



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INGENIERÍA,
INDUSTRIA Y CONSTRUCCIÓN**

CARRERA DE INGENIERÍA ELÉCTRICA

**SISTEMA EMBEBIDO ELECTRÓNICO EN KIMONO
PARA MONITORIZAR TÉCNICAS DE JUDO**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO ELÉCTRICO**

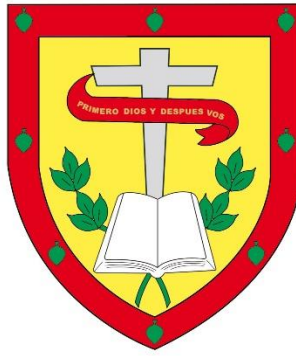
AUTOR: PABLO FABIÁN MARTÍNEZ JARA

DIRECTOR: ING. JUAN CARLOS COBOS TORRES, PhD.

CUENCA – ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INGENIERÍA,
INDUSTRIA Y CONSTRUCCIÓN**

CARRERA DE INGENIERÍA ELÉCTRICA

**SISTEMA EMBEBIDO ELECTRÓNICO EN KIMONO
PARA MONITORIZAR TÉCNICAS DE JUDO**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO ELÉCTRICO**

AUTOR: PABLO FABIÁN MARTÍNEZ JARA

DIRECTOR: ING. JUAN CARLOS COBOS TORRES, PhD.

CUENCA – ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO

DECLARATORIA DE AUTORIA Y RESPONSABILIDAD

Pablo Fabián Martínez Jara portador de la cédula de ciudadanía N° 0102367513. Declaro ser el autor de la obra: "**SISTEMA EMBEBIDO ELECTRÓNICO EN KIMONO PARA MONITORIZAR TÉCNICAS DE JUDO**", sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 04 de octubre de 2023



Pablo Fabián Martínez Jara
0102367513

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Pablo Fabián Martínez Jara, bajo mi supervisión.



Ing. Juan Carlos Cobos Torres PhD.

DIRECTOR

DEDICATORIA

Esta Tesis de Ingeniería la dedico a mi madre, Doctora María Esther Jara (†), quien me dejó su valioso legado de constante lucha y que siempre debo seguir hacia adelante en la vida, enseñándome a ser muy fuerte y que nunca es tarde para ser excelentes personas y profesionales, haciendo el bien a todo lo que nos rodea, siendo siempre una persona útil a la sociedad con humildad, honradez y generosidad.

Madre te llevo en mi corazón y desde el cielo te pido siempre me bendigas para hacer el bien y me des fuerza para seguir luchando para ser más útil a la familia y a los demás.

AGRADECIMIENTOS

Quiero agradecer a Dios por bendecir cada instante de mi vida, a mi Padre y mi Madre (†) que la llevo en mi corazón y que ahora está en el cielo, a mis hermanos y toda mi familia quienes están presentes en todo momento.

Mi cálido agradecimiento a la Universidad Católica de Cuenca, a todas las autoridades docentes y amigos de la Facultad de Ingeniería Eléctrica, por brindarme todos los conocimientos adquiridos y por su constante apoyo para lograr mis metas propuestas, de manera especial al Director de Carrera Ingeniero Daniel Icaza y a mi Tutor de tesis Ingeniero Juan Carlos Cobos, gracias a ustedes por su paciencia, dedicación y amistad, Dios les bendiga siempre.

A todos mis excompañeros y amigos de estudio, los cuales en las aulas compartimos grandes momentos amistad, generosidad y trabajo en equipo durante los años de estudio, Dios les bendiga.

RESUMEN

Hoy en día, la ciencia y la tecnología han desarrollado soluciones inteligentes, que permiten conocer sobre cualquier información en tiempos muy pequeños a cualquier instante. Actualmente, a través de sensores y equipos amigables con el ser humano, es posible obtener datos de temperatura, movimientos, velocidades, aceleración, entre otros, que juntamente con microcontroladores y lenguajes de computador, mejoran y optimizan los procesos en varias ramas.

El judo es un deporte de combate que tiene amplia aceptación a nivel mundial y las condiciones actuales de la competencia, obligan a los entrenadores a mejorar las técnicas y el rendimiento de los atletas, por lo que se ha desarrollado un sistema embebido para monitorear los movimientos del judoca con base a las aceleraciones medidas en diferentes zonas del cuerpo humano, utilizando software y hardware óptimo y especializado para la medición, transmisión y comunicación de datos.

Los resultados obtenidos exponen que, al aplicar de forma errónea la técnica Uchi mata, se producen picos de aceleración superiores a 15 m/seg^2 y presenta considerables cambios de aceleración, mientras que, al ejecutar correctamente la técnica, la aceleración en las extremidades superiores e inferiores se dispersa en un rango de 5 a 10 m/seg^2 , con menos fluctuación.

El prototipo desarrollado busca integrarse a las actuales formas de entrenamiento que, únicamente son efectuadas con respecto a la percepción y experiencia del entrenador, llegando a aportar con información clara, precisa, exacta, utilizando herramientas adaptables al kimono del judoca.

Palabras clave: judo, IoT, sensores, traje biomecánico, aceleración.

ABSTRACT

Nowadays, science and technology have developed intelligent solutions, which allow knowing about any information in a very short time at any moment. At present, through sensors and human-friendly equipment, it is possible to obtain data on temperature, movement, speed, and acceleration, which, together with microcontrollers and computer languages, improve and optimize processes in various branches.

Judo is a combat sport that is widely accepted international. The current conditions of the competition force coach to improve the techniques and performance of athletes, so it has developed an embedded system to monitor the movements of the judo player based on the accelerations measured in different areas of the human body, using optimal and specialized software and hardware for measurement, transmission and data communication.

The results obtained show that when the Uchi Mata technique is misapplied, acceleration peaks above 15 m/sec² and presents considerable changes in speed, while, when the method is performed correctly, the acceleration in the upper and lower extremities is dispersed in a range of 5 to 10 m/sec², with less fluctuation.

The developed prototype seeks to integrate itself into the current forms of training that are only carried out concerning the perception and experience of the trainer, providing clear, precise, and exact information using tools that can adapt to the judo player's kimono.

Keywords: judo, IoT, sensors, biomechanical suit, acceleration

ÍNDICE DE CONTENIDOS

DECLARATORIA DE AUTORIA Y RESPONSABILIDAD.....	ii
CERTIFICACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
RESUMEN.....	vi
ABSTRACT.....	vii
ÍNDICE DE CONTENIDOS.....	viii
LISTA DE FIGURAS.....	x
LISTA DE TABLAS.....	xii
LISTA DE ANEXOS.....	xiii
CAPITULO I.....	1
Introducción.....	1
1.1 Justificación del problema.....	6
1.2 Objetivos.....	7
1.2.1 General.....	7
1.2.2 Específicos.....	7
2. Metodología.....	9
CAPITULO II.....	11
3. Fundamentación Teórica.....	11
3.1 Hardware del sistema WD para kimono.....	11
3.1.1 NodeMCU-32S.....	11
3.1.2 MPU6050.....	13
3.1.3 74HC4067.....	14
3.1.4 Módulo μ SD.....	16
3.1.5 Módulo fuente de alimentación 3.3V/5V.....	17
3.2 Software del sistema WD para kimono.....	18
3.2.1 Arduino IDE.....	18
3.2.2 Proteus Design Suite.....	26
3.2.3 Eagle PCB Design and Electrical Schematic Software.....	27
3.3 Análisis técnico – táctico en el judo.....	29
CAPITULO III.....	31
4. Desarrollo.....	31
4.1 Prototipo v1.....	31

4.2	Prototipo v2.	35
4.3	Prototipo v3.	39
5.	Análisis de resultados y discusión.....	47
5.1	Adquisición de datos.....	47
5.2	Prototipo desarrollado vs. métodos tradicionales.	59
5.2.1	Ventajas del prototipo de traje vs. los métodos tradicionales.	59
5.2.2	Desventajas	61
	CAPITULO IV.....	62
6.	Conclusiones.....	62
7.	Recomendaciones.....	63
8.	Bibliografía.....	65
9.	Anexos.	70

LISTA DE FIGURAS

Figura 1: Judo Gi tradicional.....	2
Figura 2: Técnicas principales del judo.....	2
Figura 3: Diagrama de conexión del hardware del dispositivo propuesto.	4
Figura 4: Arquitectura del sistema propuesto.	5
Figura 5: Metodología a seguir.....	9
Figura 6: NodeMCU 32S.	12
Figura 7: Comunicación serie entre dos dispositivos.	13
Figura 8: Módulo MPU6050.	13
Figura 9: Protocolo de comunicación I2C.....	14
Figura 10: Módulo 74HC4067.	15
Figura 11: Módulo μ SD.....	16
Figura 12: Protocolo SPI.	17
Figura 13: Módulo fuente de alimentación 3.3V/5V.	18
Figura 14: Ventana principal del IDE de Arduino.	19
Figura 15: Ejemplos de declaración de variables.	20
Figura 16: Diagrama de flujo del condicional "if".....	21
Figura 17: Ejemplo del condicional "if".	21
Figura 18: Diagrama de flujo del condicional "switch".	22
Figura 19: Ejemplo del condicional "switch".	22
Figura 20: Diagrama de flujo del bucle "for".	23
Figura 21: Ejemplo del bucle "for".	23
Figura 22: Diagrama de flujo del bucle "while".	24
Figura 23: Ejemplo del bucle "while".	24
Figura 24: Uso de funciones en un programa.	25
Figura 25: Ejemplo de función.	26
Figura 26: Interfaz gráfica de Proteus.	27
Figura 27: Interfaz gráfica de Eagle PCB Design.....	28
Figura 28: Agrupación de elementos electrónicos.....	29
Figura 29: Layout terminado y placa PCB impresa, A) Circuito digitalizado, B) Circuito físico. ..	29
Figura 30: Diagrama de conexión del prototipo v1, A) Microcontrolador y un acelerómetro, B) Microcontrolador y dos acelerómetros.	32
Figura 31: Simulación del prototipo v1.	33
Figura 32: Datos adquiridos del prototipo v1 (Simulado).	34
Figura 33: Esquema físico del prototipo v1.....	34
Figura 34: Datos adquiridos del prototipo v1 (Real).	35
Figura 35: Diagrama de conexión del prototipo v2.....	35
Figura 36: Simulación del prototipo v2.	38
Figura 37: Datos adquiridos del prototipo v2 (Simulado).	39
Figura 38: Esquema físico del prototipo v2.....	39
Figura 39: Datos adquiridos del prototipo v2 (Real).	39
Figura 40: Diagrama de conexión del prototipo v3.....	40
Figura 41: Diagrama esquemático de la PCB diseñada.	41
Figura 42: PCB diseñada, A) Versión digitalizada, B) Versión pre-impresión.....	42
Figura 43: PCB diseñada para el prototipo v3, A) Vista frontal, B) Vista posterior.....	42
Figura 44: Simulación del prototipo v3.	43

Figura 45: Datos adquiridos del prototipo v3 (Simulado).....	44
Figura 46: Esquema físico del prototipo v3, A) Sistema de acelerómetros, B) Sistema electrónico completo, C) Acelerómetro implementado en el traje, D) Caja de conexiones.	45
Figura 47: Datos obtenidos del prototipo v3.	45
Figura 48: Implementación final en el traje, A) Sistema electrónico de medición, B) Sistema electrónico y traje con acelerómetros, C) Judoca con traje de medición vista frontal, D) Judoca con traje de medición vista lateral.....	46
Figura 49: Uchi mata clásico.....	48
Figura 50: Pruebas de la técnica Uchi mata, A) Inicio de Uchi mata ejecutado erróneamente; B) Final de Uchi mata ejecutado erróneamente; C) Ejecución de Uchi mata ejecutado correctamente; D) Final de Uchi mata ejecutado correctamente.	48
Figura 51: Resultados obtenidos en el sensor MPU_0, A) Eje X; B) Eje Y, C) Eje Z.....	50
Figura 52: Resultados obtenidos en el sensor MPU_1, A) Eje X; B) Eje Y, C) Eje Z.....	51
Figura 53: Resultados obtenidos en el sensor MPU_2, A) Eje X; B) Eje Y, C) Eje Z.....	53
Figura 54: Resultados obtenidos en el sensor MPU_9, A) Eje X; B) Eje Y, C) Eje Z.....	55
Figura 55: Resultados obtenidos en el sensor MPU_10, A) Eje X; B) Eje Y, C) Eje Z.....	56
Figura 56: Resultados obtenidos en el sensor MPU_11, A) Eje X; B) Eje Y, C) Eje Z.....	58

LISTA DE TABLAS

Tabla 1: Características del NodeMCU 32S.....	12
Tabla 2: Características del MPU6050.....	13
Tabla 3: Características del 74HC4067.....	15
Tabla 4: Características del módulo μ SD.....	16
Tabla 5: Características del módulo fuente de alimentación 3.3V/5V.....	18
Tabla 6: Tipo de variables en Arduino.....	20
Tabla 7: Conexiones del prototipo v1.	32
Tabla 8: Conexiones del prototipo v2.	36
Tabla 9: Conexiones del prototipo v3.	43
Tabla 10: Análisis de resultados obtenidos en el sensor MPU_0.....	50
Tabla 11: Análisis de resultados obtenidos en el sensor MPU_1.....	52
Tabla 12: Análisis de resultados obtenidos en el sensor MPU_2.....	53
Tabla 13: Análisis de resultados obtenidos en el sensor MPU_9.....	55
Tabla 14: Análisis de resultados obtenidos en el sensor MPU_10.....	57
Tabla 15: Análisis de resultados obtenidos en el sensor MPU_11.....	58

LISTA DE ANEXOS

Anexo 1: Distribución de pines de la tarjeta NodeMCU-32S.	70
Anexo 2: Tabla de verdad del multiplexor 74HC4067.....	70
Anexo 3: Código completo del prototipo v1.	73
Anexo 4: Código completo del prototipo v2.	82
Anexo 5: Disposición de los acelerómetros en el traje.	82

CAPITULO I

Introducción.

El judo es un arte marcial creado a finales del siglo XIX que se convirtió en un deporte Olímpico desde el año 1964 (Ciaccioni et al., 2019); basado en el encuentro directo de dos oponentes con grandes potenciales físicos, técnicos y tácticos (Peset Mancebo et al., 2013), y que se ha extendido a lo largo del mundo con innumerable número de atletas debido a su gran popularidad. En el Ecuador varias fuentes mencionan que los primeros indicios de prácticas de Judo en la provincia del Azuay fueron en 1964, por intermedio del norteamericano Gary, oriundo de Euclid, en el estado de Ohio, Estados Unidos. Era profesor de inglés en una institución educativa conocida de la ciudad de Cuenca, donde lo vieron practicando lo que podría ser judo (Phillips & Toscano, 2022). Normalmente un combate de judo se desarrolla en un dojo cubierto con una estera denominada “tatami” de 8m x 8m más dos o tres metros de seguridad, con una duración de hasta máximo 8 minutos dependiendo de las capacidades de los judocas (término utilizado para definir a los atletas) (Detanico & dos Santos, 2012).

Es popular debido a las bondades que ofrece, tanto físicas (fuerza, flexibilidad, musculatura, agilidad, coordinación) como psicológicas (respeto, disciplina, responsabilidad, regularidad, concentración, serenidad, entre otros) (Gutierrez-Garcia et al., 2018). Es también conocido debido a su larga tradición en la práctica para personas con discapacidad, llamándose “para judo” e incluido por primera vez en los Juegos Olímpicos de Seúl en 1988 (Pečnikar Oblak et al., 2020). La vestimenta que utilizan los peleadores se llama “judo gi” y se aprecia en la Figura 1. Consta esencialmente de una chaqueta, un cinturón (que indica el nivel de estudios del atleta) y un pantalón.



Figura 1: Judo Gi tradicional.

Fuente: (Craats, 2019).

El judo exige acciones de alta intensidad a lo largo del combate, donde se aplican altas cargas de energía, fuerza y velocidad, características coordinadas a lo largo del entrenamiento en coordinación con el entrenador. Esta última persona es la encargada de observar y recordar todas las circunstancias que se presenten con el atleta, con el objetivo de mejorar el rendimiento. Sin embargo, está comprobado que los entrenadores en ciertas ocasiones no pueden observar todo con detalle, por lo que a su vez puede derivar en una interpretación errónea de las distintas técnicas empleadas (Miarka et al., 2012; Torres-Luque et al., 2016). El autor de (Kobayashi & Sharp, 2014) ha dividido a las técnicas del judo como se detalla en la Figura 2.

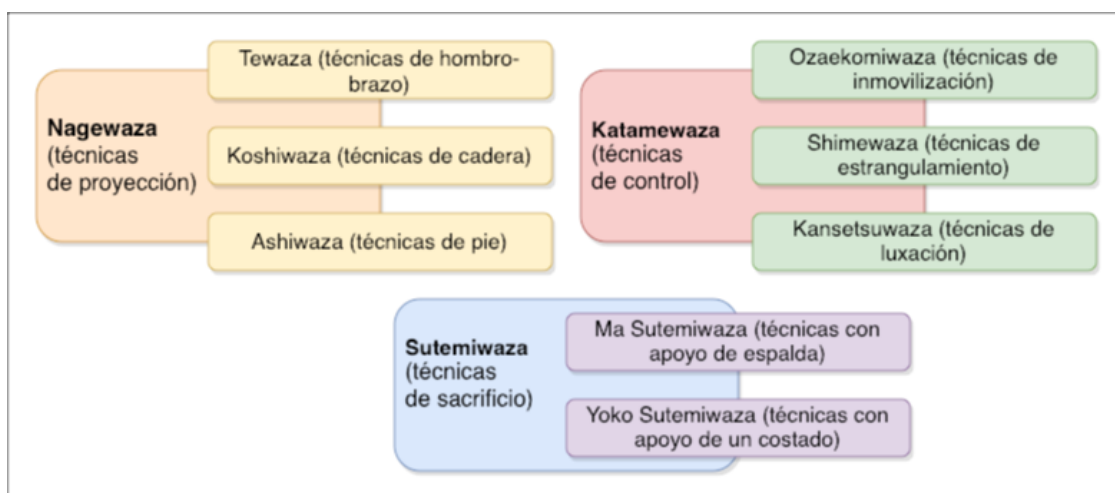


Figura 2: Técnicas principales del judo.

Fuente: (Kobayashi & Sharp, 2014).

En (Sterkowicz et al., 2013) se coincide con lo mencionado anteriormente; estos autores afirman que “la observación de la pelea en las competencias de judo es necesaria, ya que es la única oportunidad de verificar el proceso de entrenamiento del competidor”; de esto se genera una retroalimentación importante al judoca para

enfrentar nuevos desafíos, más exigencias, mejorar técnicas, habilidades y cuidarse de posibles lesiones. Lamentablemente en la realidad tanto competidores, como árbitros y entrenadores están expuestos a cometer errores y muchas veces la precisión es una característica fundamental para obtener buenas retroalimentaciones.

En la actualidad la ingeniería es una ciencia que ha podido solucionar inconvenientes relacionados a la precisión. Hoy en día casi todos los dispositivos eléctricos y electrónicos están conectados entre ellos, intercambiando parámetros en tiempo real, adquiriendo datos y generando información con gran exactitud. IoT¹ es una red abierta y compleja de objetos inteligentes que mediante sensores permite recibir información importante con el objetivo de actuar ante cambios en el entorno y compartir datos entre humanos y dispositivos electrónicos (Madakam et al., 2015). Muchas empresas internacionales dedicadas al IoT se encargan del desarrollo de los WD², que no son más que dispositivos que monitorean en tiempo real distintas variables como: ritmo cardiaco, presión sanguínea, ubicación, movimiento, velocidad, fuerza, COVID-19 (Ates et al., 2021), entre otros; mediante dispositivos que son implementados a lo largo de la vestimenta o incluso en contacto con el cuerpo humano (Iqbal et al., 2021). Se estima que, para los próximos 3 años, el 60% de dispositivos portátiles se usaran no solo para publicar datos con relación a la salud sino para controlar objetos, autenticar transacciones, aplicaciones de domótica, y otras actividades (Poongodi et al., 2020); esto se debe a la mayor disponibilidad, el menor costo y los avances de los dispositivos informáticos personales, como son los teléfonos y relojes inteligentes (R. T. Li et al., 2016).

De todas las variables mencionadas en el párrafo anterior la detección del movimiento humano ha sido la más útil y explotada durante muchos años en aplicaciones como el deporte o la medicina para evaluar riesgos de caídas, cuantificación del ejercicio, habito de las personas o el cuidado del adulto mayor; mediante sensores como acelerómetros, magnetómetros y giroscopios de 3 ejes que permiten obtener datos para un propósito específico (Haghi et al., 2017). Desde el año 2012 hasta la fecha se han registrado 24 600 artículos científicos en el buscador Google Académico, utilizando las palabras clave “wearable devices” y “sports”.

El trabajo realizado por (S. Li et al., 2020) propone un sistema IoT basado en WD para monitorear en tiempo real el ritmo cardiaco, la frecuencia respiratoria y el movimiento durante la actividad física. La singularidad de este trabajo radica en la

¹ Internet of Things – Internet de las cosas.

² Wearable devices – Tecnología corporal.

implementación de la capa de niebla en la puerta de enlace a los sensores físicos, con el objetivo de disminuir la cantidad de datos transmitidos en la nube. El método propuesto de comunicación FCE-WSN³ ha presentado alta precisión en la predicción de situaciones anormales para los deportistas. En (Kos & Kramberger, 2017) se ha realizado un mini dispositivo para detectar y registrar los movimientos y la información biométrica del usuario durante la actividad física. El hardware del dispositivo se muestra en la Figura 3:

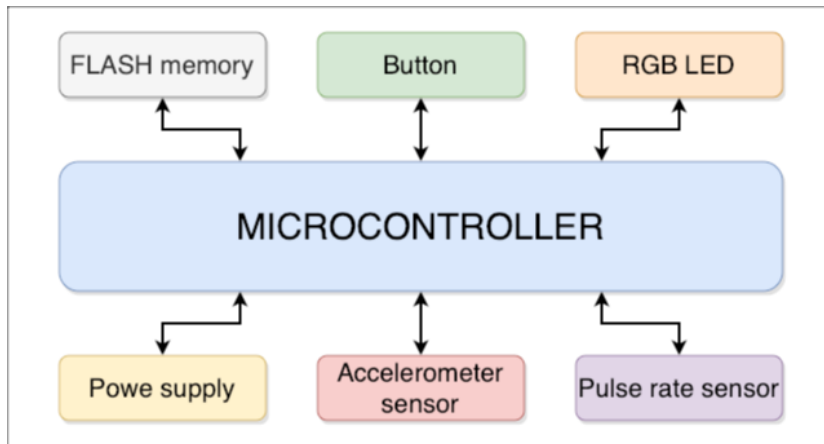


Figura 3: Diagrama de conexión del hardware del dispositivo propuesto.

Fuente: (Kos & Kramberger, 2017).

El dispositivo es capaz de enviar registrar datos de temperatura de la piel, frecuencia del pulso, velocidad y aceleración de los movimientos de la muñeca. También puede enviar los datos a la nube y posee una autonomía de 6 horas con la batería conectada al sistema. El rendimiento del sistema fue puesto a prueba en 7 jugadores de tenis con diferentes niveles de conocimiento, aportando alta precisión en los datos obtenidos. También puede ser implementado en el golf.

Investigadores (Kim & Kim, 2016) han desarrollado un sistema IoT para detectar el movimiento del pie al realizar una patada a un balón de futbol. Para esto se diseñó un WD que consta de un acelerómetro, giroscopio y magnetómetro de tres ejes que conectados a un microcontrolador envía los datos a una PC de forma inalámbrica a una velocidad de muestreo de 100 Hz. En ese mismo instante de la acción, una cámara graba los movimientos del deportista y son comparados con los valores censados, demostrando que al utilizar este dispositivo y procesando los datos con un algoritmo que

³ Fog Assisted Computational efficient Wearable sensor networks – Redes de sensores portátiles eficientes computacionales asistidos por niebla.

mide el impacto se puede obtener un sistema de bajo coste con sensores accesibles en el mercado, que generan buenos resultados para este tipo de análisis.

Otro deporte en el cual la investigación no ha pasado por alto es el baloncesto. Se ha diseñado un WD en (Nguyen et al., 2015) con el objetivo de monitorear cuando el atleta corre, camina o trota, así como reconocer el tipo de actividad que se está efectuando. Se ha utilizado un microcontrolador del fabricante STMicroelectronics como principal elemento del hardware. Tiene una autonomía de 13.2 horas, recepta información del atleta como: temperatura, presión y aceleración y recopila los datos en una tarjeta SD para su posterior procesamiento. Con los datos obtenidos se ha realizado un reconocimiento de actividades mediante un algoritmo de ML⁴ obteniendo un rendimiento del 92% y proyectando a un posible aumento de sensores en distintas partes del cuerpo.

En el vóleybol se ha desarrollado un WD (Salim et al., 2019) con el objetivo de capturar los datos del acelerómetro, giroscopio, magnetómetro y barómetro instalados en el dispositivo y a la vez clasificar los movimientos realizados utilizando técnicas de ML. El sistema que ha sido implementado se muestra en la Figura 4, consta de un pequeño dispositivo portátil colocado en una o ambas muñecas del deportista para recopilar los datos mencionados; esto a la vez es enviado a un sistema de clasificación mediante arboles de decisión y discrimina si la acción ejecutada se encuentra dentro de la base de datos. el resultado es comparado con el video grabado y se genera una retroalimentación en tiempo real que es notificada al deportista y al entrenador.

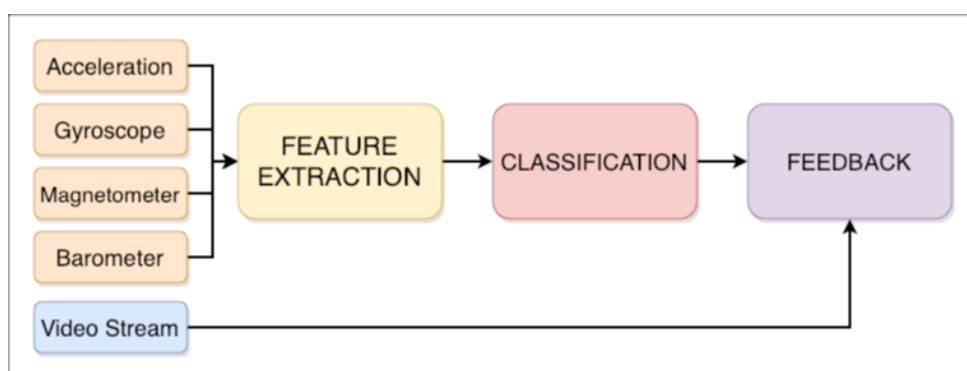


Figura 4: Arquitectura del sistema propuesto.

Fuente: (Salim et al., 2019).

