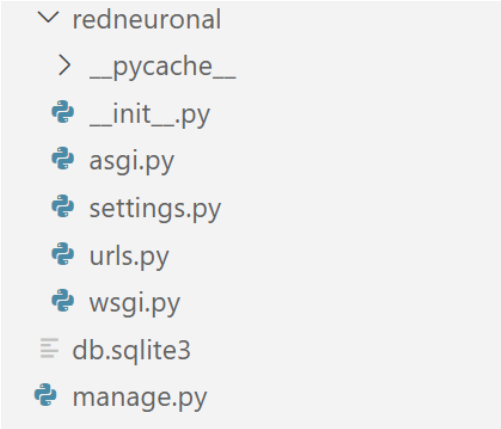
Figura 24

Comando para crear el proyecto

```
django-admin startproject redneuronal
```

Figura 25

Archivos creados al iniciar el proyecto



Estos archivos desempeñan roles específicos en la estructura y funcionamiento del proyecto Django.

Se procedió a ejecutar el servidor, en este caso será el servidor local. La figura 26 representa el comando para ello.

Figura 26

Ejecutar el servidor local

```
(entorno) PS C:\Users\dsdal\Documents\red_neuronal> python manage.py runserver 0.0.0.0:8000
```

Luego de finalizar la creación y ejecución del proyecto, se crea la aplicación en la cual podremos visualizar los gráficos necesarios, mediante el comando visualizado en la Figura 27. En la Figura 28 de muestra la aplicación creada.

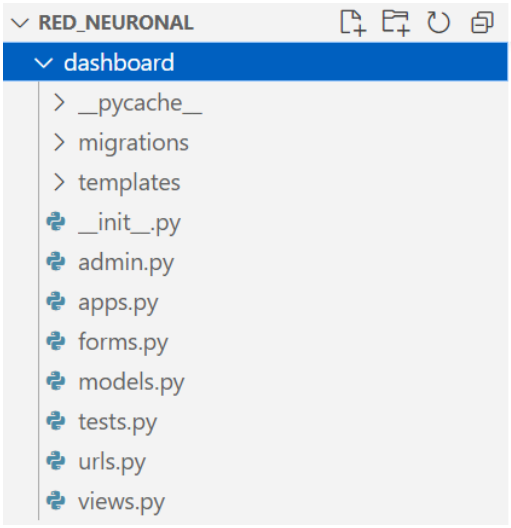
Figura 27

Comando para la creación de la aplicación

```
(entorno) PS C:\Users\dsdal\Documents\red_neuronal> python manage.py startapp dashboard
```

Figura 28

Aplicación dashboard

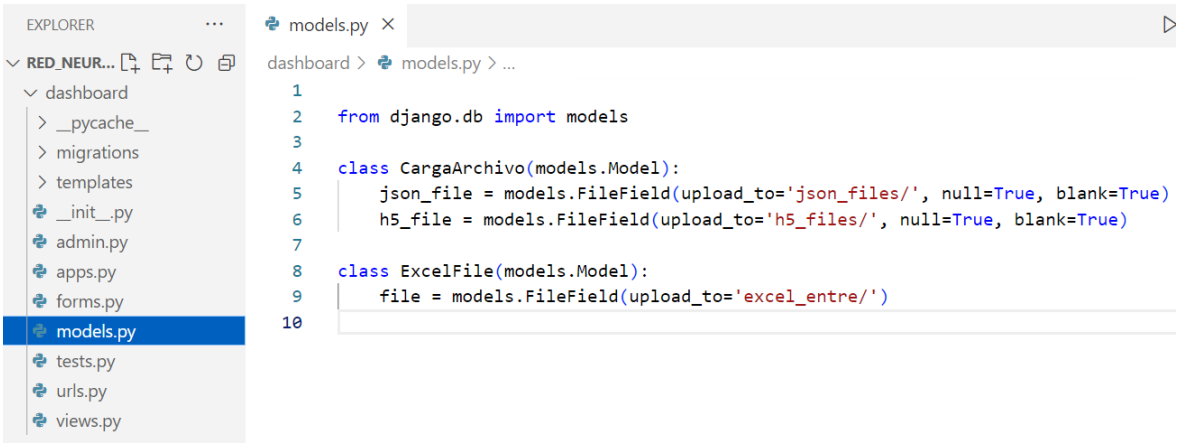


Al termino de crear la aplicación se procede a generar los modelos (models), las vistas (views), los formularios (forms) y las urls.

Los modelos actúan como la representación de tablas en la base de datos, definiendo la estructura de los datos y sus relaciones. En el contexto de la aplicación, se definen modelos para almacenar tanto los datos generados por el modelo principal como los archivos que se ingresarán para su comparación. La Figura 29 ilustra la creación de ambos modelos.

Figura 29

Modelos para la aplicación

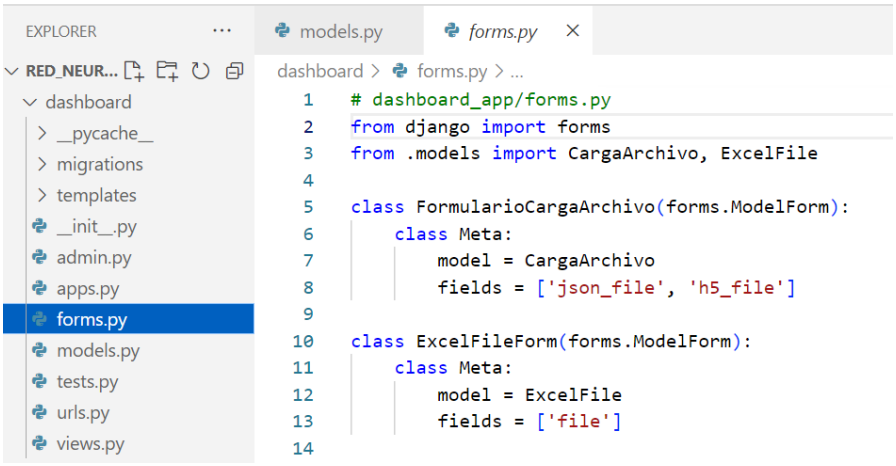


Los formularios gestionan la introducción de datos por parte del usuario en una aplicación web. Facilitaron la recolección, validación y procesamiento estructurado de datos, como en el caso de ingresar archivos en formato JSON, H5 y

un archivo Excel. En la Figura 30, se presenta el código correspondiente a ambos formularios, tanto para cargar el modelo como para los datos de entrenamiento.

Figura 30

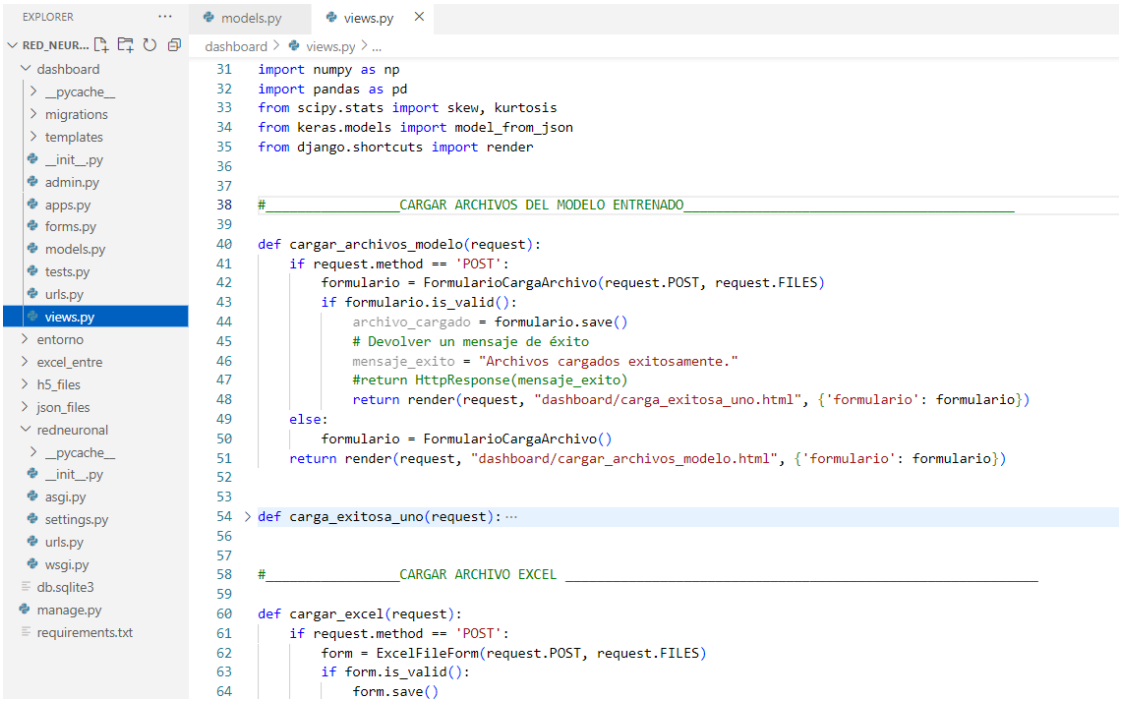
Formularios para la aplicación



Las vistas gestionan la lógica de presentación en una aplicación web. Su función principal es procesar la información recibida de la solicitud y devolver una respuesta adecuada, en este caso se muestra la respuesta en una página basada en HTML y CSS. Se muestra en la Figura 31 algunas de las vistas creadas.

