

EVALUACIÓN DE GESTOS TÉCNICOS MOTRICES CON BASE EN LA COORDINACIÓN ÓCULO-MANUAL U ÓCULO-PÉDICA A TRAVÉS DE UN SISTEMA IoT – WBAN



Trabajo de Grado, Requisito para optar al Título
de Ingeniería Electrónica y Telecomunicaciones

Miguel Santiago Catamusca Ordoñez

Mario Camilo Burbano Agredo

Director: Dr. Marlon Felipe Burbano Fernández

Codirector: Dr. Gustavo Ramírez González

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES

PROGRAMA DE INGENIERÍA ELECTRONICA Y TELECOMUNICACIONES

POPAYÁN, 2024

AGRADECIMIENTOS

A Dios padre, por iluminarnos en todo momento de nuestra trayectoria universitaria, por cada lección aprendida y por cada una de sus bendiciones en nuestra vida personal y profesional.

A nuestros padres, hermanos y familiares, por su comprensión y apoyo incondicional desde el inicio y hasta el final de nuestra carrera.

A nuestro director Marlon Felipe Burbano. por ser una luz en nuestro camino, por compartir sus conocimientos y experiencias.

A nuestro Co-Director Gustavo Ramírez González por brindarnos sus conocimientos, por darnos la oportunidad de realizar este trabajo.

A nuestros docentes, por compartir su conocimientos y experiencias, reconocemos su arduo trabajo, su misión de formar mejores profesionales y mejores seres humanos.

A nuestros compañeros y amigos, por su compañía, consejos y momentos compartidos en este proceso de nuestras vidas.

A nuestra alma mater, por darnos la oportunidad de formarnos como profesionales íntegros, dignos y capaces de ofrecer lo mejor de nosotros, en el progreso de nuestra sociedad.

CONTENIDO

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA	1
1.1 DEFINICIÓN DEL PROBLEMA.....	1
1.2. OBJETIVOS DE LA INVESTIGACIÓN	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
1.3. ALCANCE DE LA INVESTIGACIÓN	3
1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	4
CAPÍTULO 2: ANÁLISIS BIBLIOMÉTRICO.....	7
2.1 CONCEPTOS GENERALES.....	7
2.1.1 Entornos de desarrollo	7
2.1.1.1 IDE Arduino.....	7
2.1.1.2 Visual Studio Code.....	8
2.1.1.3 Jupyter Notebook.....	8
2.1.1.4 Android Studio	9
2.1.2 Protocolos HTTP	9
2.1.3 Wearable.....	10
2.1.4 WiFi.....	10
2.1.5 ESP-32.....	11
2.1.6 Lenguajes De Programación.....	11
2.1.6.1 C++.....	11
2.1.6.2 Python.....	12
2.1.6.3 Java.....	13
2.1.6.7 PHP	13
2.1.7 Librerías	14
2.1.8 Sensores Inerciales	15
2.1.9 Energía Consumida	15

2.1.10 Batería	16
2.1.10.1 Batería tipo botón	16
2.1.10.2 Batería Multipack	16
2.1.10.3 Batería de litio	16
2.1.11 Módulo FastDTW-PYTHON	17
2.1.12 XAMPP	17
2.1.13 Procesamiento Digital de Señales (PDS).....	18
2.1.14 Series de Tiempo	18
2.1.15 Dynamic Time Warping (DTW).....	20
2.2 TRABAJOS RELACIONADOS	22
2.3 ARTÍCULOS SELECCIONADOS	23
CAPÍTULO 3: DESARROLLO DE ARQUITECTURA	28
3.1 ARQUITECTURA CON BASE EN EL MODELO 4+1.....	28
3.2 PLANTEAMIENTO INICIAL DE LA SOLUCIÓN	35
CAPÍTULO 4: DESARROLLO WEARABLE, APLICACIÓN MÓVIL Y SERVIDOR	36
4.1 DEFINICIÓN DE REQUISITOS	36
4.2 REVISIÓN DEL ESTADO DEL ARTE	37
4.3 ELEMENTOS SELECCIONADOS.....	38
4.4 DISPOSITIVO WEARABLE	39
4.4.1 Estructura Hardware del Dispositivo Wearable.....	40
4.4.1.1 Carcasa del dispositivo	40
4.4.2 Pre procesamiento en el Módulo Secundario Wearable.....	42
CAPÍTULO 5: ALGORITMO PARA COMPARAR MOVIMIENTOS	69
5.1. DESARROLLO	69
5.1.1 Parámetros Generales	69
5.1.2 Calibración del Dispositivo	69
5.1.3 Filtro complementario	70

5.1.4 Análisis por DTW.....	71
5.1.5. Análisis comparativo a partir de DTW	74
5.1.6. Diagrama de Flujo	75
CAPÍTULO 6: CASO DE ESTUDIO	77
6.1 PLANTEAMIENTO DEL EXPERIMENTO.....	77
6.2 CREACIÓN DEL SET DE DATOS	77
6.3 RESULTADOS DEL ALGORITMO.....	80
CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS.	93
7.1 CONCLUSIONES	93
7.2 TRABAJOS FUTUROS.....	94
REFERENCIAS	95

LISTA DE FIGURA

	Pág.
Figura 1. Artículos por Periodo.	4
Figura 2. Diagrama Estratégico del año 2022.....	5
Figura 3. Clúster Internet Of Things.....	6
Figura 4. Componentes de una Serie de Tiempo.	19
Figura 5. Ajuste de Ventana.....	21
Figura 6. Componentes del sistema.	28
Figura 7. Arquitectura de Alto Nivel.	30
Figura 8. Casos de uso del sistema.	31
Figura 9. Diagrama de secuencia.....	32
Figura 10. Diagrama de clases del sistema.	32
Figura 11. Vista del proceso.....	33
Figura 12: Diagrama de actividades.	34
Figura 13. Diagrama de Implementación.	34
Figura 14. Funcionalidad.....	40
Figura 15. Estructura a nivel hardware del wearable.....	41
Figura 16. Esquema de conexión de la ESP32.....	41
Figura 17. Wearable Ubicado en pantorrilla.....	41
Figura 18. Componentes dispositivo wearable.	44
Figura 19. Librerías Requeridas.	44
Figura 20. Componentes Requeridos.....	45
Figura 21. Credenciales Requeridas.	45
Figura 22. Listado de almacenamiento.	46
Figura 23. Captura de Datos.....	46
Figura 24. Función Setup.	47
Figura 25. Resto de Configuración.....	47
Figura 26. División de Búsquedas.	48
Figura 27. Offset.	49
Figura 28. Librerías.	49
Figura 29. Aspectos adicionales.....	49
Figura 30. Asignaciones dadas.	51
Figura 31. Funciones Adicionales.	52

Figura 32. Estructura del Proyecto APP Móvil.....	55
Figura 33. Funciones de Evaluación.....	56
Figura 34. Variables.	56
Figura 35. Dispositivo Wearable.	57
Figura 36. Preferencias.....	57
Figura 37. Caso TRY de Espacios.....	58
Figura 38. Request agregado.	59
Figura 39. Variable Media Player de Alarma.	59
Figura 40. Botón de Registro y Caso.....	60
Figura 41. Wearable registrado.	60
Figura 42. Adicional.....	60
Figura 43. Proceso de Registro de Dispositivo.	61
Figura 44. Nombre de red y contraseña.	61
Figura 45. Mensaje de no Registro.	62
Figura 46. Interfaces de la APP Móvil.	62
Figura 47. Variable que crea la Conexión.	64
Figura 48. Ejemplo de Matriz de Costos.	72
Figura 49. Ejemplo DTW – señales reales.....	73
Figura 50. Ejemplo DTW – comparación de señales.....	73
Figura 51. Ejemplo dtw - señales ideales.	74
Figura 52. Diagrama de flujo del prototipo diseñado.	76
Figura 53. distancia entre el balón y el arco.	78
Figura 54. series temporales sobre el eje X tiro penal a la derecha.....	83
Figura 55. Series temporales sobre eje Y tiro penal a la derecha.....	83
Figura 56 Series temporales sobre el eje X tiro penal al centro.	84
Figura 57 Series temporales sobre el eje Y tiro penal al centro.	85
Figura 58. Series temporales sobre el eje X tiro penal a izquierda.	86
Figura 59. Series temporales sobre el eje Y tiro penal a la izquierda.....	86
Figura 60. Series temporales sobre el eje X tiro penal a la derecha deportista 1.....	88
Figura 61. Series temporales sobre el eje Y tiro penal a la derecha deportista 1.....	88
Figura 62. Series temporales sobre el eje X tiro penal al centro deportista 2.....	89
Figura 63. Series temporales sobre el eje Y tiro penal al centro deportista 2.....	90
Figura 64. Series temporales sobre el eje X tiro penal a la izquierda deportista 3.	91
Figura 65. Series temporales sobre el eje Y tiro penal a la izquierda deportista 3.	91

LISTA DE TABLAS

	Pág.
Tabla 1. Tendencias y cantidad de artículos	22
Tabla 2. Parámetros principales a tener en cuenta	25
Tabla 3. Definición de Requisitos	36
Tabla 4. Dispositivos Explorados.....	37
Tabla 5. Elementos Seleccionados	39
Tabla 6. Características de Dispositivos	42
Tabla 7. Características de los deportistas	79
Tabla 8. Enlaces de videos capturados a los deportistas	79
Tabla 9. Códigos QR de videos capturados a los deportistas.....	79
Tabla 10. Tabla Repositorio de Datasets de los deportistas	80
Tabla 11. Resultados de la ejecución de los dos remates restantes	92

RESUMEN

El avance tecnológico global se manifiesta a través de la introducción de nuevos dispositivos en la rutina diaria de las personas. Actualmente, es frecuente observar a individuos que emplean prendas o accesorios diseñados para mejorar diversos aspectos de su vida cotidiana, tales como la práctica deportiva, la salud o el entretenimiento. Estos objetos, conocidos como "*wearables*" [1], consisten en prendas que incorporan pequeños dispositivos electrónicos, incluyendo microprocesadores, los cuales se integran en alguna parte del cuerpo y mantienen una interacción constante con el usuario u otros sistemas con el propósito de llevar a cabo funciones específicas.

Este documento aborda inicialmente la necesidad de desarrollar métodos de evaluación para habilidades motrices, destacando la falta de atención en este aspecto dentro de las aplicaciones móviles de entrenamiento deportivo, que mayormente se enfocan en la instrucción de planes de hipertrofia muscular o resistencia. Se reconoce la ausencia de métodos de evaluación de metas deportivas sin la dependencia de dispositivos costosos o instalaciones de laboratorio.

Para respaldar la pertinencia de esta investigación, se lleva a cabo una revisión de antecedentes que destaca la importancia de abordar esta brecha en el conocimiento y su aplicabilidad en entornos prácticos. La propuesta de una arquitectura de sistema se fundamenta en el principio de evaluación generada por un sistema, proponiendo la recopilación de datos de movimientos deportivos y la realización de un análisis para su evaluación.

La elección recae en el uso de dispositivos *wearables* para la captura de datos de movimientos, centrándose específicamente en medidas de aceleración y velocidad angular. El análisis implica la comparación de datos entre movimientos ejecutados por un deportista y un deportista experto. La investigación se concentra en el estudio de señales como series temporales, combinando medidas de aceleración y velocidad angular mediante un filtro complementario que genera dos variables resultantes. Se presenta un algoritmo basado en DTW (*Dynamic Time Warping*) [2] para llevar a cabo dicho análisis de datos.

Este documento se divide en seis capítulos estructurados de la siguiente manera:

Capítulo I. Establece la problemática de la investigación, su alcance y justificación.

Capítulo II. Presenta los conceptos generales y un análisis bibliométrico realizado con el software *SciMat*, relacionado con temas *wearable*, *wlan* y tecnologías de desarrollo de sistemas.

Capítulo III. Describe la arquitectura con base en el modelo 4+1 vistas, y presenta el planteamiento inicial de la solución.

Capítulo IV. Expone el desarrollo de un prototipo funcional con base en la arquitectura construida previamente.

Capítulo V. Detalla las fases de preprocesamiento y procesamiento de datos con el algoritmo elegido

Capítulo VI. Detalla mediante un caso de estudio las pruebas realizadas para la validación del sistema.

Capitulo VII: Brinda las conclusiones del proyecto y sus posibles trabajos futuro.

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA

Este capítulo abarca todo lo relacionado con el problema de investigación que debe ser abordado a través del desarrollo de este trabajo. En consecuencia, se identifica el problema, se establecen los objetivos, se define el alcance y se presenta la justificación con el propósito de resolver la pregunta de investigación planteada.

1.1 DEFINICIÓN DEL PROBLEMA

Las aplicaciones móviles se han convertido en el medio de interacción con un sin número de servicios que facilitan las actividades de la vida cotidiana. Desde aplicaciones sencillas como una alarma para despertarse hasta otras más complejas que permiten gestionar transacciones bancarias, haciendo del dispositivo móvil (*smartphone*) una pieza fundamental para realizar alguna actividad o para ser parte de cualquier sistema que vaya a ofrecer un servicio de la cotidianidad.

En el campo del entrenamiento deportivo existen varias aplicaciones móviles que enseñan cómo llevar a cabo un plan enfocado en hipertrofia muscular o resistencia. Se evidencia que dichas aplicaciones no ofrecen un método en el cual al deportista se le permita aprender eficazmente la ejecución de los ejercicios propuestos, lo que conlleva a una incertidumbre en el momento de evaluar las metas propuestas por él. Además, este sistema presenta limitaciones para la enseñanza de habilidades motrices más complejas [3].

Las habilidades motrices presentan diferentes movimientos que involucran la coordinación óculo-manual u óculo-pédica. Estas coordinaciones también llamadas ojo-mano u ojo-pie respectivamente, son unas habilidades cognitivas complejas, que permiten realizar simultáneamente actividades en las que tenemos que utilizar los ojos y las manos o los ojos y los pies. Movimientos como patear un balón de fútbol, la recepción de un balón en voleibol, la brazada de natación, el gesto técnico en el atletismo, entre otros, son ejemplos. Estos movimientos para su mejoramiento o perfeccionamiento se dan bajo la supervisión de un instructor deportivo quien observando el movimiento del deportista brinda una evaluación al respecto. Cabe resaltar que actualmente surge la cultura del deporte como un estilo de vida, donde con ayuda de la tecnología, las personas llevan un plan de entrenamiento de manera autónoma. Sin embargo, al carecer de instructores, no se encuentra una forma en que se pueda realizar una evaluación adecuada orientada al aprendizaje de movimientos. En un contexto de autoaprendizaje, surge entonces una pregunta, ¿cómo puede darse la evaluación del deportista?

Existen tecnologías que permiten desarrollar sistemas compuestos por entornos virtuales, sensores y un equipo multidisciplinario que acompaña la ejecución y evaluación de las actividades del deportista [4]. Esto implica hacer uso de un ambiente determinado (laboratorio), además, presentan características que orientan estos sistemas a un

público que se desenvuelve en un entorno deportivo de alto rendimiento, a partir de lo anteriormente mencionado se pueden considerar como problemáticas:

- Un alto costo debido a la adquisición de diferentes dispositivos, personal especializado y espacios en laboratorios de dedicación exclusiva a estos estudios.
- Escenarios con difícil acceso para personas del común, puesto que en general, estos escenarios a modo de laboratorio están destinados a deportistas de alto rendimiento.
- Necesidad de personal especializado para la interpretación de los resultados, generalmente, en los entornos de alto rendimiento se cuenta con varias personas especializadas en la interpretación de los datos arrojados por los sensores motrices o kinemáticos, para conseguir una evaluación del deportista.

En un contexto cotidiano, las personas no son atletas de alto rendimiento y no cuentan con accesos a laboratorios o sensores de alto desempeño, mucho menos cuentan con la atención de personal especializado para que puedan hacer una evaluación adecuada que les indique si los movimientos aprendidos están realizados de forma correcta.

En este sentido, las personas comúnmente cuentan con un teléfono móvil y pueden acceder a dispositivos *wearables* a un costo razonable, estos les brindan cierta información a través de las aplicaciones de *e-health*, como pasos dados, brazadas dadas en natación, distancia recorrida en una caminata, etc. Otras veces estos dispositivos dan alarmas para retomar la actividad física. Pero no se ha encontrado una tecnología o aplicación que brinde realimentación en la realización de movimientos deportivos. Es decir, no se encuentra una tecnología que capture datos de los movimientos realizados y otorgue alguna realimentación que indique si los movimientos están correctos o requieren mejorar.

Teniendo en cuenta el escenario de una persona del común, donde puede acceder a un dispositivo móvil celular y *wearables*, surge la pregunta, ¿cómo emplear tecnologías de bajo costo en el apoyo al aprendizaje de movimientos óculo-manual u óculo-pédica?

El Internet de las Cosas *IoT* (*Internet of Things*) es una tecnología que cuenta con capacidad de procesamiento y descripción de objetos físicos a nivel sensorial, usando diferentes tipos de redes [5]. Una de estas redes son las Redes Inalámbricas de Área Corporal *WBAN* (*Wireless Body Area Network*). Este tipo de redes de comunicación inalámbrica están compuestas por dispositivos de baja potencia, se compone por una gran cantidad y variedad de *wearables* que son dispositivos electrónicos que se usan en el cuerpo humano entre otros sensores de tipo inercial, los cuales recopilan y envían los datos a una estación base para ser procesados [6]. Esta tecnología sirve para una gran variedad de aplicaciones médicas, electrónica de consumo, entretenimiento entre otros.

Teniendo en cuenta las coordinaciones óculo-manual y óculo-pédica, las redes *WBAN* y el consumo de servicios a través de aplicaciones móviles, se tiene un escenario en el cual es posible pensar que a partir de un sistema basado en dispositivos *wearable*, se puede apoyar la enseñanza y aprendizaje de diversas técnicas motrices con el fin de lograr una evaluación adecuada de forma automática por el sistema, manteniendo una relación costo-beneficio factible para el deportista, como también una usabilidad y portabilidad adecuada para el desarrollo de la actividad física.

A partir del escenario descrito se plantea la siguiente pregunta de investigación: ¿Cómo evaluar técnicas motrices a partir de un sistema basado en *IoT* usando tecnologías *WBAN* y *wearable*?

Con base a la pregunta de investigación se plantea la siguiente hipótesis: A partir de un sistema *IoT* que haga uso de dispositivos *wearable* para capturar datos inerciales de los movimientos del cuerpo humano, se puede realizar una evaluación de técnicas motrices con base en el principio de la coordinación óculo-manual u óculo-pédica.

1.2. OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. Objetivo General

Evaluar gestos técnicos motrices con base en la coordinación óculo-manual u óculo-pédica a través de un sistema *IoT-WBAN*.

1.2.2. Objetivos Específicos

1. Desarrollar un prototipo de sistema con características *IoT-WBAN* que facilite la recolección y análisis de datos específicos de movimientos del cuerpo humano basados en la coordinación óculo-manual u óculo-pédica.
2. Implementar un algoritmo que evalúe gestos técnicos motrices a partir de los datos recolectados en el primer objetivo específico.
3. Validar el prototipo de sistema en un caso de estudio basado en un gesto técnico de fútbol (tiro libre penalti).

1.3. ALCANCE DE LA INVESTIGACIÓN

Se propone el diseño de un sistema integrado para la evaluación de gestos técnicos motrices, enfocándose principalmente en la coordinación óculo-pédica y óculo-manual. Este diseño se basa en experiencias adquiridas a través de diversas fuentes, como libros, proyectos de investigación, SciMat y otra información relevante para el proyecto.

El sistema integra diferentes herramientas con el objetivo de recolectar, almacenar, filtrar, evaluar y visualizar datos de movimientos en un entorno deportivo. A través de un caso de estudio, se evalúa la eficacia del sistema y se analizan los resultados obtenidos.

Finalmente, se realiza una conclusión sobre el funcionamiento del sistema, donde se sintetizan los hallazgos y se destacan las ventajas y limitaciones identificadas durante el proceso de evaluación.

1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN

El estudio de la literatura existente se realizó mediante un análisis bibliométrico haciendo uso del *software SciMAT* versión 1.1.04 [7]. La búsqueda se realizó entre mayo y junio de 2022 en las bases de datos *ScienceDirect* y *SCOPUS*. En esta, se tuvo en cuenta los años comprendidos entre el 2017 y 2022. Partiendo del contexto de esta investigación mencionado en el planteamiento del problema, se hicieron las consultas a partir de las cadenas booleanas “*wearable AND movement*” y “*Athletes training AND IoT*”. Con base a lo anterior se obtuvo 4577 artículos. La Figura 1, muestra que en las consultas realizadas hay un incremento en las publicaciones para el periodo comprendido entre 2017 y 2022, mostrando así un interés en la comunidad científica por avanzar en estos temas.

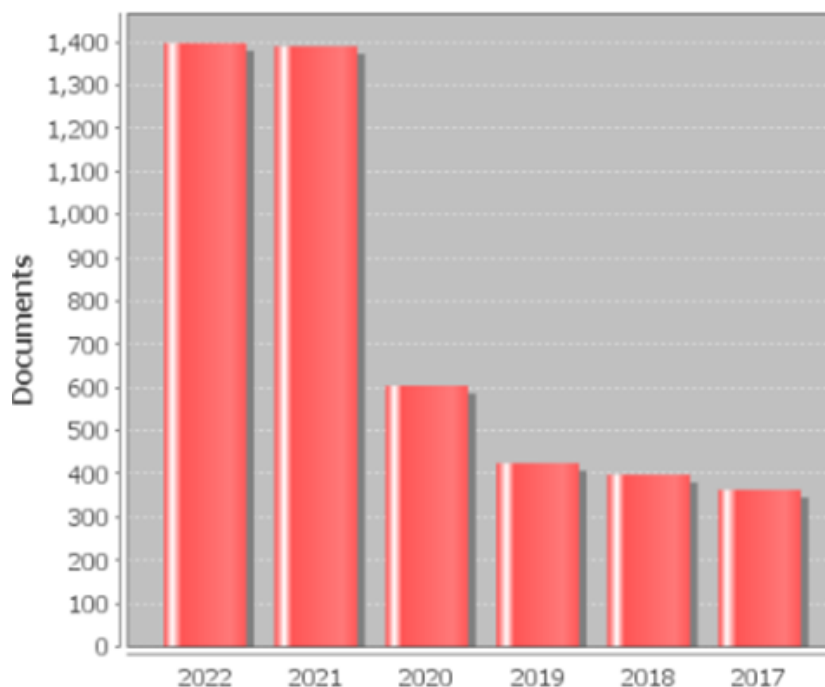


Figura 1. Artículos por Periodo. Fuente: Elaboración propia

Una de las funcionalidades de *SciMat* permite conocer tanto las tendencias de investigación como las que ya no lo son a partir de la generación de gráficos estratégicos. En estos gráficos se define con una región motora, donde se

consideran tópicos emergentes o de vanguardia. En la Figura 2 se tiene gráfico estratégico para el año 2022, en el cual el tema de *Internet Of Things*, está a la vanguardia.

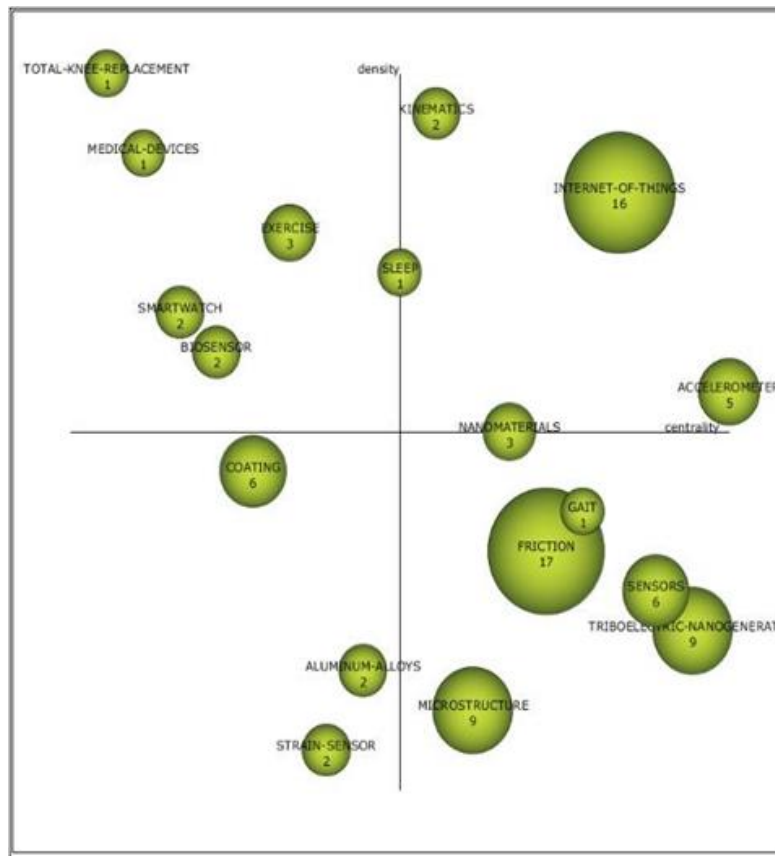


Figura 2. Diagrama Estratégico del año 2022. Fuente: Elaboración propia

Siendo el tópico de *Internet Of Things* un tema de vanguardia, se genera un gráfico de agrupamiento para ver los temas en relación (ver Figura 3). Se observa entonces, que, en este caso, se encuentran tecnologías como los dispositivos *wearables*, aplicaciones, sensores, cuidados de salud y aplicaciones móviles con una relación directa establecida con IoT. En el caso de este trabajo, se puede considerar que son temas necesarios para la construcción de un sistema que pueda dar solución al problema definido.

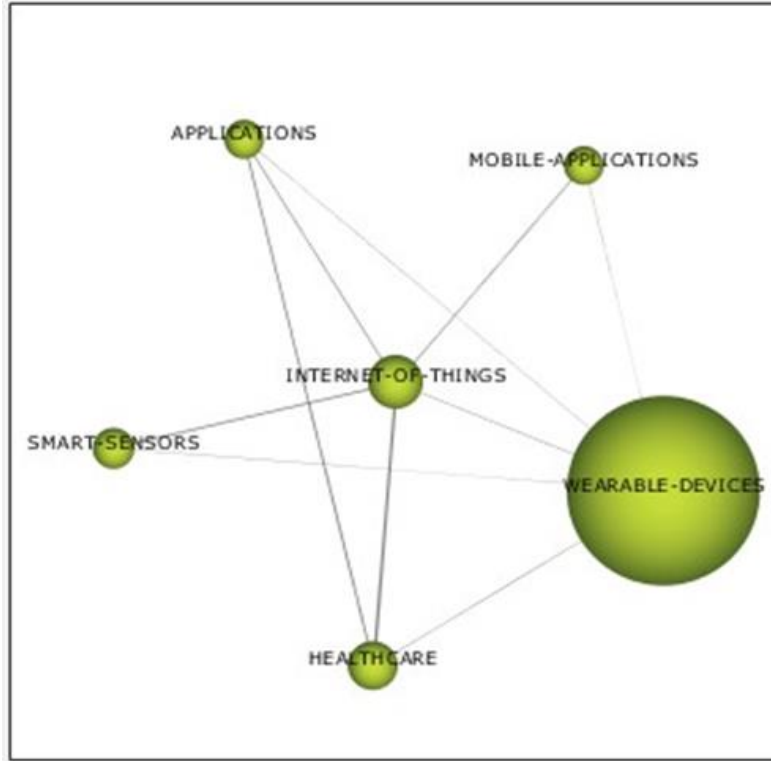


Figura 3. Clúster Internet Of Things. Fuente:Elaboracion propia

Es importante destacar que el trabajo de grado realizado tiene un enfoque significativo en el diseño de un sistema portátil que registre los movimientos involucrados en la ejecución de un movimiento, centrándose específicamente en la coordinación óculo-manual y/o óculo-pédica. Este enfoque se debe a la línea de investigación de Aplicaciones y Servicios sobre Internet - Internet de las Cosas, con énfasis en sistemas embebidos, que respalda la práctica deportiva con el propósito de evaluar la técnica. Se busca lograr una relación costo/beneficio accesible para los deportistas.

CAPÍTULO 2: ANÁLISIS BIBLIOMÉTRICO

A continuación, se presentan los conceptos generales y el estado del arte, junto con una síntesis de los trabajos de investigación relacionados con el desarrollo de dispositivos wearables utilizando la herramienta Scimat. Finalmente, se enumeran los artículos seleccionados, describiendo las contribuciones realizadas a este trabajo.

