



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE TELECOMUNICACIONES

**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR,
PROYECTO DE INVESTIGACIÓN**

TEMA:

**“SISTEMA DE ENTRENAMIENTO INTELIGENTE BASADO EN UN DISPOSITIVO DE DETECCIÓN
DE PUNTAJE PARA PETOS Y CASCOS ELECTRÓNICOS EN LA ACTIVIDAD DEL TAEKWONDO”**

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cybersecurity (seguridad cibernética)

AUTOR:

López Flores Jhonatan Eduardo

DIRECTOR:

Ing. Vásquez Ayala Carlos Alberto, MSc

Ibarra, 2025

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1725212383		
APELLIDOS Y NOMBRES:	LÓPEZ FLORES JHONATAN EDUARDO		
DIRECCIÓN:	Quito, Oe2 y S41 Conjunto Villas de Toledo		
EMAIL:	jelopezf@utn.edu.ec		
TELÉFONO FIJO:	02 2974623	TELF. MOVIL	0999040039

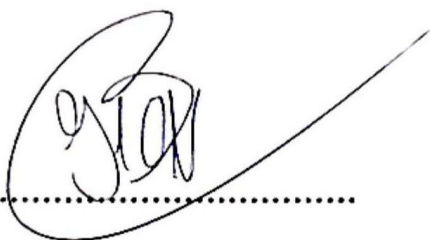
DATOS DE LA OBRA	
TÍTULO:	“SISTEMA DE ENTRENAMIENTO INTELIGENTE BASADO EN UN DISPOSITIVO DE DETECCIÓN DE PUNTAJE PARA PETOS Y CASCOS ELECTRÓNICOS EN LA ACTIVIDAD DEL TAEKWONDO”
AUTOR (ES):	LÓPEZ FLORES JHONATAN EDUARDO
FECHA:	16 de julio de 2025
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	☒ GRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES
DIRECTOR:	ING. VÁSQUEZ AYALA CARLOS ALBERTO, MSC
ASESOR:	ING. JAIME ROBERTO MICHILENA CALDERÓN, MSC

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, LÓPEZ FLORES JHONATAN EDUARDO, con cédula de identidad N° 1725212383, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 23 días del mes de julio de 2025.

EL AUTOR:

A handwritten signature in black ink, appearing to read 'Jhonatan', is written over a horizontal dotted line. A long, sweeping horizontal stroke extends from the end of the signature to the right.

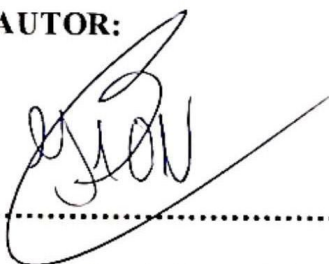
Jhonatan Eduardo López Flores

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 23 días del mes de julio de 2025.

EL AUTOR:

A handwritten signature in black ink, appearing to read 'Jhonatan', is written over a horizontal dotted line. The signature is fluid and cursive.

Jhonatan Eduardo López Flores

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 16 de julio de 2025

ING. CARLOS ALBERTO VÁSQUEZ AYALA, MSC
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

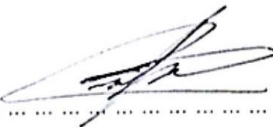

.....

Ing. Carlos Alberto Vásquez Ayala. MSc

C.C.: 1002424982

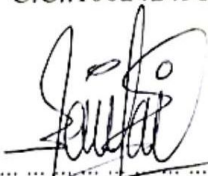
APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “SISTEMA DE ENTRENAMIENTO INTELIGENTE BASADO EN UN DISPOSITIVO DE DETECCIÓN DE PUNTAJE PARA PETOS Y CASCOS ELECTRÓNICOS EN LA ACTIVIDAD DEL TAEKWONDO”, elaborado por Jhonatan Eduardo López Flores, previo a la obtención del título de Ingeniero en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



ING. CARLOS ALBERTO VÁSQUEZ AYALA. MSC

C.C.:1002424982



ING. JAIME ROBERTO MICHILENA CALDERÓN. MSC

C.C.: 1002198438

DEDICATORIA

Este trabajo de titulación va dedicado a Lizbette, quien fue y es una parte fundamental en mi vida, la mujer que me ha apoyado en los momentos más difíciles y ha sido mi motor e inspiración para que pueda cumplir todos mis logros y me motiva a seguir esforzándome para ser una mejor persona y un mejor profesional.

A mi abuelita Yolanda, que, aunque ya no se encuentra presente, siempre está presente en mis pensamientos, ya que fue una gran persona que dejó un legado muy especial en mí, con sus enseñanzas y cariño.

Jhonatan Eduardo López Flores

AGRADECIMIENTO

En primer lugar quiero agradecer a mis padres por el apoyo y enseñanzas recibidas durante todas las etapas de mi vida.

A mi novia y a sus padres, por el apoyo constante durante esta etapa, ya que me han abierto los brazos y me han hecho parte de su familia.

A todos mis amigos, en especial a Diego e Israel, que con las risas y los buenos y malos momentos logramos superar todas las adversidades y problemas que se presentaron durante la carrera.

A todos los profesores y maestros que he tenido en mi vida, en especial a MSc. Carlos Vásquez, MSc. Jaime Michilena, MSc. Fabián Cuzme, MSc. Mauricio Domínguez y MSc. Edgar Maya, de quienes recibí todo el apoyo y las enseñanzas para poder ser un buen profesional y poder desarrollarme en el campo laboral.

Al Club de Artes Marciales Ecuadecuba, en especial al Lic. Eddy Cajigal, por la apertura para poder desarrollar este proyecto en sus instalaciones y por las enseñanzas recibidas en las artes marciales, que me han forjado a nivel físico y de espíritu.

A mis amigos de Huawei Technologies, quienes con su amistad y compañerismo han demostrado ser un gran grupo y unas excelentes personas y profesionales.

Jhonatan Eduardo López Flores

RESUMEN EJECUTIVO

El presente trabajo de titulación propone el diseño e implementación de un sistema inteligente de entrenamiento para monitorear sesiones de Taekwondo. Este dispositivo está orientado a optimizar la evaluación técnica de las patadas ejecutadas mediante el uso de sensores físicos, retroalimentación visual y generación de reportes. El sistema está conformado por tres sensores de fuerza FSR 402, dispuestos de forma triangular dentro de un pateador de precisión. Para el circuito de recopilación de datos se utiliza un Arduino Lilypad, mientras que para la transmisión de datos se emplea un módulo ESP32 que envía la información por WiFi hacia un servidor backend local desarrollado en Python, utilizando la librería FastAPI. El frontend está construido con las librerías React y Tailwind CSS, permitiendo gestionar usuarios, iniciar sesiones por tiempo o repeticiones, visualizar el estado del dispositivo en tiempo real y generar informes individuales por cada sesión y por cada alumno. La arquitectura electrónica y el uso de componentes compactos permiten que el dispositivo sea completamente portátil e inalámbrico, lo que facilita su adaptación al entorno del Club de Artes Marciales Ecuadecuba. La validación de las patadas se realiza mediante decisión automática del sistema, gracias a un algoritmo de aprendizaje KNN con umbrales de fuerza definidos. Las pruebas funcionales demostraron la eficacia del sistema para registrar y clasificar los impactos, mientras que las encuestas aplicadas a los actores clave reflejaron una alta aceptación del sistema en cuanto a funcionalidad, comodidad y utilidad en un contexto real de entrenamiento. Este proyecto representa un aporte innovador en el entrenamiento inteligente de las artes marciales, promoviendo el uso de nuevas tecnologías en las sesiones de práctica.

Palabras clave: sensores FSR, ESP32, Arduino Lilypad, Taekwondo, KNN, aprendizaje de máquina, entrenamiento deportivo, FastAPI, sistema de entrenamiento inteligente, visualización en tiempo real, artes marciales.

ABSTRACT

This thesis project proposes the design and implementation of an intelligent training system to monitor Taekwondo sessions. The device is aimed at optimizing the technical evaluation of executed kicks through the use of physical sensors, visual feedback, and automated report generation. The system consists of three FSR 402 force sensors arranged in a triangular configuration inside a precision kicking target. An Arduino Lilypad is used for the data collection circuit, while an ESP32 module handles data transmission via WiFi to a local backend server developed in Python using the FastAPI framework. The frontend is built with React and Tailwind CSS libraries, allowing for user management, the initiation of sessions based on time or repetitions, real-time device status monitoring, and the generation of individual reports per session and per student. The electronic architecture and use of compact components make the device fully portable and wireless, enabling its adaptation to the environment of the Ecuadcuba Martial Arts Club. Kick validation is performed automatically by the system through a KNN machine learning algorithm with predefined force thresholds. Functional tests demonstrated the system's effectiveness in recording and classifying impacts, while surveys conducted with key stakeholders showed high acceptance in terms of functionality, comfort, and usefulness in real training scenarios. This project represents an innovative contribution to smart martial arts training, promoting the integration of new technologies into practice sessions.

Keywords: FSR sensors, ESP32, Arduino Lilypad, Taekwondo, KNN, machine learning, sports training, FastAPI, intelligent training system, real-time visualization, martial arts.

LISTA DE SIGLAS

HTTP: "HyperText Transfer Protocol" (*Protocolo de Transferencia de Hipertexto*).

KNN: "K-Nearest Neighbors" (*K Vecinos Más Cercanos*).

CRUD: "Create, Read, Update, Delete" (*Crear, Leer, Actualizar, Eliminar*).

SQL: "Structured Query Language" (*Lenguaje de Consulta Estructurado*).

API: "Application Programming Interface" (*Interfaz de Programación de Aplicaciones*).

FSR: "Force Sensitive Resistor" (*Resistor Sensible a la Fuerza*).

IP: "Internet Protocol" (*Protocolo de Internet*).

WiFi: "Wireless Fidelity" Protocolo de comunicación inalámbrica (*Fidelidad Inalámbrica*).

LED: "Light Emitting Diode" (*Diodo Emisor de Luz*).

LAN: "Local Area Network" (*Red de Área Local*).

UUID: "Universally Unique Identifier" (*Identificador Único Universal*).

DB: "Database" (*Base de Datos*).

Índice de contenidos

I.	Antecedentes.....	25
1.1.	Tema.....	25
1.2.	Problema.....	25
1.3.	Objetivos	27
1.3.1.	Objetivo general.....	27
1.3.2.	Objetivos específicos	27
1.4.	Alcance.....	27
1.4.1.	Diagrama de bloques	29
1.4.2.	Arquitectura	30
1.5.	Justificación	31
II.	Marco teórico	32
2.1.	Historia, Desarrollo y Reglamentación del Taekwondo.....	32
2.1.1.	Origen y Evolución Histórica	32
2.1.2.	Taekwondo en el Ámbito Internacional	34
2.1.3.	Normativas Internacionales en el Taekwondo	35
2.1.4.	Impacto de los Reglamentos en Técnicas y Estrategias	37
2.2.	Biomecánica y Sistemas de Puntuación en Taekwondo	38
2.2.1.	Principios de Biomecánica en Movimientos de Taekwondo	38
2.2.2.	Análisis Biomecánico de Técnicas de Patada	39
2.2.2.1.	Bandal chagui.....	39

2.2.2.2.	Ap chagui.....	40
2.2.3.	Evolución de los Métodos de Puntuación	42
2.3.	Tecnología y Machine Learning en el Entrenamiento Deportivo	43
2.3.1.	Herramientas y Dispositivos Tecnológicos en Entrenamiento	43
2.3.2.	Machine Learning.....	44
2.3.2.1.	Tipos de Machine Learning	44
2.4.	Modelo en V para Desarrollo de Sistemas de Entrenamiento	46
2.4.1.	Introducción al Modelo en V	46
2.4.2.	Etapas del modelo en V	46
2.5.	Placas Electrónicas y Servicios en la Nube.....	48
2.5.1.	Arduino LilyPad	48
2.5.2.	Servicios en la Nube.....	50
2.5.2.1.	Almacenamiento en la nube	50
III.	Desarrollo experimental.....	52
3.1.	Situación actual.....	52
3.2.	Metodología del modelo en V y Stakeholders	54
3.2.1.	Estudio de los Stakeholders	55
3.2.2.	Construcción de atributos de los requerimientos	56
3.2.2.1.	Nomenclatura de los requerimientos	56
3.2.3.	Requerimientos de Stakeholders	57
3.2.3.1.	Encuesta de requerimientos	57
3.2.3.2.	Tamaño de la muestra	57

3.2.3.3.	Determinación de parámetros	58
3.2.4.	Requerimientos del Sistema	60
3.2.5.	Requerimientos de arquitectura	62
3.2.6.	Benchmarking (Selección de Hardware y Software).....	64
3.2.6.1.	Hardware	65
3.2.6.1.1.	Microcontrolador	65
3.2.6.1.2.	Sensor de fuerza	66
3.2.6.1.3.	Módulo de comunicación	66
3.2.6.2.	Software	68
3.2.6.2.1.	Entorno de programación	68
3.2.6.2.2.	Plataforma en la nube	69
3.2.6.2.3.	Gestor de base de datos en la nube	69
3.3.	Diseño del sistema	70
3.3.1.	Diagrama de bloques general del sistema	71
3.3.1.1.	Registro	74
3.3.1.2.	Adquisición de datos	81
3.3.1.2.1.	Circuito de alimentación de batería en esquemático	82
3.3.1.2.2.	Módulos del circuito de alimentación de batería.	83
3.3.1.2.3.	Circuito de recopilación de datos.....	84
3.3.1.2.4.	Circuito de transmisión de datos	85
3.3.1.3.	Procesamiento de datos	86
3.3.1.3.1.	Activación de sensores	86
3.3.1.3.2.	Transmisión de datos	88

3.3.1.3.3.	Algoritmo KNN	96
3.3.1.4.	Tratamiento de información	99
3.3.1.4.1.	Almacenamiento de patadas en la base de datos	100
3.3.1.4.2.	Procesamiento en el backend	103
3.3.1.5.	Alertas	105
3.3.1.5.1.	Alertas visuales	106
3.3.1.5.2.	Alertas auditivas	111
3.3.1.6.	Reporte.....	113
3.3.1.6.1.	Reportes por sesión	114
3.3.1.6.2.	Reportes por usuario.....	117
IV.	Resultados	120
4.1.	Casos de pruebas.....	120
4.1.1.	Test eléctrico.....	120
4.1.2.	Test de hardware	125
4.1.3.	Test de software	133
4.1.3.1.	Pruebas de registro de usuario	135
4.1.3.2.	Pruebas de procesamiento de la información	139
4.1.3.3.	Pruebas de tratamiento de la información	142
4.1.3.4.	Visualización de alertas	146
4.1.3.5.	Visualización de generación de reportes	150
4.1.4.	Test de aplicación.....	153
4.1.5.	Test funcional	162
4.2.	Encuestas dirigidas al instructor y estudiantes del club	173

4.2.1.	Evaluación del nivel de aceptación del sistema por parte del director/entrenador del club	173
4.2.2.	Evaluación del nivel de aceptación del sistema por parte de los estudiantes del club	182
V.	Conclusiones	188
VI.	Recomendaciones	190
VII.	Referencias	192
VIII.	Anexos	197

Índice de figuras

Figura 1 Diagrama de bloques del sistema.	29
Figura 2 Arquitectura del sistema.	30
Figura 3 Patada bandal chagui (Academia de Taekwondo OCRAM, 2018).....	40
Figura 4 Patada bandal chagui (Academia de Taekwondo OCRAM, 2018).....	41
Figura 5 Etapas del modelo en V (Cantone, 2006)	47
Figura 6 Arduino LilyPad (Arduino, 2024)	48
Figura 7 Logo Ecuadcuba Fuente: Club de Artes Marciales Ecuadcuba	52
Figura 8 Diagrama de bloques del sistema del sistema (Autor)	72
Figura 9 Arquitectura del sistema (Autor)	74
Figura 10 Diagrama de bloques del registro e inicio de sesión de usuarios (Autor)	74
Figura 11 Script del user_model (Autor)	76
Figura 12 Script del scheme user (Autor)	77
Figura 13 Script de user_crud (Autor)	77
Figura 14 Script de main (Autor).....	78
Figura 15 Base de datos relacional creada en RDS (Autor)	80
Figura 16 Conexión del back end con la base de datos (Autor)	81
Figura 17 Diagrama de bloques adquisición de datos (Autor)	81
Figura 18 Circuito de alimentación esquemático (Autor)	82
Figura 19 Circuito de alimentación con módulos de carga y regulación de voltaje (Autor)	83

Figura 20 Circuito esquemático de recopilación de datos (Autor)	84
Figura 21 Circuito esquemático de transmisión de datos (Autor)	85
Figura 22 Diagrama de bloques de procesamiento de datos (Autor).....	86
Figura 23 Diagrama de flujo activación de sensores (Autor)	87
Figura 24 Script de activación de sensores (Autor)	87
Figura 25 Diagrama de flujo transmisión de datos (Autor)	89
Figura 26 Script de transmisión de datos (Autor)	89
Figura 27 Lista de comandos válidos ESP32 (Autor).....	90
Figura 28 Función de creación de comandos (Autor).....	91
Figura 29 Función para obtener comandos pendientes ESP32 (Autor)	91
Figura 30 Función para marcar comandos ejecutados ESP32 (Autor)	92
Figura 31 Funciones para obtener y marcar comandos en el backend (Autor)	93
Figura 32 Función para enviar datos Lilypad (Autor)	94
Figura 33 Función para recibir datos desde Lilypad hacia ESP32 (Autor)	94
Figura 34 Función para enviar datos al backend desde ESP32 (Autor)	95
Figura 35 Función del backend para guardar las patadas en la base de datos (Autor)	96
Figura 36 Diagrama de flujo algoritmo KNN (Autor).....	96
Figura 37 algoritmo KNN implementado en el backend (Autor)	98
Figura 38 Diagrama de bloques del tratamiento de la información (Autor).....	99

Figura 39 Diagrama de flujo para el almacenamiento en la base de datos (Autor)	100
Figura 40 Schema para la creación de “kick result” (Autor)	102
Figura 41 Estructura lógica del procesamiento del backend (Autor)	103
Figura 42 Archivo main.py (Autor)	104
Figura 43 Modelos y esquemas del proyecto (Autor).....	105
Figura 44 Bloque de tipos de alertas (Autor).....	106
Figura 45 Diagrama de flujo alertas visuales (Autor).....	106
Figura 46 Esquemático LED de encendido (Autor)	107
Figura 47 LED de estado conexión Wi-Fi (Autor)	108
Figura 48 Función del ESP32 para gestión del LED (Autor)	109
Figura 49 Esquemático circuito de obtención de datos con LED para la sesión (Autor)	109
Figura 50 Comando para indicar el estado de la sesión mediante un LED (Autor)	110
Figura 51 Diagrama de flujo para alertas auditivas (Autor)	111
Figura 52 Esquemático del circuito de obtención de datos con buzzer (Autor)	112
Figura 53 Función para emitir alertas auditivas (Autor).....	112
Figura 54 Bloques de los reportes generados (Autor)	113
Figura 55 Diagrama de flujo para reportes de sesiones (Autor)	114
Figura 56 Función para asociar la patada a la sesión activa (Autor)	116
Figura 57 Función para generar resultados globales de la sesión (Autor)	117

Figura 58 Diagrama de flujo para reportes de usuario (Autor).....	117
Figura 59 Función para crear una sesión de entrenamiento (Autor).....	119
Figura 60 Circuito de alimentación incorporado dentro del pateador (Autor)	122
Figura 61 LED de alimentación del dispositivo encendido (Autor)	123
Figura 62 Medida de voltaje	124
Figura 63 LED de conexión WiFi encendido en el dispositivo (Autor)	127
Figura 64 Ping hacia el dispositivo desde el localhost (Autor)	128
Figura 65 Descubrimiento de la red de área local (Autor).....	129
Figura 66 Verificación del dispositivo conectado desde el frontend	129
Figura 67 Distribución física de los sensores en el pateador (Autor)	130
Figura 68 Datos recibidos de los sensores y el circuito de recopilación de datos hacia el backend (Autor).....	131
Figura 69 Datos recibidos de los sensores y el circuito de recopilación mostrados en el frontend (Autor).....	132
Figura 70 Página de presentación del sistema (Autor)	134
Figura 71 Página de inicio de sesión o registro de nuevo usuario	135
Figura 72 Página de registro de usuarios	136
Figura 73 Visualización en la base de datos del usuario creado (Autor).....	137
Figura 74 Visualización del usuario en la lista desplegable (Autor)	138
Figura 75 Verificación de la información del usuario en la página de inicio (Autor)	138

Figura 76 Prueba de pateo al implemento (Autor)	140
Figura 77 Datos recibidos de los sensores (Autor)	141
Figura 78 Procesamiento de las patadas por parte del backend (Autor).....	141
Figura 79 Búsqueda relacional del usuario en la tabla de sesiones (Autor)	143
Figura 80 Sesiones del usuario jelopezf consultadas desde el backend (Autor)	143
Figura 81 Consulta de las patadas pertenecientes a la sesión de entrenamiento (Autor)	144
Figura 82 Búsqueda de los resultados de las patadas generadas en la sesión (Autor).....	145
Figura 83 Búsqueda del resultado de la sesión por session_id (Autor)	146
Figura 84 LED de encendido del prototipo activo (Autor).....	146
Figura 85 LED de conexión con el backend encendido en el prototipo (Autor)	147
Figura 86 LED de sesión de entrenamiento iniciada activo (Autor)	148
Figura 87 Zumbador incorporado al prototipo (Autor)	149
Figura 88 Pestaña de sesiones de entrenamiento (Autor)	151
Figura 89 Reporte de una sesión de entrenamiento (Autor)	151
Figura 90 Datos globales del usuario (Autor).....	152
Figura 91 Visualización del archivo iniciador del programa (Autor).....	155
Figura 92 Sistema instalado dentro del club para pruebas (Autor).....	155
Figura 93 Base de datos de los estudiantes que realizarán la prueba (Autor)	156
Figura 94 Captura de pantalla de la pestaña nueva sesión (Autor).....	157

Figura 95 Configuración de la sesión de entrenamiento en sitio (Autor)	157
Figura 96 Alertas visuales del dispositivo activas (Autor)	158
Figura 97 Estudiante realizando el pateo al prototipo (Autor)	159
Figura 98 Datos de patadas entrando al sistema (Autor)	160
Figura 99 Análisis de los resultados generados por el sistema por parte del entrenador (Autor)	161
Figura 100 Reporte de la sesión generado por el sistema (Autor)	161
Figura 101 Registros de nuevos usuarios en el sistema (Autor)	164
Figura 102 Menú de selección del modo de entrenamiento (Autor)	165
Figura 103 Sesión de entrenamiento configurada con 10 repeticiones (Autor)	165
Figura 104 Pruebas de pateo en el implemento por repeticiones (Autor)	166
Figura 105 Valores entrantes en la sesión de entrenamiento (Autor)	166
Figura 106 Datos de la sesión de entrenamiento presentados de manera gráfica (Autor)	167
Figura 107 Resultados estadísticos y gráficos de la sesión de entrenamiento (Autor)	168
Figura 108 Nueva sesión de entrenamiento en modo tiempo 30 segundos (Autor)	169
Figura 109 Estudiante realizando el pateo al implemento (Autor)	170
Figura 110 Monitoreo durante la sesión de entrenamiento	170
Figura 111 Resultados de la sesión de tiempo	171
Figura 112 Análisis del entrenador a la sesión finalizada	172

Figura 113 Tabulación de la primera pregunta del director (Autor).....	174
Figura 114 Tabulación de la segunda pregunta de la encuesta al director (Autor)	174
Figura 115 Tabulación de la tercera pregunta de la encuesta al director (Autor).....	175
Figura 116 Tabulación de la cuarta pregunta de la encuesta al director (Autor)	176
Figura 117 Tabulación de la quinta pregunta de la encuesta al director (Autor).....	177
Figura 118 Tabulación de la sexta pregunta de la encuesta al director (Autor)	178
Figura 119 Tabulación de la séptima pregunta de la encuesta al director (Autor)	179
Figura 120 Tabulación de la octava pregunta de la encuesta al director (Autor)	179
Figura 121 Tabulación de la novena pregunta de la encuesta al director (Autor)	180
Figura 122 Tabulación de la décima pregunta de la encuesta al director (Autor)	181
Figura 123 Tabulación de la undécima pregunta de la encuesta al director (Autor)	181
Figura 124 Nombres de los alumnos encuestados (Autor)	182
Figura 125 Tabulación de la primera pregunta de la encuesta a los estudiantes (Autor)	183
Figura 126 Tabulación de la segunda pregunta de la encuesta a los estudiantes (Autor).....	184
Figura 127 Tabulación de la tercera pregunta de la encuesta a los estudiantes (Autor)	184
Figura 128 Tabulación de la cuarta pregunta de la encuesta a los estudiantes (Autor)	185
Figura 129 Tabulación de la quinta pregunta de la encuesta a los estudiantes (Autor).....	186
Figura 130 Tabulación de la quinta pregunta de la encuesta a los estudiantes (Autor).....	186

Índice de tablas

Tabla 1 Lista de stakeholders	55
Tabla 2 Nomenclaturas requerimientos	56
Tabla 3 Requerimientos de stakeholders	59
Tabla 4 Requerimientos del sistema	60
Tabla 5 Requerimientos de arquitectura	62
Tabla 6 Selección de microcontrolador	65
Tabla 7 Selección sensor de fuerza	66
Tabla 8 Selección módulo de comunicación.....	67
Tabla 9 Selección entorno de programación	68
Tabla 10 Selección plataforma en la nube	69
Tabla 11 Selección gestor de base de datos	70
Tabla 12 Test Eléctrico	120
Tabla 13 Test de Hardware	125
Tabla 14 Test de software	133
Tabla 15 Valores de los sensores usados	139
Tabla 16 Test de aplicación.....	153
Tabla 17 Test funcional.....	162

I. Antecedentes

Este capítulo detalla los fundamentos y la base para el desarrollo del proyecto de titulación, los cuales incluyen: el tema elegido, el problema identificado, los objetivos, el alcance y la justificación. Estos elementos constituyen la estructura esencial para la creación de un sistema electrónico adecuado, con el propósito de diseñar un sistema de entrenamiento inteligente para Taekwondo basado en un dispositivo de detección de puntaje para petos y cascos electrónicos.

1.1. Tema

SISTEMA DE ENTRENAMIENTO INTELIGENTE BASADO EN UN DISPOSITIVO DE DETECCIÓN DE PUNTAJE PARA PETOS Y CASCOS ELECTRÓNICOS EN LA ACTIVIDAD DEL TAEKWONDO

1.2. Problema

En el deporte competitivo del Taekwondo, la introducción de sistemas de puntuación electrónicos ha revolucionado la forma en que se llevan a cabo las competencias. Estos sistemas, diseñados para mejorar la objetividad en la puntuación, requieren una precisión y técnica impecables por parte de los deportistas para activar correctamente los sensores y asegurar la anotación de puntos (Lee & Kim, 2022). A pesar de su intención de aumentar la justicia en el deporte, la transición a la puntuación electrónica ha presentado desafíos significativos que impactan el rendimiento de los competidores.

Los practicantes de artes marciales se enfrentan a la necesidad de adaptar su técnica para satisfacer las demandas de los nuevos sistemas de puntuación electrónicos. Muchos luchan con la adaptación, lo que puede llevar a una técnica deficiente y, en consecuencia, a un rendimiento

subóptimo en las competiciones (Park & Lee, 2021). Esta brecha en la adaptación tecnológica es particularmente pronunciada entre aquellos competidores que tienen acceso limitado a recursos y entrenamiento especializado, lo que puede resultar en desmotivación y una desventaja competitiva significativa.

La situación actual subraya la necesidad imperativa de desarrollar sistemas de entrenamiento inteligentes que puedan proporcionar retroalimentación en tiempo real. Estos sistemas podrían ayudar a los deportistas a ajustar su técnica de manera más efectiva y prepararse adecuadamente para las competencias con sistemas de puntuación electrónicos (Yilmaz & Sevinc, 2021). La implementación de tecnologías de entrenamiento que se alineen con las condiciones de competición modernas es esencial para mejorar las posibilidades de éxito de los deportistas en el escenario mundial.

La transición del peto convencional al peto electrónico en Taekwondo no solo representa un cambio para los deportistas, sino también para los entrenadores, quienes requieren recursos tecnológicos adecuados para facilitar este cambio. Los entrenadores son fundamentales en el proceso de adaptación de los deportistas a la puntuación electrónica, y su capacidad para proporcionar instrucción y retroalimentación efectivas depende de su propio entendimiento y manejo de las nuevas tecnologías (Kazimierczak & Peško, 2021). Por lo tanto, es esencial que se les proporcionen las herramientas y el conocimiento necesarios para aprovechar al máximo los sistemas de puntuación electrónicos y guiar a los deportistas a través de esta transición tecnológica (Lee & Choi, 2014).

1.3. Objetivos

1.3.1. Objetivo general

Implementar un sistema de entrenamiento inteligente para Taekwondo, mediante un dispositivo, para la mejora de la técnica de patadas y la adaptación a los sistemas de puntuación electrónicos en el Club “Ecuadcuba”.

1.3.2. Objetivos específicos

- Levantar información acerca de términos técnicos y tecnológicos clave dentro del ámbito del Taekwondo.
- Definir los requerimientos del sistema mediante la metodología del modelo en V.
- Diseñar el sistema de entrenamiento inteligente para Taekwondo.
- Realizar pruebas para validar los requerimientos y funcionalidad del sistema.

1.4. Alcance

El proyecto inicia con la fase de levantamiento de información, alineada con el primer bloque del modelo en V, que se centra en la comprensión de los *stakeholders* y la recopilación de información esencial sobre términos técnicos y tecnológicos en el ámbito del Taekwondo. Esta fase implica realizar encuestas y entrevistas con entrenadores y atletas del club Ecuadcuba, para obtener una comprensión profunda y relevante del deporte y las tecnologías asociadas, estableciendo así una base sólida de conocimientos que guiará el desarrollo del sistema.

Siguiendo la metodología del modelo en V, la próxima fase del proyecto se centra en la definición de los requerimientos. Inicialmente, se realizará un análisis detallado de los

requerimientos del sistema, identificando los elementos esenciales para la implementación efectiva del prototipo, así como para el análisis, almacenamiento y tratamiento de datos. Este análisis garantizará que todas las necesidades funcionales y operativas del sistema sean comprendidas y atendidas. Posteriormente, para los requerimientos de arquitectura, se aplicará el estándar IEEE 29148 en la definición de los requerimientos de software, proporcionando una estructura clara y completa. Paralelamente, se llevará a cabo un benchmarking de hardware, evaluando y seleccionando los componentes más adecuados y eficientes para el sistema, asegurando así una integración óptima entre el hardware y el software.

En la tercera fase del proyecto, alineada con el diseño y desarrollo en el modelo en V, se abordarán varios aspectos clave del sistema. Comenzando con la "Recopilación de datos", se diseñarán un arreglo de sensores y dispositivos para capturar métricas esenciales durante las sesiones de entrenamiento de Taekwondo, asegurando la precisión y relevancia de los datos recogidos. A continuación, el "Procesamiento de datos en tiempo real" se centrará en analizar estos datos de manera rápida y eficiente mediante el uso de un microcontrolador, proporcionando retroalimentación a los atletas y entrenadores. La "Comunicación del sistema con la nube" garantizará una transmisión segura y eficiente de los datos para su posterior análisis y almacenamiento, mientras que el "Análisis, almacenamiento y tratamiento de datos en la nube" aprovechará servicios externos y algoritmos avanzados como KNN para un tratamiento más efectivo de los datos. Finalmente, se desarrollará la "Interfaz de usuario y visualización", enfocándose en asegurar que los usuarios finales puedan acceder, visualizar y comprender los datos analizados de manera intuitiva y efectiva.

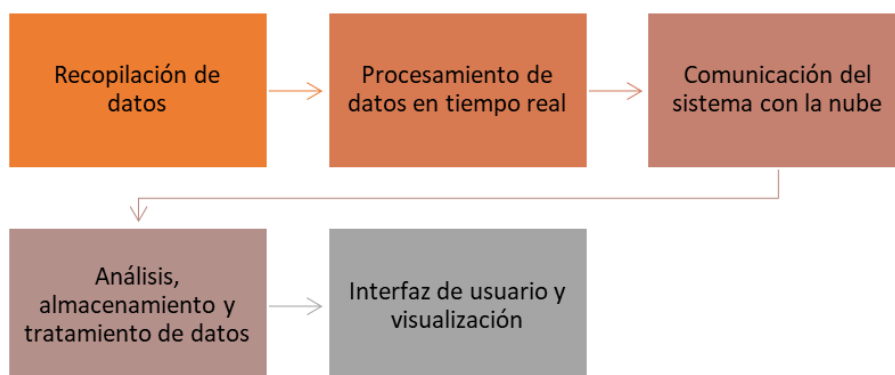
En la última fase del proyecto, alineada con la verificación y validación del modelo en V, se inicia con pruebas unitarias en la base de la "V". Estas pruebas se centran en verificar cada componente individual del sistema, asegurando su correcto funcionamiento de manera aislada. Luego, se procede con pruebas de integración, donde se evalúa la interacción y el funcionamiento conjunto de los componentes, para confirmar que el sistema integrado opera como se espera. Esta etapa es esencial para garantizar la cohesión y eficacia del sistema en su conjunto.

Ascendiendo en el modelo en V, se realizan pruebas del sistema completo para validar que cumpla con todos los requerimientos funcionales y no funcionales. Las pruebas de aceptación siguen, evaluando el sistema en un entorno real o simulado para asegurar que satisface las necesidades y expectativas de los usuarios finales. Finalmente, se verifica la alineación del sistema con los requerimientos y expectativas de los stakeholders, utilizando un checklist detallado de los requerimientos para guiar estas pruebas. Esta fase completa el ciclo del modelo en V.

1.4.1. Diagrama de bloques

En la figura 1 se plantea el diagrama de bloques que seguirá el proyecto, esto se ajustará de acuerdo con la necesidad que se requiera en el diseño del sistema.

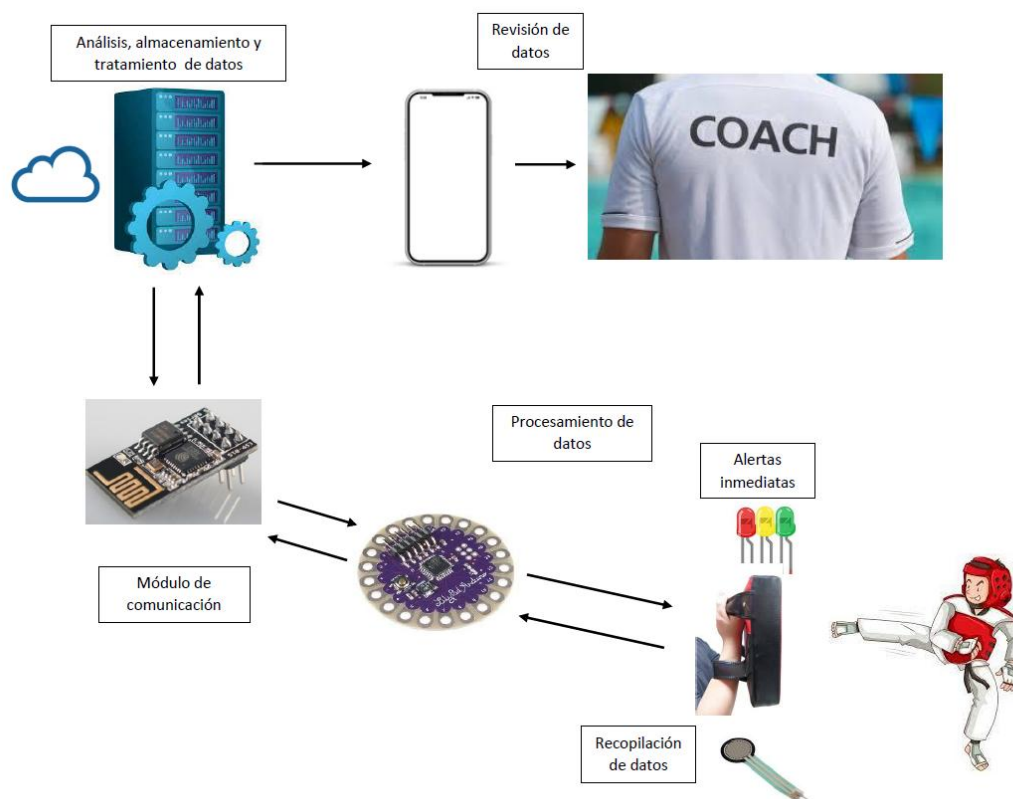
Figura 1 Diagrama de bloques del sistema (Autor)



1.4.2. Arquitectura

En la figura 2 se plantea la arquitectura inicial planteada que tendrá el proyecto, teniendo en cuenta sus fases o bloques correspondientes.

Figura 2 Arquitectura del sistema (Autor)



1.5. Justificación

De acuerdo con un análisis aplicado de la biomecánica deportiva en el Taekwondo, la correcta ejecución de las técnicas de patada es crucial para activar los sensores en los petos y cascos electrónicos, lo que subraya la importancia de la velocidad y la técnica precisa (Yao, 2023). Además, estudios han demostrado que la generación de velocidad pico y energía cinética máxima son determinantes para la efectividad de las técnicas de patada, lo que resalta la necesidad de un entrenamiento enfocado en estos aspectos para mejorar el rendimiento en competiciones (Pieter & Pieter, 1995).

La relevancia de la tecnología en el entrenamiento deportivo se hace eco en la literatura, donde se enfatiza que el análisis técnico y táctico proporcionado por herramientas tecnológicas avanzadas es fundamental para el desarrollo de estrategias de entrenamiento efectivas. Como señala un estudio del *Journal of Strength and Conditioning Research*, los elementos técnico-tácticos son vitales para el éxito en el Taekwondo juvenil, y la tecnología puede ofrecer un marco útil para la planificación y mejora del entrenamiento (Casolino et al., 2012). Esto se complementa con investigaciones que indican la importancia de adaptar las técnicas de patada para maximizar la puntuación, lo que justifica la necesidad de sistemas de entrenamiento que reflejen y se adapten a los criterios específicos de los sistemas de puntuación electrónicos (Gavagan & Sayers, 2017).

Por lo tanto, la justificación para este proyecto se basa en la necesidad evidente de mejorar la técnica de patadas en Taekwondo para cumplir con los estándares de los sistemas de puntuación electrónicos y en el potencial comprobado de la tecnología para proporcionar retroalimentación detallada y mejorar el entrenamiento.

II. Marco teórico

Este capítulo introduce conceptos clave relacionados con el uso de tecnologías en el Taekwondo, específicamente en los sistemas de puntuación electrónicos para petos y cascos. Se aborda cómo estos dispositivos impactan el rendimiento de los deportistas y cómo los atletas deben adaptarse a las demandas de la puntuación electrónica. Además, se exploran aspectos tecnológicos, como los sistemas embebidos y los sensores integrados en los dispositivos de monitoreo, que son esenciales para el entrenamiento inteligente y la mejora de la técnica de los deportistas.

2.1. Historia, Desarrollo y Reglamentación del Taekwondo

2.1.1. Origen y Evolución Histórica

El origen de las artes marciales parece estar intrínsecamente vinculado con la evolución de las civilizaciones asiáticas, siendo al mismo tiempo un reflejo de los desafíos de supervivencia y un componente cultural profundamente arraigado. Las primeras formas de combate, que probablemente surgieron en la India y China hacen más de 4,000 años, habrían evolucionado para servir tanto a necesidades defensivas como a la expresión de valores y tradiciones locales. Este carácter dual de las artes marciales como herramienta de supervivencia y como manifestación cultural es esencial para entender su expansión y adaptación en regiones como Corea y Japón, donde cada sociedad integró y modificó estas técnicas para alinearlas con sus propios contextos culturales.

En Corea, sus primeras artes marciales (subak) serían las interpretaciones de *Wei-Chia*, los sistemas externos de *chuan fa*. Los estilos externos nortños, que eran conocidos por sus técnicas de pie, se combinarían con acrobacias dando lugar con los años al *taekyon*. Uno de los estilos del *chuan fa*, basado en luxaciones e inmovilizaciones, el *chin na*, se introduciría en Japón, a través

de Corea sobre el año 900 d.C., formando la base para *aiki-jutsu*, antecesor del aikido (Bass, 2002). Mientras que en las islas Ryukyu, el *chuan fa* (kempo en japonés) pudo haber entrado directamente desde China, de la que en determinadas épocas fueron un estado tributario, originando el *okinawate* y el *tode*, que darían a su vez origen al *karatedo*.

(Martínez Guirao, 2013).

Diversos estudiosos sostienen que el intercambio cultural en los últimos años de la dinastía Tang, que gobernaba Manchuria en ese momento, jugó un papel en la introducción del sistema de lucha chino chuan fa en Koguryo. Este sistema se adaptó en Corea bajo el nombre de kwonbop. No obstante, algunos investigadores (WTF, 2004) rastrean los orígenes del taekwondo hasta las primeras evidencias arqueológicas de sistemas de combate en la península coreana, las cuales se remontan a la época de los tres reinos, específicamente en el reino de Koguryo (37 a.C. - 688 d.C.).

Tras la ocupación japonesa y al finalizar la Segunda Guerra Mundial, muchos jóvenes que habían emigrado regresaron a Corea, y comenzaron a surgir varias escuelas o kwans dedicadas a la enseñanza de artes marciales, especialmente en Seúl. Cada escuela tenía su estilo particular, influido por la formación y las experiencias previas de sus maestros, quienes en su mayoría habían alcanzado altos rangos, como cinturones negros y danes, principalmente en el karatedo japonés y el chuan fa chino. Estas influencias fueron notorias en los primeros años posteriores a la liberación, pero se fueron diluyendo con el tiempo a través de dos procesos específicos: el "proceso de coreanización," enfocado en recuperar las artes tradicionales coreanas, y el "proceso de deportivización," que destacaba los aspectos deportivos de la práctica (Martínez Guirao, 2013).

En 1971, el Parlamento coreano declaró el taekwondo deporte nacional y promovió la creación de la Kukkiwon, bajo la que se aglutinaron todas sus asociaciones, escuelas y estilos. Poco después, el 28 de mayo de 1973, se fundó la World Taekwondo Federation (WTF), siendo elegido presidente Ung Yong Kim, un abogado callado y fornido que había formado parte de las fuerzas de seguridad del presidente Park Chung Hee y trabajado como agente diplomático en Washington, la ONU y Londres.

(Díaz y Barrientos Varela, 2020)

2.1.2. Taekwondo en el Ámbito Internacional

La Federación Mundial de Taekwondo (WT) es la Federación Internacional (FI) que rige el deporte del Taekwondo y es miembro de la Asociación de Federaciones Internacionales Olímpicas de Verano (ASOIF) y del Comité Paralímpico Internacional (IPC). WT lidera el deporte de combate más inclusivo y accesible, el cual combina los valores de un antiguo patrimonio asiático con los de un deporte de élite global. El Taekwondo se desarrolla sobre una base sólida, mezclando lo tradicional con lo moderno.

La World *Taekwondo* (WT), anteriormente conocida como la Federación Mundial de Taekwondo (WTF), fue fundada el 28 de mayo de 1973. Su creación surgió como una iniciativa para estandarizar las prácticas y reglas del Taekwondo a nivel internacional, especialmente en un momento en que este arte marcial estaba ganando popularidad global.