Figura 31

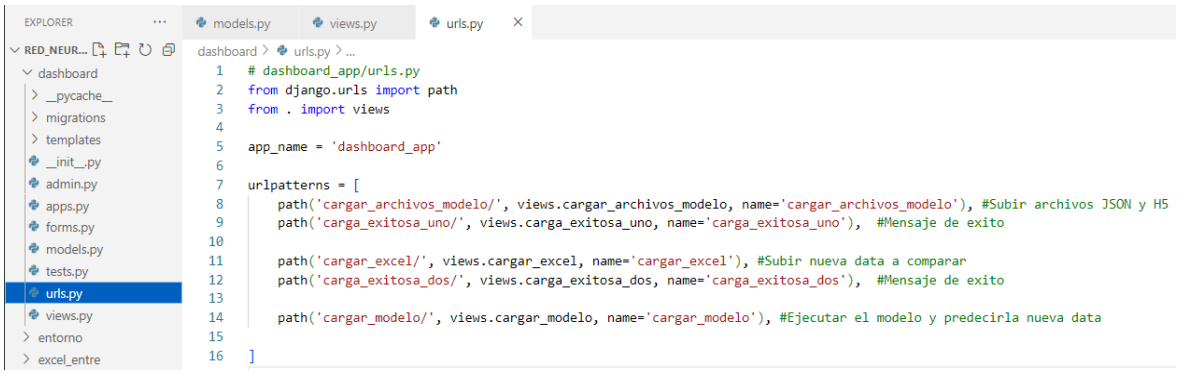
Vistas del modelo



Las urls son rutas que la aplicación necesita para poder navegar por la red. En la Figura 32 se muestran las rutas que la aplicación usó.

Figura 32

Rutas o urls de la aplicación



Concluyendo el desarrollo de la aplicación destinada a presentar el gráfico, las páginas resultantes ilustrarán las interfaces que servirán de punto de interacción con los usuarios. La Figura 33 exhibe la interfaz destinada a la carga de archivos relacionados con la red neuronal. Una vez que estos archivos se han cargado, la aplicación dirige al usuario a la página destinada a la carga del archivo para su comparación, presentada en la Figura 34. Al finalizar la carga, la aplicación redirige a la página encargada de la ejecución y visualización de las predicciones, mostrada en la Figura 35.

Figura 33

Interfaz para subir los archivos del modelo

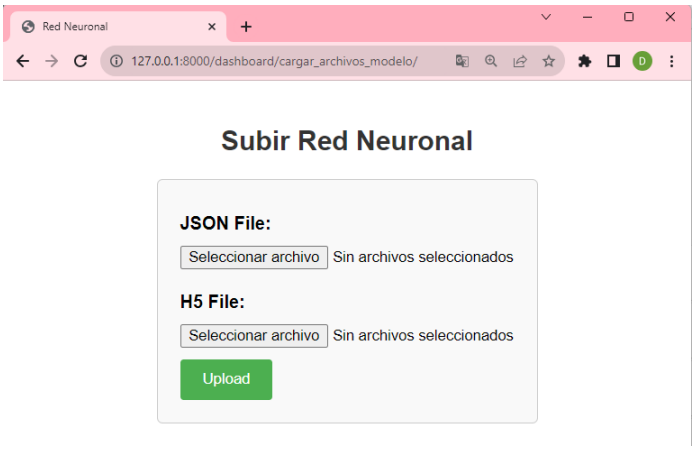


Figura 34

Interfaz para subir archivo Excel

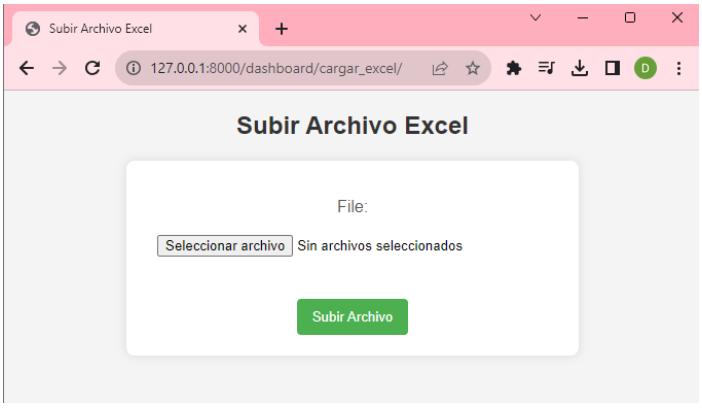
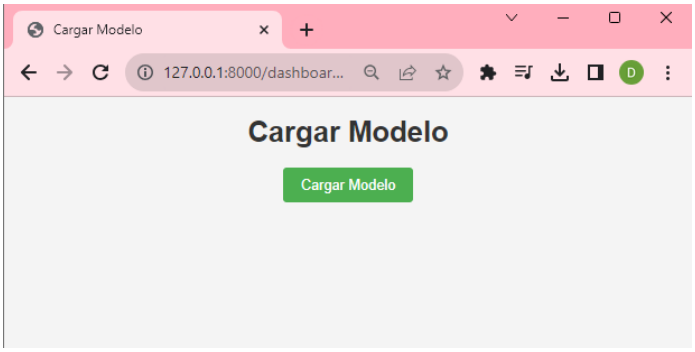


Figura 35

Interfaz para ejecutar la predicción del modelo



3.3 Resultados

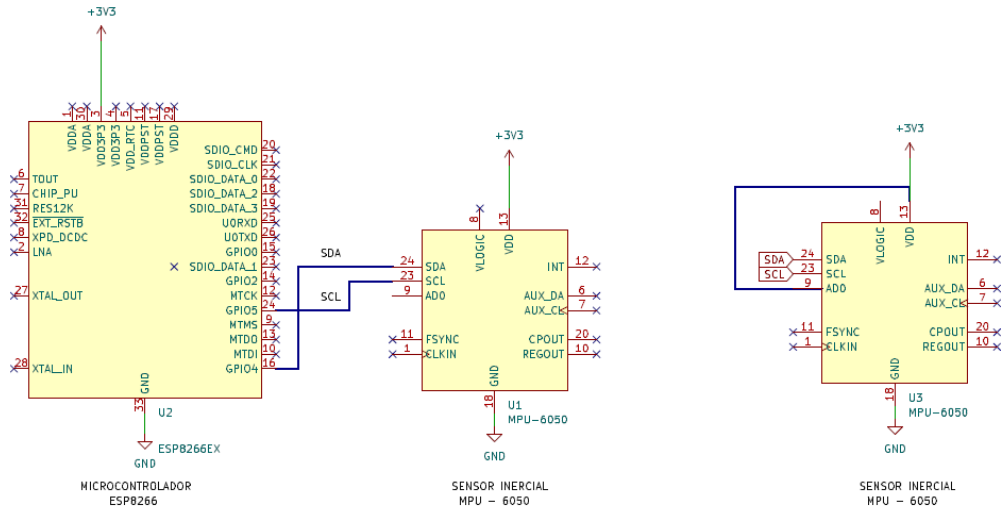
3.3.1 Hardware

a. Diseño del esquemático

El esquemático se desarrolló en Kicad, software para diseños electrónicos. En el esquemático se muestran la conexión de los dispositivos seleccionados, dos sensores inerciales MPU6050 y un microcontrolador ESP8266, tal como se aprecia en la Figura 36.