2.1 CONCEPTOS GENERALES

En esta sección, se presenta una breve descripción de los conceptos fundamentales empleados en la elaboración de este trabajo.

2.1.1 Entornos de desarrollo

2.1.1.1 IDE Arduino

El entorno de desarrollo integrado IDE Arduino (por sus siglas en inglés *integrated development environment*), es una aplicación multiplataforma la cual está escrita en un lenguaje de programación java, se conformada por un conjunto de herramientas de programación y es exclusivamente dedicada a dos lenguajes de programación, C y C++ debido a su biblioteca llamada “*Wiring*” [8].

Este entorno de programación consiste en un editor de código, un compilador, un depurador, un constructor de interfaz gráfica (GUI) e incorpora herramientas necesarias para cargar un programa ya compilado en la memoria *flash* de cualquier hardware libre [9]. Al ser una plataforma libre es compatible con placas de diferentes creadores, pero son igualmente funcionales por lo que parten de la misma base. Cuando se carga un programa en la placa, se dispone de un *bootloader*, este es un programa cargado en el microcontrolador que permite subir el código sin ayuda de hardware adicional. El *bootloader* está activo unos segundos cuando se resetea la placa, en seguida el programa que tenga cargado la placa en su memoria *Flash* comienza a funcionar. El led integrado dependiendo del tipo de placa emite un parpadeo que indica que el *bootloader* se está ejecutando [10].

Los programas escritos con este entorno se denominan *sketches*, los cuales se almacenan con la extensión de archivo .ino. El editor tiene como función cortar, pegar, buscar o reemplazar texto. La sección de mensajes proporciona comentarios y/o muestra errores al guardar y exportar, La consola muestra la salida de texto del software Arduino (IDE), incluidos los mensajes de error completos y otra información. La esquina inferior derecha de la ventana muestra la placa configurada y el puerto serie. Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar sketches y abrir el monitor en serie [11].

2.1.1.2 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero muy potente, cuenta con un soporte integrado para JavaScript, TypeScript y Node.js, sin embargo, se puede crear código en distintos lenguajes debido a diversos *plugins* o extensiones. Este entorno es multiplataforma por lo que se puede trabajar sobre sistemas operativos como Windows, GNU/Linux y macOS [12], en este entorno se puede desarrollar sitios web, aplicaciones WPF, servicios web, aplicaciones para dispositivos móviles con Xamarin, desarrollar videojuegos.

Este entorno de desarrollo cuenta con múltiples características como IntelliSense, con esta característica es posible el autocompletado y resaltado de sintaxis en la edición de código, lo que permite escribir código de una manera ágil. Además, proporciona sugerencias de código y terminaciones inteligentes con base a los tipos de variables, funciones, etc. Otra característica es la depuración que ayuda a detectar errores en el código, así, se evita que los desarrolladores tengan que revisar cada una de las líneas para encontrar errores. Cabe resaltar que VS Code es capaz de detectar errores mínimos automáticamente antes de ejecutar código o la depuración como tal [13].

Visual Studio Code cuenta con una terminal compuesta por todas las funciones, y se inicia fácilmente en el directorio de trabajo. Dicha terminal integrada puede utilizar cualquier Shell instalado en el equipo, como *PowerShell*, *Bash* o cualquier otro.

2.1.1.3 Jupyter Notebook

Jupyter Notebook es una aplicación cliente-servidor donde es posible crear y compartir documentos web con extensión *.JSON* donde estas siguen un esquema versionado y una lista ordenada de celdas de entrada y de salida. Estas celdas almacenan, entre otras cosas, código, texto (en formato *Markdown*), fórmulas matemáticas, ecuaciones o también contenido multimedia (Rich Media). El programa se ejecuta escribiendo *jupyter notebook* en una línea de comandos, es decir desde la aplicación web-cliente la cual es compatible con cualquier navegador estándar. Como requisito previo se debe instalar y ejecutar en el sistema el servidor Jupyter Notebook. Los documentos creados en Jupyter pueden exportarse, entre otros formatos, a *HTML*, *PDF*, *Markdown* o *Python* y también pueden compartirse con otros usuarios por correo electrónico, utilizando *Dropbox*, *GitHub* o mediante su visor integrado [14].

Este entorno fue creado a partir de la interfaz *Ipython Notebook*, cuenta con las mismas funcionalidades, la diferencia es la posibilidad de ejecutar código en diversos lenguajes, Esta ejecución es posible mediante la comunicación con un núcleo o *Kernel*, inicialmente el *kernel* de cálculo *python* por defecto, con el tiempo y la versatilidad que caracteriza a *jupyter notebook* ha permitido aumentar el número de núcleos disponibles incluyendo Octave, Julia, R, Haskell, Ruby, C/C++, Fortran, Java, SageMath, Scala, Matlab y Mathematica en otros [15].

Jupyter Notebook es un entorno diseñado para cumplir necesidades específicas y ajustarse al flujo de trabajo de la ciencia de datos y la simulación numérica. En una sola interfaz, los usuarios pueden escribir, documentar y ejecutar código, visualizar datos, realizar cálculos y ver los resultados. Además, el código se puede organizar en celdas independientes, por lo que es posible probar secciones del código de manera individual.

2.1.1.4 Android Studio

Es un entorno de desarrollo integrado (IDE) especialista en el desarrollo de aplicaciones móviles basados en sistemas Android [16]. Este IDE tiene como referencia el potente editor de código y sus respectivas herramientas de desarrollo de IntelliJ IDEA (entorno de desarrollo líder en java y kotlin).

Al estar basado en IntelliJ IDEA cuenta con características al momento escribir código como las funciones de asistencia de codificación que se encargan de buscar posibles errores y brindar sugerencias de mejora a medida que se escribe, mientras tanto se educa sin problemas sobre las mejores prácticas de la comunidad en codificación, nuevas funciones de lenguaje y más. Adicionalmente es compatible con sistemas operativos Windows, ChromeOS, MacOS o Linux lo cual permite que cualquier desarrollador lo pueda utilizar desde cualquier ordenador o dispositivo [17].

La estructura de un proyecto en Android Studio puede incluir uno o varios módulos con archivos de código fuente y archivos de recursos. Algunos tipos de módulos son Módulos de aplicaciones para Android, Módulos de biblioteca, Módulos de Google App Engine entre otros, este enseña los archivos de un programa en la vista de proyecto de Android. Esta vista está organizada en módulos para poder acceder de manera directa a los archivos fuente clave del proyecto. Se pueden ver todos los archivos de compilación en el nivel superior de las secuencias de comandos de Gradle.

2.1.2 Protocolos HTTP

HTTP (*Hypertext Transfer Protocol*) es el protocolo con el cual se realiza una petición de datos y recursos, como pueden ser documentos HTML. Es el punto de partida para cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, por lo que se deduce que el elemento que recibirá los datos (el cliente) inicia la petición, normalmente un navegador Web [18]. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc.

Este protocolo funciona en la capa de aplicación, y se transmite sobre el protocolo *TCP*, o el protocolo encriptado *TLS* (en-US), teóricamente es posible usar cualquier otro protocolo fiable. Debido a que este es un protocolo capaz de ampliarse, se usa no sólo para transmitir documentos de hipertexto (*HTML*), sino que además, se usa para

transmitir imágenes, vídeos, enviar datos o contenido a los servidores, como en el caso de los formularios de datos [19]. *HTTP* puede incluso ser utilizado para transmitir partes de documentos, y actualizar páginas Web en el acto.

2.1.3 Wearable

Un Wearable es todo aquel dispositivo electrónico que se puede utilizar en cualquier parte del cuerpo humano, este tiene la posibilidad de interactuar con otros aparatos para transmitir o recoger algún tipo de datos. Algunos de estos dispositivos vienen compuestos por una variedad de sensores, esto depende de su función. Como un ejemplo claro de *wearable* se tiene los relojes inteligentes (*smartwatch*) y las pulseras de actividad (*smartband*), cabe resaltar que hay muchos más con diferentes funcionalidades. Los más conocidos están diseñados para ser utilizados con actividades de entretenimiento, aunque el uso de estos accesorios es también muy valioso en el mundo de la moda, el deporte profesional, la medicina o la industria.

Esta tecnología surgió en la década de los años 60, pero hasta los últimos años no había entrado de lleno en la sociedad. Son normalmente, objetos cotidianos los cuales fueron combinados con tecnología con el fin de ampliar su funcionalidad y tener a su alcance las utilidades de otros dispositivos siendo parte de redes *ban* (*body area network*) o *lan* (*local area network*). Una de las características de los *wearables* es que siempre están encendidos y/o en funcionamiento, además son dispositivos multitarea e incluso pueden ejecutar su propio sistema operativo [20].

2.1.4 WiFi

Como tecnología de comunicación de *wearable* se tiene el **WiFi** (de la marca registrada **WiFi**). Esta tecnología permite la interconexión inalámbrica de dispositivos electrónicos, estos pueden conectarse entre sí o a internet a través de un punto de acceso de red inalámbrica.

Dicha conexión inalámbrica se realiza mediante el uso de radiofrecuencias, utilizadas para la transmisión de información. Por otra parte, se tiene una limitación, ya que el deportista deberá encontrarse dentro de un alcance específico de cobertura para poder aprovechar la señal. Generalmente, ese radio varía entre los 5 y los 150 metros de distancia con respecto al aparato emisor de dicha señal.

Existen diferentes tipos de **WiFi** [21], los cuales se clasifican según los estándares 802.11 y en función de su utilidad, varía la velocidad de la señal, estos son:

- IEEE 802.11. Creado en 1997, en el presente ya no se usa. Éste permitía una velocidad máxima de conexión de dos megabits por segundo, unos valores muy lentos para la mayor parte de las aplicaciones.

- IEEE 802.11a. Surgió en 1999 con una velocidad máxima de 54 megabits por segundo. Esta versión fue la primera en funcionar a 5 GHz, una frecuencia que bloquea fácilmente los objetos y que hace que este estándar tenga un alcance limitado.
- IEEE 802.11b, IEEE 802.11g, IEEE 802.11n. Cuentan con velocidades máximas de 11, 54 y 300 megabits por segundo y disponen de una frecuencia de 2,4 GHz, una banda casi universal de bajo consumo que los convierte en los más empleados internacionalmente.
- IEEE 802.11ac. Nacido en 2014 y conocido como *WiFi 5*, funciona a una velocidad máxima de 1.300 megabits por segundo y opera en la banda de 5GHz.

2.1.5 ESP-32

Son una serie de microcontroladores caracterizados por ser de bajo costo y de bajo consumo, cuenta con un módulo integrado compuesto por *WiFi* y *Bluetooth*, sus múltiples entornos de desarrollo de código abierto y sus bibliotecas la hacen perfectamente adecuada para llevar a cabo proyectos de todo tipo. Están compuestos por un procesador integrado con interfaces para conectarse con varios periféricos tales como sensores, el procesador está formado por dos núcleos de procesamiento cuyas frecuencias operativas pueden controlarse de manera independientemente entre 80 MHz y 240 MHz. Los periféricos del procesador facilitan la conexión a una variedad de interfaces externas.

Para elegir una placa ESP-32 hay que tener en cuenta los siguientes aspectos [22]:

- Números de pines y configuración: es importante tener acceso a la distribución de pines de la placa para poder hacer un uso correcto de ella.
- Interfaz Serie-USB y regulador de voltaje: estas dos características las tienen prácticamente todas las placas de desarrollo. Estas son las que permiten conectar la placa directamente al ordenador para ser energizada y programada.
- Conector para batería: si estás pensando en incursionar en sistemas de bajo consumo con baterías puedes optar por placas que ya incluyan conectores para baterías.
- Funciones extras: muchas placas de desarrollo para ESP-32 traen características extras como cámaras, pantallas OLED, módulos LoRa, etc.

2.1.6 Lenguajes De Programación

2.1.6.1 C++

C++ es un lenguaje de programación visto como una extensión de “C”, este cuenta con mecanismos que hacen posible el manejo de excepciones, el polimorfismo, la herencia y la sobrecarga de operadores lo cual permite la

manipulación de objetos, siendo así, un lenguaje híbrido. Por otra parte se agregaron facilidades de programación genérica, que se incorporaron a los paradigmas de programación estructurada y programación orientada a objetos [23]. Por lo tanto, se dice que C++ es un lenguaje de programación multiparadigma de tipo imperativo.

Este lenguaje tiene la capacidad de manipular directamente el hardware y los recursos del sistema, por lo que es apropiado para el desarrollo de aplicaciones de bajo nivel, como controladores de dispositivos, sistemas operativos y motores de videojuegos. Asimismo, C++ es empleado por una gran variedad de aplicaciones, entre estas se tiene el desarrollo de aplicaciones de escritorio, aplicaciones empresariales, software para sistemas embebidos y/o aplicaciones de alto rendimiento entre otros.

Algunas características que hacen a C++ la mejor opción para desarrollar sistemas embebidos son [24]:

- Eficiencia: C++ tiene control de la memoria y los recursos del sistema, lo cual posibilita una programación eficiente y un uso óptimo de los recursos limitados.
- Acceso a hardware: C++ permite acceder directamente al hardware del sistema a través de punteros y manipulación de bits, lo que es esencial en sistemas embebidos donde se requiere un control preciso de los dispositivos y periféricos.
- Control de tiempo real: C++ proporciona mecanismos para manejar restricciones de tiempo críticas, como hilos de ejecución, sincronización y programación concurrente. Esto es fundamental en sistemas embebidos en los que se deben cumplir estrictamente los plazos y se requiere una respuesta en tiempo real.
- Portabilidad: C++ es un lenguaje portátil y ampliamente compatible que se puede utilizar en una variedad de plataformas y sistemas operativos, lo que facilita el desarrollo de aplicaciones embebidas independientes del hardware subyacente.

2.1.6.2 Python

El análisis de un gran volumen de datos requiere de software capaz de procesarlo en forma idónea de acuerdo a los requerimientos establecidos, además del análisis, es necesario que permita desarrollar módulos que conformen parte de un sistema que utiliza diferentes tecnologías. Así, el lenguaje de programación Python cumple con estas características.

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos [25]. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de las plataformas.

El intérprete de Python y la extensa librería estándar se encuentran disponibles libremente en código fuente y de forma binaria para la mayoría de las plataformas desde la Web de Python [26], y se pueden distribuir libremente. El mismo sitio también contiene distribuciones y referencias a muchos módulos libres de Python de terceros, programas, herramientas y documentación adicional.

2.1.6.3 Java

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Es una plataforma fiable que permite crear muchos servicios y aplicaciones. Está diseñada para ser una plataforma independiente, lo que significa que puede ejecutarse en cualquier sistema que tenga instalado la máquina virtual de java (JVM) [27].

Las principales características de java son:

- **Orientada a objetos:** Java es un lenguaje de programación orientado a objetos, por lo cual está enfocado en crear módulos reutilizables de código. Por lo tanto, está en la capacidad de desarrollar aplicaciones bajo conceptos como clases, objetos, herencia y polimorfismo.
- **Sintaxis:** la sintaxis de Java es amigable con el desarrollador. Además, es fuertemente tipada, lo cual implica una declaración explícita de las variables y su tipo.
- **Librerías y APIs:** Java tiene una gran cantidad de librerías y APIs, las cuales proporcionan diferentes funcionalidades que apoyan el desarrollo de nuestras aplicaciones.
- **Procesos múltiples:** Java soporta la ejecución de múltiples procesos en un simple programa. Esto es muy útil para el desarrollo de aplicaciones que requieren ejecutar múltiples tareas simultáneamente.

El desarrollo y ejecución de aplicaciones en Java requiere tener instalado en el equipo el Java Development Kit (JDK). El JDK incluye el compilador de java, entorno de ejecución y otras herramientas necesarias para el desarrollo en Java. Finalmente, el ecosistema de Java es muy amplio y permanece en constante crecimiento por su utilidad en múltiples sectores industriales debido a su versatilidad, rendimiento y fiabilidad.

2.1.6.7 PHP

PHP (Hypertext Preprocessor) es un lenguaje del lado del servidor diseñado especialmente para desarrollos web [28]. Este fue creado por Rasmus Lerdorf en 1994 y desde entonces ha evolucionado y es ampliamente utilizado para construir aplicaciones web dinámicas.

Las principales características de PHP son:

- Lenguaje del lado del servidor: PHP es principalmente usado del lado del servidor, lo que significa que el código hecho con PHP se ejecuta en el servidor antes de ser enviado al navegador. Esto permite generar contenido de forma dinámica, interactuar con bases de datos y recursos en otros servidores.
- Sintaxis: la sintaxis de PHP es similar a la de otros lenguajes de programación como C o Java. Es un lenguaje débilmente tipado, esto implica que las variables no necesitan ser explícitamente declaradas con un tipo de dato.
- Desarrollo Web: PHP tiene soporte para el desarrollo web. Incluye funciones y librerías para el manejo de solicitudes HTTP, carga de archivos, manejo de sesiones, cookies y más. también se integra muy bien con varios servidores web como Apache y Nginx.
- Conexión a bases de datos: PHP tiene un amplio soporte para la conectividad a bases de datos. Proporciona una capa unificada llamada PDO (PHP Data Objects) que permite interactuar con múltiples sistemas de bases de datos, como, MySQL, PostgreSQL, Oracle y SQLite.
- Código abierto: PHP es un lenguaje de código abierto, lo que significa que es libre para usar y tiene una gran cantidad de recursos online y documentación.

Para ejecutar código de PHP, es necesario instalar un servidor web con PHP. El más común es Apache o Nginx como servidor web e instalar PHP como un módulo o como un servicio separado.

2.1.7 Librerías

Una librería o framework son un conjunto de bloques de código y funcionalidades que se utilizan para agilizar y facilitar el desarrollo de aplicaciones en cualquier lenguaje de programación [29]. Dichas librerías contienen diversos conjuntos de funciones, clases y rutinas que se pueden utilizar en la creación de un programa con el fin de llevar a cabo tareas específicas sin la necesidad de escribir el código en su totalidad.

Las librerías tienen la capacidad de entregar una abstracción de alto nivel al encapsular una funcionalidad de gran complejidad en funciones y métodos adaptables lo cual hace que sean reutilizables. Cuando se hace uso de una librería, los desarrolladores tienen la posibilidad de aprovechar el trabajo previo realizado por otros programadores, lo cual, ahorra tiempo y esfuerzo en el desarrollo de nuevas aplicaciones.

Las librerías se pueden clasificar en cuatro clases [30]:

- Librerías Estáticas: Son archivos compilados que contienen código de tipo objeto y funciones predefinidas, estas librerías están vinculadas estáticamente al programa durante la etapa de compilación, lo que quiere decir que el código de la librería se copia directamente en el ejecutable final del programa.

- **Librerías Dinámicas:** Son archivos compilados que contienen código y funciones que, a diferencia de las librerías estáticas, las librerías dinámicas se cargan en tiempo de ejecución y no se vinculan directamente con el ejecutable del programa durante la etapa de compilación.
- **Librerías Estándar:** Son aquellas que están incluidas en la distribución estándar de un lenguaje de programación. Estas librerías son de uso constante e incluyen funcionalidades básicas y comunes, las cuales son imprescindibles para desarrollar cualquier tipo de aplicación.
- **Librerías de Terceros:** Son aquellas que fueron desarrolladas por personas o equipos externos a la entidad a la cual pertenece el lenguaje de programación. Estas librerías proporcionan funcionalidades adicionales específicas para diferentes propósitos, como el manejo de bases de datos, el procesamiento de imágenes, la interacción con APIs externas, entre otros.

Las librerías en sistemas embebidos tienen un papel importante, dado que permiten el uso de operaciones básicas de bajo nivel, como el acceso a puertos GPIO (Entrada/Salida de Propósito General), hasta funciones más complejas, como el control de comunicaciones, protocolos de red, manejo de periféricos, etc.

2.1.8 Sensores Inerciales

Los sensores inerciales son dispositivos capaces de medir y registrar la aceleración lineal y angular de un objeto el cual está en movimiento. Están diseñados para medir cambios en la velocidad y la orientación con respecto a un sistema de referencia fijo [31]. Los sensores inerciales comunes utilizan una combinación de acelerómetros y giroscopios [32]. Los acelerómetros miden la aceleración lineal en tres ejes: X, Y y Z, mientras que los giroscopios miden la velocidad angular alrededor de los mismos ejes. Estos componentes trabajan de manera simultánea para proporcionar información sobre el movimiento y la orientación de un objeto.

Un sensor inercial muy común es el MPU6050 [33], el cual está compuesto por un acelerómetro de tres ejes y un giroscopio de tres ejes en un solo chip. Es un dispositivo muy requerido en aplicaciones que necesitan mediciones precisas de movimiento y orientación. Este sensor es utilizado frecuentemente en aplicaciones, como estabilización de vuelo en drones y quadcopters, seguimiento de movimiento en realidad virtual y aumentada, control de balance en robots y plataformas móviles, monitoreo de actividad física y detección de gestos a partir de dispositivos portátiles, entre otros.

2.1.9 Energía Consumida

La vida útil del dispositivo *wearable* es vital para poder llevar una buena fluidez en la recolección de datos, debido a esto se debe maximizar el tiempo de operatividad. Es por ello, la importancia de calcular el consumo de energía

total. Esta información se puede encontrar fácilmente en la hoja de datos (*datasheet*) del respectivo fabricante de cada componente que conforma el dispositivo *wearable*.

2.1.10 Batería

Una batería es un dispositivo electroquímico el cual tiene como función almacenar energía química y transformarla en energía eléctrica por medio de reacciones químicas internas. Está conformada por celdas electroquímicas, estas contienen electrodos (un ánodo y un cátodo) y un electrolítico [34].

Cuando una batería está en funcionamiento, se genera una reacción química en el interior de las celdas que libera electrones en el ánodo y son dirigidos al cátodo. Esta corriente de electrones o flujo de electrones es lo que permite energizar un dispositivo electrónico.

Las baterías se pueden clasificar por su tipo y tecnología [35], algunas de estas son de plomo-ácido, baterías de níquel-cadmio (NiCd), baterías de níquel-metal hidruro (NiMH), baterías de iones de litio (Li-ion) entre otras. Cada tipo de batería cuenta con sus propias características tales como capacidad, densidad de energía, vida útil, velocidad de carga, tamaño, forma y seguridad.

Las baterías se emplean en una gran variedad de dispositivos, desde dispositivos portátiles como teléfonos móviles y relojes inteligentes hasta vehículos eléctricos y sistemas de almacenamiento de energía a gran escala.

En seguida se listan tres tipos de batería comúnmente utilizadas en dispositivos portátiles (*wearables*):

2.1.10.1 Batería tipo botón

Es una batería de alta portabilidad, es de una sola celda y se caracteriza por su alta estabilidad ante los cambios de temperatura [36], esta se reconoce fácilmente por su forma redonda y estrecha, como desventaja se tiene que no es una batería recargable.

2.1.10.2 Batería Multipack

Es una batería de portabilidad media baja debido a sus diferentes presentaciones, manejan diferente cantidad de celdas, como ventaja se tiene que son de tipo recargables [37].

2.1.10.3 Batería de litio

Esta batería es de tipo recargable, su portabilidad depende de la cantidad de energía que almacena, sin embargo, si se hace una relación tamaño-capacidad se puede evidenciar que la energía que almacena es alta respecto a las

dimensiones de esta misma, emplea como electrolito sal de litio la cual consigue los iones necesarios para la reacción electroquímica reversible [38].

2.1.11 Módulo FastDTW-PYTHON

FastDTW (*Fast Dynamic Time Warping*) es un algoritmo de alineación temporal diseñado para comparar series temporales o secuencias de datos que pueden variar en velocidad y duración. A diferencia del algoritmo de DTW (*Dynamic Time Warping*) tradicional, que es computacionalmente costoso y puede ser lento en series temporales largas, FastDTW es una versión eficiente que proporciona una aproximación alineada temporalmente a un costo computacional mucho menor [39].

En Python, puedes utilizar FastDTW a través de la biblioteca "fastdtw" que ofrece una implementación eficiente del algoritmo FastDTW. Esta biblioteca es una opción popular para aquellos que necesitan realizar alineaciones temporales eficientes en sus datos de series temporales.

2.1.12 XAMPP

El desarrollo de sistemas que involucren el consumo de servicios a través de una red, requiere de un servidor, por esto, el entorno XAMPP ofrece diferentes tecnologías que permiten desarrollar un servidor local. Esto es, con el consumo de los recursos hardware del equipo que se utilice para desplegar el servidor [40].

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Linux: Es el sistema operativo donde estará instalado nuestra aplicación. A diferencia de Windows, Linux es una distribución libre que es segura, no requiere pago de licencias y tiene alto rendimiento.

Apache: el servidor web de código abierto es la aplicación usada globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.

MySQL/MariaDB: XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB.

PHP: Es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.

Perl: este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.

2.1.13 Procesamiento Digital de Señales (PDS)

El procesamiento digital de señales (PDS) es la técnica utilizada para tratar señales a través de algoritmos y procesos a nivel digital. Estas señales pueden ser de diversas clases, como señales de audio, señales de video, señales de radar, señales biomédicas, entre otras [41]. El objetivo principal del procesamiento digital de señales es extraer información útil de las señales, mejorar su calidad, analizar su contenido o transformarlas según la necesidad del caso.

Cuando se trata de sistemas embebidos, el procesamiento digital de señales toma un papel muy importante, este se implementa utilizando algoritmos optimizados y técnicas de programación competentes con el fin de aprovechar al máximo los recursos limitados de la plataforma. Además, los sistemas embebidos suelen incluir periféricos especializados, como sensores especializados para la recolección de datos, controladores DMA (Acceso Directo a Memoria) o aceleradores de hardware, para acelerar ciertas operaciones de procesamiento de señales y reducir la carga de la CPU principal.

Gran variedad de técnicas y algoritmos utilizados en el procesamiento digital de señales abarcan:

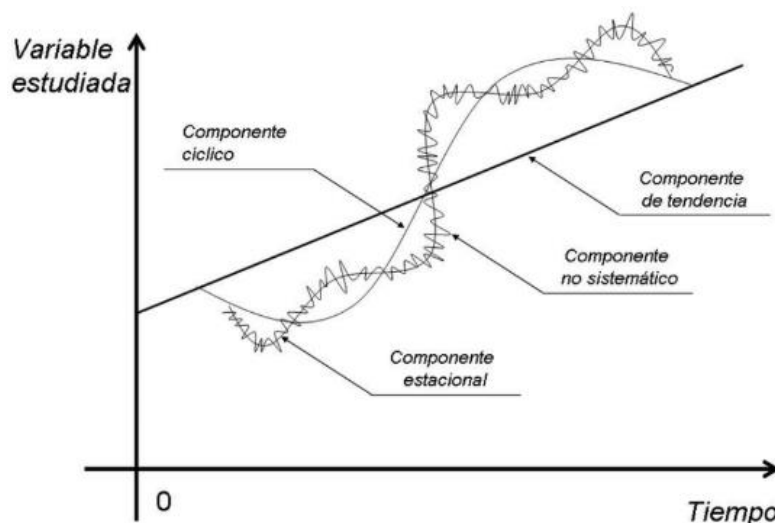
- Transformada de Fourier: se utiliza para analizar y representar una señal en el dominio de la frecuencia. Puede utilizarse para aplicar filtros, eliminar ruido o extraer características de la señal.
- Filtrado digital: implica la aplicación de filtros digitales, como filtros pasa bajos, pasa altos o pasa banda, para modificar una señal de acuerdo con ciertos requisitos.
- Compresión de datos: se refiere a la reducción del tamaño de los datos de señal para minimizar el uso de memoria o el ancho de banda requerido para transmitir la señal.
- Procesamiento de imágenes: implica el uso de algoritmos para manipular imágenes digitales, como filtrado de imágenes, detección de bordes o reconocimiento de objetos.
- Procesamiento de audio: incluye técnicas como la cancelación de ruido, la mejora de la calidad del audio, la detección de voz o la compresión de audio.

2.1.14 Series de Tiempo

Una serie temporal se refiere a una secuencia de observaciones que se registran en momentos específicos del tiempo, dispuestas en un orden cronológico y con intervalos uniformes entre ellas [42]. En esta secuencia, los datos suelen identificarse como X_t , donde $t = 1, 2, \dots, n$.