Se ha demostrado con este sistema que los WD pueden llegar a ser un complemento para la tradicional revisión de videos de las maniobras ejecutadas; en un

⁴ Machine Learning – Aprendizaje Automático.

futuro se propone realizar otras técnicas de aprendizaje automático, así como ampliar el número de sensores.

Los deportes de contacto tampoco son la excepción para realizar investigación con dispositivos IoT. KUMITRON (Echeverría & C. Santos, 2021) es un sistema WD que recepta datos de los sensores implementados en un karategui (vestimenta de los karatekas), integra técnicas de inteligencia artificial y visión por computador y permite a los practicantes del karate mejorar sus habilidades a la hora del combate. KUMITRON utiliza un dron para grabar los movimientos realizados en un método de combate llamado “kumite”; un microcontrolador recoge los valores de aceleración, ritmo cardíaco y temperatura y son enviados de forma inalámbrica a un servidor local. Utilizando la librería de código abierta OpenCV se puede realizar visión artificial para segmentar las partes del cuerpo humano en el video y reconocer si es que hay un cambio de posición entre frame y frame del video, es decir, si el karateka se movió. Con los datos obtenidos se construye un diagrama de vector de direcciones y el entrenador es el encargado de procesar la información obtenida para construir la retroalimentación con base a su conocimiento y las reglas básicas del karate. El sistema emitirá valores predichos de movimientos con respecto a los parámetros de entrenamiento del sensei (entrenador del karate).

Otros investigadores han apostado al e-textil aplicados a deportes de combate (Ma et al., 2021; Saponara, 2017; Tama et al., 2017; Ye et al., 2022); e-textil es la construcción de ropa inteligente mediante hilos conductores, galgas extensiométricas, o nano sensores que forman parte de la indumentaria.

Toda esta revisión de la literatura ha desembocado a una conclusión: el IoT, los WD y el e-textil en el deporte tienen como objetivo conocer los movimientos de los atletas, velocidades y aceleración de ejecución de movimientos y monitoreo de signos vitales con el fin de mejorar el rendimiento deportivo, perfeccionar técnicas, predecir escenarios o evitar lesiones.

1.1 Justificación del problema.

Actualmente la disciplina deportiva de Judo en nuestro país está en aumento, cada vez con nuevos integrantes; existen excelentes deportistas que han sobresalido y ya están participando en los juegos olímpicos. Así, el judo ecuatoriano aportó con tres atletas al selecto grupo que representa el país en los Juegos Olímpicos de Tokio 2020. Vanessa Chalá, Estefanía García y Lenin Preciado representaron a Ecuador en judo. Los tres, se ubicaban en zona de clasificación olímpica, según el ranking mundial en esta disciplina.

Por lo tanto, se pretende ayudar a mejorar las técnicas de los movimientos biomecánicos en los entrenamientos, para lo cual es necesario obtener información como de la velocidad, fuerza, y aceleración de estas destrezas del cuerpo humano, con el propósito de analizar y evaluar estos datos, encontrando debilidades y fortalezas de cada atleta; así será posible desarrollar mejores planes de entrenamiento efectivos, además, los entrenadores podrán encontrar los errores y encontrar las correcciones necesarias para cada uno de los deportistas.

En un deportista el entrenamiento debe ser integral, las capacidades físicas de fuerza, resistencia, velocidad son básicos en los planes de entrenamiento. Con los resultados, se espera obtener el mejor rendimiento, esto ayudará principalmente al análisis de la cinética del judoca, para lo cual se implementarán sensores detectores de velocidad y aceleración en cada articulación del cuerpo humano, codo, muñeca, cadera, rodilla y tobillo; los mismos, que por medio de conexiones con hilos conductores flexibles entregarán los datos al microcontrolador, este dispositivo por medio de un interfaz, se conectara de manera inalámbrica a un computador, el cual utilizara software adecuados para analizar y evaluar esta información por parte del entrenador.

Esta necesidad de conocer e interpretar la información de los movimientos biomecánicos, propone el diseño, construcción e implementación de un sistema electrónico para recolectar información por medio de sensores inerciales ubicados estratégicamente en un traje deportivo de Judo.

1.2 Objetivos.

1.2.1 General.

- Diseñar, construir e implementar en un traje de judo un sistema prototipo con microcontroladores, para monitorear y obtener datos de fuerza, aceleración y velocidad de los movimientos biomecánicos del atleta, por medio de sensores de aceleración de bajo costo, y su enlace inalámbrico de la información con un ordenador para que el entrenador visualice y analice las técnicas.

1.2.2 Específicos.

- Fortalecer y ampliar los conceptos sobre sistemas embebidos integrados en prendas deportivas mediante revisión de literatura científica especializada para contar con información actualizada y de calidad.

- Determinar la posición de los sensores y su cantidad mediante un juicio de expertos para que su ubicación sea correcta con respecto a las necesidades de los entrenadores.
- Diseñar y construir el sistema embebido mediante microcontroladores y sensores empotrables en prenda de vestir para contar con el sistema de medición de fuerza y velocidades.
- Medir los movimientos de los deportistas mediante marcadores en los trajes y el análisis de dichos movimientos mediante cámaras para poder contar con información adicional a los sensores físicos.
- Visualizar los datos obtenidos por los sensores mediante un ordenador y un sistema de comunicación con el Kimono.

2. METODOLOGÍA.

El presente trabajo se basa en el modelo “deductivo descriptivo”. Se inicia estudiando el judo como concepto general, principales características del deporte, movimientos biomecánicos utilizados por los atletas, las principales técnicas que se realizan en un combate o en el entrenamiento y los tipos de datos que normalmente los entrenadores requieren para la práctica diaria. El siguiente punto es la investigación sobre los distintos sistemas IoT o dispositivos WD que actualmente existen en la literatura y como han sido aplicados en el deporte, mediante sensores inalámbricos o cableados que reciban diferentes datos, que servirán para realizar la discusión de esta investigación. El último paso de esta tesis es el desarrollo e implementación de un sistema IoT basado en WD mediante sensores de velocidad y aceleración que reciban datos en tiempo real. La Figura 5 muestra un esquema de los pasos a seguir.

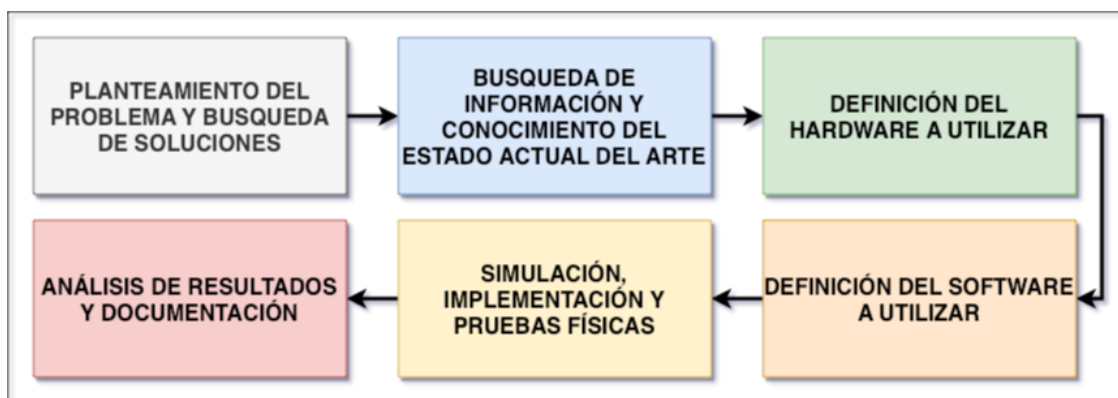


Figura 5: Metodología a seguir.

Fuente: El autor.

- En la etapa inicial de esta investigación se plantea el prototipo del WD, definiendo hardware a utilizar, cumpliendo los siguientes pasos:
 - Planteamiento del prototipo, búsqueda de sensores, microcontroladores, actuadores y demás dispositivos disponibles en el mercado que integren el hardware del WD.
 - Realizar un primer bosquejo de la ubicación de los sensores a lo largo del kimono, así como la central electrónica donde periféricos de entrada y salida serán conectados.
- Una vez que se ha proyectado el WD es necesario definir el software; se realizan los siguientes pasos:

- Los sensores, microcontrolador y actuadores a utilizar funcionan con librerías creadas por desarrolladores externos; se realiza una investigación sobre el funcionamiento de cada librería a utilizar.
 - Se desarrolla el algoritmo aplicando conceptos de programación orientada a objetos, condicionales, bucles y funciones personales.
 - Se depuran errores y se trata de reutilizar el mayor número de funciones creadas.
- C. Una adecuada implementación de un prototipo WD debe de ser simulado previo a la instalación física. Se realizan los siguientes pasos:
- Se utiliza un software de simulación que permita crear un escenario digital lo más cercano posible a la realidad utilizando distintos periféricos disponibles para comprobar el funcionamiento del software antes de la implementación física.
 - En este apartado se crea y se simula la electrónica adicional que servirá para conectar los diferentes sensores y actuadores.
- D. Se realizan las pruebas con el dispositivo WD implementado en el traje de pelea:
- Implementación de los diferentes sensores, tendido del cableado, pruebas eléctricas y electrónicas.
 - Realizar pruebas de funcionamiento, recolección y visualización de datos.
 - Analizar los datos, documentar y publicar los resultados obtenidos.

Con la siguiente investigación se pretende diseñar, simular, construir e implementar un prototipo WD en un kimono mediante un sistema IoT que involucre: sensores, microcontrolares y actuadores para monitorear y obtener datos de fuerza, velocidad y aceleración de los movimientos biomecánicos del atleta en tiempo real para que el entrenador visualice, interprete y analice los datos obtenidos con el objetivo de mejorar técnicas, posturas y evitar lesiones al ejecutar acciones propias del combate.

El presente trabajo se ha dividido de la siguiente manera: En el CAPITULO II se encuentra disponible la Fundamentación Teórica, en el CAPITULO III se realiza el desarrollo, análisis y discusión para culminar con el CAPITULO IV donde se explican las conclusiones, enumeraciones, bibliografía y los anexos a esta investigación.

CAPITULO II

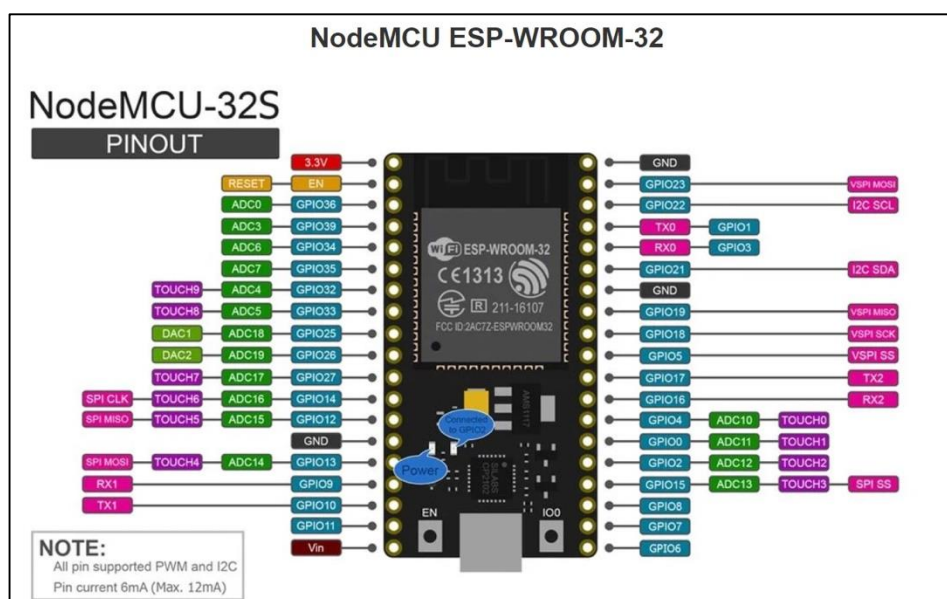
3. FUNDAMENTACIÓN TEÓRICA.

3.1 Hardware del sistema WD para kimono.

3.1.1 NodeMCU-32S.

El proyecto Arduino engloba variedad de tarjetas electrónicas con múltiples propósitos. Dependiendo del alcance del proyecto una tarjeta Arduino puede tener más o menos características, pero esencialmente consta de un microcontrolador, puertos de entrada/salida, protocolos de comunicación y un pequeño espacio de almacenamiento. Arduino es una plataforma de software libre que varias multinacionales han apostado para crear sistemas IoT; Doit es una empresa que ha desarrollado la tarjeta NodeMCU-32S (Figura 6) o llamada también DEVKIT V1.

El NodeMCU-32S es una placa de desarrollo que lleva incorporado un SoC⁵ del modelo ESP8266EX que se encarga de gestionar procesos en la CPU, puertos de entrada y salida, memoria interna y protocolos de comunicación. Además, integra un módulo ESP-12 que no es más que la memoria Flash y la antena Wi-Fi (Hernández, 2017). Las características principales del DEVKIT V1 se pueden visualizar en la Tabla 1 y la distribución de entradas/salidas se encuentra en el



Anexo 1.

⁵ System on a Chip – Sistema en chip.

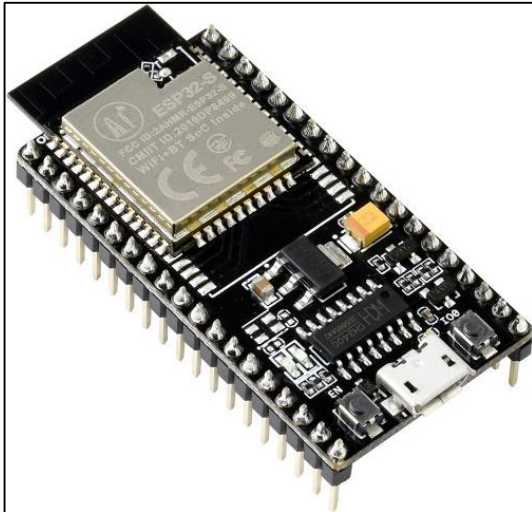


Figura 6: NodeMCU 32S.

Fuente: El autor.

Tabla 1: Características del NodeMCU 32S.

CARACTERÍSTICAS	VALOR
Resolución	12 bits
ADC	18
DAC	2 de 8 bits
SRAM	448kB
Memoria FLASH	4MB
GPIO	36
Voltaje de alimentación	2.5V – 3.6V
Voltaje de entrega	1V, 2V, 3.3V
Frecuencia	160MHz – 240MHz
Comunicaciones	Wi-Fi, Bluetooth, SPI, I2C, UART

Fuente: (Martínez & Sebastián, 2019).

3.1.1.1 Comunicación Serial.

La comunicación en serie (Figura 7) es un protocolo de comunicación entre dispositivos muy tradicional utilizado para el intercambio de datos, control y depuración, entre otros.

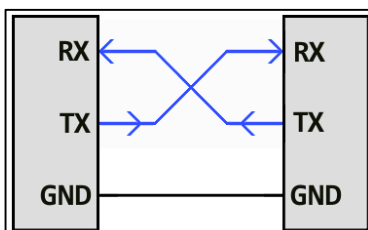


Figura 7: Comunicación serie entre dos dispositivos.

Fuente: (Robots Didácticos, 2020).

Permite el intercambio de datos de bit a bit sin necesidad de un reloj (comunicación asincrónica); es decir, se puede enviar información por un canal, mientras que se reciben datos por el otro. Para que sea esto posible se requiere de las siguientes características:

- Bits de datos: Cantidad de bits que se encuentra en el dato para la transmisión.
- Bits de sincronización: Indica cuando iniciar y detener la transmisión o recepción entre los dispositivos.
- Velocidad de baudios: Indica el número de bits por segundo que se envían, normalmente los valores son: 115200, 9600 y 4800 [baudios] (Robots Didácticos, 2020).

3.1.2 MPU6050.

El módulo MPU6050 (Figura 8) es una IMU⁶ que internamente contiene un acelerómetro de tres ejes y un giroscopio también de tres ejes, que permite obtener la aceleración [m/seg²] y la velocidad angular [rad/seg] de un objeto en movimiento. Se puede comunicar con el microcontrolador utilizando el protocolo I²C y es usualmente utilizado para aplicaciones de navegación, goniometría, estabilización, entre otros (NAYLAMP MECHATRONICS, 2016). Las características principales de este módulo se encuentran detallados en la Tabla 2.

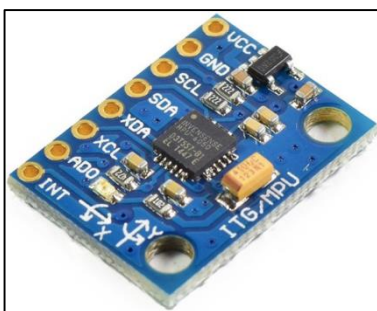


Figura 8: Módulo MPU6050.

Fuente: (NAYLAMP MECHATRONICS, 2016).

Tabla 2: Características del MPU6050.

CARACTERÍSTICAS	VALOR
-----------------	-------

⁶ Inertial Measurement Unit – Unidad de Medición Inercial.

Voltaje de operación	3V/3.3V – 5V DC
Grados de libertad (DoF)	6
Rango Acelerómetro	2g/4g/8g/16g
Rango Giroscopio	250Grad/seg, 500Grad/seg, 1000Grad/seg, 2000Grad/seg
Conversor AD	16 bits

Fuente: (NAYLAMP MECHATRONICS, 2016).

3.1.2.1 Protocolo I²C.

El protocolo I²C (Figura 9) es también un puerto que permite la comunicación entre varios dispositivos que pueden ser maestros y esclavos. Facilita la interacción entre dispositivos ya que necesita únicamente de dos líneas de señal y una “tierra” común. La forma de transmitir es la siguiente:

- SCL: Es la línea de los pulsos de reloj que sincronizan el sistema.
- SDA: Línea por donde se emiten y se reciben los datos.
- GND: Línea común entre todos los dispositivos (Robots Didácticos, 2019).

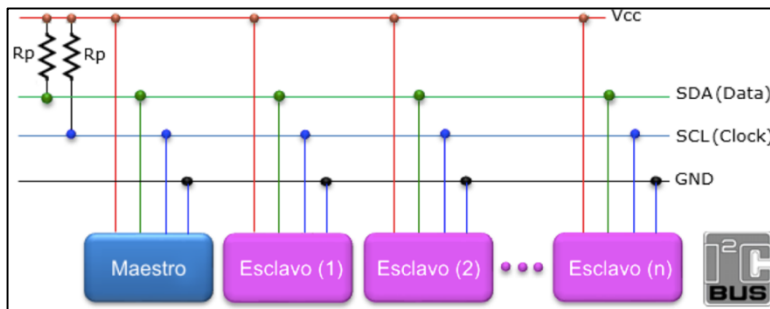


Figura 9: Protocolo de comunicación I²C.

Fuente: (Robots Didácticos, 2019).

El maestro es el que tiene el control absoluto de la comunicación, el inicia y detiene la comunicación con los esclavos, esto modificando la disponibilidad de la línea SCL. En la práctica el maestro es normalmente el microcontrolador, mientras que los esclavos son los sensores que actúan con el medio físico.

3.1.3 74HC4067.

Los multiplexores y de-multiplexores son dispositivos que permiten controlar un mayor número de entradas o salidas, mediante una señal de control. El 74HC4067 (Figura 10) es un multiplexor/de-multiplexor de 16 canales que funciona con señales analógicas o digitales y normalmente es utilizado para aplicaciones con protocolos de

comunicación serial, I²C, o SPI. La Tabla 3 muestra las características principales y la

TRUTH TABLE

S0	S1	S2	S3	\bar{E}	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

H= High Level
L= Low Level

tabla de verdad del módulo se encuentra en el

Anexo 2.

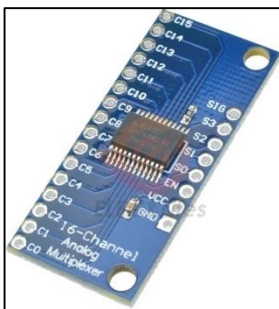


Figura 10: Módulo 74HC4067.

Fuente: (Llamas, 2016b).

Tabla 3: Características del 74HC4067.

CARACTERÍSTICAS	VALOR
Tecnología	CMOS
Compatible con voltajes	TTL
Voltaje de operación	2V – 6V
Temperatura de operación	-55° a 125°

Fuente: (Llamas, 2016b).

3.1.4 Módulo μ SD.

El módulo μ SD (Figura 11) es un dispositivo que permite almacenar diferente tipo de información en una tarjeta μ SD, mediante el bus SPI o en algunas ocasiones ciertos fabricantes han utilizado el protocolo serial o I²C. La gran ventaja de utilizar estos dispositivos es el dotar a la placa de desarrollo Arduino de memoria prácticamente ilimitada; la gran desventaja es el uso del procesador, la memoria flash y la memoria dinámica (Llamas, 2016a). Las características principales se encuentran en la Tabla 4.

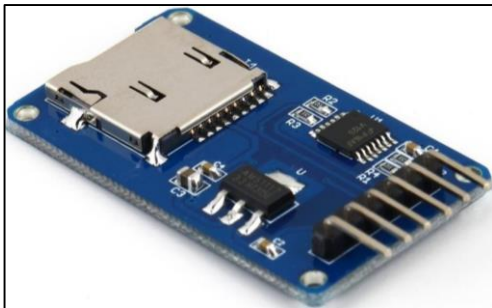


Figura 11: Módulo μ SD.

Fuente: (Llamas, 2016a).

Tabla 4: Características del módulo μ SD.

CARACTERÍSTICAS	VALOR
Voltaje de alimentación	3.3V o 5V
Regulador de tensión incorporado	3.3V
Consumo	< 200mA
Interfaz	SPI

Fuente: (Llamas, 2016a).

3.1.4.1 Protocolo SPI.

El protocolo SPI (Figura 12) es una solución síncrona que permite la comunicación entre varios dispositivos. Es muy popular debido a su simplicidad de funcionamiento ya que funcionan como maestro y esclavos, en donde el maestro es el que habilita la comunicación con un esclavo en común. Consta de las siguientes interfaces:

- MOSI: Línea de transmisión del maestro hacia el esclavo.
- MISO: Línea de transmisión del esclavo hacia el maestro.
- CLK: Línea de los pulsos del reloj que sincronizan el sistema y es enviado por el esclavo.

- SS: Línea que selecciona al esclavo y a la vez habilita la comunicación (Aprendiendo Arduino, 2013).

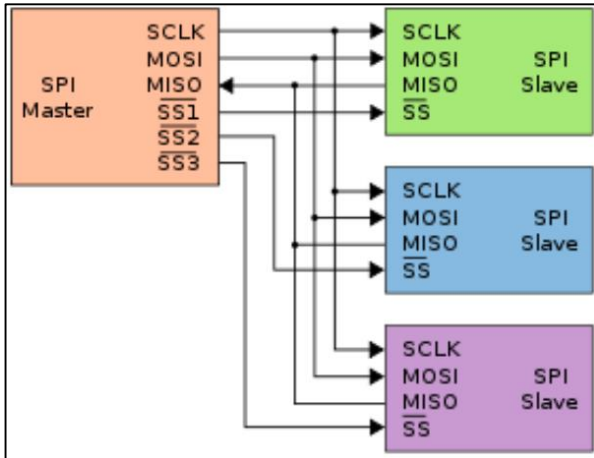


Figura 12: Protocolo SPI.

Fuente: (Aprendiendo Arduino, 2013).

3.1.5 Módulo fuente de alimentación 3.3V/5V.

Normalmente en las prácticas y al momento de “subir” un programa a una placa de desarrollo Arduino se utiliza la conexión USB, la cual tiene dos características: brinda una comunicación serial entre PC y microcontrolador y permite alimentar todos los periféricos con un voltaje DC de 5 V. El reto se encuentra cuando se necesita una autonomía o se requiere de sistemas portátiles. Las placas de desarrollo permiten diferentes formas de alimentación:

- USB (Versión 1): Explicada anteriormente, con una corriente de salida máxima de 500mA.
- Adaptador de corriente: Dependiendo de la placa se debe de proveer un valor de voltaje y corriente adecuado de manera que ningún elemento sufra un inconveniente eléctrico.
- Baterías: Se debe considerar los mAh que suministran las baterías para obtener la autonomía en la placa de desarrollo (Aprendiendo Arduino, 2016).

El módulo fuente de alimentación 3.3V/5V (Figura 13) permite alimentar a una placa de desarrollo mediante un Jack o un USB, obteniendo a la salida dos líneas diferentes de 3.3V y 5V. Las características del dispositivo se encuentran detalladas en la Tabla 5.

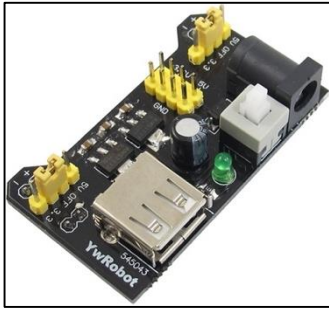


Figura 13: Módulo fuente de alimentación 3.3V/5V.

Fuente: (Aprendiendo Arduino, 2016).

Tabla 5: Características del módulo fuente de alimentación 3.3V/5V.

CARACTERÍSTICAS	VALOR
Voltaje de entrada	6.5V a 12V
Voltaje de salida	3.3V – 5V (Seleccionable)
Salida de tensión USB	5V
Corriente máxima de salida	700mA

Fuente: (Aprendiendo Arduino, 2016).

3.2 Software del sistema WD para kimono.

3.2.1 Arduino IDE.

Un IDE⁷ es un software que permite crear aplicaciones que combina distintas características necesarias para “subir” a un microcontrolador en una sola interfaz gráfica. El IDE de Arduino (Figura 14) permite escribir código compatible con todas las plataformas Arduino. Está disponible para Windows, Mac OS y Linux y tiene compatibilidad para todas las tarjetas Arduino como: Uno, Dual, Nano, Mega, LilyPad, así como la NodeMCU 32S.

⁷ Integrated Development Environment – Entorno de Desarrollo Integrado.