Figura 36

Esquemático

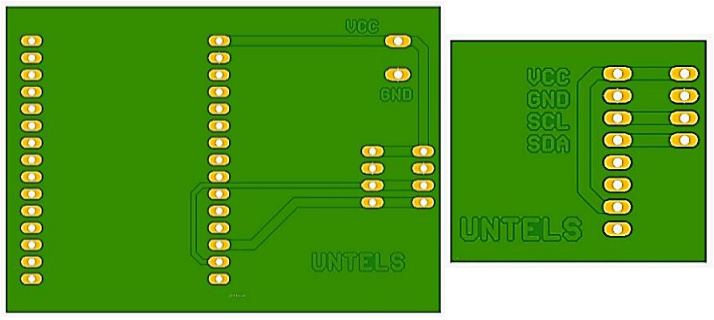


b. Diseño de la placa PCB

Aunque en el esquema se representan los componentes de forma conjunta, durante el proceso de diseño fue necesario separar las placas, dado que los sensores se ubican en distintas posiciones de la pierna. No obstante, se establecerá una conexión entre ellas mediante cables. En la Figura 37 se muestran los diseños finales de las placas.

Figura 37

Diseño final de placas



(a) (b)

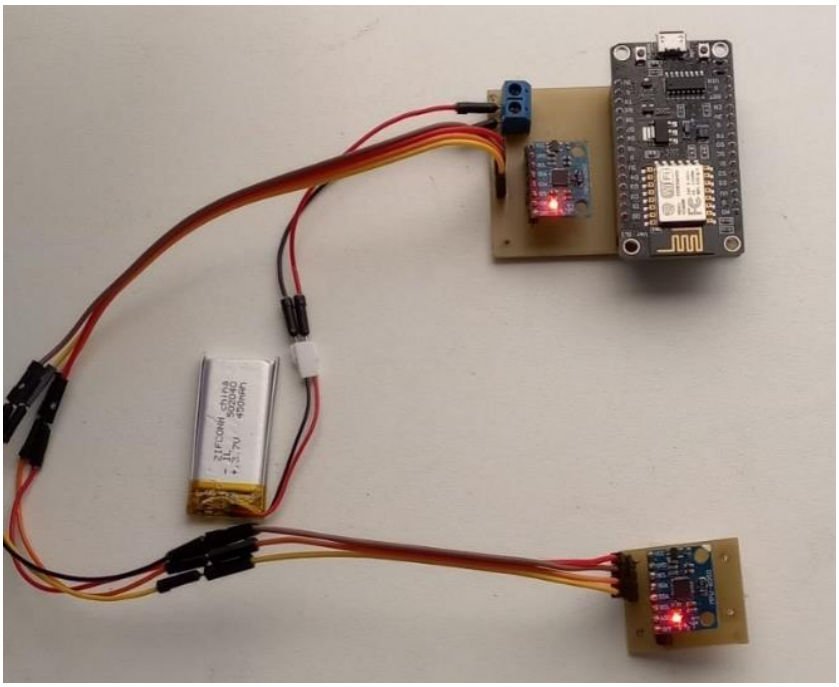
Nota. Se muestran dos placas, en la imagen (a) es el diseño de placa que aloja tanto el microcontrolador como el segundo sensor inercial, mientras que en la imagen (b) es el diseño de placa que aloja al primer sensor inercial, dicha placa estará situada en la parte inferior de la pierna.

c. Dispositivos colocados en la placa

Después de completar el diseño de las placas, se procede a fabricarlas para posteriormente ensamblar los dispositivos seleccionados. En la Figura 38, se observan los dos sensores inerciales y el microcontrolador ubicados en sus placas respectivas, conectadas a una batería de litio que suministra la energía necesaria para el funcionamiento de este dispositivo electrónico.

Figura 38

Diseño de dispositivo final



3.3.2 Red neuronal

Se comprobó la información de la red creada, llamada ‘model’, para verificar la creación tenga los valores especificados en la metodología, para ello se requirió de la función ‘.summary’, esta función un resumen detallado de la arquitectura del modelo de red neuronal creado. Se muestra en la Figura 39.

Figura 39

Función para mostrar un resumen de la red neuronal

```
model.summary()
```

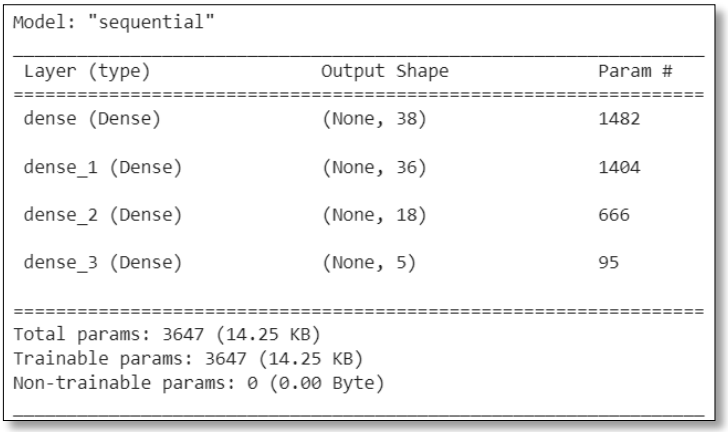
La estructura de la red neuronal se caracteriza por la presencia de capas identificadas por nombres y tipos, como las capas densas numeradas. Cada una de estas capas tiene una forma de salida específica, representada por sus dimensiones, y este patrón se mantiene de manera consistente a lo largo de las capas subsiguientes.

La sección de parámetros revela la cantidad de parámetros entrenables en cada capa, que son los valores que la red aprenderá durante el proceso de entrenamiento. El resumen total presenta el número completo de parámetros en la red, siendo 3647 en este caso. Todos estos parámetros son entrenables, lo que significa que la red tiene la capacidad de ajustarlos durante el entrenamiento para mejorar su rendimiento.

La información sobre parámetros no entrenables indica que, en este modelo específico, todos los parámetros son entrenables; no hay parámetros que permanezcan constantes durante el entrenamiento. Además, la última línea del resumen proporciona una perspectiva del tamaño total del modelo en términos de espacio en disco, indicando que la red con sus 3647 parámetros ocupa 14.25 kilobytes. Este resumen se muestra en la Figura 40 para comprender la estructura y la capacidad de aprendizaje de la red neuronal.

Figura 40

Resumen del modelo de la red neuronal



El resumen muestra que la cantidad de capas ingresadas son las mismas que se establecieron en la metodología.

La red neuronal creada está tiene definida la función de pérdida la cual evalúa qué tan lejos están las predicciones del modelo respecto a las etiquetas reales. También consta de un algoritmo de optimización que ajustará los pesos del modelo para minimizar esta

pérdida. Tiene especificada la métrica de precisión (accuracy) para evaluar el rendimiento del modelo, midiendo la proporción de predicciones correctas respecto al total.

Para el entrenamiento de 'model', el modelo creado, se usaron datos de entrenamiento y etiquetas, ajustando sus parámetros a lo largo de 120 épocas y reservando un 10% de los datos para validar el rendimiento del modelo.

El modelo se entrenó a lo largo de 120 época, la elección fue basada en experimentación y ajuste para lograr un equilibrio entre convergencia del modelo. Este número puede ser sintonizado durante la experimentación y validación del modelo en función de los resultados obtenidos. El uso de aproximadamente el 10% como conjunto de validación es una elección convencional común, pero esta fracción puede variar según la naturaleza específica del problema y el tamaño del conjunto de datos. La Tabla 10 muestra un resumen de los parámetros de la red mencionados.

Tabla 10

Parámetros de entrenamiento de la red neuronal diseñada

Parámetros	Valores	Descripción
Épocas o iteraciones	120	Cantidad de veces que el modelo recorrió el conjunto de datos durante el proceso de entrenamiento.
Validación	10 %	El conjunto de validación se utiliza para monitorear el rendimiento del modelo en datos no vistos durante el entrenamiento y evitar el sobreajuste.

La Figura 41 muestra el valor de la perdida y el valor de la precisión, obtenida del modelo diseñado.

Figura 41

Valores de perdida y precisión del modelo

Entonces para un modelo entrenado con 120 épocas y una validación de 0.1, obtenemos un valor de pérdida de 0.0829 y una precisión del 97.14% lo cual está próximo al 100%. Estos valores indican que es un entrenamiento optimo. La Tabla 11 detalla los resultados de los resultados de la evaluación del modelo.