El análisis tradicional de las series temporales se fundamenta en la premisa de que los valores observados de una variable se componen de cuatro componentes que, al actuar en conjunto, producen los valores medidos. Estos tres componentes se ilustran en la Figura 4. y son los siguientes:

- **Componente tendencia:** Se puede definir como una variación a largo plazo que ocurre en relación con el nivel medio, o como el cambio a largo plazo en la media. La tendencia se caracteriza por un movimiento gradual y continuo en la serie a lo largo del tiempo.
- **Componente estacional:** Se refiere a la presencia de patrones periódicos o variaciones que ocurren en ciertos intervalos de tiempo (como semestrales, mensuales, etc.) en muchas series temporales. Por ejemplo, las ventas al por menor en Puerto Rico tienden a aumentar durante los meses de noviembre y diciembre debido a las festividades navideñas. Estos efectos estacionales son fácilmente comprensibles y pueden medirse de manera explícita o incluso eliminarse de la serie de datos. Este proceso se conoce como desestacionalización de la serie.
- **Componente aleatorio:** Se caracteriza por no seguir ningún patrón discernible de comportamiento. En lugar de eso, representa el resultado de factores fortuitos o aleatorios que afectan la serie temporal de manera aislada, sin una estructura predecible.
- **Componente cíclico:** Se refiere a patrones de comportamiento recurrentes en una serie temporal, aunque no necesariamente siguen una periodicidad exacta como en el caso de la estacionalidad. Estos patrones pueden mostrar oscilaciones o fluctuaciones que se repiten a lo largo del tiempo, pero con variaciones en la duración o la amplitud de los ciclos.



(Fuente: Sapag 2000, pág. 96)

Figura 4. Componentes de una Serie de Tiempo. Fuente: tomado de Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

2.1.15 Dynamic Time Warping (DTW)

DTW es un algoritmo ampliamente aceptado en la comparación de series temporales. Su función principal es evaluar la similitud entre dos secuencias que pueden variar tanto en el tiempo como en el espacio. Esta similitud se cuantifica mediante el cálculo de la distancia entre las dos series temporales. Aunque su origen se encuentra en el ámbito del reconocimiento de voz, su aplicación se ha extendido a diversas áreas, como la biometría, la verificación de huellas dactilares, la escritura a mano y la minería de datos, entre otras [2].

Este algoritmo se basa en el trabajo previo realizado por investigadores en el campo del reconocimiento de voz, particularmente en el estudio realizado por el autor [44]. La premisa subyacente es la siguiente: cuando dos personas pronuncian la misma frase, pero con distintas velocidades o duraciones temporales, las representaciones de estas frases tendrán en general una forma similar, pero con deformaciones a lo largo del eje del tiempo. El propósito de DTW es eliminar estas diferencias temporales y permitir la alineación incluso entre un punto de una serie con varios puntos de la otra serie (ver Figura 5).

Para cada punto, C_k sea:

$$d(C_k) = d(I_k, J_k) = a(I_k) - b(J_k)$$

La distancia entre los valores de ambas series fue alineada. Dada una trayectoria F , se define la suma ponderada de dichas distancias.

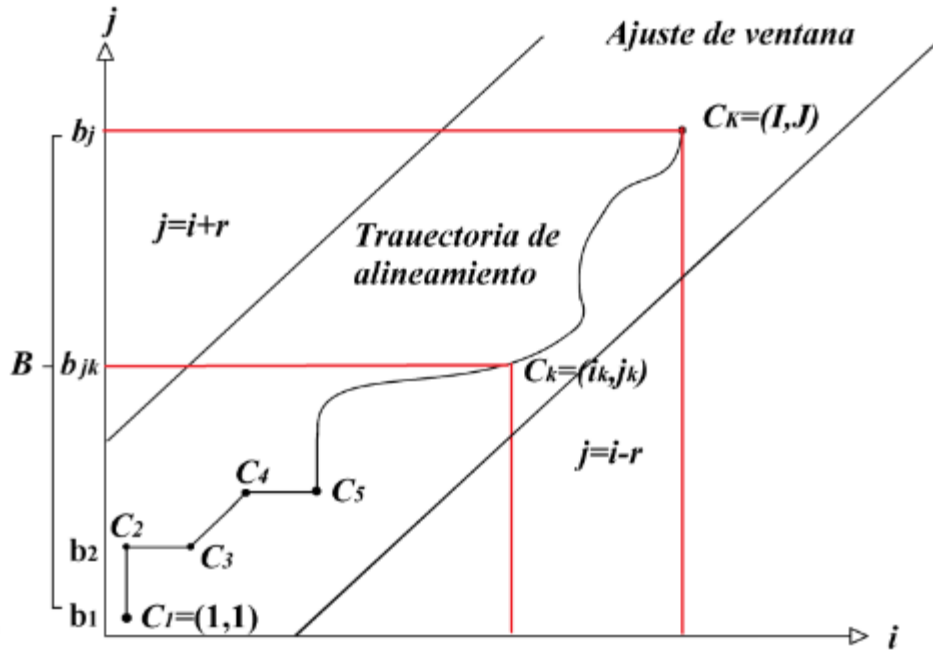


Figura 5. Ajuste de Ventana. Fuente: tomado de Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

$$S(F) = \sum_{k=1}^{k=K} d(C_k)w_k \text{ con } w_k \geq 0$$

Retomando la Figura 5, en la cual las dos secuencias temporales muestran variaciones en el tiempo, al seleccionar la trayectoria F , donde se elimina esas discrepancias temporales, se obtienen sumas más bajas en $S(F)$. Una vez que todas las diferencias originadas por las deformaciones temporales han sido corregidas, es lógico suponer que se identificará la trayectoria óptima F_{opt} y que la suma resultante $S(F_{opt})$ representará la distancia “real” entre ambas secuencias.

Lo mencionado anteriormente sirve como fundamento para la siguiente definición de la medida de similitud DTW entre dos series, A y B :

$$DTW(A, B) = \left\{ \frac{S(F)}{\sum_{k=1}^{k=K} d(C_k)w_k} \right\}$$

El denominador se incluye con el propósito de contrarrestar el impacto de la cantidad de puntos K presentes en cada trayectoria F .

2.2 TRABAJOS RELACIONADOS

El software SciMAT permite generar un informe completo con archivos *HTML (HyperText Markup Language)* de los documentos encontrados en la búsqueda realizada en el ítem 1.4. En este informe, entre otras cosas, se encuentran la cantidad de artículos relacionados a cada temática, por ejemplo, con *Internet Of Things* existen 27 artículos. El proceso anterior se realizó a lo largo del periodo seleccionado (2017 a 2022) y como resultado se obtuvo 65 artículos relacionados con las tendencias de los últimos 5 años, los datos se muestran en la Tabla 1.

Tabla 1. Tendencias y cantidad de artículos

Tendencias	Número de artículos
<i>Internet-Of-Things</i>	27
<i>Sensors</i>	16
<i>Accelerometer</i>	5
<i>Wearable</i>	6
<i>Exercice</i>	8
<i>Smartphone</i>	2

De los 65 artículos obtenidos, se hizo una revisión basada en su título y *abstract* (resumen) dejando así 26 artículos de interés. Después de una lectura detallada, fueron seleccionados 7 artículos que presentaron mayor relevancia considerando el problema atacado en esta investigación. Los artículos seleccionados, se escogieron principalmente a aquellos que buscan el desarrollo de sistemas haciendo uso de dispositivos *wearable*, enfocando sus análisis en aspectos claves que están estrechamente relacionados con la presente propuesta de trabajo de grado.

2.3 ARTÍCULOS SELECCIONADOS

A continuación, se plasman las síntesis de estos trabajos de investigación:

Algorithm for the Comparison of Human Periodic Movements Using Wearable Devices.

El presente artículo se enfoca en la enseñanza y aprendizaje de la motricidad en una modalidad virtual, por lo cual propone un algoritmo en el que se comparan dos movimientos periódicos, el que realiza un experto y el que lo imita, es decir, un aprendiz esto con el fin de determinar qué tan semejantes se realizan estos dos movimientos y retroalimentarse de ellos [43]. El algoritmo parte de la captura de datos a través de un dispositivo *wearable*, el cual está compuesto por un sensor inercial, para este caso un acelerómetro; los datos del experto y el aprendiz son capturados en un conjunto de datos de pasos de baile de salsa. Dichos datos son ajustados en función de las iteraciones de Pearson, en seguida realiza un proceso de sincronización, filtrado, normalización, *DTW (Dynamic Time Warping)*, regresión lineal y análisis de error, con el fin de hacer la comparación correspondiente de los dos conjuntos de datos. Con lo anterior, es posible determinar si los ciclos de las dos señales coinciden y en qué medida los movimientos del aprendiz se asemejan a los del experto.

Wearable muscle movement information measuring device based on acceleration sensor.

El siguiente trabajo consistió en diseñar un dispositivo *wearable* utilizando sensores de tipo inercial, para este caso un acelerómetro con el fin de obtener información de los movimientos generados por los músculos de un deportista, durante el ejercicio se realiza la recopilación de datos; el preprocesamiento de datos, la extracción y selección de características, el entrenamiento del modelo de clasificación y la evaluación de esta información muscular del movimiento se realiza mediante algoritmos de segmentación y detección de movimiento de transición ya que estos tienen la capacidad de segmentar de manera efectiva las acciones de transición incluidas en la secuencia de aceleración y, al utilizar este método para clasificar y reconocer nueve tipos de información de movimiento humano, lo cual permite una tasa de reconocimiento promedio del 98,56 % [45], cabe resaltar que toda este dispositivo está soportado en el marco del *IoT* para la transmisión de los datos y poder ser procesados en un ordenador.

An indoor positioning method integrating WiFi and wearable inertial navigation module.

Este artículo propone un método de posicionamiento en interiores, su diseño cuenta con una red *WiFi* y un módulo de navegación inercial *wearable*, permitiendo el posicionamiento de navegación estimado para peatones (PDR) [46]. Está diseñado de forma tal, que el celular no influya en la captura de los datos y utiliza un algoritmo bayesiano ponderado, el cual proporciona la posición inicial para el sistema PDR, el método es enfocado netamente en la posición de peatones, no evalúa sus movimientos, sin embargo, utiliza una conexión *WiFi* constante.

Remote healthcare monitoring using wearable IoT devices and cloud services.

El sistema de Telemedicina propuesto está basado en *Internet of Medical Things (IoMT)* y puede ser utilizado como método alternativo a la visita al médico. El objetivo principal es reducir las interacciones cara a cara, liberar más tiempo y recursos para quienes lo necesitan con urgencia. Se plantea una interfaz que conecta de forma remota al paciente y al médico [47], este mide y carga regularmente las lecturas del paciente en una base de datos, que el médico puede revisar de forma remota para decidir acciones a tomar de acuerdo al estado del paciente.

Un aporte de este sistema es el uso de los sensores que recopilan y envían datos a través de Internet utilizando el módulo *WiFi* conectado a un controlador de nodo. Además, permite almacenar los datos en una base de datos para poder realizar análisis con estos.

Cabe resaltar que, este trabajo no se centra en datos correspondientes a sensores inerciales para evaluar gesticulaciones o simples movimientos.

Real-Time monitoring of Athletes' training data based on wireless sensors.

El artículo hace un enfoque a los sistemas compuestos por sensores inerciales controlados en el marco del *IoT* a través de redes de área local *WLAN (Wireless Local Area Network)*, dichos sensores de inercia se usan de manera integral y brindan predicciones de cantidades, por ejemplo, excitador, más velocidad y campo atractivo, al tiempo que brindan pruebas de datos de orientación, cuaterniones y puntos de delineado. El trabajo realizado en este artículo [48] es un ejemplo de esta predicción, donde es posible investigar la visibilidad de inercia de los sensores, como se muestra al evacuar a dos competidores. Con esta información y las respectivas revisiones de los maestros se consigue preparación precisa para los atletas y poder explotar al máximo sus facultades sin exceder sus límites.

Wearable sensor validation of sports-related movements for the lower extremity and trunk.

El artículo propone una alternativa a la captura de movimiento óptico 3D, esta es un sistema *IMU (Inertial Measurement Unit)* contra un sistema de captura de movimiento óptico en el cual se realizó una prueba con cuarenta y nueve adultos sanos, donde estos realizaron seis movimientos comunes a una variedad de aplicaciones deportivas (cortar, correr, saltar, sentadillas con una sola pierna y giros cruzados) mientras estaban equipados simultáneamente con marcadores retrorreflectantes estándar y un sistema IMU inalámbrico [49]. Se calcularon el sesgo, el error cuadrático medio *RMSE (Root Mean Square Error)*, la precisión y el error absoluto medio *MAE (Mean Absolute Error)* para comparar los dos sistemas en las articulaciones de las extremidades inferiores y el tronco en todos los planos de movimiento y para todas las actividades, al ser un artículo que compara dos métodos de captura de movimiento no va más allá de una comparación por lo que carece de preprocesamiento de datos y por ende no aprovecha al máximo los datos obtenidos por el sistema IMU.

Sliding variance and data range for lightweight sports activity recognition with fusion of modalities.

La investigación de artículo desarrolla un sistema ágil innovador [50] basado en detección de actividad deportiva utilizando datos de series temporales provenientes de los sensores inerciales, en este caso el acelerómetro integrado en dos modalidades, el teléfono inteligente y el reloj inteligente. Esta investigación está enfocada en los ejercicios de peso ligero, tales como, saltos, flexiones, sentadillas y sentadillas con salto.

Es un sistema basado en extracción de datos de series temporales, donde se proponen dos técnicas nuevas de extracción de características; varianza deslizante y el rango de datos. El fin de estas técnicas es mejorar las deficiencias como la precisión y el tiempo de procesamiento.

Las técnicas propuestas para mejorar las deficiencias son efectivas, pero estas mejoras solo son a nivel *software*. Es evidente que carecen de mejoras a nivel *hardware* como hacer el uso de dos sensores y ponderar sus resultados, esto también ayudará a disminuir las deficiencias.

Parámetros principales comunes a tener en cuenta Tabla 2.

Tabla 2. Parámetros principales a tener en cuenta

Artículo	Parámetros				
	App móvil	Wearable	WBAN	Coordinación óculo-manual u óculo-pédica	Entorno de Autoaprendizaje
Algorithm for the Comparison of Human Periodic Movements Using Wearable Devices	×	✓	✓	✓	×
Wearable muscle movement information measuring device based on acceleration sensor	✓	✓	×	✓	×

An indoor positioning method integrating WiFi and wearable inertial navigation module	✓	✓	✓	×	×
Remote healthcare monitoring using wearable IoT devices and cloud services	×	✓	✓	×	×
Real-Time monitoring of Athletes' training data based on wireless sensors.	×	✓	✓	✓	×
Wearable sensor validation of sports- related movements for the lower extremity and trunk	×	✓	✓	✓	×
Sliding variance and data range for lightweight sports activity recognition with fusion of modalities.	×	✓	×	×	×

El empleo del software *SciMAT* para analizar tendencias y patrones en la literatura científica ha posibilitado la identificación y organización de documentos relevantes acerca de temas de investigación vinculados a dispositivos *wearables* para el entrenamiento deportivo, sistemas basados en aplicaciones móviles y algoritmos de comparación de señales. Estos hallazgos han sido fundamentales en el desarrollo del sistema propuesto en este trabajo.

Para asegurar la identificación precisa de las bases del conocimiento científico buscadas por el diseñador, se requiere llevar a cabo una revisión manual de los documentos obtenidos a través del análisis con *SciMAT*. Este proceso permitirá seleccionar de manera efectiva la información relevante y necesaria para el desarrollador.

Después del análisis realizado con *SciMAT* y la revisión manual de los artículos recopilados, se determina que el sistema debe basarse en tres etapas: *Wearable*, Servidor y Aplicación Móvil. En la etapa de *Wearable* se establece el diseño e implementación del hardware y software del dispositivo portátil. De manera similar, en la etapa de Servidor se selecciona el software que se utilizará para desarrollar la lógica encargada de procesar las solicitudes provenientes del dispositivo portátil y la Aplicación Móvil. Por último, en la etapa de Aplicación Móvil se define el diseño y el software que permitirán la interacción del deportista con el sistema.

En última instancia, se destaca una brecha de investigación significativa centrada en la integración de datos multimodales para alcanzar una comprensión más completa de los movimientos humanos. Este enfoque implica la combinación de información de acelerómetros con otros sensores como giroscopios. Además, se identifica una brecha en la investigación relacionada con el desarrollo de algoritmos específicos destinados a abordar tipos de movimientos más especializados, como movimientos articulares específicos o gestos más precisos. La validación de estos algoritmos en contextos prácticos del mundo real, como su aplicación para mejorar el rendimiento deportivo o facilitar la rehabilitación, también representa un área de investigación esencial. Otra brecha se encuentra en la exploración de métodos de aprendizaje automático más avanzados para la detección y comparación de patrones temporales en movimientos humanos, subrayando la necesidad de enfoques innovadores y eficaces en esta área. Por consiguiente, este trabajo de investigación aborda estas cuestiones.

CAPÍTULO 3: DESARROLLO DE ARQUITECTURA

En este capítulo se presenta muestra como fue desarrollada la arquitectura a partir de un modelo de 4+1 vistas que resume el funcionamiento y despliegue del sistema. Como componentes de alto nivel se obtiene que el proyecto se puede dividir en tres partes generales: el componente wearable, el componente de server y el componente app.

3.1 ARQUITECTURA CON BASE EN EL MODELO 4+1.

Como punto de partida para el desarrollo de este trabajo de grado se proponen tres componentes:

El componente wearable, el componente server y el componente app como se observa en la Figura 6. De estos componentes, se estableció una arquitectura de software desarrollada mediante el modelo 4+1 vistas [51].

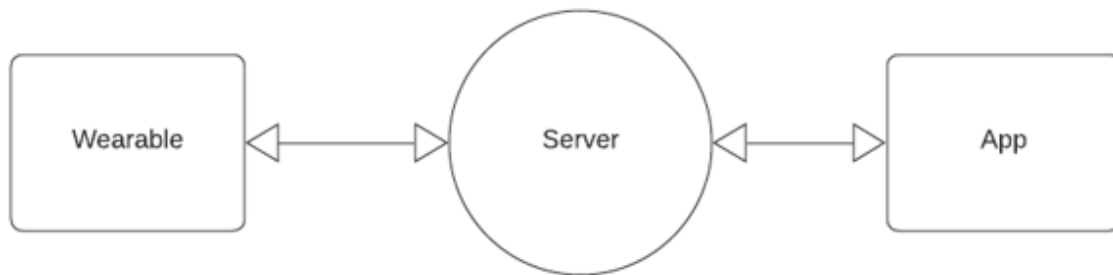


Figura 6. Componentes del sistema. Fuente: Elaboración propia

Componente Wearable:

El componente *Wearable* tiene la responsabilidad de adquirir los datos del movimiento realizado por el deportista utilizando sensores inerciales como el acelerómetro y el giroscopio. Estos datos se envían al componente *Server* a través de una conexión *WiFi*. Además, el componente *wearable* también recibe instrucciones del componente *Server* para iniciar o finalizar la captura de los datos. Por otro lado, el componente *Server* recibe solicitudes del componente *App*, las cuales posibilitan la interacción del deportista con el sistema.

Componente Server:

El componente *Server* recibe los datos recolectados por el componente *Wearable* a través de la tecnología *WiFi*. Luego, se realiza el preprocesamiento de los datos, se implementa el algoritmo de comparación correspondiente y, finalmente, se almacena la evaluación obtenida. Además, este componente procesa las solicitudes generadas por el deportista a través de su interfaz, lo que permite una interacción ágil entre los componentes del sistema y el deportista.

Componente App:

El componente App desempeña el papel de interfaz de deportista. A través de una conexión *WiFi*, envía y recibe datos desde el componente *Server*, lo que le permite tener control sobre la captura de movimientos realizada por el componente *Wearable*, así como visualizar la evaluación respectiva.

Arquitectura

Como se expone en el planteamiento del problema en el capítulo 1, la enseñanza, aprendizaje y evaluación de gestos técnicos motrices en un entorno virtual se llevan a cabo de manera general a través de sistemas complejos y costosos, que requieren que el deportista esté acompañado por un equipo multidisciplinario de profesionales. Por otro lado, existen aplicaciones móviles disponibles, pero su enfoque se limita principalmente a la hipertrofia muscular o la resistencia, dejando de lado la evaluación adecuada de los gestos técnicos. Ambos escenarios no permiten una evaluación correcta.

Análisis de requisitos

En este caso, se busca que el sistema a implementar permita la evaluación del movimiento basado en la coordinación óculo-manual u óculo-pédica realizada por el deportista sin la necesidad de la participación de un tercero. El objetivo es mejorar la enseñanza y el aprendizaje de este movimiento. Con este fin, se plantean los siguientes requisitos:

- El sistema debe capturar los datos que describen el movimiento realizado por el deportista.
- El sistema debe enviar los datos a un servidor.
- El sistema debe almacenar los datos en la base de datos que se encuentra en el servidor.
- El sistema debe comparar los movimientos enviados por el deportista con los datos almacenados en el sistema, los cuales corresponden a movimientos realizados por deportistas expertos.
- El sistema debe mostrar al deportista los resultados (evaluación) de la comparación realizada.

Arquitectura de alto nivel

La propuesta arquitectónica de alto nivel se presenta en la Figura 7. En esta se ilustra a un deportista utilizando el dispositivo *Wearable*. El dispositivo *Wearable* captura los datos generados a partir de los movimientos y los envía a través de **WiFi**. Estos datos son recibidos por el componente *Server*, donde se lleva a cabo el preprocesamiento, el uso del algoritmo de comparación y el almacenamiento de la evaluación. El componente *App* se comunica a través de **WiFi** con el componente *Server* para activar la captura de movimientos del dispositivo *Wearable* y visualizar la evaluación basada en esos datos. Además, el componente *App* genera un punto de acceso (*Mobile*

Hotspot) que permite al dispositivo *Wearable* conectarse mediante *WiFi* al componente *Server*. Con base en esta descripción, se procede a diseñar una arquitectura basada en el modelo 4+1 vistas.

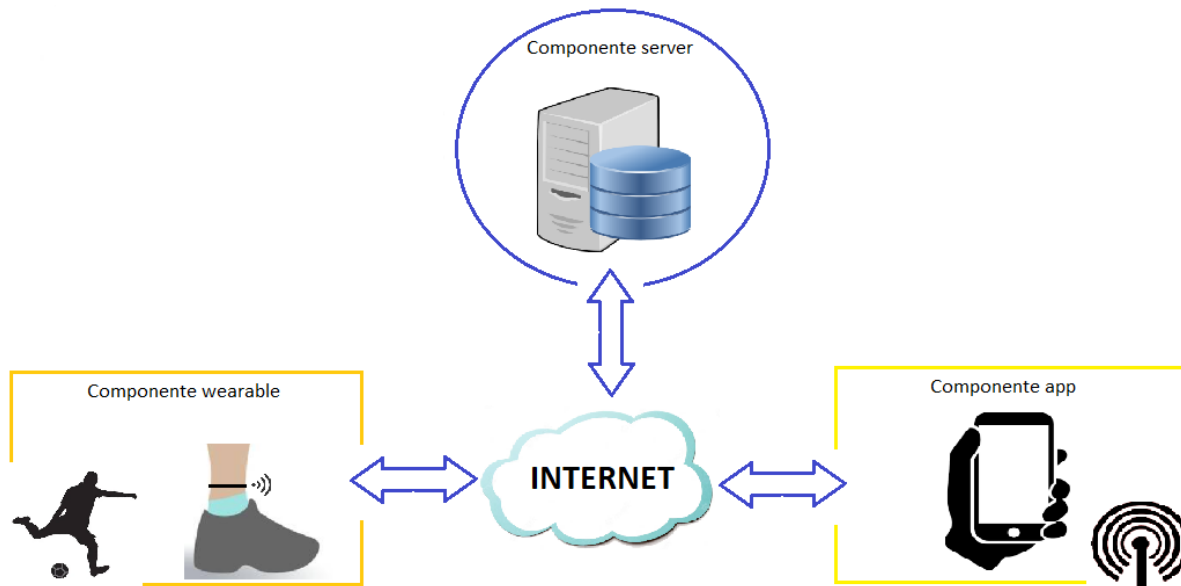


Figura 7. Arquitectura de Alto Nivel. Fuente: Elaboración propia

Arquitectura basada en el modelo 4 + 1 vistas

El modelo 4+1 vistas se utilizan para describir la arquitectura del sistema basado en el uso de múltiples puntos de vista.

Vista de escenarios

En la Figura 8 correspondiente al modelo de caso de uso del sistema, se resume el funcionamiento del sistema. En ella se presentan cuatro actores: el deportista, el componente *Wearable*, el componente *Server* y el componente *App*. El diagrama muestra que el componente *Server* posee los datos y los utiliza de acuerdo a la selección realizada por el deportista a través del componente *App*. Además, el deportista utiliza el dispositivo *Wearable* para registrar los datos generados a partir de sus movimientos al realizar el gesto técnico motriz. Una vez que los datos del deportista son registrados, el componente *Server* lleva a cabo la comparación (evaluación) y almacena los resultados en dicho componente, los cuales son visualizados posteriormente por el deportista a través del componente *App*.

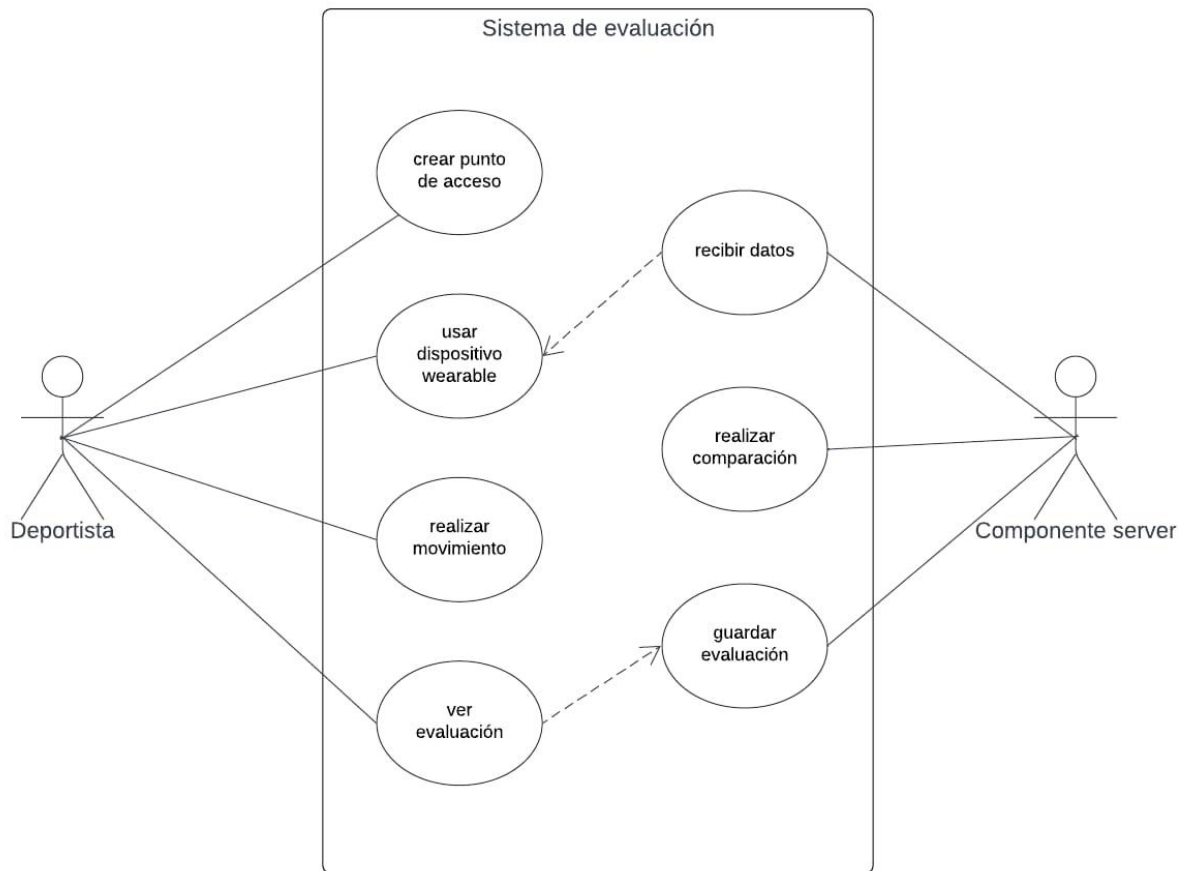


Figura 8. Casos de uso del sistema. Fuente: Elaboración propia

Esta vista muestra la funcionalidad del sistema a través de las figuras correspondientes al Diagrama de Secuencia del Sistema (Figura 9) y al Diagrama de Clases del Sistema (Figura 10). El diagrama de secuencia ilustra que el sistema se inicia cuando el deportista genera un punto de acceso (*Hotspot Mobile*) y enciende el dispositivo *Wearable* desde su dispositivo móvil. Esto permite que el dispositivo *Wearable* comience a capturar los movimientos. Los datos capturados son enviados desde el dispositivo móvil al componente *Server*, donde son comparados con un conjunto de datos obtenidos de un deportista experto. Por último, se almacena la comparación realizada y se muestra el resultado al deportista.