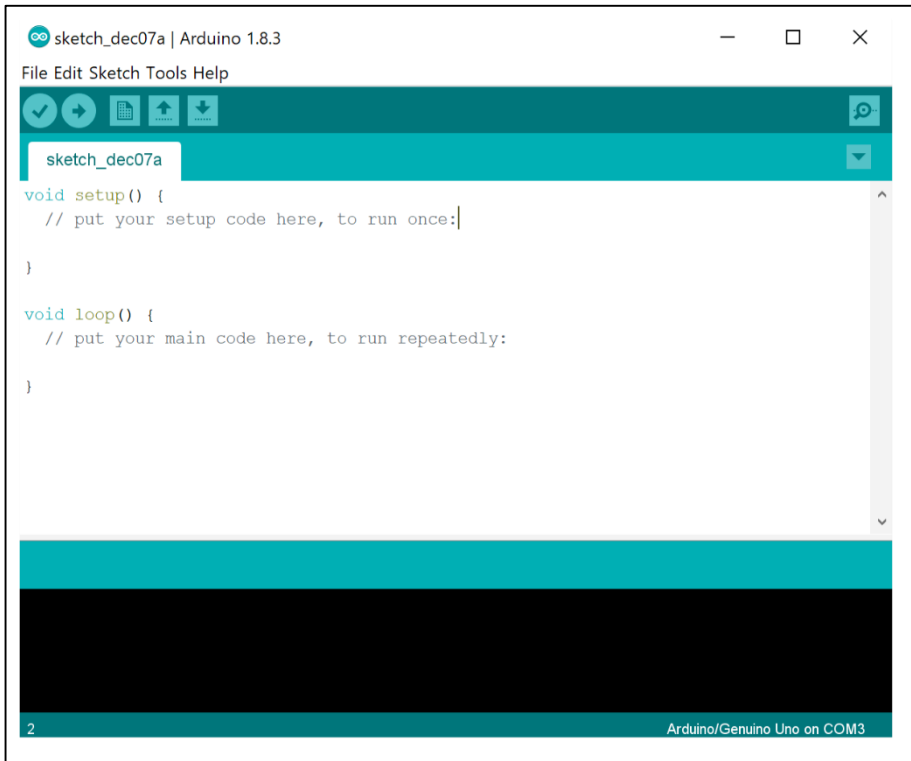


Figura 14: Ventana principal del IDE de Arduino.

Fuente: (Pan & Zhu, 2018; Peña, 2020).

La ventaja de utilizar Arduino IDE es la gran comunidad que existe en la web; controladores y demás archivos necesarios se encuentran para descargar inmediatamente y de forma gratuita. El lenguaje de programación que se utiliza es una combinación entre C y C++ (Pan & Zhu, 2018; Peña, 2020).

3.2.1.1 Librerías y variables.

Antes de conocer sobre condicionales, bucles y funciones propias del programador es necesario conocer ciertos conceptos propios de los lenguajes de programación:

- A. Librerías: Son distintas variables, definiciones, funciones, objetos, o en otras palabras código desarrollado por terceras personas con el propósito de facilitar el trabajo a los nuevos programadores.
- B. Variables: La mayoría de las veces que se trabaja con sensores es necesario guardar la información ya que son valores que cambian durante el tiempo. Una variable es un espacio en la memoria del Arduino que permite almacenar de una forma volátil la información obtenida desde el exterior, tiene un nombre un valor y un tipo. Los tipos de datos utilizados en Arduino se pueden visualizar en la Tabla 6:

Tabla 6: Tipo de variables en Arduino.

TIPO	NÚMERO DE BYTES	RANGO
Bool	1 byte	0 a 1
Byte / Unsigned Char	1 byte	0 – 255
Char	1 byte	-128 – 127
Int	2 bytes	-32768 – 32767
Word / Unsigned Int	2 bytes	0 – 65535
Long	2 bytes	-2147483648 – 2147483647
Unsigned Long	4 bytes	0 – 4294967295
Float / Double	4 bytes	-3.4028e ³⁸ a 3.4028e ³⁸
String	1 byte + x	Array de caracteres
Array	1 byte + x	Colección de variables

Fuente: (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

La forma correcta para utilizar una variable dentro del IDE de Arduino es:

- `tipo_de_la_variable nombre_de_la_variable = valor_de_la_variable;`
(Figura 15) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

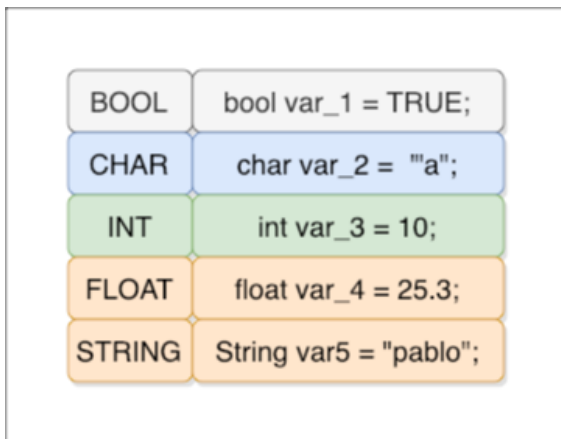


Figura 15: Ejemplos de declaración de variables.

Fuente: El autor.

3.2.1.2 Condicionales.

Muchas de las veces en un algoritmo de programación es necesario evaluar el valor de una variable para realizar alguna tarea, activar/desactivar una señal, entre otras acciones. Las sentencias condicionales más usadas en programación son el “if” y el “switch”.

- A. If: Comprueba si una condición determinada es verdadera o falsa. Puede evaluar distintas condiciones como se muestra en la Figura 16. Si cumple alguna de las condiciones que se programa entra a esa parte del código,

ejecuta las líneas programadas y regresa al punto siguiente de la instrucción if.

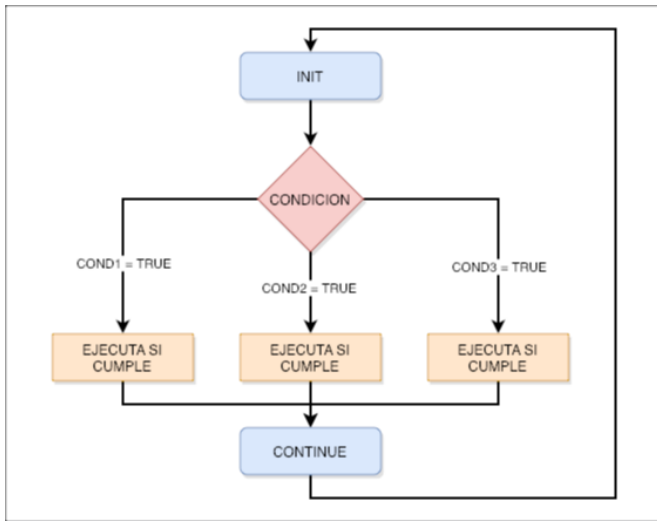


Figura 16: Diagrama de flujo del condicional "if".

Fuente: El autor.

La forma correcta para utilizar un condicional "if" dentro del IDE de Arduino es:

- palabra_reservada_if (condición) {acción_a_ejecutar} (Figura 17) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

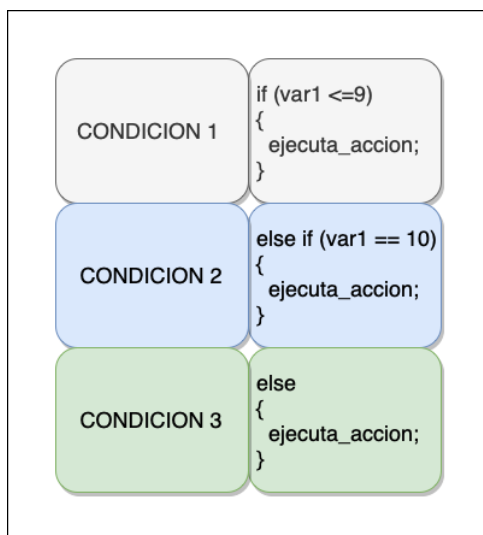


Figura 17: Ejemplo del condicional "if".

Fuente: El autor.

B. Switch: Tiene el mismo funcionamiento que el condicional if. Puede evaluar distintas condiciones como se muestra en la Figura 18.

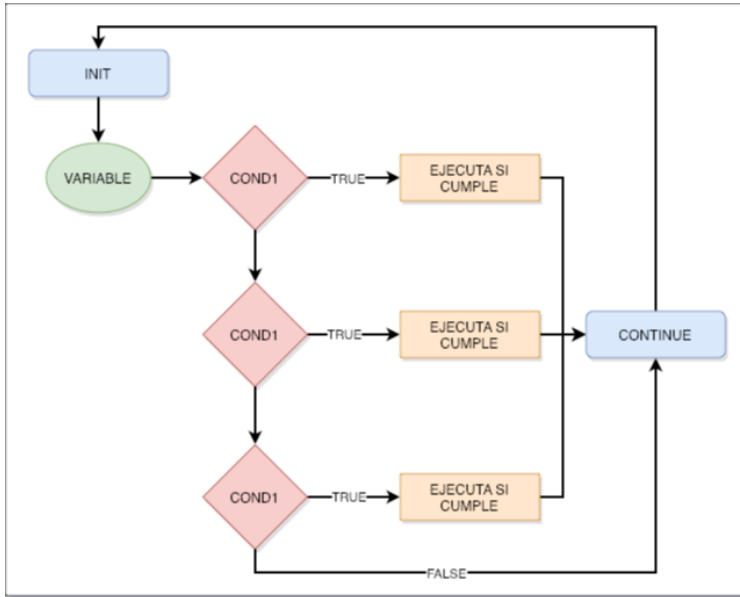


Figura 18: Diagrama de flujo del condicional "switch".

Fuente: El autor.

La forma correcta para utilizar un condicional "switch" dentro del IDE de Arduino es:

- palabra_reservada_switch (condición) {acción_a_ejecutar} (Figura 19) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

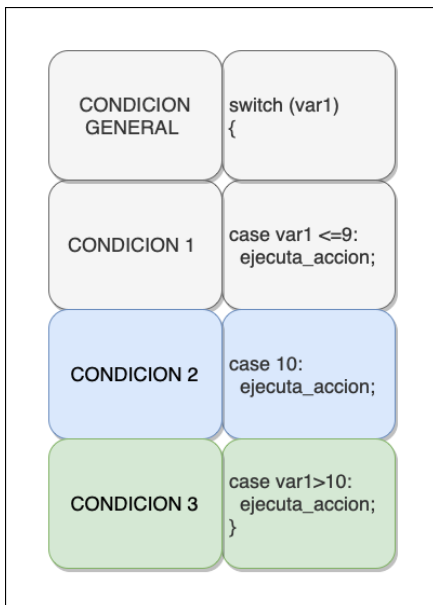


Figura 19: Ejemplo del condicional "switch".

Fuente: El autor.

3.2.1.3 Bucles.

Un bucle es un determinado número de líneas de código que se repiten mientras se cumpla una condición. En programación, los bucles más utilizados son “for” y “while”.

A. For: Se construye mediante tres valores: inicio, final e iteración. El valor de inicio de una variable cambia con respecto a la iteración y el código comprendido dentro del bucle for se ejecuta una y otra vez hasta que la variable llegue a ser el valor “final”. En la Figura 20 se muestra el diagrama de flujo.

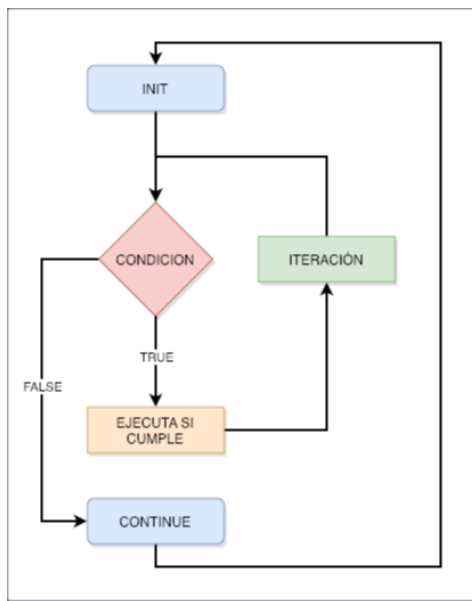


Figura 20: Diagrama de flujo del bucle "for".

Fuente: El autor.

La forma correcta para utilizar un bucle “for” dentro del IDE de Arduino es:

- palabra_reservada_for (variable valor_inicial; valor_final; iteracion) {acción_a_ejecutar}(Figura 21) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

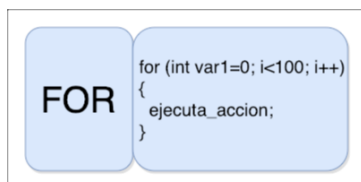


Figura 21: Ejemplo del bucle "for".

Fuente: El autor.

B. While: Tiene un funcionamiento similar al bucle for. El código que este dentro de la sentencia while se repetirá una y otra vez hasta que la variable cumpla la condición programada al inicio del while. En la Figura 22 se muestra el diagrama de flujo.

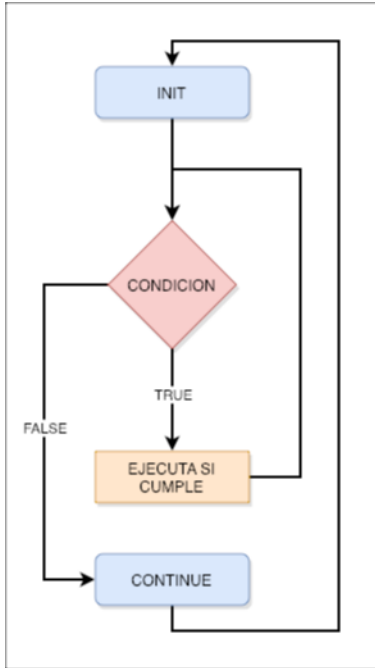


Figura 22: Diagrama de flujo del bucle "while".

Fuente: El autor.

La forma correcta para utilizar un bucle "while" dentro del IDE de Arduino es:

- palabra_reservada_while (condición) {acción_a_ejecutar}(Figura 23) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

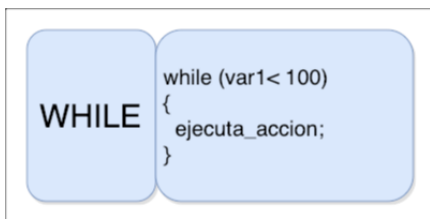


Figura 23: Ejemplo del bucle "while".

Fuente: El autor.

3.2.1.4 Funciones.

Un programa de Arduino por defecto consta de dos funciones principales que el usuario no debe de omitir:

- void setup(): Se configuran entradas, salidas, inicio de instancias y demás parámetros que se ejecutaran una única vez a lo largo del proceso.
- void loop(): Es la función principal, es un bucle infinito en donde el programador irá escribiendo todas las instrucciones que el microcontrolador repetirá una y otra vez.

Una función es un bloque de líneas de programación que tiene un tipo, un nombre, parámetros de ingreso (en algunos casos se omite), conjunto de instrucciones y devuelve (o no) un resultado. El objetivo de las funciones es el de reutilizar código a lo largo del programa, organizar el algoritmo, fácil de leer y depurar errores y reducir el número de sentencias. En la Figura 24 se puede evidenciar un ejemplo de cómo se utilizan funciones un programa en general.

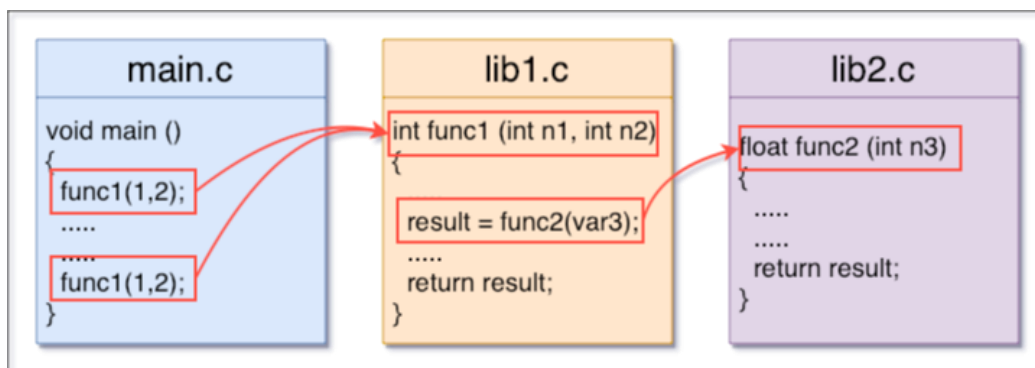


Figura 24: Uso de funciones en un programa.

Fuente: El autor.

La forma correcta para utilizar una función dentro del IDE de Arduino es:

- `tipo_de_funcion nombre_de_funcion (parámetros_de_ingreso) {acción_a_ejecutar retorno_de_valor}` (Figura 25) (Banzi & Shiloh, 2022; Boxall, 2021; Minns, 2013).

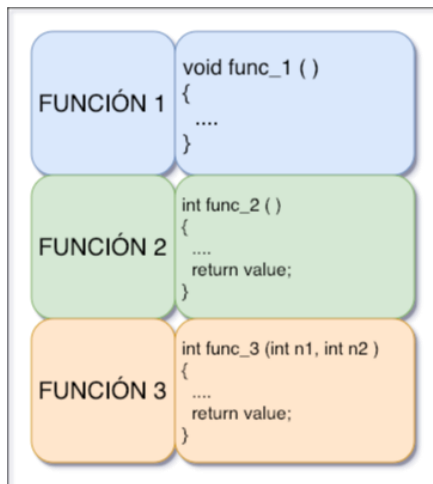


Figura 25: Ejemplo de función.

Fuente: El autor.

3.2.2 Proteus Design Suite.

El proceso previo a la implementación de cualquier proyecto electrónico es la simulación. Labcenter Electronics Ltd. es la empresa propietaria del software Proteus Design Suite. Simplemente conocido como Proteus, es un entorno de diseño que presenta al usuario una gran cantidad de dispositivos eléctricos y electrónicos que existen en el mercado, con el objetivo de realizar simulaciones a nivel de software como de hardware (Breijo, 2012).

La Figura 26 presenta la interfaz principal de Proteus; se puede evidenciar distintos dispositivos electrónicos tales como: sensores de gases, sonido, temperatura y humedad, presión, resistencias, potenciómetros, fuentes de alimentación, módulos de almacenamiento y un microcontrolador Arduino del tipo Mega. Dependiendo de cada sensor hay diferentes comunicaciones entre dispositivos y microcontrolador: I2C, SPI y Serial. Una vez que en esta plataforma el circuito cumple con los objetivos se puede realizar la implementación física de dispositivos electrónicos.

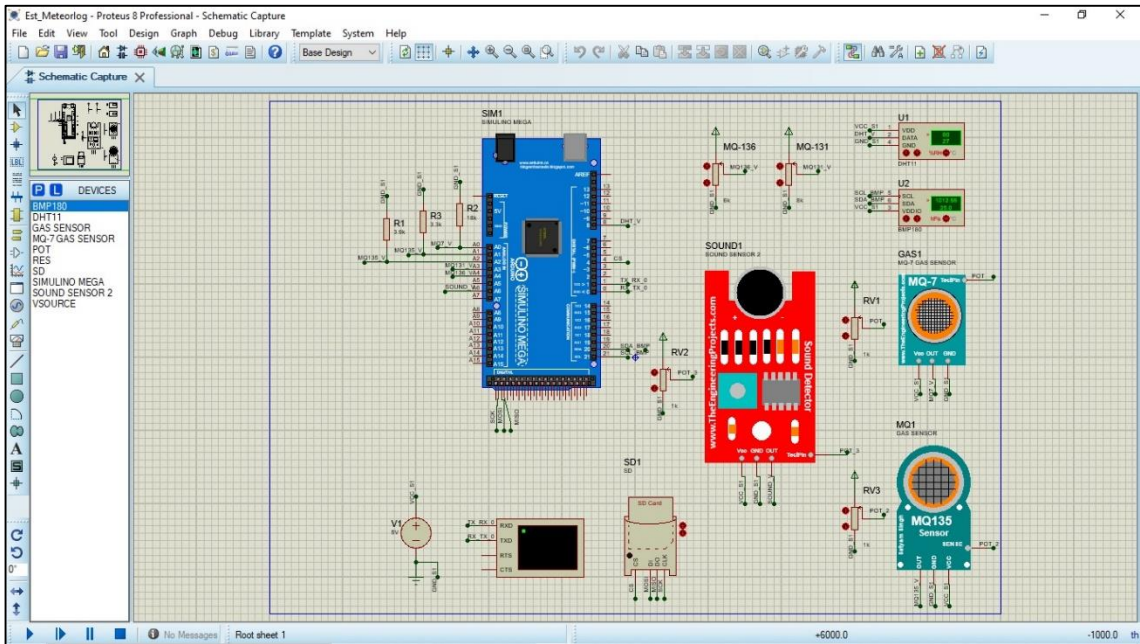


Figura 26: Interfaz gráfica de Proteus.

Fuente: El autor.

3.2.3 Eagle PCB Design and Electrical Schematic Software.

En varias ocasiones la conexión entre microcontrolador, sensores y actuadores requiere de dispositivos electrónicos intermedios como: resistencias, condensadores, pulsantes, cableado, circuitos integrados, entre otros; la integración de estos elementos genera lo que se conoce como PCB⁸. Un PCB es un dispositivo compuesto por trazados de un material conductor sobre una base aislante para realizar interconexiones entre dispositivos electrónicos. Las ventajas de utilizar una PCB en los proyectos electrónicos son muchas: conexiones fijas, menos propenso a ruido externo, disminuye la probabilidad de fallas, entre otros. Eagle PCB Design es un software diseñado por Autodesk que permite la creación de dispositivos PCB muy similar a Proteus (Torres-Ortega, 2014). La Figura 27 muestra lo que se conoce como “circuito esquemático”.

⁸ Printed circuit board – Tarjeta de circuito impreso.

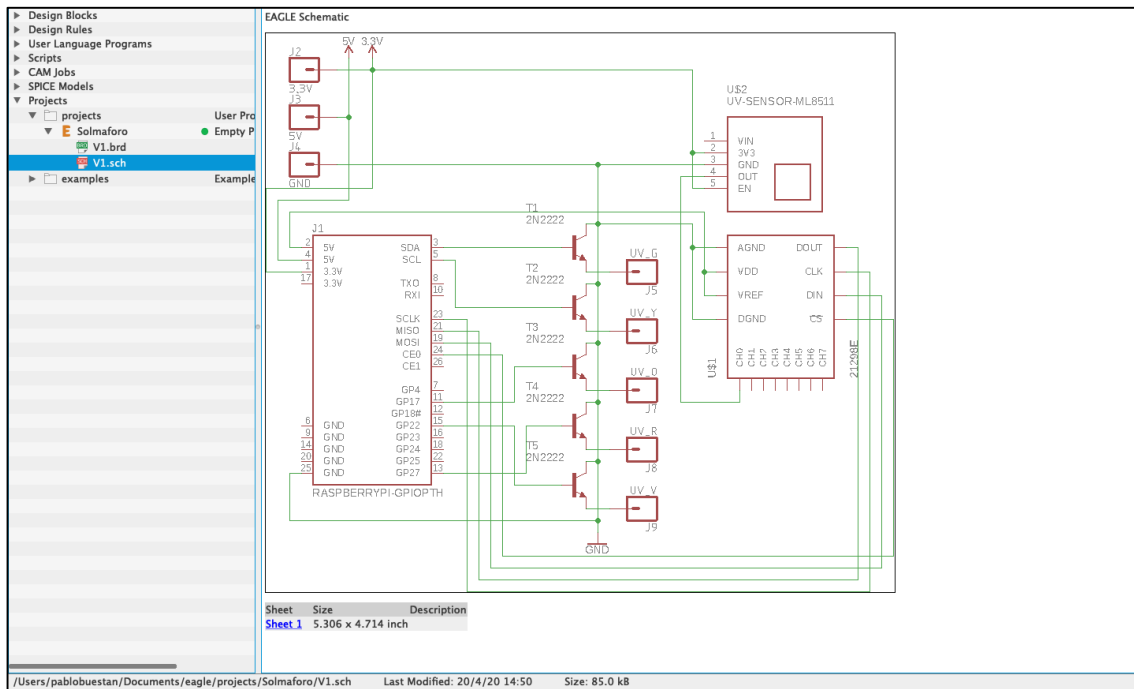


Figura 27: Interfaz gráfica de Eagle PCB Design.

Fuente: El autor.

Eagle presenta una librería muy extensa de componentes electrónicos; de igual manera que el IDE de Arduino, existen librerías creados por terceros que pueden ser encontrados en la extensa comunidad existente en la web. El usuario debe diseñar la interconexión de dispositivos, nombrar líneas de conexiones, colocar fuentes de alimentación y con las herramientas de Eagle certificar que no existe cortocircuito, o entradas/salidas sin conectar. Una vez que el usuario ha comprobado que las conexiones en el circuito esquemático son correctas se genera el “layout”. El layout es la representación digital lo más cercana posible a como quedarían los componentes en la PCB, de acuerdo con las dimensiones, posición y pistas conductoras. La Figura 28 muestra un layout antes de dimensionar la placa PCB, organizar los elementos y dibujar las conexiones electrónicas.

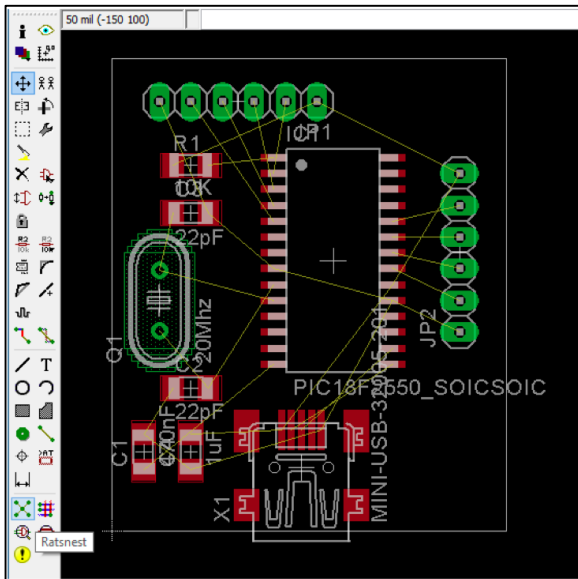


Figura 28: Agrupación de elementos electrónicos.

Fuente: (Torres-Ortega, 2014).

La Figura 29A muestra un layout completo y mediante procesos químicos se consigue la Figura 29B.

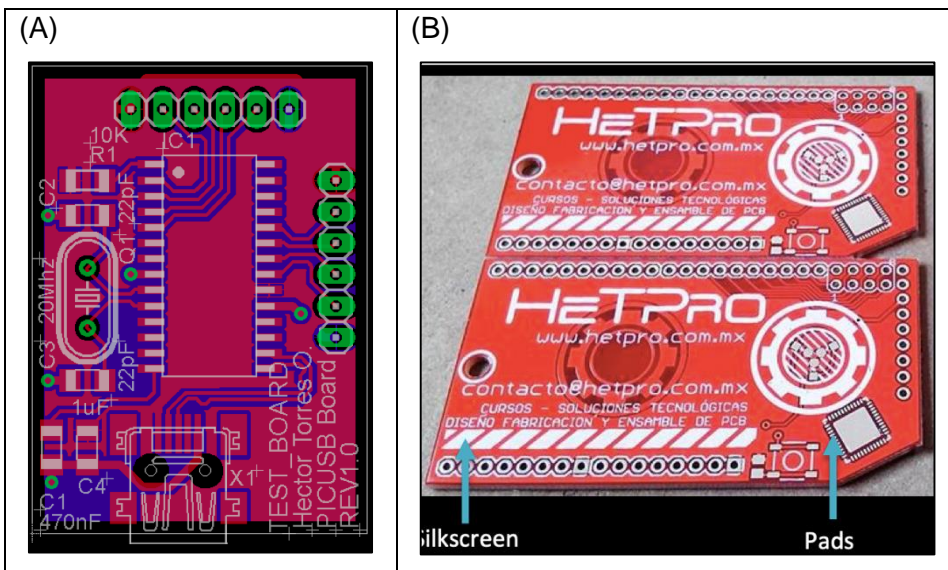


Figura 29: Layout terminado y placa PCB impresa, A) Circuito digitalizado, B) Circuito físico.