Tabla 11

Resultados de la evaluación del modelo

Resultados	Valores	Descripción
Perdida	0.0829	El valor bajo de pérdida obtenido indica que el modelo está haciendo predicciones cercanas a las etiquetas reales.
Precisión	97.14 %	Un valor alto de precisión es deseado, lo más cercano al 100%.

Los datos crudos obtenidos y categorizados fueron una cantidad de 126 130, los cuales fueron divididos entre el taño de ventana, el cual fue 40, dándonos un valor de 3 150 datos a entrenar, los cuales se ubican en una matriz. La matriz que contiene la extracción de características de los datos, consta de 38 columnas, las cuales contienen una cantidad de 3 150 muestras para el entrenamiento. Dichas muestras contienen el valor obtenido de la extracción de características como valores medios, valores máximos, desviación, y la energía. Los datos de las muestras se visualizan en la Tabla 12.

Tabla 12

Datos para el entrenamiento del modelo

Datos	Cantidad
Fila de datos crudos ingresados	126 130
Tamaño de ventana	40

Cantidad de ventanas	(126 130 / 40) – 1 ≅ 3 152
Cantidad de parámetros	38
Tamaño de matriz	[3 152 x 38]

Al tener una precisión al 97.14% se procede a realizar predicciones utilizando la red neuronal creada. Para la predicción del modelo se utilizaron un conjunto de datos nuevos, para ser comparados con la red neuronal y así poder predecir la característica de la señal con respecto a las ventanas evaluadas.

Los resultados se visualizan mediante gráficos de barras para cada muestra. Los elementos necesarios para los gráficos incluyen las clases, las predicciones, los nombres de las clases, las etiquetas y el título.

Las clases a mostrar son 'Parado', 'Caminando', 'Sentado', 'Mov. Derecha', 'Mov. Izquierda', y la cantidad de predicciones se ajusta según el número de filas de los datos, correspondiente al número de ventanas. Se establecen etiquetas para los ejes ‘x’, ‘y’ con 'Clases' y 'Probabilidades'.

Se mostraron diferentes graficas para cada uno de la clase de movimientos, es decir para la clase caminando, la clase sentado, la clase parado, el mov derecha, el mov. Izquierda.

- TODAS LAS CLASES

Para datos que contienen todos los movimientos tuvieron la siguiente cantidad de datos, los cuales de detallan en la Tabla 13.

Tabla 13

Datos para evaluar

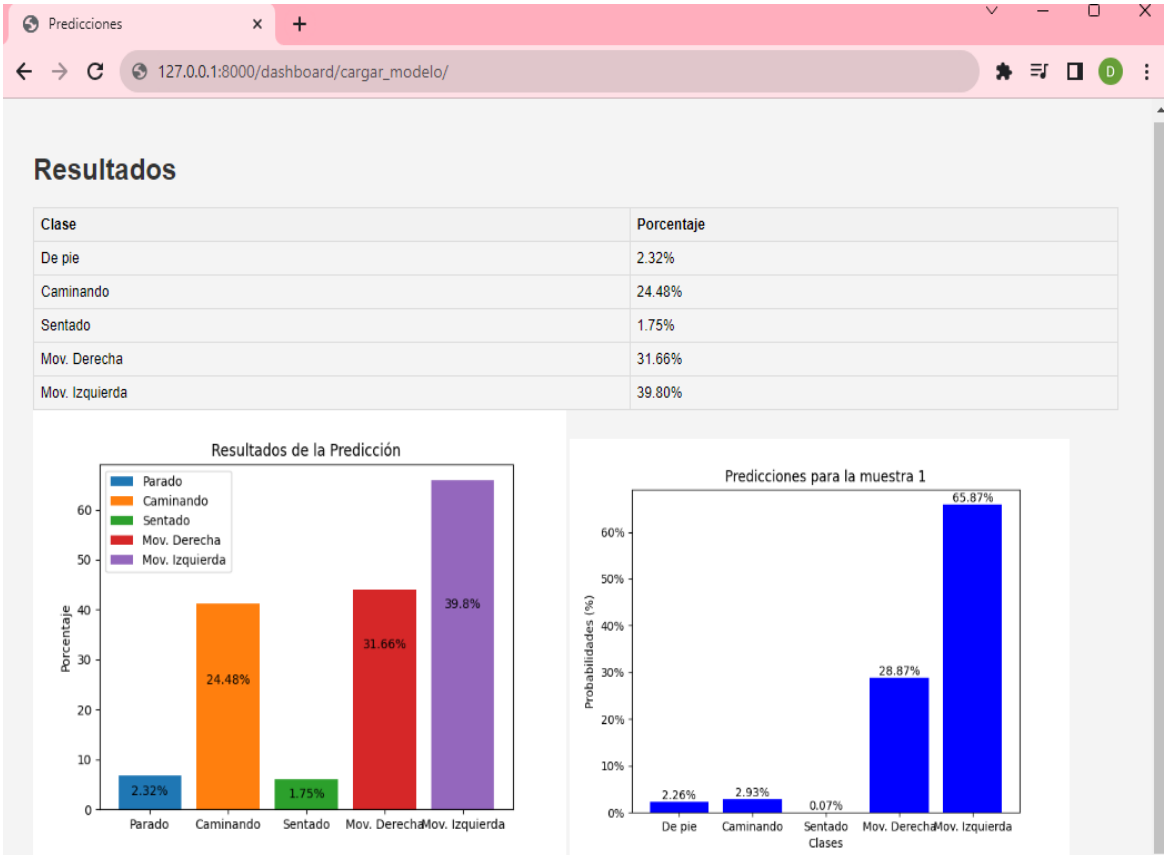
Datos	Cantidad
Fila de datos crudos ingresados	8 483
Tamaño de ventana	40
Cantidad de ventanas	(8 483/ 40) – 1 ≅ 211

Cantidad de parámetros	38
Tamaño de matriz	[211 x 38]

El grafico obtenido para los datos mencionado en la Tabla 12 se muestran la Figura 42.

Figura 42

Predicción para datos aleatorios del movimiento



(i) (ii)

Nota. Se muestra una tabla la cual muestra el valor de predicción de los datos ingresados para el total de los datos. En la sección (i) se muestran los mismos valores de la tabla en forma de barras, para una mejor visualización. En la sección (ii) muestra las predicciones, pero para una de las ventanas.

La predicción representada en la Figura 46, en la sección (ii), indica un porcentaje significativamente alto del 65.87% para la clase "Mov. Izquierda", para la muestra 1, es decir para la primera ventana. Sin embargo, también exhibe porcentajes menores para otras clases, con un 2.26% para "De Pie", un 2.93% para "Caminando", un 0.07% para "Sentado" y un 28.87% para "Mov. Derecha". Este resultado sugiere que la persona está realizando una actividad de caminar mientras ejecuta un desplazamiento hacia la izquierda, lo cual no es un movimiento convencional. Por ende, esta observación podría interpretarse como una anomalía en el patrón típico de caminar.

- CLASE CAMINANDO

Para mostrar el grafico de la clase caminando se compararon con los datos establecidos en la Tabla 14, mientras que su grafico se denota en la Figura 43.

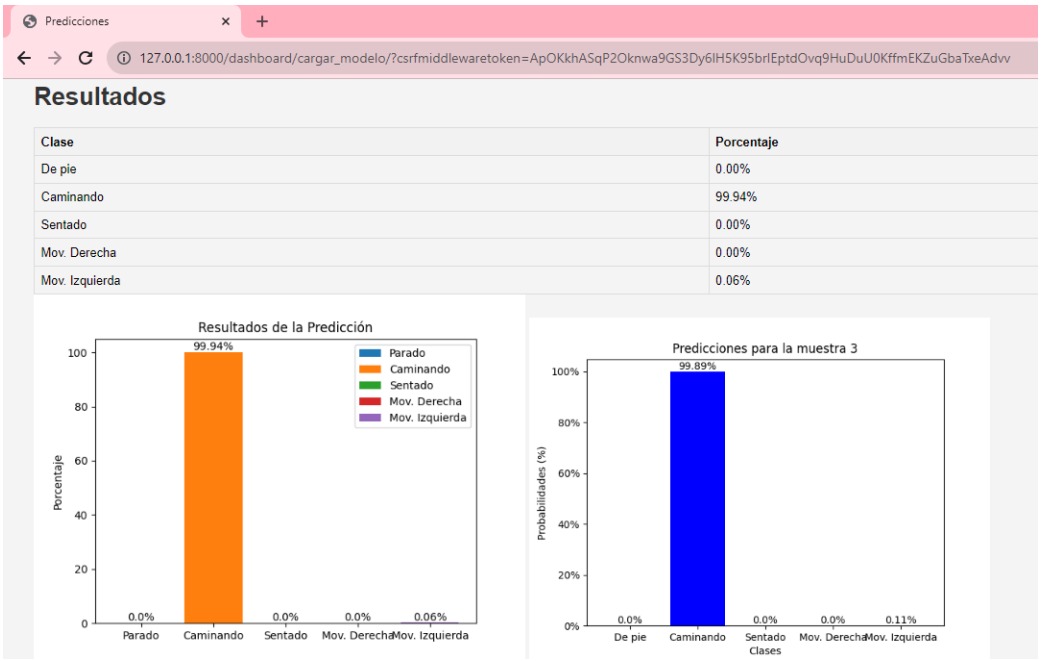
Tabla 14

Datos de entrenamiento para clase Caminando

Datos	Cantidad
Fila de datos crudos ingresados	1 265
Tamaño de ventana	40
Cantidad de ventanas	(1 265/ 40) – 1 ≅ 30
Cantidad de parámetros	38
Tamaño de matriz	[30 x 38]