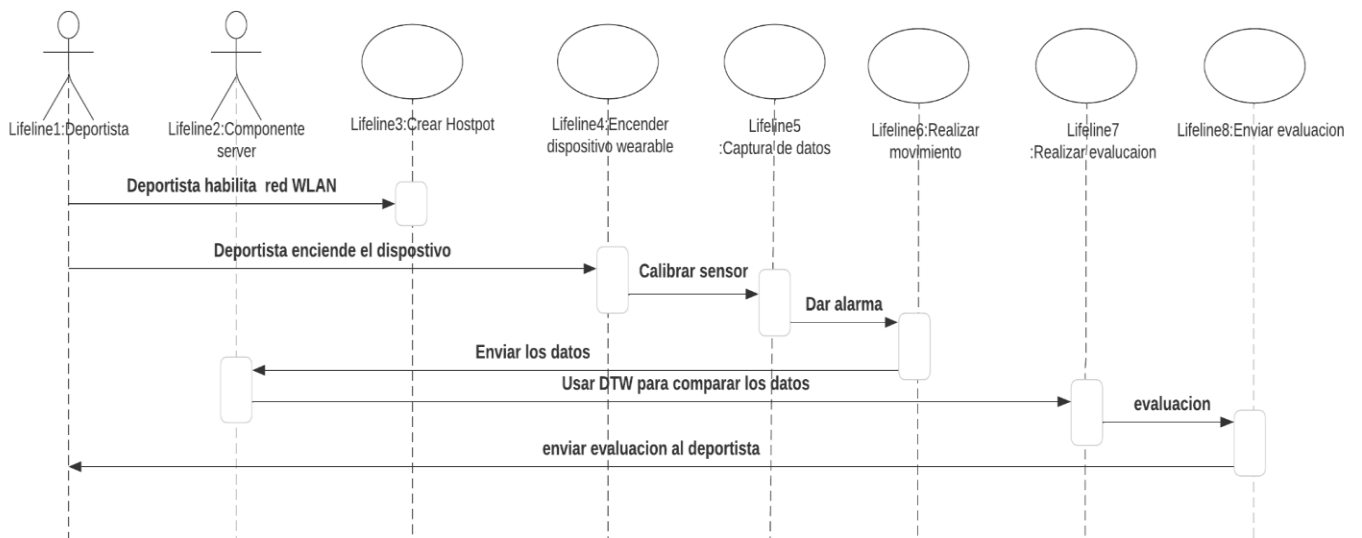


Figura 9. Diagrama de secuencia. Fuente: Elaboración propia

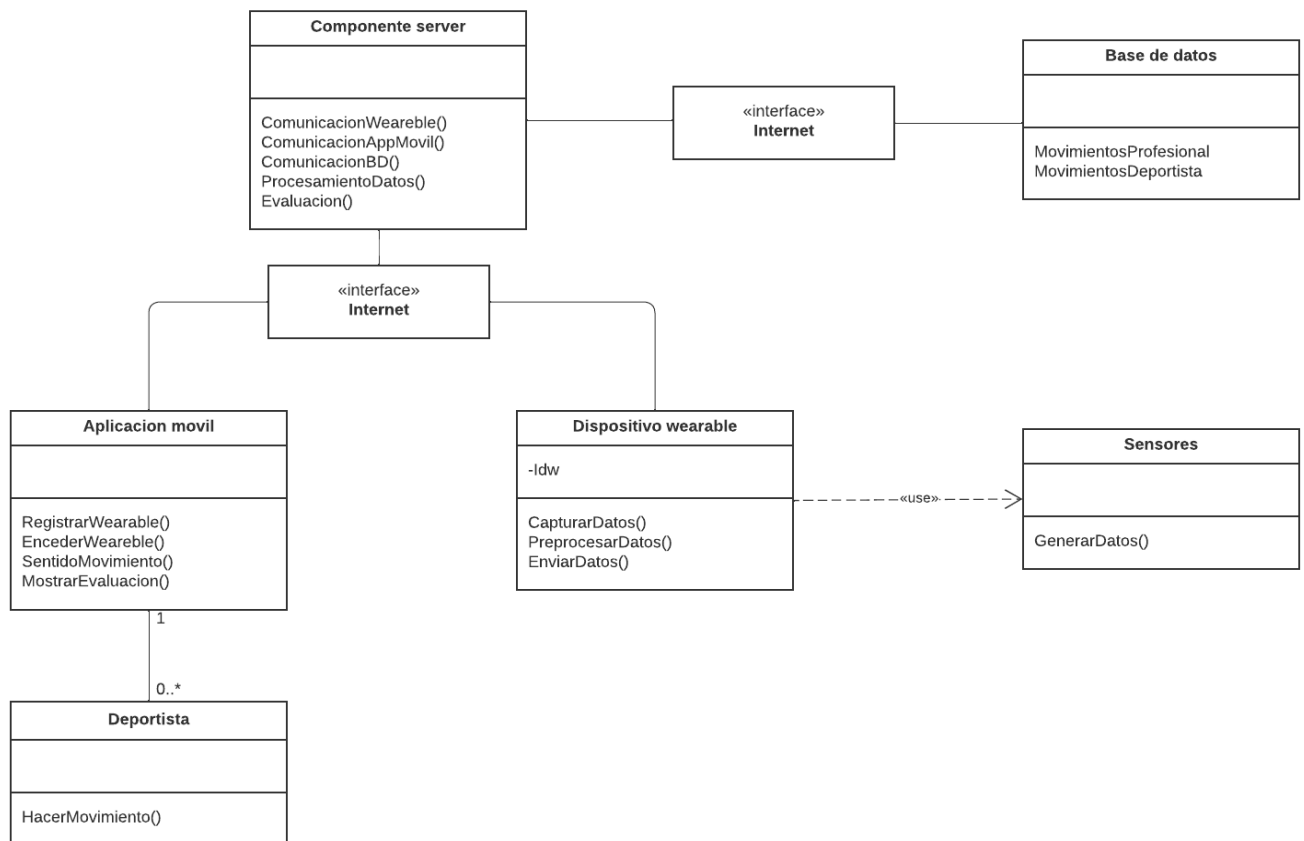


Figura 10. Diagrama de clases del sistema. Fuente: Elaboración propia

Vista de desarrollo

En la figura correspondiente al modelo de clases del sistema (Figura 11), se pueden observar las clases en tres componentes principales: el componente *Wearable*, el componente *Server* y el componente *App*. El componente *Wearable* cuenta con sensores como el acelerómetro y el giroscopio, los cuales generan datos a partir del movimiento del cuerpo humano. Además, este componente incluye un dispositivo programable que permite realizar un preprocesamiento de los datos obtenidos, para luego ser enviados al componente *Server* mediante un módulo *WiFi*.

Por su parte, el componente *Server* es responsable de recibir y gestionar las solicitudes tanto del componente *Wearable* como del componente *App*. Al recibir los datos provenientes del componente *Wearable*, realiza el procesamiento respectivo y lleva a cabo la comparación con el dataset compuesto por los datos del deportista experto almacenados en archivos (.npy). Posteriormente, el componente *Server* almacena los resultados obtenidos y los envía al deportista. Además, este componente también recibe solicitudes por parte del deportista, lo que le permite iniciar la captura de datos del dispositivo *Wearable*.

Finalmente, el componente *App* proporciona una interfaz para el deportista, donde interactúa con el componente *Server* ejecutando instrucciones para el componente *Wearable* y visualizando la comparación realizada en el componente *Server*. La comunicación entre los tres componentes se realiza a través de *WiFi*.

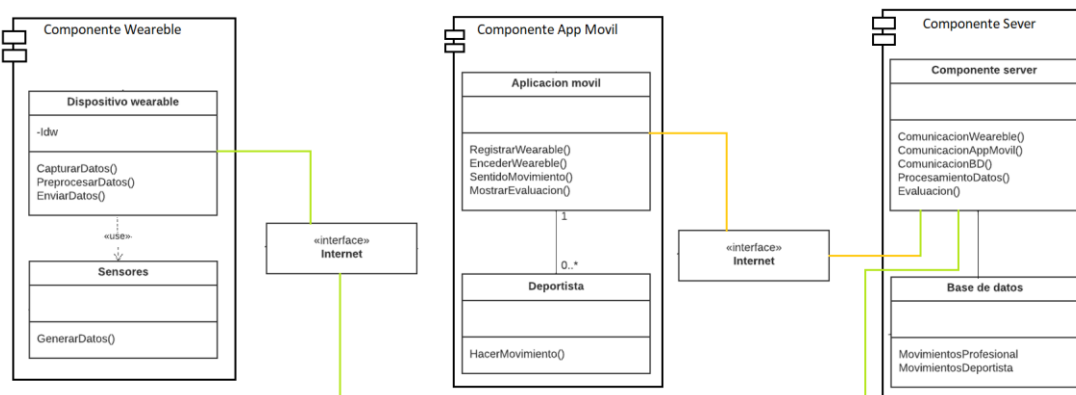


Figura 11. Vista del proceso. Fuente: Elaboración propia

En el caso del diagrama de actividad representado en la Figura 12, se presentan los aspectos dinámicos del sistema. El flujo comienza cuando el deportista enciende el dispositivo *Wearable*. A continuación, el componente *Server* recibe los datos capturados por el dispositivo *Wearable*, realiza la comparación correspondiente, almacena los resultados obtenidos y los envía al componente *App*. Finalmente, estos resultados son visualizados por el deportista.

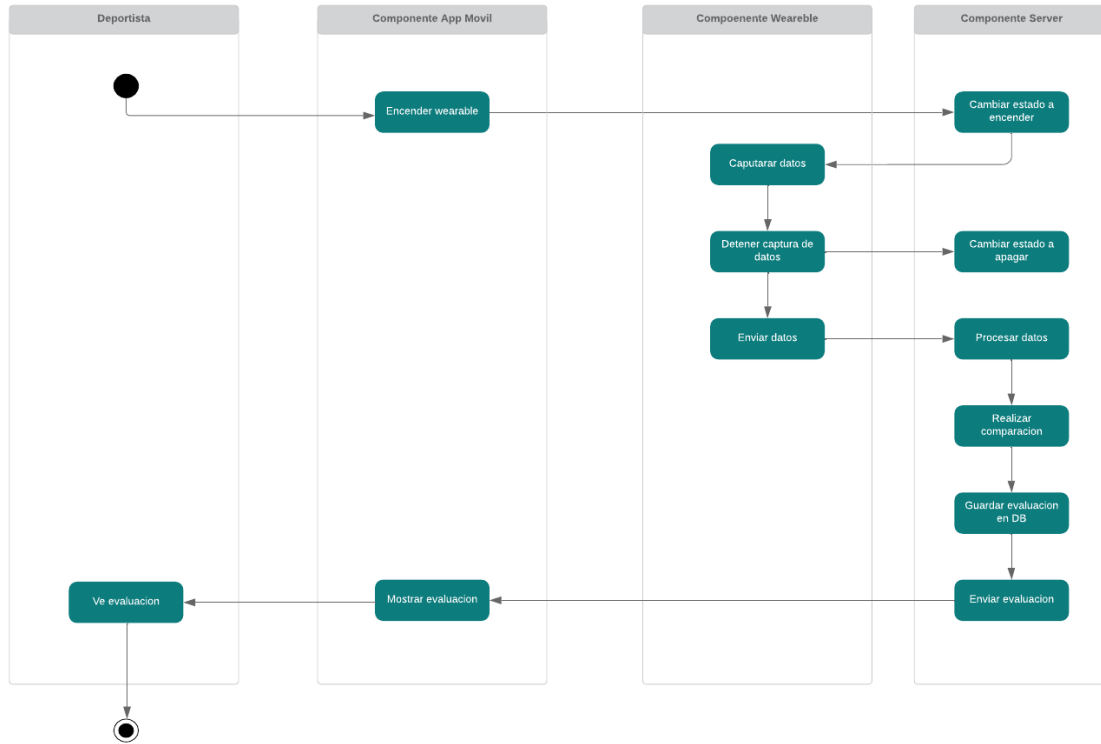


Figura 12: Diagrama de actividades. Fuente: Elaboración propia

La figura correspondiente al Diagrama de Implementación (Figura 13) muestra la representación física de la implementación del sistema. En dicho diagrama, se observa que el dispositivo *Wearable* se comunica a través de *WiFi* con el componente *Server*. A su vez, el componente *Server* se comunica mediante *WiFi* con el componente *App*, el cual es una aplicación desplegada en un dispositivo móvil, como un teléfono inteligente. Esta aplicación permite mostrar los resultados de la comparación, es decir, la evaluación obtenida.

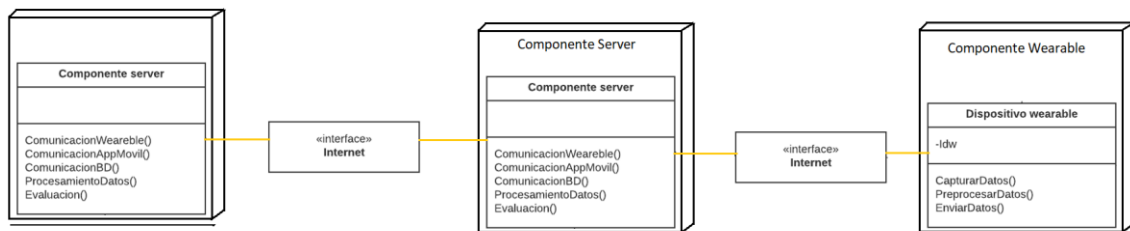


Figura 13. Diagrama de Implementación. Fuente: Elaboración propia

3.2 PLANTEAMIENTO INICIAL DE LA SOLUCIÓN

Visto desde el diagrama de componentes, el diagrama de despliegue y los componentes del sistema, se plantean tres módulos independientes e interoperables. En primer lugar, se busca el desarrollo de un dispositivo *Wearable* que cumpla con los requerimientos establecidos en esta sección. Este dispositivo consta de la capacidad de capturar datos generados a partir de los movimientos del deportista. Luego, se lleva a cabo el desarrollo de un servidor capaz de recibir y realizar el preprocesamiento de los datos obtenidos del dispositivo *Wearable*. Una vez pre procesados los datos, se plantea el algoritmo mediante el cual se analizan y se realiza la comparación (evaluación). Finalmente, se desarrolla una aplicación móvil que permite la interacción del deportista con el servidor y el dispositivo *Wearable*, permitiendo visualizar el resultado de la comparación (evaluación).

CAPÍTULO 4: DESARROLLO WEARABLE, APLICACIÓN MÓVIL Y SERVIDOR

En este capítulo se realiza el desarrollo del dispositivo *wearable*, la aplicación móvil y el servidor. Además, se establece la interoperabilidad de los tres módulos desarrollados, con el objetivo de obtener un prototipo funcional del sistema a nivel de recolección, filtrado y preprocesamiento de datos.

4.1 DEFINICIÓN DE REQUISITOS

El diseño del dispositivo con base a la arquitectura propuesta busca la captura de movimientos del cuerpo humano donde se requiere que el dispositivo logre capturar datos generados a partir de movimientos de deportistas y se espera que lo haga a lo largo de 9 segundos tomando 600 muestras. El resumen de los requerimientos se encuentra en la Tabla 3.

Tabla 3. Definición de Requisitos

POBLACIÓN OBJETIVO	
<i>Ítem</i>	<i>Características</i>
Edad	Personas entre 18 y 80 años
Peso	Entre 50 y 100 kilo Gramos
Estatura	Mayor a 1.50 metros
Limitaciones	Ninguna
Sexo	Masculino y Femenino

ACTIVIDADES FÍSICAS A ESTUDIAR	
<i>Movimiento</i>	<i>Características</i>
Extremidades del cuerpo humano y zona abdominal o lumbar	Movimientos realizados y recopilados por el dispositivo. Se espera ubicar el dispositivo en las extremidades inferiores, puede ser en pierna izquierda o derecha para este caso se dejan abiertos a criterio del caso de estudio.

VARIABLES ESPERADAS DE LA ACTIVIDAD FISICA

<i>variables</i>	<i>Descripción</i>
Aceleración	Se requiere que el dispositivo realice medición de la aceleración
Velocidad Angular	Se requiere que el dispositivo realice medición de la velocidad angular
Interfaz gráfica	Se requiere que el dispositivo tenga una interfaz gráfica que permita acceder a los datos y mediciones realizadas
Almacenamiento	Se requiere que el dispositivo envíe los datos que son generados al server en el cual se almacenaran.
Pre-procesamiento	Se requiere que el dispositivo envíe los datos que son generados al server en el cual se realice un procesamiento de la información generada y almacenada para que luego se pueda analizar de una forma profunda.
Epoch	En este caso se propone que el prototipo tenga un tiempo de uso para captura de datos de 9 segundos, pero cada ciclo será definido acorde al algoritmo que será propuesto posteriormente. En este caso, para el prototipo tendrá un epoch de 9 segundos.

4.2 REVISIÓN DEL ESTADO DEL ARTE

Este apartado se refiere a realizar una inspección de los dispositivos disponibles en el mercado que puedan ser utilizados para el desarrollo tecnológico, cumpliendo con los requerimientos establecidos anteriormente. Para ello, se investigan los sensores necesarios para capturar los movimientos de manera sencilla y eficiente, y se considera qué dispositivos pueden emplearse para este propósito. Se considera que esta actividad puede realizarse utilizando acelerómetros, magnetómetros y/o giroscopios. Un acelerómetro es capaz de medir la aceleración generada por el movimiento. Un magnetómetro puede medir los campos magnéticos presentes en el espacio. Por último, un giroscopio mantiene la misma orientación y detecta los cambios en su eje, proporcionando la posición exacta en relación con un punto de referencia. En nuestro caso, hemos optado por utilizar las mediciones de aceleración y orientación para la captura de datos, y, por lo tanto, pretendemos emplear dispositivos que contengan acelerómetros y giroscopios para el desarrollo. La Tabla 4 resume los dispositivos explorados y seleccionados con este propósito.

Tabla 4. Dispositivos Explorados

<i>Nombre</i>	<i>Tipo</i>	<i>Características</i>	<i>Procesador y memoria</i>	<i>Conectividad</i>	<i>Batería</i>	<i>Programación</i>

Microsoft Band	Commercial Band	accelerometer, GPS, Gyroscopy, HR Sensor, UV Sensor, Skin temperature, Sensor Capacitive, Barometer.	1x Cortex M4 a 0.2 GHz	Bluetooth 4.0 (low energy)	Dual 100mAh rechargeable lithium-ion polymer batteries	Android, IOS, Visual studio
SAMSUNG Gear Fit	Commercial Band	GPS, HR, Accelerometer, Gyroscope, Barometer	Gear S2 SoC	Bluetooth 4.2	200mA	Tizen STUDIO
Garmin Smart Watch (vivoactive)	Commercial Smart Watch	Accelerometer GPS Barometer Compass HR Sensor	N/A	Bluetooth 4.0 (low energy)	200mA	Linux VIRB (IOS, Android, Mac, Windows)
SparkFun's NRF52832 Breakout+IMU	Printed Circuit Board	IMU: Accelerometer Magnetometer Gyroscope	ARM Cortex-M4F	Bluetooth low energy ANT Nordic's proprietary 2.4GHz ultra low-power wireless communication	External 3,3V	Arduino Android C
ESP32 + MPU6050	programmable board	External IMU: Accelerometer and Gyroscope	Tensilica Xtensa 32bits LX6 hasta 240MHz.	Wi-Fi: 802.11b/g/n/e/i (802.11n @ 2.4 y 5 Ghz hasta 150 Mbit/s).	External 1000mA	C++, micro-python

4.3 ELEMENTOS SELECCIONADOS

De acuerdo con los requerimientos y el estado del arte, se ha seleccionado la ESP-32 y el sensor MPU6050 como parte del dispositivo *wearable*. Estos dispositivos cumplen con las características necesarias para la captura de datos a través del acelerómetro y el giroscopio, y al ser aptos para el desarrollo tanto de hardware como de software, pueden adaptarse a las necesidades específicas descritas anteriormente. La Tabla 5 resume las justificaciones para su selección.

Tabla 5. Elementos Seleccionados

Tecnología	Razón/justificación
Acelerómetro	El acelerómetro es una constante en casi todos los elementos dedicados al estudio de movimientos en las piernas.
Giroscopio	El giroscopio es una constante en casi todos los elementos dedicados al estudio de movimientos en las piernas.
WiFi bajo los estándares IEEE: 802.11b, IEEE 802.11g, IEEE 802.11n.	Cuentan con velocidades máximas de 11, 54 y 300 megabits por segundo y disponen de una frecuencia de 2,4 GHz, una banda casi universal de bajo consumo que los convierte en los más empleados internacionalmente
Batería de litio	las ventajas de la batería de litio se encuentran acorde a su compatibilidad con la tecnología de comunicación WiFi y con la placa programable ESP32.

4.4 DISPOSITIVO WEARABLE

Al emprender el desarrollo de estos dispositivos, es crucial considerar múltiples aspectos que requieren definición. Entre estos aspectos se encuentran la determinación de parámetros, el análisis de sensores, las tecnologías de comunicación, los algoritmos de funcionamiento y la elección de la tarjeta programable. Además, resulta imprescindible tener en cuenta la usabilidad y el sistema de alimentación, incluyendo la selección apropiada de la batería. Estos pasos se llevan a cabo con el propósito de obtener un dispositivo que satisfaga las necesidades específicas de este proyecto [43]. En este contexto, se procedió al desarrollo de un dispositivo *wearable* siguiendo la metodología propuesta para el diseño de dispositivos wearables [52].

Para el desarrollo del prototipo, se siguió la aproximación metodológica propuesta [52], que incluye un análisis de funcionalidad, consumo de batería y portabilidad ("*wearability*") para el diseño de un dispositivo *Wearable*. El modelo completo se encuentra resumido en la Figura 14.

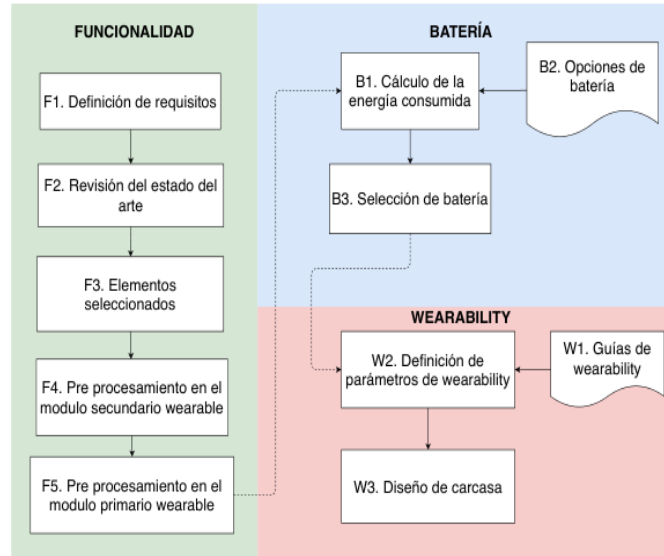


Figura 14. Funcionalidad. Fuente: Tomado de aproximación metodológica con propósito académico para el desarrollo de dispositivos wearable para extremidades inferiores del cuerpo humano [52]

Dentro del modelo propuesto, la funcionalidad constituye una categoría en la que se realiza el diseño de los módulos primarios y secundarios. Este diseño se basa en la captura de requisitos, la revisión del estado del arte y la selección de elementos pertinentes. El módulo primario tiene como función principal el procesamiento de la información, mientras que el módulo secundario se encarga de la recolección y envío de los datos al módulo primario.

4.4.1 Estructura Hardware del Dispositivo Wearable

4.4.1.1 Carcasa del dispositivo

Cuando se abordó el diseño del dispositivo wearable, fue imperativo considerar una multitud de factores críticos. Uno de estos factores clave implica reconocer que el dispositivo estará sujeto a movimientos bruscos. En respuesta a esta exigencia, se ha optado por concebir un diseño altamente funcional capaz de soportar tales movimientos sin consecuencias adversas. Este diseño se ha modelado con precisión utilizando SolidWorks y se ha materializado mediante una impresora 3D (consulte la Figura 15). Posteriormente, se emprendió una exhaustiva investigación para evaluar una variedad de materiales, con el propósito de identificar uno que garantice la integridad del wearable sin limitar la libertad de movimiento. Finalmente, se ha seleccionado un material que corresponde a un tipo de resorte ancho, el cual puede ser asegurado mediante pretinas, cumpliendo así con los rigurosos requisitos de resistencia y flexibilidad.

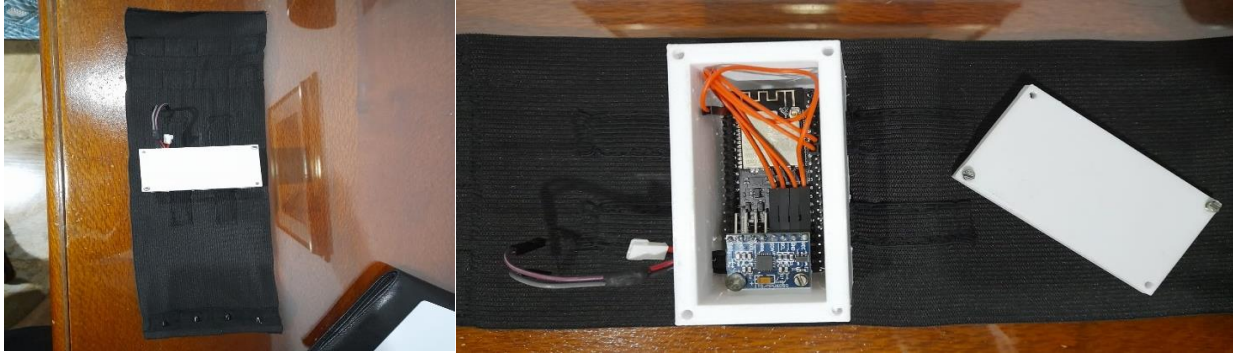


Figura 15. Estructura a nivel hardware del wearable. Fuente: Elaboración propia

Posteriormente, se procedió a establecer las conexiones necesarias entre los diversos módulos. Para llevar a cabo esta tarea, se emplearon cables de tipo jumpers, siguiendo el esquema representado en la Figura 16.

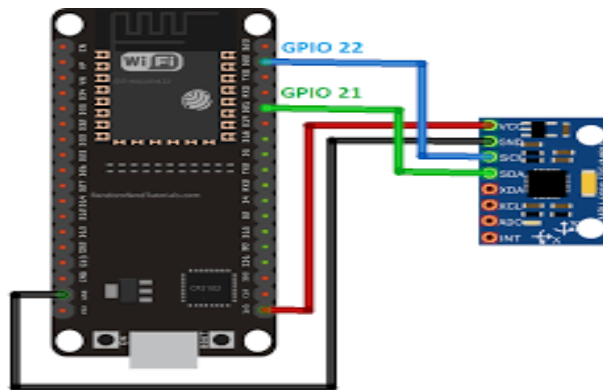


Figura 16. Esquema de conexión de la ESP32. Fuente: Elaboración propia

El resultado de este proceso es el dispositivo wearable, como se muestra en la Figura 17, el cual se puede colocar en una extremidad inferior como la pantorrilla derecha o izquierda, donde desempeñará sus funciones.