La fundación de la WT se realizó en Seúl, Corea del Sur, y fue un paso crucial para el desarrollo y la difusión del Taekwondo como un deporte global. La organización se estableció con

el objetivo de promover el Taekwondo en todo el mundo, unificar las reglas de competición, y eventualmente lograr su inclusión en los Juegos Olímpicos.

Aunque en 1963 el taekwondo ya había sido reconocido como deporte en Corea, todavía no ocupaba esa posición internacionalmente. Pero, con los cambios que se venían produciendo en los últimos años, principalmente su homogeneización y deportivización, pronto se conseguiría tal estatus.

(Martínez Guirao, 2013).

Los primeros Juegos Olímpicos en los que se incluyó la modalidad de taekwondo en el calendario fueron los de Seúl 1988, en los que, a propuesta del Comité Organizador Nacional, apareció como deporte de exhibición. En esa edición participaron 120 deportistas y Corea ganó 7 de los 8 pesos en la categoría masculina. En los Juegos de Barcelona 1992 nuevamente se propuso el taekwondo como uno de los deportes de exhibición, junto con la pelota vasca y el hockey sobre patines. En la consecución de este logro fundamental para nuestro deporte ya que es necesario que una modalidad deportiva haya sido deporte de exhibición en dos ediciones de los Juegos para poder entrar en el calendario oficial olímpico

(Díaz y Barrientos Varela, 2020)

2.1.3. Normativas Internacionales en el Taekwondo

De acuerdo con Arias (2018), las reglas para determinar los puntos válidos en Taekwondo son las siguientes:

Áreas de Puntuación:

- El Tronco: Se considera el área de color azul o roja del protector de tronco.
- La Cabeza: Se considera toda la cabeza por encima de la línea inferior del protector de cabeza.

Criterios de Punto(s) Válido(s):

- Se otorgarán punto(s) cuando una técnica permitida impacte en las áreas de puntuación del tronco con el nivel adecuado de impacto.
- Se otorgarán punto(s) cuando una técnica permitida impacte en las áreas de puntuación de la cabeza.
- La validez de la técnica, el nivel de impacto y/o del contacto válido en las áreas de puntuación será determinado por el sistema de puntuación electrónico, excepto las técnicas de puño. Estas valoraciones del PSS no estarán sujetas a Repetición Instantánea de Video.
- El Comité Técnico de la WT determinará el nivel de impacto y la sensibilidad requerida del PSS, usando diferentes escalas en consideración de las categorías de peso, sexo y grupos de edades. En ciertas circunstancias, el delegado Técnico podrá recalibrar la validez del nivel de impacto.

Los Puntos Válidos Son:

- Un (1) punto por un puño válido al protector del tronco.
- Dos (2) puntos por una patada válida al protector de tronco.
- Cuatro (4) puntos por una patada válida con giro al protector de tronco.
- Tres (3) puntos por una patada válida a la cabeza.

- Cinco (5) puntos por una patada válida con giro a la cabeza.
- Un (1) punto se otorgará por cada “Gam-jeom” dado al competidor oponente.

2.1.4. Impacto de los Reglamentos en Técnicas y Estrategias

Si se observan los combates de Pekín 2008 y se comparan con los de Río 2016, se intuye que el protagonismo de la bandal chagui se ha visto reducido significativamente, incrementándose, sin embargo, el uso de técnicas lineales como el miro chagui, especialmente con la pierna adelantada, tanto en ataque como en contra. También se recurrió a este recurso técnico como fase inicial de las correcciones que se dirigían luego a otra zona del cuerpo o con otra trayectoria. No obstante, con el uso de los dispositivos Adidas-KP&P (sistemas electrónicos) parece ser que las acciones de corrección al tronco ven limitada su efectividad o su empleo se acotará a deportistas que consigan ejecutarlas con una mayor contundencia (Díaz y Barrientos Varela, 2020).

La distinta asignación de puntos entre el tronco y la cabeza y la utilización de dispositivos electrónicos propició, hasta el momento, un estilo de competidores longilíneos y muy flexibles en detrimento de otros más potentes o explosivos. La implantación de cascos electrónicos que registraban la existencia o no de contacto y no la potencia de este había incidido todavía más en esta situación, generando auténticos “esgrimistas” cuyo objetivo era alcanzar la cabeza del oponente con un “toque” y no realizar un auténtico golpeo.

Los cambios en los sistemas de registro de puntuación propiciaron también que se generasen nuevas acciones técnicas, bautizadas con nombres no tradicionales (pues no existen en el taekwondo de base). Una de ellas es el monkey kick o golpeo semicircular hacia dentro, que permite contactar con la planta del pie sobre el protector de tronco del adversario en distancia de clinch.

(Díaz y Barrientos Varela, 2020).

Otra de ellas es el scorpion kick, técnica que consiste en golpear la nuca del adversario realizando un nako chagui por la espalda en el cuerpo a cuerpo —con guardia cerrada— que permite el contacto de la planta del pie con la nuca del oponente. Existen auténticos especialistas en el uso de esta técnica, como el tunecino Oussama Oueslati, a quien favorecía, para poder ejecutarla, su gran estatura y envergadura de piernas. La polémica desatada por la falta de espectacularidad que los espectadores han criticado en la competición de taekwondo de Río 2016 provocó que la WTF introdujese nuevos cambios que buscan hacer a este deporte más atractivo manteniendo la objetividad, e impedir su deriva hacia una modalidad más “esgrimística” que de golpes.

2.2. Biomecánica y Sistemas de Puntuación en Taekwondo

2.2.1. Principios de Biomecánica en Movimientos de Taekwondo

La biomecánica permite comprender los diferentes factores que permiten realizar determinado gesto técnico, factores como velocidad fuerza, velocidad de ejecución de la técnica, ángulos, etc. (Monterrosa, Díaz, y Jiménez, 2008; Granda, Loachamin, y Arla, 2016),

El análisis biomecánico cumple cuatro etapas (Reyes, 2013). En primer lugar, un análisis biomecánico cumple con la etapa de un diagnóstico de las deficiencias técnicas del deportista, posteriormente se determina las correcciones biomecánicas en función al diagnóstico realizado, la aplicación de ejercicios para un perfeccionamiento en el aspecto técnico, y finalmente la evaluación de la efectividad. También se afirma que para un análisis biomecánico en el taekwondo deben considerarse aspectos como tiempo de ejecución, de impacto, fuerza máxima y fuerza relativa. (Torres, 2010)

2.2.2. Análisis Biomecánico de Técnicas de Patada

2.2.2.1. Bandal chagui

Monterrosa y colaboradores (2018) realizaron un análisis biomecánico de la patada bandal chagui en el arte marcial del taekwondo. En su estudio, demostraron la importancia de la velocidad y la masa del segmento como factores clave, los cuales se combinan en diferentes ecuaciones aplicadas en la investigación. Observaron que la formación de los ángulos, especialmente en el segmento que realiza el gesto técnico, mostraba un patrón de campana invertida en todos los sujetos, indicando un inicio en un ángulo amplio que se cierra en la mitad del movimiento y se vuelve a abrir al alcanzar el objetivo.

En cuanto a los recorridos del centro de masa del cuerpo humano, encontraron que el sujeto con mayor masa logró un mayor desplazamiento. Este dato sugiere que los sujetos con mayor proporción de masa desplazan más para alcanzar el objetivo. Sin embargo, es importante destacar que los datos del centro de gravedad pueden variar en sujetos obesos debido a la dificultad en la toma de datos del punto antropométrico para la medición.

Los autores enfatizaron la precisión en la medición del desplazamiento del centro de gravedad, ya que de ello depende el número de ecuaciones que conforman los datos de manera secuencial. Además, señalaron que la aceleración, un valor que conforma la fuerza por su ecuación física, depende de valores de la velocidad final y la velocidad inicial sobre el tiempo que transcurre la acción.

En términos de velocidad, el sujeto con el valor más alto fue AM, con una velocidad de 11,3 m/s, equivalente a 40.6 km/h. La velocidad se determinó por el movimiento de masa del

centro de gravedad del pie, considerando el testigo en el fotograma para las mediciones de los desplazamientos del pie en el espacio.

Finalmente, Monterrosa et al. (2018) concluyeron que no se puede realizar una investigación biomecánica sin los soportes físicos y matemáticos, ya que de ellos depende el factor numérico y comparativo. Además, resaltaron la importancia de la fisiología, la antropometría y otras áreas de la salud en el análisis del movimiento.

En la figura 3 se aprecian los movimientos por pasos que se deben seguir para realizar la patada bandal chagui.

Figura 3 Patada bandal chagui (Academia de Taekwondo OCRAM, 2018)



2.2.2.2. Ap chagui

Loachamín, et al., (2017) realizaron un análisis biomecánico de la patada Ap Chagüi en taekwondistas de diferentes niveles de cinturón. El estudio permitió identificar características clave para la modelación del entrenamiento deportivo, en línea con la literatura existente (Estevan, Jandacka y Falco, 2018). Mediante la prueba U de Mann-Whitney, se encontraron diferencias en los gestos técnicos entre los taekwondistas de cinturón blanco y negro, destacando la influencia de la velocidad en el éxito motriz.

Se observó que no existían diferencias significativas en el ángulo del brazo durante las distintas fases de la patada Ap Chagüi entre los dos grupos, aunque los taekwondistas de cinturón negro mostraron mayores rangos promedio. En cuanto a los miembros inferiores, se detectaron diferencias significativas en el rendimiento técnico en el ángulo de la pierna en la primera fase de la patada, con mayores rangos promedio en los deportistas de cinturón negro.

El estudio también reveló que el centro de gravedad y la velocidad son factores determinantes en la técnica de patada, con diferencias significativas en la velocidad inicial y final entre los dos grupos. Estos hallazgos sugieren que el rendimiento deportivo es multifactorial y debe considerarse en conjunto con otros aspectos como los modelos de entrenamiento y las características individuales de los deportistas.

En la figura 4 se aprecian los movimientos por pasos que se deben seguir para realizar la patada ap chagui.

Figura 4 Patada bandal chagui (Academia de Taekwondo OCRAM, 2018)



2.2.3. Evolución de los Métodos de Puntuación

El sistema de puntuación electrónica en Taekwondo fue implementado oficialmente por la Federación Mundial de Taekwondo (WT, anteriormente conocida como WTF) en los eventos internacionales a partir del año 2009. Este cambio representó un avance significativo en la forma de juzgar y puntuar los combates, buscando aumentar la objetividad y precisión en la evaluación de los golpes. La adopción de este sistema se dio después de varios años de desarrollo y pruebas, y su implementación coincidió con un período de modernización y profesionalización del Taekwondo como deporte olímpico.

Según Morales (2010), el reglamento de arbitraje en Taekwondo establece lo siguiente respecto al uso de protectores equipados electrónicamente:

1. Los puntos válidos que se realicen en la zona media del torso serán registrados automáticamente por el transmisor incluido en los protectores electrónicos. Si se ejecuta una técnica de giro válida en el protector del torso y se marca un "punto válido" automáticamente por el transmisor del protector electrónico del torso, los jueces procederán a anotar la "validación del giro".
2. En cuanto a los puntos válidos en la cabeza, estos serán marcados por el juez utilizando un pulsador electrónico o una hoja de papel de juez. En el caso de una técnica de giro válida en la cabeza, los jueces registrarán los puntos válidos y procederán a "validar el giro".

2.3. Tecnología y Machine Learning en el Entrenamiento Deportivo

2.3.1. Herramientas y Dispositivos Tecnológicos en Entrenamiento

En la actividad física y el deporte la introducción de las TIC's se ha convertido en un elemento denotativo que no queda al margen del desarrollo deportivo y la contemporaneidad, su empleo ha permitido gestionar la eficiencia y eficacia del proceso de preparación, lo que se corrobora en múltiples investigaciones con un enfoque práctico, los cuales, según Pulido, Sánchez-Oliva, Sánchez-Miguel, González-Ponce & García-Calvo (2016) tienen como punto de encuentro el empleo de herramientas tecnológicas tales como uso de dispositivos electrónicos, emails, páginas webs, vídeos, acelerómetros, entre otros.

En correspondencia la armonización entre deporte e informática se ha logrado visualizar a través de innumerables elementos, dentro de los que se encuentran los softwares empleados para perfeccionar sistemas de planificación, para el análisis biomecánico de la técnica, para el análisis de acciones de alto nivel de complejidad, para captura de datos por telemetría, se han creado herramientas, aportadas por el ordenador, dentro de las que se destaca el "esporttester" hoja de

anotación, plataforma de fuerzas, electro miógrafo, cicloergómetro, analizador de gases, GPS, y muchos otros de gran valor

(Guillén et al., 2018)

2.3.2. Machine Learning

Machine learning es la ciencia que hace que los ordenadores “aprendan” a partir de los datos. En vez de programar, paso a paso, cada solución específica para cada necesidad planteada, tal y como se realiza en el enfoque de la programación convencional, el área de machine learning está dedicada al desarrollo de algoritmos genéricos que pueden extraer patrones de diferentes tipos de datos. De esta manera, un programa de machine learning destinado, por ejemplo, a clasificar números escritos a mano, no va a diferir sustancialmente de un programa destinado a la clasificación de las imágenes de señales de tráfico: ambos se basarán en la existencia de algún tipo de algoritmo de machine learning que clasifique datos etiquetados.

(Bobadilla, 2020).

2.3.2.1. Tipos de Machine Learning

Con el objetivo de poder abordar cualquier tarea específica, el ingeniero de datos debe conocer algunos conceptos importantes de machine learning, así como las diferentes opciones existentes, las medidas de calidad más utilizadas, etc. Los conocimientos básicos incluyen la identificación de las tareas, empezando por la clasificación de los problemas de machine learning en alguno de los siguientes tipos:

- Aprendizaje supervisado

- regresión
- clasificación
- Aprendizaje no supervisado
 - clustering (agrupamiento)
 - reducción de dimensiones
- Aprendizaje semi-supervisado
 - Aprendizaje por refuerzo

(Bobadilla, 2020).

El aprendizaje supervisado en machine learning se aplica cuando cada dato, o conjunto de datos de entrada (muestra), tiene asociada una etiqueta. Por ejemplo, en un conjunto de imágenes en las que cada una contiene metadatos como etiquetas (“perro”, “pájaro”, “gato”), el modelo puede ser entrenado para predecir la etiqueta de una nueva imagen no incluida en el conjunto inicial. Este proceso constituye un problema de clasificación.

El aprendizaje no supervisado, en cambio, utiliza datos no etiquetados. Una de sus aplicaciones más conocidas es el clustering, donde el objetivo es agrupar muestras para identificar patrones. Este tipo de agrupamiento es útil, por ejemplo, para clasificar clientes en servicios en línea o agrupar productos en un comercio electrónico. En el aprendizaje semi supervisado, se trabaja con conjuntos de datos donde solo una parte de las muestras están etiquetadas, combinando métodos supervisados y no supervisados.

El aprendizaje por refuerzo se inspira en mecanismos naturales, permitiendo que el algoritmo aprenda a partir de la interacción con un entorno real o simulado, adaptándose a través

de recompensas o penalizaciones recibidas del entorno, y presenta gran potencial para futuras aplicaciones.

2.4. Modelo en V para Desarrollo de Sistemas de Entrenamiento

2.4.1. Introducción al Modelo en V

Para la Unidad de Desarrollo Tecnológico en Inteligencia Artificial, el modelo V o de Cuatro Niveles es un método robusto utilizado para proyectos pequeños que involucran hasta cinco personas y su objetivo es regular el proceso, de manera que se minimicen los riesgos y se garantice la calidad del proyecto.

(Álvarez, 2017).

El modelo en V se propone como una alternativa más eficiente que el modelo en cascada, ya que organiza el desarrollo en un esquema donde cada etapa de creación tiene una fase paralela de verificación y validación. Esto se fundamenta en el principio de que cada fase debe concluir con un resultado comprobable.

2.4.2. Etapas del modelo en V

El modelo en cuestión se estructura en cuatro etapas clave, cada una correspondiente a una fase distinta en el ciclo de vida del desarrollo de software.

En la primera etapa, el enfoque principal es el cliente para quien se está desarrollando el sistema; esta etapa marca tanto el comienzo como la conclusión del proyecto e incluye el análisis de requerimientos, cuyo resultado exitoso se mide por la aprobación del usuario respecto a lo acordado para el sistema.

La segunda etapa se centra en los aspectos funcionales del sistema, abarcando la fase de diseño, cuyo éxito se determina por la agrupación efectiva de los resultados de las pruebas de integración.

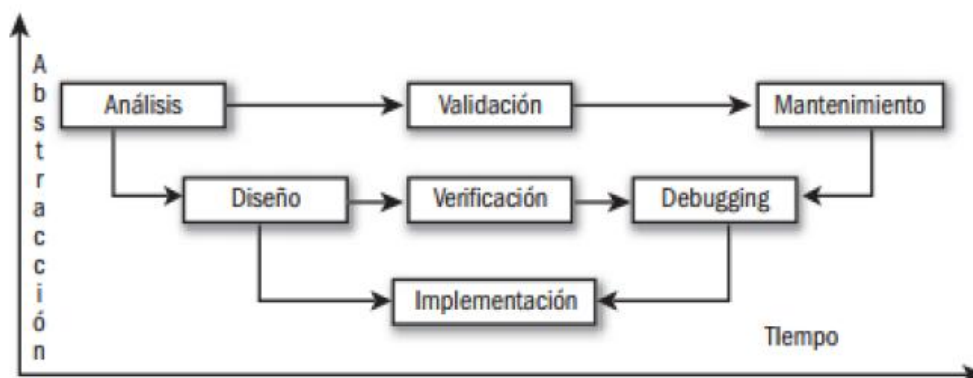
La tercera etapa se dedica a definir los componentes de hardware y software más adecuados para el sistema, correspondiendo a la fase de arquitectura del sistema, y su éxito se evalúa a través de los resultados obtenidos en las pruebas de integración.

Finalmente, la cuarta etapa se ocupa del desarrollo de los elementos y módulos del sistema, en la fase de implementación, donde el criterio de éxito son las pruebas unitarias realizadas.

A medida que se progresa a través de los niveles en el modelo de desarrollo, la relación entre cada fase de desarrollo y su correspondiente fase de verificación y validación se va distanciando, lo que se representa gráficamente como una 'V'. Esta distancia gráfica busca reflejar el tiempo transcurrido entre una fase de desarrollo y su fase de verificación correspondiente. Debido a que cada fase del proceso es sometida a validación y verificación, es común que se produzcan iteraciones repetidas entre los niveles, lo que puede resultar en un proceso de desarrollo extenso y laborioso, a pesar de la solidez del método.

En la figura 5 se ve la representación gráfica del esquema del Modelo en V con cada una de sus etapas.

Figura 5 Etapas del modelo en V



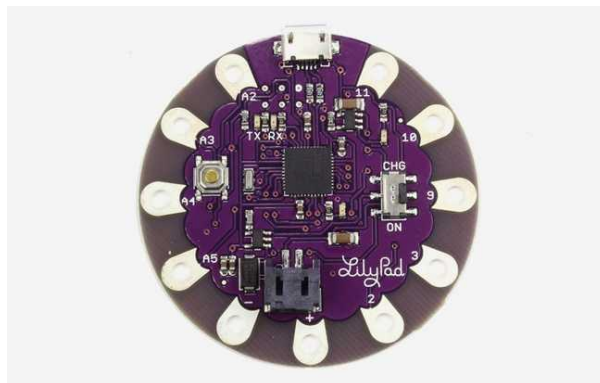
2.5. Placas Electrónicas y Servicios en la Nube

2.5.1. Arduino LilyPad

El LilyPad Arduino USB se fundamenta en el microcontrolador ATmega32u4 y está equipado con 9 pines de entrada/salida digital, de los cuales cuatro pueden funcionar como salidas PWM y otros cuatro como entradas analógicas. Además, cuenta con un resonador de 8 MHz, una conexión micro USB, un conector JST para baterías LiPo de 3.7V y un botón de reset. Este dispositivo incluye todo lo necesario para operar el microcontrolador, permitiendo su conexión a un ordenador mediante un cable USB o su alimentación a través de una batería para su puesta en marcha.

La figura 6 brinda una representación visual de la placa Arduino LilyPad que será usada dentro del proyecto.

Figura 6 Arduino LilyPad (Arduino, 2024)



Una característica distintiva del LilyPad Arduino USB, en comparación con modelos anteriores de LilyPad, es que el ATmega32u4 incorpora la funcionalidad de comunicación USB, lo que hace innecesario el uso de un adaptador USB a serie externo. Esto posibilita que el LilyPad Arduino USB sea reconocido por un ordenador conectado como un ratón o teclado, además de actuar como un puerto serial virtual (CDC) / COM. Las implicaciones adicionales de esta característica para el funcionamiento de la placa se detallan en la página de inicio de LilyPad Arduino

El ATmega32U4 del LilyPad Arduino USB ya viene con un bootloader preinstalado, lo que permite cargar nuevo código sin necesidad de un programador de hardware externo. Este proceso se realiza utilizando el protocolo AVR109.

Además, existe la posibilidad de programar el microcontrolador directamente, obviando el bootloader, a través del encabezado de Programación en Serie en Circuito (ICSP). Aunque los orificios son demasiado pequeños para insertar pines directamente, se pueden utilizar pines macho en el conector ISP del programador y presionarlos contra el encabezado ICSP en la placa (desde arriba).

(Arduino, 2024)

2.5.2. Servicios en la Nube

El concepto Cloud, en español Nube, engloba un conjunto de aplicaciones y servicios informáticos cuya principal característica es que se encuentran alojados en Internet. El usuario puede acceder instantáneamente, en cualquier momento y esté donde esté, a sus datos. El Instituto Nacional de Estándares y Tecnologías (NIST) da una definición formal sobre el significado de Cloud: ‘Cloud es un modelo para permitir un acceso de red adecuado, desde cualquier sitio y bajo demanda a un conjunto compartido de recursos informáticos que se pueden proporcionar rápidamente y lanzar con un mínimo esfuerzo de gestión o interacción con el proveedor de servicios.’ Suele decirse que cualquier cosa que pueda hacerse en informática puede trasladarse a la Nube, es decir, a la Red. La Nube permite a los usuarios preocuparse sólo de lo que quieren hacer y no de cómo deben configurar todo su entorno para conseguir lo que necesitan. Gracias a esto, los usuarios podrán centrarse en utilizar una aplicación sin preocuparse de si tiene instalado un software específico o de las características del equipo.

(Celaya Luna, A. 2014).

2.5.2.1. Almacenamiento en la nube

El almacenamiento en la nube o cloud storage es un nuevo modelo de almacenamiento eficiente, que proporciona seguridad a los datos que en él se almacenan debido a la independencia de los datos y demás información de los propios equipos. Esta característica permite que los datos estén siempre disponibles independientemente del equipo de trabajo, protegiendo de esta manera estos ante indisponibilidad de dichos equipos, así como proporcionando acceso a la información desde cualquier ubicación o dispositivo.

(Asta Alarcón, 2017)

El almacenamiento en la nube, o cloud storage, es un modelo moderno y eficiente que ofrece alta seguridad para los datos al desvincularlos de los dispositivos físicos. Esta independencia garantiza que la información esté siempre accesible, sin depender del equipo específico de trabajo, protegiéndola contra posibles fallos o inactividad de estos dispositivos y permitiendo el acceso a los datos desde cualquier lugar y dispositivo.

El almacenamiento en la nube de forma pública es el de uso común, donde la gente común lo utiliza con el propósito de mantener información en la nube y que en muchas ocasiones no representa ningún riesgo su pérdida o modificación por terceros, esta información normalmente es alojada en servidores públicos externos al usuario donde este puede acceder de forma gratuita y/o en ocasiones cancelando un mínimo costo por este tipo de servicio.

El almacenamiento en la nube de forma privada es el tipo de configuración que se puede encontrar a nivel de las instalaciones de una empresa en donde se requiere compartir información internamente entre sus sucursales, por ejemplo, en ella la configuración es mucho más avanzada puesto quien la administre es quien decide quien almacena la información, con quien la comparte y el tipo de información que puede alojar.

(Buechley & Eisenberg, 2008)

III. Desarrollo experimental

En el siguiente capítulo se abordarán los aspectos relacionados con el diseño del sistema electrónico, incluyendo la determinación de los mecanismos y parámetros necesarios. Se detallará también la selección del hardware y software adecuados para la detección de puntajes en el sistema de Taekwondo, utilizando la metodología del modelo en V. Además, se desarrollará la interfaz que permitirá la visualización de los resultados, facilitando la retroalimentación en tiempo real durante las sesiones de entrenamiento.

3.1. Situación actual

El Club de Artes Marciales “Black Panthers Ecuadcuba” se encuentra ubicado en el sur de Quito, fundado en el año 2010 por el maestro 4° DAN Lic. Eddy Julio Cajigal, esta academia a lo largo de los años ha entrenado a docenas de personas dentro de las artes marciales, haciendo especial énfasis en el Taekwondo, ya que cuenta con cursos permanentes de otras artes marciales como Kick Boxing y Defensa personal; actualmente, este club se encuentra reconocido y avalado por la Asociación de Taekwondo de Pichincha, quien es el ente rector del deporte a nivel provincial.

La figura 7 muestra el logo actual del Club de Artes Marciales Black Panthers Ecuadcuba.

Figura 7 Logo Ecuadcuba Fuente: Club de Artes Marciales Ecuadcuba



Al momento el club cuenta con alrededor de 60 alumnos de distintas edades y categorías, existen practicantes novatos, intermedios y avanzados, en las categorías Preinfantil 2 (6-7 años), Infantil A (8-9 años), Infantil B (10-11 años), Cadetes (12-14 años), Junior (15-17 años) y Senior (+18 años), tanto en la modalidad de Poomase como en la de Combate.

En el club existen alrededor de 31 alumnos novatos, 24 estudiantes intermedios y 7 deportistas avanzados de distintas categorías, por lo que la mayor parte de los practicantes la comprenden novatos que están en un proceso formativo técnico y táctico, quienes requieren de una preparación enfocada al desarrollo básico del arte marcial y del deporte.

Existe un grupo de deportistas del club que pertenecen a las distintas categorías de las preselecciones y selecciones de la Concentración Deportiva de Pichincha en la modalidad de combate, quienes asisten regularmente a entrenamientos dentro de las instalaciones de la institución, representando al club y a la provincia en eventos provinciales, nacionales e internacionales.

Haciendo énfasis en la modalidad de combate, el sistema de puntuación actual se lo realiza con marcación electrónica, la cual se la realiza con sensores tanto en las empuñaduras como en los petos y cascos, los cuales transmiten de manera inalámbrica una señal de la fuerza y los puntos de

contacto válidos a la mesa central de los jueces, que, de acuerdo con la zona del impacto, sea peto o cabeza, asigna la puntuación correspondiente de acuerdo con la siguiente descripción:

- 1 punto: puño válido al peto.
- 2 puntos: ataque de pie válido al peto.
- 4 puntos: ataque de pie con giro válido al peto.
- 3 puntos: ataque de pie válido a la cabeza.
- 5 puntos: ataque de pie con giro válido a la cabeza.

Para la modalidad de combate, las sesiones de entrenamiento son planificadas y organizadas por el director del club, en estas se busca mejorar el aspecto técnico y táctico en los estudiantes de nivel novatos y reafirmar estos ámbitos en los practicantes intermedios y avanzados, para lograr este cometido se practican sesiones de entrenamiento de 60 minutos, las cuales tienen distintas modalidades, existen ejercicios por repeticiones contadas por el entrenador y por un tiempo determinado, cada tipo tiene una finalidad específica, que está relacionada con el plan de enseñanza del club.

Teniendo en consideración que el director del club es la parte fundamental del programa de enseñanza de los estudiantes, se considera como un ente primordial para poder hacer el análisis de los requerimientos que debe tener el proyecto, así como también considerar a un grupo de estudiantes para que brinden su opinión sobre las características que les beneficiarían a ellos como usuarios del prototipo.

3.2. Metodología del modelo en V y Stakeholders

El modelo V es un enfoque de desarrollo de sistemas y software que representa las fases del ciclo de vida del proyecto en forma de V secuencial. Las etapas de desarrollo están directamente vinculadas a las actividades de verificación y validación de este modelo. La v

desciende a una fase descendente que involucra actividades de diseño y desarrollo, mientras que la fase ascendente involucra pruebas y verificación para garantizar que cada nivel de desarrollo cumpla con los requisitos definidos.

Los stakeholders son aquellos actores que tienen un interés directo o indirecto en un proyecto o sistema. Podrían ser un factor de desarrollo, un factor de éxito del proyecto o una combinación de ambos. Durante el ciclo de vida del proyecto, las partes interesadas, como los usuarios finales, los clientes, los patrocinadores, los miembros del equipo de desarrollo, los reguladores y otros actores clave, deben tener en cuenta sus expectativas y necesidades.

3.2.1. Estudio de los Stakeholders

Los stakeholders son fundamentales dentro del presente trabajo, ya que son quienes participan de forma directa e indirecta en el desarrollo y diseño del prototipo, ellos son quienes determinarán los parámetros necesarios que deberá cumplir el diseño para que este sea funcional de acuerdo con sus necesidades tanto como usuario, como entrenador o como desarrollador del prototipo.

Los Stakeholders definirán los requerimientos necesarios para que el sistema pueda trabajar de una manera correcta, de acuerdo con sus necesidades enfocadas al objetivo principal del proyecto.

En la Tabla 1 se aprecian los stakeholders que se ven involucrados dentro del proyecto.

Tabla 1 Lista de stakeholders

Lista de Stakeholders	
1.	Club de Artes Marciales Ecuadecuba

-
2. Estudiantes novatos de Taekwondo
 3. Deportistas intermedios y avanzados de Taekwondo
 4. Lic. Eddy Cajigal (Director y entrenador del club)
 5. MSc. Carlos Vásquez (Director del trabajo de titulación)
 6. MSc. Jaime Michilena (Asesor del trabajo de titulación)
 7. Jhonatan López (Diseñador del proyecto)
-

Fuente: Autor

Para poder cumplir con las expectativas que se presentan por los stakeholders se plantean los requerimientos de arquitectura en hardware y software, lo cual brinda un diseño integral que complementa las necesidades de cada usuario con la funcionalidad del sistema.

3.2.2. Construcción de atributos de los requerimientos

Es fundamental definir las características de los atributos de los requisitos, basándose en tres pilares clave: los *stakeholders*, el sistema y la arquitectura. La correcta integración y desarrollo de estos componentes es crucial para asegurar el cumplimiento exitoso de los objetivos del proyecto.

3.2.2.1. Nomenclatura de los requerimientos

En la Tabla 2 se definen los requerimientos necesarios del proyecto con sus respectivos acrónimos, para que de esta manera sea más fácil el manejo de la información.

Tabla 2 Nomenclaturas requerimientos

Tipo de requerimiento	Acrónimo
Stakeholders	STSR
Sistema	YSR
Arquitectura	SSRH

Fuente: Autor

3.2.3. Requerimientos de Stakeholders

El sistema debe cumplir con las necesidades y expectativas del usuario, por lo que se han enumerado ciertos requisitos de usuario, priorizándolos en tres niveles: alta, media y baja. La prioridad alta corresponde a aquellos parámetros esenciales que el usuario espera ver en el prototipo, mientras que la prioridad baja incluye aspectos que pueden ser opcionales. En la Tabla 3 se presentan los requisitos de los *stakeholders* clasificados según esta priorización.

3.2.3.1. Encuesta de requerimientos

La presente encuesta está dirigida a los estudiantes novatos y deportistas intermedios y avanzados del Club de Artes Marciales “Ecuadcuba”, con la finalidad de recopilar información sobre los requerimientos que los usuarios esperan que fuesen parte del prototipo, ya que son ellos quienes podrán hacer uso del sistema y realizarán el pateo correspondiente.

La encuesta constará de 7 preguntas las cuales serán de estructura cerrada para que sea más fácil brindar las respuestas a las personas encuestadas. El formato de la encuesta se encuentra adjunta en el Anexo 2

3.2.3.2. Tamaño de la muestra

El tamaño de la muestra es el número de personas el cuál es necesario aplicar la encuesta para que los datos obtenidos puedan ser determinados como confiables. El tamaño de la muestra se obtiene a partir de la siguiente fórmula.

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$

Donde:

n =tamaño de la muestra

N = tamaño de la población

σ = Desviación estándar de la población, generalmente es de un valor contante 0.5.

Z = nivel de confianza.

e = error muestral (va desde 1% hasta 9%)

3.2.3.3.Determinación de parámetros

En este segmento se definirán las variables con las que se trabajarán en este muestreo, aplicándolas en la fórmula.

Para determinar el valor de N se tomó como referencia la cantidad de estudiantes que asisten regularmente a entrenamientos y llevan más de 6 meses formando parte del club “Ecuadcuba” (Anexo 1).

N = 40 estudiantes regulares que tienen más de 6 meses de formación

$\sigma = 0.5$

$Z = 1.96$

$e = 0.09$

$$n = \frac{40(0.5)^2(1.96)^2}{(40 - 1)(0.09)^2 + (0.58)^2(1.96)^2}$$

$$n = 23.89 \approx 24$$

Como resultado para un tamaño de muestra ideal se obtiene que la encuesta debe ser realizada a 24 estudiantes del club “Ecuadcuba”.

Para obtener los datos de una manera mucho más eficiente la encuesta (Anexo 2) se realiza a través de la plataforma Google Forms, ya que los estudiantes pueden acceder de una manera mucho más fácil a la encuesta mediante un enlace URL o un código QR.

De igual forma se estructura una encuesta dirigida únicamente hacia el director del club (Anexo 4), ya que mediante esta se puede obtener información importante para ser considerada al momento de determinar los requerimientos.

Con base en los resultados obtenidos de las encuestas realizadas (Anexo 3 y Anexo 5) a partes interesadas directas, como el director y estudiantes del Club de Artes Marciales Ecuadcuba, se presentan los requisitos operativos y de usuario en la Tabla 3. Esta tabla detalla los requisitos y expectativas percibidas de cada *stakeholder*, proporcionando una visión integral de las necesidades identificadas.

Tabla 3 Requerimientos de *stakeholders*

STSR				
N°	Requerimientos	Prioridad		
		Alta	Media	Baja
Requerimientos operacionales				
STSR1	El sistema inteligente debe ser implementado dentro del club de Artes Marciales Ecuadcuba	X		
STSR2	Sistema compacto	X		

STSR3	El sistema inteligente debe contar con una base de datos para el registro de usuarios	X	
STSR4	Poseer una interfaz de visualización para el entrenador o evaluador	X	
STSR5	Brindar una retroalimentación para el usuario	X	
STSR6	El sistema debe permitir la opción de trabajar por repeticiones o por tiempo de ejecución	X	
Requerimientos de uso			
STSR7	Interfaz amigable con el usuario	X	
STSR8	Visualización en tiempo real de las patadas	X	
STSR9	Sesión de entrenamiento establecida	X	
STSR10	Conteo total de golpes correctos e incorrectos por cada sesión de entrenamiento	X	
STSR11	Tener una interfaz configurable para poder trabajar en modo por repeticiones o por tiempo	X	
STSR12	Poseer una interfaz de visualización del impacto de los golpes por sensor		X

Fuente: Autor

3.2.4. Requerimientos del Sistema

Los requerimientos funcionales del sistema especifican el comportamiento esperado y las funciones esenciales que debe cumplir, abarcando tanto la operatividad práctica como las necesidades físicas del sistema. En la Tabla 4 se detallan los requerimientos funcionales clave, indispensables para asegurar una implementación efectiva y alineada con los objetivos del proyecto.

Tabla 4 Requerimientos del sistema

SYSR			
Nº	Requerimientos	Prioridad	Relación

		Alta	Media	Baja	
Requerimiento de uso					
SYSR1	El sistema debe estar dentro de una sesión de entrenamiento con dirección de un entrenador	X			STSR1
SYSR2	Comunicación entre el microcontrolador Arduino LilyPad y Windows	X			
SYSR3	Lectura de datos en tiempo real	X			STSR8
SYSR4	El sistema debe tener una interfaz para visualizar la gráfica del impacto de los golpes			X	STSR12
SYSR5	Retroalimentación visual para el usuario	X			STSR4
SYSR6	Conexión con una base de datos para poder tener un registro de usuarios			X	STSR3
Requerimientos de performance					
SYSR7	Mostrar una ponderación de golpes correctos e incorrectos al finalizar la sesión	X			STSR10
SYSR8	Notificaciones dirigidas hacia el entrenador		X		STSR4
SYSR9	Mostrar una gráfica de la distribución de los golpes			X	STSR12
SYSR10	Acceso a la información de la guía de usuario para la ejecución de los distintos modos		X		
SYSR11	Tener una plataforma fácil de usar para los practicantes y entrenadores	X			STSR7
Requerimientos de estados					
SYSR12	El sistema debe tener un menú de selección del tipo de entrenamiento		X		STSR11
SYSR13	El sistema debe tener un botón de inicio de la sesión	X			
SYSR14	El sistema debe tener un botón para cancelar la sesión	X			
SYSR15	El sistema debe tener una opción para visualizar el historial de entrenamientos		X		STSR3
Requerimientos de interfaz					
SYSR16	El sistema debe interactuar con los sensores de fuerza	X			
SYSR17	Obtención en tiempo real de los valores de fuerza	X			

SYSR18	El sistema debe contar con una interfaz fácil de usar para el usuario y entrenador	X	STSR7
SYSR19	La computadora debe tener una conexión inalámbrica para transmitir y recibir datos	X	
SYSR20	Ordenador con comunicación serial compatible con Arduino	X	
Requerimientos físicos			
SYSR21	El sistema debe ser compacto adaptado a un pateador	X	STSR2
SYSR22	Sensores de fuerza distribuidos en el pateador	X	
SYSR23	Tener comunicación mediante un enlace inalámbrico	X	
SYSR24	Tener un espacio adecuado para colocar la PC	X	
SYSR25	Área de entrenamiento adecuada para poder hacer uso del sistema	X	STSR1

Fuente: Autor

3.2.5. Requerimientos de arquitectura

En la Tabla 5 se presentan los requerimientos de arquitectura (SRSH), los cuales definen los requisitos de diseño del sistema en términos lógicos, de diseño, software, hardware y componentes eléctricos. Estos requerimientos proporcionan la base para seleccionar los elementos de hardware y software adecuados, garantizando la integración efectiva y el funcionamiento óptimo del sistema. Las especificaciones arquitectónicas establecen así las directrices esenciales para la implementación de la tecnología en el proyecto.

Tabla 5 Requerimientos de arquitectura

SRSH					
Nº	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
Requerimientos lógicos					
SRSH1	Compatibilidad entre Windows y Arduino	X			SYSR3

SRSH2	Compatibilidad entre Arduino LilyPad y el módulo de comunicación inalámbrica	X	
SRSH3	Compatibilidad de librerías de Arduino para Arduino LilyPad	X	
SRSH4	Conexión de base de datos con la nube	X	STSR3
SRSH5	Compatibilidad entre el módulo de comunicación y el servidor	X	SYSR23
Requerimientos de diseño			
SRSH6	Dispositivo ubicado estratégicamente para que puedan soportar golpes sin dañarse	X	
SRSH7	Interfaz amigable con el usuario	X	STSR7
SRSH8	El dispositivo debe estar ubicado en una zona que no sea manipulado por el usuario	X	
SRSH9	Los sensores de fuerza deben estar distribuidos adecuadamente para poder generar gráficas precisas	X	STSR12
SRSH10	El dispositivo no debe afectar a la movilidad del dispositivo o su funcionalidad	X	
Requerimientos de software			
SRSH11	Se necesita de un software compatible con Arduino para Windows	X	
SRSH12	Se requiere compatibilidad entre librerías de Arduino	X	SYSR2
SRSH13	Compatibilidad con librerías para la creación de la interfaz gráfica	X	
SRSH14	Se necesita de una plataforma en la nube adecuada para soportar las necesidades del proyecto, permitiendo la escalabilidad.	X	
SRSH15	Se necesita un sistema operativo con soporte para lenguajes de programación de código abierto que permita la creación de aplicaciones personalizadas.	X	
Requerimientos de hardware			
SRSH16	El sistema debe ser compatible con Windows	X	
SRSH17	Se requiere de sensores de fuerza que sean flexibles	X	
SRSH18	Se requiere de una plataforma que tenga la capacidad de realizar el procesamiento de datos	X	

SRSH19	Los sensores deben soportar al menos 8 Newtons de fuerza	X	
SRSH20	El módulo de comunicación debe ser compatible con Windows y con Arduino	X	SYSR19
SRSH21	Indicadores visuales para la visualización del estado del dispositivo		X STSR5
Requerimientos eléctricos			
SRSH22	Duración de la batería de al menos 60 minutos	X	
SRSH23	Fuente de alimentación para carga de la batería	X	
SRSH24	Adaptadores para la distribución de energía	X	

Fuente: Autor

3.2.6. Benchmarking (Selección de Hardware y Software)

La elección de hardware y software en el proyecto se fundamenta en los requisitos previamente definidos para satisfacer las necesidades de las partes interesadas (STRS, SYSR, SRSH). Para garantizar una selección óptima, se emplean tablas comparativas que permiten evaluar cada componente en función de su alineación con los requisitos establecidos. La evaluación se realiza mediante un sistema de puntuación, donde:

- "2" indica que el componente cumple completamente con el requisito,
- "1" señala un cumplimiento parcial, y
- "0" refleja la falta de cumplimiento.

Esta metodología facilita una comparación objetiva, asegurando que el hardware y software seleccionados sean los más adecuados para el proyecto.

3.2.6.1. Hardware

Para la selección de Hardware se realiza considerando los parámetros de la Tabla 5 en la que se demuestran los requerimientos de arquitectura. Aquí se definen los elementos de hardware que serán usados, entre ellos el microcontrolador Arduino LilyPad, los sensores y el módulo de comunicación.

3.2.6.1.1. Microcontrolador

Para la elección de este parámetro se lo realiza con base en los parámetros planteados a continuación en la Tabla 6, en donde se determinará el mejor microcontrolador que se adapte con las características y requerimientos del sistema, se plantean 5 requerimientos a tomar en cuenta para 3 opciones de microcontroladores.

Tabla 6 Selección de microcontrolador

Selección de microcontrolador						
Hardware	Requerimientos					Valor total
	SRSH1	SRSH6	SRSH8	SRSH10	SRSH16	
Arduino UNO	2	0	0	0	2	4
Arduino Nano	2	1	1	1	2	7
Arduino LilyPad	2	2	2	2	2	10
Cumple totalmente	2					
Cumple parcialmente	1					
No cumple	0					

Fuente: Autor

Elección: Para este parámetro se considera como una mejor opción el Arduino LilyPad, ya que este se adapta mejor a los parámetros de arquitectura, ya que este puede ser embebido al pateador, Arduino Nano y Arduino UNO no se adaptan de manera correcta con los parámetros debido a que no cumplen con las características de tamaño y forma necesarias para ser ideales para este proyecto.

3.2.6.1.2. Sensor de fuerza

Para la elección de este parámetro se lo realiza con base en los parámetros planteados en la Tabla 7, en donde se determinará el mejor sensor de fuerza que se adapte con las características y requerimientos del sistema, se plantean 5 requerimientos a tomar en cuenta para 3 opciones de sensores.