Figura 43

Predicción para datos para clase Caminando



(i) (ii)

Nota. Se muestra una tabla la cual muestra el valor de predicción de los datos ingresados para el total de los datos. En la sección (i) se muestran los mismos valores de la tabla en forma de barras, para una mejor visualización. En la sección (ii) muestra las predicciones, pero para la ventana 3.

La predicción representada en la Figura 47, en la sección (ii), indica un porcentaje significativamente alto del 99.89% para la clase "Caminando", para la muestra 3, es decir para la tercera ventana. Sin embargo, también exhibe porcentajes menores para otras clases, con un 0.11% para "Mov. Izquierda", mientras que los otros movimientos se encuentran por debajo del 0%. Este resultado sugiere que la persona está realizando una actividad de caminar.

IV. DISCUSIÓN DE RESULTADOS

La discusión de los resultados se centra en la evaluación del algoritmo basado en redes neuronales desarrollado para la predicción de parámetros de la marcha. En comparación con el estudio de Nur-Masruriyah et al. (2019), que se enfocó en la predicción de accidentes cerebrovasculares en pacientes diabéticos, se observa una diferencia en las tasas de predicción. Mientras que el estudio previo logró una precisión del 95.15%, el algoritmo desarrollado en la presente tesis alcanzó una precisión del 97.14%. Aunque esta cifra se sitúa ligeramente por debajo de las expectativas teóricas, aún se considera aceptable y dentro de rangos aceptables.

En relación con el error, que se esperaba cercano a 0 según el marco teórico, el valor calculado fue de 0.0829. Aunque no alcanza el ideal teórico, este resultado sugiere que el modelo tiene una capacidad adecuada para generalizar y predecir patrones con precisión.

Una adecuada capacidad de la red neuronal para generalizar y predecir patrones con precisión. Estos resultados respaldan la eficacia del modelo desarrollado, aunque también sugieren áreas donde se podría buscar mejoras para optimizar aún más su rendimiento.

V. CONCLUSIONES

En conclusión, este trabajo de investigación logró el desarrollo de un sistema electrónico con el objetivo central de estimar los movimientos de la marcha del cuerpo humano. Este sistema es capaz de detectar y analizar movimientos como estar de pie, caminar, sentarse, así como movimientos hacia la derecha o izquierda.

La implementación del sistema electrónico permitió la captura de datos de movimiento durante la marcha. Se llevó a cabo una selección de dispositivos para el desarrollo del sistema, optando finalmente por dos sensores inerciales y un microcontrolador. Este proceso de elección implicó la comparación de diversos dispositivos, garantizando así una selección adecuada y eficiente. Además, se desarrollaron placas de circuito impreso para la disposición de los dispositivos, contribuyendo a un sistema más estable durante la extracción de datos.

La estructura de la red neuronal ha sido diseñada considerando las características del modelo, incluyendo una capa inicial para las características de entrada, seguida de capas intermedias y una capa de salida. Esta disposición establece una base para la representación de datos y la toma de decisiones, y se sugiere realizar pruebas con diferentes cantidades de capas para mejorar la precisión del modelo. Durante el proceso de entrenamiento, se configuró el modelo función de pérdida y el optimizador. Este procedimiento contribuye a refinar la red neuronal para realizar predicciones precisas en datos no vistos, demostrando así la efectividad y robustez del enfoque implementado en este estudio. Se recomienda optimizar la función de pérdida para acercarla lo más posible a 0 y tener una precisión cercana al 100%.

La evaluación del dashboard para redes neuronales es efectivo para la representación gráfica de los resultados de predicciones. Los gráficos, especialmente las barras que ilustran las probabilidades por clase o movimientos, facilitan la interpretación de la complejidad de los datos, permitiendo a los usuarios identificar rápidamente las categorías con mayor certeza.

VI. REFERENCIAS BIBLIOGRÁFICAS

Acevedo, P. E. (2020). *Plataforma de fuerza para terapia de equilibrio (Doctoral dissertation, Universidad de Talca. Facultad de Ingeniería).*

Alpaydin, E. (2014). *Introduction to Machine Learning.*

Cerda, A. L. (2014). *Manejo del trastorno de marcha del adulto mayor. Revista Médica Clínica Las Condes, 265-275.*https://doi.org/10.1016/s0716-8640(14)70037-9

Claudio, I. R. (2012). *Entrenamiento robótico como medio de rehabilitación para la marcha. Evidencia Médica e Investigación En Salud, 5(2), 46-54.*

Finn, A. (12 de julio 2023). *El giroscopio.* Disponible en: http://www.sc.ehu.es/sbweb/fisica_/solido/rotacion/giroscopo/giroscopo.html.

Gomez, K. (2021) *Diseño de un dispositivo para la rehabilitación pasiva en pacientes con diagnósticos de lumbalgia aguda*. http://hdl.handle.net/20.500.12404/20876

Gujarathi, T., y Bhole, K. (2019,). *Gait analysis using imu sensor. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE*. https://doi.org/10.1109/icccnt45670.2019.8944545

Jacob, A., Zakaria, W. y Mohd, R. (2016). *Implementation of IMU sensor for elbow movement measurement of Badminton players*. https://doi.org/10.1109/roma.2016.7847813

Jesús, F. R. (2021). *Rehabilitación de la marcha en el accidente cerebrovascular. Sistemas robotizados.* http://hdl.handle.net/10347/27766

Luna, J. (2017). *Sistema electrónico para el análisis de la marcha humana en el plano sagital. Universidad del Valle.* https://bibliotecadigital.univalle.edu.co/handle/10893/15812

Luna, J. (2019). *Desarrollo de un sistema de adquisición para la evaluación del balance corporal en base a la medida del movimiento del tronco.* http://hdl.handle.net/20.500.12404/15817

Martín Bragado, I. (2003). *Física General*. *https://www.ucm.es/data/cont/docs/18-2020-04-15-Ignacio-martin-bragado.pdf*

Matarollo, V. (2013). *Applications of artificial neural networks in chemical problems.* https://doi.org/10.5772/51275

Nur-Masruriyah, A., Djatna, T., Dewi, M., Handayani, H., & Wahiddin, D. (2019). "*Predictive Analytics For Stroke Disease [Sesión de conferencia]"*.Fourth International Conference on Informatics and Computing (ICIC). Semarang, Indonesia. doi: 10.1109/ICIC47613.2019.8985716

Olascoaga, R. y Ascue , S. (2021). "*Development of an algorithm with neural networks for the prediction of ACV in diabetic patients*". Universidad Autónoma del Perú. https://hdl.handle.net/20.500.13067/1522

Oppenheim, A. (2011). *Tratamiento de señales en tiempo discreto (Tercera edición.). Pearson Educación.*

Organización Mundial de la Salud: OMS. (2022). *Trastornos musculoesqueléticos*. https://www.who.int/es/news-room/fact-sheets/detail/musculoskeletal-conditions

Python.org. (2023, 06 de agosto). *Python.org*. https://www.python.org/

Qiu, S., Liu, L., Wang, Z. Shengming, L., Zhao, H., Wang, J., Li, J. y Tang, K.(2019). *Body Sensor Network-Based Gait Quality Assessment for Clinical Decision-Support via Multi-Sensor Fusion. IEEE Access, 7, 59884-59894.* https://doi.org/10.1109/access.2019.2913897

Rodríguez, P. (2017). *Protocolo de evaluación de un sistema para medición de parámetros de tiempo de la marcha humana. ResearchGate.* https://www.researchgate.net/publication/321492779_Protocolo_de_Evaluacion_de_un_Sistema_para_Medicion_de_Parametros_de_Tiempo_de_la_Marcha_Humana

SuperDataScience Team. (2018). The Ultimate Guide to Convolutional Neural Networks (CNN). Recuperado de https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

Tribeño, E. (2021). *Diseño de un dispositivo portátil para la rehabilitación pasiva de muñeca con tres grados de libertad*. http://hdl.handle.net/20.500.12404/21169

Valcarce, A. (2014). *Fisica: Momento de Inercia y Aceleracion Angular. Pontificia Universidad Catolica de Chile.* Disponible en: https://www.astro.puc.cl/~avalcarc/FIS109A/17_MomentoInercia.pdf

Winter, D.A. (2009*). Biomechanics and Motor Control of Human Movement. 4th Edition, John Wiley & Sons, Hoboken.* http://dx.doi.org/10.1002/9780470549148

Zurada, J. M. (2005). *Introduction to Artificial Neural Systems*.