Fuente: (Torres-Ortega, 2014).

3.3 Análisis técnico – táctico en el judo.

Los autores de (Gras-Consuegra et al., 2021) definen al análisis técnico – táctico como un par dinámico de preparación que busca lograr soluciones a los problemas físicos e intelectuales encontrados en las competencias o en los entrenamientos. Dichos autores han realizado una investigación que busca encontrar errores de ejecución en el nague-waza de las técnicas base, de enlace y combinadas, mediante el criterio de un

controlador que visualiza y controla los pasos deficientes o correctos y emite un puntaje de evaluación, utilizando la metodología del modelo de Matveiv.

Según (García, 2018), el análisis técnico – táctico permite a los entrenadores encontrar las fortalezas y debilidades, tanto de sus alumnos como de sus adversarios, observando la ejecución de técnicas en específico o patrones como: dominancia del oponente, posición con respecto al rival, zonas de ataque y movimientos preferidos, tanto en el entrenamiento como en el combate. En esta investigación se hace una revisión de la literatura para entender cuáles son las metodologías actuales implementadas para el análisis técnico – táctico, concluyendo que la eficacia en el combate y el éxito en el empleo de distintos movimientos depende de la atención a las fases que suceden a lo largo del combate, a la duración y a las acciones requeridas durante la lucha.

En (Murillo Saucedo, 2016) se define al análisis técnico – táctico como la capacidad para la corrección de errores en agarres, posturas, combinaciones, ataques y defensas de judoca en un combate o entrenamiento. El propósito de esta investigación es aprender a impartir y controlar una sesión de entrenamiento de judo aplicando distintas metodologías, una de ellas el análisis técnico – táctico, evaluando el dominio, destreza y eficacia y ejecución en las distintas técnicas; los factores que se analizan son: ataques directos e indirectos, acciones defensivas, combinaciones, contrataques, entre otros. El análisis ha permitido encontrar las fortalezas y debilidades de cada judoca y retro alimentar al entrenador con información obtenida en el entrenamiento.

La metodología actual busca solucionar los problemas como movimientos, ataques, defensas, acciones y otras características que se presentan durante el entrenamiento o en una competencia. Como se explicó en este apartado, los entrenadores corrigen estos errores posteriores al combate; se visualiza todo lo relacionado con el alumno y el contrincante y se formulan comentarios. La intervención de la electrónica y el planteamiento del prototipo de esta investigación busca aportar con información adicional y precisa al análisis técnico – táctico actual.

CAPITULO III

4. DESARROLLO.

Para encontrar el prototipo adecuado se han realizado distintos modelos que gradualmente van integrando sensores y actuadores que cumplan con las exigencias físicas, eléctricas y electrónicas. Previo a la construcción del modelo físico se han realizado simulaciones del circuito y del algoritmo programado. Se cumplen los siguientes pasos:

- Se conoce el funcionamiento del módulo MPU6050 con la integración al Arduino. Se envían los datos del sensor al microcontrolador utilizando el protocolo I²C y se muestran valores abriendo el puerto Serial del PC.
- Un segundo módulo MPU6050 es programado. Se añade un multiplexor para controlar la señal SDA de cada acelerómetro.
- Como parte de la construcción del prototipo final los datos son visualizados en la PC; sin embargo, la adquisición de valores medidos debe guardarse localmente. El módulo μ SD es agregado.
- Una PCB es diseñada con el objetivo de contrarrestar los errores en cableado que generan 11 sensores MPU6050, un multiplexor 74HC4067, un módulo μ SD y un microcontrolador Arduino.
- Se realiza el tendido del hilo conductor a lo largo del traje, se conectan los sensores en varios puntos específicos para la medición y se realizan pruebas.
- El prototipo final es implementado, se realizan pruebas finales y se publican resultados.

4.1 Prototipo v1.

El diagrama de conexión se encuentra en la Figura 30.

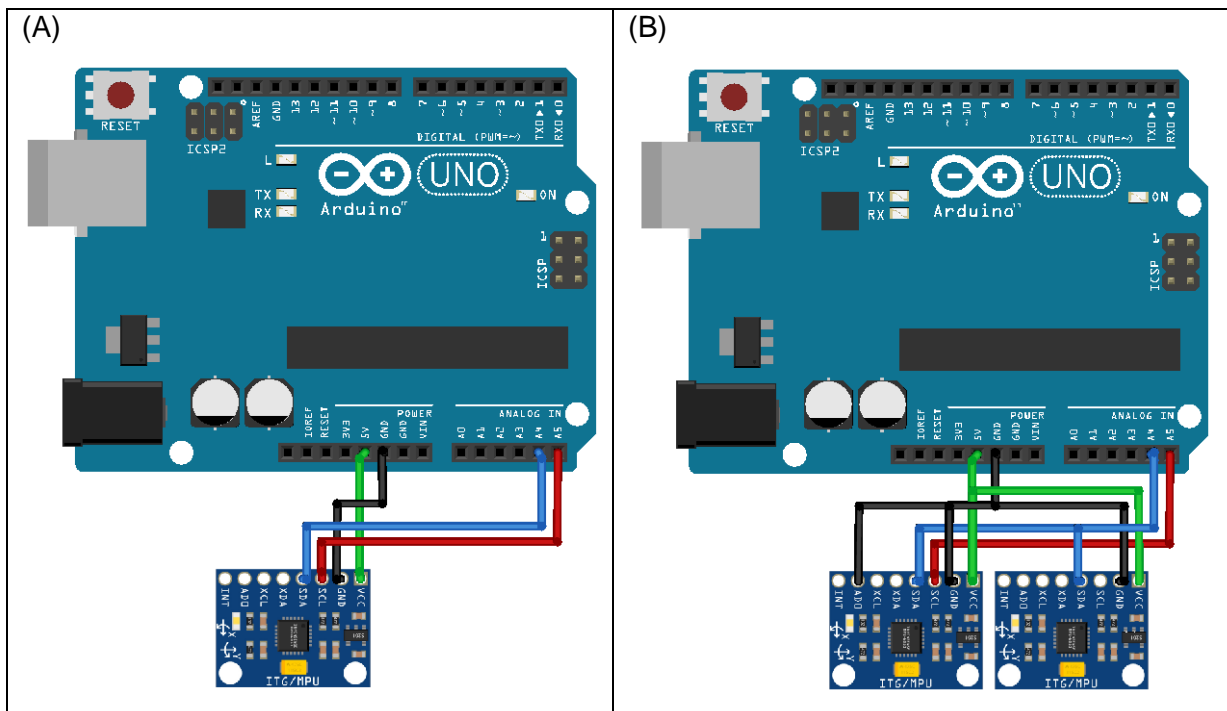


Figura 30: Diagrama de conexión del prototipo v1, A) Microcontrolador y un acelerómetro, B) Microcontrolador y dos acelerómetros.

Fuente: El autor.

En todo prototipo que incluye programación y dispositivos externos con propósitos diferentes, es recomendable realizar un algoritmo por cada módulo, realizar la configuración y adquisición de datos y luego integrar todo en un solo archivo de programación con múltiples funciones. Este primer diseño consta de un Arduino Uno (funciona de igual manera que el NodeMCU 32S) y un módulo MPU6050 como se puede ver en la Figura 30A. En la Tabla 7, se muestra la configuración de las Entradas/Salidas.

Tabla 7: Conexiones del prototipo v1.

ARDUINO	MPU6050
5V	VCC
GND	GND
A4	SDA
A5	SCL

Fuente: El autor.

Todos los módulos externos que utilicen el protocolo I²C para comunicarse con el microcontrolador disponen de una dirección hexadecimal única que viene dada por el fabricante. En el caso del MPU6050 puede ser 0x68 o 0x69, dependiendo del valor de conexión del PIN AD0.

Adquirir los datos de un solo módulo es sencillo, ya que se utiliza la librería externa `Adafruit_MPU6050.h` en el IDE de Arduino. El proceso de configuración de este dispositivo es el siguiente:

- Se crea un objeto utilizando la clase `Adafruit_MPU6050`.
- En la función `setup()` se inicializa el objeto utilizando el método `.begin()`.
- En la función `loop()` se crean variables para cada uno de los valores de X, Y, Z tanto del acelerómetro como del giroscopio y mediante el método `getEvent()` se actualizan estos valores con los datos adquiridos.
- El proceso de adquisición se repite indefinidamente ya que se encuentra dentro de la función `loop()`.

De igual manera, la lectura de dos sensores MPU6050 no presenta dificultad. Se realiza el proceso explicado anteriormente con la singularidad de conectar el PIN ADO a GND e indicar en el método `begin()` la dirección hexagonal (0x69), todo esto en el segundo módulo. La conexión se presenta en la Figura 30B.

Antes de realizar algún prototipo físico es recomendable realizar una simulación para verificar que el software y hardware cumplan con lo que se desea. Proteus tiene una amplia librería en la que se encuentra el Arduino UNO; sin embargo, hasta el momento no se dispone de un catálogo de acelerómetros, por lo que se ha utilizado un módulo que realiza otra función pero que utiliza el protocolo de comunicación I²C. La Figura 31 presenta el diagrama de conexiones en el simulador.

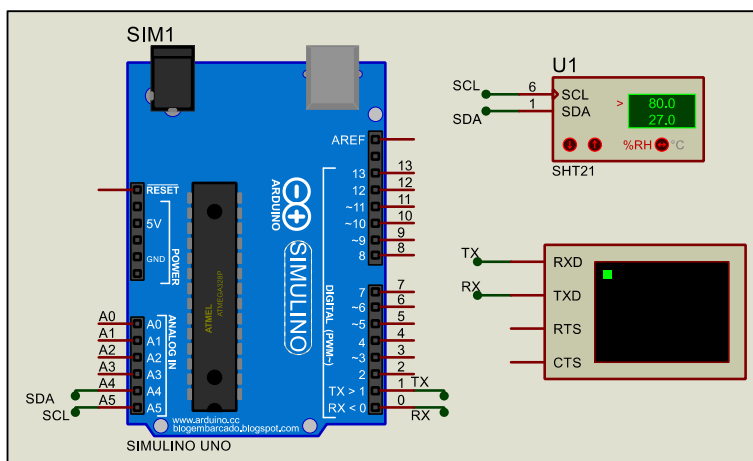
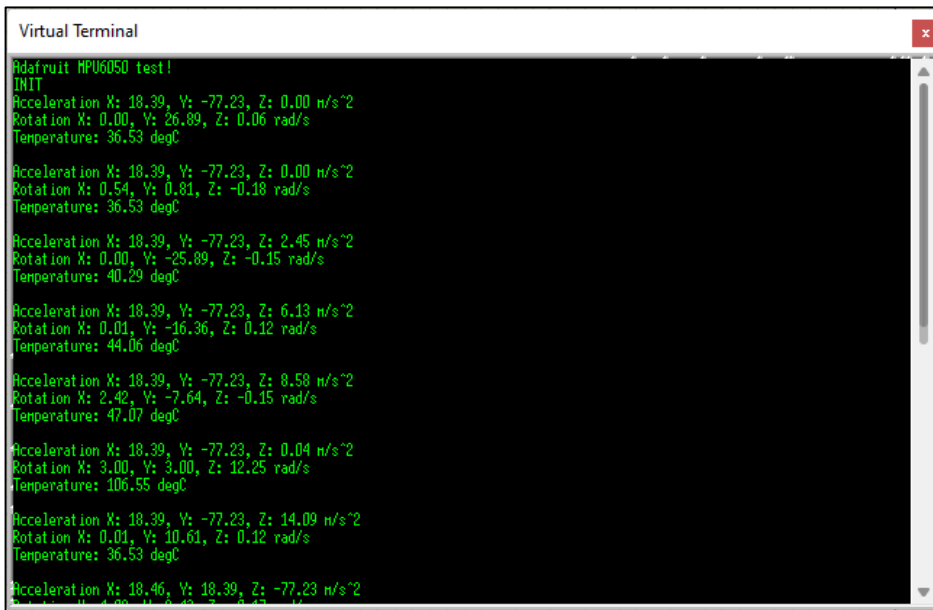


Figura 31: Simulación del prototipo v1.

Fuente: El autor.

Proteus permite la visualización de datos utilizando la comunicación Serial. Los datos adquiridos se muestran en la Figura 32:



```
Virtual Terminal
Adafruit MPU6050 test!
INIT
Acceleration X: 18.39, Y: -77.23, Z: 0.00 m/s^2
Rotation X: 0.00, Y: 26.89, Z: 0.06 rad/s
Temperature: 36.53 degC

Acceleration X: 18.39, Y: -77.23, Z: 0.00 m/s^2
Rotation X: 0.54, Y: 0.81, Z: -0.18 rad/s
Temperature: 36.53 degC

Acceleration X: 18.39, Y: -77.23, Z: 2.45 m/s^2
Rotation X: 0.00, Y: -25.89, Z: -0.15 rad/s
Temperature: 40.29 degC

Acceleration X: 18.39, Y: -77.23, Z: 6.13 m/s^2
Rotation X: 0.01, Y: -16.36, Z: 0.12 rad/s
Temperature: 44.06 degC

Acceleration X: 18.39, Y: -77.23, Z: 8.58 m/s^2
Rotation X: 2.42, Y: -7.64, Z: -0.15 rad/s
Temperature: 47.07 degC

Acceleration X: 18.39, Y: -77.23, Z: 0.04 m/s^2
Rotation X: 3.00, Y: 3.00, Z: 12.25 rad/s
Temperature: 106.55 degC

Acceleration X: 18.39, Y: -77.23, Z: 14.09 m/s^2
Rotation X: 0.01, Y: 10.61, Z: 0.12 rad/s
Temperature: 36.53 degC

Acceleration X: 18.46, Y: 18.39, Z: -77.23 m/s^2
Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
Temperature: 36.53 degC
```

Figura 32: Datos adquiridos del prototipo v1 (Simulado).

Fuente: El autor.

Una vez constatado que el algoritmo de programación y las conexiones son las correctas, se procede a realizar el prototipo físico (Figura 33).

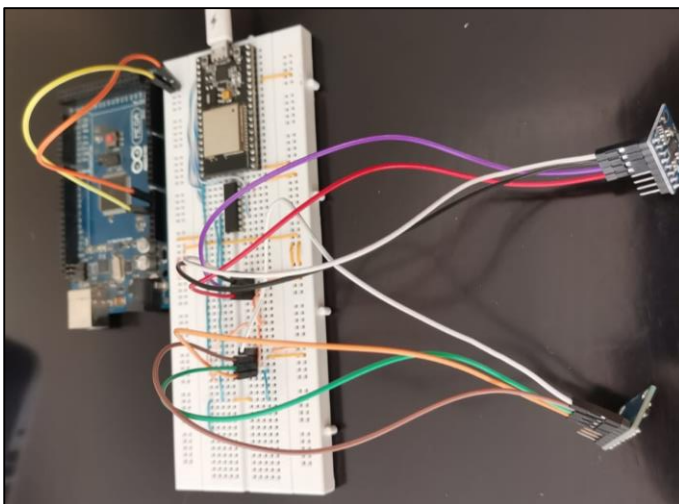


Figura 33: Esquema físico del prototipo v1.

Fuente: El autor.

Para el prototipo físico se implementó la configuración de la Figura 30B; esta es la razón por la cual, se adquieren valores de dos módulos MPU6050, tal y como se presenta en la Figura 34. El código completo se puede encontrar en el Anexo 3.

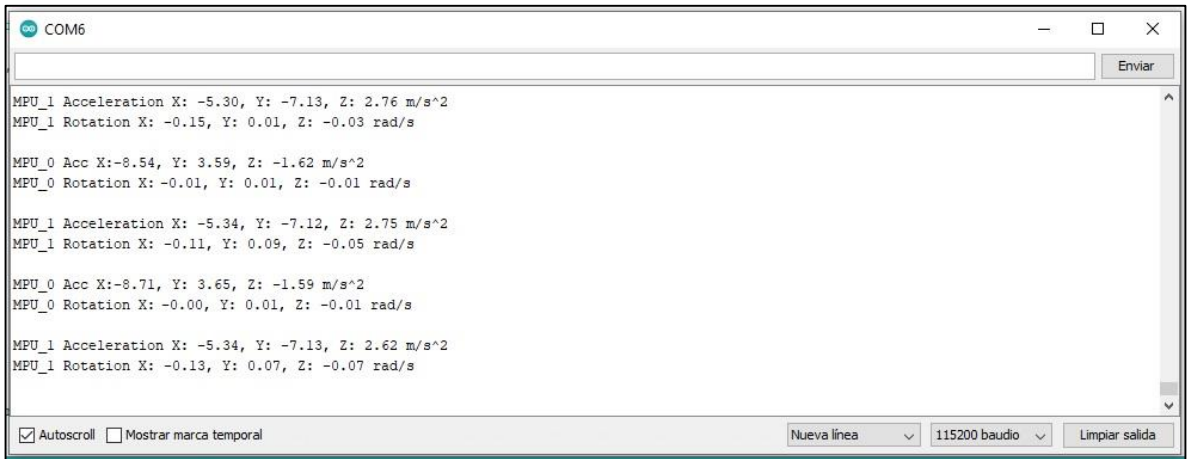


Figura 34: Datos adquiridos del prototipo v1 (Real).

Fuente: El autor.

4.2 Prototipo v2.

El diagrama de conexión se encuentra en la Figura 35.

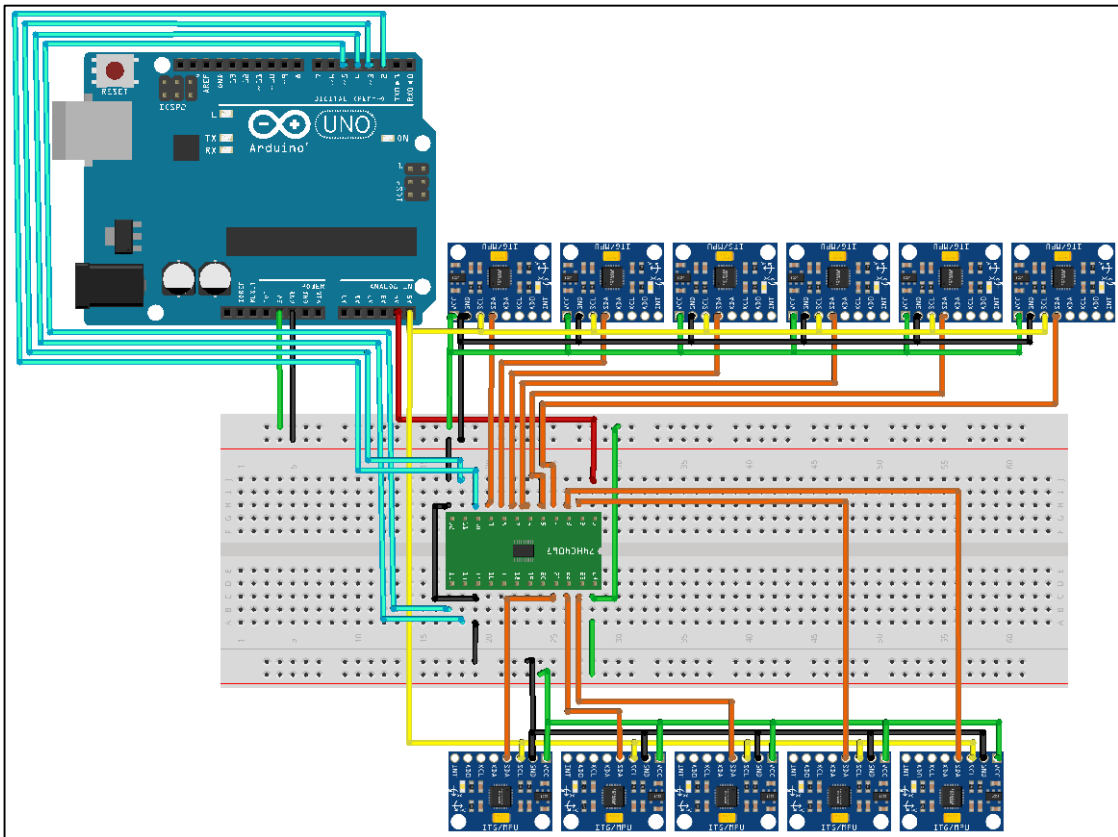


Figura 35: Diagrama de conexión del prototipo v2.

Fuente: El autor.

Al tratar de integrar un tercer sensor acelerómetro/giroscopio se produjo un error. Esto se debe a que todos los módulos MPU6050 comparten las direcciones hexagonales 0x68 y 0x69, el microcontrolador recibe las señales SDA enviadas desde los sensores y se ocasiona un conflicto al momento de configurar los distintos objetos e inicializarlos ya que las direcciones hexagonales se encuentran duplicadas. El multiplexor 74HC4067 se añade al circuito y permite que múltiples señales SDA y una única dirección hexagonal (0x68) puedan derivar en una sola salida y no ocasionen conflictos en los objetos creados. En la Tabla 8 se muestra la configuración de las Entradas/Salidas.

Tabla 8: Conexiones del prototipo v2.

ARDUINO	MPU6050	74HC4067
5V	VCC	VCC
GND	GND	GND – EN
-	SDAx	Cx (cada sensor se conecta en una única entrada Ej. SDA0 – C0)
A4	-	SIG
A5	SCL (Todos los SCL se unen)	-
D2	-	S0
D3	-	S1
D4	-	S2
D5	-	S3

Fuente: El autor.

Al tener doce señales SDA correspondientes a cada módulo MPU6050, se debe seleccionar una única señal para la adquisición de datos en el Arduino Uno. Cada SDA se ha conectado a una entrada Cx del multiplexor y dependiendo de los valores que se encuentren las entradas S0, S1, S2 y S3 (que tiene relación directa a las salidas digitales D2, D3, D4 y D5) se selecciona uno u otro acelerómetro/giroscopio. La salida SIG es la encargada de enviar los valores actuales de aceleración y velocidad angular de un sensor en específico y se conecta directamente a la entrada A4 microcontrolador. El algoritmo es explicado brevemente:

- Se crea un objeto por cada sensor MPU6050 utilizando la clase Adafruit_6050.

- En la función setup () se configuran a D2-D5 como salidas y se inicializan todos los objetos utilizando begin ().
- Para la selección de cada sensor se debe de modificar los valores de D2-D5, cumpliendo la tabla de verdad del

TRUTH TABLE

S0	S1	S2	S3	\bar{E}	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

H= High Level
L= Low Level

- Anexo 2 y permitiendo realizar doce combinaciones (desde 0000 hasta 1101).
- En la función loop() se crean variables para cada uno de los valores de X, Y, Z tanto del acelerómetro como del giroscopio (de cada uno de los 12 sensores) y mediante el método getEvent() se actualizan estos valores con los datos adquiridos.
- El proceso de adquisición se repite indefinidamente, ya que se encuentra dentro de la función loop().

La Figura 36 presenta el diagrama de conexiones en el simulador.

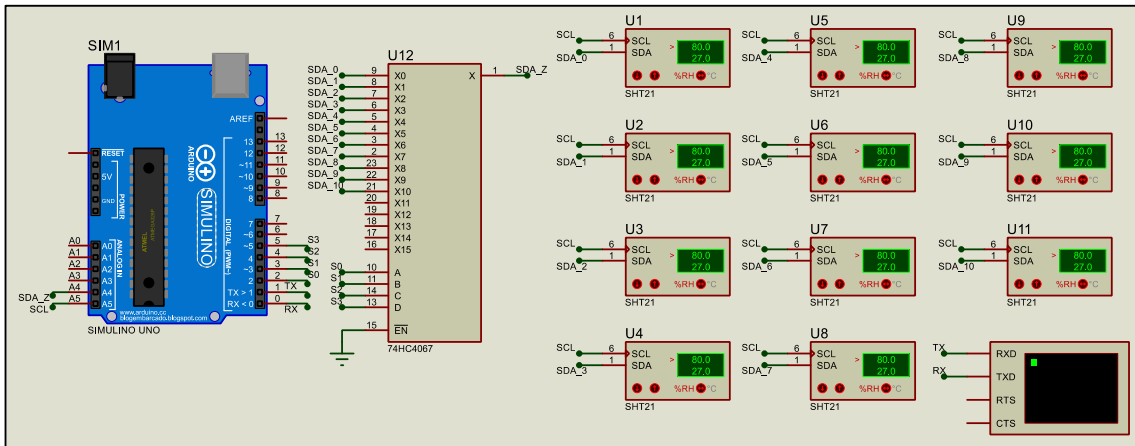


Figura 36: Simulación del prototipo v2.

Fuente: El autor.

Los datos adquiridos en Proteus se presentan en la Figura 37:

```

Virtual Terminal

mpu_6 Found!
mpu_7 Found!
mpu_8 Found!
mpu_9 Found!
mpu_10 Found!
mpu_11 Found!
MPU_6 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_6 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_7 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_7 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_8 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_8 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_9 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_9 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_10 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_10 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_11 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
  
```

Figura 37: Datos adquiridos del prototipo v2 (Simulado).

Fuente: El autor.

Una vez constatado que el algoritmo de programación y las conexiones son las correctas, se procede a realizar el prototipo físico (Figura 38).

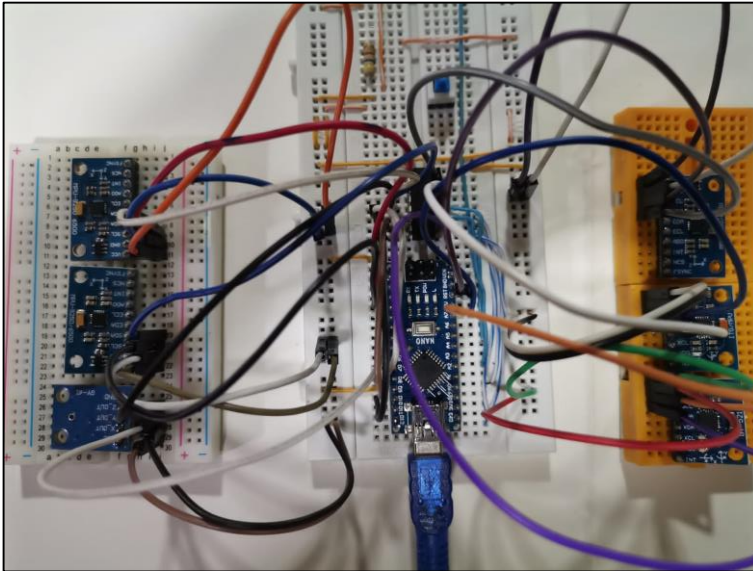


Figura 38: Esquema físico del prototipo v2.

Fuente: El autor.