Tabla 7 Selección sensor de fuerza

Selección de sensor de fuerza						
Hardware	Requerimientos					Valor total
	SRSH6	SRSH8	SRSH9	SRSH17	SRSH19	
FSR 402	2	2	2	2	1	9
FSR 406	2	2	1	2	1	8
SEN-10293	1	2	1	1	1	6
Cumple totalmente	2					
Cumple parcialmente	1					
No cumple	0					

Fuente: Autor

Elección: Para este parámetro se considera como una mejor opción el sensor FSR 402, ya que este se adapta mejor a los parámetros de arquitectura, ya que abarca un área menor de contacto, lo que resultará más útil para este proyecto, FSR 406 y SEN-10293 no se adaptan de manera correcta con los parámetros debido a que el sensor FSR 406 abarca un área extensa que no se adapta a los golpes, el sensor SEN-10293 no es adecuado debido a que este sensor funciona con base a vibraciones no sobre impactos y su forma no se adapta de manera correcta para embeberlo en el sistema.

3.2.6.1.3. Módulo de comunicación

Para la elección de este parámetro se lo realizará con base en los parámetros planteados a continuación en la Tabla 8, en donde se determinará el mejor módulo de comunicación que se adapte con las características y requerimientos del sistema, se plantean 5 requerimientos a tomar en cuenta para 3 opciones de módulos de comunicación.

Tabla 8 Selección módulo de comunicación

Selección de módulo de comunicación						
Hardware	Requerimientos					Valor total
	SRSH2	SRSH5	SRSH6	SRSH10	SRSH20	
ESP32	2	2	2	1	2	9
HM-10 Bluetooth	1	2	2	1	1	7
NRF24L01	2	2	1	2	2	9
Cumple totalmente	2					
Cumple parcialmente	1					
No cumple	0					

Fuente: Autor

Elección: Para este parámetro se consideran dos opciones con puntaje similar, los módulos ESP 32 y NRF24L01 se adaptan de manera similar al sistema, el parámetro que definirá el módulo de comunicación a usar será la facilidad de compra y disponibilidad. Ambos módulos permiten tener una comunicación a través de Wifi lo que brinda un rango de cobertura aceptable con una frecuencia de 2.4 GHz y un protocolo fiable que cualquier dispositivo inteligente actual puede reconocer e interpretar las señales, el módulo HM-10 Bluetooth, no se adapta de manera correcta ya que la tecnología Bluetooth a pesar de trabajar con la misma frecuencia de 2.4 GHz no trabaja de manera correcta a distancias de más de 10 metros.

3.2.6.2. Software

Para el siguiente punto se va a definir el software adecuado para el proyecto, esto se lo hace con base en los criterios de la Tabla 9 referente a los parámetros de arquitectura, para este paso es fundamental tomar en cuenta los entornos en los que se va a desarrollar el prototipo, considerando el uso del microcontrolador Arduino LilyPad, el cual debe tener una correcta comunicación con la base de datos y la representación visual de los mismos para poder tener una buena interacción con el usuario y entrenador.

3.2.6.2.1. Entorno de programación

Para la elección de este parámetro se va a realizar una comparación de distintos entornos de programación de Arduino dentro de la Tabla 9, en donde se toman en cuenta 4 parámetros relacionados para elegir el entorno que mejor se ajuste al proyecto.

Tabla 9 Selección entorno de programación

Selección de entorno de programación					
Software	Requerimientos				Valor total
	SRS5	SRS11	SRS12	SRS15	
Arduino IDE	2	2	1	1	5
Arduino Web Editor	2	2	1	1	6
Visual Studio Code (Python)	2	2	2	2	8
Cumple totalmente	2				
Cumple parcialmente	1				
No cumple	0				

Fuente: Autor

Elección: De acuerdo con los resultados obtenidos en la Tabla 9 se observa como el entorno Visual Studio Code (Python) es una mejor opción como entorno de programación, ya que

presenta una mejor compatibilidad con librerías necesarias tanto como para el microcontrolador como para la comunicación, además de ser gratuita y tener un amplio soporte por parte de la comunidad.

3.2.6.2.2. Plataforma en la nube

Para este punto se tomaron en cuenta 4 parámetros para definir que plataforma en la nube va a ser la ideal para poder realizar el procesamiento de datos y gestión de estos, se tomaron en cuenta 3 opciones y los resultados se detallan en la Tabla 10.

Tabla 10 Selección plataforma en la nube

Selección de plataforma en la nube					
Hardware	Requerimientos				Valor total
	SRSH5	SRSH11	SRSH13	SRSH14	
AWS	2	2	2	2	8
Google Colab	2	1	1	2	6
Microsoft Azure	1	1	1	1	4
Cumple totalmente	2				
Cumple parcialmente	1				
No cumple	0				

Fuente: Autor

Elección: De acuerdo con los resultados obtenidos en la Tabla 10 se observa que la mejor plataforma en la nube para este proyecto es AWS, esto debido a que tiene más herramientas asociadas a IoT y *machine learning*, lo cual lo convierte en una opción mucho más versátil y que se puede adaptar mejor en caso de trabajar con su capa gratuita o de ser necesario con costos moderados en un escenario de más demanda de recursos.

3.2.6.2.3. Gestor de base de datos en la nube

Para este punto se plantean 3 opciones de gestores de base de datos en la nube, las cuales serán evaluadas y comparadas mediante tres parámetros de requerimientos, cabe señalar que estos gestores se encuentran dentro de la plataforma AWS ya que esta fue seleccionada para poder realizar el procesamiento de datos.

Tabla 11 Selección gestor de base de datos

Selección de gestor de base de datos				
Hardware	Requerimientos			Valor total
	SRS4	SRS5	SRS14	
Amazon RDS	2	2	2	6
Dynamo DB	2	2	2	6
Amazon Aurora	2	1	1	4
Cumple totalmente	2			
Cumple parcialmente	1			
No cumple	0			

Fuente: Autor

Elección: De acuerdo con los resultados obtenidos en la Tabla 11 se obtiene que tanto Amazon RDS y Dynamo DB tienen una ponderación equivalente, ambos gestores se adaptan de manera similar al proyecto, por lo que se realiza una elección con base en el costo que se pudiese generar en el proyecto, Amazon RDS presenta un manejo mucho más eficiente de los datos de sensores y se puede escalar con mayor facilidad, lo cual reduce los costos de manera significativa en caso de necesitar mayor capacidad de guardado de datos en el proyecto.

3.3. Diseño del sistema

Con los componentes de hardware y software del sistema definidos, se procede a establecer las directrices para el diseño del sistema orientado a la medición y evaluación de las patadas en

taekwondo. Este proceso considera los criterios previamente analizados, seleccionados específicamente para facilitar el desarrollo de un sistema que pueda capturar, procesar y analizar datos de fuerza e impacto generados por las técnicas de pateo propias de este arte marcial.

Durante esta fase, se especifican las funciones principales del sistema a través de diagramas de bloques y diagramas de flujo. Estas herramientas ofrecen una visión clara y estructurada de los procesos involucrados, desde la detección del impacto mediante sensores, pasando por el procesamiento de los datos en tiempo real, hasta la transmisión y visualización de los resultados. Este enfoque asegura una integración lógica entre los componentes, permitiendo que el sistema opere de manera eficiente y cumpla con los requerimientos específicos del proyecto.

Además, esta metodología ayuda a identificar posibles desafíos relacionados con la calibración de los sensores, la gestión de datos en tiempo real y la comunicación con los módulos externos, como la interfaz de usuario. Con una planificación detallada, se busca garantizar que el sistema sea capaz de medir la efectividad de las patadas, contribuyendo al entrenamiento y evaluación técnica de los practicantes de taekwondo.

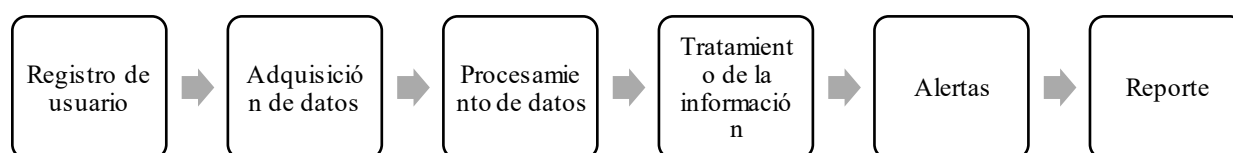
3.3.1. Diagrama de bloques general del sistema

El diseño del sistema de entrenamiento inteligente se organiza en seis fases fundamentales, las cuales abarcan el proceso completo de funcionamiento de este. Cada una de estas fases incluye subprocesos específicos que se relacionan directamente con las funciones principales a cumplir, asegurando un flujo de trabajo ordenado y eficiente.

Para brindar una visión general clara, en la Figura 8 a continuación se presenta un diagrama de bloques que ilustra las seis fases del sistema. Este diagrama ofrece una comprensión visual de

las actividades clave realizadas en cada etapa de la arquitectura del sistema. Las fases abarcan desde la captación de datos iniciales mediante los sensores hasta el análisis, procesamiento y visualización de los resultados obtenidos, proporcionando un marco estructurado que guía todo el proceso.

Figura 8 Diagrama de bloques del sistema del sistema (Autor)



En las secciones siguientes, se detallará cada fase de manera individual, explorando los subprocesos y componentes que intervienen en cada etapa. Este enfoque permitirá comprender cómo cada fase contribuye al objetivo global del sistema, que es medir y analizar parámetros clave de las patadas, como la fuerza y la velocidad, y proporcionar retroalimentación útil para mejorar el rendimiento de los practicantes.

El sistema diseñado para el monitoreo y evaluación de las patadas en taekwondo se organiza en seis bloques principales, cada uno enfocado en una etapa específica del proceso. A continuación, se describen los bloques de manera adaptada:

En el primer bloque, los usuarios tienen la opción de registrarse o iniciar sesión mediante una interfaz gráfica. Para registrarse, deben proporcionar un nombre de usuario y una contraseña personal. Estos datos se almacenan en una base de datos alojada en AWS. Cuando los usuarios inician sesión correctamente, el sistema utiliza su nombre para identificar quién genera datos y sesiones de entrenamiento, asegurando una gestión personalizada de la información.

En el segundo bloque, se lleva a cabo la adquisición de datos mediante sensores de fuerza integrados en las herramientas de entrenamiento, como pateadores. Los sensores registran parámetros clave como la fuerza del impacto en distintos puntos del pateador. En este bloque también actúa el módulo de transmisión de datos para poder asociarlos en la plataforma de desarrollo.

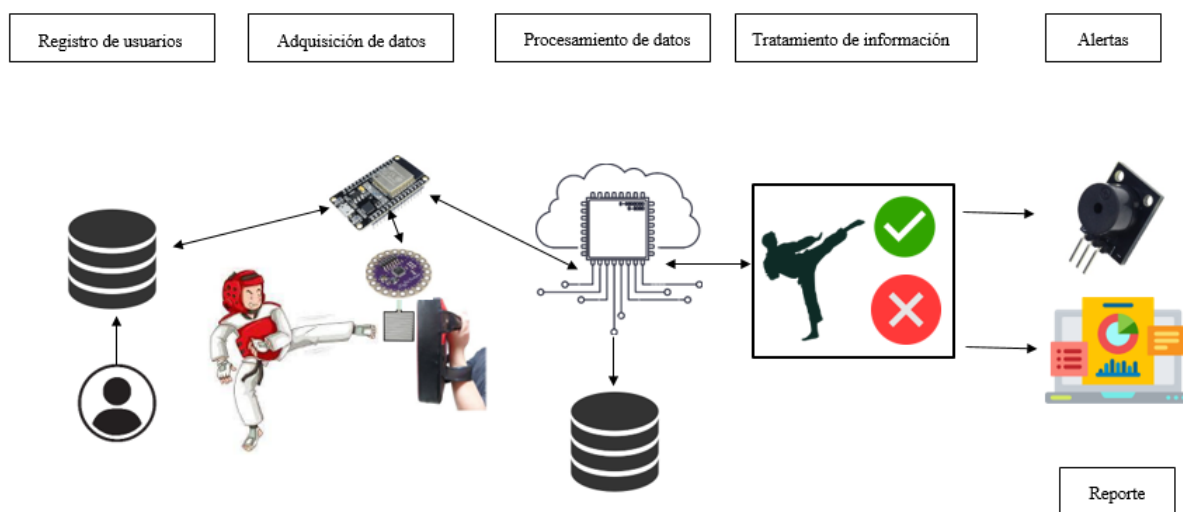
El tercer bloque se centra en el procesamiento de la información capturada. Los datos obtenidos de los sensores se utilizan para determinar los puntos en donde existió una mayor cantidad de fuerza. Con esta información, se genera un mapa de constelaciones el cual se usa para analizar la efectividad de las patadas.

En el cuarto bloque, se trata la información procesada para clasificar las patadas ejecutadas como correctas o incorrectas. Los resultados se visualizan en tiempo real mediante una interfaz gráfica accesible para el usuario y el instructor. Además, el sistema detalla los errores cometidos, indicando el número de la o las patadas incorrectas.

El quinto bloque está dedicado a la generación de alertas. Estas se activan cuando el número de patadas incorrectas supera un número determinado, indicando una sesión inválida al entrenador, para que este tome la decisión correspondiente de terminar o no dicha sesión.

Por último, el sexto bloque se encarga de la generación de reportes detallados. Estos reportes incluyen el total de patadas ejecutadas, el mapa de constelaciones generado de la sesión y una recomendación por parte del sistema.

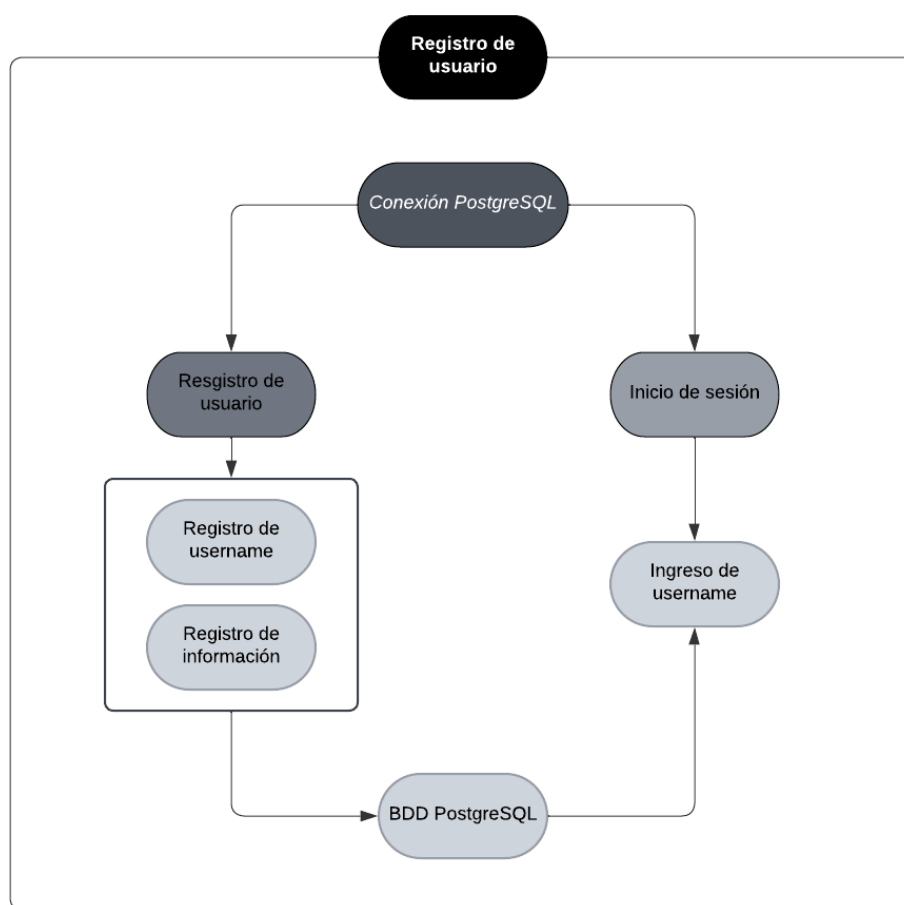
A continuación, en la Figura 9, se muestra la arquitectura del sistema, con base en el diagrama de bloques, el cual cumple con el proceso para poder hacer el uso del sistema.

Figura 9 Arquitectura del sistema (Autor)

3.3.1.1. Registro

Dentro del bloque de registros se tiene la base de datos que será necesaria para poder establecer los usuarios del sistema, los cuales serán registrados y de esta manera tener un seguimiento personalizado de cada uno de ellos, ya que cada sesión que realice será registrada y el resultado será almacenado para que el entrenador pueda realizar un análisis posterior y de esta manera tenga un registro histórico de la evolución que ha tenido el practicante, viendo mejoras o deterioros de las técnicas propuestas. En la figura 10 se observa el proceso lógico para el registro e inicio de sesión de los usuarios que harán uso del sistema.

Figura 10 Diagrama de bloques del registro e inicio de sesión de usuarios (Autor)

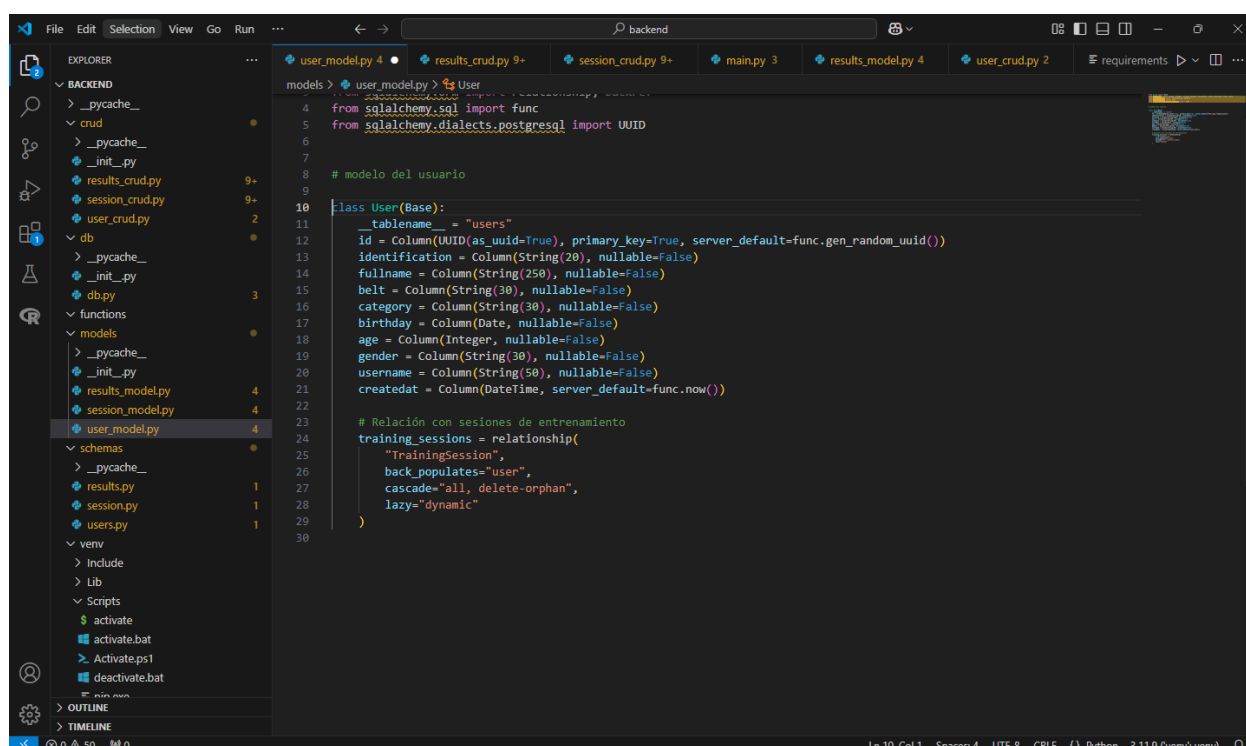


El sistema está desarrollado en Python, utilizando el framework FastAPI para construir una API RESTful eficiente, modular y fácil de escalar. Este framework permite implementar funciones para manejar solicitudes HTTP como GET, POST, PUT y DELETE, cada una diseñada para cumplir con operaciones específicas. Estas funciones están organizadas en rutas que responden a acciones relacionadas con usuarios, sesiones de entrenamiento, resultados y archivos. Además, el sistema incluye validaciones para asegurar que los datos enviados por los clientes sean consistentes con los requerimientos de cada operación.

La arquitectura del sistema está dividida en cuatro componentes principales: models, schemas, CRUD y el archivo principal (main.py).

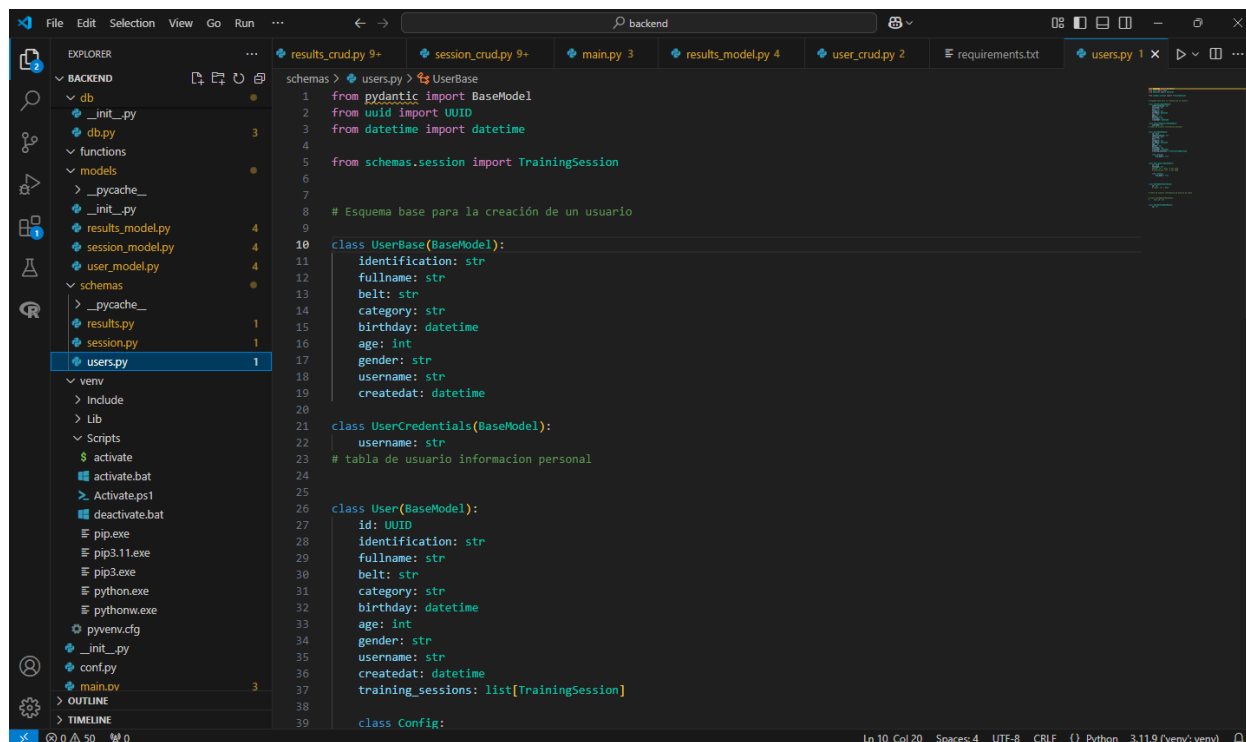
Models: Representan la estructura lógica de las tablas de la base de datos, utilizando SQLAlchemy como ORM (Object Relational Mapper). Cada modelo define las propiedades y relaciones de las entidades del sistema, como usuarios, sesiones de entrenamiento y resultados. Esto permite interactuar con la base de datos de manera declarativa y eficiente. En la Figura 11 se ve un modelo del Model que será usado para los usuarios

Figura 11 Script del user_model (Autor)



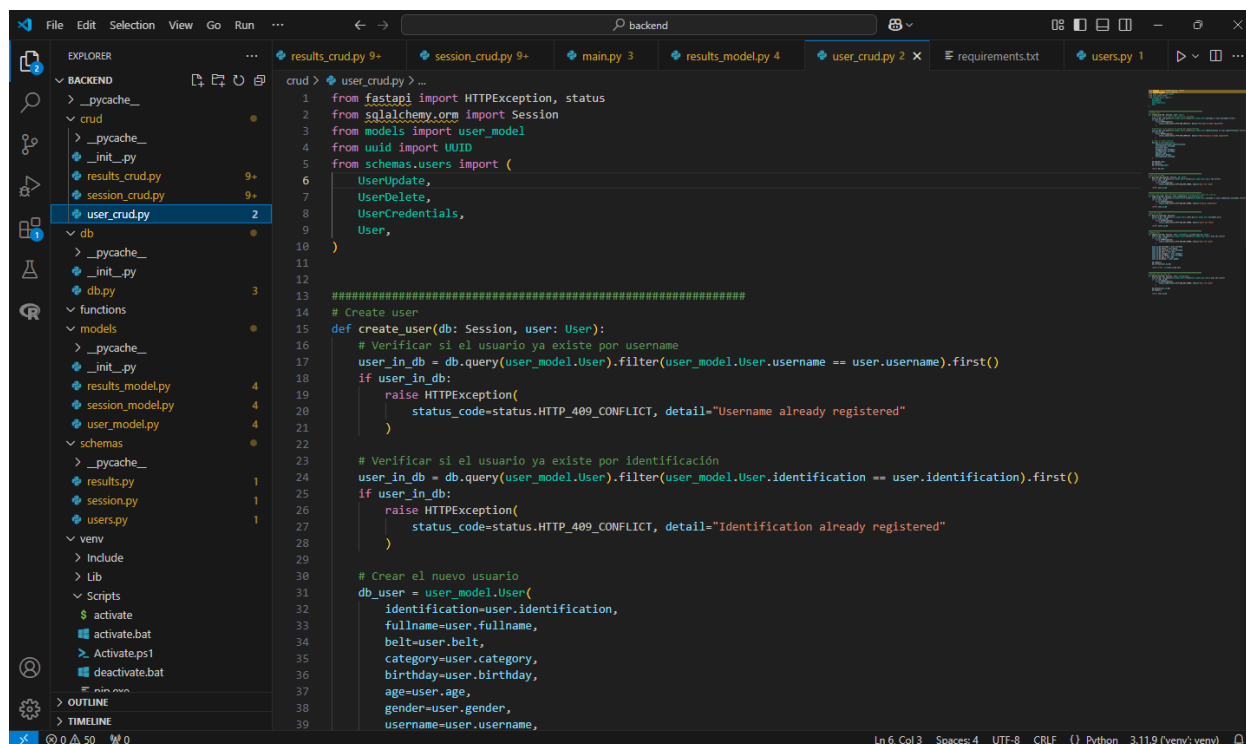
Schemas: Están basados en Pydantic y se utilizan para validar y serializar los datos que entran y salen de la API. Existen diferentes tipos de esquemas, como Base (estructura básica), Create (para datos de entrada al crear nuevos registros), Update (para modificaciones) y Response (para estructurar las respuestas enviadas al cliente). En la Figura 12 se observa un ejemplo de cómo será la estructura de los Schemas, en este caso para el user Schema

Figura 12 Script del scheme user (Autor)



CRUD: Contiene las funciones que implementan las operaciones básicas de interacción con la base de datos, como crear, leer, actualizar y eliminar registros. Estas funciones encapsulan la lógica de negocio, asegurando una separación clara entre la capa de datos y la capa de presentación. En la Figura 13 se visualiza un ejemplo de los scripts CRUD que serán usados dentro del proyecto.

Figura 13 Script de user_crud (Autor)



El archivo `main.py` actúa como el punto de entrada del sistema, configurando la aplicación, registrando los endpoints y definiendo los middlewares necesarios. Uno de los middlewares más importantes es CORS, que permite el acceso desde diferentes orígenes, haciendo que la API sea accesible para clientes web y móviles. El archivo también incluye la inicialización de la base de datos, usando `Base.metadata.create_all()` para garantizar que las tablas necesarias sean creadas al iniciar la aplicación. En la Figura 14 se ve una parte del código del archivo `main`, el cual contiene las funciones y las une para formar el sistema.

Figura 14 Script de `main` (Autor)

```

1 from fastapi import FastAPI, Depends, HTTPException, status, Form, UploadFile, File
2 from sqlalchemy.orm import Session
3 from db.db import engine, get_db, Base
4 from fastapi.middleware.cors import CORSMiddleware
5 from models import User, Session, Results
6 from schemas.results import SessionResultCreate, SessionResult, SessionResultBase, SessionResultDelete, SessionResultUpdate
7 from schemas.session import TrainingSessionBase, TrainingSession, TrainingSessionCreate, TrainingSessionDelete, TrainingSessionUpdate
8 from schemas.users import User, UserUpdate, UserDelete, UserPublic, UserBase
9 from crud import user_crud, session_crud, results_crud
10 import os
11
12 # Initialize database
13 Base.metadata.create_all(bind=engine)
14
15 # Initialize FastAPI app
16 app = FastAPI()
17
18 # CORS middleware
19 app.add_middleware(
20     (function) allow_credentials: Any
21     allow_credentials=True,
22     allow_methods=["*"],
23     allow_headers=["*"],
24 )
25
26 @app.get("/")
27 def get_main():
28     return {"message": "Welcome to the Training Sessions API"}
29
30 # ***** USERS *****
31
32 @app.post("/api/user/register", response_model=User)
33 async def create_user_endpoint(user: UserBase, db: Session = Depends(get_db)):
34     return user_crud.create_user(db=db, user=user)
35
36 @app.get("/api/user/{id}", response_model=User)
37 async def get_user_by_uid_endpoint(id: str, db: Session = Depends(get_db)):
38     return user_crud.get_user_by_uid(db=db, id=id)

```

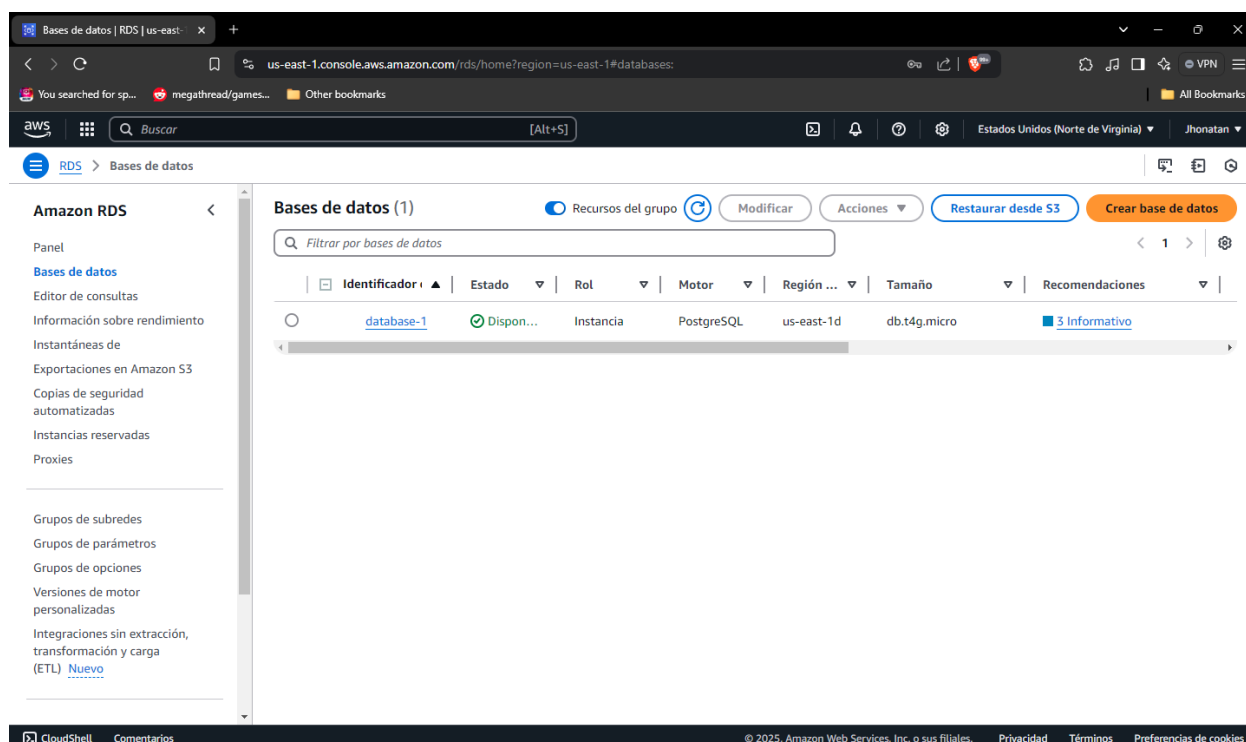
Cuando se recibe una solicitud HTTP, la API la dirige al endpoint correspondiente definido en main.py. Los datos recibidos pasan primero por los schemas para validación. Luego, la lógica CRUD se encarga de procesar la solicitud interactuando con los modelos y la base de datos. La respuesta generada por el CRUD es serializada nuevamente mediante un esquema de respuesta y enviada al cliente en formato JSON. Este flujo asegura que cada operación sea consistente, validada y estructurada correctamente.

Todo el proceso referente al sistema de base de datos se aloja dentro del servicio RDS de AWS, el cual proporciona una solución escalable y administrada para bases de datos relacionales. AWS RDS permite elegir entre varios gestores de bases de datos como MySQL, MariaDB, Oracle y SQL Server, entre otros. En este caso, se opta por utilizar PostgreSQL debido a su robustez, soporte para consultas avanzadas, extensibilidad y manejo eficiente de grandes volúmenes de

datos. PostgreSQL es ideal para aplicaciones que requieren integridad de datos y características avanzadas como índices personalizados y transacciones complejas.

En la figura 15 se observa la interfaz del gestor de base de datos RDS de AWS activo y se visualiza que se encuentra disponible para su funcionamiento.

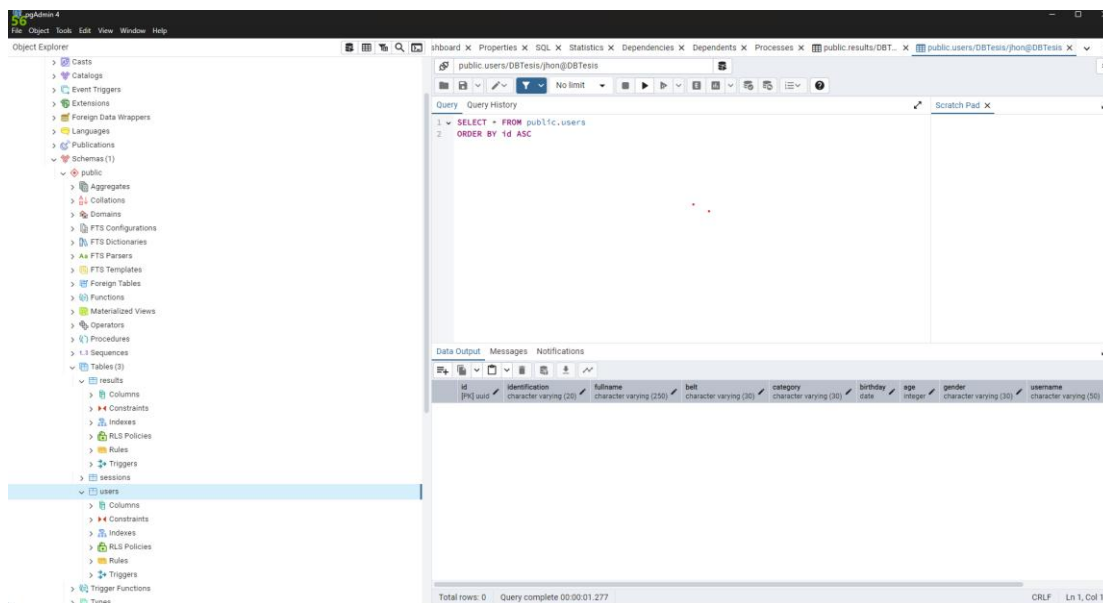
Figura 15 Base de datos relacional creada en RDS (Autor)



El diseño lógico de la base de datos se estructura en varias tablas que representan las entidades principales del sistema, como usuarios, sesiones de entrenamiento, resultados y archivos relacionados. Estas tablas están optimizadas mediante relaciones clave foránea para garantizar la integridad referencial y minimizar la redundancia de datos. Todas las operaciones de lectura y escritura hacia la base de datos se gestionan a través de SQLAlchemy, que abstrae las consultas

SQL y ofrece una interacción programática mediante modelos, tal como se observa en la Figura 16.

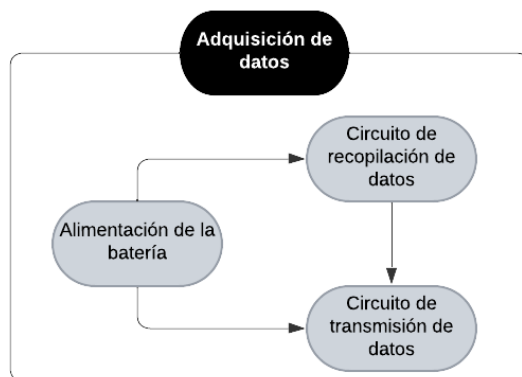
Figura 16 Conexión del back end con la base de datos (Autor)



3.3.1.2. Adquisición de datos

En este bloque se presenta el proceso de la adquisición de datos, este consiste en el circuito de batería y alimentación tanto para el microcontrolador, como para el módulo de comunicación y todos los sensores y periféricos que se involucren, por lo que se propone el diagrama de bloques de la Figura 17.

Figura 17 Diagrama de bloques adquisición de datos (Autor)



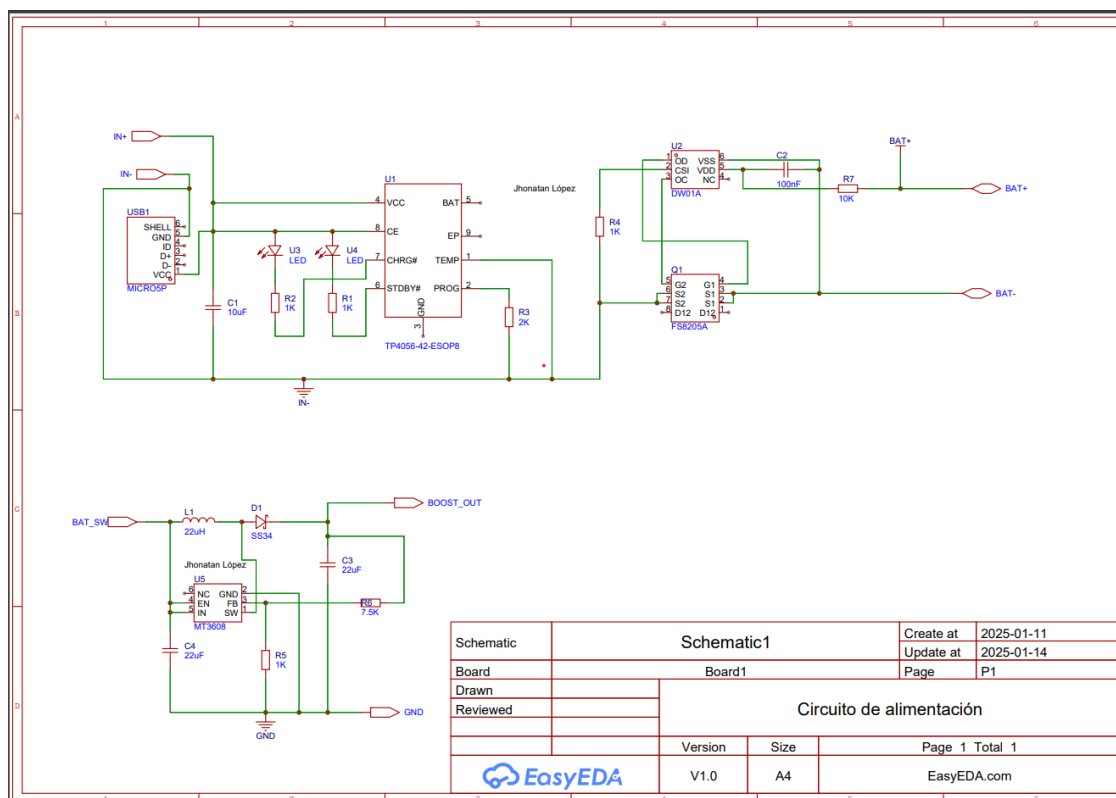
3.3.1.2.1. Circuito de alimentación de batería en esquemático

En este bloque se realiza el diseño del circuito de alimentación de la batería, el cual brindará de la potencia necesaria para alimentar los circuitos de recopilación y transmisión de datos, para esto se realizó un modelo en el que se toma un circuito integrado TP4056, el cual tiene como finalidad cargar y descargar la batería, de manera segura, para evitar que exista un sobre voltaje y también evitar que la batería se descargue a un valor menor que el detallado por el fabricante y así no exista un daño en la misma.

También se consideró un circuito elevador de voltaje, ya que comúnmente las baterías emiten un voltaje de 3.7 voltios, por lo que es necesario incrementar el voltaje a 5 voltios para que ambos circuitos puedan alimentarse y no existan problemas de falta de energía durante las sesiones de entramiento.

En la Figura 18 se observa el circuito esquemático del sistema de alimentación que será usado para el proyecto.

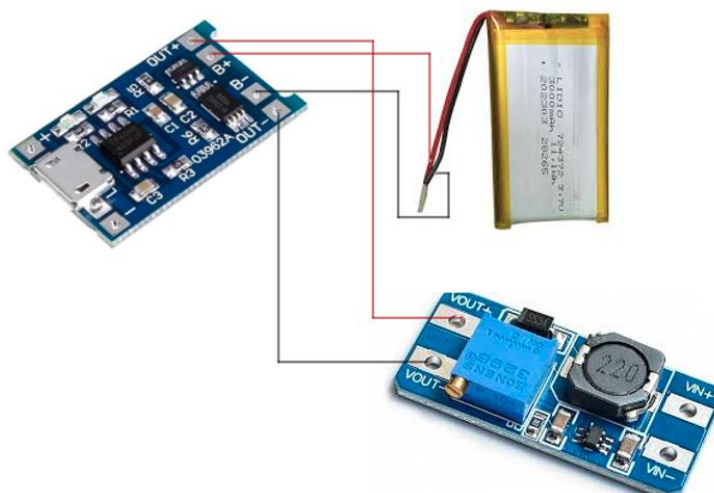
Figura 18 Circuito de alimentación esquemático (Autor)



3.3.1.2.2. Módulos del circuito de alimentación de batería.

Para este caso se hizo uso de dos módulos comerciales que tienen las características de cargador y regulador de salida de voltaje, así como de protección del circuito en caso de sobre descarga y cortocircuito. Los módulos usados son: Módulo de carga TP4056 y Módulo de regulación de voltaje MT3608 representados en la Figura 19.