ANEXOS

Anexo 1. Matriz de consistencia

TESIS: “IMPLEMENTACIÓN DE SISTEMA ELECTRÓNICO PARA LA MEDICIÓN Y ANÁLISIS DE LA MARCHA MEDIANTE ACELEROMETROS Y GIROSCOPOS”

PLANTEAMIENTO DEL PROBLEMA	OBJETIVOS DE LA INVESTIGACIÓN	METODOLOGÍA DE LA INVESTIGACIÓN
PROBLEMA GENERAL: La escasez de sistemas electrónicos adecuados basados en sensores que adquieran información de la zona en estudio (pierna), para el monitoreo de la marcha en personas con problemas de movilidad. **PROBLEMAS ESPECÍFICOS:** • ¿Cómo implementar el circuito electrónico para la adquisición del movimiento de la marcha en personas adultas mayores? • ¿Cómo se puede desarrollar un algoritmo basado en redes neuronales para evaluar la marcha y detectar anomalías? • ¿Cómo se puede desarrollar una aplicación que represente de manera gráfica los datos recolectados en relación a las anomalías de la marcha?	**OBJETIVO GENERAL:** Implementar un sistema electrónico basado en acelerómetros y giróscopos para el monitoreo de la marcha en personas con problemas de movilidad. **OBJETIVOS ESPECÍFICOS:** • Implementar un sistema electrónico embebido inalámbrico para la adquisición del movimiento de la marcha en personas adultas. • Desarrollar de un algoritmo basado en redes neuronales para evaluar la marcha y detectar anomalías. • Desarrollar una aplicación que permita la representación gráfica de los datos obtenidos de la marcha.	**TIPO DE INVESTIGACIÓN:** Cuantitativa. **NIVEL DE INVESTIGACIÓN:** Experimental, Descriptiva, Aplicada. **METODO:** • Implementación de circuito electrónico • Procesamiento de datos de los sensores. • Representación gráfica de los datos.

ANEXO 2. Glosario de términos.

Acelerómetro: Sensor que mide la aceleración lineal de un objeto en una dirección especifica.

Cadencia: El número de pasos dados por una persona en un período de tiempo determinado. Se refiere a la frecuencia o la cantidad de pasos que una persona da por minuto mientras camina o corre. Expresado en pasos por minuto (ppm).

Chip: Pieza pequeña la cual contiene circuitos electrónicos y componentes miniaturizado, también conocido como circuito integrado.

Ciclo de Marcha: El ciclo completo de movimiento de una pierna, desde el contacto inicial del talón con el suelo hasta el siguiente contacto del mismo talón.

Desplazamiento: Cambio neto en la posición de un objeto desde su punto inicial al punto final.

EMG (Electromiografía): Técnica de diagnóstico médico que se utiliza para evaluar y registrar la actividad eléctrica de los músculos.

Fase de Apoyo: La parte del ciclo de marcha en la que el pie está en contacto con el suelo, dividida en la fase de apoyo inicial y la fase de apoyo terminal.

Fase de Balanceo: La parte del ciclo de marcha en la que el pie está en el aire y se mueve hacia adelante para prepararse para el siguiente paso.

Giroscopio: Sensor encargado de medir la velocidad angular o tasa de rotación de un objeto alrededor de un eje especifico.

IMU (Inertial Measurement): Unidad de Medición Inercial Conjunto de sensores que incluye acelerómetros, giroscopios y, en algunos casos, magnetómetros, utilizados para obtener datos sobre la orientación, posición y movimiento de un objeto.

IoT (Internet Of Things): Conocido en español como Internet de las cosas es una red de dispositivos físicos y objetos pues están conectados entre sí mediante internet, permitiendo así la recopilación y transmisión de datos para la interacción y la toma de decisiones automáticas o remotas.

Marcha Humana: La forma en que los seres humanos se desplazan de un lugar a otro, caracterizada por el movimiento rítmico y coordinado de las extremidades.

MEMS (Micro Electro Mechanical System): Los sistemas microelectromecánicos en español, se refiere a la tecnología que combina elementos mecánicos y electrónicos.

Microcontrolador: Circuito integrado que incluye un procesador, memoria y periféricos en un solo paquete, diseñado para controlar dispositivos y sistemas.

MPU (Multiple Process Unit): La unidad de procesamiento central se compone principal de dispositivos electrónicos, realiza operaciones de procesamiento y control.

OMS (Organización Mundial de la Salud): Dirige asuntos relacionados con la salud a nivel mundial, tiene su sede en Ginebra, Suiza.

RAM (Random Access memory): La memoria de acceso aleatorio es un tipo de memoria temporal, almacena y accede rápidamente a datos que están siendo utilizados activamente por el sistema operativo y las aplicaciones en ejecución, es una memoria volátil ya que su contenido se borra cuando se apaga la energía.

RNA (Redes Neuronales Artificiales): Modelo de inteligencia artificial inspirados en el funcionamiento del cerebro humano, se componen por nodos interconectados que trabajan en conjunto para procesar información y resolver problemas complejos.

ROM (Read Only Memory): Tipo de memoria donde los datos se escriben durante su fabricación y no pueden modificarse ni borrarse posteriormente, memoria de solo lectura. Se utiliza para almacenar programas y datos fundamentales para el funcionamiento del sistema, como firmware y configuración básica del hardware.

ANEXO 3. Hoja de datos del sensor inercial MPU6050

6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/*g*	
SELF-TEST RESPONSE Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	**FS_SEL=0**					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/ √ Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE	Programmable Range	5		256	Hz	
OUTPUT DATA RATE	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME ZRO Settling (from power-on)	**DLPFCFG=0** to ±1°/s of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		*g*	
	AFS_SEL=1		±4		*g*	
	AFS_SEL=2		±8		*g*	
	AFS_SEL=3		±16		*g*	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/*g*	
	AFS_SEL=1		8,192		LSB/*g*	
	AFS_SEL=2		4,096		LSB/*g*	
	AFS_SEL=3		2,048		LSB/*g*	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		*mg*	1
	Z axis		±80		*mg*	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		*mg*	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µg/ √Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		mg/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

6.4 Electrical Specifications, Continued

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
SERIAL INTERFACE						
SPI Operating Frequency, All Registers Read/Write	MPU-6000 only, Low Speed Characterization		100 ±10%		kHz	
	MPU-6000 only, High Speed Characterization		1 ±10%		MHz	
SPI Operating Frequency, Sensor and Interrupt Registers Read Only	MPU-6000 only		20 ±10%		MHz	
I^2C Operating Frequency	All registers, Fast-mode			400	kHz	
	All registers, Standard-mode			100	kHz	
I^2C ADDRESS	AD0 = 0		1101000			
	AD0 = 1		1101001			
DIGITAL INPUTS (SDI/SDA, AD0, SCLK/SCL, FSYNC, /CS, CLKIN)						
V_{IH}, High Level Input Voltage	MPU-6000	0.7*VDD			V	
	MPU-6050	0.7*VLOGIC			V	
V_{IL}, Low Level Input Voltage	MPU-6000			0.3*VDD	V	
	MPU-6050			0.3*VLOGIC	V	
C_I, Input Capacitance			< 5		pF	
DIGITAL OUTPUT (SDO, INT)						
V_{OH}, High Level Output Voltage	R_{LOAD}=1MΩ; MPU-6000	0.9*VDD			V	
	R_{LOAD}=1MΩ; MPU-6050	0.9*VLOGIC			V	
V_{OL1}, LOW-Level Output Voltage	R_{LOAD}=1MΩ; MPU-6000			0.1*VDD	V	
	R_{LOAD}=1MΩ; MPU-6050			0.1*VLOGIC	V	
$V_{OL.INT1}$, INT Low-Level Output Voltage	OPEN=1, 0.3mA sink Current			0.1	V	
Output Leakage Current	OPEN=1		100		nA	
t_{INT}, INT Pulse Width	LATCH_INT_EN=0		50		µs	
DIGITAL OUTPUT (CLKOUT)						
V_{OH}, High Level Output Voltage	R_{LOAD}=1MΩ	0.9*VDD			V	
V_{OL1}, LOW-Level Output Voltage	R_{LOAD}=1MΩ			0.1*VDD	V	