Para el prototipo físico se implementaron los doce sensores. Se fue realizando la inspección del funcionamiento de cada uno, luego únicamente dos, posterior fueron tres hasta completar todos los MPU6050. Los resultados se visualizan en la Figura 39. El código completo se puede encontrar en el Anexo 4.

```
COM6
MPU_10 Rotation X: -0.14, Y: 0.02, Z: -0.03 rad/s
MPU_11 Acceleration X: 6.51, Y: 1.51, Z: -7.81 m/s^2
MPU_11 Rotation X: 0.02, Y: 0.03, Z: -0.03 rad/s
MPU_6 Acceleration X: -1.45, Y: 2.58, Z: 10.33 m/s^2
MPU_6 Rotation X: -0.01, Y: 0.01, Z: -0.01 rad/s
MPU_7 Acceleration X: 0.00, Y: -23.20, Z: 16.55 m/s^2
MPU_7 Rotation X: -1.30, Y: 1.36, Z: 0.00 rad/s
MPU_8 Acceleration X: 3.61, Y: -4.21, Z: 7.62 m/s^2
MPU_8 Rotation X: -0.01, Y: 0.01, Z: -0.01 rad/s
MPU_9 Acceleration X: 4.34, Y: 7.43, Z: 4.32 m/s^2
MPU_9 Rotation X: 0.00, Y: -0.04, Z: 0.01 rad/s
```

Figura 39: Datos adquiridos del prototipo v2 (Real).

Fuente: El autor.

4.3 Prototipo v3.

El diagrama de conexión se encuentra en la Figura 40.

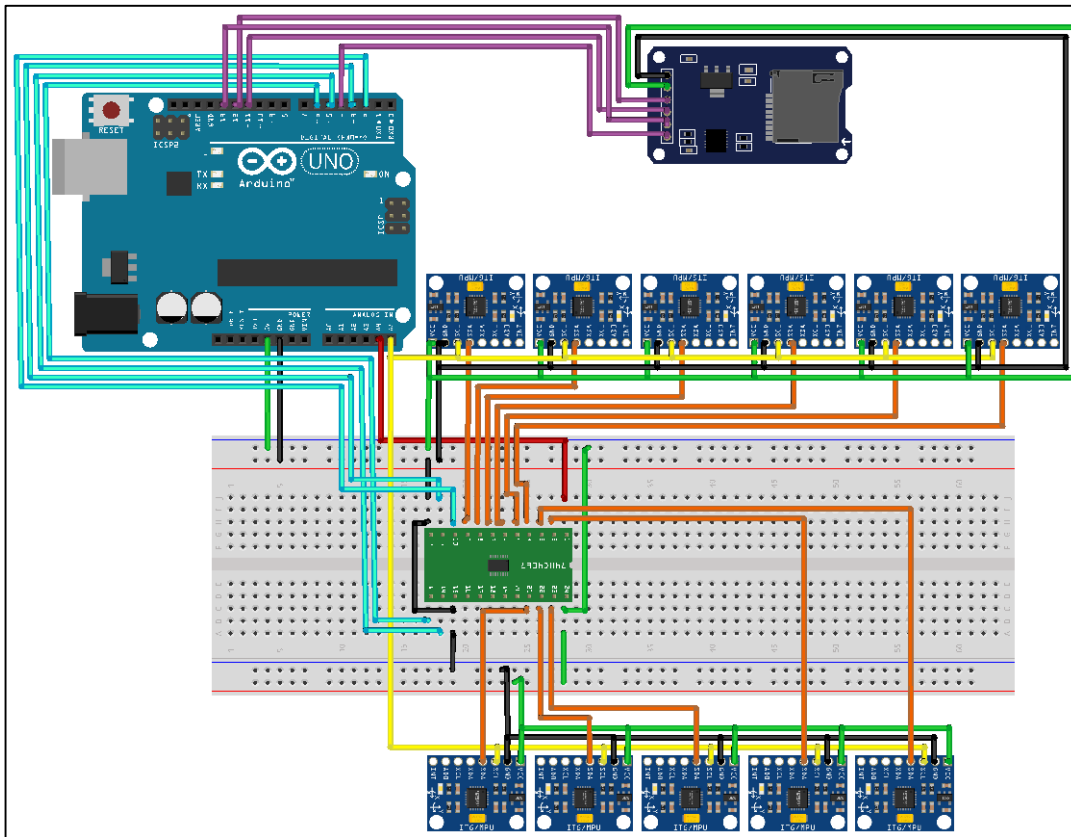


Figura 40: Diagrama de conexión del prototipo v3.

Fuente: El autor.

Los errores que más se presentan en el desarrollo de un circuito eléctrico y electrónico es el cableado, las conexiones, los empalmes y el protoboard. Estos inconvenientes son normales en la fase de desarrollo de un prototipo, pero por ningún motivo pueden presentarse en el producto final. Para eso, se ha diseñado una PCB que permita interconectar sensores, fuentes de voltaje y actuadores reduciendo el cableado, empalmes, ruido eléctrico y gestionando de una mejor manera la distribución de los dispositivos electrónicos. Utilizando EAGLE se realiza el diseño esquemático presentado en la Figura 41:

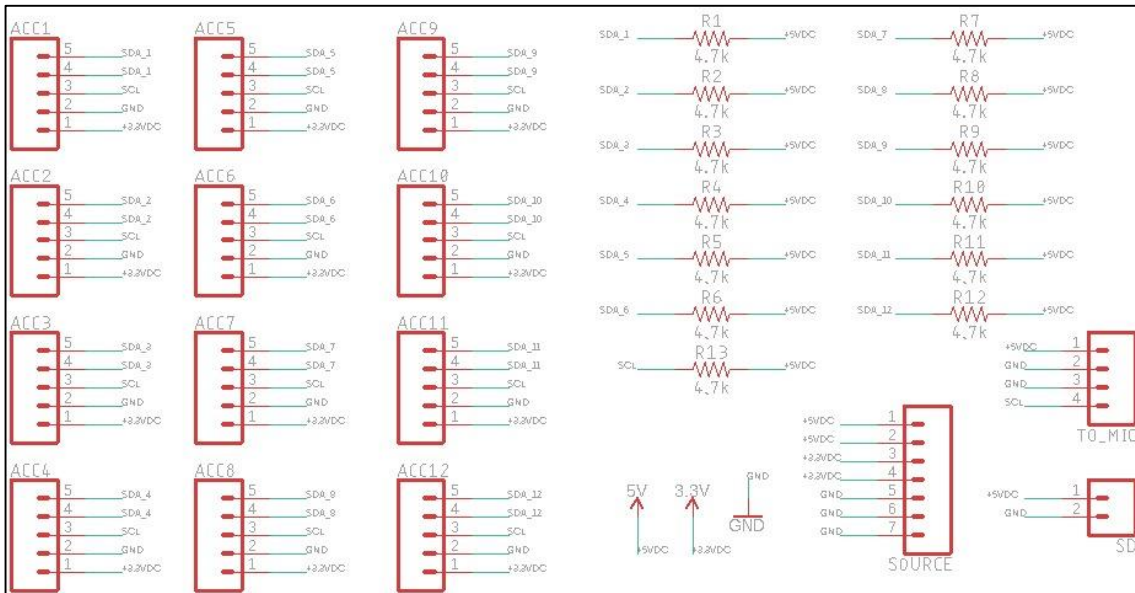


Figura 41: Diagrama esquemático de la PCB diseñada.

Fuente: El autor.

Se muestra los distintos dispositivos electrónicos que forman parte de la PCB. Se tiene una resistencia (R1-R12) de 4.7kΩ por cada MPU6050, que permite la comunicación mediante el protocolo I²C. Los elementos ACC1-ACC12 corresponden a peinetas de conexión tipo macho, destinados para la conexión de cada acelerómetro/giroscopio. Se tiene SOURCE, SD y TO_MIC que son de igual manera peinetas destinados para la alimentación del microcontrolador y módulo de almacenamiento. Con este diagrama se diseña el PCB, que se muestra en la Figura 42.

(A)	(B)
-----	-----

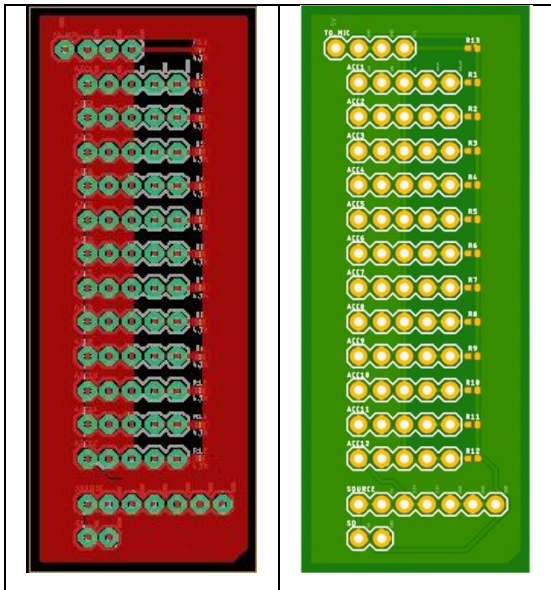


Figura 42: PCB diseñada, A) Versión digitalizada, B) Versión pre-impresión.

Fuente: El autor.

La Figura 42A muestra el resultado de colocar los agujeros donde se soldarán las peinetas y las resistencias y la capa de GND, importante para evitar el ruido y la creación de pistas de cobre. En la Figura 42B, se presenta el diseño final listo para enviar a imprimir. La PCB obtenida físicamente se visualiza en la Figura 43.

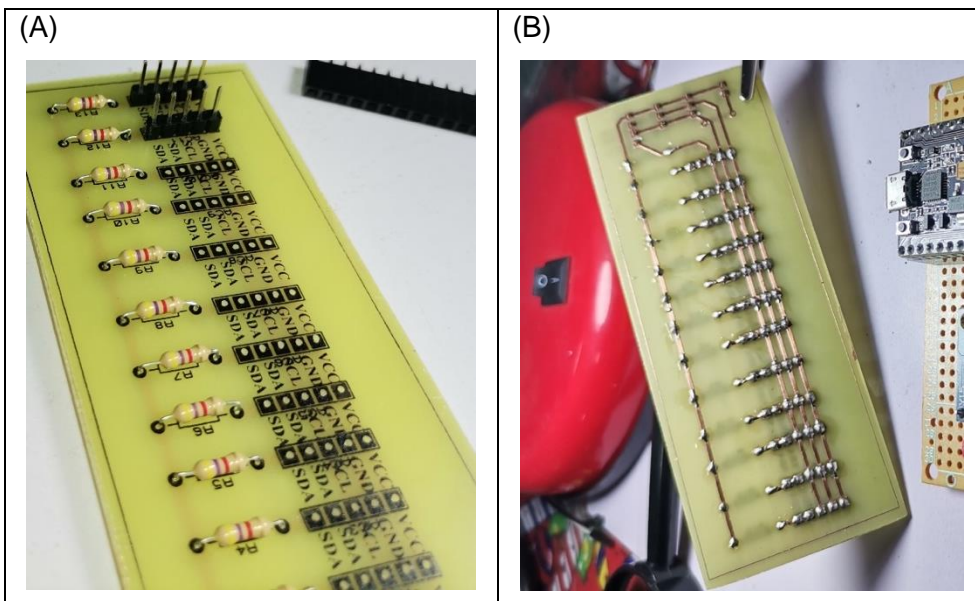


Figura 43: PCB diseñada para el prototipo v3, A) Vista frontal, B) Vista posterior.

Fuente: El autor.

Adicional a la PCB diseñada se ha integrado el módulo μ SD que guardará los valores de cada uno de los sensores MPU6050. Por último, utilizando un hilo conductor se ha bordado todas las conexiones destinadas para VCC, GND, SDA y SCL de cada

uno de los acelerómetros/giroscopio. En la Tabla 9, se muestra la configuración de las Entradas/Salidas.

Tabla 9: Conexiones del prototipo v3.

ARDUINO	MPU6050	74HC4067	MÓDULO μ SD
5V	VCC	VCC	VCC
GND	GND	GND – EN	GND
-	SDAx	Cx	-
A4	-	SIG	-
A5	SCL	-	-
D2	-	S0	-
D3	-	S1	-
D5	-	S2	-
D6	-	S3	-
D4	-	-	CS
D11	-	-	MOSI
D12	-	-	MISO
D13	-	-	CSK

Fuente: El autor.

La integración del módulo μ SD obliga a realizar una modificación en las salidas digitales asignadas al multiplexor 74HC4067. De esta manera, los datos son almacenados en la tarjeta μ SD en un formato .txt. La Figura 44 presenta el diagrama de conexiones en el simulador.

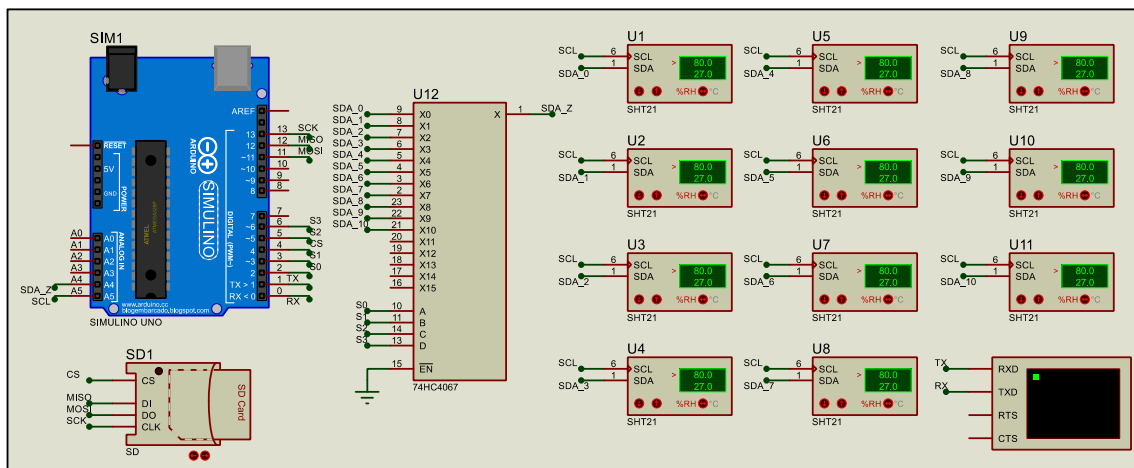


Figura 44: Simulación del prototipo v3.

Fuente: El autor.

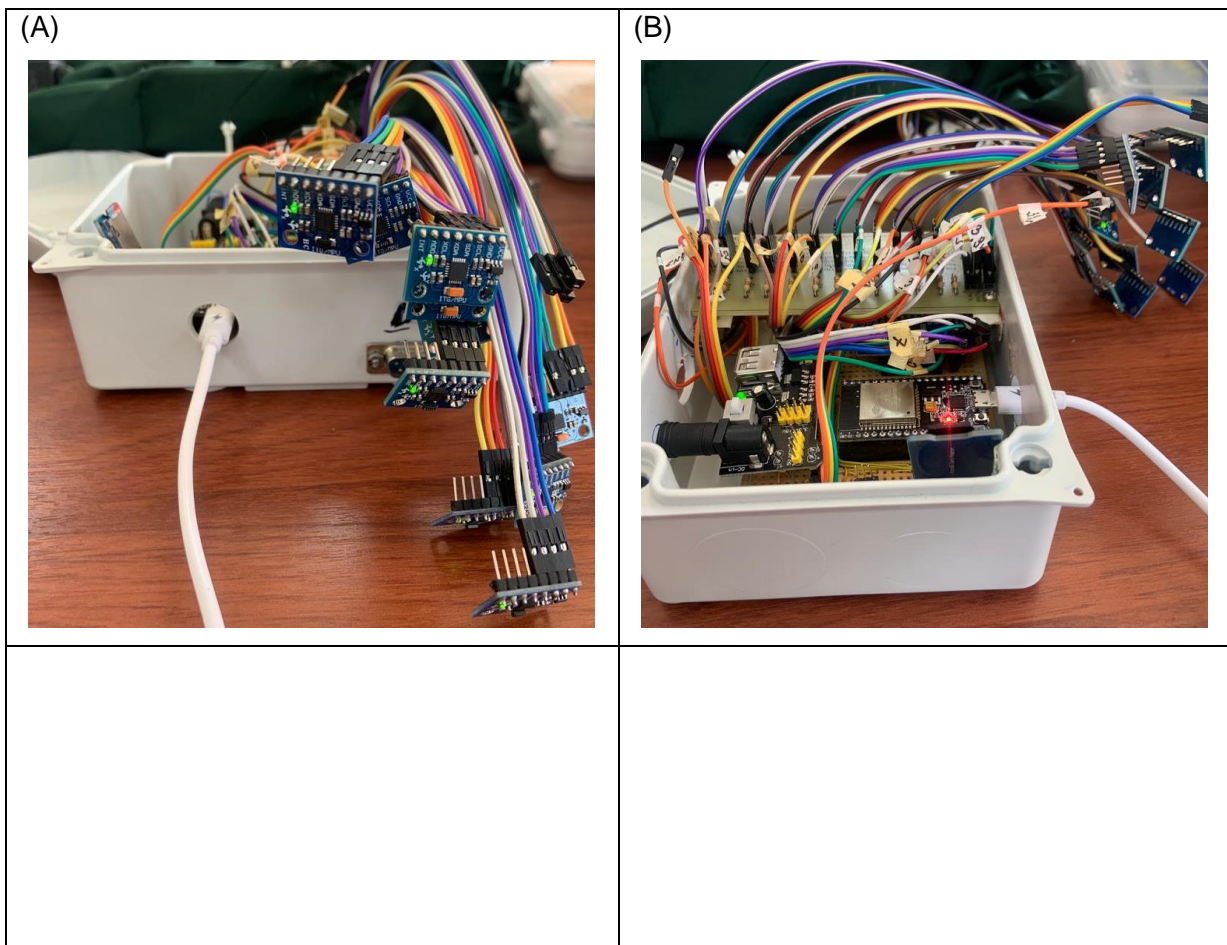
Los datos adquiridos en Proteus están representados en la Figura 45.

```
Virtual Terminal
mpu_6 Found!
mpu_7 Found!
mpu_8 Found!
mpu_9 Found!
mpu_10 Found!
mpu_11 Found!
MPU_6 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_6 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_7 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_7 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_8 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_8 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_9 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_9 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_10 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
MPU_10 Rotation X: 0.00, Y: 0.00, Z: 0.00 rad/s
MPU_11 Acceleration X: 0.00, Y: 0.00, Z: 0.00 m/s^2
```

Figura 45: Datos adquiridos del prototipo v3 (Simulado).

Fuente: El autor.

Una vez constatado que el algoritmo de programación y las conexiones son las correctas, se procede a realizar el prototipo físico (Figura 46).



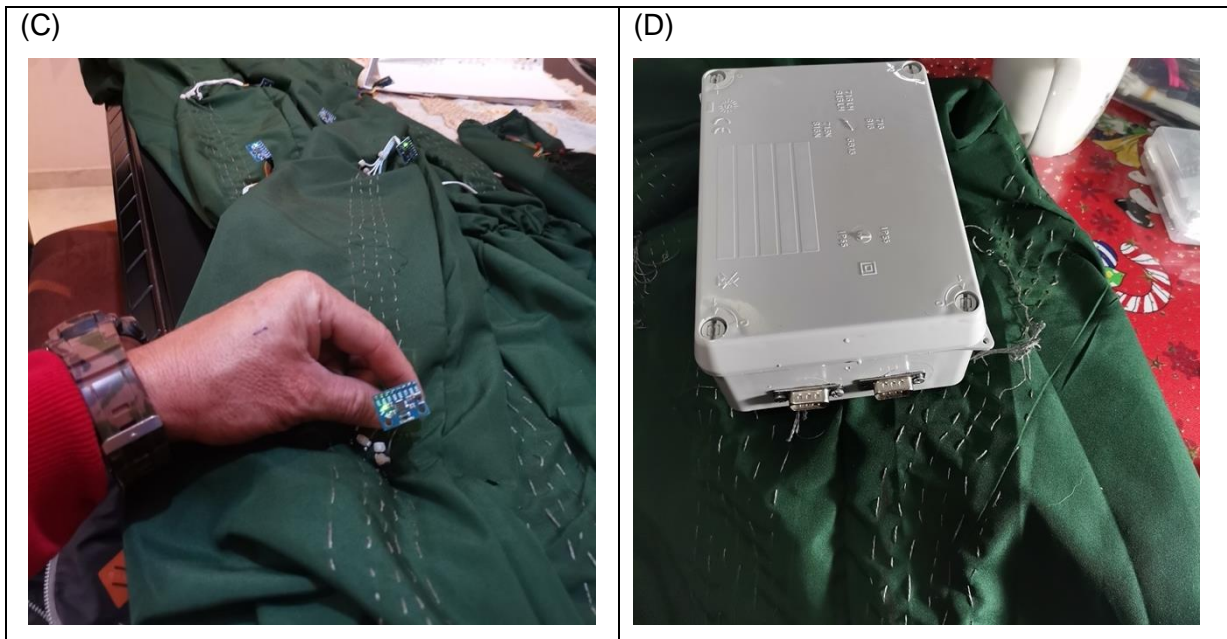


Figura 46: Esquema físico del prototipo v3, A) Sistema de acelerómetros, B) Sistema electrónico completo, C) Acelerómetro implementado en el traje, D) Caja de conexiones.

Fuente: El autor.

Se obtuvieron resultados de seis sensores MPU sin inconvenientes como se muestra en la Figura 47.

```

COM6
MPU_10 Rotation X: -0.14, Y: 0.02, Z: -0.03 rad/s
MPU_11 Acceleration X: 6.51, Y: 1.51, Z: -7.81 m/s^2
MPU_11 Rotation X: 0.02, Y: 0.03, Z: -0.03 rad/s
MPU_6 Acceleration X: -1.45, Y: 2.58, Z: 10.33 m/s^2
MPU_6 Rotation X: -0.01, Y: 0.01, Z: -0.01 rad/s
MPU_7 Acceleration X: 0.00, Y: -23.20, Z: 16.55 m/s^2
MPU_7 Rotation X: -1.30, Y: 1.36, Z: 0.00 rad/s
MPU_8 Acceleration X: 3.61, Y: -4.21, Z: 7.62 m/s^2
MPU_8 Rotation X: -0.01, Y: 0.01, Z: -0.01 rad/s
MPU_9 Acceleration X: 4.34, Y: 7.43, Z: 4.32 m/s^2
MPU_9 Rotation X: 0.00, Y: -0.04, Z: 0.01 rad/s

```

Figura 47: Datos obtenidos del prototipo v3.

Fuente: El autor.

Al conectar el acelerómetro/giroscopio número siete el sistema dejaba de inicializar correctamente los sensores. Luego de una serie de depuraciones tanto de software como de hardware, se evidenció que el hilo conductor presentaba una resistencia alta a medida que la distancia aumentaba entre la central y el sensor. Al no recibir la señal SDA y SCL de los MPU6050 en la función setup() el sistema ingresa a un bucle infinito de manera que no se adquieran datos erróneos de medición. Se optó por utilizar un cable conductor de 4 hilos de cobre que reemplaza a todo el hilo conductor bordado en el traje. Además, para evitar guardar datos cuando el sujeto se encuentra

estático se ha añadido un pulsante, ubicado en la manga derecha del traje, que permite grabar (o no) los datos adquiridos. El resultado se presenta en la Figura 48:

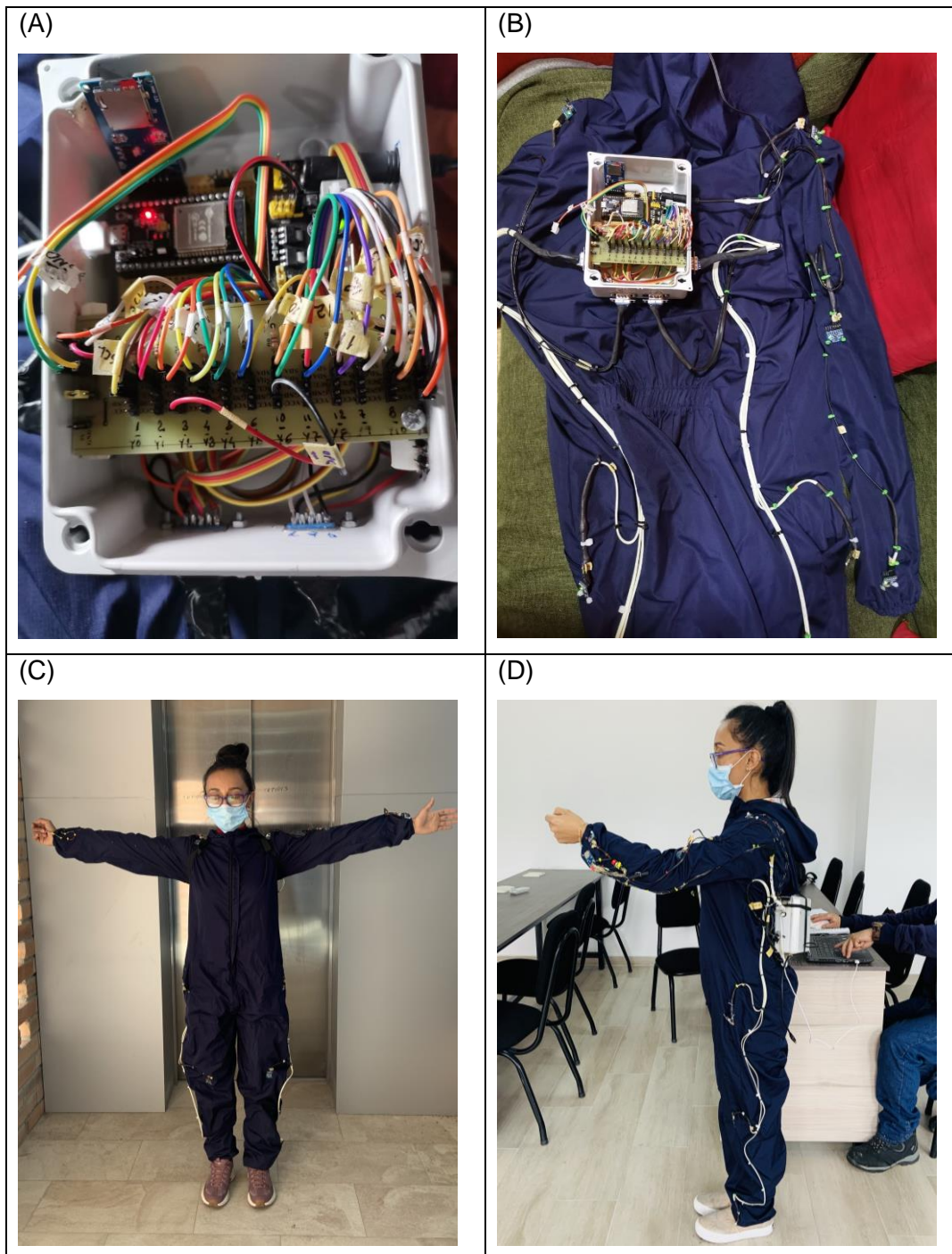


Figura 48: Implementación final en el traje, A) Sistema electrónico de medición, B) Sistema electrónico y traje con acelerómetros, C) Judoca con traje de medición vista frontal, D) Judoca con traje de medición vista lateral.