Figura 19 Circuito de alimentación con módulos de carga y regulación de voltaje (Autor)

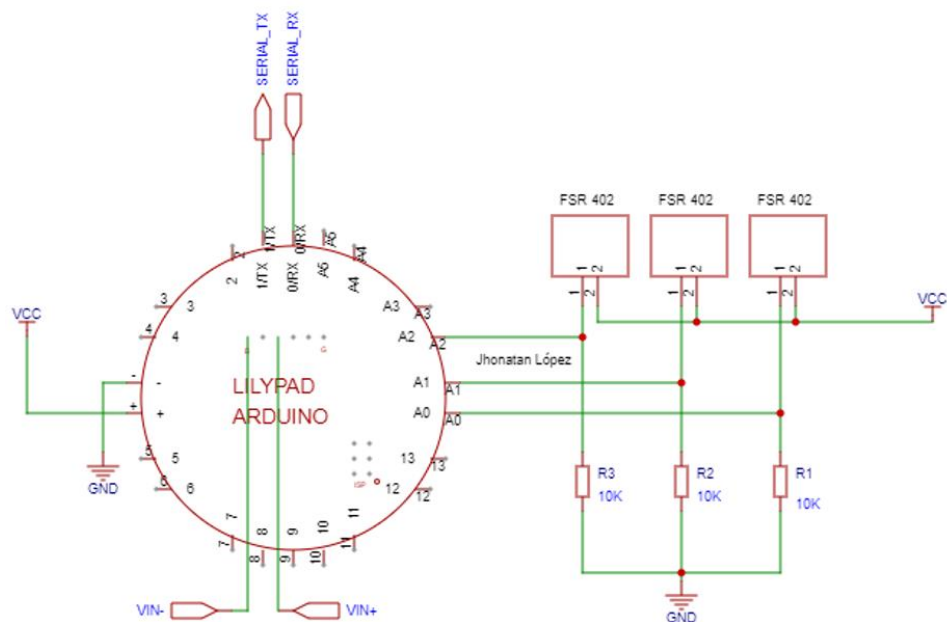


3.3.1.2.3. Circuito de recopilación de datos

Para el diseño de este circuito se tomó como elemento principal el microcontrolador Arduino LilyPad, el cual cuenta con entradas y salidas analógicas, las cuales son necesarias para poder implementar los sensores de fuerza FSR402, estos sensores se implementarán mediante un circuito pulldown, para poder crear una división de voltaje a tierra y de esta manera se obtengan valores de fuerza y que el microcontrolador pueda leerlos y mediante los pines de transmisión y recepción serial comunicarlos hacia el circuito de transmisión de datos.

En la figura 20 se muestra el esquema del circuito con el microcontrolador, sensores y resistencias que generarán la distribución pull down.

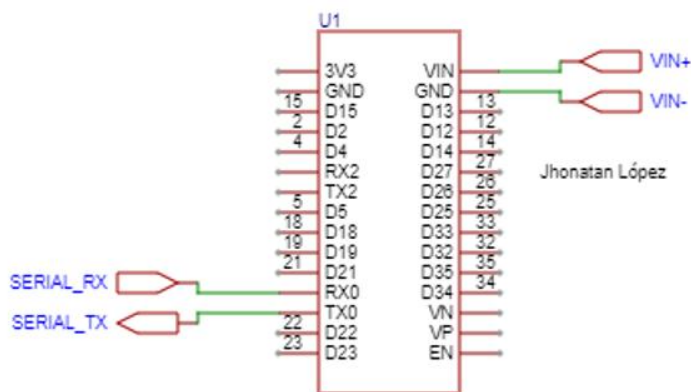
Figura 20 Circuito esquemático de recopilación de datos (Autor)



3.3.1.2.4. Circuito de transmisión de datos

El circuito de transmisión de datos consiste en un módulo ESP32, el cual tendrá una comunicación serial con el Arduino LilyPad para recibir y también tendrá una entrada de voltaje desde el circuito de alimentación, su representación visual se refleja en la Figura 21.

Figura 21 Circuito esquemático de transmisión de datos (Autor)

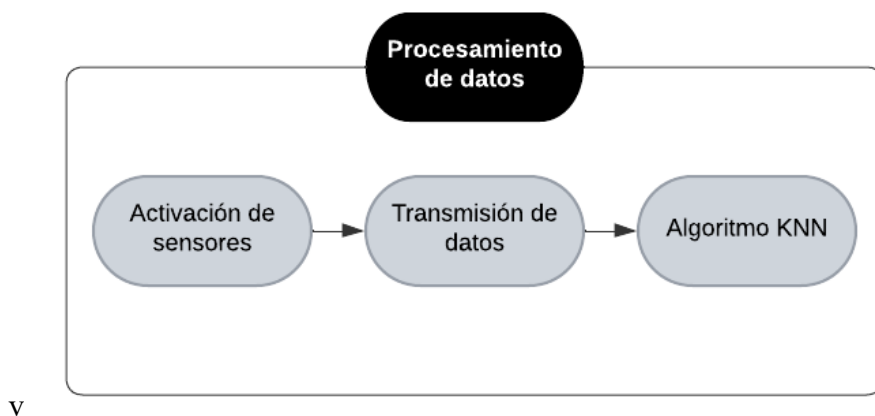


3.3.1.3. Procesamiento de datos

El bloque de procesamiento denota el flujo que tendrán los datos después de ser recibidos por los sensores, como primer paso se tiene la activación de sensores, posterior a esto se transmitirán los datos hacia el módulo ESP32, una vez se reciben los datos en el servidor, se procede a la implementación del algoritmo KNN para que de esta manera se pueda determinar si un golpe o patada es válida y se logre implementar el aprendizaje de la máquina.

La Figura 22 muestra el diagrama de bloques de esta parte del proyecto.

Figura 22 Diagrama de bloques de procesamiento de datos (Autor)

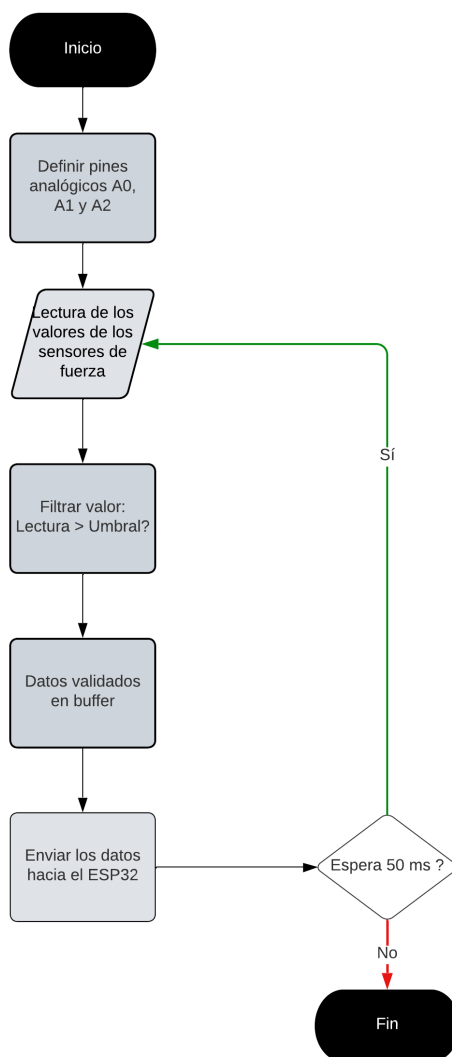


3.3.1.3.1. Activación de sensores

En esta sección se realiza la configuración del microcontrolador para poder hacer el proceso de activación de sensores y estos puedan recibir los datos cuando se ejecute una patada, además, se añaden parámetros para que se calcule el valor máximo de la fuerza de cada sensor y se transmita en una variable que cree el mapa de constelación, esta información se transmitirá en forma de comunicación serial hacia el módulo de transmisión de datos.

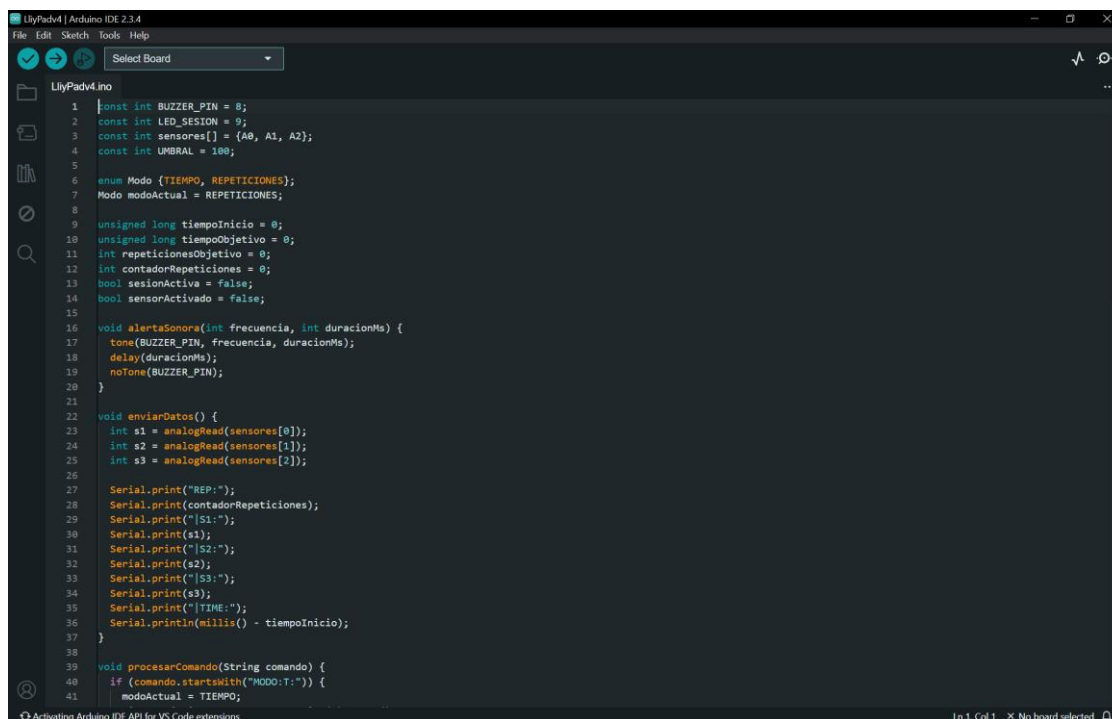
La Figura 23 detalla el diagrama de flujo de cómo se activan los sensores dentro de este bloque, para que pueda recibir los datos y transmitirlos por comunicación serial.

Figura 23 Diagrama de flujo activación de sensores (Autor)



La Figura 24 muestra el código que se ejecuta en el microcontrolador LilyPad para recibir los datos de los sensores de fuerza.

Figura 24 Script de activación de sensores (Autor)



```

1  const int BUZZER_PIN = 8;
2  const int LED_SESSION = 9;
3  const int sensores[] = {A0, A1, A2};
4  const int UMBRAL = 100;
5
6  enum Modo {TIEMPO, REPETICIONES};
7  Modo modoActual = REPETICIONES;
8
9  unsigned long tiempoInicio = 0;
10 unsigned long tiempoObjetivo = 0;
11 int repeticionesObjetivo = 0;
12 int contadorRepeticiones = 0;
13 bool sesionActiva = false;
14 bool sensorActivado = false;
15
16 void alertaSonora(int frecuencia, int duracionMs) {
17   tone(BUZZER_PIN, frecuencia, duracionMs);
18   delay(duracionMs);
19   noTone(BUZZER_PIN);
20 }
21
22 void enviarDatos() {
23   int s1 = analogRead(sensores[0]);
24   int s2 = analogRead(sensores[1]);
25   int s3 = analogRead(sensores[2]);
26
27   Serial.print("REP:");
28   Serial.print(contadorRepeticiones);
29   Serial.print("|S1:");
30   Serial.print(s1);
31   Serial.print("|S2:");
32   Serial.print(s2);
33   Serial.print("|S3:");
34   Serial.print(s3);
35   Serial.print("|TIME:");
36   Serial.println(millis() - tiempoInicio);
37 }
38
39 void procesarComando(String comando) {
40   if (comando.startsWith("MODOS:")) {
41     modoActual = TIEMPO;

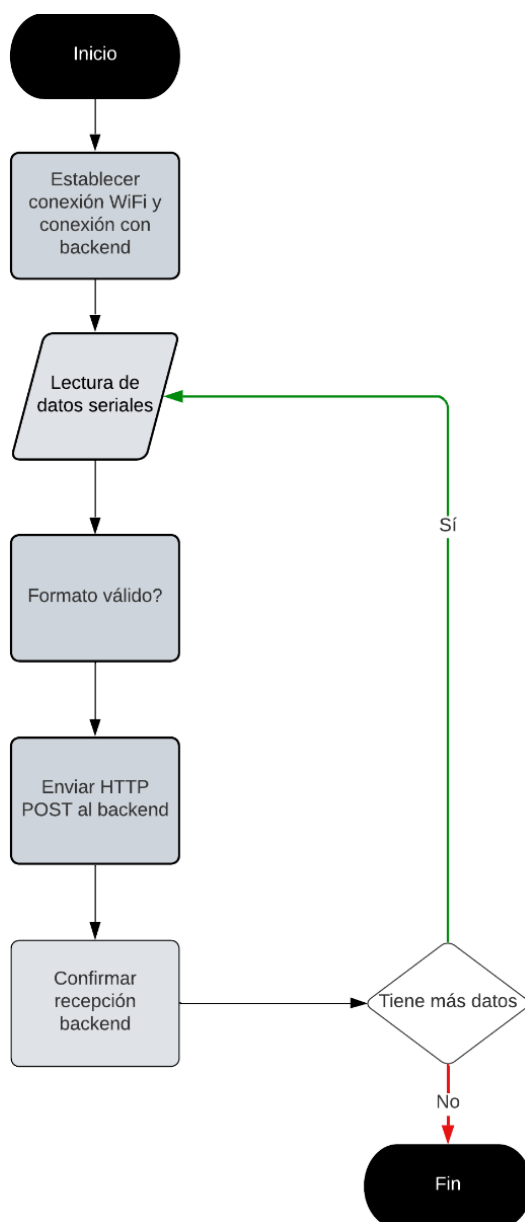
```

3.3.1.3.2. Transmisión de datos

En este punto se considera como los datos se van a transmitir hacia el backend desde el dispositivo, se toma en cuenta que el dispositivo debe tener una conexión WiFi en el lugar en donde se ubique el dispositivo, ya que mediante este protocolo se va a enviar la información hacia la dirección IP 192.168.137.1, puerto 2003, en donde se encuentra alojado el servicio.

El ESP32 recibe los datos seriales, gestiona la conexión WiFi y los envía al backend en Python, donde se almacenan en una base de datos relacional con tablas específicas para patadas individuales, el script tiene un proceso de verificación para la transmisión de los datos, para que no exista una pérdida de información.

La Figura 25 detalla el diagrama de flujo de cómo se transmiten los datos desde el módulo ESP32 hacia el backend.

Figura 25 Diagrama de flujo transmisión de datos (Autor)

La Figura 26 muestra el código que se ejecuta en el módulo de comunicación para conectarse a una red WiFi y transmitir los datos hacia el backend.

Figura 26 Script de transmisión de datos (Autor)

```

scriptesp32.ino
1  #include <WiFi.h>
2  #include <HTTPClient.h>
3  #include <ArduinoJson.h>
4
5  // ===== CONFIGURACIÓN WIFI =====
6  const char* SSID = "MSI 8638";
7  const char* PASSWORD = "nuyt2430";
8  const char* SERVER_URL = "http://192.168.137.1:2003";
9
10 // Configuración de IP estática para el ESP32
11 IPAddress local_IP(192, 168, 137, 100);
12 IPAddress gateway(192, 168, 137, 1);
13 IPAddress subnet(255, 255, 255, 0);
14 IPAddress primaryDNS(8, 8, 8, 8);
15 IPAddress secondaryDNS(8, 8, 4, 4);
16
17 // ===== CONFIGURACIÓN DE HARDWARE =====
18 #define RX_PIN 16 // Conectar a TX del Lilypad
19 #define TX_PIN 17 // Conectar a RX del Lilypad
20 #define LED_WIFI 13
21
22 // Serial para el Lilypad
23 HardwareSerial lilypadSerial(2);
24
25 // ===== VARIABLES GLOBALES =====
26 unsigned long lastUpdate = 0;
27 const int updateInterval = 1000; // Consultar comandos cada 5 segundos
28
29 // Lista de comandos válidos
30 const String comandosValidos[] = {"INICIAR", "DETENER", "MOD0"};
31
32 // ===== FUNCIONES =====
33
34 void conectarWifi() {

```

Para este punto se tomó en cuenta que el ESP32 debe enviar comandos válidos para que el Arduino Lilypad pueda recibir los parámetros y comenzar la sesión, por lo que se establece que únicamente pueden ser procesados una lista de comandos válidos, los cuales serán enviados por el backend y a su vez se almacenarán en una tabla dedicada.

En la figura 27 se visualiza la declaración de los comandos válidos que tiene el script del ESP32 para poder hacer una correcta gestión de la información que recibe.

Figura 27 Lista de comandos válidos ESP32 (Autor)

```

// Lista de comandos válidos
const String comandosValidos[] = {"INICIAR", "DETENER", "MOD0"};

```

En la figura 28 se visualiza la función de crear comando, la cual almacena el comando en la base de datos y la asocia a la sesión activa

Figura 28 Función de creación de comandos (Autor)

```
def create_command(db: Session, command: CommandCreate):
    """
    Crea un comando y lo asocia a la sesión activa.
    Se busca la sesión activa (status == "in_progress") y se asigna su ID.
    """
    # Buscar la sesión activa (se asume que solo hay una sesión activa por usuario o globalmente)
    session_active = db.query(session_model.TrainingSession).filter(
        session_model.TrainingSession.status == "in_progress"
    ).first()

    if not session_active:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail="No active session found. Please start a session first."
        )

    db_command = Command(
        id=uuid.uuid4(), # Genera un nuevo UUID
        content=command.content,
        session_id=session_active.id, # Se asocia automáticamente a la sesión activa
        executed=False
    )
    db.add(db_command)
    db.commit()
    db.refresh(db_command)
    return db_command
```

Una vez que el comando sea enviado desde el backend se guardará en la base de datos, el ESP32 hará una petición constante para buscar los comandos que estén pendientes mediante una función GET, en donde hará una solicitud a la base de datos para detectar el comando que no haya sido ejecutado y proceder con el mismo, una vez que un comando sea recibido y procesado por el ESP32, este enviará mediante una función PUT una señal en donde se marcará como ejecutado el comando.

En la Figura 29 se muestra la función obtenerComandosPendientes en el ESP32 para que pueda recibir la información desde el backend.

Figura 29 Función para obtener comandos pendientes ESP32 (Autor)

```

void obtenerComandosPendientes() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = String(SERVER_URL) + "/api/comandos";
        http.begin(url);
        int httpCode = http.GET();

        if (httpCode == HTTP_CODE_OK) {
            String payload = http.getString();
            Serial.println("Comandos pendientes: " + payload);

            DynamicJsonDocument doc(1024);
            DeserializationError error = deserializeJson(doc, payload);
            if (!error) {
                JsonArray commandsArray = doc.as<JsonArray>();
                for (JsonObject cmd : commandsArray) {
                    String commandId = cmd["id"].as<String>();
                    String content = cmd["content"].as<String>();

                    if (esComandoValido(content)) {
                        enviarComandoLilypad(content);
                        marcarComandoComoEjecutado(commandId);
                    } else {
                        Serial.println("Comando inválido recibido: " + content);
                    }
                }
            } else {
                Serial.print("Error parseando JSON: ");
                Serial.println(error.c_str());
            }
        } else {
            Serial.printf("Error en GET /api/comandos: %d\n", httpCode);
        }
        http.end();
    } else {
        Serial.println("WiFi desconectado, no se pueden obtener comandos.");
    }
}

```

En la Figura 30 se muestra la función `marcarComandoComoEjecutado` la cual permitirá que el ESP32 le indique al backend cuando un comando ya fue ejecutado.

Figura 30 Función para marcar comandos ejecutados ESP32 (Autor)


```

void marcarComandoComoEjecutado(String commandId) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = String(SERVER_URL) + "/api/comandos/" + commandId;
        http.begin(url);
        http.addHeader("Content-Type", "application/json");

        int httpCode = http.PUT("{}");
        if (httpCode == HTTP_CODE_OK) {
            Serial.println("Comando marcado como ejecutado: " + commandId);
        } else {
            Serial.printf("Error en PUT /api/comandos/%s: %d\n", commandId.c_str(), httpCode);
        }
        http.end();
    } else {
        Serial.println("WiFi desconectado, no se puede marcar comando.");
    }
}
}

```

En la Figura 31 se ven las funciones del backend para poder realizar las consultas del ESP32 y para marcar el comando como ejecutado en la base de datos.

Figura 31 Funciones para obtener y marcar comandos en el backend (Autor)

```

@app.get("/api/comandos", response_model=List[Command])
async def obtener_comandos_pendientes(db: Session = Depends(get_db)):
    """
    Retorna la lista de comandos pendientes (aquellos con executed == False).
    """
    try:
        return commands_crud.get_pending_commands(db=db)
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
            detail=f"Error obteniendo comandos: {str(e)}"
        )

@app.put("/api/comandos/{command_id}", response_model=Command)
async def marcar_comando_completado(command_id: UUID, db: Session = Depends(get_db)):
    """
    Marca un comando como ejecutado.
    """
    try:
        updated_command = commands_crud.mark_command_completed(db=db, command_id=command_id)
        if not updated_command:
            raise HTTPException(status_code=404, detail="Command not found")
        return updated_command
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
            detail=f"Error actualizando comando: {str(e)}"
        )

```

A partir de que el ESP32 comunique al Lilypad que la sesión acaba de iniciar se tiene que enviar las patadas desde el circuito de recepción de datos, hacia el de transmisión y posterior al

backend para que pueda guardar las patadas de esa sesión en la base de datos, por lo que se debe tener una tabla que permita el guardado de esta información y así también una función en el ESP32 que permita realizar esta acción.

En la Figura 32 se visualiza como el Lilypad envía por comunicación serial los datos de cada patada hacia el ESP32 mediante la función enviarDatos.

Figura 32 Función para enviar datos Lilypad (Autor)

```
void enviarDatos() {  
    int s1 = analogRead(sensores[0]);  
    int s2 = analogRead(sensores[1]);  
    int s3 = analogRead(sensores[2]);  
  
    Serial.print("REP:");  
    Serial.print(contadorRepeticiones);  
    Serial.print("|S1:");  
    Serial.print(s1);  
    Serial.print("|S2:");  
    Serial.print(s2);  
    Serial.print("|S3:");  
    Serial.print(s3);  
    Serial.print("|TIME:");  
    Serial.println(millis() - tiempoInicio);  
}
```

En la Figura 33 se aprecia la función declarada en el ESP32 leerDatosLilypad para poder recibir de manera clara y organizada las patadas de la sesión activa.

Figura 33 Función para recibir datos desde Lilypad hacia ESP32 (Autor)

```

void leerDatosLilypad() {
  if (lilypadSerial.available()) {
    String datos = lilypadSerial.readStringUntil('\n'); // Leer hasta el salto de línea
    datos.trim();
    Serial.println("Datos recibidos del Lilypad: [" + datos + "]");
    // Si el mensaje indica que la sesión ha finalizado, procesarlo como finalización
    if (datos.startsWith("SESION_COMPLETADA")) {
      finalizarSesion(datos);
    }
    else if (datos.startsWith("SESION_DETENIDA")) {
      // Llamamos a cancelarSesion() o stopSesion() según tu lógica
      // Si "DETENIDA" implica "canceled":
      cancelarSesion(datos);
    }
    else {
      // Caso normal: procesar como datos de patada
      enviarDatosAlBackend(datos);
    }
  }
}

```

En la Figura 34 se puede ver como el ESP32 realiza el envío de cada patada verificando los parámetros que debe contener cada una mediante una función POST hacia el backend.

Figura 34 Función para enviar datos al backend desde ESP32 (Autor)

```

void enviarDatosAlBackend(String datos) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String url = String(SERVER_URL) + "/api/kicks";
    http.begin(url);
    http.addHeader("Content-Type", "application/json");

    // Variables para extraer los valores del string recibido
    int rep, s1, s2, s3, time;
    int matched = sscanf(datos.c_str(), "REP:%d|S1:%d|S2:%d|S3:%d|TIME:%d", &rep, &s1, &s2, &s3, &time);

    if (matched == 5) {
      // Crear el JSON que espera el backend para crear una patada.
      // NOTA: No se envía session_id, ya que el backend lo asociará automáticamente.
      DynamicJsonDocument doc(256);
      doc["rep"] = rep;
      doc["s1"] = s1;
      doc["s2"] = s2;
      doc["s3"] = s3;
      doc["time"] = time;

      String jsonPayload;
      serializeJson(doc, jsonPayload);
      Serial.println("JSON a enviar: " + jsonPayload);

      int httpCode = http.POST(jsonPayload);
      if (httpCode == HTTP_CODE_OK) {
        Serial.println("Datos enviados correctamente al backend.");
      } else {
        Serial.printf("Error enviando datos al backend: %d\n", httpCode);
      }
    } else {
      Serial.println("Error parseando la cadena recibida del Lilypad.");
    }
    http.end();
  } else {
    Serial.println("WiFi desconectado, no se pueden enviar datos.");
  }
}

```

En la Figura 35 se observa la función del backend para recibir las patadas del ESP32 y como se guardan en la base de datos, asociando a la sesión activa

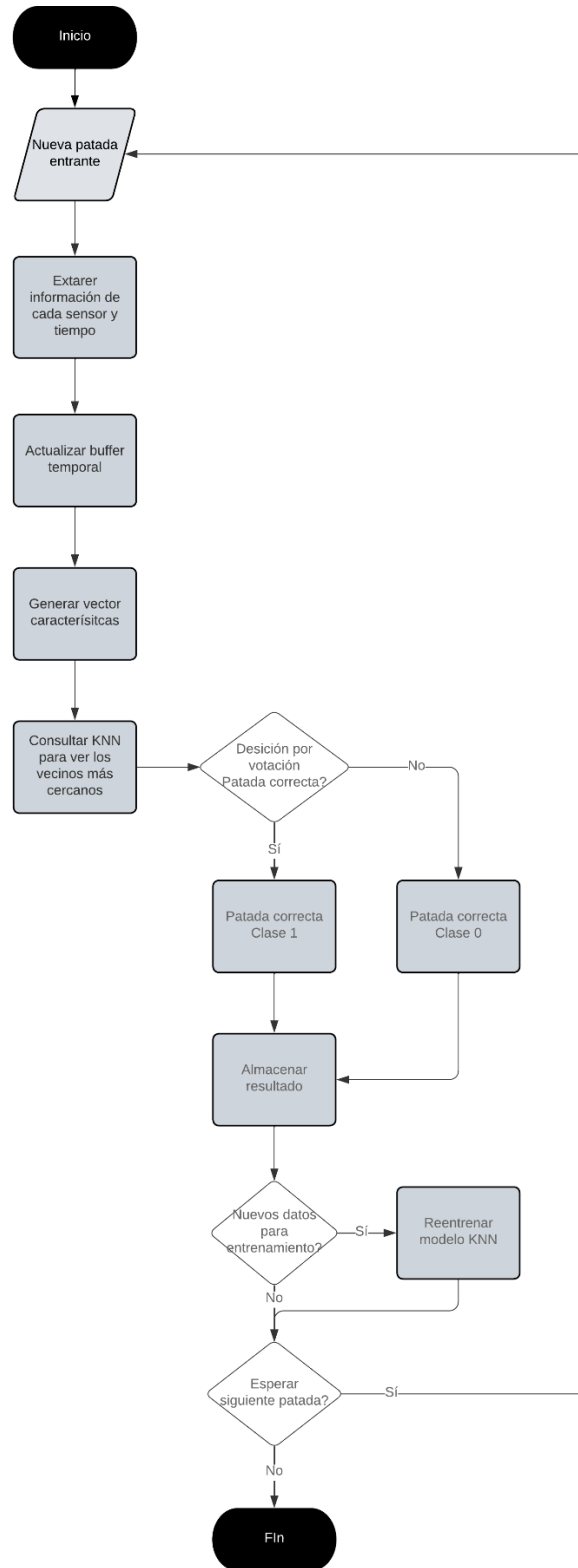
Figura 35 Función del backend para guardar las patadas en la base de datos (Autor)

```
@app.post("/api/kicks", response_model=Kick)
async def create_kick_endpoint(kick: KickCreate, db: Session = Depends(get_db)):
    """
    Endpoint para que el ESP32 envíe los datos de una patada.
    Se espera un JSON con el siguiente formato:
    {
        "rep": 1,
        "s1": 50.0,
        "s2": 20.0,
        "s3": 100.0,
        "time": 3500.0
    }
    """
    try:
        print("Datos recibidos:", kick.dict()) # <-- Depuración
        new_kick = kicks_crud.create_kick(db, kick)
        return new_kick
    except HTTPException as e:
        raise e
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
            detail=f"Error creating kick: {str(e)}"
        )
```

3.3.1.3.3. Algoritmo KNN

En la Figura 36 se describe el diagrama de flujo que seguirá la información a medida que se reciban los datos, para obtener un resultado positivo o negativo.

Figura 36 Diagrama de flujo algoritmo KNN (Autor)



El algoritmo KNN implementado analiza cada patada mediante tres etapas clave:

- a) Captura y contextualización, donde los valores de los sensores (S1, S2, S3) se combinan con metadatos temporales (tiempo absoluto y orden en la secuencia) y se almacenan en un buffer que mantiene las últimas N patadas;
- b) Generación de características, que integra no solo los valores instantáneos de los sensores, sino también estadísticas históricas (promedios móviles y máximos de la ventana temporal) para capturar patrones dinámicos.;
- c) Se calcula la distancia euclidiana entre el vector de características actual y el dataset etiquetado, seleccionando los k vecinos más cercanos (ej: $k = 3$) para una decisión por votación mayoritaria (ej: ≥ 2 votos para "correcta").

El sistema mejora su precisión mediante retroalimentación activa: cada nueva patada etiquetada manualmente (*correcta/incorrecta*) se añade al dataset de entrenamiento, actualizando el modelo KNN sin reiniciar el servicio. Este enfoque considera el orden temporal de las patadas para identificar secuencias válidas (ej: combinaciones de técnicas), usando el endpoint /entrenar para incorporar datos. La integración con el flujo de entrenamiento garantiza que el algoritmo evolucione con el usuario, optimizando la detección de patrones específicos de su estilo de Taekwondo, como la distribución de fuerza en patadas laterales o frontales.

En la Figura 37 se visualiza el proceso del algoritmo KNN impuesto en una función de Python, en donde se recibirán los datos enviados por el sistema electrónico.

Figura 37 algoritmo KNN implementado en el backend (Autor)

```

97 #####
98 # KNN algorithm
99 def process_all_unprocessed_kicks(db: Session):
100     """
101     Procesa todas las patadas no procesadas (Kick.processed == False),
102     sin filtrar por sesión. El modelo KNN se entrena usando solo s1, s2 y s3,
103     ignorando time para el análisis, pero time se almacena en el resultado como referencia.
104     Además, si la predicción del KNN no es "correct", se verifica manualmente que al menos
105     dos de los tres sensores tengan valores  $\geq 250$  para marcar la patada como válida.
106     """
107     # Obtener todas las patadas no procesadas
108     unprocessed_kicks = db.query(Kick).filter(Kick.processed == False).all()
109     if not unprocessed_kicks:
110         return {"message": "No unprocessed kicks found."}
111
112     # Obtener datos históricos de KickResult para entrenar el modelo KNN
113     historical_kicks = db.query(KickResult).all()
114     if not historical_kicks:
115         # Dataset de ejemplo si no hay datos históricos
116         historical_kicks = [
117             {"s1": 200, "s2": 150, "s3": 180, "time": 500, "label": "correct"},
118             {"s1": 100, "s2": 80, "s3": 90, "time": 600, "label": "incorrect"},
119             {"s1": 250, "s2": 220, "s3": 210, "time": 550, "label": "correct"},
120             {"s1": 90, "s2": 70, "s3": 85, "time": 650, "label": "incorrect"},
121         ]
122     # Usamos el dataset de ejemplo en forma de diccionario
123     X_train = np.array([[hk["s1"], hk["s2"], hk["s3"]] for hk in historical_kicks])
124     y_train = np.array([hk["label"] for hk in historical_kicks])
125 else:
126     # Suponemos que historical_kicks son objetos con atributos
127     X_train = np.array([hk.s1, hk.s2, hk.s3] for hk in historical_kicks)
128     y_train = np.array([hk.classification for hk in historical_kicks])
129
130     if len(X_train) < 1:
131         return {"message": "Not enough data to train KNN."}
132
133     # Ajustar n_neighbors según el número de muestras
134     n_neighbors = min(3, len(X_train))

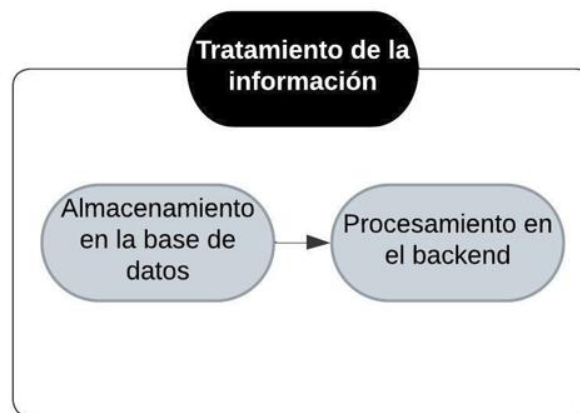
```

3.3.1.4. Tratamiento de información

En el bloque de tratamiento de la información se hará el almacenamiento de las variables de las patadas correctas obtenidas y el procesamiento para poder integrarlas en cada una de las sesiones de entrenamiento que correspondan y de esta manera se pueda tener un registro detallado por usuario y sesión de entrenamiento ejecutada.

En la Figura 38 se observan los bloques que tendrá este proceso para lograr cumplir el tratamiento de la información del sistema.

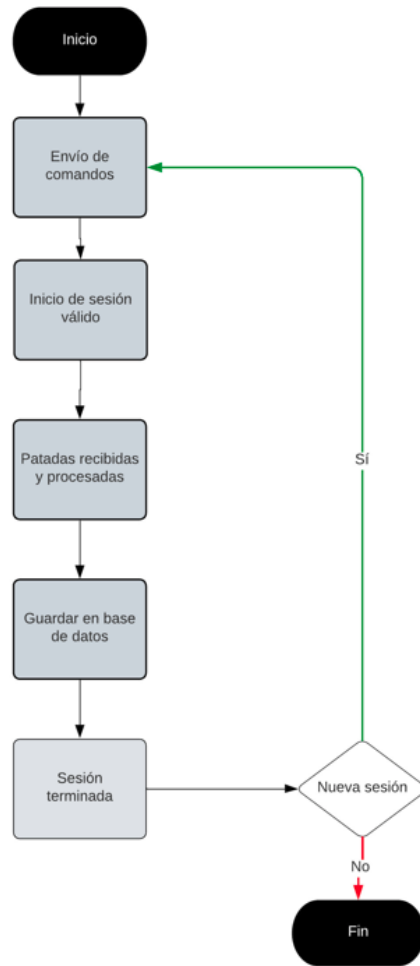
Figura 38 Diagrama de bloques del tratamiento de la información (Autor)



3.3.1.4.1. Almacenamiento de patadas en la base de datos

En la figura 39 se observa el diagrama de flujo para guardar las patadas procesadas en la base de datos.

Figura 39 Diagrama de flujo para el almacenamiento en la base de datos (Autor)



Una vez que se capturan las patadas mediante los sensores del prototipo, cada patada se almacena en la base de datos en una tabla específica, en este caso, "kicks" junto con su información asociada: valores de fuerza de cada sensor, un número de repetición y un timestamp que indica el momento en que se registró. Cada patada se asocia a una sesión de entrenamiento mediante un identificador (`session_id`), y esta sesión, a su vez, se vincula a un usuario. De esta forma, se puede rastrear qué patadas pertenecen a qué sesión y, a su vez, a qué usuario, permitiendo así una gestión organizada de los datos.

Después de que las patadas se almacenan, se ejecuta un proceso de análisis que utiliza un algoritmo de clasificación en este caso KNN, proceso que se explicó en la sección 3.3.1.3.3 para evaluar si cada patada es correcta o no. El algoritmo toma como entrada los valores de fuerza de los sensores (ignorando el tiempo, que se guarda solo como referencia) y genera una predicción que determina la validez de la patada. Una vez procesada, cada patada se actualiza en la base de datos para indicar que ya fue evaluada, y se crea un registro en otra tabla en este caso "kick_results", que almacena el resultado del análisis junto con datos adicionales como la confianza del modelo y el promedio de fuerza.

Finalmente, la información de las patadas procesadas se utiliza para calcular resultados globales de la sesión. El sistema agrega los registros de patadas (los resultados individuales) asociados a una misma sesión para determinar el total de patadas, la cantidad de patadas correctas y el porcentaje de acierto, entre otros parámetros.

En la Figura 39 se visualiza el esquema para la creación de los resultados de cada patada en la base de datos.

Figura 40 Schema para la creación de “kick result” (Autor)

```

class KickResultBase(BaseModel):
    kick_id: str
    session_id: str
    s1: int
    s2: int
    s3: int
    time: Optional[int] = None # Solo referencia, NO se usa en KNN
    classification: str
    confidence: Optional[float] = 0.0
    average_force: float

class KickResultCreate(KickResultBase):
    pass

class KickResultUpdate(BaseModel):
    classification: Optional[str]
    confidence: Optional[float]
    average_force: Optional[float]

class KickResultResponse(KickResultBase):
    id: str
    created_at: datetime

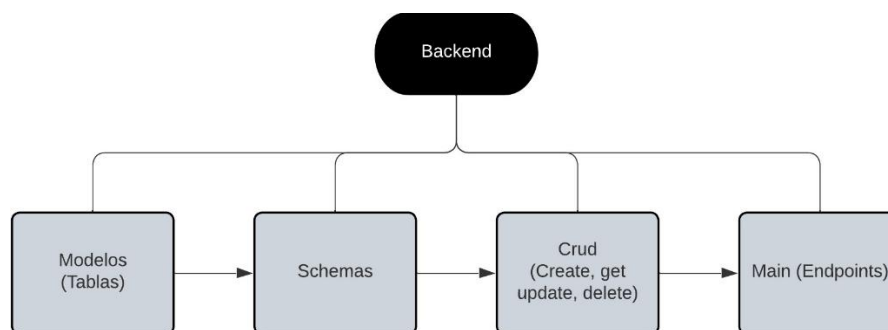
class Config:
    orm_mode = True

```

3.3.1.4.2. Procesamiento en el backend

En la figura 41 se observa la estructura lógica del backend para el procesamiento, control y correcto funcionamiento del proyecto.

Figura 41 Estructura lógica del procesamiento del backend (Autor)



El procesamiento del backend comienza con la configuración del archivo principal (main.py), donde se inicializan las dependencias necesarias, como los frameworks y librerías para la gestión del servidor, la base de datos y la API. Aquí se define la instancia de la aplicación

utilizando FastAPI y se configuran las rutas principales que manejarán las solicitudes HTTP. Además, se establece la conexión con la base de datos mediante SQLAlchemy o un motor de bases de datos similar, asegurando que el sistema pueda interactuar con los datos de manera eficiente.

En la Figura 42 se observa el archivo `main.py` el cual contiene todas las funciones para poder integrar el sistema con todos los pasos previos y se importan las librerías necesarias.

Figura 42 Archivo `main.py` (Autor)

```

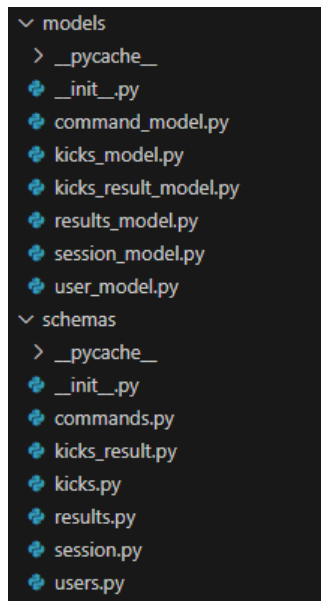
main.py > get_main
1  from datetime import datetime
2  from fastapi import FastAPI, Depends, HTTPException, status, APIRouter
3  from sqlalchemy.orm import Session
4  from pydantic import BaseModel
5  from db.db import engine, get_db, Base
6  from uuid import UUID
7  from fastapi.middleware.cors import CORSMiddleware
8  from models import user_model, session_model, results_model, command_model, kicks_result_model
9  from schemas.results import ResultsCreate, ResultsResponse, ResultsUpdate
10 from schemas.session import TrainingSessionBase, TrainingSession, TrainingSessionCreate, TrainingSessionDelete, TrainingSessionUpdate
11 from schemas.users import User, UserUpdate, UserDelete, UserPublic, UserBase
12 from schemas.commands import CommandCreate, Command
13 from schemas.kicks import Kick, KickCreate, KickBase
14 from schemas.kicks_result import KickResultUpdate, KickResultBase, KickResultCreate, KickResultResponse
15 from schemas.commands import CommandCreate, Command
16 from crud import kicks_result_crud, user_crud, session_crud, results_crud, commands_crud, kicks_crud
17 from crud.kicks_crud import process_all_unprocessed_kicks
18 from crud.results_crud import generate_session_results
19 import os
20 from typing import Optional, List, Dict
21
22
23 # Initialize database
24 Base.metadata.create_all(bind=engine)
25
26 # Initialize FastAPI app
27 app = FastAPI()
28
29 @app.get("/")
30 def get_main():
31     return {"message": "Welcome to the Training Sessions API"}
32
33 # ***** USERS *****

```

La gestión de la base de datos se lleva a cabo a través de modelos que definen la estructura de las tablas y schemas que validan los datos entrantes y salientes. A partir de estos modelos, se implementan los controladores y servicios que gestionan las operaciones del sistema, como el registro de nuevas sesiones de entrenamiento, la consulta de datos históricos y el almacenamiento de resultados después del procesamiento con el algoritmo KNN.

En la figura 43 se puede ver la estructura del backend de los modelos y esquemas que contiene el sistema

Figura 43 Modelos y esquemas del proyecto (Autor)



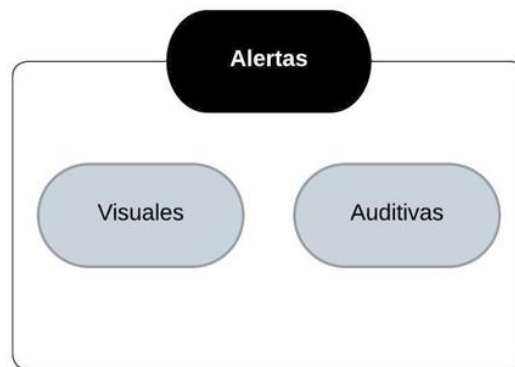
Estos controladores se exponen a través de endpoints, permitiendo que la aplicación frontend o cualquier cliente externo pueda interactuar con el sistema de manera segura y eficiente.

3.3.1.5. Alertas

El sistema integra alertas visuales y auditivas para proveer retroalimentación directa durante el entrenamiento. Las alertas visuales se gestionan mediante tres LEDs que muestran los estados clave del prototipo, mientras que las alertas auditivas usan un buzzer para comunicar eventos críticos de la sesión de entrenamiento. Ambos tipos de componentes permiten que el usuario tenga una mejor retroalimentación para que sea más amigable y permita ver de manera más fácil el estado del prototipo o de la sesión.

En la figura 44 se observan los tipos de alertas que tiene el prototipo para poder avisar al usuario el estado.