Figura 17. Wearable Ubicado en pantorrilla. Fuente: Elaboración propia

4.4.2 Pre procesamiento en el Módulo Secundario Wearable

El módulo secundario se refiere al dispositivo encargado de la calibración, recolección, codificación y envío de datos, relacionados con los movimientos del deportista. Esta unidad debe estar equipada con componentes adecuados para obtener datos a partir de los movimientos del deportista, así como tecnologías de comunicación inalámbrica y capacidad para conectarse con periféricos externos. El dispositivo debe contar con un nivel adecuado de procesamiento y tener un tamaño conveniente para no interferir con los movimientos del deportista. Para seleccionar los dispositivos adecuados para el desarrollo del prototipo propuesto, se ha realizado una revisión del estado del arte y se han evaluado diferentes opciones. Finalmente, se ha optado por los dispositivos ESP-32 y MPU6050. Para este caso, se ha establecido una frecuencia de muestreo de 66.7 Hz y un tiempo de envío de 15 ms, estos se eligieron con el fin de tener la mejor afinidad del dispositivo sin causarle algún daño a sus componentes sobrepasando sus límites. Las principales características de estos dispositivos se encuentran resumidas en la Tabla 6.

Tabla 6. Características de Dispositivos

<i>ESP32</i>	<i>Características</i>
Banda de Frecuencia	2.4GHz ISM y 5GHz
Tasa de datos en el aire	Wi-Fi 802.11b: Tasa de datos máxima: 11 Mbps Wi-Fi 802.11g: Tasa de datos máxima: 54 Mbps Wi-Fi 802.11n: Tasa de datos máxima: 150 Mbps a 2.4 GHz (1x1 SISO), 300 Mbps a 2.4 GHz (2x2 MIMO)
Modulación	BPSK, QPSK, 16-QAM, 64-QAM y 256-QAM
Micro controlador	CPU Xtensa LX6 doble núcleo, 32bits
Memoria flash	4MB
Memoria RAM	520KB
Puertos GPIO	36 a 40 configurables
Temporizadores/Contadores	3 de 16 bits, 2 de 64 bits y varios contadores de 8 bits dependiendo de cómo se configuren sus puertos.
Regulador de Voltaje	3.3 a 5 voltios

Aplicaciones	Wearable, Beacons, Periféricos de computadora, Controles remotos para televisión, Set Top Box y sistemas multimedia, Sensores deportivos y de estado físico, Sensores de salud y estilo de vida, Controladores de juegos para computadoras, Electrodomésticos inteligentes.
Periféricos	puertos Serie Universal Asíncrona por Redes de Transmisión (UART), Interfaz de Periférico Serial (SPI), Inter-Integrated Circuit (I2C), Modulación por Ancho de Pulso (PWM), Puertos de Entrada/Salida de Propósito General (GPIO) entre otros.
<i>MPU6050</i>	<i>Características</i>
Acelerómetro	Rango de medición: $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$
Giroscopio	Rango de medición: ± 250 , ± 500 , ± 1000 o ± 2000 grados por segundo (dps).
Frecuencia de Muestreo	Adaptable
Consumo de Energía	Bajo
<i>Batería de litio</i>	<i>Características</i>
Amperaje	1200mAH
dimensiones	46x25x8mm

Una vez seleccionados los dispositivos que forman parte del prototipo wearable, se procede a programarlos utilizando el entorno de desarrollo IDE de Arduino y diferentes librerías como Adafruit_MPU6050.h, Adafruit_Sensor.h, Wire.h y MPU6050. h. Estas librerías permiten crear y gestionar de manera adecuada las funciones necesarias para obtener los datos a partir de los movimientos del deportista. Además, proporcionan un amplio conjunto de funciones que facilitan el desarrollo del software para el dispositivo wearable.

A continuación, se ofrece una breve descripción del desarrollo de la aplicación, la cual se programa en la ESP-32. En la Figura 18 se muestra un diagrama de los componentes básicos que proporciona una visión general de la estructura del dispositivo wearable. Este diagrama muestra la organización de los componentes hardware, sus interfaces y las dependencias entre ellos.



Figura 18. Componentes dispositivo wearable. Fuente: Elaboración propia

El desarrollo del dispositivo wearable se programa en lenguaje de programación C++ utilizando el IDE de Arduino. Este lenguaje proporciona tanto estructuras comunes de lenguajes de alto nivel como condiciones que permiten un control a un nivel más bajo. Como todo se programa en C++, el dispositivo wearable está compuesto por un conjunto de funciones, donde la función principal `setup()` es la primera que se ejecuta. En resumen, los componentes del sistema incluyen código fuente (archivos `wearable.ino`), código de librerías con interfaz (archivos de cabecera `.h`), tablas de bases de datos, ejecutables y documentos. Además, el sistema contiene prototipos de funciones, variables globales, constantes, macros (`#define`) y directivas (`#pragma`) que forman parte de la interfaz pública del módulo.

El módulo secundario del dispositivo wearable se encarga de recolectar datos del acelerómetro y el giroscopio, y enviarlos al componente Server mediante WiFi. Este módulo se divide en tres partes principales: la definición de las librerías y variables globales, la función `setup()` y la función `loop()`.

Librerías y variables globales:

El código implementado en la ESP-32 inicia instalando las librerías esenciales para establecer la conexión WiFi y utilizar el sensor MPU6050. La totalidad de este código se encuentra adjunta en los anexos, específicamente en el apéndice D.

Las librerías necesarias para el proyecto son las siguientes:

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <list>
#include <WiFi.h>
#include <HTTPClient.h>
#include <MPU6050.h>
```

Figura 19. Librerías Requeridas. Fuente: Elaboración propia

Las librerías Adafruit_MPU6050, Adafruit_Sensor.h y Wire.h son utilizadas para aprovechar las funcionalidades del sensor MPU6050. En el entorno de trabajo, se requiere importar la librería "list" para utilizar listas. Además, se utilizan las librerías WiFi.h y HTTPClient.h para establecer la conexión WiFi y enviar y recibir solicitudes al servidor, respectivamente. Por último, la librería MPU6050.h brinda las funcionalidades necesarias para realizar el preprocesamiento de los datos.

En el proceso de preprocesamiento realizado en el dispositivo wearable, se comienza creando un objeto de tipo MPU6050. También se define una constante que representa el número de muestras utilizadas para calcular el offset. Por último, se utilizan dos variables para almacenar los valores obtenidos de cada componente del acelerómetro y giroscopio.

```
MPU6050 mpuOF;  
const int numSamples = 100;  
int16_t offsetAccX, offsetAccY, offsetAccZ;  
int16_t offsetGyroX, offsetGyroY, offsetGyroZ;
```

Figura 20. Componentes Requeridos. Fuente: Elaboración propia

Una vez importadas las librerías, se procede a definir las variables y constantes necesarias. En particular, las constantes "ssid" y "password" son fundamentales en el código del dispositivo wearable, ya que representan las credenciales necesarias para establecer la conexión a una red WiFi. En el contexto del sistema propuesto, se trata de un Hotspot. A continuación, se muestra su definición:

```
const char* ssid = "Nombre de la red";  
const char* password = "Clave de la red";
```

Figura 21. Credenciales Requeridas. Fuente: Elaboración propia

Una vez definidos los parámetros para la conexión al Hotspot, es necesario definir una variable que permita recibir el resultado de ejecutar una función que consulte en la base de datos si se inicia o no la captura de los datos, así, la variable res se encarga de esto. Además, se debe tener un valor por defecto en 0, lo cual indica que el dispositivo wearable no iniciará a tomar datos sin la instrucción dada por el deportista mediante la aplicación móvil. Se hace necesario que cada dispositivo wearable debe tener un identificador único en el sistema. Finalmente, se crean las seis listas que van a almacenar los datos recolectados de los sensores en sus tres ejes, todo lo anteriormente mencionado se observa a continuación:

```

int res;
String encender = "0";
String idw = "w003";

std::list<float> lax;
std::list<float> lay;
std::list<float> laz;
std::list<float> lgx;
std::list<float> lgy;
std::list<float> lgz;

```

Figura 22. Listado de almacenamiento. Fuente: Elaboración propia

Una vez creadas las listas, es necesario crear un iterador para cada una de ellas, lo cual nos permitirá manipular dicha estructura de datos. Además, se definen seis variables globales, una por cada componente correspondiente a los sensores acelerómetro y giroscopio. Estas variables serán utilizadas para almacenar los datos que se enviarán al servidor. Finalmente, se crea un objeto de la clase Adafruit_MPU6050, el cual nos brinda acceso a diversas funciones de la librería, incluyendo la captura de datos.

```

list<float>::iterator it_lax = lax.begin();
list<float>::iterator it_lay = lay.begin();
list<float>::iterator it_laz = laz.begin();
list<float>::iterator it_lgx = lgx.begin();
list<float>::iterator it_lgy = lgy.begin();
list<float>::iterator it_lgz = lgz.begin();

String ax;
String ay;
String az;
String gx;
String gy;
String gz;

Adafruit_MPU6050 mpu;

```

Figura 23. Captura de Datos. Fuente: Elaboración propia

Función setup():

La función setup() se ejecuta una sola vez al encender el dispositivo wearable y contiene las configuraciones necesarias para establecer la conexión tanto con el servidor como entre la ESP-32 y el sensor MPU6050. Además, se utiliza la salida de Arduino como medida de prueba para validar el proceso realizado. A continuación, se muestra la configuración correspondiente en el código:

```
Serial.begin(115200);
while (!Serial)
  delay(10);
Serial.println("Sistema IoT-WBAN");
```

Figura 24. Función Setup. Fuente: Elaboración propia

Se hace uso de la función begin() para establecer la conexión a internet mediante WiFi, utilizando las credenciales definidas previamente (ssid y password). Se implementa un bucle while() para esperar a que se establezca la conexión correctamente. Durante este proceso, se muestra en la consola el mensaje "Conectando a WiFi...". Una vez que la conexión WiFi se ha establecido con éxito, se imprime un mensaje en la consola indicando que la operación ha sido exitosa, junto con la dirección IP local asignada al dispositivo. A partir de este punto, se puede continuar con el resto de la configuración y ejecución del programa.

```
WiFi.begin(ssid, password);
Serial.print("Conectando");
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("Conexion establecida");
Serial.print("IP local: ");
Serial.println(WiFi.localIP());
```

Figura 25. Resto de Configuración. Fuente: Elaboración propia

El sensor MPU6050 puede estar en diferentes posiciones, por lo tanto, es necesario ajustar el offset en los seis componentes de los dos sensores. Esto garantiza que los datos recolectados se obtengan desde la misma posición del sensor.

Primero, se inicializan los offset que se hayan cargado en el sensor y se agrega un pequeño retraso. Luego, se establecen los offset del acelerómetro a cero. A continuación, se utiliza un bucle for() para establecer los nuevos valores de offset en función del número de muestras definido anteriormente.

Dentro del bucle for(), se acumulan los valores de offset del acelerómetro. Al finalizar el bucle, se divide cada componente por el número de muestras para obtener el valor de ajuste correspondiente para cada componente.

```
mpuOF.initialize();
delay(1000);
offsetAccX = offsetAccY = offsetAccZ = 0;

for (int i = 0; i < numSamples; i++) {
    int16_t accX, accY, accZ;
    mpuOF.getAcceleration(&accX, &accY, &accZ);
    offsetAccX += accX;
    offsetAccY += accY;
    offsetAccZ += accZ;
    delay(10);
}
offsetAccX /= numSamples;
offsetAccY /= numSamples;
offsetAccZ /= numSamples;
```

Figura 26. División de Búsquedas. Fuente: Elaboración propia

Al igual que en el caso anterior para el sensor de giroscopio, se toman sus offset y se fijan a 0, se hace un ciclo for() en el cual se toman los valores de offset para el giroscopio y se van acumulando, al finalizar el ciclo se tiene un valor de offset para cada componente.

```

offsetGyroX = offsetGyroY = offsetGyroZ = 0;
for (int i = 0; i < numSamples; i++) {
  int16_t gyroX, gyroY, gyroZ;
  mpuOF.getRotation(&gyroX, &gyroY, &gyroZ);

  offsetGyroX += gyroX;
  offsetGyroY += gyroY;
  offsetGyroZ += gyroZ;

  delay(10);
}
offsetGyroX /= numSamples;
offsetGyroY /= numSamples;
offsetGyroZ /= numSamples;

```

Figura 27. Offset. Fuente: Elaboración propia

El proceso anterior se realizó utilizando la librería MPU6050.h. A continuación, se utiliza la librería Adafruit_MPU6050.h para llevar a cabo la validación de la conexión del dispositivo wearable. Esto se realiza nuevamente utilizando la salida de Arduino para obtener información sobre el estado de la conexión.

```

if (!mpu.begin()) {
  Serial.println(" Error al encontrar el sensor MPU6050");
  while (1) {
    delay(10);
  }
}

Serial.println("MPU6050 encontrado");

```

Figura 28. Librerías. Fuente: Elaboración propia

Finalmente, se realiza un ajuste del rango de operación de los sensores, así como la aplicación de un filtro, con el fin de optimizar la funcionalidad del dispositivo.

```

mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
Serial.print("Accelerometer range set to: ");
mpu.setGyroRange(MPU6050_RANGE_1000_DEG);
Serial.print("Gyro range set to: ");
mpu.setFilterBandwidth(MPU6050_BAND_260_HZ);
Serial.print("Filter bandwidth set to: ");

```

Figura 29. Aspectos adicionales. Fuente: Elaboración propia

Función loop():

La función loop(), como su nombre indica, se ejecuta de forma repetitiva. En esta función se llaman a todas las demás funciones que permiten capturar los datos del deportista y enviarlos al servidor a través de *WiFi*. A continuación, se analizará en detalle el resultado de la ejecución de estas funciones.

Función estado():

La función estado() devuelve un valor que indica si el dispositivo wearable debe iniciar la captura de datos. El valor de retorno se asigna a la variable res. A continuación, se define una variable entera llamada m con un valor de 600, que indica el número de veces que se capturarán los datos del movimiento del deportista.

Se realiza una validación condicional utilizando el condicional “if” basado en el valor retornado por la función estado(). Si el valor es igual a 1, se ingresa al ciclo. Dentro del condicional “if”, se ejecuta un ciclo do while que llama a la función leerDatos(). Esta función captura los datos utilizando el sensor MPU6050. El ciclo do while se repetirá 600 veces. Una vez finalizado el ciclo, se sale del do while y se ejecuta la función cambiarEstado() para desactivar la captura de datos.

Después, se asignan los datos almacenados en las listas a los iteradores definidos previamente. Además, se asigna un carácter especial a las seis variables mencionadas anteriormente. Estas asignaciones se realizan con el objetivo de manipular adecuadamente los datos recolectados por el sensor y establecer valores iniciales para su posterior envío al servidor.

```

res = estado();
int m = 600;

if (res == 1) {
    do {
        leerDatos();
        m = m - 1;
    } while (m > 0);
    cambiarEstado();

    it_lax = lax.begin();
    it_lay = lay.begin();
    it_laz = laz.begin();
    it_lgx = lgx.begin();
    it_lgy = lgy.begin();
    it_lgz = lgz.begin();

    ax = '@';
    ay = '@';
    az = '@';
    gx = '@';
    gy = '@';
    gz = '@';
}

```

Figura 30. Asignaciones dadas. Fuente: Elaboración propia

Una vez obtenidos los datos y asignados a las variables iteradoras, se realiza un ciclo while que se repetirá un número de veces igual a la longitud de una de las listas que almacenan los datos en una componente de uno de los sensores (acelerómetro o giroscopio). Se selecciona la lista lax como referencia para determinar la cantidad de repeticiones del ciclo.

Dentro del ciclo while, se recorren todas las listas y se almacenan en las seis variables correspondientes los valores respectivos de cada componente. Cada valor se separa por un carácter especial "@". Este procedimiento permite tener las seis variables con los datos recolectados y separados de forma adecuada.

A continuación, se llama a la función `enviarDatos()`, encargada de enviar los datos al servidor. Por último, se realiza una limpieza de las listas, eliminando cualquier dato que pueda afectar futuras capturas de datos.

El último condicional “else” corresponde al primer “if” mencionado. Si la variable `res` tiene un valor de 0, se imprime el mensaje "Wearable apagado" y, en consecuencia, no se realizará la captura ni el envío de datos.

```
while (it_lax != lax.end()) {
    ax = ax + String(*it_lax++) + "@";
    ay = ay + String(*it_lay++) + "@";
    az = az + String(*it_laz++) + "@";
    gx = gx + String(*it_lgx++) + "@";
    gy = gy + String(*it_lgy++) + "@";
    gz = gz + String(*it_lgz++) + "@";
}

enviarDatos();

lax.clear();
lay.clear();
laz.clear();
lgx.clear();
lgy.clear();
lgz.clear();
} else {
    Serial.println("wearable apagado");
}
```

Figura 31. Funciones Adicionales. Fuente: Elaboración propia

estado():

En esta función se realiza una consulta a la base de datos, específicamente a la tabla "estado", para validar si el dispositivo wearable puede comenzar a capturar datos. Para ello, se utiliza la función `WiFi.status()` para verificar si hay conexión con el servidor. Luego, se utiliza un objeto `HTTPClient` y se llama a la función `begin()` con el parámetro que permite establecer la conexión al script "estado.php", el cual contiene la lógica para realizar la consulta en la tabla mencionada.

Para enviar la información de consulta (idw), se utiliza el método POST(). De esta manera, se establece la estructura necesaria para realizar la consulta y se maneja la respuesta del servidor, que es el valor de retorno de esta función.

cambiarEstado():

En esta función se cambia el estado del dispositivo wearable para apagarlo. Primero se verifica el estado de la conexión con el servidor utilizando la función WiFi.status(). Si la conexión está establecida, se utiliza un objeto HTTPClient para realizar una conexión al servidor y ejecutar una consulta que apaga el dispositivo wearable. Para ello, se envía el identificador del dispositivo (idw) mediante el método POST.

La respuesta del servidor se maneja de acuerdo a las necesidades, pero la función en sí no devuelve ningún valor. En resumen, esta función se utiliza para detener o apagar la captura de datos por parte del dispositivo wearable.

leerDatos():

Esta función desempeña un papel crucial, ya que se encarga de capturar los datos generados por el dispositivo wearable. En primer lugar, se definen seis variables que se utilizarán para almacenar los datos obtenidos de cada captura de movimientos. Se utiliza el objeto MPU para acceder a la función getEvent(), que permite obtener los datos del acelerómetro y el giroscopio. A partir de cada sensor, se extraen las componentes respectivas y se almacenan en las variables mencionadas anteriormente. De esta manera, se logra la captura precisa de los datos generados por el dispositivo wearable.

Además, en esta función se realiza la aplicación del offset, lo cual resulta fundamental para obtener mediciones precisas al simular la misma posición del dispositivo wearable. Esto garantiza que las variaciones en la posición del dispositivo no afecten negativamente las mediciones del sensor MPU6050. En consecuencia, se aplica el ajuste offsetAcc a cada componente del acelerómetro y el ajuste offsetGyro al giroscopio correspondiente. De esta manera, se logra compensar cualquier desviación y obtener medidas más precisas y confiables.

Los valores obtenidos anteriormente se añaden a las seis listas globales que fueron definidas al principio del script. Esto permite almacenar los datos recolectados para su posterior procesamiento. Además, con el fin de visualizar los datos recolectados por el dispositivo wearable, se imprimen por consola. Asimismo, se utiliza un delay() de 15 segundos, lo que significa que el dispositivo wearable realizará una nueva captura de datos cada 15 segundos. Esto permite establecer un intervalo de tiempo para la toma de medidas periódicas.

enviarDatos():

Esta función es responsable de enviar los datos recolectados por el dispositivo wearable al servidor. Primero se verifica la conexión utilizando la función `WiFi.status()`. Si la conexión es exitosa, se crea un objeto `HTTPClient` que se utiliza para enviar los datos recolectados mediante la función `begin()` y el método `POST`. Una vez enviados los datos, el servidor devuelve una respuesta que se maneja para verificar si el proceso fue exitoso o no. Sin embargo, esta función en sí no devuelve ningún valor.

APP MÓVIL

Para el desarrollo del componente app, se considera aspectos fundamentales de funcionamiento como:

- Diseñada para dispositivos móviles con sistema operativo Android.
- Conexión vía WiFi con el componente server.
- Intuitiva para el deportista.
- Visualizar mediante gráficas y valores numéricos el resultado de la evaluación.
- Vincular el dispositivo wearable.
- Crear el punto de acceso (*Hotspot*).
- Baja latencia.

En nuestro caso empleamos Android Studio para el desarrollo de la aplicación móvil.

- Definición de requisitos
- Revisión del estado del arte
- Tecnologías de desarrollo seleccionadas
- Librerías
- Recursos gráficos
- Librería Volley: es una biblioteca HTTP que facilita y agiliza el uso de redes en apps para Android.
- Volley ofrece los siguientes beneficios:
 - Programación automática de solicitudes de red
 - Varias conexiones de red simultáneas
 - Almacenamiento de respuestas en caché y en disco transparentes con coherencia de caché en HTTP estándar
 - Compatibilidad con la priorización de solicitudes
 - API de cancelación de solicitudes (permite cancelar una única solicitud, o bien establecer bloques o grupos de solicitudes para cancelar)
 - Personalización sencilla, por ejemplo, de reintentos o retiradas

- Ordenamiento sólido que permite completar correctamente la IU con datos recuperados de forma asíncrona de la red
- Herramientas de depuración y rastreo

Volley se destaca por sus operaciones de tipo RPC que se usan para completar la IU, por ejemplo, obtener una página de resultados de la búsqueda como datos estructurados. Se integra fácilmente con cualquier protocolo y, además, incluye compatibilidad con strings sin procesar, imágenes y JSON. Dado que proporciona compatibilidad integrada con las funciones que necesitas, Volley elimina la necesidad de escribir código estándar y te permite concentrarte en la lógica que es específica de tu app.

A continuación, se presenta la estructura del proyecto de la aplicación móvil, el código completamente detallado se encuentra en los anexos, específicamente en el apéndice A.

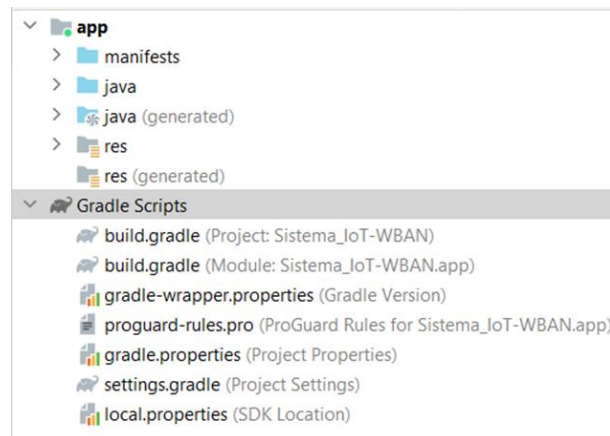


Figura 32. Estructura del Proyecto APP Móvil. Fuente: Elaboración propia

La clase HomeFragment se encargará de ejecutar realizar la evaluación por parte del deportista. Así, en esta clase deportista puede seleccionar la dirección en la que desea ser evaluado y la aplicación a modo de alarma indicará cuando el deportista puede iniciar la ejecución del movimiento. Finalmente, la aplicación debe ofrecer la evaluación del movimiento realizado.

De acuerdo a lo anterior, la aplicación deberá realizar peticiones HTTP hacia el servidor, para esto se utiliza la librería Volley y como primer paso se debe crear un objeto que permita el manejo de las solicitudes que se procesarán con esta librería.

```
RequestQueue requestQueue;
```

Luego se procede a definir las variables que reciben la selección de deportista. Se utilizan tres botones para definir la dirección que el deportista quiere que se le evalúe su movimiento. Además, se definen las variables que recibirán el resultado de la evaluación realizada, el nombre del dispositivo wearable y un efecto visual que indique al deportista que el sistema está procesando la evaluación.

```
final ImageButton imgBtnIzq = binding.imgBtnIzq;
final ImageButton imgBtnCen = binding.imgBtnCen;
final ImageButton imgBtnDer = binding.imgBtnDer;

final TextView resultado = binding.textViewResultado;
final TextView wearable = binding.textViewWearable;
final ProgressBar cargando = binding.progressBar;
```

Figura 33. Funciones de Evaluación. Fuente: Elaboración propia

La segunda parte del onCreateView está enfocada a la acción que se ejecutará de acuerdo a la orientación dada por el deportista. Así, con el evento iniciado, el sistema utiliza la función operacionGeneral() para procesarla. Esta función requiere cuatro parámetros, tres de estos son los mencionados anteriormente, en la definición de las variables y el cuarto parámetro es un valor numérico con el cual se identifica el sentido que el deportista desea ser evaluado.

```
imgBtnIzq.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { operacionGeneral(wearable,resultado, cargando, direccion: "1"); }
});

imgBtnCen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { operacionGeneral(wearable,resultado, cargando, direccion: "2"); }
});

imgBtnDer.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { operacionGeneral(wearable,resultado, cargando, direccion: "3"); }
});
```

Figura 34. Variables. Fuente: Elaboración propia

La función operacionGeneral() consta de dos partes principales, la primera válida si existe registro de un dispositivo wearable en la aplicación móvil, de ser así se procede a la segunda parte de la función, en el caso contrario, el

deportista tendrá bloqueado las funcionalidades de la aplicación hasta que registre un dispositivo wearable, este procedimiento se explicara más adelante.

```
private void operacionGeneral(TextView wearable, TextView resultado,ProgressBar cargando,String direccion){
    String idw = cargarPreferencias(wearable);
    if(idw == "wearable no registrado"){...}
    try {...} catch (Exception e){
        Toast.makeText(getActivity().getApplicationContext(), text: "Error", Toast.LENGTH_SHORT).show();
    }
}
```

Figura 35. Dispositivo Wearable. Fuente: Elaboración propia

En la imagen anterior se observa que se está utilizando la función cargarPreferencias(). Esta recibe un parámetro de tipo TextView, en el cual se cargará un mensaje de acuerdo a si tiene o no registrado el wearable. En esta función se hace uso del método getSharedPreferences(), el cual valida si existe almacenado en sus logs un valor de texto, este resultado se almacena en la variable de tipo SharedPreferences preferenes. Luego, esta variable se valida nuevamente obteniendo el string, entonces, si tiene una valor se retorna de lo contrario retorna el mensaje “wearable no registrado”, el cual lo verá el deportista y servirá para realizar la validación vista en la función operacionGeneral().

```
private String cargarPreferencias(TextView wearable) {
    SharedPreferences preferences = getActivity().getSharedPreferences( name: "idw", Context.MODE_PRIVATE);

    String idw = preferences.getString( s: "idw", s1: "wearable no registrado");
    wearable.setText(idw);

    return idw;
}
```

Continuando con la validación del registro del wearable, se toma el valor retornado por cargarPreferencias() y se valida, si no hay un wearable registrado, la aplicación mostrará esto al deportista y no le permitirá realizar la captura de movimientos.

```
private void operacionGeneral(TextView wearable, TextView resultado,ProgressBar cargando,String direccion){
    String idw = cargarPreferencias(wearable);
    if(idw == "wearable no registrado"){
        Toast.makeText(getActivity().getApplicationContext(), text: "Registre el wearable", Toast.LENGTH_SHORT).show();
        return;
    }
}
```

Figura 36. Preferencias. Fuente: Elaboración propia

Una vez se valida que el deportista tiene registrado el dispositivo wearable, se ejecutan una serie de funciones las cuales interactúan con el componente server y permiten que el dispositivo wearable pueda capturar los datos de los movimientos realizados por el deportista. La función encenderWearable() tiene dos parámetros, el primero es la url del .php que enciende en wearable, por ello es necesario identificar qué dispositivo wearable va a encender, así, el

segundo parámetro es el idw(identificador del wearable). Luego, se ejecuta la función ejecutarAlarma(), ésta indica al deportista cuando puede iniciar el movimiento, porque debe existir una pausa en la cual el dispositivo wearable se calibra. Finalmente, se tiene dos componentes, el primero un componente visual que permite al deportista verificar que el sistema está realizando todo el procesamiento para evaluar su movimiento a partir de la ejecución realizada y el segundo componente son espacios de tiempos necesarios para que el sistema pueda realizar el procesamiento anteriormente mencionado.