Fuente: El autor.

5. ANÁLISIS DE RESULTADOS Y DISCUSIÓN.

5.1 Adquisición de datos.

La disposición de los sensores alrededor del traje se ve reflejado en el Anexo 5. El sistema de medición de aceleraciones cumple los siguientes pasos:

- El prototipo desarrollado está alimentado por una única fuente de poder para todo el sistema, tanto los sensores, los actuadores y el microcontrolador.
- Una vez que ha sido energizado el prototipo, se inicializan todos los periféricos y se realiza un “barrido” de sensores con el objetivo de encontrar errores de lectura y escritura previo a la adquisición. Si no se han presentado errores, se continua con el siguiente paso caso contrario el sistema se detiene hasta que la falla sea detectada y solucionada.
- Posterior a este acontecimiento, el sistema de almacenamiento es iniciado e inmediatamente comienza la adquisición de aceleraciones en cada uno de los sensores.
- Para no obtener valores erróneos por movimientos involuntarios del judoca antes y después de las pruebas, el atleta debe de mantener pulsado el botón que se encuentra en la manga derecha. El judoca deja de presionar este pulsante botón únicamente dejándolo sin presionar cuando se esté ejecutando la técnica a analizar.
- En ese lapso los valores medidos en todos los sensores son guardados en la memoria de almacenamiento μ SD. El proceso de medición culmina en este momento.

Las características para las pruebas realizadas son las siguientes:

- La técnica realizada se le conoce como Uchi mata y la ejecución se puede ver en la Figura 49. El Tori⁹ agarra el cuello del Uke¹⁰, gira su cuerpo hacia el Uke y lo levanta a una altura por encima de la oreja. La idea de esta técnica es desestabilizar al Uke en diagonal el frente y luego lo lanza hacia abajo (Cumming Martial Arts Center, 2018).

⁹ Atleta que realiza la proyección o ejecuta la técnica.

¹⁰ Judoca que es derribado en la ejecución de la técnica.

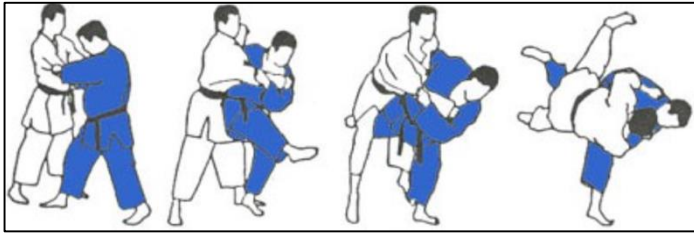


Figura 49: Uchi mata clásico.

Fuente: (Cumming Martial Arts Center, 2018).

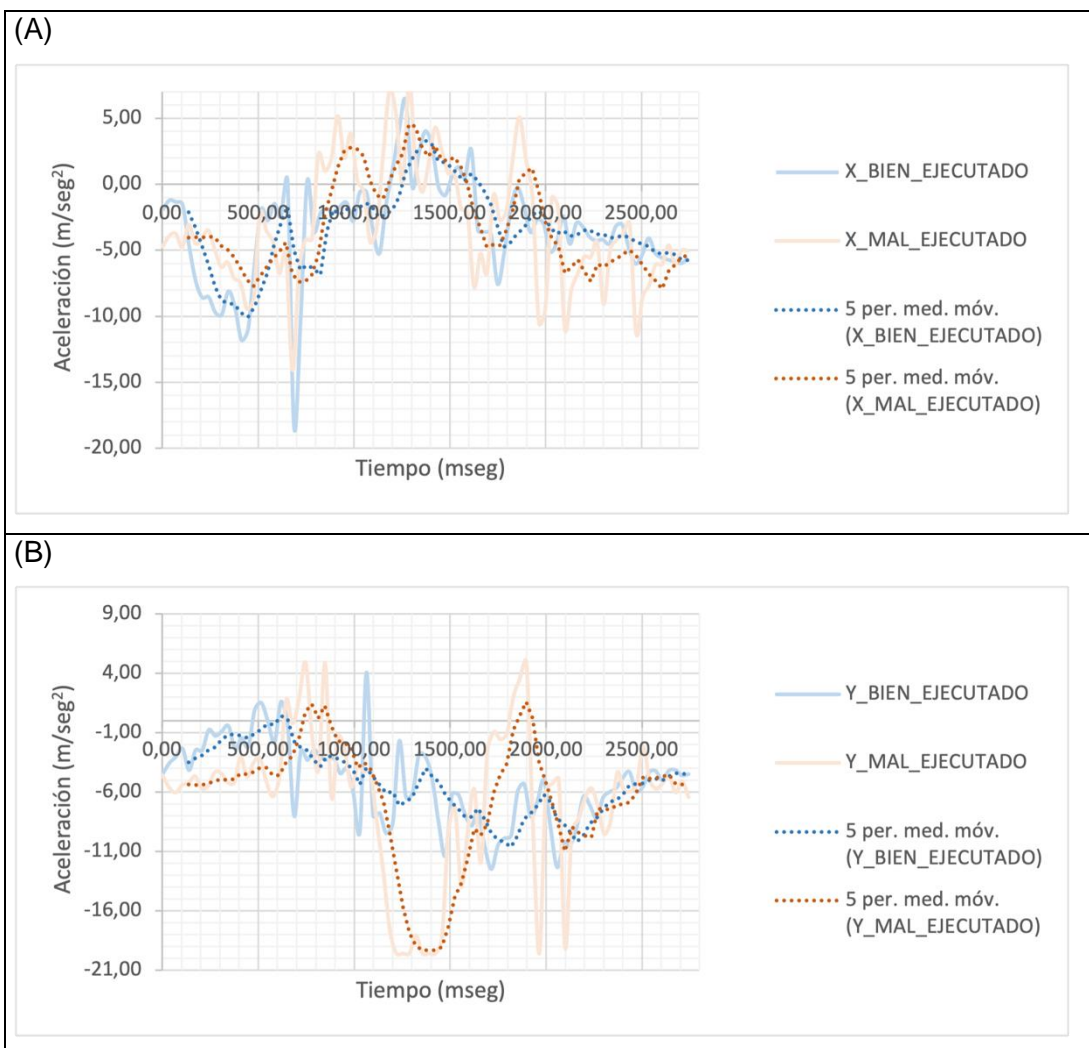
- La judoca realizó la técnica Uchi mata de una manera incorrecta (Figura 50A y Figura 50B) y correcta (Figura 50C y Figura 50D).



Figura 50: Pruebas de la técnica Uchi mata, A) Inicio de Uchi mata ejecutado erróneamente; B) Final de Uchi mata ejecutado erróneamente; C) Ejecución de Uchi mata ejecutado correctamente; D) Final de Uchi mata ejecutado correctamente.

Fuente: El autor.

- Cada prueba realizada duro entre uno y dos segundos aproximadamente.
- Se analiza un único sensor por ejecución, ya sea realizada correcta o incorrectamente.
- En total se obtuvieron 2952 datos de los sensores MPU_0, MPU_1, MPU_2, MPU_9, MPU_10 y MPU_11, en un total de 12 pruebas realizadas.
- Con datos obtenidos se ha realizado una comparación entre las pruebas ejecutadas correcta e incorrectamente entre los ejes X, Y, Z de cada uno de los sensores. Se hace un análisis completo del comportamiento de las líneas de tendencia de las aceleraciones de cada uno de los ejes en los distintos sensores que integran las 12 pruebas, obteniendo lo siguiente:
 - MPU_0: Este sensor está ubicado cerca del hombro izquierdo según el Anexo 5. La Figura 51 muestra los resultados obtenidos.



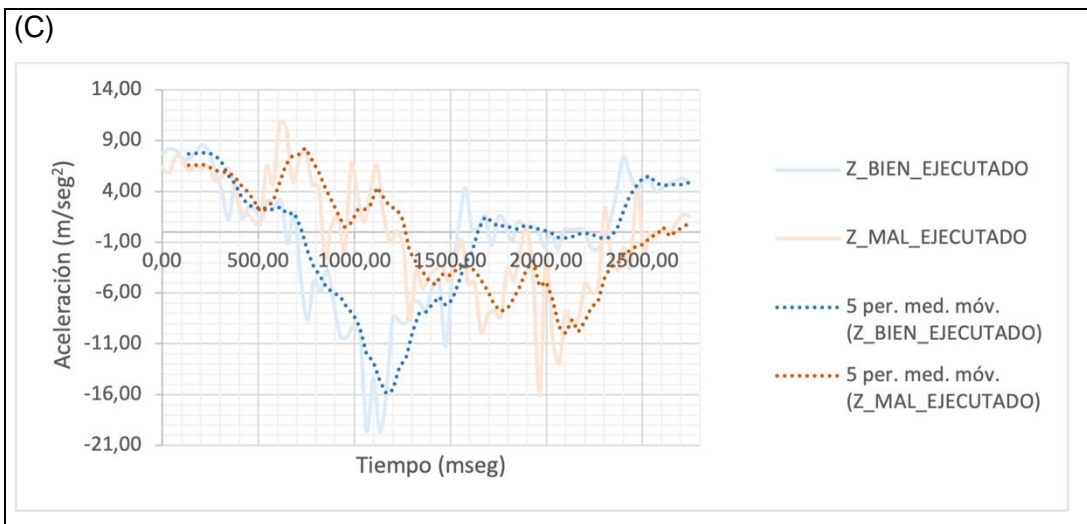


Figura 51: Resultados obtenidos en el sensor MPU_0, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

- En la Tabla 10 se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_0.

Tabla 10: Análisis de resultados obtenidos en el sensor MPU_0.

MPU_0
ACELERACIONES OBTENIDAS
<p>Con relación a las aceleraciones obtenidas en este sensor, se puede apreciar que en el eje Y se ha provocado la mayor aceleración posible con un valor aproximado de 20m/seg² en la técnica mal ejecutada. En el eje Z, también ocurre algo parecido. En un instante de tiempo, el valor de la aceleración ha llegado a ser 16m/seg² cuando ha sido bien ejecutado.</p> <p>Según el criterio de la judoca, al momento de realizar la prueba errónea, intencionalmente realizó un movimiento en sentido contrario a la dirección a la cual el oponente está descendiendo, producto de la técnica, mientras que, en una ejecución correcta el eje donde más se debe generar una fuerza es en Z; ya que, iniciada la técnica, se ha realizado un levantamiento de los hombros, producto de la fuerza que necesita la Tori para aplicar el Uchi mata.</p>
TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA
<p>En este sentido, el eje X tiene una similitud entre las dos pruebas realizadas. Esto quiere decir que al realizar las técnicas correcta y erróneamente, este eje no se vio afectado. En el eje Y, las gráficas tienden a seguir una misma trayectoria al inicio y al final, mientras que durante el transcurso ya pierde el sentido, debido a la intencionalidad de la judoca de aplicar movimientos no acordes a la técnica. De igual manera ocurre lo mismo en el eje Z.</p>

Fuente: El autor.

- MPU_1: Este sensor está ubicado cerca del codo izquierdo según el Anexo 5. La Figura 52 muestra los resultados obtenidos.

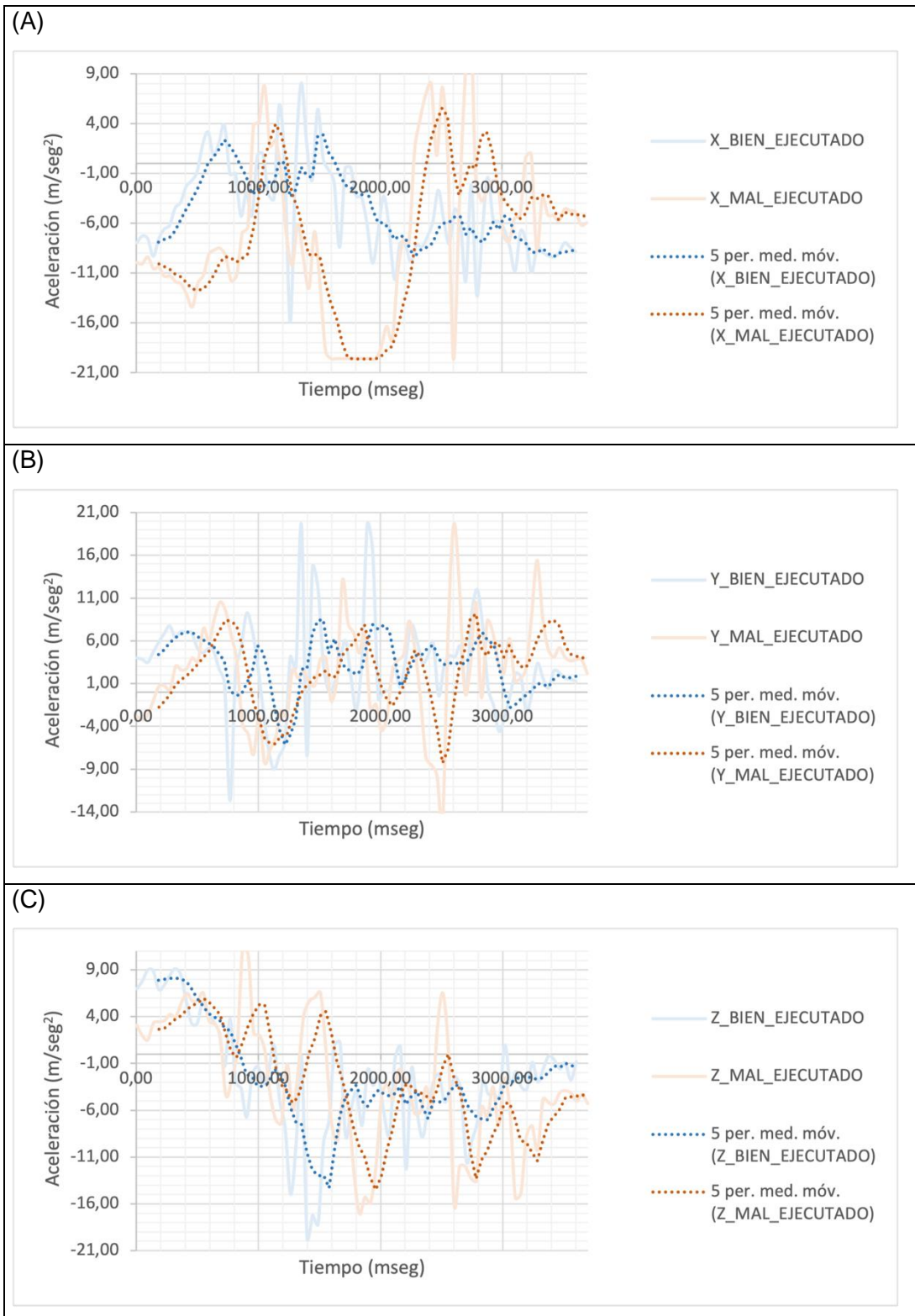


Figura 52: Resultados obtenidos en el sensor MPU_1, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

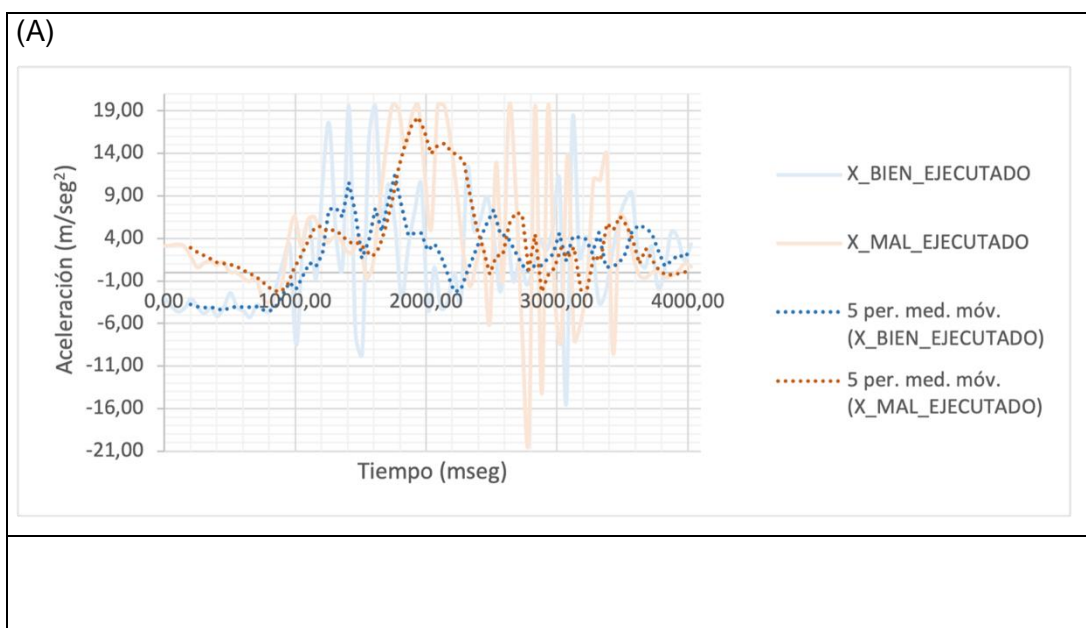
- En la Tabla 11, se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_1.

Tabla 11: Análisis de resultados obtenidos en el sensor MPU_1.

MPU_1
ACELERACIONES OBTENIDAS
<p>La mayor aceleración obtenida se encuentra en el eje X, con un valor aproximado de 21m/seg² en una prueba mal ejecutada, seguido del eje Z, con aceleraciones similares de 16m/seg², entre prueba errónea y correcta.</p> <p>Como parte del análisis, la Tori ha mencionado que en esta zona del cuerpo no se forja una fuerza que genere movimientos hacia adelante, atrás, izquierda o derecha; únicamente de arriba hacia abajo, tal y como se visualiza en el eje Z. Es probable, que el pico creado en el eje X, es producto del movimiento erróneo intencional de la judoca.</p>
TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA
<p>Inicialmente se puede apreciar que existe un desfase entre gráficas. Esto ocurre esencialmente debido a la velocidad, aceleración y fuerza de ejecución de la técnica, errores de coordinación entre intervalos de tiempo y de pulsación del botón, que ha simple visto son imperceptibles; sin embargo, el microprocesador trabaja en escala de milisegundos y almacena tiempos diferentes para las mismas pruebas, sea errónea o correcta. Las gráficas tienden a seguir un patrón y en rasgos globales tienen una similitud. En ciertas ocasiones, se aprecia que la judoca genera mayor aceleración en la prueba errónea. Es probable que en este aspecto la Tori emplea mayor fuerza, con el objetivo de culminar más rápido y ejecutar la técnica correctamente.</p>

Fuente: El autor.

- MPU_2: Este sensor está ubicado cerca de la muñeca izquierda según el Anexo 5. La Figura 53 muestra los resultados obtenidos.



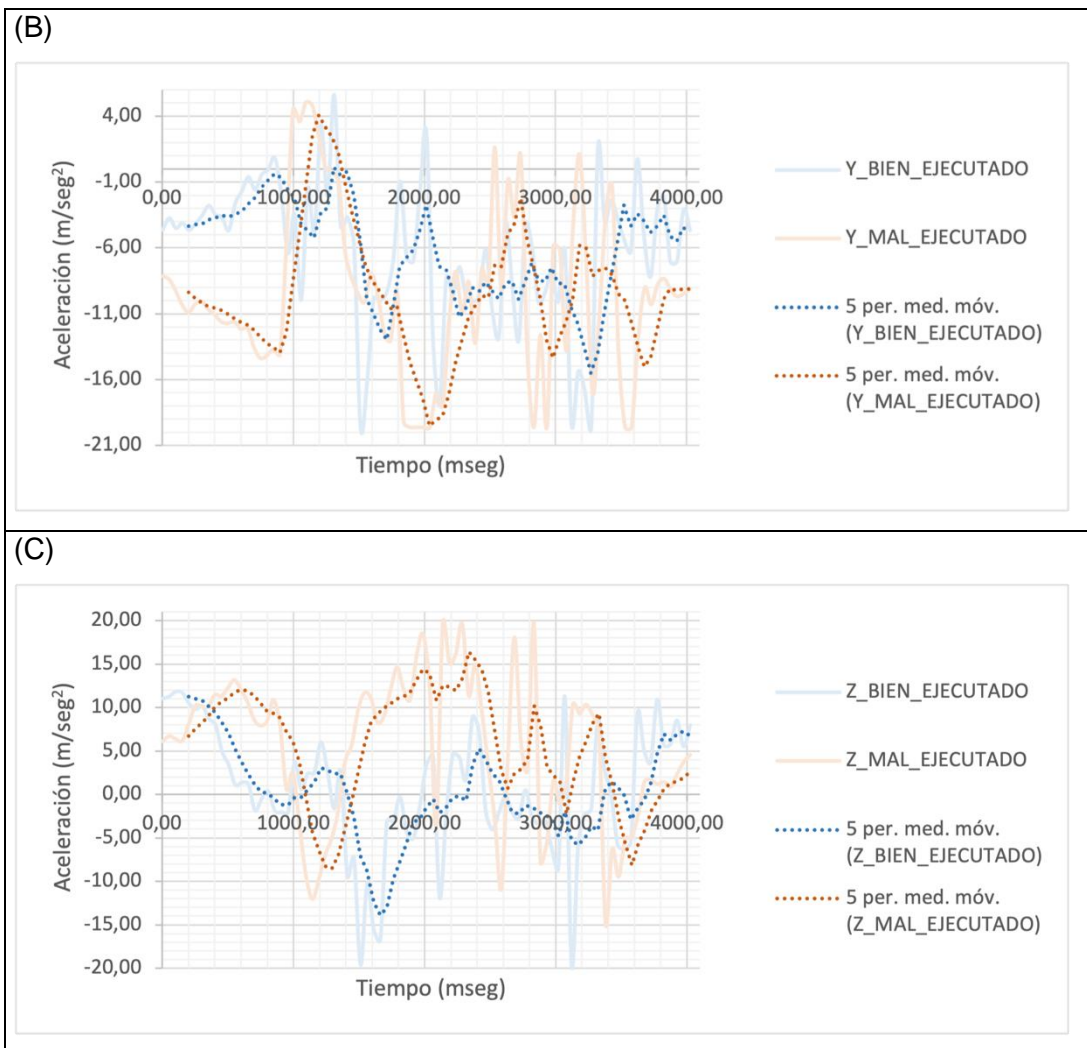


Figura 53: Resultados obtenidos en el sensor MPU_2, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

- En la Tabla 12, se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_2.

Tabla 12: Análisis de resultados obtenidos en el sensor MPU_2.

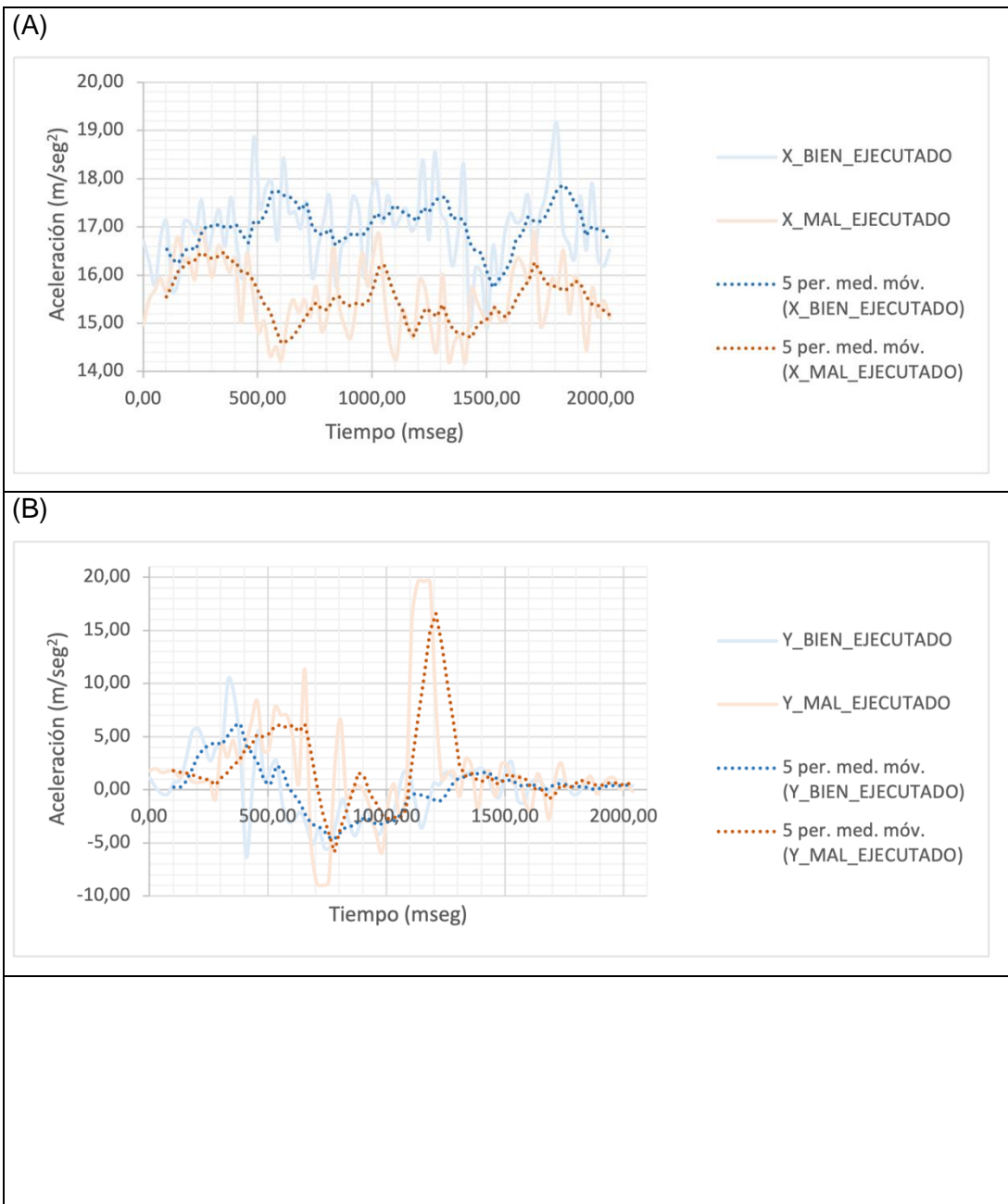
MPU_2
ACELERACIONES OBTENIDAS
<p>La mayor aceleración se encuentra en el eje Y con un valor aproximado de 20m/seg² en un Uchi mata erróneo, seguido de una aceleración de 15m/seg² ejecutado correctamente. Si bien la mayor aceleración ejecutada correctamente se encuentra en el eje Z, el eje X presenta mayor número de aceleraciones por encima de los 5 m/seg, llegando a tener entre 5 y 6 picos. La atleta ha comentado que, durante el trayecto de movimiento del brazo izquierda, la zona de la muñeca tiende a realizar un movimiento hacia delante, como consecuencia se aprecian esos picos en el eje X. Las aceleraciones generadas en los otros ejes, sobre todo en Z, es producto del movimiento accionado para desplazar de arriba hacia abajo al contrincante.</p>

TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA

Esta zona presenta un análisis complejo debido a la fluctuación que presentan los valores. Las gráficas que más similitud tienen corresponden al eje X. Se han realizado múltiples pruebas y se ha notado que tanto el sensor MPU_2, como el MPU_11, al estar posicionados en una zona terminal del traje como lo es la muñeca y el tobillo respectivamente, sufren mayor movimiento, y por ende ingresa un mayor ruido a los datos obtenidos. No se puede obtener una similitud ni una tendencia entre las dos pruebas en este sensor.