Figura 44 Bloque de tipos de alertas (Autor)

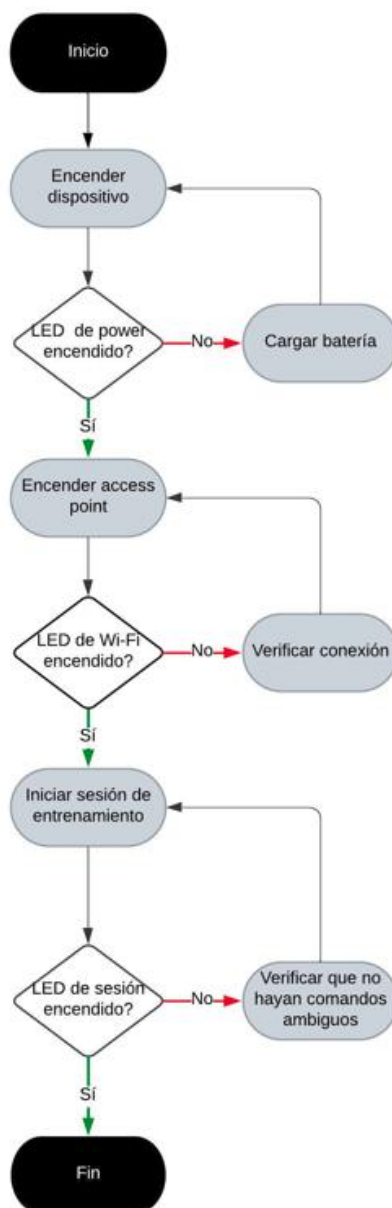


3.3.1.5.1. Alertas visuales

Para el prototipo existen tres alertas visuales que tendrán distintas funciones, una para indicar el estado general del dispositivo, es decir, si este se encuentra encendido y con batería suficiente para poder funcionar, uno conectado al ESP32 para indicar si la conexión mediante Wi-Fi se encuentra conectada y lista para recibir comandos desde el backend, y un led para indicar el estado activo de la sesión.

En la Figura 45 se puede ver el propósito para cada una de las alertas visuales mediante un diagrama de flujo.

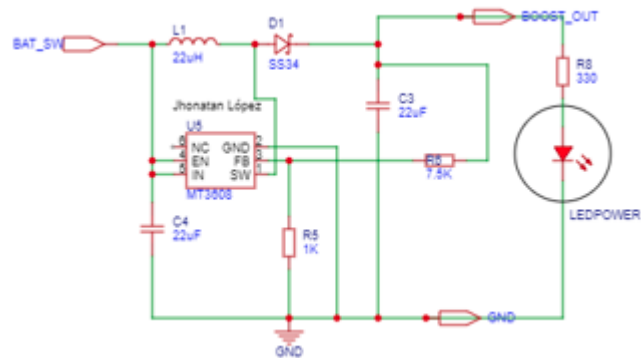
Figura 45 Diagrama de flujo alertas visuales (Autor)



Para el primer caso es necesario conectar un led directamente desde la fuente de alimentación para poder verificar que el dispositivo está obteniendo una alimentación correcta de 5 voltios y la batería no se encuentre descargada.

En la Figura 46 se aprecia el circuito lógico para el funcionamiento de este LED.

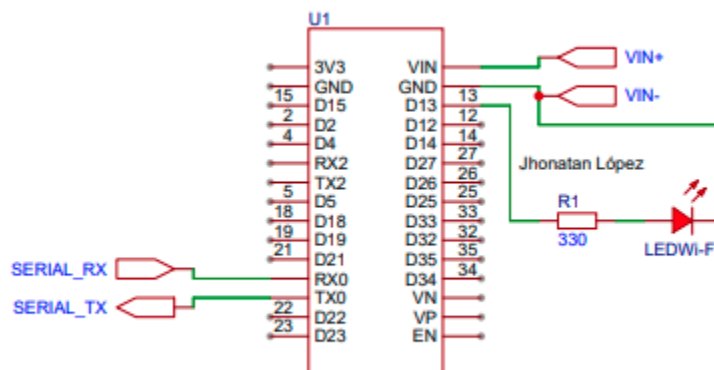
Figura 46 Esquemático LED de encendido (Autor)



Para el segundo LED se hace una conexión en el circuito de transmisión de datos, para poder verificar que la conexión Wi-Fi este activa, en este punto se toma el pin digital 13 del módulo ESP32 y se procede a conectar el LED y crear la configuración correspondiente para que funcione de manera correcta.

En la Figura 47 se aprecia el esquemático del circuito de transmisión de datos con el LED indicador del estado de la conexión Wi-Fi

Figura 47 LED de estado conexión Wi-Fi (Autor)



En la Figura 48 se puede ver la función para el ESP32 y que de esta manera encienda el LED para que el usuario visualice de manera más fácil el estado.

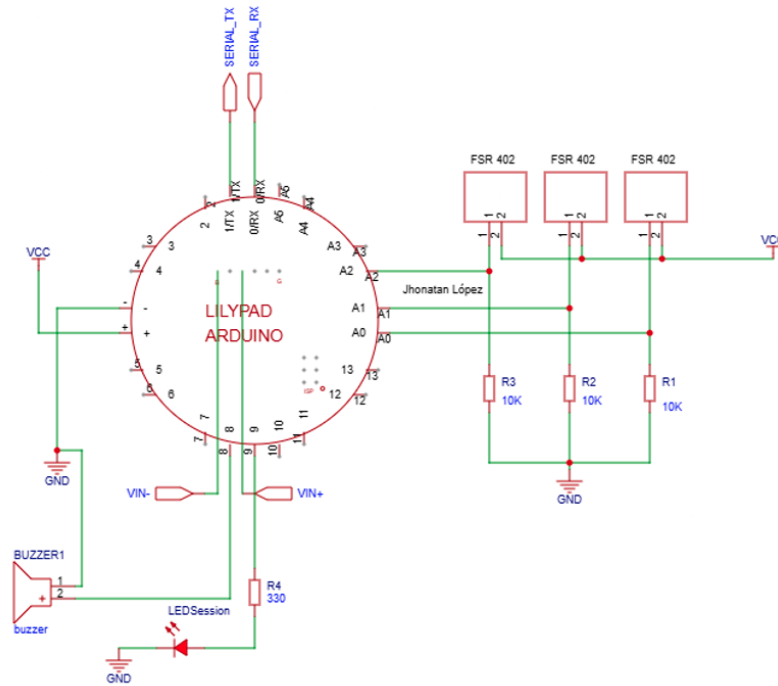
Figura 48 Función del ESP32 para gestión del LED (Autor)

```
void gestionarConexion() {
    static bool prevState = false;
    if (WiFi.status() != WL_CONNECTED) {
        digitalWrite(LED_WIFI, LOW);
        if (prevState) {
            Serial.println("Conexión WiFi perdida. Reconectando...");
            conectarWiFi();
        }
        prevState = false;
    } else {
        if (!prevState) {
            digitalWrite(LED_WIFI, HIGH);
            prevState = true;
        }
    }
}
```

Para el tercer y último LED se realizó una conexión en el pin digital 9 del circuito de obtención de datos, para que de esta manera se encienda el LED cada vez que una sesión esté activa, es decir cuando reciba el comando INICIAR hasta que tenga una señal de sesión completada o el comando DETENER.

En la Figura 49 se verifica el esquemático del circuito de obtención de datos en donde tiene un LED en el pin 9 para indicar el estado de la sesión de manera visual.

Figura 49 Esquemático circuito de obtención de datos con LED para la sesión (Autor)



En la Figura 50 se observa el comando `procesarComando`, el cual será el encargado de activar o desactivar el LED de acuerdo con el estado actual de la sesión.

Figura 50 Comando para indicar el estado de la sesión mediante un LED (Autor)

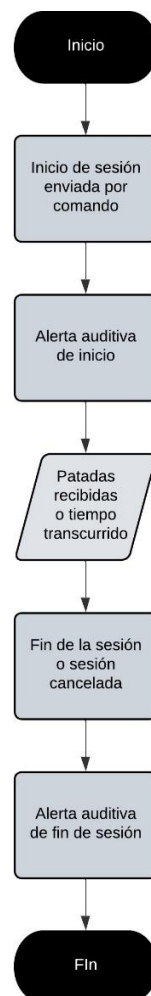
```
void procesarComando(String comando) {
  if (comando.startsWith("MODOT:")) {
    modoActual = TIEMPO;
    tiempoObjetivo = comando.substring(7).toInt() * 1000;
    Serial.println("CONFIG:TIEMPO");
  }
  else if (comando.startsWith("MODOR:")) {
    modoActual = REPETICIONES;
    repeticionesObjetivo = comando.substring(7).toInt();
    Serial.println("CONFIG:REPETICIONES");
  }
  else if (comando == "INICIAR") {
    if (!sesionActiva) {
      sesionActiva = true;
      tiempoInicio = millis();
      Serial.println("SESION_INICIADA");
      contadorRepeticiones = 0; // Resetear contador al iniciar
      digitalWrite(LED_SESSION, HIGH);
      alertaSonora(3000, 1000);
    }
  }
  else if (comando == "DETENER") {
    sesionActiva = false;
    digitalWrite(LED_SESSION, LOW);
    alertaSonora(1000, 1000);
    Serial.println("SESION_DETENIDA");
  }
}
```

3.3.1.5.2. Alertas auditivas

Para las alertas auditivas se consideran los casos de inicio y fin de sesión, por lo que estará directamente relacionado con el circuito de obtención de datos y es en este en donde se realiza la configuración e instalación del buzzer para que genere las alertas auditivas y el usuario pueda escuchar directamente la indicación de inicio o fin.

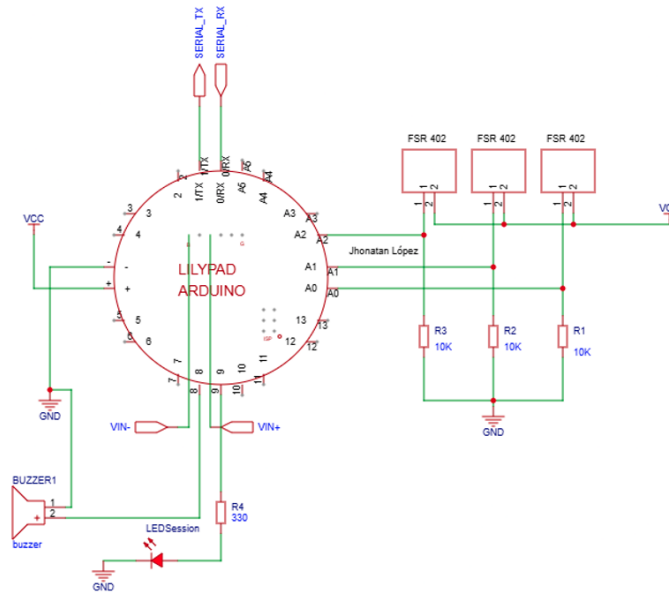
En la Figura 51 se detalla en diagrama de flujo que tendrán las alertas auditivas para cada sesión que se genere.

Figura 51 Diagrama de flujo para alertas auditivas (Autor)



Para realizar la conexión física se toma como pin designado el número 9 del Arduino Lilypad, el cual será el encargado de generar las señales hacia el buzzer y que este indique al usuario el inicio o fin de la sesión.

Figura 52 Esquemático del circuito de obtención de datos con buzzer (Autor)



El buzzer emite tonos diferenciados para cada evento importante. Al inicio de la sesión, el buzzer reproduce un tono ascendente (de 3000 Hz a 1000 Hz durante 1 segundo), confirmando que la sesión ha comenzado correctamente. Cuando la sesión finaliza exitosamente, se emite un tono grave largo (1000 Hz durante 1.5 segundos) para indicar que se alcanzó el objetivo.

En la Figura 53 se observa el comando procesarComando, el cual será el encargado de emitir las alertas sonoras, cuando inicie y finalice la sesión.

Figura 53 Función para emitir alertas auditivas (Autor)

```

void procesarComando(String comando) {
  if (comando.startsWith("MODOT:")) {
    modoActual = TIEMPO;
    tiempoObjetivo = comando.substring(7).toInt() * 1000;
    Serial.println("CONFIG:TIEMPO");
  }
  else if (comando.startsWith("MODOR:")) {
    modoActual = REPETICIONES;
    repeticionesObjetivo = comando.substring(7).toInt();
    Serial.println("CONFIG:REPETICIONES");
  }
  else if (comando == "INICIAR") {
    if(!sesionActiva) {
      sesionActiva = true;
      tiempoInicio = millis();
      Serial.println("SESION_INICIADA");
      contadorRepeticiones = 0; // Resetear contador al iniciar
      digitalWrite(LED_SESION, HIGH);
      alertaSonora(3000, 1000);
    }
  }
  else if (comando == "DETENER") {
    sesionActiva = false;
    digitalWrite(LED_SESION, LOW);
    alertaSonora(1000, 1000);
    Serial.println("SESION_DETENIDA");
  }
}

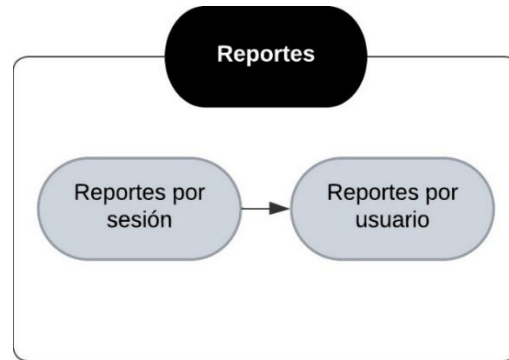
```

3.3.1.6. Reporte

Para el bloque de reportes se generaron dos aspectos importantes para que el entrenador pueda evaluar el desempeño de los alumnos, se consideran reportes por sesión de entrenamiento y por usuario.

En la Figura 54 se visualizan los bloques que contiene esta sección, para que el sistema sea mucho más amigable con el usuario y tenga una mejor funcionalidad.

Figura 54 Bloques de los reportes generados (Autor)



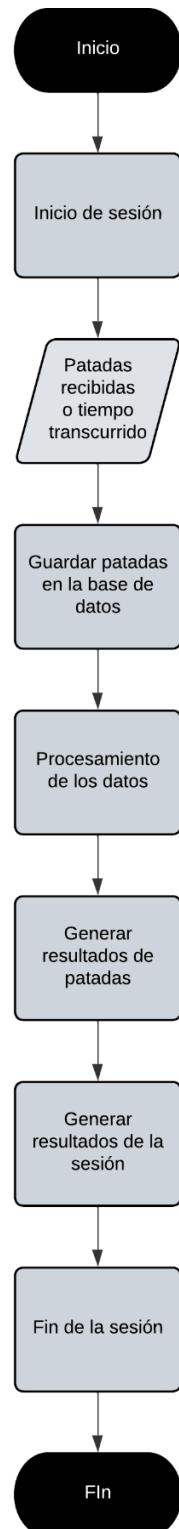
Cada usuario tiene un perfil donde puede acceder a su historial de sesiones, lo que le permite al entrenador realizar un seguimiento de su progreso a lo largo del tiempo. La tabla muestra las sesiones por fecha, métricas de rendimiento y estadísticas clave, asegurando que los usuarios puedan recuperar fácilmente la información relevante.

Para cada sesión, se presentan métricas detalladas de rendimiento, incluyendo el número de patadas ejecutadas, la distribución de fuerza y la precisión de los golpes. El sistema muestra estas métricas a través de visualizaciones dinámicas como gráficos y mapas de constelación, permitiendo a los usuarios y entrenadores analizar la efectividad del entrenamiento.

3.3.1.6.1. Reportes por sesión

En la Figura 55 se muestra el diagrama de flujo que tiene el reporte por sesión

Figura 55 Diagrama de flujo para reportes de sesiones (Autor)



Para poder generar los reportes por sesión de entrenamiento es necesario que al momento de crear una de ellas en el backend se asocien todas las patadas que pertenecen a esa sesión,

referenciando el UUID de la sesión a cada una de las patadas y que de esta manera se tenga un control claro de las patadas que se ejecutan y guardan en la base de datos.

Para lograr esto se realiza una función en el backend para que asocie las patadas que se generan a la sesión activa mediante una búsqueda en la base de datos con la sesión que tenga el estado “in_progress”

En la figura 56 se visualiza la función para asociar la patada a la sesión activa.

Figura 56 Función para asociar la patada a la sesión activa (Autor)

```
#####
# Create kick
def create_kick(db: Session, kick: KickCreate):
    # Buscar la sesión activa más reciente
    session_in_db = db.query(session_model.TrainingSession).filter(
        session_model.TrainingSession.status == "in_progress"
    ).order_by(session_model.TrainingSession.start_time.desc()).first()

    if not session_in_db:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail="No active session found"
        )

    # Crear la patada asignándole la sesión activa
    db_kick = kicks_model.Kick(
        session_id=session_in_db.id, # Asignar la sesión activa
        rep=kick.rep,
        s1=kick.s1,
        s2=kick.s2,
        s3=kick.s3,
        time=kick.time,
        processed=False
    )

    db.add(db_kick)
    db.commit()
    db.refresh(db_kick)

    return db_kick
```

Una vez se conoce que patadas pertenecen a una sesión y se realizó el proceso del algoritmo KNN para verificar si la patada es o no correcta se procede con la generación de los resultados de

la sesión con los datos directamente obtenidos del resultado de cada patada para poder formar un resultado global de cada sesión.

Para lograr este cometido se realiza una función PUT, en donde se generan los resultados globales de la sesión y se guarda en la base de datos, haciendo referencia a la sesión de entrenamiento correspondiente.

En la Figura 57 se visualiza la función que permite generar los resultados globales de la sesión.

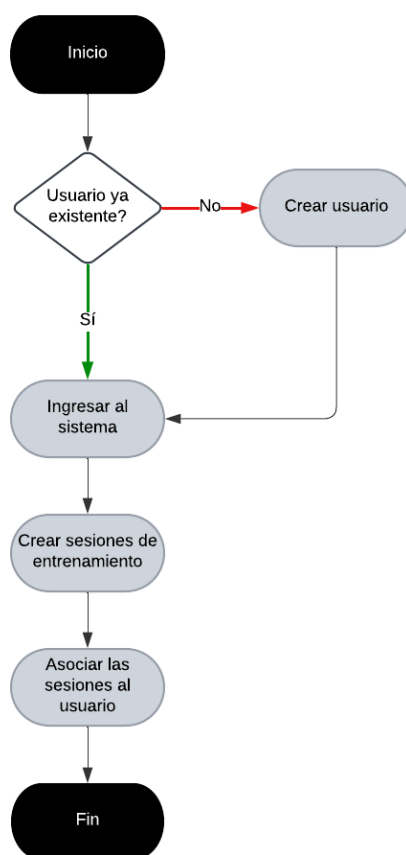
Figura 57 Función para generar resultados globales de la sesión (Autor)

```
@app.put("/api/results/session/{session_id}", response_model=ResultsResponse)
async def generate_session_results_endpoint(session_id: UUID, db: Session = Depends(get_db)):
    """
    Genera los resultados globales de la sesión (totales, correctas, incorrectas, etc.)
    usando los KickResult ya existentes para esa sesión.
    """
    try:
        result = generate_session_results(db, session_id)
        return result
    except HTTPException as he:
        raise he
    except Exception as e:
        raise HTTPException(status_code=status.HTTP_500_INTERNAL_SERVER_ERROR, detail=str(e))
```

3.3.1.6.2. Reportes por usuario

En la Figura 58 se visualiza el diagrama de flujo que se debe seguir para generar los reportes por usuario.

Figura 58 Diagrama de flujo para reportes de usuario (Autor)



Para poder generar los reportes por usuario el entrenador deberá ingresar al sistema de cada estudiante y ahí podrá observar todo el rendimiento que ha tenido a lo largo de las sesiones creadas y ejecutadas.

Los reportes por usuario estarán diseñados para ofrecer un análisis detallado del rendimiento de cada atleta en sus sesiones de entrenamiento. Cada usuario tendrá acceso a su propio panel de control, donde podrá visualizar un historial completo de sus sesiones organizadas cronológicamente.

En la Figura 59 se visualiza la función para crear una nueva sesión, en donde se requiere obligatoriamente el ID de un usuario.

Figura 59 Función para crear una sesión de entrenamiento (Autor)

```
#####  
# Create session  
def create_session(db: Session, session: TrainingSessionCreate):  
    db_session = session_model.TrainingSession(  
        user_id=session.user_id,  
        mode=session.mode,  
        repetitions=session.repetitions,  
        duration=session.duration,  
        status="in_progress"  
    )  
    db.add(db_session)  
    db.commit()  
    db.refresh(db_session)  
    return db_session
```

IV. Resultados

En este capítulo se presentan los resultados obtenidos de manera global para el sistema, estas serán llevadas a cabo en el Club de Artes Marciales Ecuadecuba “Black Panthers”, con el objetivo de verificar el correcto funcionamiento a nivel físico y lógico, mediante los distintos tests se verifican que cumplan con los requerimientos de stakeholders, sistema y arquitectura que fueron planteados para garantizar un sistema adecuado a las necesidades actuales.

4.1. Casos de pruebas

Se describen los casos que se llevan a cabo junto con los parámetros a evaluar, así como los resultados obtenidos en cada uno de ellos, con la finalidad de ver el funcionamiento específico del sistema. A continuación, se presentan mediante tablas los tests realizados al dispositivo, cada uno contienen los prerequisites, pasos realizados y resultados esperados.

4.1.1. Test eléctrico

En la Tabla 12 se muestra la tabla del proceso de evaluación para la parte eléctrica del dispositivo

Tabla 12 Test Eléctrico

Test eléctrico	Subsistema de prueba	Eléctrico
	¿Prueba de despliegue?	Sí/No
Descripción:		

Prueba inicial del sistema enfocada en verificar la correcta alimentación eléctrica del dispositivo. Se valida el funcionamiento del circuito de carga y elevación de voltaje compuesto por el módulo TP4056 y el MT3608, así como el encendido de los módulos Lilypad y ESP32. Esta prueba garantiza que los componentes puedan operar de manera estable sin interrupciones de suministro.

Prerrequisitos:

- a. Ensamblaje del circuito de alimentación (batería de litio 3.7v 3000mAh, módulo TP4056, y módulo MT3608).
- b. Conexión de salida de 5V regulada hacia el Lilypad y ESP32.
- c. Disponibilidad de multímetro digital para verificación de voltaje.

Pasos:

- Verificación de conexión de los módulos TP4056 y MT3608
- Comprobación del voltaje adecuado para los circuitos
- Verificación de encendido del dispositivo

Resultado esperado:

Todos los componentes deben encenderse correctamente y mantenerse estables. El voltaje debe mantenerse dentro del rango de operación (4.8 - 5.2V) para garantizar un funcionamiento confiable del sistema de adquisición y transmisión de datos.

Fuente: Autor

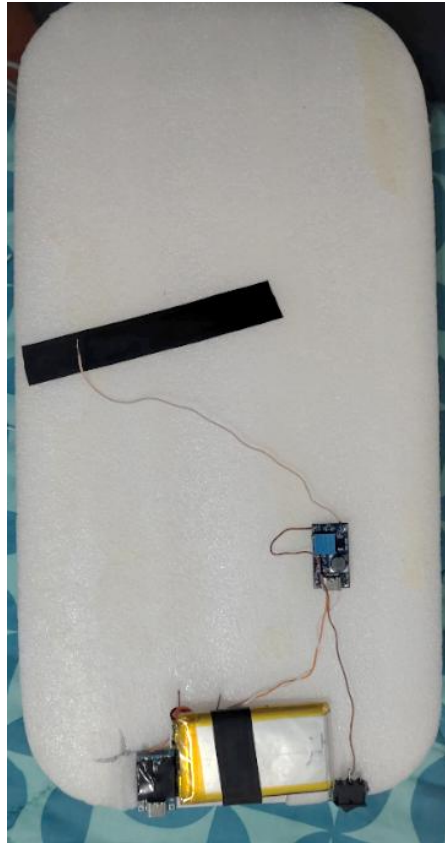
Resultado obtenido

Las pruebas eléctricas se verifican por el inicio y activación de los circuitos de obtención y transmisión de datos, para que funcionen estos deben ser alimentados con un voltaje de corriente continua de 5 voltios más menos 0.01 voltios , por lo que el circuito de alimentación brinda ese valor desde la batería, elevando el voltaje de 3.7 voltios y mediante un circuito rectificador mantiene el valor constante para que no exista daños en el circuito, así también cuenta con un circuito de control de sobre y bajo voltaje y prevención de cortocircuito, desconectando el suministro en caso de que se pueda tener el caso, protegiendo así la integridad de los componentes.

De igual manera para darle una protección adicional al prototipo, se colocó un disipador de aluminio para prevenir un sobrecalentamiento de los módulos de carga y control de voltaje y a su vez de la batería.

En la Figura 60 se observa el circuito de alimentación ya estructurado dentro del pateador.

Figura 60 Circuito de alimentación incorporado dentro del pateador (Autor)



Para verificar el inicio del dispositivo se colocó un LED directamente a la alimentación de los 5 voltios para verificar cuando el proyecto se encuentre encendido y de esta manera tener una representación visual clara de cuando el prototipo ya esté obteniendo la energía necesaria para funcionar.

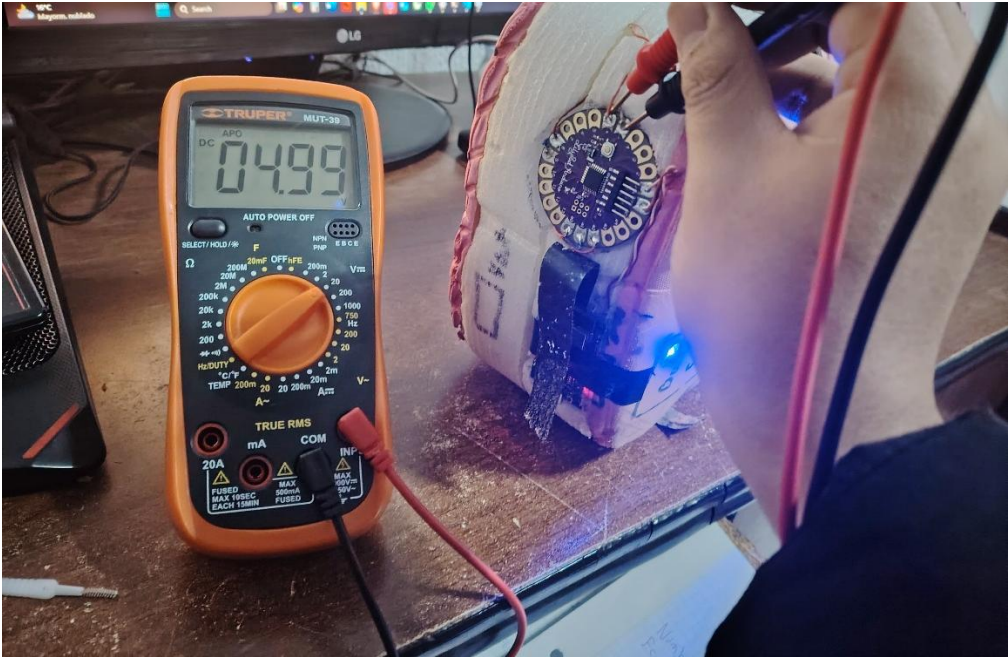
En la figura 61 se puede visualizar el LED de la alimentación de 5 voltios encendido para poder visualizar el estado del dispositivo.

Figura 61 LED de alimentación del dispositivo encendido (Autor)



En la Figura 62 se observa la medición de voltaje en la entrada del microcontrolador

Figura 62 Medida de voltaje



4.1.2. Test de hardware

En la Tabla 13 se muestran los pasos para realizar el test de los componentes del dispositivo.

Tabla 13 Test de Hardware

Test de hardware	Subsistema de prueba	Hardware
	¿Prueba de despliegue?	Sí/No
Descripción: Esta prueba tiene como objetivo verificar el correcto funcionamiento de la comunicación entre los diferentes módulos del sistema: el circuito de adquisición de datos (Arduino Lilypad con sensores FSR 402), el circuito de transmisión (ESP32), y la red local a través de WiFi. Se		

validan las conexiones físicas, la respuesta de los sensores y la comunicación serial entre los microcontroladores.

Prerrequisitos:

- a. Montaje completo del sistema de adquisición y transmisión en el pateador.
- b. Carga de los scripts en el Arduino Lilypad y el ESP32.
- c. Establecimiento de conexión WiFi local para el ESP32.
- d. Interfaz de consola incorporada para monitoreo de datos enviados.

Pasos:

- Verificación de conexión entre el circuito de transmisión de datos y la red local
- Verificación de la correcta comunicación entre el circuito de obtención de datos y el circuito de transmisión de datos
- Verificación del funcionamiento de los sensores

Resultado esperado:

El sistema debe detectar correctamente la fuerza aplicada en los sensores FSR y mostrar los valores en consola. Los datos deben transmitirse desde el Arduino Lilypad hacia el ESP32 por serial y luego ser retransmitidos por WiFi a la IP local establecida. Las lecturas deben reflejar variaciones consistentes al aplicar fuerza, sin pérdida de paquetes ni errores de transmisión.

Fuente: Autor

Resultado obtenido

Para verificar una correcta conexión entre el circuito de transmisión de datos y la red de área local tenemos un indicador LED el cual nos notifica de manera visual que el dispositivo se encuentra conectado hacia la red local, de igual manera podemos comprobar mediante un ping a la dirección IP 192.168.137.100, la cual se encuentra estática en el ESP32, de igual manera la interfaz gráfica nos permite observar mediante una API dedicada el estado del dispositivo.

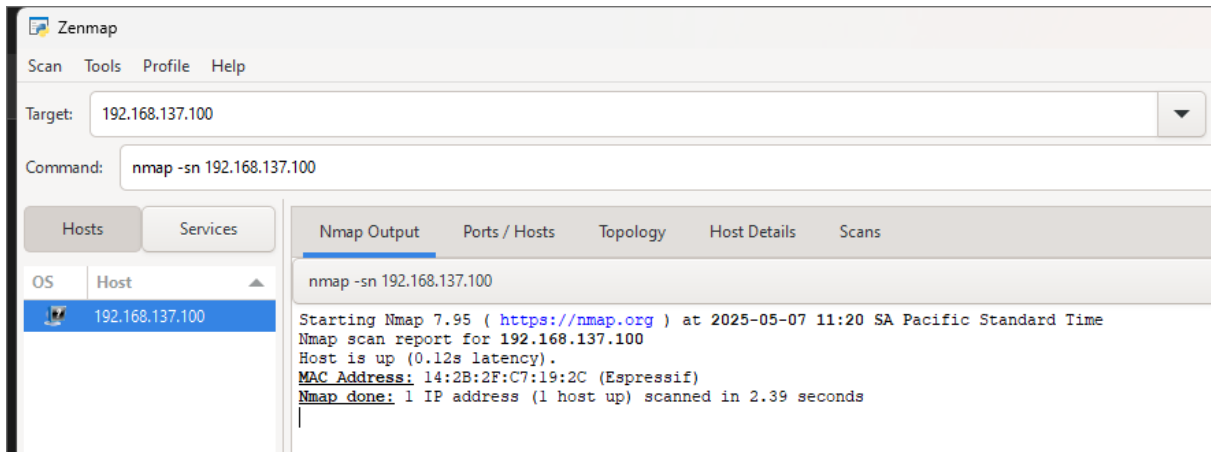
En la Figura 63 se visualiza el indicador LED de la conexión a la red local mediante WiFi encendido en el dispositivo.

Figura 63 LED de conexión WiFi encendido en el dispositivo (Autor)



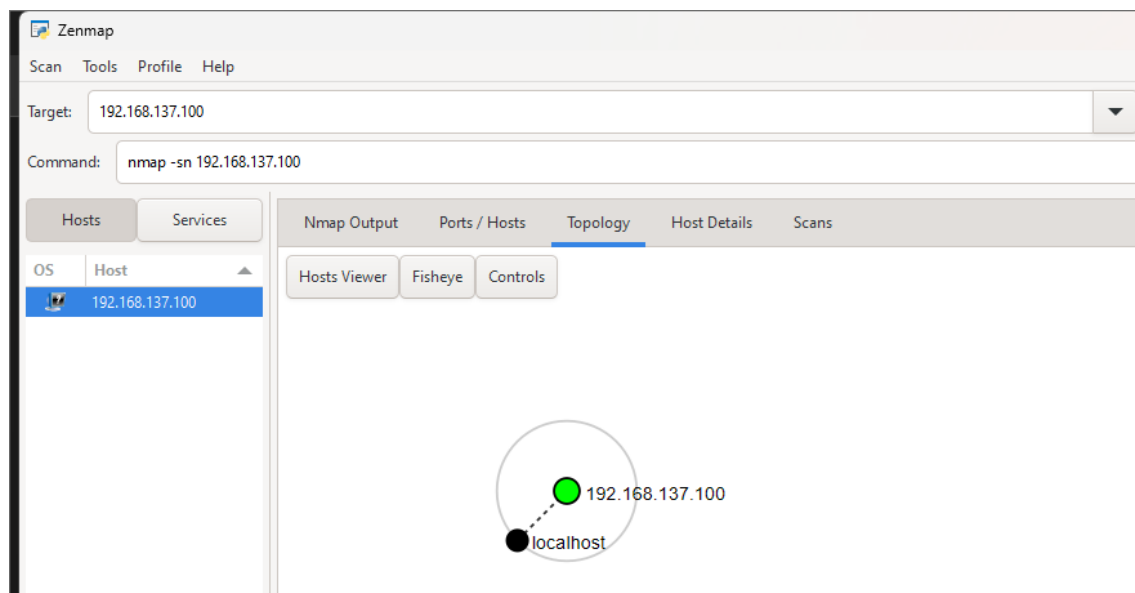
En la Figura 64 se puede visualizar la correcta comunicación entre la red de área local y el dispositivo mediante la función PING.

Figura 64 Ping hacia el dispositivo desde el localhost (Autor)



En la Figura 65 se puede visualizar el descubrimiento del dispositivo desde el localhost y de esta manera constatar que se encuentra conectado y operativo

Figura 65 Descubrimiento de la red de área local (Autor)



En la Figura 66 se puede visualizar el estado actual del dispositivo desde el frontend en la barra de navegación, verificando así que este se encuentra activo y listo para funcionar.

Figura 66 Verificación del dispositivo conectado desde el frontend



Para la verificación correcta de la actividad de los sensores y la comunicación de estos con el circuito de transmisión, es necesario realizar una prueba con golpes a los sensores y verificar que estos reciban la información adecuada y la transmitan hacia el circuito de transmisión de datos y este a su vez hacia el backend.

En la Figura 67 se visualiza la distribución de los sensores dentro del pateador, los cuales están ubicados de manera tal que se busca una adecuada focalización del golpe que se da.

Figura 67 Distribución física de los sensores en el pateador (Autor)



En la Figura 68 se visualizan desde el backend los valores generados por los sensores, que son recibidos y procesados desde el circuito de recopilación de datos.

Figura 68 Datos recibidos de los sensores y el circuito de recopilación de datos hacia el backend (Autor)

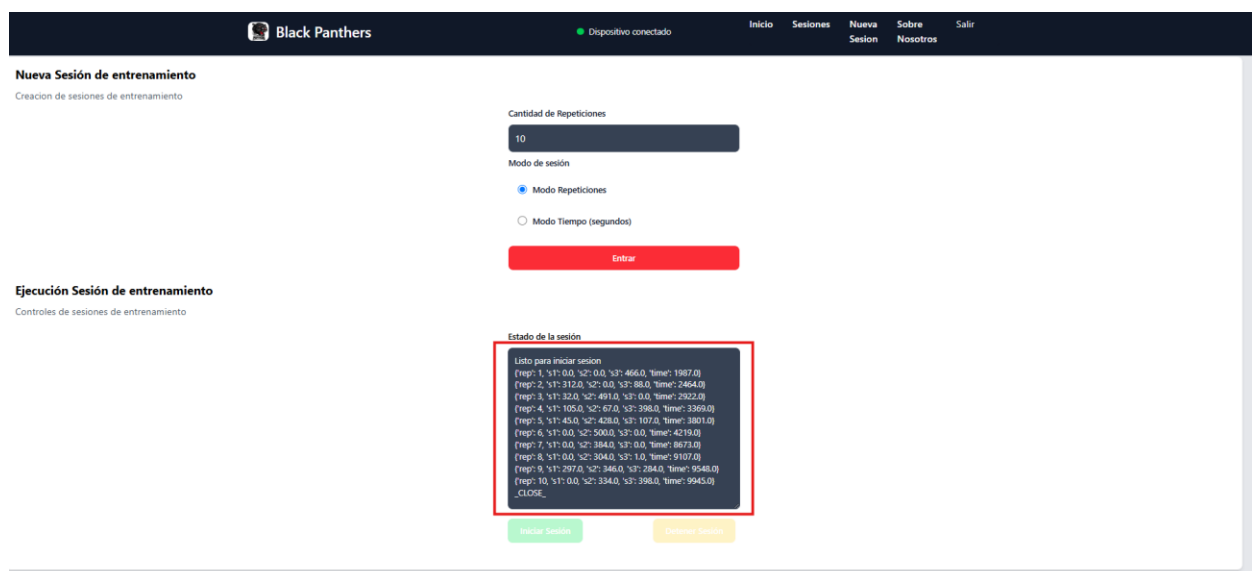
```

C:\Windows\system32\cmd.e X + v
INFO: 192.168.137.100:56674 - "GET /api/comandos HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53112 - "POST /api/session-with-command HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 192.168.137.100:56675 - "GET /api/comandos HTTP/1.1" 200 OK
INFO: ('127.0.0.1', 53117) - "WebSocket /api/ws/95663ad5-dc6f-429d-bbb2-dcb493e8751d" [accepted]
WS Fase1
registrando sesion con ID: 95663ad5-dc6f-429d-bbb2-dcb493e8751d
conexion registrada {'95663ad5-dc6f-429d-bbb2-dcb493e8751d': <starlette.websockets.WebSocket object at 0x00000228192EE3D0>}
WS Fase2
WS Fase3
WS Fase4
INFO: connection open
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 192.168.137.100:56676 - "PUT /api/comandos/8b145b1d-6de1-4499-9816-98c2715bef26 HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53120 - "POST /api/comandos/iniciar HTTP/1.1" 200 OK
INFO: 192.168.137.100:56677 - "GET /api/comandos HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53125 - "POST /api/comandos/iniciar HTTP/1.1" 200 OK
INFO: 192.168.137.100:56678 - "PUT /api/comandos/8b95ba26-7099-4eca-8fb7-190b0bfbb97 HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
INFO: 127.0.0.1:53080 - "GET /api/status HTTP/1.1" 200 OK
fase 1
Session activa <models.session_model.TrainingSession object at 0x0000022819339510>
Session activa 2 95663ad5-dc6f-429d-bbb2-dcb493e8751d
Datos recibidos: {'rep': 1, 's1': 0.0, 's2': 0.0, 's3': 466.0, 'time': 1987.0}
Conexion activa {'95663ad5-dc6f-429d-bbb2-dcb493e8751d': <starlette.websockets.WebSocket object at 0x00000228192EE3D0>}
Sesion que envio en el kick: 95663ad5-dc6f-429d-bbb2-dcb493e8751d
envio el ws: <starlette.websockets.WebSocket object at 0x00000228192EE3D0>
Si existe ws

```

En la Figura 69 se puede observar desde el frontend como se reciben los datos de los sensores y el circuito de recopilación de datos.

Figura 69 Datos recibidos de los sensores y el circuito de recopilación mostrados en el frontend (Autor)



4.1.3. Test de software

En la Tabla 14 se detallan los requerimientos que se evalúan en los componentes de software del sistema.

Tabla 14 Test de software

Test de software	Subsistema de prueba	Software
	¿Prueba de despliegue?	Sí/No
Descripción: Pruebas funcionales del software centradas en la visualización de datos, funcionamiento de interfaz gráfica, procesamiento y tratamiento de información, y generación de reportes.		
Prerrequisitos: a. Diseño de la base de datos e implementación de la lógica de backend. b. Configuración del entorno de desarrollo Visual Studio con Vite y Tailwind CSS. c. Programación de endpoints y rutas para el funcionamiento del frontend. d. Integración del backend FastAPI con el frontend React.		
Pasos:		

- Visualización de registro de usuarios.
- Visualización del procesamiento de información.
- Visualización del tratamiento de la información.
- Visualización de alertas.
- Visualización de reportes de sesiones.

Resultado esperado:

El sistema debe permitir la visualización del flujo completo del entrenamiento: desde el registro del usuario, creación y monitoreo de sesiones, hasta la presentación de alertas y reportes dinámicos. Cada acción en la interfaz debe estar sincronizada con el backend y ser visible en el frontend, permitiendo que los datos capturados por los sensores sean procesados, analizados y visualizados en tiempo real o como parte del historial del usuario.

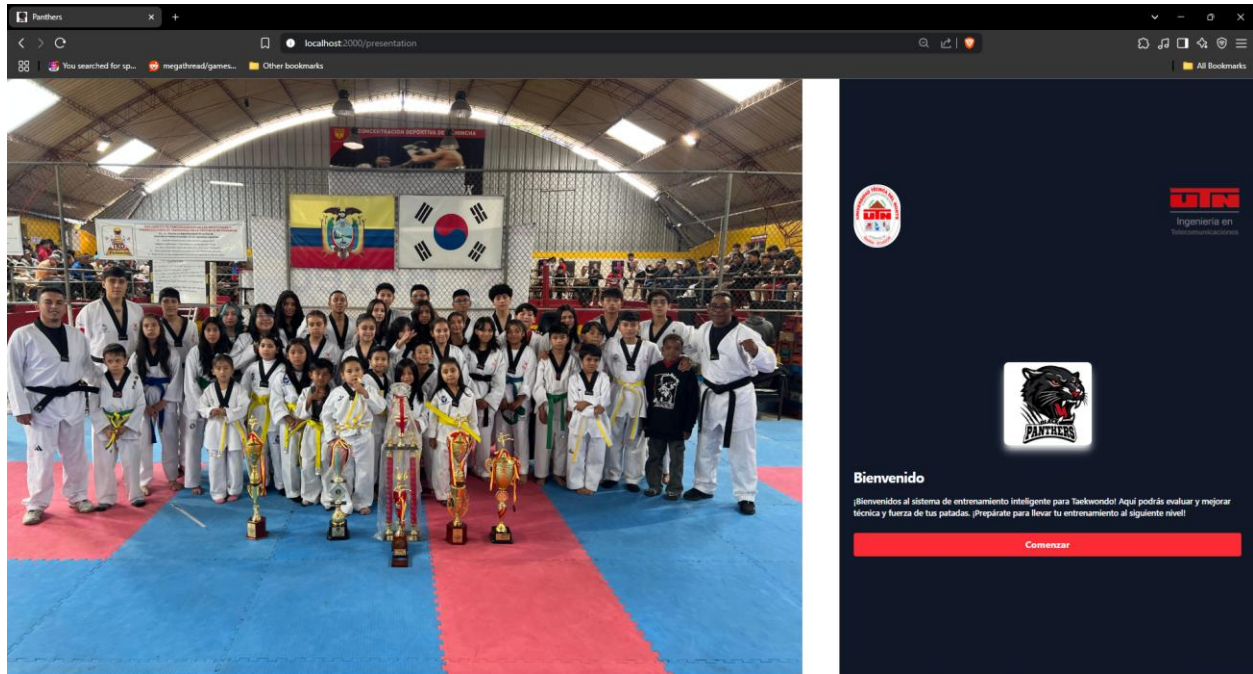
Fuente: Autor

Resultado obtenido:

El resultado obtenido se visualiza mediante el frontend del sistema, el cual permite navegar por todas las funcionalidades de este, ya que este se encuentra organizado de tal manera para que sea de fácil uso y amigable con el usuario, manteniendo los parámetros de calidad necesarios y que se mantenga sencillo en la manera de usar.

En la Figura 70 se visualiza la página de presentación del sitio para dar inicio al sistema.