```
try {
    encenderWearable( URL: "http://192.168.1.6/sistemaIoTWBAN/cambiarEstado.php", idw);
    ejecutarAlarma();

    cargando.setVisibility(View.VISIBLE);

    Runnable runnable = new Runnable() {...};

    Handler handler = new Handler(Looper.getMainLooper());
    handler.postDelayed(runnable, delayMillis: 16000);

    Runnable runnable1 = new Runnable() {...};
    Handler handler1 = new Handler(Looper.getMainLooper());
    handler1.postDelayed(runnable1, delayMillis: 22000);

} catch (Exception e){
    Toast.makeText(getActivity().getApplicationContext(), text: "Error", Toast.LENGTH_SHORT).show();
}
```

Figura 37. Caso TRY de Espacios. Fuente: Elaboración propia

Como se observa en la imagen anterior, se utiliza el bloque try() catch {} esto para manejar cualquier excepción que suceda en la ejecución de la lógica de la aplicación móvil, el comportamiento exitoso fue descrito anteriormente, sin embargo, si hay una falla, se le informará al deportista mediante un mensaje.

A continuación se describe el funcionamiento detallado de la función encenderWearable(), como se mencionó anteriormente esta recibe dos parámetros la url del servidor para encender el dispositivo wearable y el idw de dicho wearable. Primero se crea la solicitud de tipo POST con la url, entonces si la respuesta del servidor es exitosa, se tendrá un mensaje “Wearable encendido” de lo contrario se mostrará el error. Una vez que maneja la respuesta del servidor, se tiene la sección en donde se envían los datos necesarios para que en el servidor se pueda procesar la solicitud, para esto se utiliza map<>, en la cual se pueden asignar los dos parámetros necesarios para encender el dispositivo wearable. Así, se observa que se crea la variable parámetros y se le asignan dos parámetros, el primero el idw con el valor del idw y el segundo encender con el valor 1, con esto se indica que wearable se encenderá. Finalmente, el request creado se agrega al requestQueue.

```

private void encenderWearable(String URL,String idw){
    StringRequest stringRequest=new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(getActivity().getApplicationContext(), text: "Wearable encendido", Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getActivity().getApplicationContext(), error.toString(), Toast.LENGTH_SHORT).show();
        }
    }){
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parametros=new HashMap<String, String>();

            parametros.put( k: "idw", idw);
            parametros.put( k: "encender", v: "1");

            return parametros;
        }
    };
    requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
    requestQueue.add(stringRequest);
}

```

Figura 38. Request agregado. Fuente: Elaboración propia

Es fundamental que la aplicación sea amigable para el deportista, por ello la aplicación cuenta con una alarma que indica el momento en el cual el deportista ya puede ejecutar su movimiento. A continuación, se observa que la función ejecutarAlarma() crea una variable MediaPlayer alarma, la cual lee el archivo de audio que se agrego para esta funcionalidad.

```

private void ejecutarAlarma(){
    final MediaPlayer alarma;
    alarma = MediaPlayer.create(getActivity(),R.raw.inicio);
    alarma.start();
}

```

Figura 39. Variable Media Player de Alarma. Fuente: Elaboración propia

La segunda parte de la aplicación móvil es el encargado de registrar el dispositivo wearable. Así, se declaran tres variables con las cuales se puede visualizar si hay un dispositivo registrado, en tal caso la funcionalidad sería actualizar el idw, otro escenario es si no hay un dispositivo wearable registrado, en tal caso se ingresa el idw del dispositivo wearable correspondiente.

Se inicia declarando tres variables, las cuales permiten visualizar si hay o no un dispositivo wearable registrado, otra variable controla el campo input y finalmente una variable que controla el evento de un botón que permite registrar o actualizar el idw según sea el caso.

```

final EditText campoIdw= binding.editTextIdw;
final Button btnregistrar = binding.btnRegistrar;
final TextView registroWearable = binding.textViewRegistroWearable;

```

Figura 40. Botón de Registro y Caso. Fuente: Elaboración propia

En la secuencia de ejecución de esta sección de la aplicación móvil, la primer función que se ejecuta en cargarPreferencias() cuyo parámetro es un TextView. El idw que se registra se almacena en las preferencias de la aplicación móvil, estas permiten almacenar pequeños volúmenes de datos con una estructura clave valor, por ello, se consulta este archivo con la clave idw, este resultado se almacena en la variable preferencias.

Una vez se tiene el resultado de la consulta anterior, se utiliza el método getString() el cual tiene dos opciones de retorno, si existe en los archivos preference con la clave idw se retorna este valor, de lo contrario se retorna el mensaje “wearable no registrado”. Cual sea el retorno se almacena en la variable idw la cual se setea al TextView registroWearable. Así, el deportista puede observar que dispositivo wearable tiene registrado o si tiene que registrar uno.

```

private void cargarPreferencias(TextView registroWearable) {
    SharedPreferences preferences = getActivity().getSharedPreferences( name: "idw", Context.MODE_PRIVATE);
    String idw = preferences.getString( s: "idw", s1: "wearable no registrado");
    registroWearable.setText(idw);
}

```

Figura 41. Wearable registrado. Fuente: Elaboración propia

El evento del botón que permite registrar el idw del dispositivo wearable tiene dos secciones fundamentales, la primera valida que el campo input no este vacio, en caso tal, se mostrara un mensaje “el idw wearable no debe ser nulo”, sino se invoca la funcion guardarIdw() que consta de dos parametros, el valor del EditText y el TextView.

```

btnregistrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(campoIdw.getText().length() == 0){
            Toast.makeText(getActivity().getApplicationContext(), text: "el idw wearable no debe ser nulo", Toast.LENGTH_SHORT).show();
        }else {
            guardarIdw(campoIdw, registroWearable);
        }
    }
});

```

Figura 42. Adicional. Fuente: Elaboración propia

La función guardarIdw() primero obtiene el valor que corresponde a la clave idw almacenada en las preferencias de la aplicación móvil, este valor se almacena en la variable preferencias. Luego, en la variable idw, se parece a string el valor que ingreso el deportista. Posteriormente, se crea un editor para el valor almacenado en la variable

preference y finalmente con el metodo put() del editor creado se fija el valor ingresado por el deportista, con la clave idw.

La segunda parte de la función limpia el campo input, visualiza en el TextView el idw que se acaba de registrar y finalmente muestra un mensaje “Wearable registrado” confirmándole al deportista que el proceso de registrar o actualizar un dispositivo wearable fue satisfactorio.

```
private void guardarIdw(EditText campoIdw, TextView registroWearable) {  
  
    SharedPreferences preferences = getActivity().getSharedPreferences("idw", Context.MODE_PRIVATE);  
  
    String idw = campoIdw.getText().toString();  
  
    SharedPreferences.Editor editor = preferences.edit();  
    editor.putString("idw", idw);  
  
    campoIdw.setText("");  
    registroWearable.setText(idw);  
    Toast.makeText(getActivity().getApplicationContext(), "Wearable registrado", Toast.LENGTH_SHORT).show();  
  
    editor.commit();  
}
```

Figura 43. Proceso de Registro de Dispositivo. Fuente: Elaboración propia

La última parte de la aplicación móvil corresponde a visualizar las instrucciones para que el deportista conozca las credenciales del hotspot que debe crear para que el dispositivo wearable se puede conectar via WiFi al sistema. Se inicia definiendo dos variables correspondientes TextView en donde se almacenarán las credenciales, de acuerdo al idw registrado y almacenado en el archivo preference. Luego, se invoca la función cargarPreferencias() cuyos parametros son las variables anteriormente mencionadas.

```
final TextView nombreRed = binding.textViewNombreRed;  
final TextView contraseña = binding.textViewContraseña;  
  
cargarPreferencias(nombreRed, contraseña);
```

Figura 44. Nombre de red y contraseña. Fuente: Elaboración propia

Como se mencionó anteriormente, en el archivo preference se busca un valor por una clave, en este caso idw, luego, con este valor se fija el nombre de la red con el valor retornado y para la contraseña es dos veces el valor retornado. En caso de que no exista un valor para la clave idw, se muestra un mensaje “wearable no registrado”.

```

private void cargarPreferencias(TextView nombreRed, TextView contrasena) {
    SharedPreferences preferences = getActivity().getSharedPreferences("idw", Context.MODE_PRIVATE);

    String idw = preferences.getString("idw", "wearable no registrado");
    nombreRed.setText("Nombre de la red: "+idw);
    contrasena.setText("Contraseña: "+idw+idw);
}

```

Figura 45. Mensaje de no Registro. Fuente: Elaboración propia

Finalmente, a continuación Figura 46, se muestra las tres interfaces que componen la aplicación móvil:

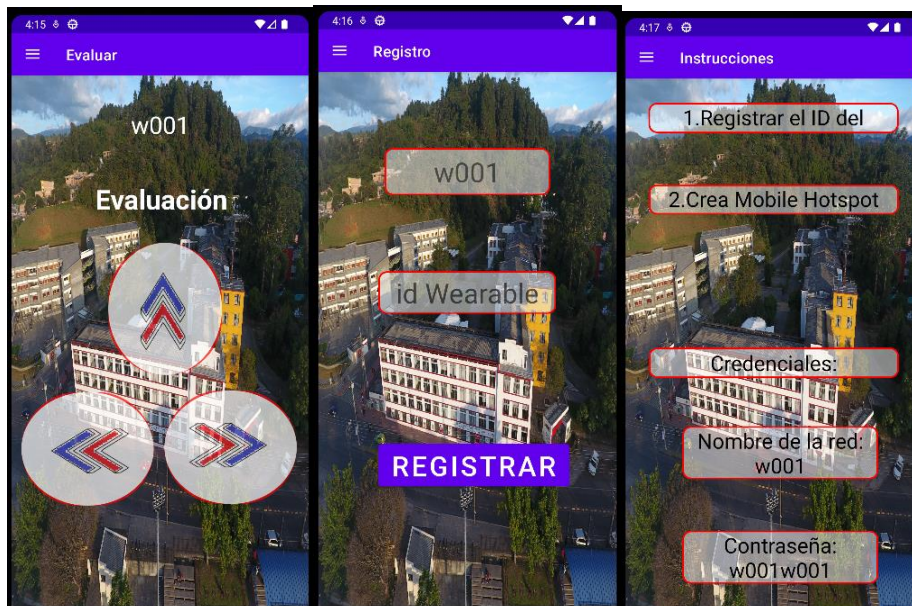


Figura 46. Interfaces de la APP Móvil. Fuente: Elaboración propia

SERVIDOR

De acuerdo a la estructura planteada, el componente server recibe, procesa y responde peticiones desde el componente app y el componente wearable mediante WiFi. Para su desarrollo se siguieron los siguientes pasos:

1. Tecnologías de desarrollo seleccionadas

Como se mencionó anteriormente XAMPP ofrece un entorno en el cual se puede desarrollar toda la lógica del lado del servidor. Se utiliza PHP para crear los scripts y MySQL como base de datos. Además, para codificar el script que realiza el análisis de los datos que permitan obtener la evaluación, se utiliza Python.

2. Definición de requisitos

El servidor debe cumplir con las siguientes funciones:

- Conexión con la base de datos: el servidor debe mantener una comunicación constante con la base de datos. Para lograr esto, se debe crear un script .php que establezca la conexión entre el servidor y la base de datos. Esto permitirá realizar consultas al servidor desde el dispositivo wearable o desde la aplicación móvil.
- Guardar datos enviados desde el componente wearable: es necesario almacenar de manera adecuada los datos recibidos del dispositivo wearable en la tabla denominada "datos_wearable" de la base de datos.
- Cambiar el estado del dispositivo wearable: se utiliza un script .php que permite cambiar el estado encendido (1) a apagado (0) o viceversa, esto con el objetivo de iniciar o detener la captura de datos por parte del dispositivo wearable, esta información se debe manejar en la tabla "estado".
- Realizar evaluación y guardar el resultado: es la función principal del componente server, aquí se utiliza un script .php para procesar la solicitud enviada desde la aplicación móvil, así, se obtiene los datos asociados a un idw(identificador dispositivo wearable), finalmente, en un script .py se realiza la evaluación y este resultado se almacena en la tabla "resultados".
- Cambiar el estado de los datos recolectados: después de realizar la evaluación, es necesario actualizar el estado de los datos recopilados desde el dispositivo wearable para evitar su uso en evaluaciones futuras. Este proceso implica la modificación de la tabla "datos_wearable" utilizando el identificador del dispositivo wearable (idw).

El código correspondiente a cada archivo que compone el componente server, implementado en el lenguaje PHP, se encuentra detallado en los anexos, específicamente en el apéndice B.

Crear la base de datos: en el servidor, es esencial almacenar los datos recolectados desde el dispositivo wearable, así como otras tablas necesarias para cumplir con los objetivos del proyecto. Por lo tanto, se procede a crear una base de datos que consta de cuatro tablas.

3. Librerías

PyMySQL es un paquete de Python que se utiliza para interactuar con bases de datos MySQL . Proporciona una forma sencilla de conectarse y ejecutar consultas en bases de datos MySQL desde aplicaciones Python. Es compatible con diferentes implementaciones de Python, como CPython, PyPy e IronPython. La instalación y el uso de PyMySQL son simples y directos, lo que lo convierte en una opción conveniente para trabajar con bases de datos MySQL en entornos Python.

4. Scripts - MySQL

conexion.php:

- El script conexion.php crea la conexión con la base de datos, requiere de cuatro parámetros para poder crear una conexión:
- User: esta variable almacena el nombre de deportista con el cual se creó la base de datos.
- pass: esta variable almacena la contraseña de la base de datos, en este caso no es necesario usarla, por lo tanto, no se colocó contraseña
- server: esta variable almacena la dirección donde está almacenada la base de datos, al utilizar XAMPP, se coloca el localhost.
- db: esta variable almacena el nombre de la base de datos.
- con: finalmente esta variable recibe el resultado de ejecutar la función mysqli_connect() que crea la conexión a la base de datos.