Fuente: El autor.

- MPU_9: Este sensor está ubicado cerca de la cadera, en la zona lateral izquierda según el Anexo 5. La Figura 54 muestra los resultados obtenidos.



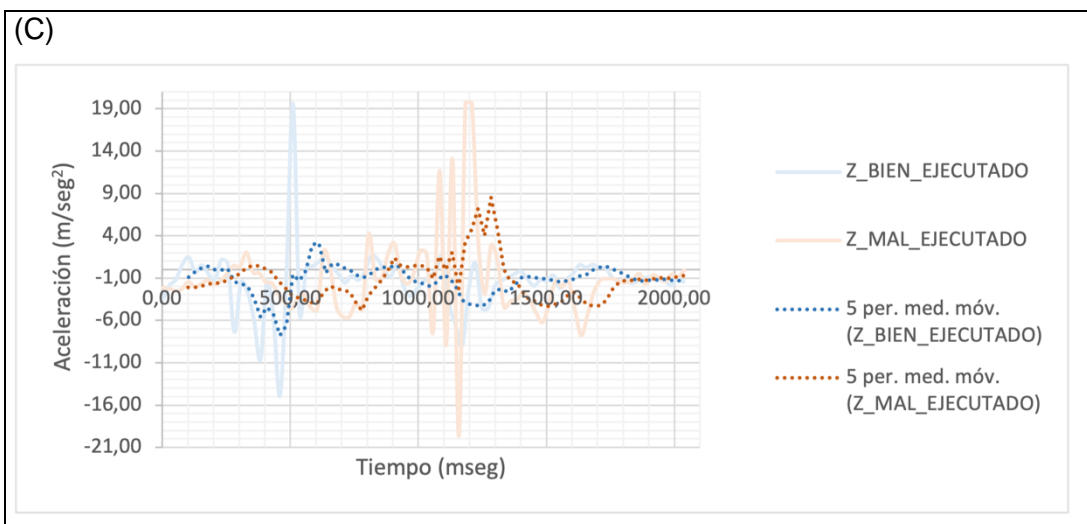


Figura 54: Resultados obtenidos en el sensor MPU_9, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

- En la Tabla 13, se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_9.

Tabla 13: Análisis de resultados obtenidos en el sensor MPU_9.

MPU_9
ACELERACIONES OBTENIDAS
La aceleración con mayor registro se encuentra en el eje X, con un valor aproximado de 18 m/seg ² al ejecutar satisfactoriamente la técnica y con un valor de 15 m/seg ² aproximadamente cuando se ha ejecutado erróneamente, mostrado en el eje Y. El eje X presenta un rango de aceleraciones que oscilan entre los 14 y los 18 m/seg ² en los dos casos. Esto se debe al movimiento que hace la Tori previo al levantamiento del contrincante.
TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA
El movimiento generado por la judoca es girar el cuerpo entero de izquierda a derecha (o viceversa, según la posición de inicio) con el objetivo de ofrecer la espalda al contrincante. Este movimiento, se ve reflejado en el eje X. Se mantiene constante, o al menos dentro de un rango establecido y sigue una misma tendencia. Lo mismo ocurre con el eje Z y con Y, tienen una similitud considerable, con la peculiaridad de no generar mucha aceleración, que da lugar a tener una mínima fuerza en estos ejes.

Fuente: El autor.

- MPU_10: Este sensor está ubicado cerca del hombro izquierdo según el Anexo 5. La Figura 55 muestra los resultados obtenidos.

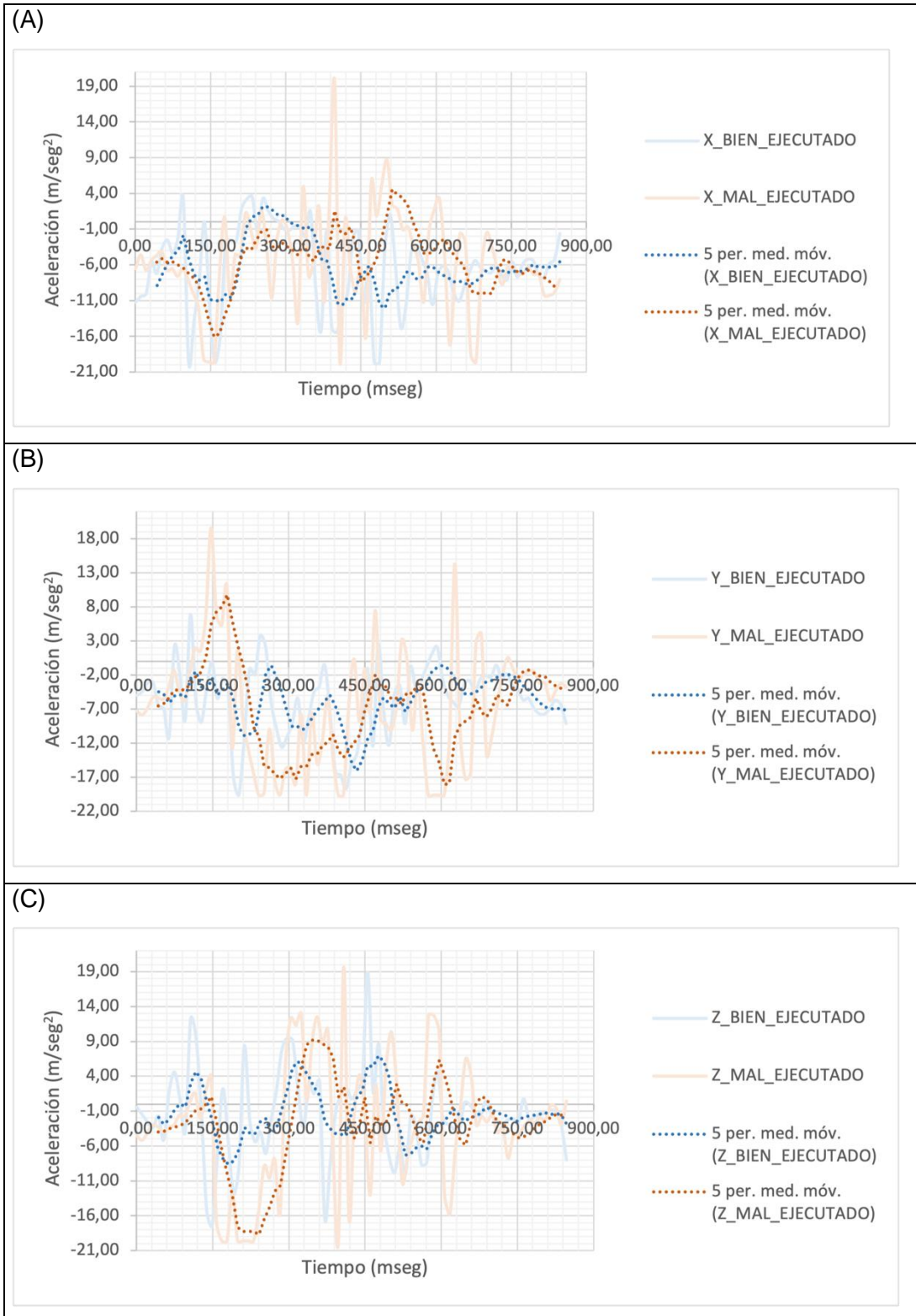


Figura 55: Resultados obtenidos en el sensor MPU_10, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

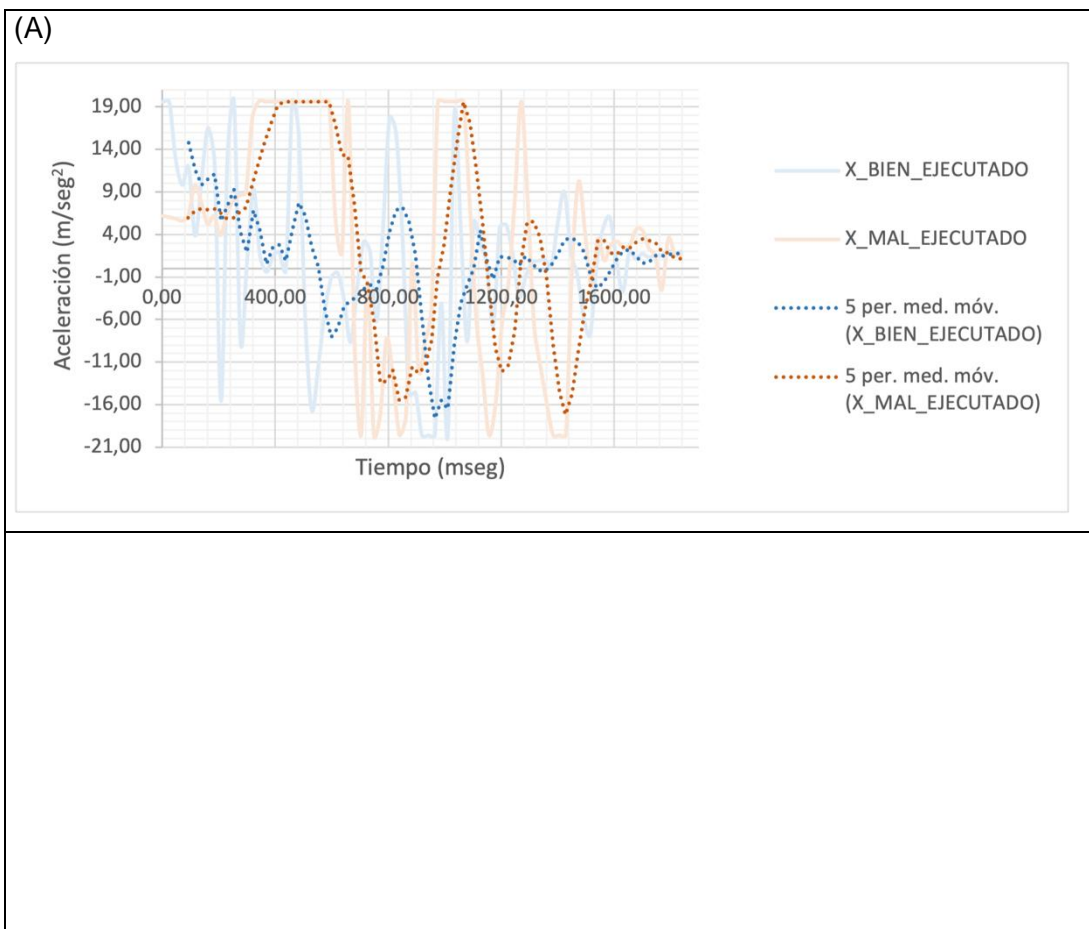
- En la Tabla 14, se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_10.

Tabla 14: Análisis de resultados obtenidos en el sensor MPU_10.

MPU_10
ACELERACIONES OBTENIDAS
<p>El mayor registro de aceleración se encuentra en el eje Z con un valor aproximado de 20m/seg² con una técnica empleada erróneamente, mientras que en el eje Y se encuentra una aceleración de 17 m/seg² cuando se ha ejecutado correctamente.</p> <p>La pierna de la judoca analizada sigue la misma dirección que la zona de la cadera que no tiene contacto directo con el contrincante. Esta es la razón por la que nuevamente, se presenta una aceleración entre rangos de 6 y m/seg en el eje X, durante casi toda la técnica. El eje Z y el eje X presentan cierta similitud, pero con valores con desfase. Además, la judoca realiza movimientos involuntarios que generaron mayor aceleración y desplazamiento.</p>
TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA
<p>Las gráficas presentan cierta tendencia y similitud. Nuevamente, se presenta cierto desfase y mínimas fluctuaciones en el eje Z, producto de la posición final de la judoca. Al finalizar con esta pierna como único punto de equilibrio, la judoca en muy pocas veces tiende a tambalear y se produce un mínimo ruido en los valores de este sensor.</p>

Fuente: El autor.

- MPU_11: Este sensor está ubicado cerca del hombro izquierdo según el Anexo 5. La Figura 56 muestra los resultados obtenidos.



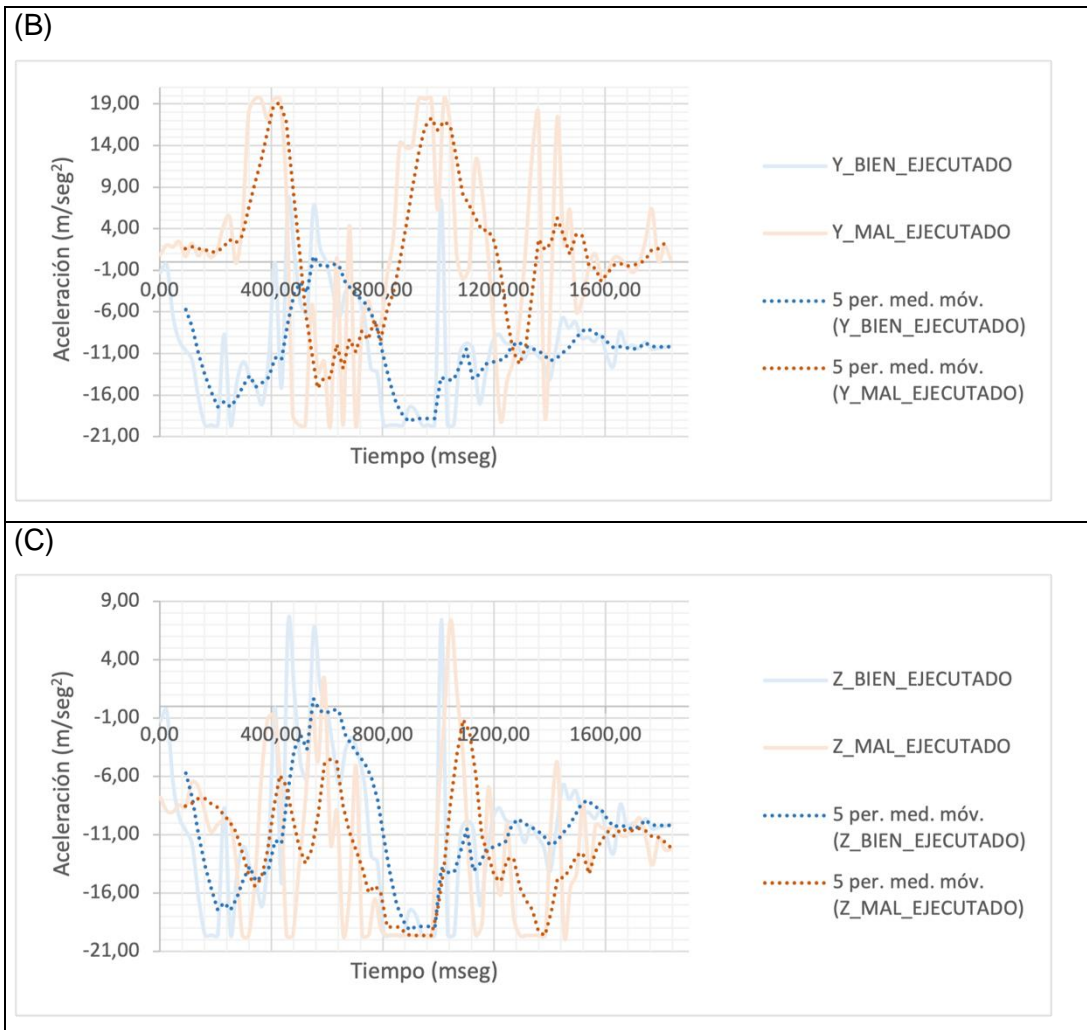


Figura 56: Resultados obtenidos en el sensor MPU_11, A) Eje X; B) Eje Y, C) Eje Z.

Fuente: El autor.

- En la Tabla 15, se hace un análisis comparativo entre las pruebas ejecutadas y valores de aceleración del sensor MPU_11.

Tabla 15: Análisis de resultados obtenidos en el sensor MPU_11.

MPU_11
ACELERACIONES OBTENIDAS
En este sensor se han presentado varios picos de aceleraciones que sobrepasan los 17 m/seg ² , tanto en la técnica ejecutada erróneamente como en la prueba correcta.
TÉCNICA EJECUTADA CORRECTA VS. INCORRECTA
Ocurre el mismo evento del sensor MPU_2. Al estar implementado en una zona que tiende a absorber todo el ruido, genera datos que no tienen una tendencia o similitud entre pruebas. Otro factor por considerar es el desequilibrio de la judoca al equilibrar con una única pierna.

Fuente: El autor.

5.2 Prototipo desarrollado vs. métodos tradicionales.

Según (Barnet, 2014) el Uchi mata ha sido dividido en tres fases para la ejecución de la técnica:

- **Desequilibrio:** Se producen mediante un empuje o tire, por parte del Tori al Uke.
- **Posición para atacar:** Una vez que se ha logrado el desequilibrio, el Tori se desplaza hasta el cuerpo del Uke según la posición deseada previo a la técnica por ejecutar.
- **Ataque:** Comprende el periodo desde que el Uke se desprende del piso, hasta que hace contacto con el suelo.

Siguiendo este contexto, una de las formas empleadas como método tradicional para comprobar la correcta ejecución de técnicas del judo es mediante indicadores biomecánicos, aplicando una metodología empírica, mediante la percepción visual. El sensei en todas las fases antes mencionadas, visualiza la distancia de los pies, distancia del agarre, ángulos entre muslos y piernas, altura del centro de gravedad, desplazamientos, altura de codos, cadera, direcciones, entre otros (Barnet, 2014); se aplican modelos matemáticos y se obtiene la retroalimentación deseada. Esta misma metodología es aplicada en (Santiago et al., 2014). Se analiza el agarre, el desequilibrio, la posición de las extremidades y las diferentes fases de la técnica Koshi Guruma, mediante la emisión del feedback del entrenador, juntamente con un modelo matemático para obtener la frecuencia de repetición de errores en la técnica. Es por lo que esta metodología de retroalimentación (mayormente utilizada) por parte del entrenador a los judocas, utilizando únicamente la vista y analizando in situ el desarrollo de la técnica y la frecuencia de errores en la ejecución, ha sido utilizada como medio de comparación con el prototipo desarrollado en este documento.

5.2.1 Ventajas del prototipo de traje vs. los métodos tradicionales.

- **Datos obtenidos:** El prototipo de traje desarrollado en este documento consta de 11 sensores, los cuales están instalados en puntos específicos que actualmente miden aceleraciones; la integración de este valor genera la velocidad y la segunda integración de este nuevo dato es el desplazamiento del sensor. Los datos ingresan a un sistema IoT que funciona a velocidades muy rápidas, permitiendo que ningún dato se pierda. Como se visualizó anteriormente, cada acelerómetro puede generar información que permite conocer si el judoca está realizando (o

no) correctamente las técnicas, con respecto a las aceleraciones medidas. En los métodos tradicionales esto no ocurre. No se tiene una información sobre el comportamiento de las aceleraciones, desplazamientos ni velocidades en diferentes zonas de análisis. Únicamente, se debe de confiar en la experiencia del entrenador, que analiza los entrenamientos, explica las fallas y corrige los errores.

- **Tiempo real:** El microprocesador utilizado permite la comunicación con otros dispositivos mediante protocolos y conexiones vistas anteriormente. Esto quiere decir que, los datos adquiridos pueden ser visualizados en tiempo real. Con este prototipo de traje diseñado, el entrenador puede acceder a esta información al momento que se esté realizando las distintas técnicas, de manera que pueda generar esa retroalimentación al instante, mejorar posturas, evitar lesiones, entre otros. Esto no es posible mediante los métodos tradicionales, ya que comúnmente requieren de cámaras o dispositivos que registran la actividad de los judocas y son consultados posteriormente por el entrenador. Esto genera un inconveniente, debido a que se pierde la conexión y comunicación entre judoca y entrenador, la cual debe de ser fluida, exacta e inmediata.
- **Innovación:** El IoT es una rama de la ciencia que está en constante actualización. Mucha investigación se realiza utilizando sensores y actuadores IoT y están al alcance del mercado de la región. Hoy en día, los dispositivos IoT cuentan ya con técnicas de Inteligencia Artificial que, mediante visión por computador pueden realizar el mismo trabajo que un entrenador, a una escala milimétrica de detalle y exactitud. Los métodos tradicionales no necesitan de innovación, es más es probable que no puedan innovar, ya que el único medio de verificación es la experiencia y la percepción del entrenador. Si bien, es importante la opinión del entrenador, es necesario utilizar la tecnología para generar un espacio de cooperación entre humano y máquina y poder satisfacer a las necesidades y niveles que presentan los actuales judocas.
- **Costos:** La inversión de la electrónica implementada en el prototipo del traje de este documento está valorado en un precio estimado de \$70 - \$80 USD. Con esta cantidad de dinero, se puede desarrollar un traje capaz de medir aceleración, velocidad, desplazamiento y fuerza en tiempo real, generar una base de datos de los valores adquiridos, ser

consultados por el entrenador para tomar acciones que mejoren las técnicas del judo. En los métodos tradicionales, normalmente se utilizan equipos de video para revisar las grabaciones de entrenamiento y realizar la retroalimentación a los judocas. Una cámara de video de media – alta gama esta sobre los \$500 USD, y cumple una única función, que es grabar los entrenamientos, llegando a costar 7 u 8 veces más de lo invertido en el traje del tipo DIY.

5.2.2 Desventajas

- **Cableado:** Los cables siempre serán un problema en todos sistema eléctrico y electrónico. En este caso no es la excepción, se tiene una central que integra: alimentación, microcontrolador, sistema de almacenamiento. Aquí es donde inicia y termina todo el cableado, para generar intercambio de datos con los sensores. En las pruebas se tuvieron ciertos inconvenientes ya que por una u otra técnica aplicada el cableado incomodaba a la judoca, se desprendían de la prenda o en el peor de los casos se cortaba la comunicación, inhabilitando el sistema.
- **Post-procesado de datos:** Los valores actualmente son guardados en una memoria que tienen un formato listo para ser importados a una tabla de análisis de datos (puede ser una tabla de Microsoft Excel). En las condiciones actuales, se necesita de un post-procesado de datos, para generar las distintas graficas que permitan conocer los valores de aceleración de cada punto de análisis. Es necesario que el entrenador tenga conceptos mínimos sobre el uso y manejo de datos para filtrar, graficar y compartir esta información.
- **Innovación:** La mayor parte de entrenadores de judo son adultos mayores, con años de experiencia y que han aplicado la técnica de percepción una y otra vez obteniendo buenos resultados. Siempre va a existir el rechazo al cambio y sobre todo a la adaptación de nuevas tecnologías. Es probable que la integración de este prototipo en los entrenamientos de judo no sea del agrado del sensei, generando incomodidad, repercusión y el deseo de seguir utilizando técnicas tradicionales.

CAPITULO IV

6. CONCLUSIONES.

- El traje desarrollado permitió encontrar las aceleraciones en diferentes puntos del cuerpo humano al efectuar la técnica Uchi mata. Con este tipo de información, los entrenadores pueden observar que zona es la que está ejerciendo mayor (o menor) aceleración en el judoca y por consiguiente en donde se aplica más (o menos) fuerza. También se puede observar cuando el atleta está realizando correctamente (o erróneamente) una técnica, con respecto a la posición y al movimiento realizado en los sensores ubicados en las extremidades superiores e inferiores. Este traje puede aportar de mucha información e integrarse sin ningún inconveniente a las prácticas diarias del judo, siendo un aporte al entrenamiento.
- Los acelerómetros son dispositivos capaces de compartir importante información durante varias décadas en múltiples disciplinas. En el deporte han incursionado generando datos con relación al movimiento, desplazamiento y aceleración de cierta zona de interés en los atletas. En el judo, se ha demostrado que con este tipo de sensores se puede conocer qué región del cuerpo es, la que mayor trabajo realiza el judoca. De hecho, con este dispositivo se ha evidenciado que, entre una técnica Uchi mata bien y mal ejecutada existe varias diferencias de la aceleración aplicada entre una zona y otra. También se ha demostrado que la zona menos propensa a fluctuaciones ha sido la cadera (sensor MPU_9), mientras que en las muñecas y tobillos se han obtenido datos con considerables diferencias.
- La metodología actual para la implementación y el desarrollo de software y hardware de los componentes en el traje biomecánico ha permitido encontrar diferentes errores que podrían ser solventados. Uno de ellos es la variación de la señal adquirida en las muñecas y tobillos del judoca. Una de las posibles soluciones sería cambiar la prenda que sirve para la instalación de los dispositivos electrónicos. Actualmente, el material utilizado en la fabricación del traje es 100% poliéster, según el fabricante. La prenda más adecuada para disminuir errores de medición por movimientos involuntarios del atleta es un traje del tipo licra. Esto debido

a que este material, se adhiere completamente al cuerpo humano y no permite que el propio traje adquiriera movimientos involuntarios, y por ende genere datos con ruido.

- Otra solución para limitar errores en el censado es la comunicación inalámbrica de los acelerómetros. Cada sensor utiliza cuatro cables, tanto para la alimentación y la comunicación con el microcontrolador. A lo largo de la implementación de los dispositivos electrónicos en el traje, se generaron varios problemas en comunicación, alimentación y datos erróneos. Utilizar microcontroladores de bajo consumo como es el caso del NodeMCU 32S por cada acelerómetro, disminuiría los errores en conexiones y mejoraría el tráfico de datos entre sensores y microcontrolador. Sin embargo, aumentaría un valor adicional al costo total de implementación, pero se mejoraría características de desempeño y eficacia.
- Este tipo de traje ha sido probado con una única técnica de judo, como es el caso del Uchi mata. Es posible utilizar este mismo traje para otros tipos de técnicas y poco a poco ir generando una base de datos de los entrenamientos. Con esta futura información, el traje se podría ir adaptando a los sistemas de IA y los algoritmos de visión por computador que se han desarrollado y se han conocido al realizar la revisión del estado actual del arte. Esto conlleva a tener un sistema más inteligente y que podría aportar con información más detallada a los entrenamientos y a los judocas.