Figura 70 Página de presentación del sistema (Autor)

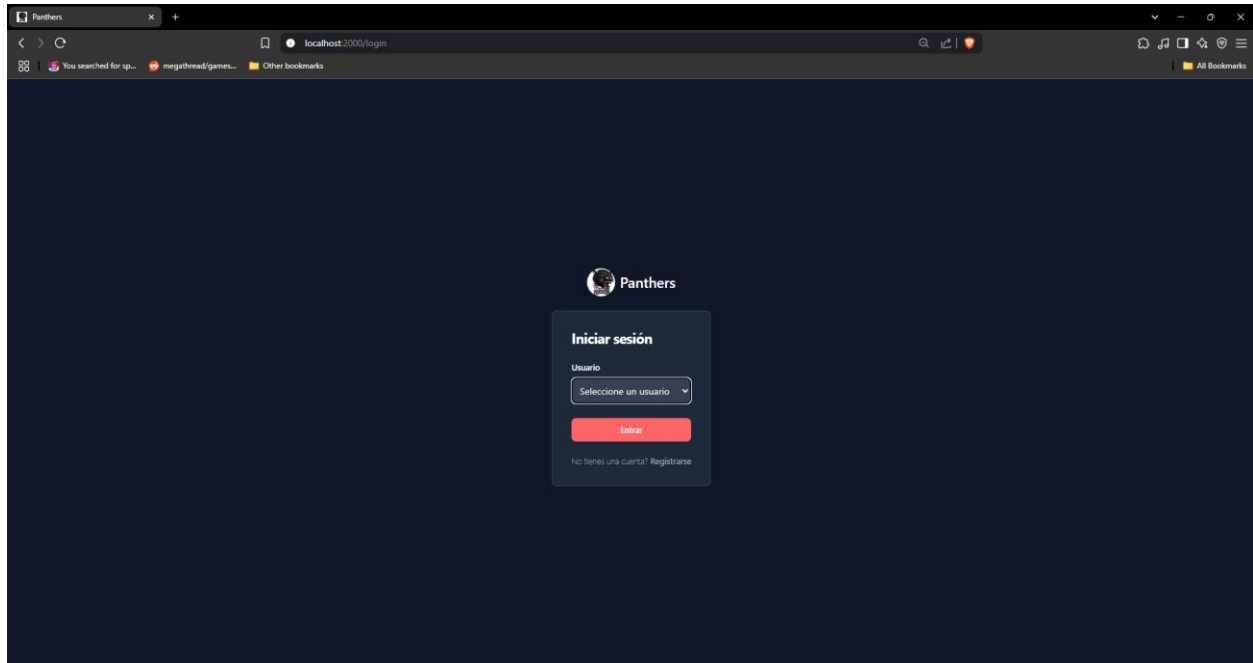


4.1.3.1. Pruebas de registro de usuario

Al ingresar al sistema, el usuario tiene dos opciones, el iniciar sesión si ya se encuentra registrado en el sistema o de no ser el caso el de crear su perfil y registrarse en el sistema, en caso de que el usuario ya se encuentre registrado se podrá elegir desde una lista desplegable el nombre de usuario ya existente.

En la Figura 71 se visualiza la página para poder iniciar sesión o crear un nuevo usuario en caso de que la persona no se encuentre registrada en el sistema.

Figura 71 Página de inicio de sesión o registro de nuevo usuario



Dentro de la página de registro se deben llenar datos importantes para poder tener un control del estudiante o deportista y de esta manera el entrenador pueda en cualquier momento verificar los antecedentes de este y evaluar de manera mucho más objetiva.

En la Figura 72 se visualiza la página para poder crearse un nuevo usuario en caso de que la persona no se encuentre registrada.

Figura 72 Página de registro de usuarios

Registrarse

Identificación: 1725212383

Nombre Completo: Jhonatan Eduardo López F

Cinturón: Negro 1º DAN

Categoría: Avanzado

Genero: Masculino

Fecha de Nacimiento: 03/19/2000

Username: jelopezf

Buttons: Registrarse, Volver

En la Figura 73 se puede visualizar el usuario ya creado en la base de datos y que puede ser visualizado en esta.

Figura 73 Visualización en la base de datos del usuario creado (Autor)

pgAdmin 4

Object Explorer

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X DBTesis/jhon@DB... X DBTesis/jhon@DBTesis X

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

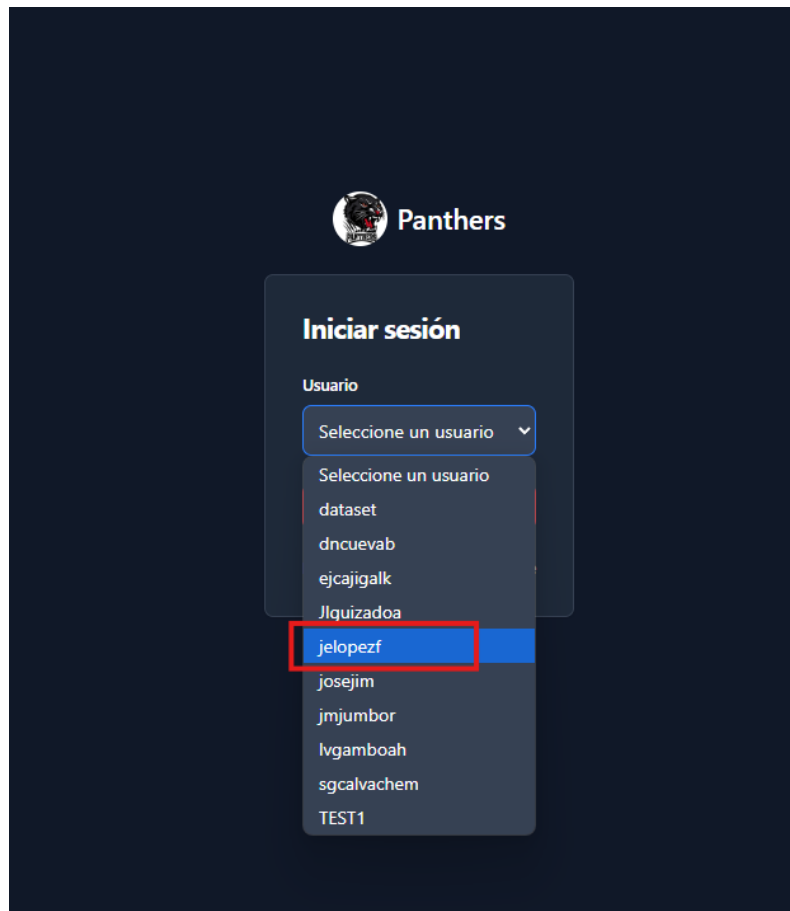
C:\Program Files\PostgreSQL\17\pgAdmin 4\runtime>pgcli.exe "host=database-1.cp2w6qwc4717.us-east-1.rds.amazonaws.com port=5432 dbname=DBTesis user=jhon sslmode=prefer connect_timeout=10" 2>>41
pgcli (17.2, postgres 16.3)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See pgcli reference
page "Notes for Windows users" for details.
SQL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: none)
Type "help" for help.

DBTesis=> SELECT *
DBTesis-> FROM users
DBTesis-> WHERE identification = '1725212383';
dat
-----
ac92ed0d-bdef-479c-8c00-8b7582153dfe | 1725212383 | Jhonatan Eduardo López Flores | Negro | Avanzado | 2000-03-19 | 25 | Masculino | jelopezf | 2025-04-24 20:33:57.191651
(1 row)

DBTesis=>
```

En la Figura 74 se puede visualizar el nombre de usuario ya existente en el frontend para poder iniciar sesión de una manera más rápida y fácil.

Figura 74 Visualización del usuario en la lista desplegable (Autor)



En la Figura 75 se puede visualizar en la página de inicio la información registrada del usuario creado.

Figura 75 Verificación de la información del usuario en la página de inicio (Autor)



4.1.3.2. Pruebas de procesamiento de la información

Para las pruebas de procesamiento de la información se va a verificar como llegan los datos de los sensores, como los recibe el backend y la manera en que el algoritmo KNN determina si son o no correctas las patadas que recibió.

Los sensores se encuentran ubicados en el centro del pateador, por lo que serán activados al momento que se genere una patada mayor al umbral determinado en el código para evitar falsos positivos de activación, el valor mostrado en el sensor es de 100 unidades, considerando que los sensores se encuentran recubiertos para evitar un daño en los mismos, de esta manera se creará como una patada válida si supera ese valor, de igual manera se tiene un valor máximo de 1000, de esta manera se puede hacer una validación con los tres sensores instalados, y programar parámetros adecuados para que mediante el algoritmo pueda detectar la validez de cada una de las patadas.

En la Tabla 15 se pueden visualizar los valores máximos y mínimos de los sensores

Tabla 15 Valores de los sensores usados

Sensor	Valor mínimo	Valor mínimo válido	Valor máximo
Sensores FSR 402	100	250	1000

Nota: Los valores son unidades calibradas con una resistencia de 10k ohmios y sensores de una resistencia máxima de 10 newtons. Fuente: Autor

Para poder realizar la prueba realizamos una cantidad determinada de patadas hacia el implemento para poder verificar como los sensores detectan el golpe y procesan dichos valores

En la Figura 76 se observa una demostración del autor realizando patadas al implemento para verificar los datos recibidos.

Figura 76 Prueba de pateo al implemento (Autor)



Los datos obtenidos por el sistema se basan en los parámetros previos y son mostrados de la siguiente manera como se detalla en la Figura 77, los cuales son presentados para que el profesor pueda visualizar en una escala de 0 a 1000 la fuerza recibida en cada sensor, junto con un número de patada y una marca de tiempo.

Figura 77 Datos recibidos de los sensores (Autor)

Estado de la sesión

```

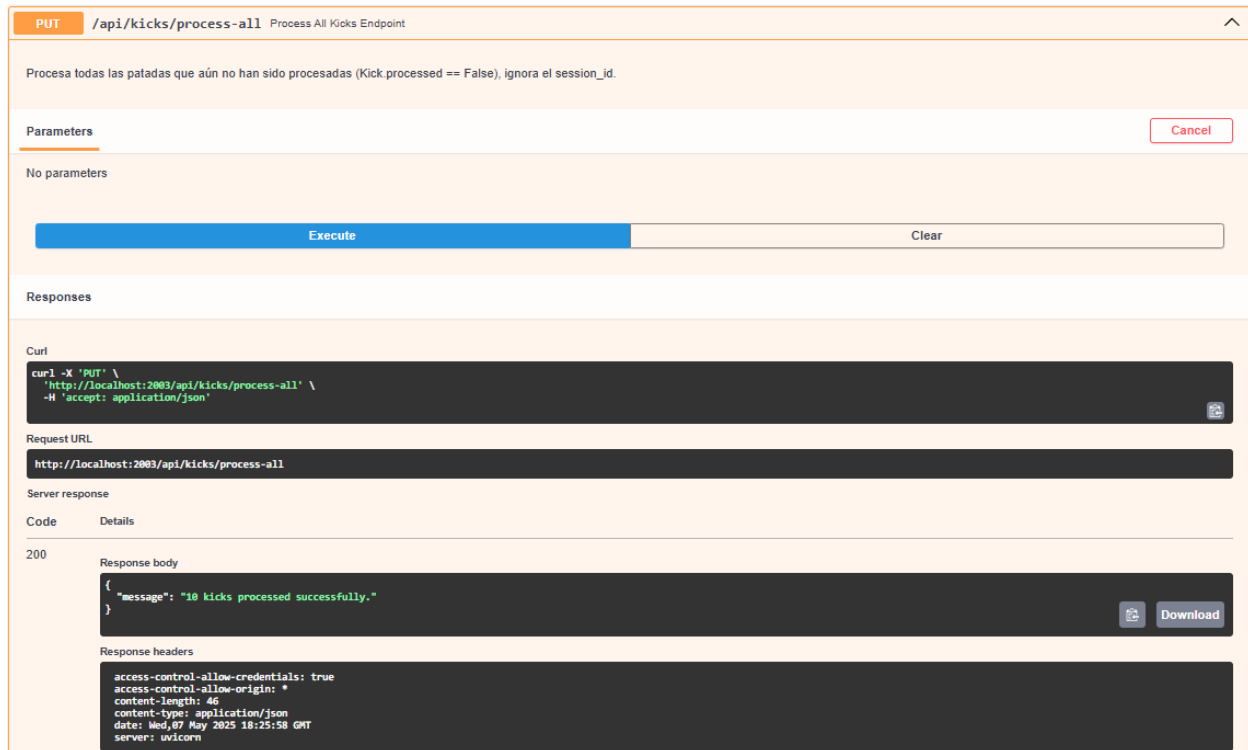
Listo para iniciar sesion
{'rep': 1, 's1': 0.0, 's2': 0.0, 's3': 557.0, 'time': 3865.0}
{'rep': 2, 's1': 492.0, 's2': 698.0, 's3': 299.0, 'time': 7088.0}
{'rep': 3, 's1': 395.0, 's2': 513.0, 's3': 420.0, 'time': 9890.0}
{'rep': 4, 's1': 0.0, 's2': 360.0, 's3': 0.0, 'time': 12730.0}
{'rep': 5, 's1': 0.0, 's2': 97.0, 's3': 0.0, 'time': 12992.0}
{'rep': 6, 's1': 1.0, 's2': 689.0, 's3': 349.0, 'time': 15315.0}
{'rep': 7, 's1': 0.0, 's2': 703.0, 's3': 550.0, 'time': 18878.0}
{'rep': 8, 's1': 599.0, 's2': 522.0, 's3': 501.0, 'time': 21678.0}
{'rep': 9, 's1': 0.0, 's2': 674.0, 's3': 0.0, 'time': 24154.0}
{'rep': 10, 's1': 0.0, 's2': 538.0, 's3': 0.0, 'time': 26540.0}
_CLOSE_

```

A continuación, se presenta la prueba de como el sistema realiza el análisis de las patadas emitidas y a su vez las determina como correctas e incorrectas, esto se realiza mediante el algoritmo KNN, el cual gracias a sus características puede continuar mejorando y brindando un mejor análisis a medida que los golpes sean o no adecuados.

En la Figura 78 se muestra el endpoint encargado del procesamiento de las patadas y como estas son clasificadas mediante el algoritmo KNN.

Figura 78 Procesamiento de las patadas por parte del backend (Autor)



4.1.3.3. Pruebas de tratamiento de la información

Para las pruebas del tratamiento de información es necesario verificar que todos los datos se encuentren almacenados en una base de datos creada específicamente para el sistema, ya que es necesario que la información se encuentre disponible en todo momento que se quiera iniciar una nueva sesión o consultar sesiones anteriores para verificar el progreso del estudiante.

Como primer punto se necesita que cada usuario cuente con sus propias sesiones de entrenamiento y que a su vez estas cuenten con sus patadas ejecutadas, los resultados de dichas patadas y los resultados finales que se generarán, para esto se va a realizar la prueba en la base de datos, que tenga la clave foránea del usuario asignada en cada una de las sesiones creadas.

En la Figura 79 se observa la búsqueda que se realiza para obtener todas las sesiones del usuario jelopezf en la base de datos.

Figura 79 Búsqueda relacional del usuario en la tabla de sesiones (Autor)

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X DBTesis/jhon@DB... X DBTesis/jhon@DB... X public.sessions/D... X DBTesis/jhon@DBTesis X

Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\PostgreSQL\17\pgAdmin 4\runtime>"C:\Program Files\PostgreSQL\17\pgAdmin 4\runtime\psql.exe" "host=database-1.cp2w6qwc4717.us-east-1.rds.amazonaws.com port=5432 dbname=DBTesis user=jhon sslmode=prefer connect_timeout=10" 2>> psql (17.2, server 16.3)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference page "Notes for Windows users" for details.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: none)
Type "help" for help.

DBTesis=> SELECT *
DBTesis-> FROM sessions
DBTesis-> WHERE user_id = 'ac92cd0d-bdef-479c-8c00-8b7582153d6e';

	id	status	user_id	mode	repetitions	duration	start_time	end_time
01 completed	f638d116-4035-4822-bcef-b856f75e003a	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	5	0	2025-04-24 20:34:50.134816	2025-04-24 20:35:26.0630	
59 completed	1b26eb89-560f-45b0-8a36-ae3c4e0c0e5a	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	5	0	2025-05-04 01:02:15.869382	2025-05-04 01:03:15.2429	
3 completed	f49b4618-00df-4810-964e-5bb375dc3356	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	7	0	2025-05-04 01:43:28.26751	2025-05-04 01:44:13.6575	
71 canceled	1832e1be-a56a-4781-b934-cf0e6de91aae	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-04 02:15:28.472268	2025-05-04 02:23:27.9561	
52 canceled	bcaaece7-9813-453a-8418-22a6890c0884	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-04 02:24:01.120227	2025-05-04 02:34:34.0383	
65 completed	614db798-af5b-42c9-8500-73e46b1d8974	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	4	0	2025-05-04 03:30:19.638133	2025-05-04 03:39:30.1537	
7 completed	b5782662-9807-40ac-a38b-3c81af23d65e	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	7	0	2025-05-04 03:40:45.059665	2025-05-04 03:41:27.5420	
94 canceled	3ef54682-3796-4fec-9b92-8929b778ad21	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:05:46.012953	2025-05-05 15:17:00.0903	
21 canceled	b20b298d-b864-4c9b-8ca6-89232f6a2c6e	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:17:07.364558	2025-05-05 15:17:57.7866	
17 canceled	5f6f6abe-fb23-4935-8bf7-3eb2b1b5feba	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:25:52.551526	2025-05-05 15:26:11.8926	
07 canceled	3adb7850-a7a9-40ce-8ea4-b0fec8870969	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:31:47.15697	2025-05-05 15:32:11.0517	
6 completed	0c0fdec9-987f-4e23-9532-8914a67a8d1e	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:32:11.520208	2025-05-05 15:33:09.5091	
02 completed	f3d3a067-4a43-4355-9d40-7ecd72485672	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:38:43.272366	2025-05-05 15:39:23.7238	
72 canceled	0e5b462f-65dc-454a-b29c-66b17da11c33	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:41:42.745222	2025-05-05 15:42:21.4509	
83 canceled	b104d078-697f-465f-91e2-ade36bb9c44e	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	6	0	2025-05-05 15:47:01.566314	2025-05-05 15:47:54.4471	
48 canceled	aleb0741-6f18-4803-a8ba-69068206628c	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	4	0	2025-05-05 15:48:45.504515	2025-05-05 15:49:16.7793	
17 completed	e14cb594-5385-4c94-8f42-c07f738b28d8	ac92cd0d-bdef-479c-8c00-8b7582153d6e	R	4	0	2025-05-05 15:54:43.334752	2025-05-05 15:55:10.9664	
	bf024cf5-ec05-4b3f-943b-4b45e71bca0	ac92cd0d-bdef-479c-8c00-8b7582153d6e	T	0	4	2025-05-05 15:57:59.836971	2025-05-05 15:58:20.6626	

De igual manera en el backend existe un endpoint dedicado para este propósito, por lo que se puede realizar la misma búsqueda desde el backend tal como se muestra en la Figura 80

Figura 80 Sesiones del usuario jlopezf consultadas desde el backend (Autor)

GET /api/user/{user_id}/sessions Get Training Sessions By User

Parameters

Name Description

user_id * required
string
(path)

ac92cd0d-bdef-479c-8c00-8b7582153d6e

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:2003/api/user/ac92cd0d-bdef-479c-8c00-8b7582153d6e/sessions' \
  -H 'accept: application/json'
```

Request URL

http://localhost:2003/api/user/ac92cd0d-bdef-479c-8c00-8b7582153d6e/sessions

Server response

Code Details

200

Response body

```
{
  "mode": "R",
  "repetitions": 10,
  "duration": 0,
  "id": "d238cb8b-899d-4878-bf99-b575a864ae7",
  "user_id": "ac92cd0d-bdef-479c-8c00-8b7582153d6e",
  "start_time": "2025-05-07T18:21:17.269440",
  "end_time": "2025-05-07T18:21:50.298830",
  "status": "completed"
},
{
  "mode": "R",
  "repetitions": 10,
  "duration": 0,
  "id": "95663ad5-dc6f-429d-bbb2-dcb493e8751d",
  "user_id": "ac92cd0d-bdef-479c-8c00-8b7582153d6e",
  "start_time": "2025-05-07T16:39:12.177500",
  "end_time": "2025-05-07T16:39:52.555231",
  "status": "completed"
},
{
  "mode": "R",
  "repetitions": 10,
  "duration": 0,
  "id": "2b849dd2-9140-4746-bd5b-8138721aee70",
  "user_id": "ac92cd0d-bdef-479c-8c00-8b7582153d6e",
  "start_time": "2025-05-06T22:23:23.176290",
  "end_time": null,
  "status": null
}
```

Download

De igual manera se puede visualizar las patadas que están relacionadas con una sesión de entrenamiento, para esto se puede realizar una consulta en la base de datos y obtener todas las que corresponden a dicha sesión y de esta manera constatar la correcta relación de la base de datos.

En la Figura 81 se puede observar la consulta de las patadas que pertenecen a una sesión de entrenamiento, la cual contiene 10 patadas asociadas, cada una con sus valores de los sensores respectivos.

Figura 81 Consulta de las patadas pertenecientes a la sesión de entrenamiento (Autor)

```
DBTesis=> SELECT *
DBTesis-> FROM kicks WHERE session_id = 'd230cb0b-899d-4878-8f98-b57ba8648ae7';
      id      | rep | s1 | s2 | s3 | time | processed | session_id
-----+-----+-----+-----+-----+-----+-----+-----
17daf16e-cb89-490e-8d49-d7fac69e7246 | 5 | 0 | 97 | 0 | 12992 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
70135f81-e643-480b-a016-e24f5b3850a2 | 7 | 0 | 703 | 550 | 18878 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
7dd948ab-5fa1-466c-8e61-df1e5cde5721 | 1 | 0 | 0 | 557 | 3865 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
8b845f22-589e-4dad-be11-3cf8edf54c8f | 4 | 0 | 360 | 0 | 12730 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
8e551a97-a699-4d59-9a67-afe76d056d65 | 8 | 599 | 522 | 501 | 21678 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
bd0ce7e0-9271-4496-b89c-0f22e598eeca | 9 | 0 | 674 | 0 | 24154 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
bd930b13-9b84-4146-ae3c-44750d1e507a | 10 | 0 | 538 | 0 | 26540 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
bece8768-3e75-4c7c-9239-9cca83340ba9 | 3 | 395 | 513 | 420 | 9890 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
da6b5e26-6e21-46f9-99c4-cfdffa7f3a900 | 2 | 492 | 698 | 299 | 7088 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
f7cd6227-93b3-4e5b-a172-5bf647c4c6f2 | 6 | 1 | 689 | 349 | 15315 | t | d230cb0b-899d-4878-8f98-b57ba8648ae7
(10 rows)

DBTesis=>
```

Para el siguiente paso se verifican los resultados de las patadas ya procesadas, estas se encuentran vinculadas tanto a la tabla kicks como a la tabla sessions, por lo que se puede buscar mediante el id de la sesión todas las patadas ya procesadas.

En la Figura 82 se observa la búsqueda de los resultados de las patadas generadas en la sesión, mediante el session_id correspondiente

Figura 82 Búsqueda de los resultados de las patadas generadas en la sesión (Autor)

```
DBTesis=> SELECT *
DBTesis-> FROM kick_results WHERE session_id = 'd230cb0b-899d-4878-8f98-b57ba8648ae7';
      confidence | id | average_force | kick_id | created_at | session_id | s1 | s2 | s3 | time | classification |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
adb62c58-3f76-4698-8759-93043b64dcbd | 7dd948ab-5fa1-466c-8e61-df1e5cde5721 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 0 | 557 | 3865 | correct |
1 | 185.66666666666666 | 2025-05-07 18:25:59.578483 |
89d5174a-0456-4a12-a9a4-fa5cc7a789ce | da6b5e26-6e21-46f9-99c4-cfdffa7f3a900 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 492 | 698 | 299 | 7088 | correct |
1 | 496.33333333333333 | 2025-05-07 18:25:59.580472 |
d2cf9237-103b-44ed-a266-92075c83c42d | bece8768-3e75-4c7c-9239-9cca83340ba9 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 395 | 513 | 420 | 9890 | correct |
1 | 442.66666666666667 | 2025-05-07 18:25:59.582471 |
9c750134-7f2e-403d-9654-c58dba2f3df8 | 8b845f22-589e-4dad-be11-3cf8edf54c8f | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 360 | 0 | 12730 | correct |
1 | 120 | 2025-05-07 18:25:59.583471 |
d0429561-8b45-4b61-a79c-27be3c34da1 | 17daf16e-cb89-490e-8d49-d7fac69e7246 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 97 | 0 | 12992 | incorrect |
1 | 32.33333333333333 | 2025-05-07 18:25:59.584471 |
5a9659ae-e7b6-4af2-be00-ddb00cf7be58 | f7cd6227-93b3-4e5b-a172-5bf647c4c6f2 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 1 | 689 | 349 | 15315 | correct | 0.6
6666666666666666 | 346.33333333333333 | 2025-05-07 18:25:59.585979 |
0eb14a49-c8f5-4c11-80d0-c88ded785daf | 70135f81-e643-480b-a016-e24f5b3850a2 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 703 | 550 | 18878 | correct |
1 | 417.66666666666667 | 2025-05-07 18:25:59.58902 |
d3f45385-72c7-4585-a005-fd1f1de93c75 | 8e551a97-a699-4d59-9a67-afe76d056d65 | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 599 | 522 | 501 | 21678 | correct |
1 | 540.66666666666666 | 2025-05-07 18:25:59.591987 |
f5c69338-d559-4365-9a8a-e0bd512bf375 | bd0ce7e0-9271-4496-b89c-0f22e598eeca | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 674 | 0 | 24154 | correct | 0.6
6666666666666666 | 224.66666666666666 | 2025-05-07 18:25:59.59399 |
715a66a9-e2dd-446a-8208-5e975e714830 | bd930b13-9b84-4146-ae3c-44750d1e507a | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 0 | 538 | 0 | 26540 | correct | 0.6
6666666666666666 | 179.33333333333334 | 2025-05-07 18:25:59.596559 |
(10 rows)
```

Como último paso se procede a verificar los resultados generales de la sesión, por lo que se busca mediante el session_id en la tabla results el resultado, en donde se ven resultados de promedio de fuerza, cantidad de patadas correctas e incorrecta y la precisión de la sesión.

En la Figura 83 se visualiza la búsqueda realizada en la base de datos para los resultados de la sesión

Figura 83 Búsqueda del resultado de la sesión por session_id (Autor)

```
DBTesis=> SELECT *
DBTesis-> FROM results WHERE session_id = 'd230cb0b-899d-4878-8f98-b57ba8648ae7';
      id | session_id | total_kicks | avg_force | correct_kicks | incorrect_kicks | accuracy | session_time | cre
-----+-----+-----+-----+-----+-----+-----+-----+-----
2da317fb-c4c7-45be-bfd0-3a38ee44efed | d230cb0b-899d-4878-8f98-b57ba8648ae7 | 10 | 298.56666666666666 | 9 | 1 | 90 | 0 | 2025-05-07
18:51:18.755143
(1 row)

DBTesis=>
```

4.1.3.4. Visualización de alertas

Para poder realizar la verificación de las alertas visuales se realiza una prueba del prototipo en los casos que se generen las condiciones para que se activen los LEDs, para el primer caso se enciende el dispositivo presionando el interrupto a modo on, para que se alimente todo el sistema y se enciendan los módulos.

En la Figura 84 se puede verificar el LED de energía encendido en el prototipo.

Figura 84 LED de encendido del prototipo activo (Autor)



Para la siguiente alerta visual se realiza la conexión del prototipo con la red local que tiene el sistema incorporado y de esta manera se comprueba que el ESP32 se encuentra correctamente conectado, con la dirección IP asignada y con una comunicación constante con el backend.

En la Figura 85 se aprecia el LED de comunicación con el backend encendido, así como el LED de encendido del prototipo.

Figura 85 LED de conexión con el backend encendido en el prototipo (Autor)



Para la última alerta visual es necesario configurar la sesión de entrenamiento e iniciarla para que el tercer LED se encienda e indique al usuario que la sesión de entrenamiento se encuentra activa y el dispositivo está listo para recibir los golpes de la sesión.

En la Figura 86 se puede ver el tercer LED encendido para indicar que la sesión de entrenamiento se encuentra activa y lista para recibir patadas.

Figura 86 LED de sesión de entrenamiento iniciada activo (Autor)



Para las alertas auditivas el zumbador se encuentra junto a las alertas visuales, directamente apuntando hacia el entrenador o la persona que sostiene el implemento para que pueda ser escuchado de mejor manera, este se activa cuando inicia la sesión, cuando completa por tiempo o repeticiones y cuando se cancela, de esta manera se informa directamente a los usuarios los estados de la sesión.

En la Figura 87 se puede visualizar el zumbador colocado dentro del dispositivo para que pueda emitir las alertas de la sesión

Figura 87 Zumbador incorporado al prototipo (Autor)



4.1.3.5. Visualización de generación de reportes

Para la visualización de los reportes, se tiene una pestaña específica para este propósito, aquí se pueden encontrar todas las sesiones que tiene un usuario, en donde se indica, el id único de la sesión, el estado si fue completado o cancelado, la fecha en la que se establece la sesión, la hora de inicio y finalización, así como el modo y la cantidad de repeticiones que tuvo en ese entrenamiento.

En la Figura 88 se observa la pestaña de sesiones de entrenamiento que posee un usuario, en donde se indican los parámetros necesarios.

Figura 88 Pestaña de sesiones de entrenamiento (Autor)

ID SESSION	STATUS	FECHA	HORA INICIO	HORA FIN	MODOS	NRO PATADAS	ACTION
05cc05fb ...	✗ Cancelado	2025-05-09	02:33:47	02:35:11	Repeticiones	10	Ver
d230cb0b ...	✓ Completado	2025-05-07	18:21:17	18:21:50	Repeticiones	10	Ver
95663ad5 ...	✓ Completado	2025-05-07	16:39:32	16:39:52	Repeticiones	10	Ver
2b8494d2 ...	✓ Completado	2025-05-06	22:23:23	19:00:00	Repeticiones	10	Ver
3b926478 ...	✗ Cancelado	2025-05-06	22:22:31	19:00:00	Repeticiones	10	Ver

1 de 14 →

Para poder analizar un reporte se puede generar los resultados de dicha sesión, para esto, se crea una nueva pestaña en donde se pueden visualizar todos los datos relevantes de la sesión, con gráficas en donde se verifican los valores de fuerza de cada sensor por patada y también la cantidad de patadas correctas e incorrectas analizadas por el algoritmo.

En la Figura 89 se puede ver el reporte de una sesión de entrenamiento para el análisis del entrenador.

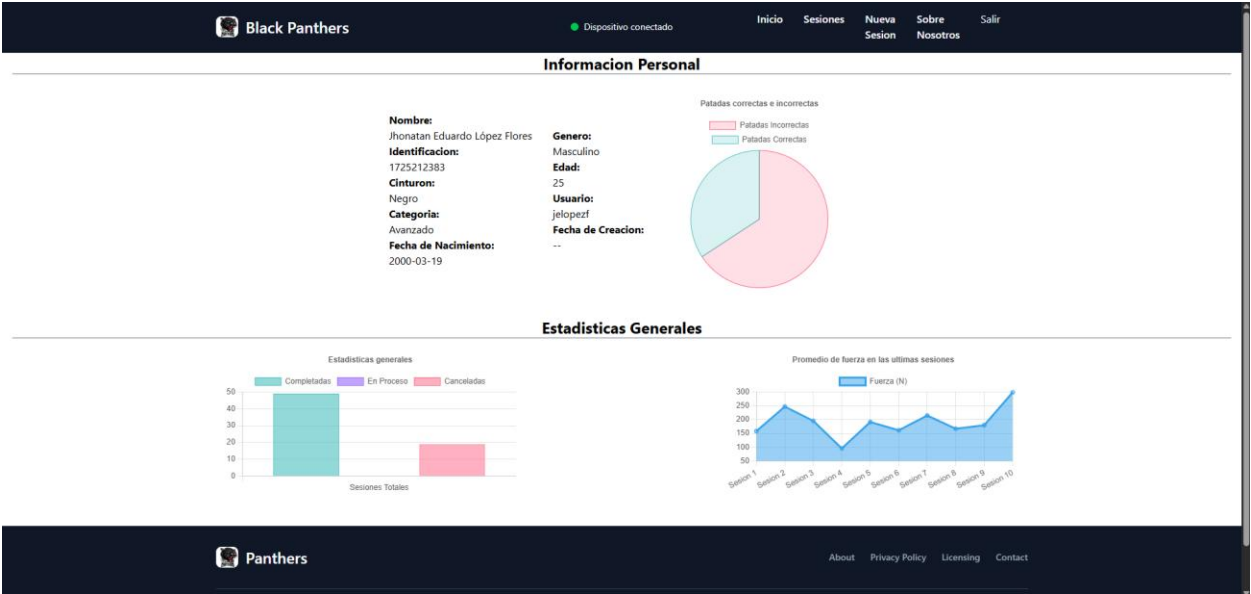
Figura 89 Reporte de una sesión de entrenamiento (Autor)



De igual manera se generan gráficas generales del usuario en la pestaña de inicio, aquí se puede visualizar el global de patadas correctas e incorrectas, la cantidad de sesiones completadas y canceladas y la fuerza promedio de todas las sesiones, de esta manera, se pueden ver parámetros relevantes del estudiante.

En la Figura 90 se puede visualizar la pestaña de inicio en donde se muestran las gráficas globales de las sesiones.

Figura 90 Datos globales del usuario (Autor)



4.1.4. Test de aplicación

En la Tabla 16 se puede ver el proceso del test de aplicación del prototipo.

Tabla 16 Test de aplicación

Test de aplicación	Subsistema de prueba	Aplicación
	¿Prueba de despliegue?	Sí/No
Descripción:		
Pruebas de funcionalidad completa de la aplicación en condiciones reales de uso, desde la autenticación de usuarios hasta el monitoreo de sesiones y visualización de resultados, comprobando que todos los módulos se integren correctamente.		

Prerrequisitos:

- a. Base de datos configurada con usuarios y sesiones previas.
- b. Conexión estable entre el circuito de adquisición de datos y el backend.
- c. Implementación del frontend funcional y conectado al servidor.
- d. Usuario registrados para uso del sistema en tiempo real junto con un entrenador.

Pasos:

- Visualización de usuarios registrados en la base de datos.
- Acceso al panel para crear nuevas sesiones.
- Monitoreo activo de patadas durante la sesión.
- Visualización automática de alertas durante la ejecución.
- Consulta de resultados y reportes al finalizar la sesión.

Resultado esperado:

El sistema debe permitir al entrenador acceder a los perfiles de usuarios y crear sesiones de entrenamiento nuevas. Durante la sesión, debe capturar correctamente los datos de las patadas, procesarlos, mostrar alertas y registrar los resultados en la base de datos. Finalizada la sesión, los datos deben visualizarse claramente desde el historial, permitiendo comparar y analizar el rendimiento del atleta.

Resultado obtenido:

Para verificar el funcionamiento de la aplicación se tomaron como referencia 15 estudiantes del club de artes marciales, se eligieron 3 alumnos por categoría junto con el director del club, para así lograr constatar como válido la prueba.

Como punto de partida de esta prueba, se coloca la PC dentro del club de Artes Marciales Ecuadrcuba, así también se le entrega el implemento al entrenador para que este haga uso de este en la sesión de entrenamiento con los alumnos del club, se inicializa el proyecto desde un archivo .batch el cual hará más fácil que el sistema se levante y los usuarios puedan acceder a él.

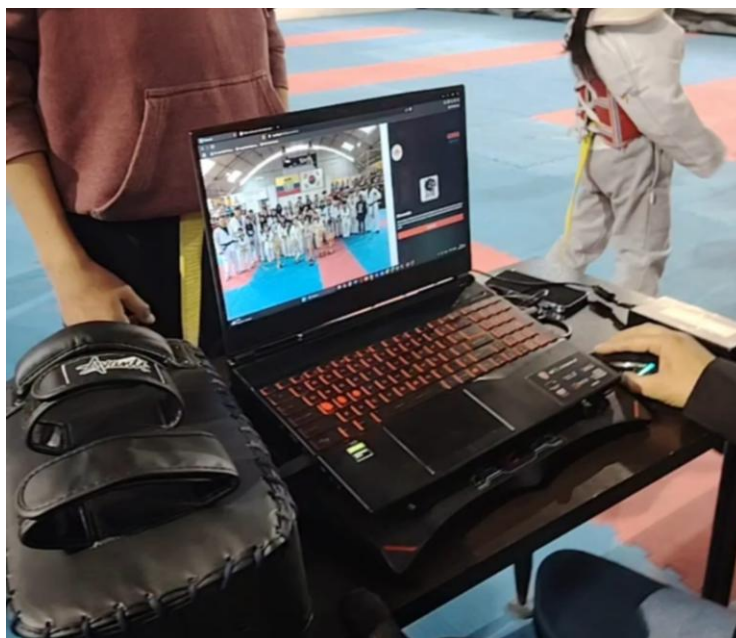
En la Figura 91 se observa la carpeta del proyecto con el archivo para inicializar el programa.

Figura 91 Visualización del archivo iniciador del programa (Autor)

Name	Date modified	Type	Size
backend	3/5/2025 19:56	File folder	
frontend	12/5/2025 21:01	File folder	
.gitignore	24/4/2025 12:29	Git Ignore Source ...	1 KB
Iniciar aplicación	5/5/2025 15:47	Windows Batch File	1 KB
package	24/4/2025 12:29	JSON Source File	1 KB
package-lock	24/4/2025 12:29	JSON Source File	3 KB
README	24/4/2025 12:29	Markdown Source...	1 KB

En la Figura 92 se puede ver el sistema instalado dentro del club para iniciar con la sesión de entrenamiento.

Figura 92 Sistema instalado dentro del club para pruebas (Autor)



Se procede a realizar el registro de los alumnos dentro de la base de datos para tener un control del avance a lo largo de las sesiones, el entrenador verificó los datos de los estudiantes dentro de la base de datos y los usuarios mostrados en el frontend.

En la Figura 93 se puede ver la base de datos de los usuarios registrados para iniciar las pruebas.

Figura 93 Base de datos de los estudiantes que realizarán la prueba (Autor)

id	identification	fullname	belt	category	birthday	gender	username
-4fca-bc8b-5816964e1e72	175 029 71	Dilan Nicolas Cueva Buñay	Verde	Blanco	2009-05-05	Masculino	dncuevab
-4ec9-a405-664d2097d52e	17134 04 2	Josue Mauricio Jumbo Ramos	Azul Rojo	Intermedio	2009-07-16	Masculino	jmjumbor
-48b2-95f0-38ff8016e50	120 032423	Silvana Gabriela Calvache Moreno	Blanco Amarillo	Blanco	2011-02-14	Femenino	sgcalvachem
-4684-bfd4-dcce06cfc1c23	1750 02 40	Luciana Valentina Gamboa Hurtado	Azul Rojo	Intermedio	2007-04-14	Femenino	lvgamboah
-4d78-8d73-cb6f4819a664	17 4705 79	Jefrey Leonel Guizado Anchundia	Azul Rojo	Avanzado	2013-09-16	Masculino	jlguizadoa
-4a0f-930e-4f19b1a538ff	17 90 4471	Guillermo Matias Loza Purcachi	Amarillo	Blanco	2009-02-25	Masculino	gmlozap
-4c27-a6b0-c02598f07c60	175 3 4976	Thiago Paul Teca Mora	Amarillo	Blanco	2017-11-05	Masculino	tpotecam
-4227-874d-a3b9509b6cae	17 755 855	Liam Jesus Muyulema Velasco	Verde Azul	Intermedio	2016-06-28	Masculino	ljmuyulemav
-4e72-940c-563c353d35ba	175 58 76	Mathuis Alexander Vera Flores	Amarillo	Blanco	2017-07-01	Masculino	maveraf
-4568-8aea-cdd3d32549c6	175528 5 2	Yarely Fernanda Rosero Espinoza	Azul Rojo	Avanzado	2014-03-22	Femenino	yfroseroe
-4134-9c67-671311f47858	17 5145420	David Andres Cueva Buñay	Verde	Avanzado	2014-02-28	Masculino	dacuevab
-4705-b168-bc4a57c335ac	175 8700 4	Luciana Isabella Cango Santillan	Blanco Amarillo	Blanco	2018-07-05	Femenino	licangos
-4e45-bfd9-78441e719035	17 357 985	Dilan Sebastian Saltos Bermeo	Amarillo Verde	Blanco	2011-08-01	Masculino	dsaltosb
-4ff0-8bfe-562b4da3f682	175 68 886	Matias Joaquin Lara Teran	Rojo Negro	Avanzado	2011-08-24	Masculino	mjlarat
-4b67-9848-a57d4209def1	172 0076 8	Daniel Leonardo Bastidas Almachi	Rojo	Intermedio	2005-11-22	Masculino	dbbastidasa
-4cf2-b050-0843bb5adca1	172 5 5025	Bryan Alexander Espinosa Egas	Azul Rojo	Intermedio	1996-04-15	Masculino	baespinosae

Durante las sesiones de entrenamiento programadas por el entrenador se configuraron una por una para poder hacer el entrenamiento personalizado, para esto se accedió a la pestaña de nueva sesión para poder configurarlas y enviar los comandos hacia el dispositivo.

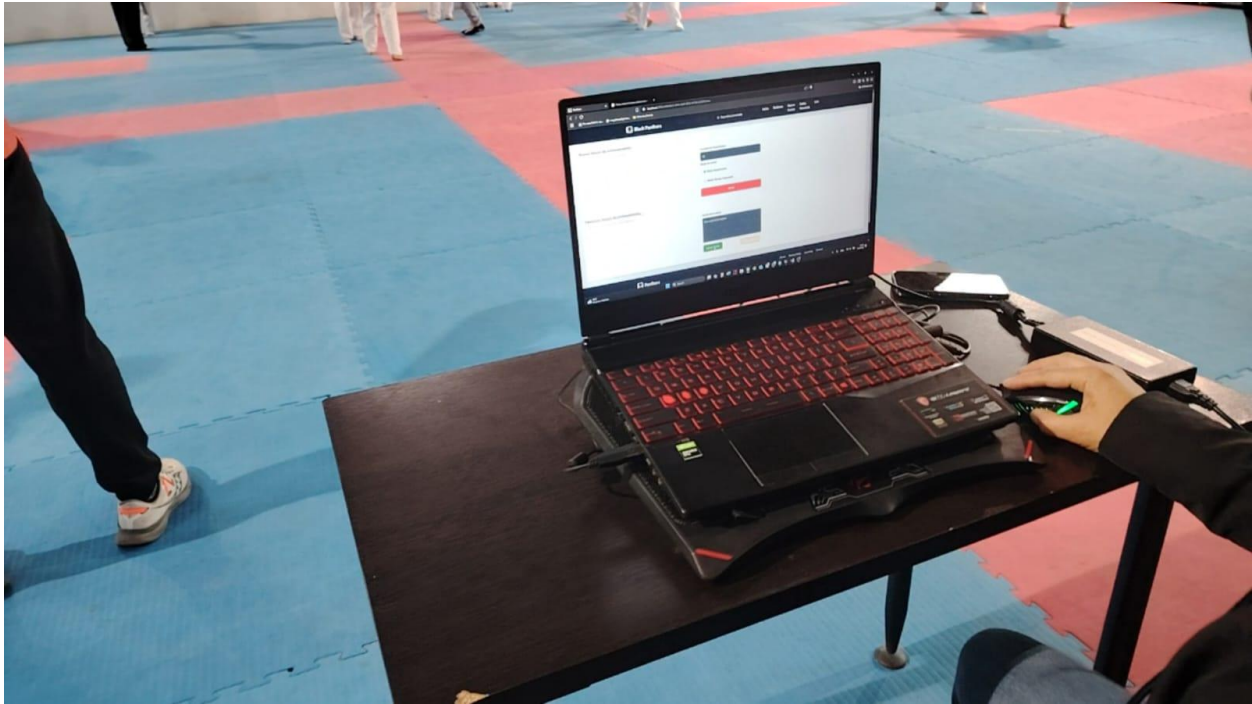
En la Figura 94 se visualiza la pestaña de nueva sesión para indicar el modo y la cantidad de repeticiones que tendrá la sesión.