```
<?php
$user = "root";
$pass = "";
$server = "localhost";
$db = "sistema_iot_wban";
$con = mysqli_connect($server, $user, $pass, $db);
?>
```

Figura 47. Variable que crea la Conexión. Fuente: Elaboración propia

Este script debe ser importado en todos los scripts que necesiten conectarse a la base de datos. Al importarlo en otro script, se obtendrá un objeto de conexión que se puede utilizar para ejecutar consultas y realizar operaciones en la base de datos según sea necesario.

guardar.php

El script de guardar.php recibe los datos provenientes del dispositivo wearable. Se debe importar el script conexion.php, una vez establecida la conexión a la base de datos es posible guardar los datos recolectados por el dispositivo wearable. Estos datos se reciben mediante el método POST, son siete valores los cuales corresponden a seis listas que contienen los valores de los registros obtenidos por el dispositivo wearable correspondientes a los tres ejes del acelerómetro, tres ejes del giroscopio y el séptimo valor es el idw (identificador dispositivo wearable).

El script `guardar.php` tiene validaciones que permite certificar que todos los datos fueron recibidos por el servidor, de no ser así, la lógica del script genera error el cual es enviado a la aplicación móvil con lo cual el deportista se entera de que la captura de los datos no fue exitosa y así, pueda volver a ejecutar el ciclo.

Los datos almacenados en las seis listas están separados por el carácter especial "@" . Para poder manejar estos datos de forma individual, es necesario recorrer las listas y separarlos utilizando la función `explode()`. Este proceso nos permite obtener seis nuevas listas que contienen únicamente los registros obtenidos del sensor, los cuales están listos para ser manipulados y enviados a la tabla "datos_wearables".

Una vez obtenidas las listas, se procede a realizar el query para guardar los datos en la tabla "datos_wearables". Para esto, se utiliza un ciclo "for()" que recorre la longitud de una de las seis listas (ya que todas tienen la misma longitud). En cada iteración del ciclo, se toma cada dato almacenado en la variable "consulta" y se guarda como un registro en la tabla. Estos datos incluyen información del acelerómetro, giroscopio, identificador del wearable (idw) y estado. El campo "estado" es especialmente importante en esta tabla, ya que, si su valor es 0, indica que esos registros no han sido utilizados en el proceso de evaluación, mientras que, si su valor es 1, indica que esos datos no serán tomados en cuenta en el proceso de evaluación en curso.

Luego, utilizando la variable de conexión a la base de datos (con), se utiliza la función `mysqli_query()` con la query creada (consulta) como parámetro. Esto permite guardar los registros obtenidos del dispositivo wearable en la base de datos. El resultado de esta ejecución se almacena en la variable "resultado", la cual se valida posteriormente para verificar si el proceso se ejecutó correctamente o no.

cambiarEstado.php

Cuando el servidor recibe una petición desde la aplicación móvil para encender el dispositivo wearable, se inicia el proceso de captura de los movimientos realizados por el deportista y posteriormente se envían estos datos al servidor. Para encender el wearable, se reciben dos campos en el servidor a través del método POST: el identificador único del wearable (idw) y el campo "encender". El valor de "encender" puede ser 1 para indicar que se desea encender el dispositivo. En la petición realizada desde la aplicación móvil, este parámetro se establece como 1 para encender el wearable.

En el script `cambiarEstado.php`, se almacenan en las variables "idw" y "encender" los valores mencionados anteriormente. A continuación, se realiza una validación para verificar si estos campos tienen algún valor. En caso de que ocurra algún error durante la ejecución del script, se genera una alerta para notificar al deportista y se cancela el proceso. Sin embargo, si no hay errores, el script continúa con su ejecución normal.

Una vez validados los campos de la solicitud, ya sea desde la aplicación móvil para encender el dispositivo wearable o desde el dispositivo wearable para apagarlo, se procede a crear una consulta de actualización (UPDATE). Esta consulta se encarga de modificar el campo "encender" en la tabla "estado" del dispositivo wearable correspondiente, utilizando el valor del campo "idw" como identificador.

Una vez creada la consulta, se ejecuta mediante el objeto de conexión a la base de datos (con) para realizar la actualización.

Finalmente, se retorna la respuesta de la ejecución y se cierra la conexión.

cambiarEvaluados.php

Para almacenar los datos recolectados por el dispositivo wearable en la tabla "datos_wearables" y mantenerlos después de la evaluación, es necesario utilizar un campo adicional en la tabla para identificar qué datos han sido utilizados en el proceso de evaluación. Esto se logra mediante el campo "estado".

El campo "estado" se utiliza para marcar los datos que han sido procesados, de manera que un valor de "0" indica que los datos no han sido utilizados en la evaluación, mientras que un valor de "1" indica que los datos han sido procesados y no serán tomados en cuenta en un nuevo proceso de evaluación.

De esta manera, el campo "estado" permite mantener la tabla lista para almacenar nuevos datos y poder identificar fácilmente cuáles han sido utilizados previamente.

El script "cambiarEvaluados.php" debe incluir la importación del script "conexión.php" para establecer la conexión con la base de datos. Después de crear la conexión, se recibe el parámetro "idw" a través del método POST, que representa el identificador único del dispositivo wearable. A continuación, se realiza una consulta de actualización (Update) en la tabla "datos_wearables" para establecer el campo "estado" en 1, indicando que el registro correspondiente ha sido utilizado en el proceso de evaluación.

La ejecución de la consulta se realiza mediante el objeto de conexión "con" utilizando la función "mysqli_query()". De esta manera, se actualiza el campo "estado" en la tabla "datos_wearables" para indicar que el registro ha sido evaluado.

Es importante destacar que este proceso asume que ya se ha establecido la conexión con la base de datos y se ha validado adecuadamente el parámetro "idw" antes de ejecutar la consulta de actualización.

Finalmente, se valida el resultado del query, indicando si fue exitoso o no el cambio de estado de los registros.

consultar.php

El dispositivo wearable puede tener dos estados en relación a una actividad: encendido o apagado. Para validar este estado, se utiliza la tabla "estado". En el script "consultar.php", se recibe el parámetro "idw" del dispositivo wearable a través del método GET.

evaluacion.php

Del lado del servidor, una de las funciones principales es llevar a cabo la evaluación. Para lograr esto, se utiliza el script "evaluacion.php". A diferencia de los scripts anteriores, este no importa el archivo "conexion.php", ya que la conexión con la base de datos se realizará desde el script "evaluacion.py". En consecuencia, el script "evaluacion.php" funciona como intermediario. Recibe dos valores a través del método POST: el "idw" (identificador único del dispositivo wearable) y "dirección", que indica qué valores almacenados en el servidor se utilizarán en la evaluación. Se valida si los datos existen y, en caso afirmativo, se almacenan en las variables "idw" y "dirección" según corresponda. Finalmente, se realizan tres procesos:

- Definir un array para almacenar la respuesta que retorne al ejecutar el script "evaluacion.py".
- Ejecutar la función "exec()" con la dirección de la ubicación del script "evaluacion.py" debido a que el servidor es local, con lo anterior se puede ejecutar un "script .py" en el entorno de un script de php. Además, se pasan los datos idw y dirección para que posteriormente sean manejados según corresponda en el script "evaluacion.py", finalmente se pasa un segundo parámetro correspondiente al array creado anteriormente.
- Obtener la respuesta generada por la ejecución del script "evaluación.py". El código asociado al archivo que constituye el algoritmo DTW, ejecutado en el componente server e implementado en el lenguaje Python, está detallado en los anexos, concretamente en el apéndice C.

En el servidor se almacenará este script en cual realizará toda la lógica para determinar la evaluación del movimiento realizado por el deportista. Este se divide en dos grandes bloques principales, el primero se encarga de la gestión con la base de datos, esto incluye la conexión, consultas para obtener los datos del deportista y al final de la evaluación, insertar este resultado en la base de datos. El segundo bloque es el encardado de tomar los datos recolectados del movimiento del deportista, estos datos provienen de dos sensores y respecto a tres ejes, por lo cual se unifican en dos secuencias una para X y otra para Y dejando a Z como referencia. Una vez se tienen la secuencia_x y la secuencia_y se cargan los datos de referencia según la corresponda, estas secuencias se tienen que escalar antes de ser utilizadas en el *DTW*. Al tener dos secuencias, se obtendrán dos resultados, así, estos se unifican mediante la obtención de su magnitud, finalmente, con base a la dirección y el valor unificado se entrega una evaluación.

A lo largo de este capítulo se desarrolló un sistema destinado a la captura de datos generados por el movimiento del deportista. Este sistema consta de tres componentes principales: en primer lugar, el componente del *server*, creado mediante PHP y Python, y desplegado con XAMPP para garantizar su funcionalidad de manera eficaz. En segundo lugar, el componente de la app, diseñado en Java utilizando el entorno de desarrollo Android Studio. Por último, el componente *wearable* se ha desarrollado en C++, utilizando el entorno de desarrollo integrado (IDE) de Arduino.

CAPÍTULO 5: ALGORITMO PARA COMPARAR MOVIMIENTOS

5.1. DESARROLLO

5.1.1 Parámetros Generales

Con el objetivo de establecer una comparación entre dos individuos que ejecutan una misma habilidad motora, en la cual uno de ellos es un experto en los movimientos y el otro intenta imitarlos, es necesario obtener datos que puedan ser analizados. El artículo [52] describe el desarrollo de un algoritmo para este propósito.

Como se aprecia en el capítulo 2, estos datos pueden obtenerse a través de dispositivos portátiles o móviles, siempre y cuando no restrinjan el movimiento natural del deportista.

En este caso específico, se han seleccionado las mediciones de velocidad angular y aceleración como la fuente principal de información sobre los movimientos realizados. Los datos obtenidos se presentan en forma de series temporales, y en el caso de las mediciones de aceleración y velocidad angular que se presentan en su forma más cruda, son de naturaleza multivariable, donde se tiene básicamente el intervalo de tiempo t , los datos de la aceleración en los ejes “x”, “y” y “z” en medidas de (g) o (m/s^2) y los datos de la velocidad angular en los ejes “x”, “y” y “z” en medidas de radianes por segundo (rad/s) o grados por segundo ($^\circ/s$).

Por otra parte, en vista de que el movimiento que se requiere comparar es de una sola repetición, se optó por asignarle un rango de tiempo en el cual se asegura que dicho movimiento sea ejecutado en su totalidad. Con el fin de no despreciar ningún valor que conforma su descripción, se le asigna un registro cada 15ms. Por lo tanto, cada serie de tiempo está conformada por 600 muestras.

5.1.2 Calibración del Dispositivo

Debido a que los datos en los distintos ejes pueden fluctuar en función de la posición del dispositivo en una persona, es decir, el eje vertical del acelerómetro idealmente debería estar alineado perfectamente con la gravedad terrestre. Sin embargo, resulta complicado aplicar este parámetro en un entorno poco controlado, ya que no todas las personas podrán lograr dicha alineación, es necesario aplicar un proceso a nivel algorítmico que permite corregir este problema.

La corrección de este problema se conoce como calibración [33], para poder llevar a cabo dicho proceso es necesario trabajar con una función llamada `offset()`, esta es utilizada para calcular y aplicar una compensación de sesgo a las mediciones en tiempo real. La compensación de sesgo implica establecer un valor de desplazamiento constante a las mediciones para corregir el sesgo sistemático. A nivel matemático, la función `offset` se puede definir como:

- $medicion_calibrada = medicion_original - offset$

donde:

- `medicion_calibrada`: es el valor corregido de la medición después de aplicar el offset.
- `medicion_original`: es el valor sin corregir de la medición obtenida del MPU6050.
- `offset`: es el valor de desplazamiento o sesgo que se calcula durante el proceso de alineación.

5.1.3 Filtro complementario

Dado que el algoritmo de comparación DTW opera con señales univariantes, es importante considerar que los datos proporcionados por el sensor MPU 6050 son multivariantes, ya que consta de seis dimensiones: tres del acelerómetro y tres del giroscopio. Por esta razón, se vuelve necesario aplicar el filtro complementario [53].

El objetivo del filtro complementario es combinar los datos provenientes del acelerómetro y el giroscopio, con el fin de obtener una estimación más precisa y estable de la orientación en tiempo real del sensor. Como consecuencia de esta combinación, la cantidad de datos se reduce de seis a dos.

El funcionamiento del filtro complementario se basa en la idea de que el giroscopio es preciso a corto plazo, pero propenso a errores acumulativos a largo plazo, mientras que el acelerómetro es preciso a largo plazo, pero sensible al ruido y a las aceleraciones transitorias, entonces el filtro utiliza las mediciones del giroscopio para estimar la orientación a corto plazo y las mediciones del acelerómetro para corregir los errores acumulativos a largo plazo. La fórmula general para el filtro complementario es:

- $\text{ángulo_estimado_x} = \text{factor} * (\text{ángulo_giroscopio_x} + \text{cambio_ángulo_giroscopio_x} * \text{tiempo_delta}) + (1 - \text{factor}) * \text{ángulo_acelerómetro_x}$
- $\text{ángulo_estimado_y} = \text{factor} * (\text{ángulo_giroscopio_y} + \text{cambio_ángulo_giroscopio_y} * \text{tiempo_delta}) + (1 - \text{factor}) * \text{ángulo_acelerómetro_y}$

Dónde:

- `ángulo_estimado_x`: Es la estimación de la orientación del sensor respecto a X con Z de referencia.
- `ángulo_estimado_y`: Es la estimación de la orientación del sensor respecto a Y con Z de referencia.
- `ángulo_giroscopio_x`: Es la medición del giroscopio respecto a X.
- `ángulo_giroscopio_y`: Es la medición del giroscopio respecto a Y.
- `cambio_ángulo_giroscopio_x`: Es el cambio en el ángulo del giroscopio desde la última medición respecto a X.
- `cambio_ángulo_giroscopio_y`: Es el cambio en el ángulo del giroscopio desde la última medición respecto a Y.
- `tiempo_delta`: Es el tiempo transcurrido desde la última medición.
- `ángulo_acelerómetro_x`: Es la medición del acelerómetro respecto a X.
- `ángulo_acelerómetro_y`: Es la medición del acelerómetro respecto a Y.

- **factor:** Es un valor que determina la influencia relativa del giroscopio y el acelerómetro en la estimación final. Suele ser un valor entre 0 y 1, donde 0 significa confiar completamente en el acelerómetro y 1 significa confiar completamente en el giroscopio.

El factor de ponderación (factor) se ajusta según las características y requisitos específicos de la aplicación. Un valor más alto del factor dará más peso al giroscopio, lo que permitirá una respuesta más rápida a los cambios en la orientación, pero también puede introducir errores acumulativos a largo plazo. Un valor más bajo del factor dará más peso al acelerómetro, lo que proporcionará una corrección a largo plazo, pero puede resultar en una respuesta más lenta a los cambios en la orientación.

En resumen, el filtro complementario utiliza una combinación ponderada de las mediciones del giroscopio y el acelerómetro para obtener una estimación más precisa y estable de la orientación del sensor, aprovechando las fortalezas de cada sensor y compensando sus debilidades.

5.1.4 Análisis por DTW

La evaluación de dos secuencias temporales a través del método de DTW (Dynamic Time Warping o Alineamiento Temporal Dinámico) se fundamenta en la generación de una matriz de costos y la determinación de una trayectoria óptima a lo largo de esta matriz para cuantificar la similitud entre las secuencias temporales. A continuación, se proporciona una explicación más exhaustiva del procedimiento matemático involucrado:

Como resultado de aplicar el filtro complementario, literal 5.1.3. Se obtienen cuatro series de tiempo, donde A y B corresponden respectivamente a las muestras obtenidas del deportista y del deportista experto respecto al eje "X" y C y D corresponden respectivamente a las muestras obtenidas del alumno y del profesional respecto al eje "Y", cabe resaltar que estas con longitudes respectivas n , m , n' y m' donde $n=m=n'=m'$ y corresponden a la cantidad de muestras, revisar literal 5.1.1 de tal forma que $A = \{a_1, a_2, \dots, a_{600}\}$, $B = \{b_1, b_2, \dots, b_{600}\}$, $C = \{c_1, c_2, \dots, c_{600}\}$ y $D = \{d_1, d_2, \dots, d_{600}\}$.

En seguida se procede a la creación de las matrices de costos "Co y Co'", sus tamaños están dados por $(n \times m)$ y $(n' \times m')$, donde cada elemento $Co(i,j)$ representa la "diferencia" entre los puntos a_i de la serie A y b_j de la serie B y $Co'(i,j)$ representa la "diferencia" entre los puntos c_i de la serie C y d_j de la serie D. Estas diferencias pueden ser una métrica de distancia, como la distancia euclidiana o la distancia absoluta.

$$Co(i, j) = de(a_i, b_j)$$

$$Co'(i, j) = de'(c_i, d_j)$$

y_n	$y_n - x_1$	$y_n - x_2$	$y_n - x_3$	$y_n - x_4$	\dots	$y_n - x_{(n-2)}$	$y_n - x_{(n-1)}$	$y_n - x_n$
$y_{(n-1)}$	$y_{(n-1)} - x_1$	$y_{(n-1)} - x_2$	$y_{(n-1)} - x_3$	$y_{(n-1)} - x_4$	\dots	$y_{(n-1)} - x_{(n-2)}$	$y_{(n-1)} - x_{(n-1)}$	$y_{(n-1)} - x_n$
$y_{(n-2)}$	$y_{(n-2)} - x_1$	$y_{(n-2)} - x_2$	$y_{(n-2)} - x_3$	$y_{(n-2)} - x_4$	\dots	$y_{(n-2)} - x_{(n-2)}$	$y_{(n-2)} - x_{(n-1)}$	$y_{(n-2)} - x_n$
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
y_4	$y_4 - x_1$	$y_4 - x_2$	$y_4 - x_3$	$y_4 - x_4$	\dots	$y_4 - x_{(n-2)}$	$y_4 - x_{(n-1)}$	$y_4 - x_n$
y_3	$y_3 - x_1$	$y_3 - x_2$	$y_3 - x_3$	$y_3 - x_4$	\dots	$y_3 - x_{(n-2)}$	$y_3 - x_{(n-1)}$	$y_3 - x_n$
y_2	$y_2 - x_1$	$y_2 - x_2$	$y_2 - x_3$	$y_2 - x_4$	\dots	$y_2 - x_{(n-2)}$	$y_2 - x_{(n-1)}$	$y_2 - x_n$
y_1	$y_1 - x_1$	$y_1 - x_2$	$y_1 - x_3$	$y_1 - x_4$	\dots	$y_1 - x_{(n-2)}$	$y_1 - x_{(n-1)}$	$y_1 - x_n$
	x_1	x_2	x_3	x_4	\dots	$x_{(n-2)}$	$x_{(n-1)}$	x_n

Figura 48. Ejemplo de Matriz de Costos. Fuente: Tomado de Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

Cuando las distancias son pequeñas, las muestras son similares, y si las distancias son altas, las muestras son distantes. Con la matriz de costos se define el camino más eficiente, es decir, se busca recorrer los datos de la matriz donde el costo sea menor. En este caso, la forma es partiendo de un punto $c = (i, j)$, se desea hacer un paso k para otro punto $ck = (ik, jk)$, para lo cual se busca en sus tres lados adyacentes el valor mínimo de tal forma que vaya registrándose el mejor camino, considerando que el costo menor entre los datos representa la cercanía entre los datos en los conjuntos de datos. En caso de valores similares, el salto se realiza hacia su diagonal superior derecha. En este caso, partiendo desde la posición.

$(x_1, y_1) = (y_1 - x_1)$, que corresponde a la posición $c = (i, j)$

Sus lados adyacentes son $(x_2, y_1) = (y_1 - x_2)$, $(x_2, y_2) = (y_2 - x_1)$, $(x_1, y_2) = (y_2 - x_1)$;

Donde el salto $ck = (ik, jk)$, se hará hacia el valor mínimo de estos lados adyacentes. es decir:

$\text{Min}((y_1 - x_2), (y_2 - x_1), (y_2 - x_1))$

Se tiene un ejemplo con datos iniciales para los vectores "v" y "u", que consta de 125 muestras. En la Figura 49 se puede apreciar la trayectoria mencionada.

En este contexto, se puede observar la aproximación entre las dos señales, donde cada línea que conecta las señales representa la ruta óptima identificada en la matriz de costos, tal como se muestra en la Figura 50.

Es importante considerar que el ejemplo presentado en esta sección corresponde a las señales antes de aplicarles filtros, normalización o ajustes. Por lo tanto, es posible observar discontinuidades tanto en la dirección vertical como horizontal al inicio y al final del recorrido identificado por la matriz de costos.

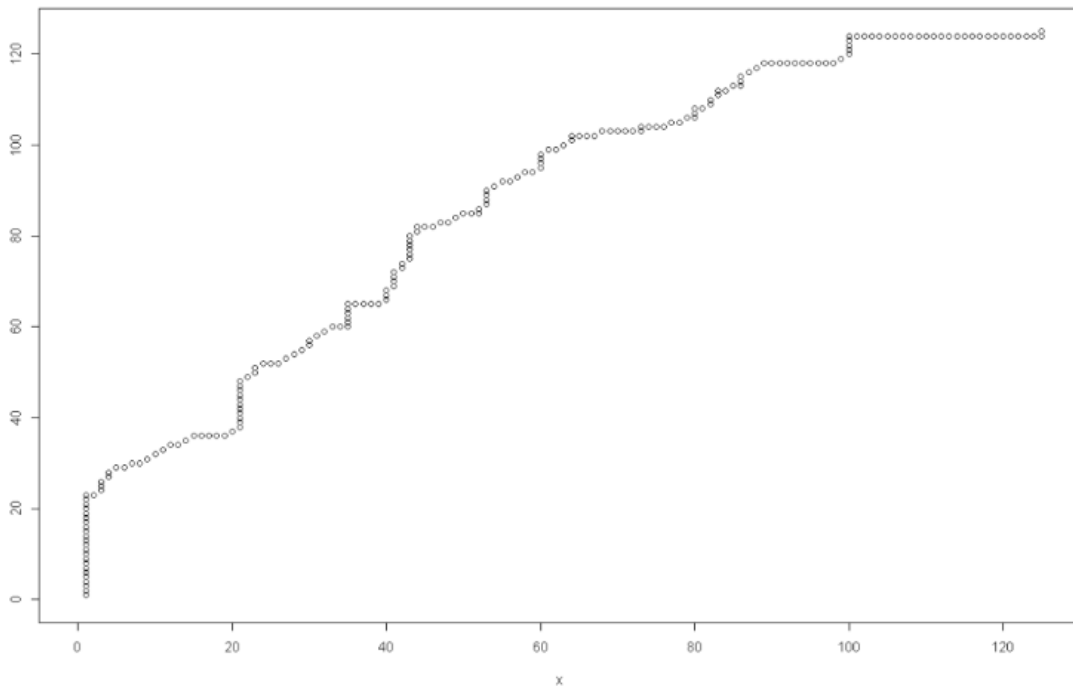


Figura 49. Ejemplo DTW – señales reales. Fuente: Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

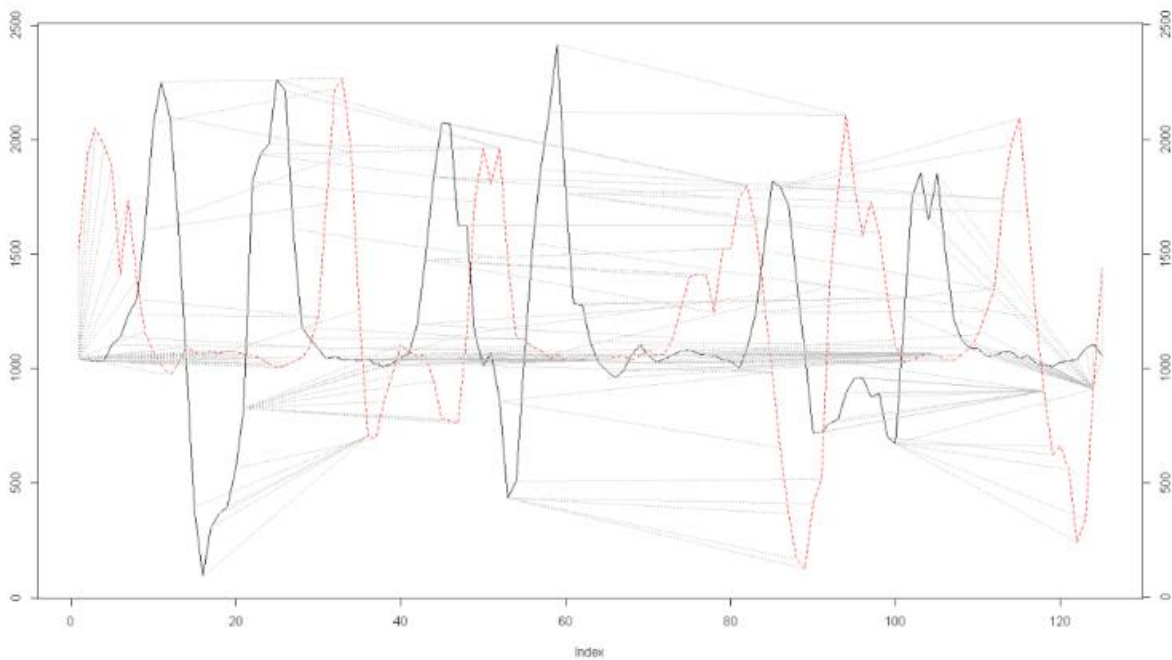


Figura 50. Ejemplo DTW – comparación de señales. Fuente: Tomado de Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

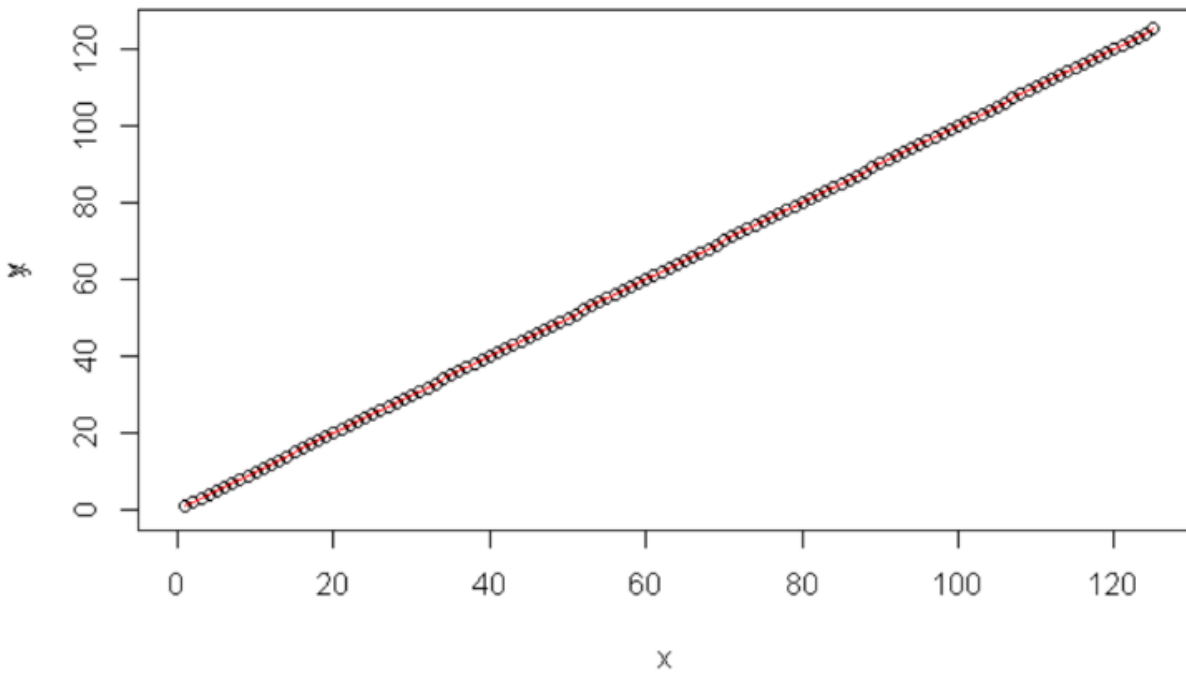


Figura 51. Ejemplo dtw - señales ideales. Fuente: Tomado de Algoritmo para la comparación de movimientos periódicos humanos en el contexto de la enseñanza y el aprendizaje de habilidades motrices [43]

5.1.5. Análisis comparativo a partir de DTW

En conclusión, se puede decir que las señales sometidas al DTW (Dynamic Time Warping o Alineamiento Temporal Dinámico) se parecen cuando el resultado del DTW arroja un valor de distancia bajo o una ruta óptima que sigue una trayectoria cercana a la diagonal principal de la matriz de costos. Aquí tienes algunos indicadores que sugieren que las señales son similares:

Distancia DTW baja: Cuando el valor de la distancia DTW calculada es bajo, esto indica que las dos señales se asemejan en gran medida. Cuanto más cercano a cero sea este valor, mayor será la similitud entre las señales.

Ruta óptima cercana a la diagonal principal: Al observar la ruta óptima trazada a través de la matriz de costos, si esta ruta se asemeja a la diagonal principal o a una línea recta, es un indicio de similitud. Esto significa que el algoritmo ha encontrado una correspondencia casi directa entre los puntos en las dos señales.