7. RECOMENDACIONES.

- Hay varias formas de reducir el ruido generado por los acelerómetros al momento de censar los movimientos del atleta. El próximo paso para mejorar el sistema de medición de aceleraciones es la actualización a un método inalámbrico de comunicación. Hoy en día en el mercado existen muchas soluciones de comunicación sin cable, mediante tecnología Bluetooth, Wi-Fi, protocolo ZigBee, entre otros. Otra forma es la mencionada anteriormente, utilizar un microcontrolador por cada acelerómetro y enviar la información a una central que controla el flujo de datos y emite los resultados.
- Actualmente cada sensor tiene su eje de coordenadas debido a la posición que ha sido implementado en el traje. Esto conlleva a tener

múltiples ejes y generar información más compleja de procesar. Se recomienda generar un punto de referencia de manera que todos estén referenciados al mismo punto de origen en el eje cartesiano.

- Sería importante verificar el comportamiento del traje para distintas técnicas de judo, así como ampliar las distintas disciplinas de análisis, tanto para deportes de contacto, como otros deportes que generen interés en la región y puedan ser parte del entrenamiento diario, con el objetivo de mejorar posturas, aplicación de técnicas, entre otros, de manera que se pueda mejorar el rendimiento deportivo.
- Sería recomendable buscar tecnología de sistemas microcontrolados integrables en ropa que sean más potentes y que permitan realizar las mediciones de una mejor manera, pues el sistema de control debería integrarse en el traje deportivo para no causar molestias o interferencias en las técnicas. Lamentablemente, esta tecnología aún no está disponible en el país.

8. BIBLIOGRAFÍA.

Aprendiendo Arduino. (2013, November 13). *Bus SPI*. Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/2016/11/13/bus-spi/>

Aprendiendo Arduino. (2016, November 9). *Alimentación Arduino*. Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/2016/11/09/alimentacion-arduino/>

Ates, H. C., Yetisen, A. K., Güder, F., & Dincer, C. (2021). Wearable devices for the detection of COVID-19. *Nature Electronics*, 4(1), 13–14. <https://doi.org/10.1038/s41928-020-00533-1>

Banzy, M., & Shiloh, M. (2022). *Getting started with Arduino*. Maker Media, Inc.

Barnet, D. (2014). *Análisis del comportamiento de los indicadores biomecánicos de la técnica Uchi-mata del deporte Judo* [Bachelor's Thesis]. Universidad de Ciencias de la Cultura Física y el Deporte "Manuel Fajardo."

Boxall, J. (2021). *Arduino workshop: A Hands-On introduction with 65 projects*. No starch press.

Breijo, E. G. (2012). *Compilador C CCS y simulador PROTEUS para microcontroladores PIC*. Marcombo.

Ciaccioni, S., Condello, G., Guidotti, F., & Capranica, L. (2019). Effects of Judo Training on Bones: A Systematic Literature Review. *Journal of Strength and Conditioning Research*, 33(10), 2882–2896. <https://doi.org/10.1519/JSC.0000000000002340>

Craats, R. (2019). *Judo*. Weigl Publishers.

Cumming Martial Arts Center. (2018). *Uchi Mata*. CMAC-JUDO.Com. <http://cmac-judo.com/gokyo-no-waza/nage-waza-throwing-yechniku/dai-nikyo---green-belt/uchi-mata.html>

Detanico, D., & dos Santos, S. G. (2012). Avaliação específica no judô: Uma revisão de métodosE. In *Revista Brasileira de Cineantropometria e Desempenho Humano* (Vol. 14, Issue 6, pp. 738–748). <https://doi.org/10.5007/1980-0037.2012v14n6p738>

Echeverria, J., & C. Santos, O. (2021). KUMITRON: Artificial intelligence system to monitor karate fights that synchronize aerial images with physiological

and inertial signals. *26th International Conference on Intelligent User Interfaces-Companion*, 37–39.

García, R. H. (2018). *Judo nuevas perspectivas sobre Metodología y Entrenamiento*. Wanceulen Editorial.

Gras-Consuegra, M. de J., Reyes-Palau, N. C., & Samada-Cruz, Y. (2021). Control técnico-táctico de combinaciones en el nague-waza del judo. *Arrancada*, 21(39), 125–135. <https://revistarrancada.cujae.edu.cu/index.php/arrancada/article/view/395>

Gutierrez-Garcia, C., Astrain, I., Izquierdo, I., Gomez-Alonso, M. T., & Yague, J. M. (2018). Effects of judo participation in children: a systematic review. *Ido Movement for Culture. Journal of Martial Arts Anthropology*, 18(4), 63–73.

Haghi, M., Thurow, K., & Stoll, R. (2017). Wearable devices in medical internet of things: Scientific research and commercially available devices. *Healthcare Informatics Research*, 23(1), 4–15. <https://doi.org/10.4258/hir.2017.23.1.4>

Harris Semiconductor SCHS209C. (2003). CD74HC4067, CD74HCT4067. In *Sparkfun.com* (pp. 2–2). Texas Instruments.

Hernández, L. (2017). *NodeMCU tutorial paso a paso desde cero*. Programafacil.Com. <https://programafacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

Iqbal, S. M. A., Mahgoub, I., Du, E., Leavitt, M. A., & Asghar, W. (2021). Advances in healthcare wearable devices. In *npj Flexible Electronics* (Vol. 5, Issue 1). Nature Research. <https://doi.org/10.1038/s41528-021-00107-x>

Kim, W., & Kim, M. (2016). Soccer kick detection using a wearable sensor. *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 1207–1209. <https://doi.org/10.1109/ICTC.2016.7763408>

Kobayashi, K., & Sharp, H. E. (2014). *Sport of Judo*. Tuttle Publishing.

Kos, M., & Kramberger, I. (2017). A Wearable Device and System for Movement and Biometric Data Acquisition for Sports Applications. *IEEE Access*, 5, 6411–6420. <https://doi.org/10.1109/ACCESS.2017.2675538>

Li, R. T., Kling, S. R., Salata, M. J., Cupp, S. A., Sheehan, J., & Voos, J. E. (2016). Wearable Performance Devices in Sports Medicine. In *Sports Health* (Vol.

8, Issue 1, pp. 74–78). SAGE Publications Inc.
<https://doi.org/10.1177/1941738115616917>

Li, S., Zhang, B., Fei, P., Shakeel, P. M., & Samuel, R. D. J. (2020). Computational efficient wearable sensor network health monitoring system for sports athletics using IoT. In *Aggression and Violent Behavior*. Elsevier Ltd.
<https://doi.org/10.1016/j.avb.2020.101541>

Llamas, L. (2016a, October 16). *Leer y escribir en una tarjeta SD o microSD con Arduino*. LUIS LLAMAS - Ingeniería, Informática y Diseño.
<https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Llamas, L. (2016b, November 19). *Más salidas/entradas en Arduino con multiplexor CD74HC4067*. LUIS LLAMAS - Ingeniería, Informática y Diseño.
<https://www.luisllamas.es/mas-salidas-y-entradas-en-arduino-con-multiplexor-cd74hc4067/>

Ma, Y., Ouyang, J., Raza, T., Li, P., Jian, A., Li, Z., Liu, H., Chen, M., Zhang, X., Qu, L., Tian, M., & Tao, G. (2021). Flexible all-textile dual tactile-tension sensors for monitoring athletic motion during taekwondo. *Nano Energy*, 85.
<https://doi.org/10.1016/j.nanoen.2021.105941>

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 03(05), 164–173. <https://doi.org/10.4236/jcc.2015.35021>

Martínez, H., & Sebastián, A. (2019). *Sistema domótico modular con automatización y control de dispositivos eléctricos, basado en Nodemcu-32S* [Bachelor's Thesis]. Universidad del Bío-Bío.

Miarka, B., Julio, U. F., del Vecchio, F. B., Calmet, M., & Franchini, E. (2012). Technique and tactic in Judo: A review. *Revista de Artes Marciales Asiáticas*, 5(1), 91–112.

Minns, P. D. (2013). *C Programming for the PC the MAC and the Arduino microcontroller system*. Author House.

Murillo Saucedo, E. R. (2016). *El perfeccionamiento de la técnica del judo* [Master's Thesis]. Universidad Autónoma de Nuevo León.

NAYLAMP MECHATRONICS. (2016). *Tutorial MPU6050, acelerómetro y giroscopio*. NAYLAMP MECHATRONICS.

https://naylorpmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html

Nguyen, L. N. N., Rodríguez-Martín, D., Català, A., Pérez-López, C., Samà, A., & Cavallaro, A. (2015). Basketball activity recognition using wearable inertial measurement units. *ACM International Conference Proceeding Series*, 07-09-September-2015. <https://doi.org/10.1145/2829875.2829930>

Pan, T., & Zhu, Y. (2018). Getting Started with Arduino. In *Designing Embedded Systems with Arduino: A Fundamental Technology for Makers* (pp. 3–16). Springer Singapore. https://doi.org/10.1007/978-981-10-4418-2_1

Pečnikar Oblak, V., Karpljuk, D., Vodičar, J., & Simenko, J. (2020). Inclusion of people with intellectual disabilities in judo: A systematic review of literature. *Archives of Budo*, 16, 245–260.

Peña, C. (2020). *Arduino IDE: Domina la programación y controla la placa*. RedUsers.

Peset Mancebo, M. F., Ferrer Sapena, A., Villamón Herrera, M., González Moreno, L. M., Toca Herrera, J.-L., & Aleixandre Benavent, R. (2013). Scientific literature analysis of Judo in Web of Science. *Archives of Budo*, 9(2), 81–91.

Phillips, J. T. G., & Toscano, B. A. (2022). Pasajes de la historia del Judo Ecuatoria o (Parte 1). *Observatorio Iberoamericano Economía y Sociedad de Japón*.

Poongodi, T., Krishnamurthi, R., Indrakumari, R., Suresh, P., & Balusamy, B. (2020). Wearable Devices and IoT. In V. K. and K. R. and A. Md. A. R. Balas Valentina E. and Solanki (Ed.), *A Handbook of Internet of Things in Biomedical and Cyber Physical System* (pp. 245–273). Springer International Publishing. https://doi.org/10.1007/978-3-030-23983-1_10

Robots Didácticos. (2019, June 24). *Descripción y funcionamiento del Bus I2C*. Robots Didácticos. <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>

Robots Didácticos. (2020, June 20). *¿Qué es la comunicación serie?* Robots Didácticos. <http://robots-argentina.com.ar/didactica/que-es-la-comunicacion-serie/>

Salim, F. A., Haider, F., Tasdemir, S. B. Y., Naghashi, V., Tengiz, I., Cengiz, K., Postma, D. B. W., & van Delden, R. (2019). Volleyball action modelling for

behavior analysis and interactive multi-modal feedback. *Proceeding of the 15th International Summer Workshop on Multimodal Interfaces*.

Santiago, A. G., Lage, I. P., Carral, J. M. C., & Pérez, C. A. (2014). Análisis del error en la técnica de judo Koshi-guruma mediante T-Patterns. *Revista Internacional de Medicina y Ciencias de La Actividad Física y Del Deporte*, 14(55), 1–8.

Saponara, S. (2017). Wearable Biometric Performance Measurement System for Combat Sports. *IEEE Transactions on Instrumentation and Measurement*, 66(10), 2545–2555. <https://doi.org/10.1109/TIM.2017.2677679>

Sterkowicz, S., Sacripanti, A., & Sterkowicz_Przybycien, K. (2013). Techniques frequently used during London Olympic judo tournaments: a biomechanical approach. *ArXiv Preprint ArXiv:1308.0716*.

Tama, D., Gomes, P., Abreu, M. J., Souto, A. P., & Carvalho, H. (2017). Intelligent textiles for martial arts. *XIVth International Izmir Textile and Apparel Symposium*.

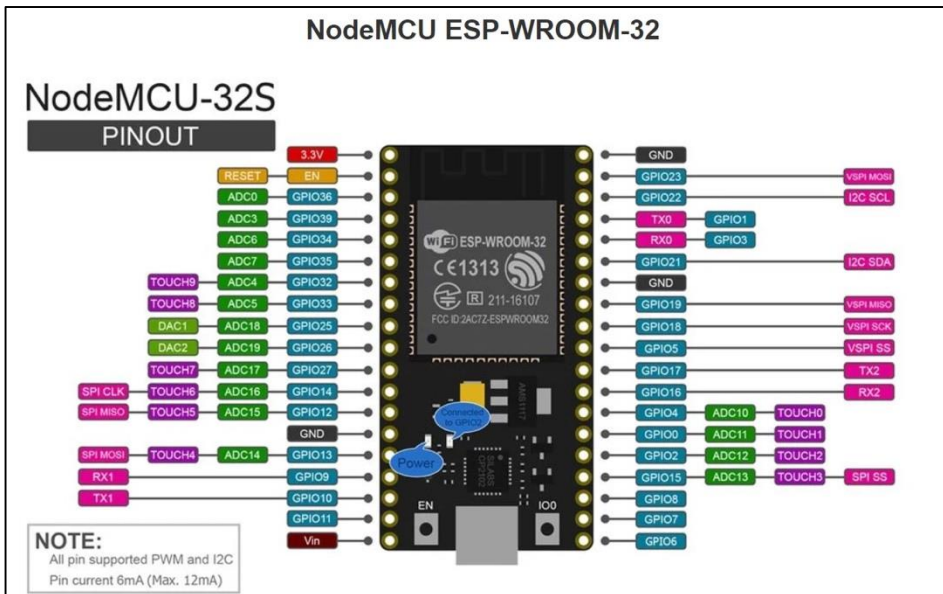
Torres-Luque, G., Hernández-García, R., Escobar-Molina, R., Garatachea, N., & Nikolaidis, P. T. (2016). Physical and physiological characteristics of judo athletes: An update. In *Sports* (Vol. 4, Issue 1). MDPI. <https://doi.org/10.3390/sports4010020>

Torres-Ortega, H. (2014). Guía de Diseño PCB con EAGLE. *H. Torres-Ortega, Guía de Diseño PCB Con EAGLE*.

Xukyo. (2020). *Crear una interfaz web para controlar su ESP32 NodeMCU*. AranaCorp. <https://www.aranacorp.com/es/crear-una-interfaz-web-para-controlar-su-esp32-nodemcu/>

Ye, X., Shi, B., Li, M., Fan, Q., Qi, X., Liu, X., Zhao, S., Jiang, L., Zhang, X., Fu, K., Qu, L., & Tian, M. (2022). All-textile sensors for boxing punch force and velocity detection. *Nano Energy*, 107114. <https://doi.org/10.1016/j.nanoen.2022.107114>

9. ANEXOS.



Anexo 1: Distribución de pines de la tarjeta NodeMCU-32S.

Fuente: (Xukyo, 2020).

TRUTH TABLE

S0	S1	S2	S3	\bar{E}	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

H= High Level
L= Low Level

Anexo 2: Tabla de verdad del multiplexor 74HC4067.

Fuente: (Harris Semiconductor SCHS209C, 2003).

```
x
1 / 3 | [Navigation icons] | 67,5% | [Zoom icons] | [Tools icons]
// Basic demo for accelerometer readings from Adafruit MPU6050

// ESP32 Guide: https://RandomNerdTutorials.
com/esp32-mpu-6050-accelerometer-gyroscope-arduino/
// ESP8266 Guide: https://RandomNerdTutorials.
com/esp8266-nodemcu-mpu-6050-accelerometer-gyroscope-arduino/
// Arduino Guide: https://RandomNerdTutorials.
com/arduino-mpu-6050-accelerometer-gyroscope/

#include<Adafruit_MPU6050.h>
#include<Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
  case MPU6050_RANGE_2_G:
    Serial.println("+2G");
    break;
  case MPU6050_RANGE_4_G:
    Serial.println("+4G");
    break;
  case MPU6050_RANGE_8_G:
    Serial.println("+8G");
    break;
  case MPU6050_RANGE_16_G:
    Serial.println("+16G");
    break;
  }
```











```

}
mpu.setGyroRange(MPU6050_RANGE_500_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
  Serial.println("+/- 250 deg/s");
  break;
case MPU6050_RANGE_500_DEG:
  Serial.println("+/- 500 deg/s");
  break;
case MPU6050_RANGE_1000_DEG:
  Serial.println("+/- 1000 deg/s");
  break;
case MPU6050_RANGE_2000_DEG:
  Serial.println("+/- 2000 deg/s");
  break;
}

mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
  Serial.println("260 Hz");
  break;
case MPU6050_BAND_184_HZ:
  Serial.println("184 Hz");
  break;
case MPU6050_BAND_94_HZ:
  Serial.println("94 Hz");
  break;
case MPU6050_BAND_44_HZ:
  Serial.println("44 Hz");
  break;
case MPU6050_BAND_21_HZ:
  Serial.println("21 Hz");
  break;
case MPU6050_BAND_10_HZ:
  Serial.println("10 Hz");
  break;
case MPU6050_BAND_5_HZ:
  Serial.println("5 Hz");
  break;
}

Serial.println("");
delay(100);

```

```
3 / 3 |     67,5%      
```

```
}  
  
void loop() {  
  /* Get new sensor events with the readings */  
  sensors_event_t a, g, temp;  
  mpu.getEvent(&a, &g, &temp);  
  
  /* Print out the values */  
  Serial.print("Acceleration X: ");  
  Serial.print(a.acceleration.x);  
  Serial.print(", Y: ");  
  Serial.print(a.acceleration.y);  
  Serial.print(", Z: ");  
  Serial.print(a.acceleration.z);  
  Serial.println(" m/s^2");  
  
  Serial.print("Rotation X: ");  
  Serial.print(g.gyro.x);  
  Serial.print(", Y: ");  
  Serial.print(g.gyro.y);  
  Serial.print(", Z: ");  
  Serial.print(g.gyro.z);  
  Serial.println(" rad/s");  
  
  Serial.print("Temperature: ");  
  Serial.print(temp.temperature);  
  Serial.println(" degC");  
  
  Serial.println("");  
  delay(500);  
}
```

Anexo 3: Código completo del prototipo v1.

Fuente: El autor.

```
pm_v3.pdf x
1 / 9 67,5%
#include<Adafruit MPU6050.h>
#include<Adafruit_Sensor.h>
#include <Wire.h>
#include <SD.h>

/* ***** CREACION DE OBJETOS MPU6050 ***** */
Adafruit MPU6050 mpu_0;
Adafruit MPU6050 mpu_1;
Adafruit MPU6050 mpu_2;
Adafruit MPU6050 mpu_3;
Adafruit MPU6050 mpu_4;
Adafruit MPU6050 mpu_5;
Adafruit MPU6050 mpu_6;
Adafruit MPU6050 mpu_7;
Adafruit MPU6050 mpu_8;
Adafruit MPU6050 mpu_9;
Adafruit MPU6050 mpu_10;
Adafruit MPU6050 mpu_11;
/* ***** ASIGNACION DE SALIDAS DIGITALES ***** */
int s0 = 32; //0
int s1 = 33; //7
int s2 = 25; //8
int s3 = 26; //0
int puls = 27;
int buttonState = 0;
int ledPin = 2;

//Mux in "SIG" pin
int SIG_pin = 0;
/* ***** MPU INIT ***** */
void mpu_init(int channel)
{
  delay(50);
  if (channel == 0)
  {
    sendMux(0);
    if (!mpu_0.begin())
    {
      Serial.println("Failed to find mpu_0");
      while (1)
      {
        delay(10);
      }
    }
    Serial.println("mpu_0 Found!");
  }
}
```

```
pm_v3.pdf x
2 / 9 67,5%
else if (channel == 1)
{
  sendMux(1);
  if (!mpu_1.begin())
  {
    Serial.println("Failed to find mpu_1");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_1 Found!");
}
else if (channel == 2)
{
  sendMux(2);
  if (!mpu_2.begin())
  {
    Serial.println("Failed to find mpu_2");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_2 Found!");
}
else if (channel == 3)
{
  sendMux(3);
  if (!mpu_3.begin())
  {
    Serial.println("Failed to find mpu_3");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_3 Found!");
}
else if (channel == 4)
{
  sendMux(4);
  if (!mpu_4.begin())
  {
    Serial.println("Failed to find mpu_4");
    while (1)
```

```
pm_v3.pdf x
3 / 9 67,5%
{
  delay(10);
}
}
Serial.println("mpu_4 Found!");
}
else if (channel == 5)
{
  sendMux(5);
  if (!mpu_5.begin())
  {
    Serial.println("Failed to find mpu_5");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_5 Found!");
}
else if (channel == 6)
{
  sendMux(6);
  if (!mpu_6.begin())
  {
    Serial.println("Failed to find mpu_6");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_6 Found!");
}
else if (channel == 7)
{
  sendMux(7);
  if (!mpu_7.begin())
  {
    Serial.println("Failed to find mpu_7");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_7 Found!");
}
else if (channel == 8)
```

```
pm_v3.pdf x
4 / 9 67,5%
{
  sendMux(8);
  if (!mpu_8.begin())
  {
    Serial.println("Failed to find mpu_8");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_8 Found!");
}
else if (channel == 9)
{
  sendMux(9);
  if (!mpu_9.begin())
  {
    Serial.println("Failed to find mpu_9");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_9 Found!");
}
else if (channel == 10)
{
  sendMux(10);
  if (!mpu_10.begin())
  {
    Serial.println("Failed to find mpu_10");
    while (1)
    {
      delay(10);
    }
  }
  Serial.println("mpu_10 Found!");
}
else if (channel == 11)
{
  sendMux(11);
  if (!mpu_11.begin())
  {
    Serial.println("Failed to find mpu_11");
    while (1)
    {
```

```
pm_v3.pdf x
5 / 9 67,5%
delay(10);
}
}
Serial.println("mpu_11 Found!");
}
}
/* ***** MPU GET ***** */
void mpu_get(int channel)
{
  sensors_event_t a, g, temp;
  if (channel == 0)
  {
    sendMux(0);
    mpu_0.getEvent(&a, &g, &temp);
  }
  else if (channel == 1)
  {
    sendMux(1);
    mpu_1.getEvent(&a, &g, &temp);
  }
  else if (channel == 2)
  {
    sendMux(2);
    mpu_2.getEvent(&a, &g, &temp);
  }
  else if (channel == 3)
  {
    sendMux(3);
    mpu_3.getEvent(&a, &g, &temp);
  }
  else if (channel == 4)
  {
    sendMux(4);
    mpu_4.getEvent(&a, &g, &temp);
  }
  else if (channel == 5)
  {
    sendMux(5);
    mpu_5.getEvent(&a, &g, &temp);
  }
  else if (channel == 6)
  {
    sendMux(6);
    mpu_6.getEvent(&a, &g, &temp);
  }
  else if (channel == 7)
```

```
pm_v3.pdf x
6 / 9 67,5%
{
  sendMux(7);
  mpu_7.getEvent(&a, &g, &temp);
}
else if (channel == 8)
{
  sendMux(8);
  mpu_8.getEvent(&a, &g, &temp);
}
else if (channel == 9)
{
  sendMux(9);
  mpu_9.getEvent(&a, &g, &temp);
}
else if (channel == 10)
{
  sendMux(10);
  mpu_10.getEvent(&a, &g, &temp);
}
else if (channel == 11)
{
  sendMux(11);
  mpu_11.getEvent(&a, &g, &temp);
}
Serial.print("MPU_");
Serial.print(channel);
Serial.print(" Acc X: ");
Serial.print(a.acceleration.x);
Serial.print(", Y: ");
Serial.print(a.acceleration.y);
Serial.print(", Z: ");
Serial.print(a.acceleration.z);
Serial.println(" m/s^2");
String value_a_acc_x = String(a.acceleration.x,2);
String value_a_acc_y = String(a.acceleration.y,2);
String value_a_acc_z = String(a.acceleration.z,2);
save_data(channel, value_a_acc_x, value_a_acc_y, value_a_acc_z);
}

//void save_data(float ax, float ay, float az, float gx, float gy , float gz)
void save_data(int channel, String ax, String ay, String az)
{
  File myFile = SD.open("/sensores.txt", FILE_APPEND);//abrimos el archivo
  if (myFile)
  {
    Serial.println("Escribiendo SD: ");
  }
}
```

```
pm_v3.pdf x
7 / 9 67,5%
myFile.print(channel);
myFile.print(",X,");
myFile.print(ax);
myFile.print(",Y,");
myFile.print(ay);
myFile.print(",Z,");
myFile.println(az);
}
else
{
  Serial.println("No se puede escribir");
}
myFile.close();
}

void sendMux(int channel)
{
  int controlPin[] = {s0, s1, s2, s3};

  int muxChannel[16][4]=
  {
    {0,0,0,0}, //channel 0
    {1,0,0,0}, //channel 1
    {0,1,0,0}, //channel 2
    {1,1,0,0}, //channel 3
    {0,0,1,0}, //channel 4
    {1,0,1,0}, //channel 5
    {0,1,1,0}, //channel 6
    {1,1,1,0}, //channel 7
    {0,0,0,1}, //channel 8
    {1,0,0,1}, //channel 9
    {0,1,0,1}, //channel 10
    {1,1,0,1}, //channel 11
    {0,0,1,1}, //channel 12
    {1,0,1,1}, //channel 13
    {0,1,1,1}, //channel 14
    {1,1,1,1} //channel 15
  };

  //loop through the 4 sig
  for(int i = 0; i < 4; i ++)
  {
    digitalWrite(controlPin[i], muxChannel[channel][i]);
  }
}
```

```
pm_v3.pdf x
8 / 9 67,5%
void setup()
{
  Serial.begin(115200);
  /* ***** PINMODE ***** */
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(puls, INPUT);
  pinMode(ledPin, OUTPUT);
  /* ***** MPU INIT ***** */
  mpu_init(0);
  mpu_init(1);
  mpu_init(2);
  mpu_init(3);
  mpu_init(4);
  mpu_init(5);
  mpu_init(6);
  mpu_init(7);
  mpu_init(8);
  mpu_init(9);
  mpu_init(10);
  mpu_init(11);
  /* ***** SD INIT ***** */
  Serial.print("Iniciando SD ...");
  if (!SD.begin())
  {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");
}

void loop()
{
  buttonState = digitalRead(puls);
  if (buttonState == LOW)
  {
    mpu_get(0);
    mpu_get(1);
    mpu_get(2);
    mpu_get(3);
    mpu_get(4);
    mpu_get(5);
    mpu_get(6);
    mpu_get(7);
  }
}
```

```
pm_v3.pdf x
9 / 9 67,5%
mpu_get(8);
mpu_get(9);
mpu_get(10);
mpu_get(11);
digitalWrite(ledPin, HIGH);
}
else
{
digitalWrite(ledPin, LOW);
}
}
```

Anexo 4: Código completo del prototipo v2.

Fuente: El autor.



Anexo 5: Disposición de los acelerómetros en el traje.

Fuente: El autor.

AUTORIZACION DE PUBLICACION EN EL REPOSITORIO INSTITUCIONAL

Yo, Pablo Fabián Martínez Jara portador de la cédula de ciudadanía N.º 0102367513. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación "Sistema embebido electrónico en kimono para monitorizar técnicas de judo" de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizo a la Universidad para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, 04 de octubre de 2023



Pablo Fabián Martínez Jara

0102367513