Figura 94 Captura de pantalla de la pestaña nueva sesión (Autor)

The screenshot shows a web application interface for 'Black Panthers'. The top navigation bar includes a logo, a status indicator 'Dispositivo conectado', and menu items: 'Inicio', 'Sesiones', 'Nueva Sesión', 'Sobre Nosotros', and 'Salir'. The main content area is titled 'Nueva Sesión de entrenamiento' with the subtitle 'Creación de sesiones de entrenamiento'. It features a 'Valor' input field set to '10', a 'Modo de sesión' section with radio buttons for 'Modo Repeticiones' (selected) and 'Modo Tiempo (segundos)', a red 'Entrar' button, and a section titled 'Ejecución Sesión de entrenamiento' with the subtitle 'Controles de sesiones de entrenamiento'. Below this, the 'Estado de la sesión' shows a message 'No se ha iniciado ninguna sesión de entrenamiento' and two buttons: 'Iniciar Sesión' (green) and 'Detener Sesión' (yellow). The bottom footer contains the 'Black Panthers' logo and links for 'About', 'Privacy Policy', 'Licensing', and 'Contact'.

En la Figura 95 se visualiza la configuración de la sesión de entrenamiento en el club con los estudiantes.

Figura 95 Configuración de la sesión de entrenamiento en sitio (Autor)



Una vez iniciada la sesión el dispositivo recibirá los comandos de configuración e inicio, lo cual generará las alertas visuales y auditivas para indicar que la sesión ha comenzado, se ha cancelado o ha finalizado

En la Figura 96 se puede observar el dispositivo con las alertas visuales activas para indicar el estado de este al entrenador.

Figura 96 Alertas visuales del dispositivo activas (Autor)



Durante la sesión de entrenamiento se puede ir evidenciando las patadas que va registrando el dispositivo y como este las envía y va registrando en la base de datos para que puedan ser almacenadas y analizadas por el algoritmo KNN.

En la Figura 97 se puede ver al estudiante realizando el pateo al dispositivo con el apoyo del entrenador.

Figura 97 Estudiante realizando el pateo al prototipo (Autor)



En la Figura 98 se observan los datos que entran desde el dispositivo hacia el sistema, mostrando las patadas efectuadas por el estudiante.

Figura 98 Datos de patadas entrando al sistema (Autor)

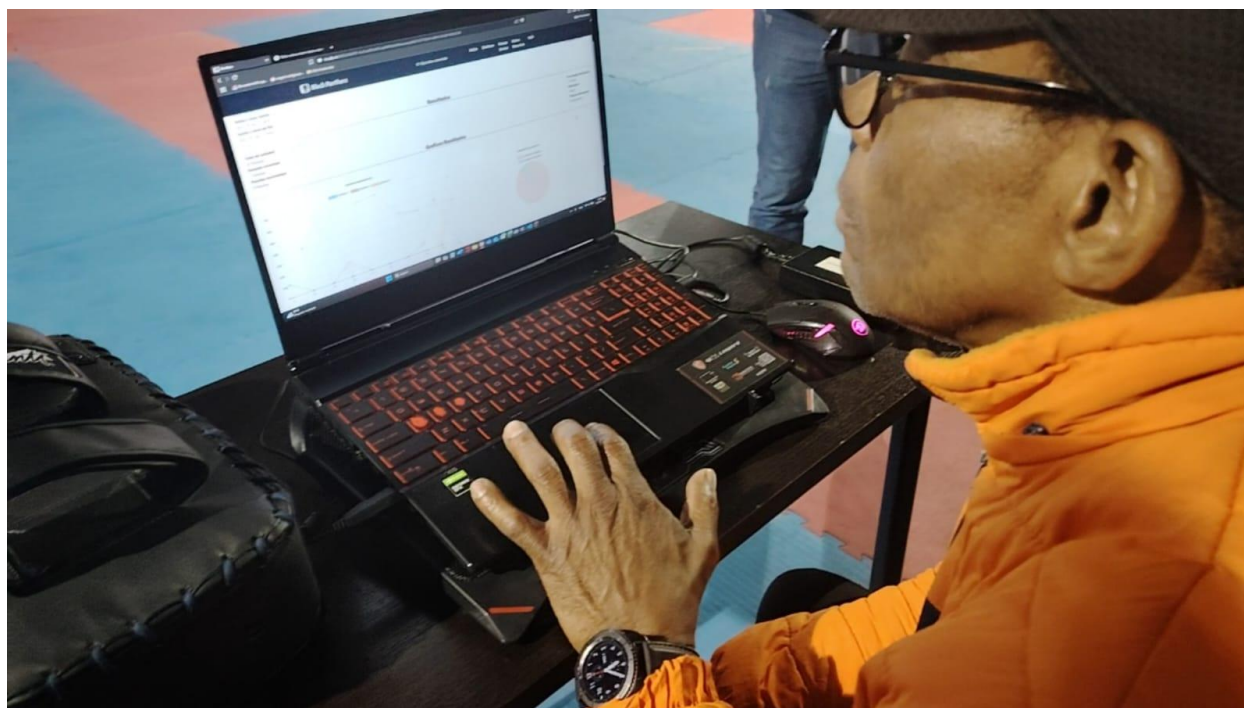
The screenshot shows the 'Black Panthers' web application interface. At the top, there's a navigation bar with links: Inicio, Sesiones, Nueva Sesión, Sobre Nosotros, and Salir. A status indicator shows 'Dispositivo conectado'. The main content area is divided into two sections: 'Nueva Sesión de entrenamiento' (Creation of training sessions) and 'Ejecución Sesión de entrenamiento' (Controls of training sessions). In the 'Nueva Sesión' section, there's a form with 'Cantidad de Repeticiones' set to 10, 'Modo de sesión' with 'Modo Repeticiones' selected, and a red 'Entrar' button. The 'Ejecución Sesión' section shows the 'Estado de la sesión' as 'Listo para iniciar sesión' and displays a table of session data. At the bottom, there are 'Iniciar Sesión' and 'Detener Sesión' buttons.

Rep.	s1	s2	s3	time
1	432.0	0.0	0.0	5386.0
2	294.0	2.0	0.0	7858.0
3	0.0	342.0	0.0	10504.0
4	0.0	593.0	0.0	13283.0

Al finalizar la sesión, el profesor tiene la posibilidad de verificar los resultados obtenidos de la misma, marcando la cantidad de patadas correctas e incorrectas, de igual manera dando un promedio total de la fuerza de la sesión, de esta manera él puede interpretar el desempeño del estudiante durante todas las patadas y de esta manera brindar una retroalimentación adecuada para que pueda mejorar en la capacidad necesaria.

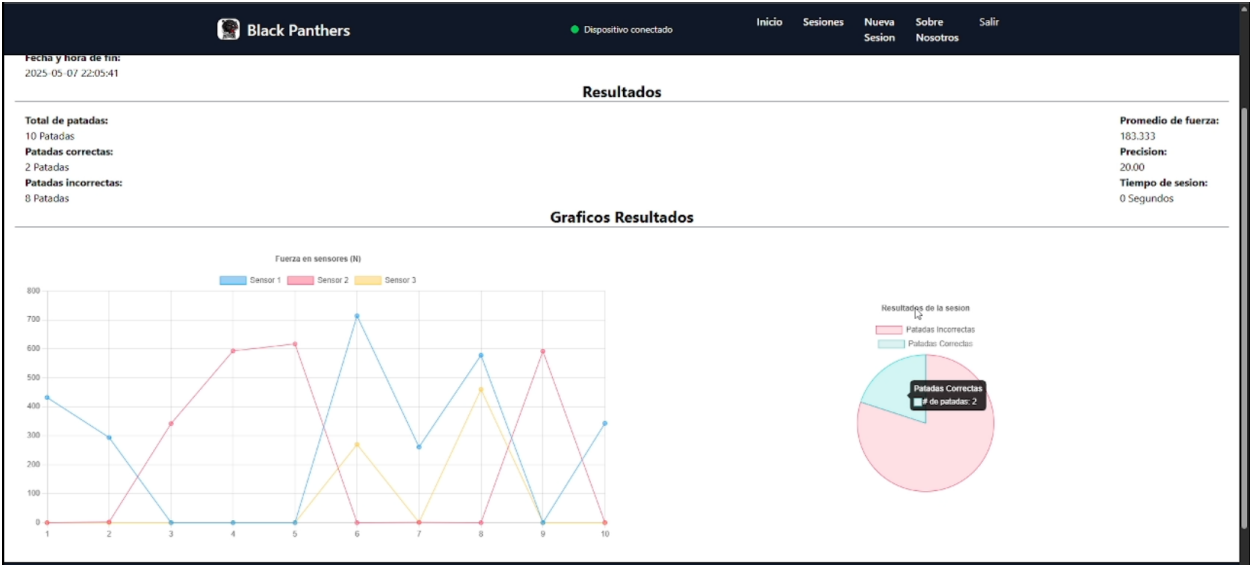
En la Figura 99 se puede visualizar al entrenador observando los resultados obtenidos en la sesión de entrenamiento.

Figura 99 Análisis de los resultados generados por el sistema por parte del entrenador (Autor)



En la Figura 100 se ven los reportes generados por el sistema para la interpretación del profesor.

Figura 100 Reporte de la sesión generado por el sistema (Autor)



4.1.5. Test funcional

En la Tabla 17 se verifican los procedimientos a realizar para validar la funcionalidad del proyecto.

Tabla 17 Test funcional

Test funcional	Subsistema de prueba	Funcionalidad
	¿Prueba de despliegue?	Sí/No
Descripción: Pruebas integrales del sistema para validar el correcto funcionamiento de todos los módulos en conjunto, incluyendo los procesos de encendido, interacción entre usuarios y entrenadores, ejecución de sesiones, análisis de patadas y generación de reportes finales.		

Prerrequisitos:

- a. Sistema completamente ensamblado y energizado.
- b. Usuarios registrados en el sistema junto con un entrenador.
- c. Interfaz gráfica desplegada correctamente.
- d. Módulos de adquisición, procesamiento y visualización activos.

Pasos:

- Encendido y verificación inicial del sistema.
- Inicio de sesión por parte del entrenador.
- Creación de una sesión de entrenamiento para un usuario.
- Transmisión de parámetros de entrenamiento al dispositivo.
- Ejecución del entrenamiento y recepción de patadas por parte del sistema.
- Análisis y clasificación automática de cada patada.
- Generación automática del reporte con métricas de desempeño.
- Visualización de resultados por parte del entrenador y del usuario.

Resultado esperado:

El sistema debe permitir una operación fluida desde su encendido hasta la finalización del entrenamiento. Debe identificar cada patada recibida, clasificarlas en correctas o incorrectas, procesar la información en tiempo real y reflejar los datos en reportes visuales claros y

completos. Además, el entrenador debe poder supervisar todo el proceso y acceder a la información histórica del usuario.

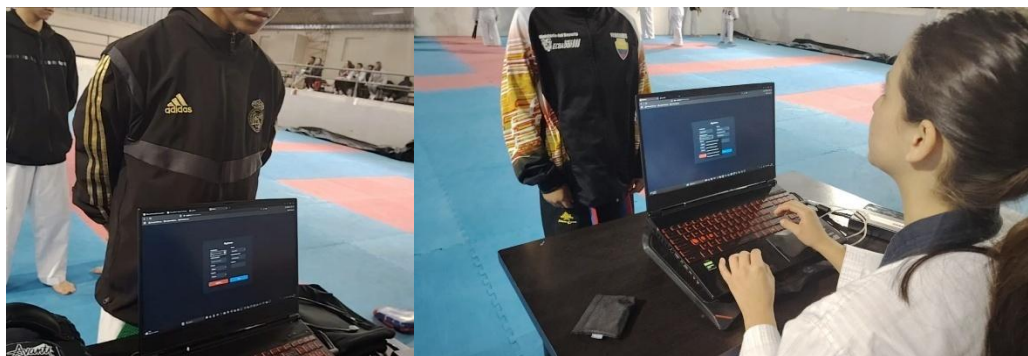
Fuente: Autor

Resultado obtenido:

A través de las pruebas realizadas dentro del club a 15 alumnos, se verifica que el sistema funciona correctamente, como primer punto se verificó que el dispositivo, el servidor y la interfaz gráfica se enciendan de manera correcta.

En la Figura 101 se puede visualizar el sistema funcionando de manera correcta y como se realizan registros de nuevos usuarios.

Figura 101 Registros de nuevos usuarios en el sistema (Autor)



Con los registros de los usuarios completos, se comienza con la sesión de entrenamiento, ingresando al perfil del estudiante que realizará las pruebas correspondientes, el entrenador puede configurar el dispositivo para que funcione en modo repeticiones o por tiempo. Es importante verificar que el dispositivo marque su estado como conectado.

En la Figura 102 se visualiza la pestaña de nueva sesión, en donde se tienen los modos de entrenamiento para la actividad a realizar.

Figura 102 Menú de selección del modo de entrenamiento (Autor)

The screenshot shows a web form titled 'Nueva Sesión de entrenamiento' with the subtitle 'Creacion de sesiones de entrenamiento'. It features a 'Valor' input field with the placeholder text 'Ingrese el valor'. Below this is the 'Modo de sesión' section, which includes two radio button options: 'Modo Repeticiones' (selected) and 'Modo Tiempo (segundos)'. At the bottom of the form is a red 'Entrar' button.

Una vez elegido el modo de la sesión de entrenamiento, el entrenador coloca la cantidad de repeticiones o tiempo que considera conveniente para la actividad planificada.

En la Figura 103 se puede observar configurado el modo repeticiones con 10 unidades y la alerta de la sesión iniciada.

Figura 103 Sesión de entrenamiento configurada con 10 repeticiones (Autor)

This screenshot shows the same 'Nueva Sesión de entrenamiento' form, but now with the value '10' entered in the 'Valor' field. The 'Modo Repeticiones' radio button remains selected. Below the 'Entrar' button, there is a new section titled 'Ejecución Sesión de entrenamiento' with the subtitle 'Controles de sesiones de entrenamiento'. This section includes a 'Estado de la sesión' label and a dark blue box containing the text 'No se ha iniciado ninguna sesión de entrenamiento'. At the bottom of this section are two buttons: 'Iniciar Sesión' (green) and 'Detener Sesión' (yellow).

Después de realizar la configuración de la sesión, el estudiante comienza con el pateo al implemento y en un proceso paralelo se verifica como los datos van ingresando al sistema, aquí el entrenador es un factor muy importante en la supervisión de la actividad.

En la Figura 104 se verifica como el estudiante realiza el pateo y el entrenador verifica los datos recibidos en el sistema.

Figura 104 Pruebas de pateo en el implemento por repeticiones (Autor)



En la Figura 105 se visualizan los datos que se obtienen cuando se recibe una patada en el sistema, mostrando los valores de cada uno de los sensores de fuerza y el tiempo en el que entra.

Figura 105 Valores entrantes en la sesión de entrenamiento (Autor)

Black Panthers

Inicio Sesiones Nueva Sesión Sobre Nosotros Salir

Nueva Sesión de entrenamiento
Creación de sesiones de entrenamiento

Tiempo en segundos
10

Modo de sesión
☒ Modo Tiempos
 ☐ Modo Repeticiones

Entrar

Ejecución Sesión de entrenamiento
Controles de sesiones de entrenamiento

Estado de la sesión

Lista para iniciar sesión

```

[resp: 1, s1: 0.0, s2: 494.0, s3: 0.0, time: 1811.0]
[resp: 2, s1: 279.0, s2: 0.0, s3: 0.0, time: 4293.0]
[resp: 3, s1: 661.0, s2: 0.0, s3: 500.0, time: 6984.0]
[resp: 4, s1: 0.0, s2: 13.0, s3: 402.0, time: 2432.0]

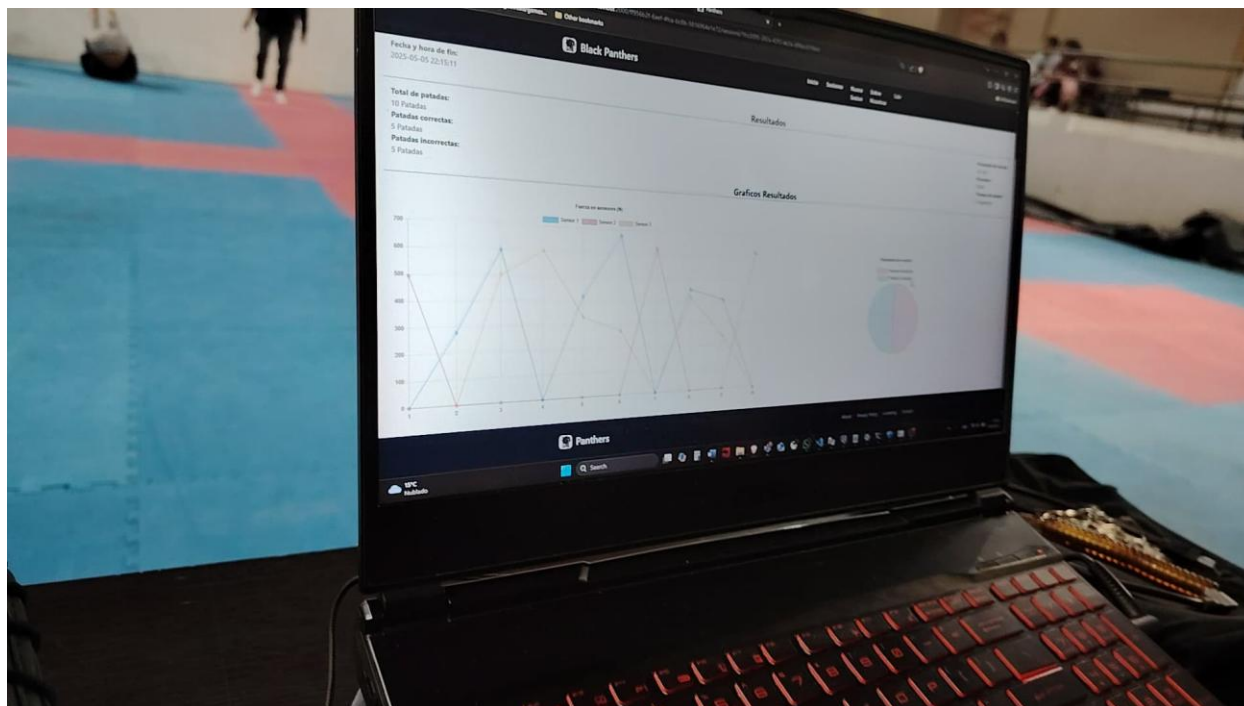
```

Final Sesión Definir Sesión

Al finalizar el entrenamiento se obtiene la alerta sonora del implemento y se procede a verificar los resultados obtenidos en la sesión de entrenamiento. Para este punto es importante que el entrenador verifique los resultados que se obtienen, para que él determine cuál fue el rendimiento que tuvo el practicante durante la sesión de entrenamiento.

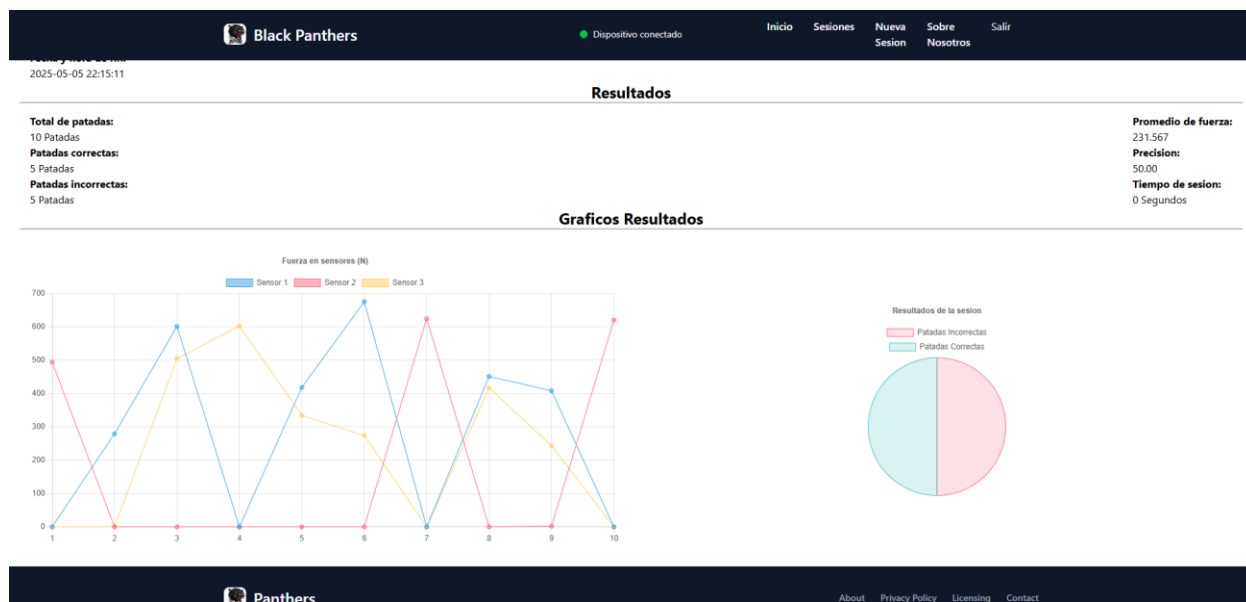
En la Figura 106 se visualizan los resultados de la sesión de entrenamiento.

Figura 106 Datos de la sesión de entrenamiento presentados de manera gráfica (Autor)



En la Figura 107 se ilustran los detalles de la sesión ejecutada, para esto se crea una gráfica de línea en donde se puede ver como fue el golpe por cada patada y por cada sensor, por lo que se puede comprobar que el 50% de las patadas fue correcta, teniendo una potencia media y los golpes dispersos, indicando que el estudiante no patea siempre en el mismo punto, si no que varía un poco en cada patada.

Figura 107 Resultados estadísticos y gráficos de la sesión de entrenamiento (Autor)



Para el modo de tiempo, se eligió a otro estudiante para que desempeñe el modo tiempo, para esto se estableció una sesión de 30 segundos para que el estudiante pueda desarrollar cualquier cantidad de patadas en el tiempo determinado, para esto se genera una nueva sesión de entrenamiento estableciendo los parámetros de esta

En la Figura 108 se puede visualizar la pantalla de generar una nueva sesión en el modo repeticiones con 30 segundos.

Figura 108 Nueva sesión de entrenamiento en modo tiempo 30 segundos (Autor)

Nueva Sesión de entrenamiento
 Creacion de sesiones de entrenamiento

Valor
 30

Modo de sesión

☐ Modo Repeticiones

☒ Modo Tiempo (segundos)

Entrar

Una vez configurada la sesión de entrenamiento se da inicio a la misma y se realiza el monitoreo del pateo del estudiante por parte de un entrenador, para visualizar como llegan los datos y se muestran en la interfaz.

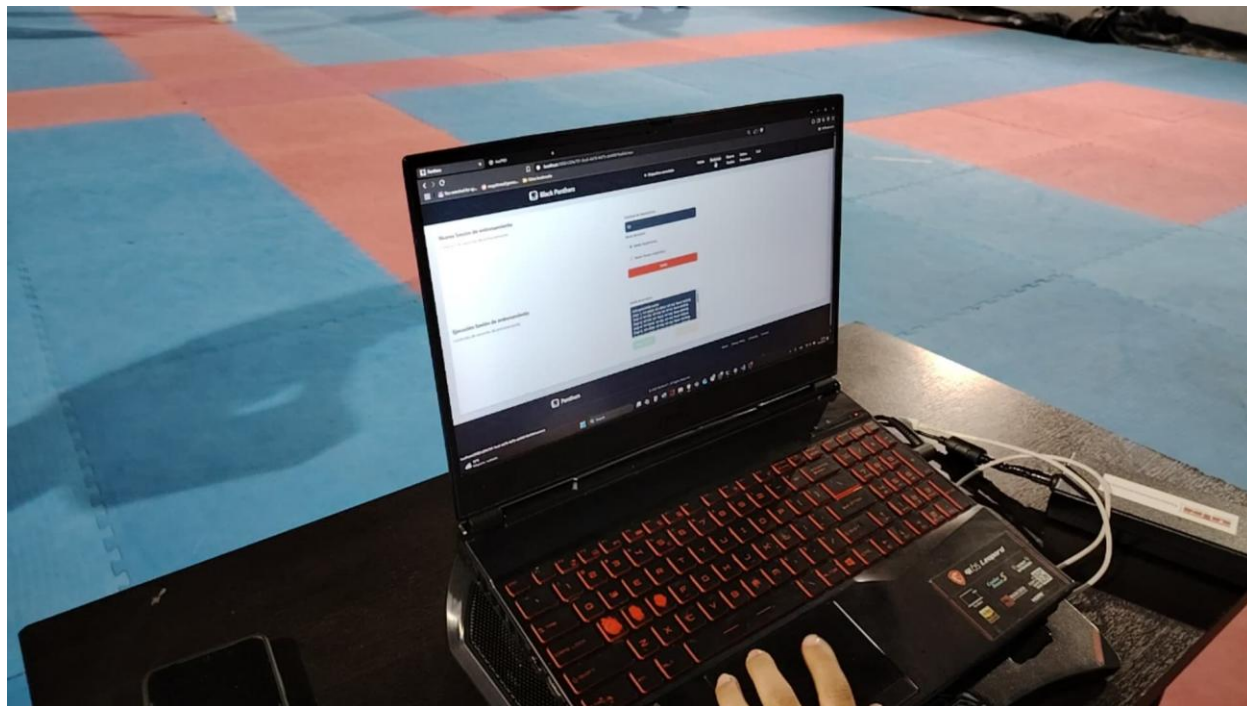
En la Figura 109 se puede visualizar al estudiante realizando el pateo al implemento con apoyo del entrenador para darle instrucciones.

Figura 109 Estudiante realizando el pateo al implemento (Autor)



En la Figura 110 se observa la monitorización de la sesión de entrenamiento durante el pateo del practicante.

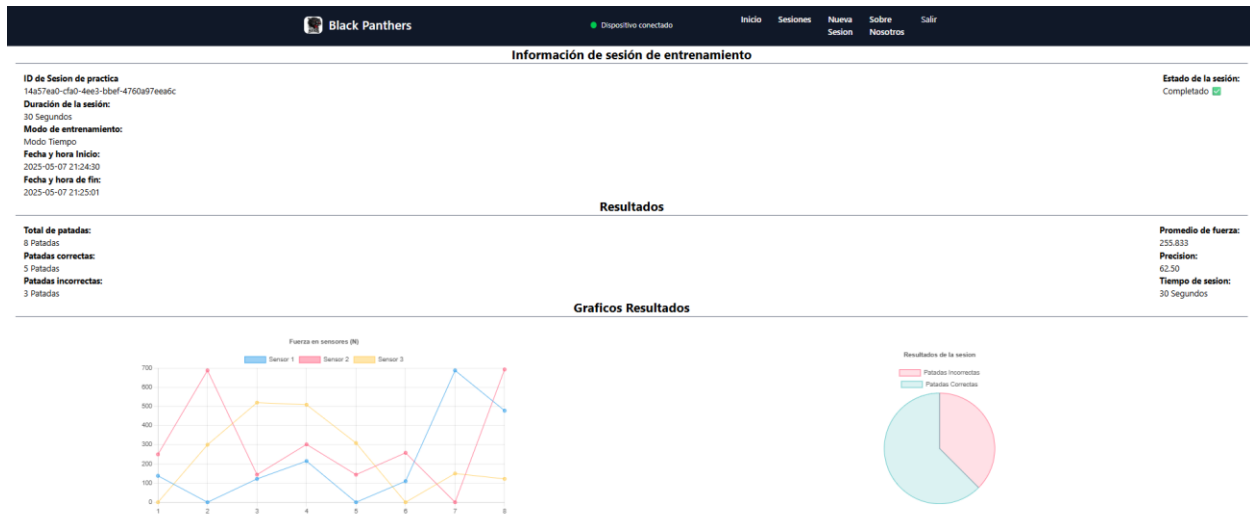
Figura 110 Monitoreo durante la sesión de entrenamiento



Al finalizar la sesión de entrenamiento, el entrenador hace la verificación de los resultados obtenidos, con apoyo en las gráficas obtenidas se puede ver como fue el desempeño del estudiante, cumpliendo los 30 segundos establecidos para la sesión.

En la Figura 111 se puede ver el reporte de los resultados obtenidos, en donde se encuentran las patadas realizadas y los valores de fuerza obtenidas en cada una de ellas, de igual forma se pueden ver cuantas patadas correctas e incorrectas se obtuvieron.

Figura 111 Resultados de la sesión de tiempo



En la Figura 112 se visualiza al entrenador haciendo el análisis correspondiente de la sesión de entrenamiento, para verificar cuales fueron los resultados de la sesión, cuantas patadas fueron correctas y cuál fue la fuerza registrada en cada una de ellas por los 5 sensores.

Figura 112 Análisis del entrenador a la sesión finalizada



Las pruebas realizadas durante las sesiones de entrenamiento a los distintos alumnos del club permitieron obtener los resultados esperados, mostrando como es el desempeño durante su pateo y con la ayuda del entrenador mejorar la postura adecuada del pie para que una patada sea considerada válida, tal como sucede en los petos electrónicos.

4.2. Encuestas dirigidas al instructor y estudiantes del club

Con el objetivo de evaluar el nivel de aceptación del sistema de entrenamiento inteligente, se realizaron dos encuestas dirigidas a los estudiantes y al director del Club de Artes Marciales Ecuadecuba. Las encuestas fueron elaboradas con base en el funcionamiento del sistema, tras la implementación de las pruebas de software y sesiones reales de entrenamiento. Las preguntas estuvieron orientadas a aspectos de funcionalidad, facilidad de uso, precisión del análisis y utilidad de las funcionalidades del sistema. Los resultados obtenidos permiten identificar los requerimientos más valorados por los usuarios y validar la efectividad del sistema en un entorno real de entrenamiento.

4.2.1. Evaluación del nivel de aceptación del sistema por parte del director/entrenador del club

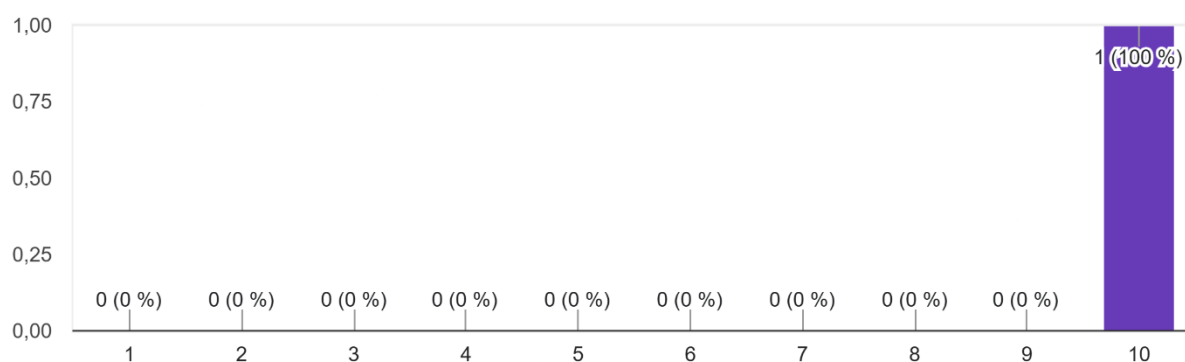
Para la realización de esta encuesta se hizo uso de la plataforma Google Forms, en donde se formularon 11 preguntas de estructura cerrada hacia el director del Club de Artes Marciales “Ecuadecuba”, Lic. Eddy Cajigal, en estas preguntas se busca evaluar del 1 al 10, siendo 10 la máxima satisfacción, los parámetros integrales del sistema, para poder verificar la aceptación de este y obtener métricas sobre la conformidad del proyecto.

Pregunta 1: Se evalúa la funcionalidad de crear nuevos usuarios dentro de la plataforma, como respuesta única se obtuvo una valoración de 10 lo que representa una satisfacción del 100% respecto a este parámetro. En la Figura 113 se visualiza la primera pregunta de la encuesta.

Figura 113 Tabulación de la primera pregunta del director (Autor)

¿Qué tan útil le resultó la funcionalidad para crear nuevos usuarios en la plataforma?

1 respuesta

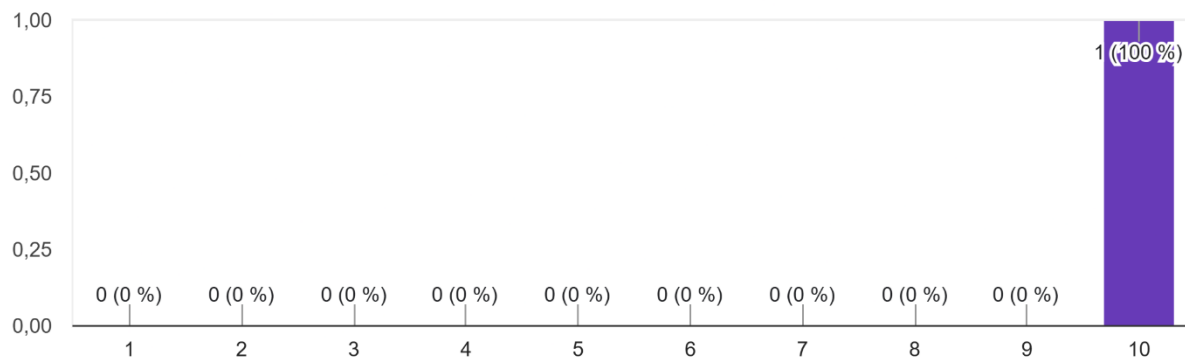


Pregunta 2: En esta pregunta se busca evaluar la facilidad y efectividad para crear nuevas sesiones de entrenamiento en el perfil de cada usuario, obteniendo una valoración del 100% de aceptación. En la Figura 114 se visualiza la respuesta a la segunda pregunta de la encuesta.

Figura 114 Tabulación de la segunda pregunta de la encuesta al director (Autor)

¿Cómo evaluaría la facilidad y efectividad del sistema para crear nuevas sesiones de entrenamiento?

1 respuesta

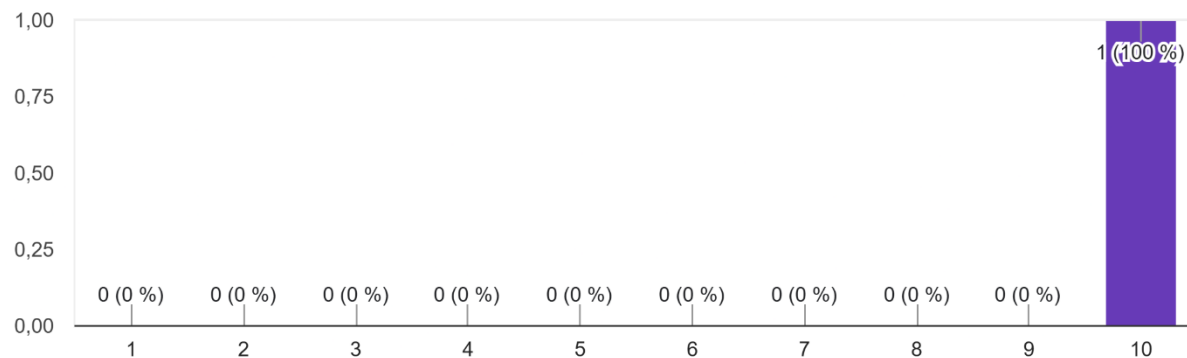


Pregunta 3: En esta pregunta se busca evaluar la facilidad y efectividad para crear nuevas sesiones de entrenamiento en el perfil de cada usuario, este parámetro se obtuvo una calificación de 10 sobre 10, lo que demuestra una aceptación de este parámetro. En la Figura 115 se observa la respuesta a la tercera pregunta de la encuesta.

Figura 115 Tabulación de la tercera pregunta de la encuesta al director (Autor)

¿Qué tan adecuada fue la opción de seleccionar el modo de entrenamiento por repeticiones o por tiempo?

1 respuesta

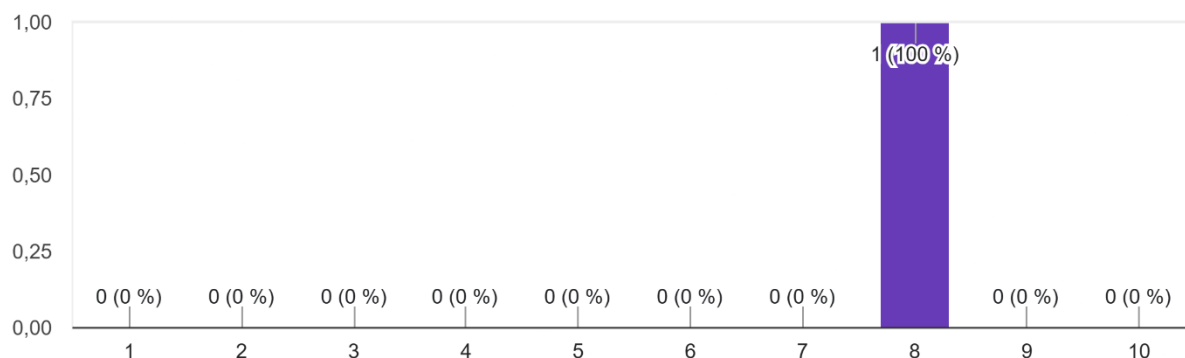


Pregunta 4: En esta pregunta se busca evaluar que tan claras y útiles fueron las alertas visuales y auditivas para el seguimiento de las sesiones, en esta pregunta se obtuvo una calificación equivalente al 80%, lo cual indica que si bien cumple con el requerimiento existen espacios de mejora pueden ser tomados en cuenta. En la Figura 116 se puede ver la respuesta obtenida en la pregunta 4.

Figura 116 Tabulación de la cuarta pregunta de la encuesta al director (Autor)

¿Qué tan claras y útiles fueron las alertas visuales y auditivas para el seguimiento de las sesiones?

1 respuesta

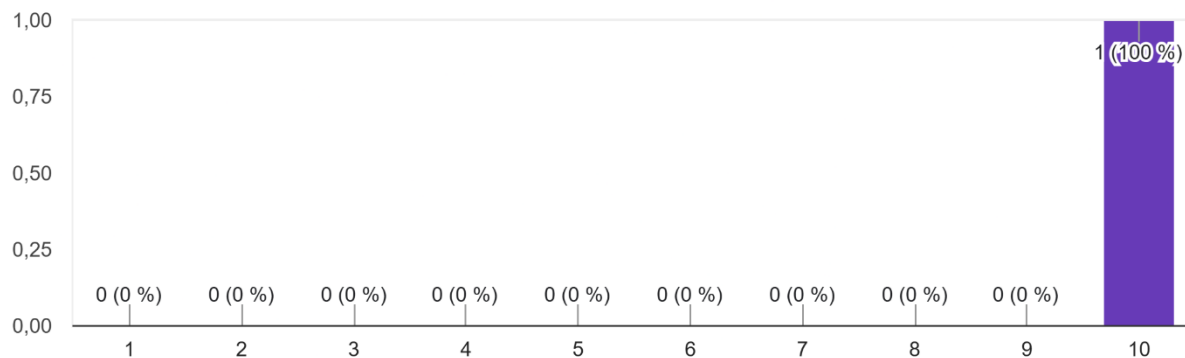


Pregunta 5: En esta pregunta se busca determinar qué tan fácil se le hizo al director acceder y entender los reportes individuales por alumnos generados por el sistema, en este punto se verifica que se obtuvo una validación del 100% por parte del director. En la Figura 117 se visualiza la respuesta del director del club.

Figura 117 Tabulación de la quinta pregunta de la encuesta al director (Autor)

¿Qué tan fácil fue acceder y entender los reportes individuales por alumno generados por el sistema?

1 respuesta

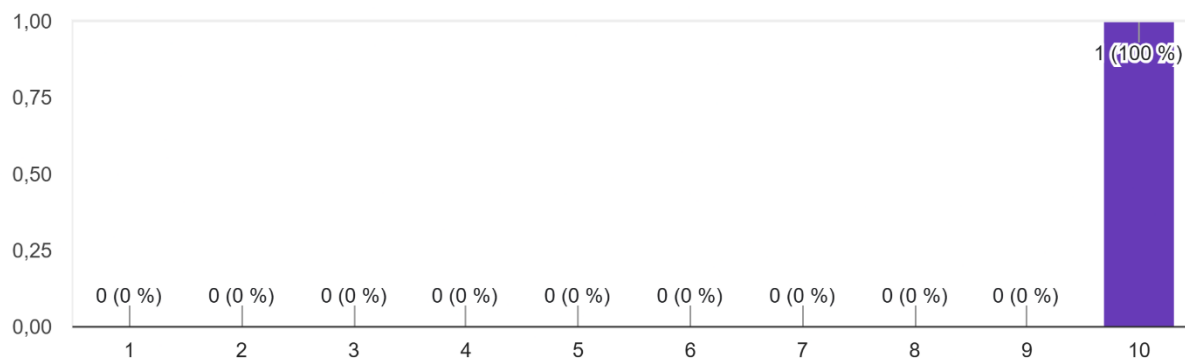


Pregunta 6: En esta pregunta se busca valorar el diseño de la interfaz de la visualización, en este parámetro se obtuvo una calificación de 10, demostrando que se está de acuerdo con la interfaz propuesta. En la Figura 118 se puede ver el resultado de la pregunta 6 de la encuesta.

Figura 118 Tabulación de la sexta pregunta de la encuesta al director (Autor)

¿Cómo valora el diseño de la interfaz de visualización?

1 respuesta

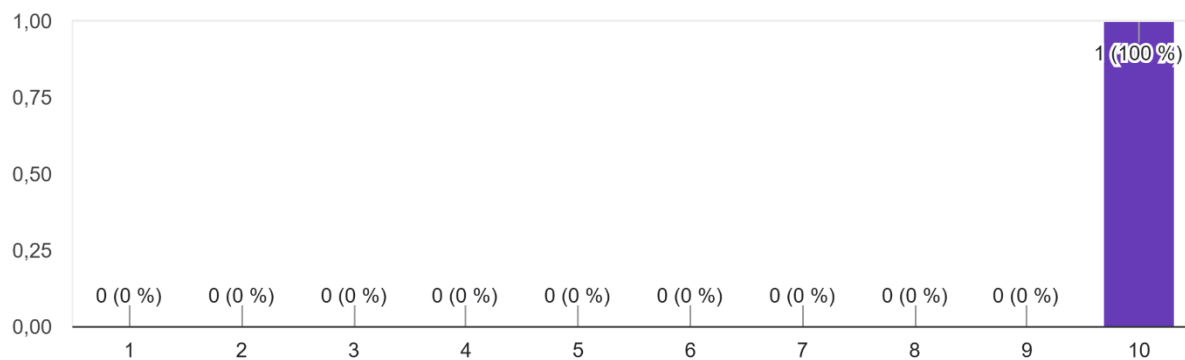


Pregunta 7: En este apartado se busca verificar que tan satisfactoria fue la retroalimentación al usuario durante o al final del entrenamiento, haciendo referencia a los datos mostrados en tiempo real, como resultado se obtuvo una satisfacción del 100%, lo cual indica que se realizó de manera correcta. En la Figura 119 se puede observar el resultado obtenido de la pregunta 7 de la encuesta.