ANEXO 4. Hoja de datos del microcontrolador ESP8266

1.2. Specifications

Table 1-1. Specifications

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: –91 dbm (11 Mbps)
		802.11 g: –75 dbm (54 Mbps)
		802.11 n: –72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5 V ~ 3.6 V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	–40 °C ~ 125 °C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

📖 ***Note:***

The TX power can be configured based on the actual user scenarios.

ANEXO 5. Código de calibración del sensor inercial MPU6050

```
// Librerías I2C para controlar el mpu6050
// la librería MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito
MPU6050 sensor;

// Valores RAW (sin procesar) del acelerómetro y giroscopio en los ejes x,y,z
int16_t ax, ay, az;
int16_t gx, gy, gz;

//Variables usadas por el filtro pasa bajos
long f_ax,f_ay, f_az;
int p_ax, p_ay, p_az;
long f_gx,f_gy, f_gz;
int p_gx, p_gy, p_gz;
int counter=0;

//Valor de los offsets
int16_t ax_o,ay_o,az_o;
int16_t gx_o,gy_o,gz_o;

void setup() {
  Serial.begin(57600);   //Iniciando puerto serial
  Wire.begin();           //Iniciando I2C
  sensor.initialize();    //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");

  // Leer los offset los offsets anteriores
  ax_o=sensor.getXAccelOffset();
  ay_o=sensor.getYAccelOffset();
  az_o=sensor.getZAccelOffset();
  gx_o=sensor.getXGyroOffset();
  gy_o=sensor.getYGyroOffset();
  gz_o=sensor.getZGyroOffset();

  Serial.println("Offsets:");
  Serial.print(ax_o); Serial.print("\t");
  Serial.print(ay_o); Serial.print("\t");
  Serial.print(az_o); Serial.print("\t");
  Serial.print(gx_o); Serial.print("\t");
  Serial.print(gy_o); Serial.print("\t");
  Serial.print(gz_o); Serial.print("\t");
  Serial.println("Envié cualquier carácter para empezar la calibración");
  // Espera un carácter para empezar a calibrar
  while (true){if (Serial.available()) break;}
  Serial.println("Calibrando, no mover IMU");
}

void loop() {
```

```
55   // Leer las aceleraciones y velocidades angulares
56   sensor.getAcceleration(&ax, &ay, &az);
57   sensor.getRotation(&gx, &gy, &gz);
58
59   // Filtrar las lecturas
60   f_ax = f_ax-(f_ax>>5)+ax;
61   p_ax = f_ax>>5;
62   f_ay = f_ay-(f_ay>>5)+ay;
63   p_ay = f_ay>>5;
64   f_az = f_az-(f_az>>5)+az;
65   p_az = f_az>>5;
66   f_gx = f_gx-(f_gx>>3)+gx;
67   p_gx = f_gx>>3;
68   f_gy = f_gy-(f_gy>>3)+gy;
69   p_gy = f_gy>>3;
70   f_gz = f_gz-(f_gz>>3)+gz;
71   p_gz = f_gz>>3;
72
73   //Cada 100 lecturas corregir el offset
74   if (counter==100){
75     //Mostrar las lecturas separadas por un [tab]
76     Serial.print("promedio:"); Serial.print("\t");
77     Serial.print(p_ax); Serial.print("\t");
78     Serial.print(p_ay); Serial.print("\t");
79     Serial.print(p_az); Serial.print("\t");
80     Serial.print(p_gx); Serial.print("\t");
81     Serial.print(p_gy); Serial.print("\t");
82     Serial.println(p_gz);
83
84     //Calibrar el acelerómetro a 1g en el eje z (ajustar el offset)
85     if (p_ax>0) ax_o--;
86     else {ax_o++;}
87     if (p_ay>0) ay_o--;
88     else {ay_o++;}
89     if (p_az-16384>0) az_o--;
90     else {az_o++;}
91
92     sensor.setXAccelOffset(ax_o);
93     sensor.setYAccelOffset(ay_o);
94     sensor.setZAccelOffset(az_o);
95
96     //Calibrar el giroscopio a 0°/s en todos los ejes (ajustar el offset)
97     if (p_gx>0) gx_o--;
98     else {gx_o++;}
99     if (p_gy>0) gy_o--;
100    else {gy_o++;}
101    if (p_gz>0) gz_o--;
102    else {gz_o++;}
103
104    sensor.setXGyroOffset(gx_o);
105    sensor.setYGyroOffset(gy_o);
106    sensor.setZGyroOffset(gz_o);
107    counter=0;
108  }
109  counter++;
110 }
```

ANEXO 6. Código recolector de datos

```
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

MPU6050 sensor1;
MPU6050 sensor2;

int16_t ax1, ay1, az1;
int16_t gx1, gy1, gz1;
int16_t ax2, ay2, az2;
int16_t gx2, gy2, gz2;

void setup() {
  Serial.begin(57600);
  Wire.begin();
  // Inicializa el primer sensor
  sensor1.initialize();
  if (sensor1.testConnection()) {
    Serial.println("Sensor 1 iniciado correctamente");
  } else {
    Serial.println("Error al iniciar el sensor 1");
  }

  // Inicializa el segundo sensor (ajustar la dirección I2C)
  sensor2.initialize();
  if (sensor2.testConnection()) {
    Serial.println("Sensor 2 iniciado correctamente");
  } else {
    Serial.println("Error al iniciar el sensor 2");
  }
}

void loop() {
  // Leer las aceleraciones y velocidades angulares del primer sensor
  sensor1.getAcceleration(&ax1, &ay1, &az1);
  sensor1.getRotation(&gx1, &gy1, &gz1);

  // Leer las aceleraciones y velocidades angulares del segundo
sensor
  sensor2.getAcceleration(&ax2, &ay2, &az2);
  sensor2.getRotation(&gx2, &gy2, &gz2);

  float ax1_m_s2 = ax1 * (9.81 / 16384.0);
  float ay1_m_s2 = ay1 * (9.81 / 16384.0);
  float az1_m_s2 = az1 * (9.81 / 16384.0);
  float gx1_deg_s = gx1 * (250.0 / 32768.0);
  float gy1_deg_s = gy1 * (250.0 / 32768.0);
  float gz1_deg_s = gz1 * (250.0 / 32768.0);

  float ax2_m_s2 = ax2 * (9.81 / 16384.0);
  float ay2_m_s2 = ay2 * (9.81 / 16384.0);
  float az2_m_s2 = az2 * (9.81 / 16384.0);
  float gx2_deg_s = gx2 * (250.0 / 32768.0);
  float gy2_deg_s = gy2 * (250.0 / 32768.0);
```

```
  float gz2_deg_s = gz2 * (250.0 / 32768.0);

  // Mostrar las lecturas
  Serial.print("Sensor 1: ");
  Serial.print(ax1_m_s2); Serial.print("\t");
  Serial.print(ay1_m_s2); Serial.print("\t");
  Serial.print(az1_m_s2); Serial.print("\t");
  Serial.print(gx1_deg_s); Serial.print("\t");
  Serial.print(gy1_deg_s); Serial.print("\t");
  Serial.print(gz1_deg_s); Serial.print("\t");
  Serial.print("5");
  Serial.println();

  Serial.print("Sensor 2: ");
  Serial.print(ax2_m_s2); Serial.print("\t");
  Serial.print(ay2_m_s2); Serial.print("\t");
  Serial.print(az2_m_s2); Serial.print("\t");
  Serial.print(gx2_deg_s); Serial.print("\t");
  Serial.print(gy2_deg_s); Serial.print("\t");
  Serial.print(gz2_deg_s); Serial.print("\t");
  Serial.print("5");
  Serial.println();

  delay(100);
}
```