Figura 119 Tabulación de la séptima pregunta de la encuesta al director (Autor)

¿Qué tan satisfactoria fue la retroalimentación ofrecida al usuario durante o al final del entrenamiento? (Datos mostrados en tiempo real)

1 respuesta

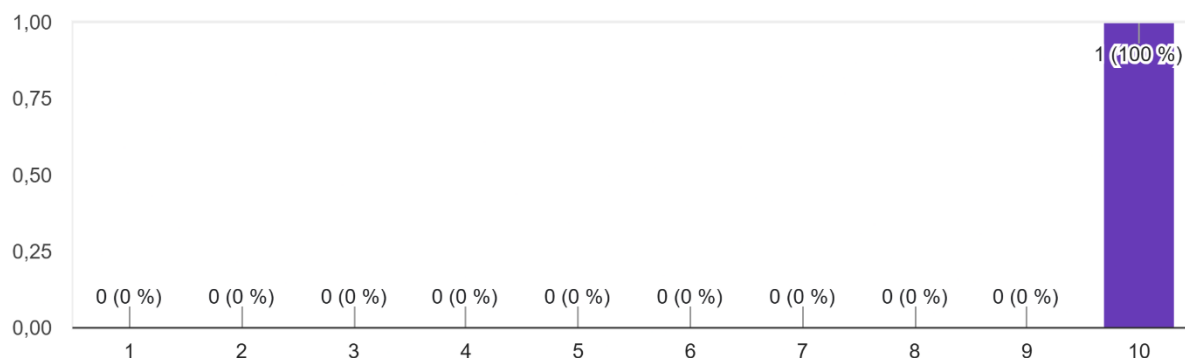


Pregunta 8: En esta pregunta se considera que tan fácil fue el uso de la interfaz en general para el director del club, en este caso se demuestra que existe un 100% de aceptación para este parámetro. En la Figura 120 se puede visualizar la respuesta número 8 de la encuesta.

Figura 120 Tabulación de la octava pregunta de la encuesta al director (Autor)

¿Cómo considera la facilidad del uso de la interfaz en general?

1 respuesta

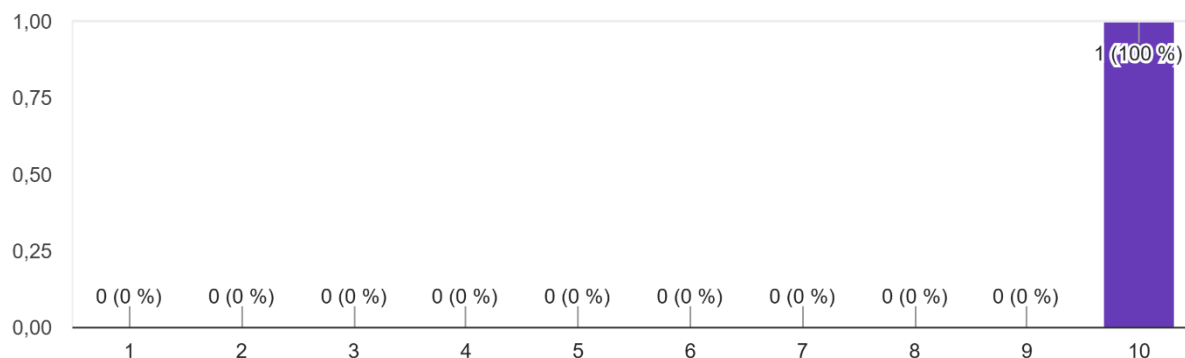


Pregunta 9: En este punto se verifica que tan útil considera el director que es el sistema para el seguimiento y mejora técnica y táctica dentro del club, para este punto el encuestado marcó 10 puntos, lo que equivale al 100% de aprobación, En la Figura 121 se visualiza la respuesta obtenida a esta pregunta.

Figura 121 Tabulación de la novena pregunta de la encuesta al director (Autor)

¿Qué tan útil considera el sistema para el seguimiento y mejora técnico y táctica dentro del club?

1 respuesta

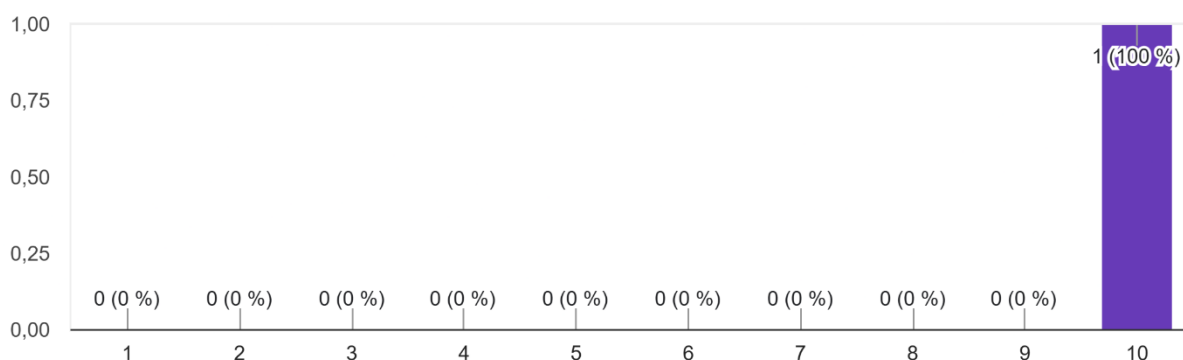


Pregunta 10: En esta pregunta se evalúa si el sistema ayudará a mejorar el desempeño para competencias con petos electrónicos en el futuro, a lo cual el director del club concordó con un 100 % que será útil en este ámbito. En la Figura 122 se muestra el resultado obtenido en la pregunta 10 de la encuesta.

Figura 122 Tabulación de la décima pregunta de la encuesta al director (Autor)

¿Considera que este sistema ayudará a mejorar el desempeño para competencias con petos electrónicos?

1 respuesta

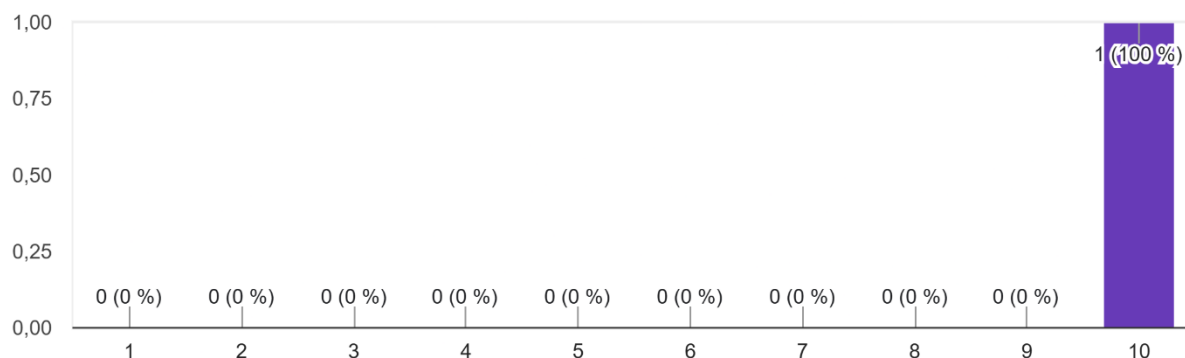


Pregunta 11: En este apartado se busca como punto final, determinar una valoración del sistema en general, el director en este punto tiene una aceptación del 100%, lo cual demuestra que se encuentra conforme con el sistema desarrollado. En la Figura 123 se visualiza la respuesta que se obtuvo en la pregunta 11.

Figura 123 Tabulación de la undécima pregunta de la encuesta al director (Autor)

Finalmente, ¿qué valoración general le da al funcionamiento completo del sistema?

1 respuesta



De acuerdo con los datos obtenidos y los resultados obtenidos por la encuesta, se pudo constatar la mejora técnica de los estudiantes al momento de realizar el pateo, ya que se considera que al conocer cómo funcionan los sensores de fuerza realizan una mejor posición de la postura técnica del pie, además, al ver con la fuerza que realizan los golpes se pudo mejorar la precisión del golpe al momento de ejecutar las patadas ya que se buscan los puntos de contacto adecuados al momento de ejecutar las técnicas.

4.2.2. Evaluación del nivel de aceptación del sistema por parte de los estudiantes del club

Para la encuesta enfocada a los estudiantes del club sobre la funcionalidad del prototipo y su uso se tomaron 15 estudiantes, los cuales fueron seleccionados de acuerdo con las distintas categorías, se eligieron 3 alumnos por cada una, para tener una variedad de practicantes adecuada para poder apreciar las características del sistema.

En la Figura 124 se puede ver el listado de nombres de los alumnos encuestados.

Figura 124 Nombres de los alumnos encuestados (Autor)

Ingrese su nombre y apellido

15 respuestas

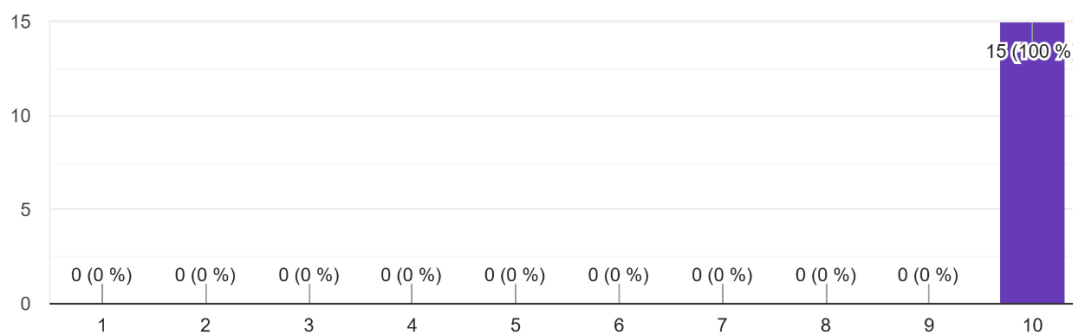
Bryan Espinosa
Josué Jumbo
Dilan Cueva
Dilan Saltos
Matías Lara
Yarely Rosero
Liam Muyulema
David Cueva
Leonel Guizado

Pregunta 1: En este apartado se busca evaluar que tan cómodo les resulto el dispositivo respecto al pateo durante la sesión, los 15 encuestados respondieron en unanimidad que fue muy cómodo el realizar el pateo, lo que indica una correcta disposición del equipo en un 100%. En la Figura 125 se puede visualizar el resultado obtenido de la encuesta.

Figura 125 Tabulación de la primera pregunta de la encuesta a los estudiantes (Autor)

¿Qué tan cómodo resultó el dispositivo para el pateo durante el entrenamiento?

15 respuestas

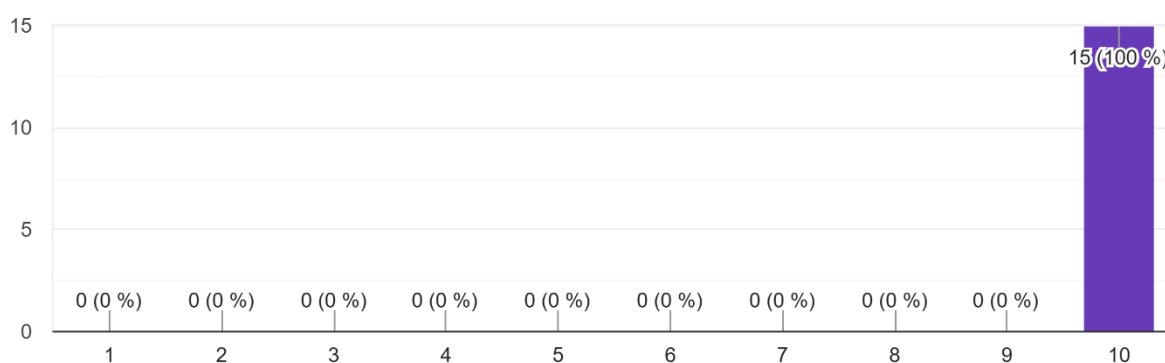


Pregunta 2: En esta parte se busca valorar si los estudiantes consideran que el sistema es compacto y no interfirió con el movimiento durante las sesiones de entrenamiento, a lo cual se obtuvo el 100% de respuestas de aceptación con respecto a este parámetro. En la Figura 126 se puede ver el resultado de la encuesta a esta pregunta.

Figura 126 Tabulación de la segunda pregunta de la encuesta a los estudiantes (Autor)

¿Considera que el sistema es compacto y no interfiere con el movimiento?

15 respuestas

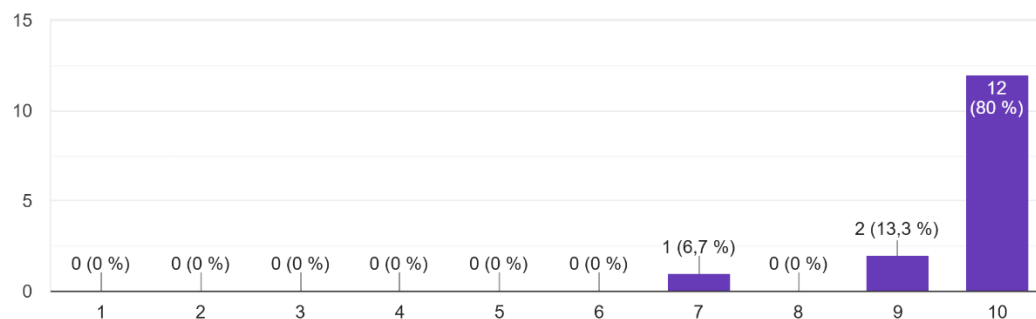


Pregunta 3: En la pregunta 3 se consulta a los encuestado que tan claras y efectivas fueron las alertas auditivas para el seguimiento de las sesiones, los encuestados en su mayoría contestaron que lo valoran con 10, pero no fue unánime, existen encuestados que respondieron que consideran que existen puntos de mejora para este criterio, por lo que se obtuvo como resultado promedio 96.7% de satisfacción. En la Figura 127 se observa la gráfica de las respuestas obtenidas.

Figura 127 Tabulación de la tercera pregunta de la encuesta a los estudiantes (Autor)

¿Qué tan claras y efectivas fueron las alertas auditivas (sonidos) para el seguimiento de las repeticiones o el tiempo?

15 respuestas

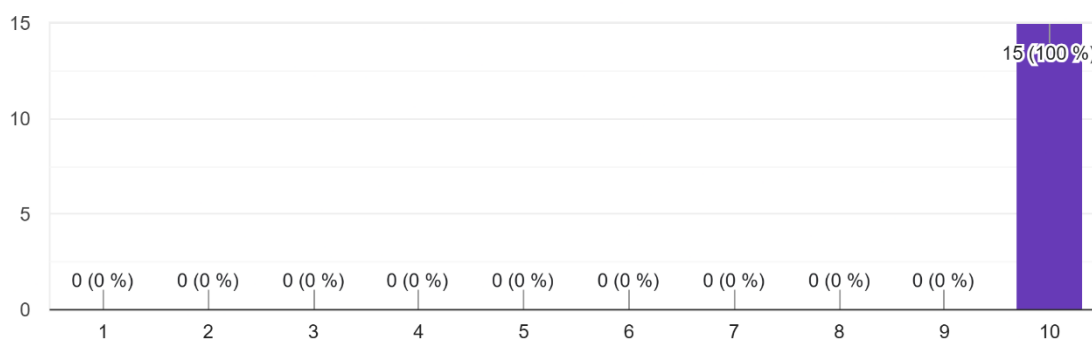


Pregunta 4: En este punto se busca preguntar a los usuarios, que tan fácil se les hizo comprender el conteo de golpes correctos e incorrectos, a lo cual todos respondieron con una valoración de 10 puntos, lo cual demuestra que el conteo de golpes fue fácil de entender para todos los encuestados con una ponderación del 100%. En la Figura 128 se puede visualizar las respuestas de los alumnos encuestados.

Figura 128 Tabulación de la cuarta pregunta de la encuesta a los estudiantes (Autor)

¿Qué tan fácil fue comprender el conteo de golpes correctos e incorrectos?

15 respuestas

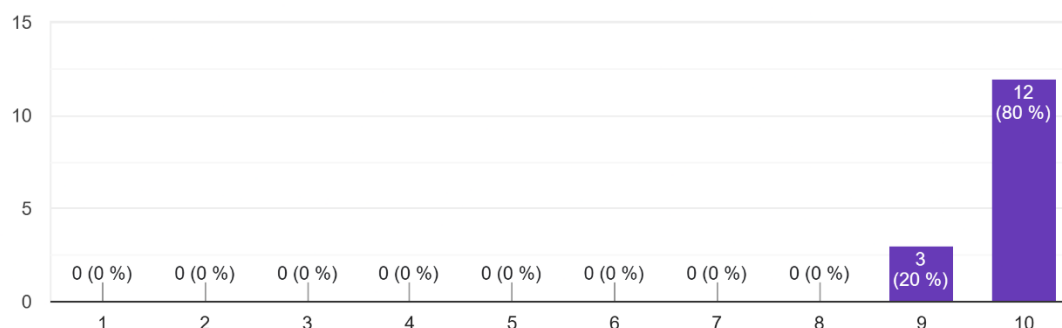


Pregunta 5: En esta pregunta se cuestiona que tan confiable consideran que identificó correctamente la patada correcta e incorrecta, si bien la mayoría de encuestados respondió que el dispositivo es muy confiable con una valoración de 10, existen encuestados que dieron una valoración de 9 puntos, lo cual demuestra que este parámetro puede ser mejor explicado durante el análisis de los reportes, pese a eso el parámetro se puede considerar como exitoso ya que tuvo una aprobación del 98%. En la Figura 129 se visualiza la ponderación de la pregunta.

Figura 129 Tabulación de la quinta pregunta de la encuesta a los estudiantes (Autor)

¿Qué tan confiable percibió el sistema para identificar correctamente una patada?

15 respuestas

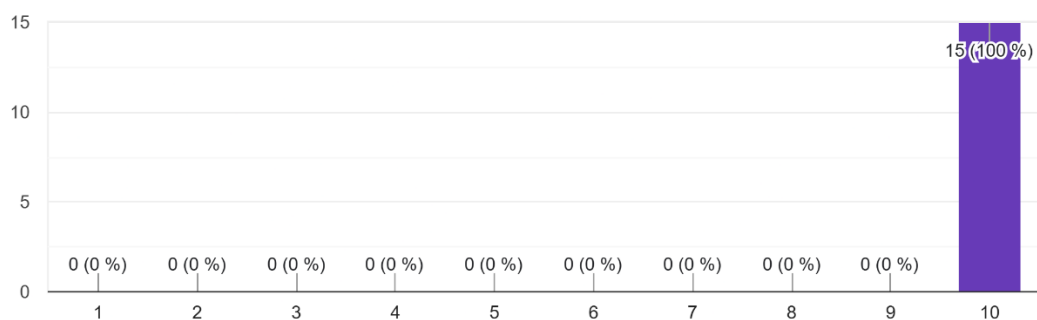


Pregunta 6: Como última pregunta, se consulta a los estudiantes del club, cómo valoran la experiencia del uso del sistema durante sus sesiones de entrenamiento, a los cual todos respondieron con una calificación de 10, lo que equivale al 100% de satisfacción general. En la Figura 130 se puede visualizar la valoración dada por los encuestados a esta pregunta.

Figura 130 Tabulación de la quinta pregunta de la encuesta a los estudiantes (Autor)

¿Cómo valoraría la experiencia completa del entrenamiento con este sistema?

15 respuestas



Los estudiantes después de haber realizado las técnicas de pateo con el prototipo demostraron de manera clara que mejoraron la técnica al momento de ejecutar las patadas, esto es visible al momento de validar los resultados en las competencias, ya que su desempeño ha tenido mejoras considerables, algunos de los estudiantes que hicieron uso del proyecto lograron obtener medallas a nivel nacional e internacional, demostrando su mejoría en el marcaje de puntos.

V. Conclusiones

Mediante la revisión bibliográfica y de los requerimientos obtenidos mediante encuestas y entrevistas con el director y los estudiantes del Club Ecuadecuba, se identificaron las necesidades de stakeholders, sistema y arquitectura para que el sistema permita mejorar el proceso de entrenamiento mediante el monitoreo y análisis de las patadas realizadas durante una sesión de entrenamiento de Taekwondo. Este punto fue clave para poder determinar el diseño físico y lógico del dispositivo, así también de las funcionalidades del prototipo.

Se establecieron los requerimientos de hardware y software necesarios para realizar la construcción del sistema. Para el hardware se hizo uso de sensores de fuerza resistiva (402), como microcontrolador fue seleccionado el Arduino Lilypad para el circuito de recopilación de datos, para la comunicación entre el backend y el prototipo fue usado el módulo ESP32, el cual usa el protocolo Wi-Fi a 2.4 GHz para buscar un mejor rango de distancia de operación. Para el software, se desarrolló el backend en el lenguaje Python con el uso de la librería FastAPI, para la gestión de sesiones, usuarios y almacenamiento de datos, para el frontend se hizo uso de React con Tailwind CSS que permite visualizar el desempeño del deportista en tiempo real y por historial.

Se diseñó un sistema electrónico en una disposición triangular centrada dentro del pateador con 3 sensores FSR, ubicada adecuadamente para detectar la distribución de la fuerza de cada patada. La disposición y calibración de los sensores permitió establecer un umbral de detección adecuado para la validación de las patadas mediante un algoritmo KNN, lo que facilitó y automatizó la categorización de golpes como correctos e incorrectos con base en los valores de fuerza de cada sensor.

Se incorporaron funcionalidades de control como la creación de sesiones con valores de repeticiones o por una cantidad determinada de segundos, la gestión de usuarios mediante la plataforma interactiva y la transmisión continua de datos y envío de comandos desde el frontend hacia el módulo ESP32. De igual manera, se implementó un sistema de alertas visuales (LEDs) y auditivas (buzzer) que permiten al estudiante y al entrenador verificar el estado actual del prototipo durante la sesión de entrenamiento.

Se realizaron pruebas funcionales con estudiantes y con el director del club, en las cuales se comprobó la correcta ejecución de todos los requerimientos estipulados por stakeholders, sistema y arquitectura. Las encuestas aplicadas mostraron una satisfacción muy alta de aceptación y utilidad tanto para el director del club como los estudiantes involucrados en las pruebas, el promedio de aceptación en general fue mayor al 90%.

VI. Recomendaciones

Para garantizar el correcto funcionamiento del sistema, se recomienda al entrenador que mantenga el dispositivo cargado completamente antes de cada jornada de entrenamiento, así como verificar la conexión del ESP32 a la red WiFi con la dirección IP 192.168.137.1.

Se debe tener un cuidado especial con las partes sensibles del dispositivo (parte superior) para evitar fracturas en los microcontroladores o en los cables de conexión, de igual manera se debe tomar en cuenta que un entrenador supervise en todo momento el uso adecuado del prototipo para que este no tenga problemas en su funcionamiento.

En la configuración de sesiones se sugiere a los entrenadores colocar la cantidad de tiempo o repeticiones adecuadas a la capacidad específica que se busca desarrollar en el estudiante, para que las mediciones que se hagan a nivel histórico se vean reflejadas de manera adecuada.

Se recomienda que antes de iniciar la sesión se constate mediante las alertas visuales y la interfaz gráfica que el dispositivo se encuentre conectado y listo para ser usado, garantizando la correcta recepción y envío de datos entre el backend y el ESP32.

Se sugiere mantener actualizados los datos de los usuarios en el sistema, de manera que los reportes muestren información real de la situación actual del estudiante y de esta manera el entrenador pueda hacer su análisis de manera más objetiva y colocar procesos de mejora adecuados.

Para mejora del sistema se recomienda tener un módulo de análisis más avanzado que permita calcular automáticamente las tendencias de mejora en fuerza y precisión con inteligencia artificial. Asimismo, se sugiere incorporar un asistente de entrenamiento virtual, que brinde

sugerencias de rutinas de acuerdo a la edad, cinturón y nivel del estudiante, buscando mejorar capacidades específicas.

Como punto final, se recomienda que el frontend realice directamente un análisis comparativo entre sesiones similares, en donde se busque similitudes entre sesiones pasadas y actuales para que brinde al entrenador un análisis comparativo real del avance del estudiante.

VII. Referencias

Academia de Taekwondo OCRAM (2018). Algunas patadas básicas
<https://www.ocramtaekwondo.com/2018/08/31/algunas-patadas-basicas/>

Antolínez Díaz, R. O., & Ruda Cleves, J. L. (2014, febrero 20). Almacenamiento en la nube. <http://repository.unipiloto.edu.co/handle/20.500.12277/2969>

Arduino (2024). LilyPad Arduino USB. Arduino.cc.
<https://docs.arduino.cc/retired/boards/LilyPad-arduino-usb/>

Arias, Y. (2018). REGLAMENTO E INTERPRETACIÓN DE ARBITRAJE EN COMBATE DE TAEKWONDO. https://mastkd.com/wp-content/uploads/2018/09/Kyorugi-01_ES_REGLAMENTO-E-INTERPRETACION-DE-ARBITRAJE-EN-COMBATE-DE-TAEKWONDO-WT.pdf

Álvarez, M. (2017). Desarrollo de una mejora en la metodología V en la fase de pruebas de software implementado en el campo de la aeronáutica (tesis).
<https://ciateq.repositorioinstitucional.mx/>. Recuperado de:
<http://ciateq.repositorioinstitucional.mx/jspui/handle/1020/125>.

Asta Alarcón, J. (2017). CLOUD STORAGE.
<https://www.lawebdelprogramador.com/pdf/1939-CLOUD-STORAGE.html>

Bobadilla, J. (2020). Machine Learning y Deep Learning: Usando Python, Scikit y Keras: (1 ed.). RA-MA Editorial. <https://elibro.net/es/lc/utnorte/titulos/222698>

Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: Toward wearable engineering for everyone. IEEE Xplore. <https://ieeexplore.ieee.org/document/4487082>

Celaya Luna, A. (2014). Cloud: herramientas para trabajar en la nube: (ed.). Editorial ICB. <https://elibro.net/es/lc/utnorte/titulos/56046>

Cervantes Gómez, J., & Gómez Fuentes, M. (2020). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. Universidades, 62(52), 37-47. Recuperado a partir de <http://udualerreu.org/index.php/universidades/article/view/183>

De La Vega Saransig, D.D., Castillo López, A.P., y Loachamin Aldaz, E.M. (2021). Diferencias biomecánicas en el taekwondo entre la categoría clasificado y novato realizando la patada mondolyo furyo chagui. Lecturas: Educación Física y Deportes, 26(276), 75-89. <https://doi.org/10.46642/efd.v26i276.2918>

De Rufino, P. (2017). TAEKWONDO: ORIGEN Y EVOLUCIÓN (tesis). <https://buleria.unileon.es>. Retrieved from https://buleria.unileon.es/bitstream/handle/10612/7963/RUFINO%20A%D1ORBE_PEDRO%20DE_JULIO_2017.pdf?sequence=1.

Dellal, A., da Silva, C. D., Hill-Haas, S., Wong, delP., Natali, A. J., De Lima, J. R., Bara Filho, M. G., Marins, J. J., Garcia, E. S., & Chamari, K. (2012). Heart rate monitoring in soccer: interest and limits during competitive match play and training, practical application. Journal of strength and conditioning research, 26(10), 2890–2906. <https://doi.org/10.1519/JSC.0b013e3182429ac7>

Díaz, J. E. B. & Barrientos Varela, M. (2020). Taekwondo: técnica, táctica, estrategia, competición: (ed.). Editorial Paidotribo. <https://elibro.net/es/lc/utnorte/titulos/161645>

Estevan I, Jandacka D, Falco C. Effect of stance position on kick performance in taekwondo. Journal of sports sciences. 2013;31(16):1815-22.

Fava, L. A., Gastón, D., Ferraresso, A., Boccari, E., & Díaz, F. J. (2020). Inteligencia y tecnologías aplicadas al deporte de alto rendimiento. Unlp.edu.ar. <http://sedici.unlp.edu.ar/handle/10915/104242>

Fava, L. A., Vilches Antão, D. G., Díaz, F. J., Pagano, M., & Romero Dapozo, J. R. (2018, July 13). Tecnología Aplicada Al Deporte de Alto Rendimiento. SEDICI. <http://sedici.unlp.edu.ar/handle/10915/68039>

Fernández-Echeverría, C., Mesquita, I., Conejero, M., & Moreno, M. P. (2019). Importancia de una herramienta tecnológica en la gestión de información en el deporte. Percepción del staff técnico de un equipo de voleibol de alto nivel. Revista Iberoamericana de Ciencias de La Actividad Física y El Deporte, 7(3), 57. <https://doi.org/10.24310/riccafd.2018.v7i3.5541>

Granda, C.A., Loachamin, E.M., y Arla, S.M. (2016). Biomechanical analysis of the Taekwondo Ap-Chagui kick. Lecturas: Educación Física y Deportes, 21(218), 1-7. <https://www.efdeportes.com/efd218/biomechanical-analysis-of-the-taekwondo.htm>

Guillén, L., Herrera, A., & Yaxel, A. de la R. (2018). Las herramientas tecnológicas TIC's como elemento alternativo para el desarrollo del componente físico. Dialnet, 34, 222-229. <https://dialnet.unirioja.es/descarga/articulo/6736351.pdf>

J. E. Martínez Guirao, Una etnografía de las artes marciales: procesos de cambio y adaptación cultural en el taekwondo. Alicante: ECU, 2013. [En Línea] Disponible en: <https://elibro.net/es/lc/utnorte/titulos/62372>

León Yas, M. A. & Rojas Ordoñez, J. D. (2018). Aplicación móvil Gps con conexión a red inalámbrica redundante para la localización de animales terrestres medianos y grandes vía rest. Recuperado de: <http://hdl.handle.net/11349/14202>.

Loachamín, E., Mena, F., Durán, E., & Maquiera, G. (2017). Diferencias biomecánicas en la patada Ap Chagüi entre taekwondocas de cinturón blanco y negro. Revista Cubana de Investigaciones Biomédicas, 36(2), 159–168. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03002017000200013

Monterrosa, A., Díaz, A., y Jiménez, C.D. (2008). Análisis biomecánico de la patada bandal chagui del arte marcial taekwondo. Lecturas: Educación Física y Deportes, 13(121), 1-5. <https://www.efdeportes.com/efd121/analisis-biomecanico-de-la-patada-bandal-chagui-taekwondo.htm>

Morales, J. (2010). REGLAMENTO PARA EL ARBITRAJE DE TAEKWONDO. <https://www.fmtaekwondo.es/wp-content/uploads/2012/09/Reglamento-Arbitraje.pdf>

Torres, I. (2010). Comparación de la eficiencia de la patada circular al pecho y a la cara en taekwondo según la distancia de ejecución. International Journal of Sport Science, 269-279. <https://doi.org/10.5232/ricyde2010.02102>

World Taekwondo. (2023). VISION MISSION, STRATEGY. Worldtaekwondo.org.
<http://www.worldtaekwondo.org/about-wt/about.html>

VIII. Anexos

Anexo 1 (Encuesta dirigida al director del club para determinar la cantidad de alumnos que tienen más de 6 meses de formación)

The image shows a screenshot of a Google Forms survey. At the top, it says 'No se pueden editar las respuestas'. The title of the survey is 'Encuesta cantidad de alumnos Club Ecuadcuba'. Below the title, there is a red asterisk indicating that the question is mandatory. The first question is 'Ingrese su nombre *' with the answer 'Eddy Cajigal Kindelan'. The second question is 'Ingrese la cantidad de estudiantes que tienen más de 6 meses en el Club Ecuadcuba *' with the answer '40'.

Anexo 2 (Encuesta dirigida a los alumnos del Club Ecuadcuba)

Levantamiento de información para los requerimientos de los estudiantes del Club Ecuadcuba

1. ¿Considera usted fundamental que el sistema de entrenamiento cuente con una interfaz intuitiva y fácil de usar?

a. Sí

b. No

2. ¿Le resultaría útil la representación visual de sus patadas a través de un mapa de constelaciones?

a. Sí

b. No

3. ¿Considera necesario que el dispositivo tenga modo de repeticiones y el modo de tiempo según sus necesidades?

a. Sí

b. No

4. ¿Cree usted que recibir retroalimentación sobre su desempeño contribuirá a mejorar su técnica?

a. Sí

b. No

5. ¿Considera útil disponer de una representación gráfica de sus resultados para monitorear su progreso?

a. Sí

b. No

6. ¿Es relevante para usted que el sistema utilice tecnología moderna para evaluar de forma objetiva su rendimiento?

a. Sí

b. No

7. ¿Le gustaría que el sistema le ofreciera un perfil para sus sesiones de entrenamiento?

a. Sí

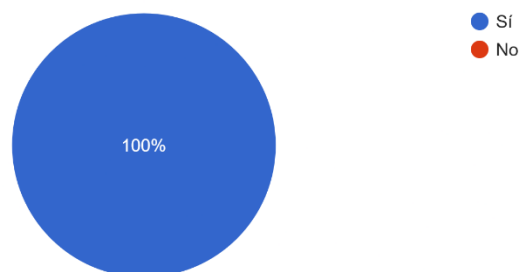
b. No

Anexo 3 (Resultados y análisis de encuesta para estudiantes Anexo 2)

Pregunta 1: En los resultados de la pregunta 1 de la encuesta dirigida a los estudiantes se obtuvo como opción unánime que el 100% de los estudiantes quisieran una interfaz amigable y fácil de usar para el sistema de entrenamiento, por lo que se considera como prioridad alta cumplir este requerimiento.

¿Considera usted que la interfaz del sistema debe ser amigable y fácil de usar?

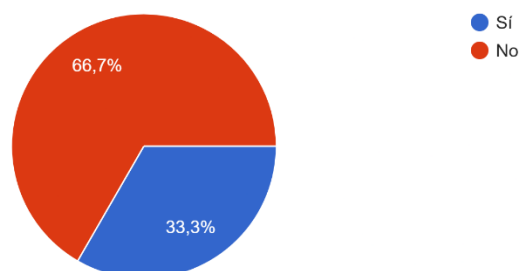
24 respuestas



Pregunta 2: En los resultados de la pregunta 2 se obtuvo que el 66.67% de los estudiantes no considera útil tener una representación visual de las patadas mediante el uso de un mapa de constelaciones, por lo que al tener una mayoría considerable que no identifica como importante este requerimiento se considera con una prioridad baja.

¿Le resultaría útil la representación visual de sus patadas a través de un mapa de constelaciones?

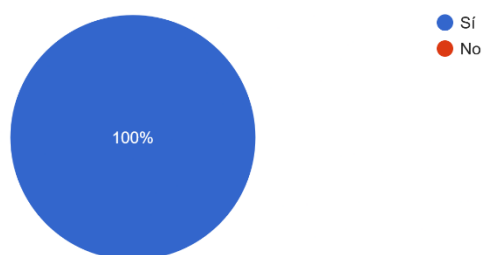
24 respuestas



Pregunta 3: Para la pregunta 3 se visualiza que el 100% de los encuestados considera como necesario el que el dispositivo tenga el modo de repeticiones y el modo de tiempo, por lo que para este parámetro a ser considerado se lo cataloga que tenga una prioridad alta.

¿Considera necesario que el dispositivo tenga modo de repeticiones y el modo de tiempo según sus necesidades?

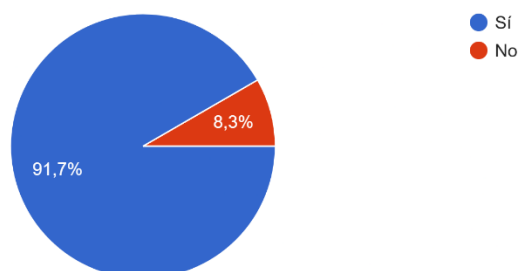
24 respuestas



Pregunta 4: Los estudiantes del club consideran como importante recibir retroalimentación sobre el desempeño realizado en la sesión de entrenamiento con un 91.7% a favor, por lo que este parámetro se le asigna con una importancia media que debe cumplir el sistema.

¿Cree usted que recibir retroalimentación sobre su desempeño contribuirá a mejorar su técnica?

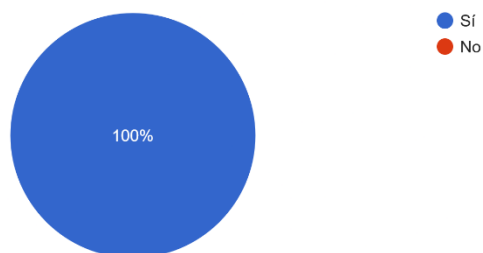
24 respuestas



Pregunta 5: En la pregunta 5 se denota una tendencia positiva al considerar como útil que una representación gráfica de los resultados obtenidos de la sesión de entrenamiento, ya que el 100% de los encuestados dijo que sí a esta pregunta y se le asigna como prioridad alta en los requerimientos.

¿Considera útil disponer de una representación gráfica de sus resultados para monitorear su progreso?

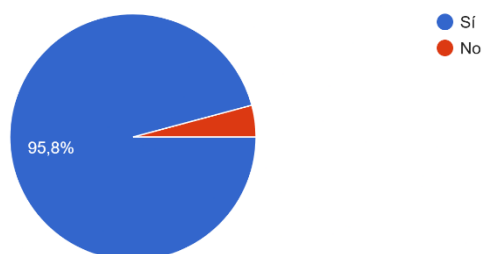
24 respuestas



Pregunta 6: Para los alumnos del club es relevante que el sistema utilice tecnología moderna para evaluar su rendimiento durante las sesiones, por lo que el 95.8% contestó que sí es importante el contar con este requerimiento, por lo que se le asigna una importancia alta.

¿Es relevante para usted que el sistema utilice tecnología moderna para evaluar de forma objetiva su rendimiento?

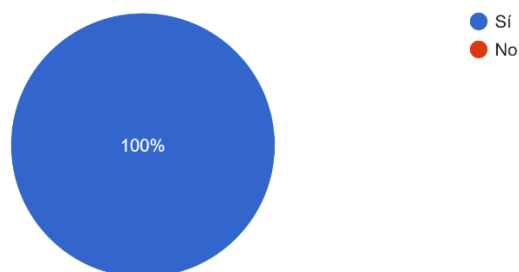
24 respuestas



Pregunta 7: En la pregunta 7 se obtiene que el 100% de los encuestados cree que es necesario tener un perfil para las sesiones de entrenamiento y que sea individual para cada uno, por lo que este requerimiento es marcado como prioridad alta.

¿Le gustaría que el sistema le ofreciera un perfil para sus sesiones de entrenamiento?

24 respuestas



Anexo 4 (Encuesta dirigida al director del Club Ecuadcuba)

1. ¿Considera usted que la implementación de un sistema de entrenamiento inteligente puede mejorar la calidad y eficiencia de los entrenamientos en el club?

a. Muy de acuerdo

b. De acuerdo

c. En desacuerdo

2. ¿Le resulta relevante que el sistema sea compacto y se integre fácilmente en las instalaciones actuales del club?

a. Muy importante

b. Importante

c. Poco importante

3. ¿Es importante para usted que el sistema proporcione datos objetivos sobre el desempeño de los estudiantes?

- a. Muy relevante
- b. Relevante
- c. No relevante

4. ¿Considera necesario que el sistema genere reportes detallados que faciliten la supervisión y la toma de decisiones?

- a. Muy importante
- b. Importante
- c. No es necesario

5. ¿Qué tan importante es para usted disponer de mecanismos de retroalimentación para corregir las técnicas de los atletas?

- a. Muy importante
- b. Importante
- c. Poco importante

6. ¿Considera importante que el sistema brinde la opción de trabajar en modo de repeticiones y en modo de tiempo contrarreloj?

a. Sí, definitivamente

b. Sí, probablemente

c. No

Anexo 5 (Resultados y análisis de encuesta para el director Anexo 4)

Pregunta 1: Para la pregunta 1 se obtuvo como respuesta que el director del club se encuentra muy de acuerdo con que se implemente un sistema de entrenamiento inteligente para mejorar la calidad y eficiencia de los entrenamientos de sus estudiantes.

¿Considera usted que la implementación de un sistema de entrenamiento inteligente puede mejorar la calidad y eficiencia de los entrenamientos en el club?

1 respuesta



Pregunta 2: El director del club considera muy importante que el sistema a implementarse sea compacto y pueda integrarse fácilmente dentro de las instalaciones actuales del club y del entorno de entrenamiento.

¿Le resulta relevante que el sistema sea compacto y se integre fácilmente en las instalaciones actuales del club?

1 respuesta



Pregunta 3: En la pregunta 3 se obtuvo que se considera como muy importante que el sistema proporcione datos objetivos sobre el desempeño de los estudiantes.

¿Es importante para usted que el sistema proporcione datos objetivos sobre el desempeño de los estudiantes?

1 respuesta



Pregunta 4: Para la pregunta 4, el director del club considera como necesario que el sistema genere reportes detallados que faciliten la supervisión y que esto ayude a la toma de decisiones propias del entrenador.

¿Considera necesario que el sistema genere reportes detallados que faciliten la supervisión y la toma de decisiones?

1 respuesta



Pregunta 5: El director del club considera importante que el sistema brinde mecanismo de retroalimentación, ya que menciona que el trabajo directo del entrenador es brindar ese tipo de mecanismos.

¿Qué tan importante es para usted disponer de mecanismos de retroalimentación para corregir las técnicas de los atletas?

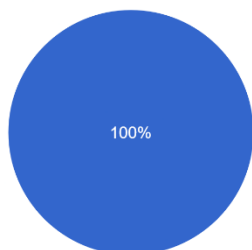
1 respuesta



Pregunta 6: Para la pregunta 6 el encuestado contesta que es muy importante que el sistema de entrenamiento brinde la opción de elegir el modo de trabajo, sea por tiempo en contrarreloj o por repeticiones definidas.

¿Considera importante que el sistema brinde la opción de trabajar en modo de repeticiones y en modo de tiempo contrareloj?

1 respuesta

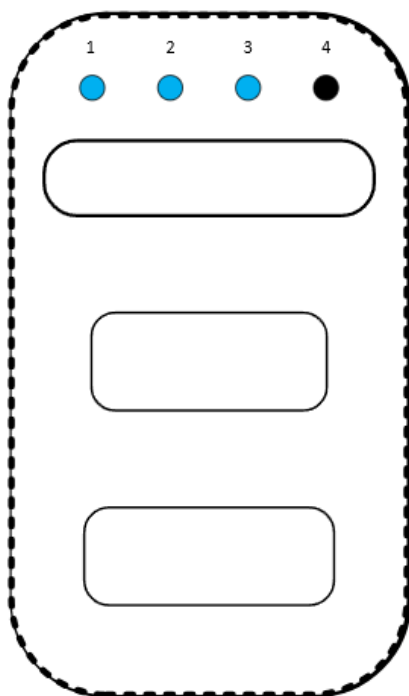


- Sí, definitivamente
- Sí, probablemente
- No

Anexo 6 (Layout del prototipo)

Para una correcta denotación del funcionamiento de las alertas visuales y auditivas, se presenta la vista frontal del prototipo para indicar el funcionamiento de cada uno de los LEDs y el zumbador o buzzer

Vista trasera



SIMBOLOGÍA

- 1: LED de conexión a la red local
- 2: LED de encendido
- 3: LED de sesión activa
- 4: Zumbador

De igual manera se presenta la vista inferior en donde se denotan el botón de encendido y apagado y la ubicación de puerto USB C para la carga del dispositivo.

Vista inferior



SIMBOLOGÍA

5: Botón de encendido/apagado

6: Puerto USB C para carga