ANEXO 7. Código red neuronal

```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.stats import skew, kurtosis
from keras.models import Sequential
from keras.layers import Dense

# carga los datos
dataset = pd.read_excel("/content/datos__.xlsx")

# extraccion de data
data=dataset[['acx','acy','acz', 'gx','gy','gz','acx2','acy2','acz2',
'gx2','gy2','gz2' ]]
result=dataset[['parado','caminando','sentado','mov.derecha','mov.izquierda']]
plt.plot(data)

# cálculo de la energía
def energy(signal):
  s2=signal**2
  energy=sum(s2)
  return energy

# extracción de características
# valor medio, valor máximo, curtosis, energía ...
ventanas = []
caracteristicas = {}
tamaño_ventana = 40
j=0;
numiter=round(len(data)/40)-1
nuevacaract = np.zeros((numiter,38))
nueva_target = np.zeros((numiter,5))

for i in range(0, len(data)-tamaño_ventana, tamaño_ventana):
    ventana = data[i:i + tamaño_ventana]
    ventanas.append(ventana)
    nuevacaract[j][0]=ventana['acx'].mean()
    nuevacaract[j][1]=ventana['acy'].mean()
    nuevacaract[j][2]=ventana['acz'].mean()
    nuevacaract[j][3]=ventana['acx'].std()
    nuevacaract[j][4]=ventana['acy'].std()
    nuevacaract[j][5]=ventana['acz'].std()
    nuevacaract[j][6]=ventana['acx'].max()
    nuevacaract[j][7]=ventana['acy'].max()
    nuevacaract[j][8]=ventana['acz'].max()
    nuevacaract[j][9]=skew(ventana['acx'])
    nuevacaract[j][10]=skew(ventana['acy'])
    nuevacaract[j][11]=skew(ventana['acz'])
    nuevacaract[j][12]=kurtosis(ventana['acx'])
    nuevacaract[j][13]=kurtosis(ventana['acy'])
    nuevacaract[j][14]=kurtosis(ventana['acz'])
    nuevacaract[j][15]=energy(ventana['acx'])
    nuevacaract[j][16]=energy(ventana['acy'])
```

```
    nuevacaract[j][17]=energy(ventana['acz'])
    nuevacaract[j][18]=ventana['acx2'].mean()
    nuevacaract[j][19]=ventana['acy2'].mean()
    nuevacaract[j][20]=ventana['acz2'].mean()
    nuevacaract[j][21]=ventana['acx2'].std()
    nuevacaract[j][22]=ventana['acy2'].std()
    nuevacaract[j][23]=ventana['acz2'].std()
    nuevacaract[j][24]=ventana['acx2'].max()
    nuevacaract[j][25]=ventana['acy2'].max()
    nuevacaract[j][26]=ventana['acz2'].max()
    nuevacaract[j][27]=skew(ventana['acx2'])
    nuevacaract[j][28]=skew(ventana['acy2'])
    nuevacaract[j][29]=skew(ventana['acz2'])
    nuevacaract[j][30]=kurtosis(ventana['acx2'])
    nuevacaract[j][31]=kurtosis(ventana['acy2'])
    nuevacaract[j][32]=kurtosis(ventana['acz2'])
    nuevacaract[j][33]=energy(ventana['acx2'])
    nuevacaract[j][34]=energy(ventana['acy2'])
    nuevacaract[j][35]=energy(ventana['acz2'])
    nuevacaract[j][36]=ventana['gx'].max()
    nuevacaract[j][37]=ventana['gy'].max()
    nueva_target[j]=result[tamaño_ventana+i-1:i + tamaño_ventana]
    j=j+1


"""** Creacion de la red neuronal **"""
training_data = nuevacaract[:,0:38]
target_data=nueva_target

#crear el modelo
model = Sequential()
model.add(Dense(38, input_dim=38, activation='relu'))
model.add(Dense(36, activation='relu'))
model.add(Dense(18, activation='relu'))
model.add(Dense(5, activation='softmax'))

model.summary()

model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

model.fit(training_data,
          target_data,
          epochs=120,
          validation_split=0.1)

#evaluar el modelo con los datos
loss, accuracy=model.evaluate(training_data, target_data)



"""** DATOS A ENTRENAR **"""
```

```
test_data = pd.read_excel("/content/data_entre.xlsx")
data_entre=test_data[['acx','acy','acz', 'gx','gy','acx2','acy2','acz2']]

ventanas = []
caracteristicas = {}
tamaño_ventana = 40
j=0;
numiter=round(len(data_entre)/40)-1
nuevacaract_e = np.zeros((numiter,38))
nueva_target_e = np.zeros((numiter,5))

for i in range(0, len(data_entre)-tamaño_ventana, tamaño_ventana):
    ventana = data_entre[i:i + tamaño_ventana]
    ventanas.append(ventana)
    nuevacaract_e[j][0]=ventana['acx'].mean()
    nuevacaract_e[j][1]=ventana['acy'].mean()
    nuevacaract_e[j][2]=ventana['acz'].mean()
    nuevacaract_e[j][3]=ventana['acx'].std()
    nuevacaract_e[j][4]=ventana['acy'].std()
    nuevacaract_e[j][5]=ventana['acz'].std()
    nuevacaract_e[j][6]=ventana['acx'].max()
    nuevacaract_e[j][7]=ventana['acy'].max()
    nuevacaract_e[j][8]=ventana['acz'].max()
    nuevacaract_e[j][9]=skew(ventana['acx'])
    nuevacaract_e[j][10]=skew(ventana['acy'])
    nuevacaract_e[j][11]=skew(ventana['acz'])
    nuevacaract_e[j][12]=kurtosis(ventana['acx'])
    nuevacaract_e[j][13]=kurtosis(ventana['acy'])
    nuevacaract_e[j][14]=kurtosis(ventana['acz'])
    nuevacaract_e[j][15]=energy(ventana['acx'])
    nuevacaract_e[j][16]=energy(ventana['acy'])
    nuevacaract_e[j][17]=energy(ventana['acz'])
    nuevacaract_e[j][18]=ventana['acx2'].mean()
    nuevacaract_e[j][19]=ventana['acy2'].mean()
    nuevacaract_e[j][20]=ventana['acz2'].mean()
    nuevacaract_e[j][21]=ventana['acx2'].std()
    nuevacaract_e[j][22]=ventana['acy2'].std()
    nuevacaract_e[j][23]=ventana['acz2'].std()
    nuevacaract_e[j][24]=ventana['acx2'].max()
    nuevacaract_e[j][25]=ventana['acy2'].max()
    nuevacaract_e[j][26]=ventana['acz2'].max()
    nuevacaract_e[j][27]=skew(ventana['acx2'])
    nuevacaract_e[j][28]=skew(ventana['acy2'])
    nuevacaract_e[j][29]=skew(ventana['acz2'])
    nuevacaract_e[j][30]=kurtosis(ventana['acx2'])
    nuevacaract_e[j][31]=kurtosis(ventana['acy2'])
    nuevacaract_e[j][32]=kurtosis(ventana['acz2'])
    nuevacaract_e[j][33]=energy(ventana['acx2'])
    nuevacaract_e[j][34]=energy(ventana['acy2'])
    nuevacaract_e[j][35]=energy(ventana['acz2'])
    nuevacaract_e[j][36]=ventana['gx'].max()
    nuevacaract_e[j][37]=ventana['gy'].max()
    nueva_target_e [j]=result[tamaño_ventana+i-1:i + tamaño_ventana]
    j=j+1


```

```
"""**ENTRENAMIENTO DE LA RED NEURONAL**"""
y_pred = model.predict(nuevacaract_e)


"""**Guardar la red**"""
model_json=model.to_json()
with open("model.json", "w") as json_file:
  json_file.write(model_json)
model.save_weights("model.h5")


"""**Utilizar la red guardada**"""
from keras.models import model_from_json

json_file = open('/content/model.json','r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
loaded_model.load_weights("/content/model.h5")

"""**Compilar modelo guardado**"""
loaded_model.compile(loss='mean_squared_error', optimizer='adam',
metrics=['binary_accuracy'])

"""**probar modelo**"""
data_test = nuevacaract_e
predicciones = loaded_model.predict(data_test)
```