Alineamiento visual: Si al representar las dos señales en un gráfico o plano, se observa que se superponen o siguen una trayectoria cercana entre sí, esto también indica similitud.

Análisis cualitativo: Además de los aspectos cuantitativos, es importante realizar un análisis cualitativo de las señales. Si se tiene conocimiento previo sobre las características de las señales y se evidencia que comparten patrones, tendencias o comportamientos similares, entonces se puede afirmar que se parecen.

Teniendo en cuenta estas premisas que determinan el parentesco del par de señales sometidas al DTW, se procede a comparar las series $A = \{a_1, a_2, \dots, a_{600}\}$, y $B = \{b_1, b_2, \dots, b_{600}\}$, con el fin de determinar la similitud del movimiento respecto al eje “X” teniendo al eje “Z” como referencia y a comparar las series $C = \{c_1, c_2, \dots, c_{600}\}$ y $D = \{d_1, d_2, \dots, d_{600}\}$, con el fin de determinar la similitud del movimiento respecto al eje “Y” teniendo al eje “Z” como referencia.

Una vez que se han obtenido dichos valores, se procede a asignarles una equivalencia en porcentaje de manera independiente. Idealmente, un valor de 0 se interpreta como un 100% de similitud. Luego, se combina estos dos porcentajes mediante una ponderación para calcular un porcentaje total que califica la similitud entre el movimiento del deportista y el del deportista experto.

5.1.6. Diagrama de Flujo

En la Figura 52. Se tiene el diagrama de flujo el cual muestra toda la secuencia de operación del todo el sistema diseñado, a partir del componente App Móvil, se da la instrucción de encender el dispositivo wearable. La petición es gestionada por el componente server, en el cual se intenta establecer una conexión con la base de datos. En caso de que la conexión no tenga éxito, se reintenta hasta lograrla. Una vez establecida la conexión, se ejecuta la función correspondiente para cambiar el estado y encender el dispositivo wearable. Luego, se lleva a cabo el proceso de calibración, que implica el manejo del sesgo, como se describe en el literal 5.1.2 Calibración del Dispositivo. Posteriormente, el deportista realiza su movimiento, generando dos series temporales recolectadas por los sensores inerciales. Una vez completada la captura de datos, se realiza otra petición al servidor para cambiar el estado en la tabla correspondiente y apagar el dispositivo wearable.

En el dispositivo wearable, se realiza un procesamiento para transmitir los datos, junto con la función utilizada para el envío de los mismos, como se explica en el literal 4.4.2 Pre procesamiento en el Módulo Secundario Wearable. Una vez que el servidor recibe estos datos, establece una conexión con la base de datos y los guarda en la tabla correspondiente.

En el componente server, se leen los datos almacenados en la base de datos, lo que proporciona las series temporales del giroscopio y acelerómetro ya calibradas. Estas dos series temporales se procesan en el filtro complementario para fusionarlas y obtener una señal precisa, como se explica en el literal 5.1.3 Filtro complementario Posteriormente, la señal resultante se compara con la señal del deportista experto utilizando DTW, que genera una matriz de costos para estimar la evaluación, como se detalla en el literal 5.1.5. Análisis comparativo a partir de DTW. El resultado se almacena en la tabla correspondiente de la base de datos y se envía a la aplicación móvil, donde el deportista puede ver la evaluación de la ejecución de su movimiento.

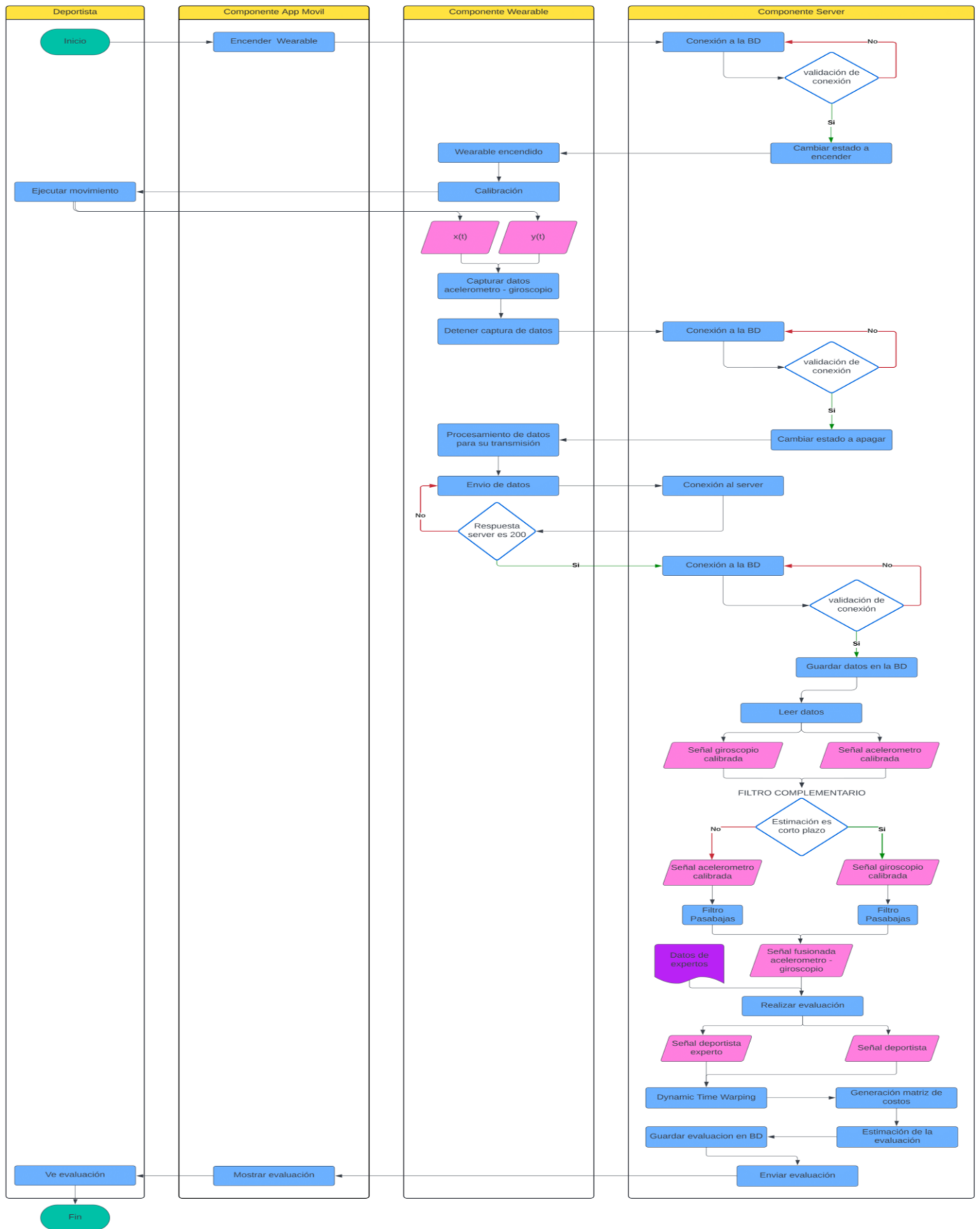


Figura 52. Diagrama de flujo del prototipo diseñado. Fuente: Elaboración propia

CAPÍTULO 6: CASO DE ESTUDIO

En el sexto capítulo, se procede a validar el algoritmo introducido en el capítulo quinto, utilizando datos recopilados con el sistema desarrollado en el capítulo cuarto, en un entorno controlado. En primer lugar, se realiza una descripción detallada de la planificación del experimento, que incluye información acerca de los participantes. A continuación, se aborda la creación del conjunto de datos y se proporciona información acerca de su disponibilidad, así como de los vídeos correspondientes, los cuales se pueden descargar desde un sitio web. Por último, se exponen los resultados del algoritmo, tanto en escenarios donde las muestras son similares como en aquellos donde difieren.

6.1 PLANTEAMIENTO DEL EXPERIMENTO

El sistema propuesto se basa en la arquitectura resumida en el capítulo 3, donde se describen los componentes: *wearable*, server y app móvil, diseñados para la recolección y análisis de datos. El componente *wearable* tiene la responsabilidad de adquirir información acerca de los movimientos del deportista a través de sensores colocados en su cuerpo. Luego, este componente transmite dicha información al servidor a través de una conexión *WiFi*. El componente app móvil, que previamente se instala en un dispositivo móvil (como una *tablet* o un *smartphone*) y se conecta al dispositivo *wearable* a través de *WiFi*, actúa como interfaz entre el *wearable* y el servidor. Es el encargado de enviar comandos al *wearable* para iniciar la recolección de datos generados por el deportista y de transmitir estos datos al servidor. Una vez que los datos se someten a un módulo de preprocesamiento, procesamiento y almacenamiento en el servidor, se utilizan para proporcionar retroalimentación al deportista. Los resultados de este proceso se reenvían al dispositivo móvil para que puedan ser visualizados en pantalla. En este escenario, se aprovecha la capacidad de procesamiento del servidor utilizado.

El funcionamiento del sistema propuesto se resume anteriormente. El proceso se inicia cuando el deportista utiliza el dispositivo portátil para seleccionar la dirección en la cual va a patear, luego, la app móvil mediante una alarma indica que ya puede realizar el movimiento. Es importante destacar que el conjunto de datos de los movimientos del profesional debe haber sido previamente cargado en el servidor. Una vez completada esta etapa, en el servidor se realizará la comparación de movimientos. Finalmente, el deportista, a través del componente de la aplicación móvil, recibirá la evaluación de su movimiento.

6.2 CREACIÓN DEL SET DE DATOS

En primer lugar, se realizó un desarrollo del dispositivo a modo de prototipo con base a una metodología propuesta para el desarrollo de dispositivos *wearable* [52]. El dispositivo *wearable* desarrollado a modo de prototipo se realizó con el uso de los dispositivos ESP32, módulo 6050 y una batería de 1200 mA, el cual es capaz de recolectar los datos a partir de un acelerómetro y un giroscopio los cuales están integrados en el módulo anteriormente nombrado. Los datos se tomaron con una frecuencia de muestreo de 66.7 Hz. Estos datos son enviados a través de *WiFi* para finalmente

ser recolectados y enviados a un server con una aplicación alojada en un dispositivo móvil, en este caso, desarrollada en un entorno Android.

El resultado proporciona información derivada de la captura de datos a través de la aceleración, inclinación y rotación generadas por los movimientos del cuerpo. Los datos recopilados muestran un comportamiento de serie de temporal multivariante. En cada intervalo de tiempo “t”, se registran los datos de aceleración en los ejes “x”, “y”, y “z”, expresados en unidades de (g) o (m/s^2), así como los datos de inclinación y rotación en los ejes “x”, “y”, y “z”, medidos en radianes por segundo (rad/s).

Para la creación del dataset, se pone a consideración un contexto de evaluación de una habilidad motriz que involucre movimientos no periódicos. Se escoge así tres estilos diferentes de tiro al arco con un balón como movimientos para la creación del dataset.

Para la captura de los movimientos, se seleccionan cinco personas: dos son deportistas expertos en el área y tres son deportistas novatos. Sus características se resumen en la Tabla 7.

Se preparó un escenario para un movimiento de tipo tiro penal, que consiste en un lanzamiento sin barrera desde una distancia prefijada de 5m (Figura 53) con la oportunidad de obtener un gol directo donde la única oposición es la del guardameta. Este proceso se divide en dos pasos: el primero es un pre-trote, y el segundo implica patear el balón al arco. En este segundo paso, los deportistas pueden elegir a qué lado del arco van a lanzar: izquierdo, centro o derecho, cabe resaltar que el remate debe realizarse con la pierna derecha.

Durante la prueba, se registraron tanto el inicio como el final de cada tiro penal. Se tomaron 600 muestras en un período de 9 segundos para cada tiro. En la Tabla 8 y Tabla 9, se encuentran los enlaces y códigos QR de acceso a los vídeos capturados durante la experiencia de los diferentes tiros penales, así como las personas que los llevaron a cabo y en la Tabla 10 se encuentra los enlaces de acceso al set de datos de cada deportista .



Figura 53. distancia entre el balón y el arco. Fuente: Elaboración propia

Tabla 7. Características de los deportistas

	<i>Deportista experto 1</i>	<i>Deportista experto 2</i>	<i>Deportista 1</i>	<i>Deportista 2</i>	<i>Deportista 3</i>
<i>Edad</i>	25	26	30	22	63
<i>Altura</i>	167cm	165cm	164cm	152cm	173cm
<i>Peso</i>	70Kg	73Kg	78Kg	60Kg	84Kg






Tabla 8. Enlaces de videos capturados a los deportistas

Remate tiro penal hacia la derecha	deportista experto 1 deportista experto 2 deportista 1 deportista 2 deportista 3	https://youtu.be/errXOqoo8d8 https://youtu.be/4vi4OVM2hyg https://youtu.be/YzR8Jx12IkM https://youtu.be/3rlMrWG6JsA https://youtu.be/KovjDFaAUnU
Remate tiro penal hacia el centro	deportista experto 1 deportista experto 2 deportista 1 deportista 2 deportista 3	https://youtu.be/BOL2beew6Pc https://youtu.be/MQPvo61ZXq8 https://youtu.be/DV55aajNUBg https://youtu.be/ndZkBgwTP7g https://youtu.be/pkzqPO2xZ4U
Remate tiro penal hacia la izquierda	deportista experto 1 deportista experto 2 deportista 1 deportista 2 deportista 3	https://youtu.be/u45226sST4c https://youtu.be/K6YaO-R3BMM https://youtu.be/Gg7PeoVVkuc https://youtu.be/KpIC5-B9Jc4 https://youtu.be/2npAgSeeTL4

Tabla 9. Códigos QR de videos capturados a los deportistas

	deportista experto 1	deportista experto 2	deportista 1	deportista 2	deportista 3
Remate tiro penal hacia la derecha					
Remate tiro penal hacia el centro					
Remate tiro penal hacia la izquierda					

Tabla 10. Tabla Repositorio de Datasets de los deportistas

<i>DEPORTISTA</i>	<i>REPOSITORIO</i>
deportista experto 1	https://github.com/camilo-burbano/Sistema_IOT_WBAN/tree/Datasets/Deportista%20experto%201 
deportista experto 2	https://github.com/camilo-burbano/Sistema_IOT_WBAN/tree/Datasets/Deportista%20experto%202 
deportista 1	https://github.com/camilo-burbano/Sistema_IOT_WBAN/tree/Datasets/Deportista%201 
deportista 2	https://github.com/camilo-burbano/Sistema_IOT_WBAN/tree/Datasets/Deportista%202 
deportista 3	https://github.com/camilo-burbano/Sistema_IOT_WBAN/tree/Datasets/Deportista%203 

6.3 RESULTADOS DEL ALGORITMO

El conjunto de datos recopilados se utiliza para realizar comparaciones entre las muestras y determinar el grado de proximidad entre ellas según el algoritmo propuesto [2]. En este contexto, se propone medir la diferencia entre las rectas como método de comparación. Las comparaciones realizadas bajo estos parámetros indicarán el nivel de similitud en los movimientos. Del conjunto de datos, se seleccionarán muestras de un grupo de deportistas expertos, que incluyen los movimientos correspondientes al deportista experto 1 y al deportista experto 2, y de deportistas no expertos, que incluyen el deportista 1, el deportista 2 y el deportista 3.

El movimiento que se abordará en este caso de estudio es el de patear un penal en fútbol. Este gesto implica un análisis detallado de varios aspectos, que abarcan la técnica, la biomecánica y la toma de decisiones. A continuación, se presenta una descripción general de los elementos clave para caracterizar este movimiento:

- **Posición del Cuerpo:** El jugador se coloca frente al balón en la posición adecuada para el disparo. Los pies deben estar cerca del balón y paralelos a la portería.
- **Selección de la Técnica:** El jugador decide qué tipo de disparo realizará, como un tiro colocado, un disparo fuerte o un disparo de efecto.
- **Paso de aproximación:** El jugador da unos pasos de aproximación hacia el balón. La longitud de esta aproximación puede variar según la preferencia del jugador.
- **Colocación del Pie:** El pie de apoyo se coloca al lado del balón, y el pie de disparo se coloca detrás del balón. La colocación del pie de disparo varía según el tipo de tiro.
- **Ángulo de Disparo:** El jugador decide el ángulo de disparo para superar al portero y evitar que detenga el balón.
- **Fuerza y Precisión:** Se aplica la fuerza necesaria al disparo mientras se busca la precisión. La parte interior del pie suele utilizarse para lograr un mayor control y precisión.
- **Efecto (si es necesario):** El jugador puede aplicar efecto al balón para hacer que se mueva de manera impredecible en el aire, dificultando la tarea del portero.
- **Factor de Distancia:** La distancia entre el punto de penalti y la portería también es un factor crítico, ya que afecta la velocidad y la colocación del disparo.
- **Elección de Lado:** El jugador decide si pateará el balón al lado derecho, izquierdo o al centro, con el objetivo de engañar al portero.

La caracterización del movimiento para patear un penalti involucra una combinación de técnica, habilidades mentales y toma de decisiones. La práctica y la experiencia son fundamentales para perfeccionar este movimiento y mejorar las posibilidades de anotar un gol. La adaptación a diferentes situaciones y la capacidad de reaccionar a las acciones del portero también son aspectos esenciales en la ejecución exitosa de un penalti.

Con base a lo anterior se plantea el caso de estudio, en el cual un deportista a través de la aplicación móvil seleccionará el lado de preferencia para rematar, seguido de esto el dispositivo *wearable* tomará los datos respectivos del movimiento mediante sensores (acelerómetro y giroscopio), luego, el servidor procesará la información y finalmente, el deportista podrá observar la clasificación respectiva del siguiente grupo: Excelente, Sobresaliente, Bueno, Regular o Malo.

La métrica previamente mencionada se fundamenta en categorías como "Excelente", "Sobresaliente", "Bueno", "Regular" o "Malo", proporcionando así una evaluación más detallada y granular en comparación con una única medida. Esta mayor granularidad posibilita una comprensión más profunda de las variaciones presentes en los resultados. Asimismo, los rangos establecidos facilitan la identificación de tendencias o patrones a lo largo del tiempo.

Para ilustrar, si los resultados se encuentran predominantemente en el rango "Malo" en varios intentos y en el rango "Excelente" en otros, esto podría revelar información valiosa sobre la variabilidad y posibles tendencias estacionales.

La aplicación de esta métrica también simplifica la interpretación al agrupar los datos en categorías más amplias y comprensibles. Este enfoque resulta especialmente útil para deportistas que buscan comprender la evaluación de sus movimientos.

Basándonos en lo expuesto, y tomando en consideración el valor resultante de la matriz de costo derivada de la ejecución del DTW, se puede generar un rango de valores mediante una técnica conocida como evolución progresiva. Esta técnica propone una métrica que aumenta de manera lineal para representar una evolución progresiva o acumulativa de un fenómeno o proceso en eventos. De esta manera, con los resultados del DTW y empleando esta métrica, se pueden generar los rangos necesarios para una evaluación más detallada y precisa.

Es necesario crear datos de referencia recopilados de los dos deportistas expertos con el propósito de establecer los rangos que se utilizarán para asignar una calificación. Dado que hay dos señales, estas se procesarán por separado, y al final se combinarán los resultados en un valor final mediante la siguiente ecuación:

$$valor_{final} = \sqrt{x^2 + y^2}$$

El valor_final se empleará como referencia para establecer los rangos correspondientes a las cinco categorías de calificación mencionadas previamente. Este proceso se realiza mediante el uso de la técnica de alineamiento dinámico de tiempo (DTW). Con esta metodología, se consideran dos series temporales que reflejan el comportamiento de los sensores (acelerómetro y giroscopio) en relación con sus ejes X y Y. Se evaluó el resultado de la aplicación de **DTW** a las series temporales generadas por dos deportistas expertos en tres tipos diferentes de tiros penales.

- Remate tiro penal hacia la derecha: En primer lugar, se analizó el comportamiento en el eje X y posteriormente en el eje Y. Se observa que el comportamiento de las señales a lo largo del tiempo, con 600 muestras, es notablemente similar. Esto era de esperarse, ya que ambos deportistas tienen un dominio técnico que les permite ejecutar el movimiento de manera óptima. Por lo tanto, el camino óptimo en ambos ejes se asemeja a una línea casi recta, como se muestra en la Figura 54 y Figura 55.

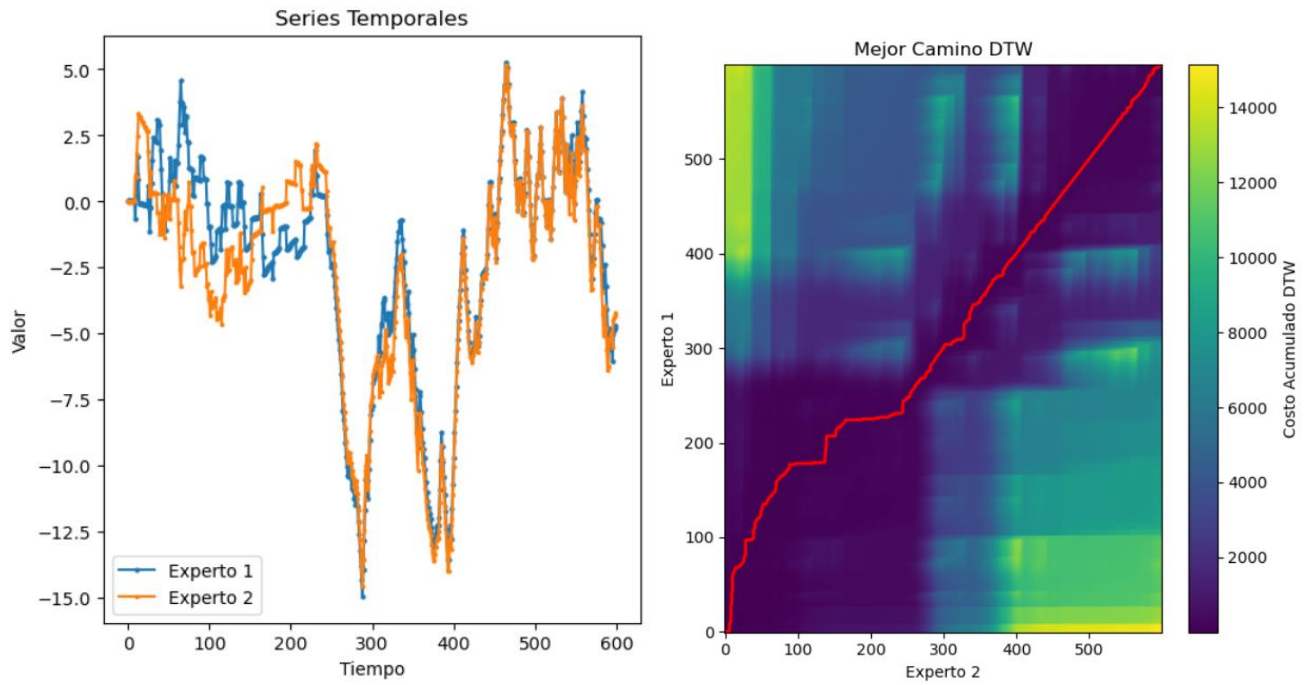


Figura 54. series temporales sobre el eje X tiro penal a la derecha. Fuente: Elaboración propia

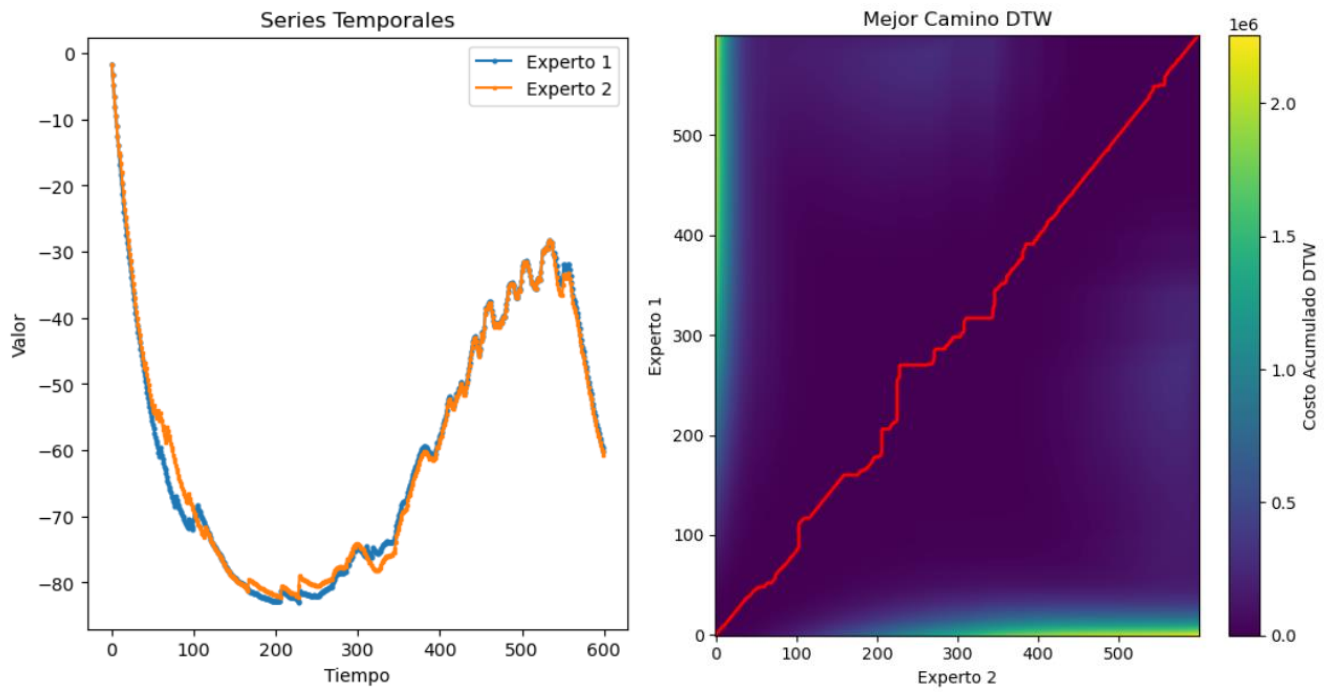


Figura 55. Series temporales sobre eje Y tiro penal a la derecha. Fuente: Elaboración propia

Al aplicar el *DTW* a las anteriores series temporales se obtienen las siguientes distancias:

Para el comportamiento en el eje X: 265.5318490921648.

Para el comportamiento en el eje Y: 220.72908011315025.

Continuando con el análisis, al aplicar la ecuación de magnitud se obtiene un valor de 345.29, el cual se puede aproximar a 345. Con este valor, se establecen los siguientes rangos:

valores = [345,690,1035,1380]

Así, “Excelente” se obtiene si el valor final es menor o igual a 345, “Sobresaliente” se obtiene si el valor final es mayor a 345 y menor a 690, “Bueno” se obtiene si el valor final es mayor a 690 y menor a 1065, “Regular” se obtiene si el valor final es mayor a 1035 y menor a 1380, finalmente si el valor final es mayor a 1380 es obtiene como evaluación “Malo”.

- Remate tiro penal hacia el centro: se realiza un análisis similar al caso anterior, con la única diferencia de que ambos deportistas expertos ejecutan el tiro penal al centro. Como resultado, se obtienen las siguientes gráficas y resultados que se muestran en la Figura 56 y Figura 57. En estas figuras, es claramente evidente una marcada similitud entre las secuencias temporales.

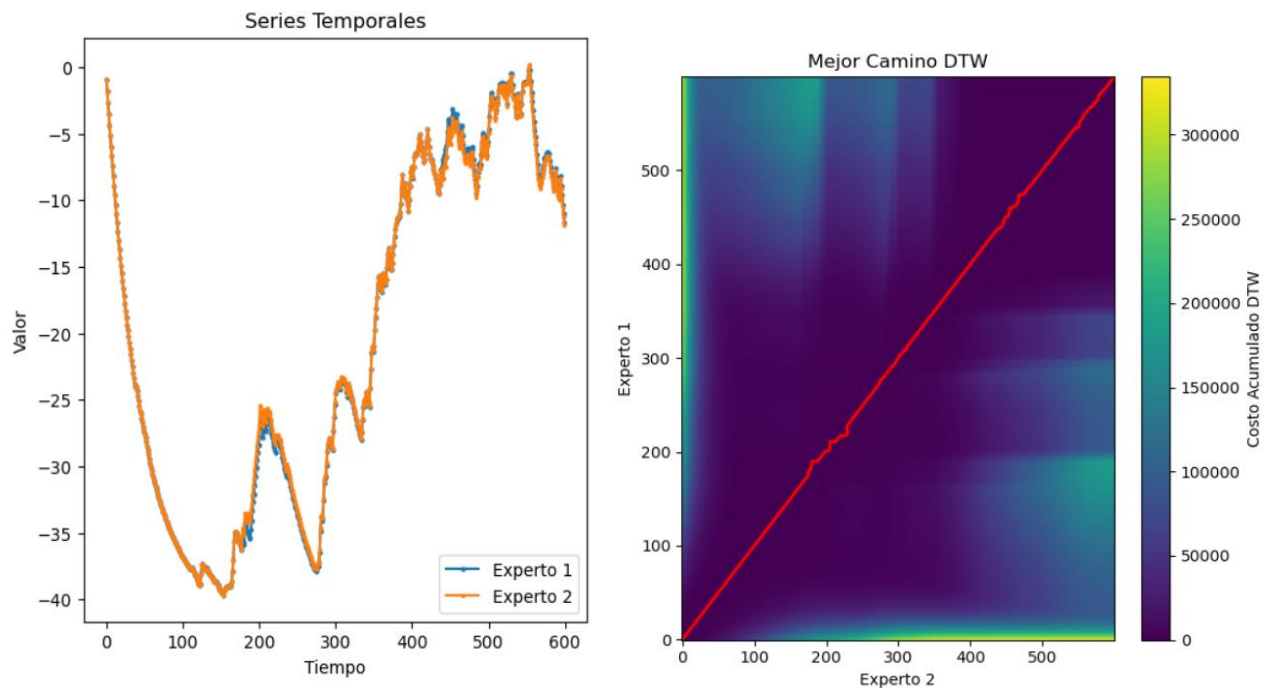


Figura 56 Series temporales sobre el eje X tiro penal al centro. Fuente: Elaboración propia

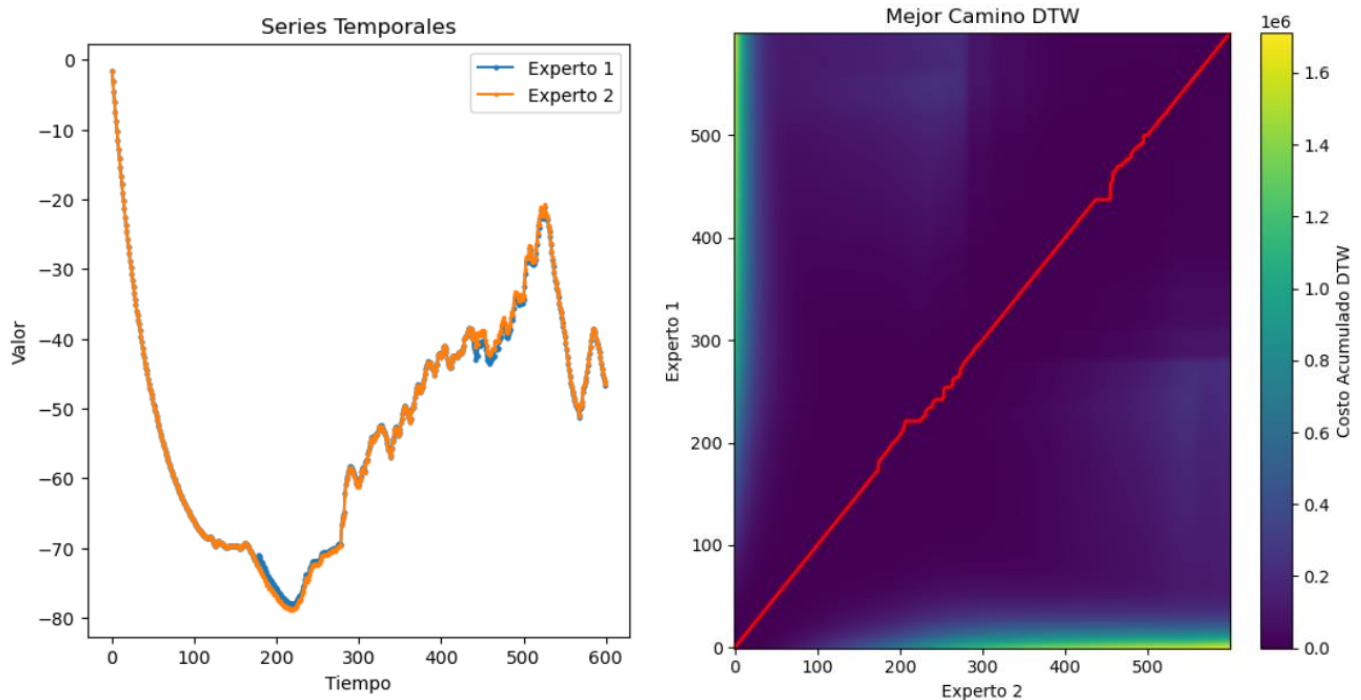


Figura 57 Series temporales sobre el eje Y tiro penal al centro. Fuente: Elaboración propia

Al aplicar el *DTW* a las anteriores series temporales se obtienen las siguientes distancias:

Para el comportamiento en el eje X: 74.43133433681257

Para el comportamiento en el eje Y: 46241359968383

Aplicando la ecuación de magnitud se obtiene un valor de 125.03, el cual se aproxima a 125, con este valor se fija los siguientes rangos:

valores = [125,250,375,500]

Así, “Excelente” se obtiene si el valor final es menor o igual a 125, “Sobresaliente” se obtiene si el valor final es mayor a 125 y menor a 250, “Bueno” se obtiene si el valor final es mayor a 250 y menor a 375, “Regular” se obtiene si el valor final es mayor a 375 y menor a 500, finalmente si el valor final es mayor a 500 es obtiene como evaluación “Malo”.

- Remate tiro penal hacia la izquierda: Se lleva a cabo un análisis similar a los casos anteriores, con la única diferencia de que ambos deportistas expertos realizan el tiro penal en esa dirección. Como resultado, se obtienen las siguientes gráficas y resultados que se presentan en la Figura 58 y Figura 59.

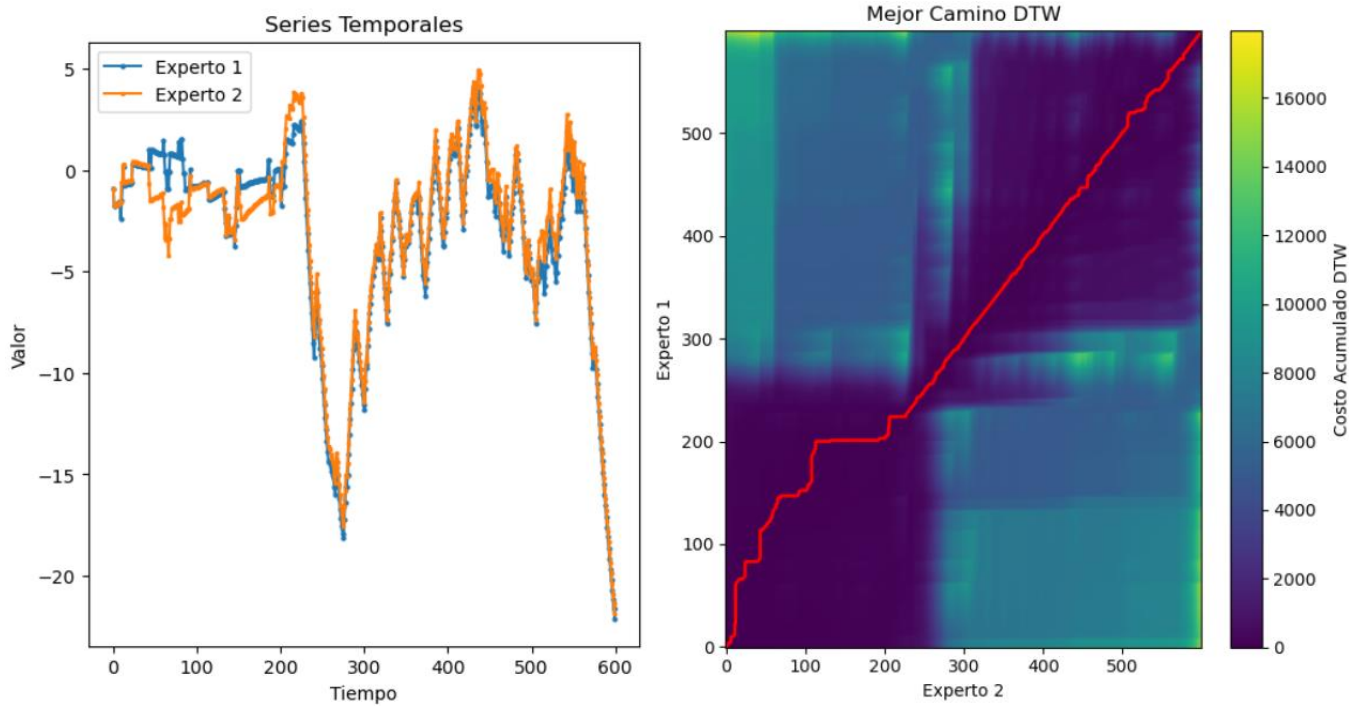


Figura 58. Series temporales sobre el eje X tiro penal a izquierda. Fuente: Elaboración propia

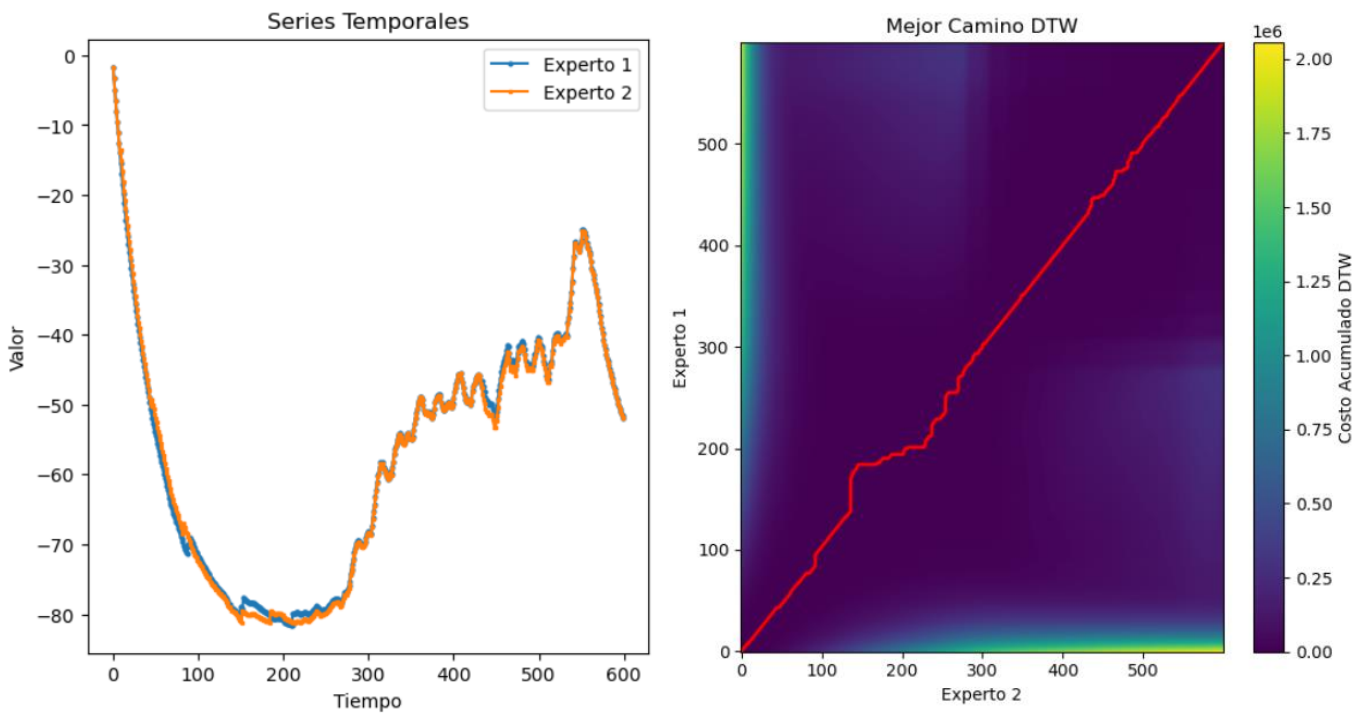


Figura 59. Series temporales sobre el eje Y tiro penal a la izquierda. Fuente: Elaboración propia

Al aplicar el *DTW* a las anteriores series temporales se obtienen las siguientes distancias:

Para el comportamiento en el eje X: 74.43133433681257

Para el comportamiento en el eje Y: 46241359968383

Aplicando la ecuación de magnitud se obtiene un valor de 345.29, el cual se aproxima a 345, con este valor se fija los siguientes rangos:

valores = [292,584,876,1168]

Así, “Excelente” se obtiene si el valor final es menor o igual a 292, “Sobresaliente” se obtiene si el valor final es mayor a 292 y menor a 584, “Bueno” se obtiene si el valor final es mayor a 584 y menor a 876, “Regular” se obtiene si el valor final es mayor a 876 y menor a 1168, finalmente si el valor final es mayor a 1168 es obtenido como evaluación “Malo”.

Cabe destacar que estas distancias son las más apropiadas desde un punto de vista práctico. Debido a la gran cantidad de muestras, es altamente improbable que se obtenga una distancia de 0. No obstante, se puede apreciar una gran similitud en las gráficas presentadas en las figuras 54 a 59.

Una vez que se han establecido los rangos para cada calificación y se han generado las series temporales de los dos deportistas expertos, el siguiente paso consiste en permitir que otros tres deportistas utilicen el sistema desarrollado. Estos deportistas llevarán a cabo el remate de tiro penal en una de las tres direcciones especificadas. Este proceso servirá para validar integralmente el sistema, tanto en términos de hardware como de software.

Deportista 1: Remate de tiro penal a la derecha.

Una vez el deportista 1 ha seleccionado patear hacia la derecha, el dispositivo *wearable* captura todos los datos y son los que el servidor va a hacer el análisis. así, de los 600 registros generados por el deportista se obtienen las secuencias temporales en el eje X y eje Y, las cuales se van a comparar respecto a las secuencias temporales del deportista experto 1. A continuación se observa las gráficas de la comparación de las secuencias temporales, junto a la matriz de costos con el mejor camino:

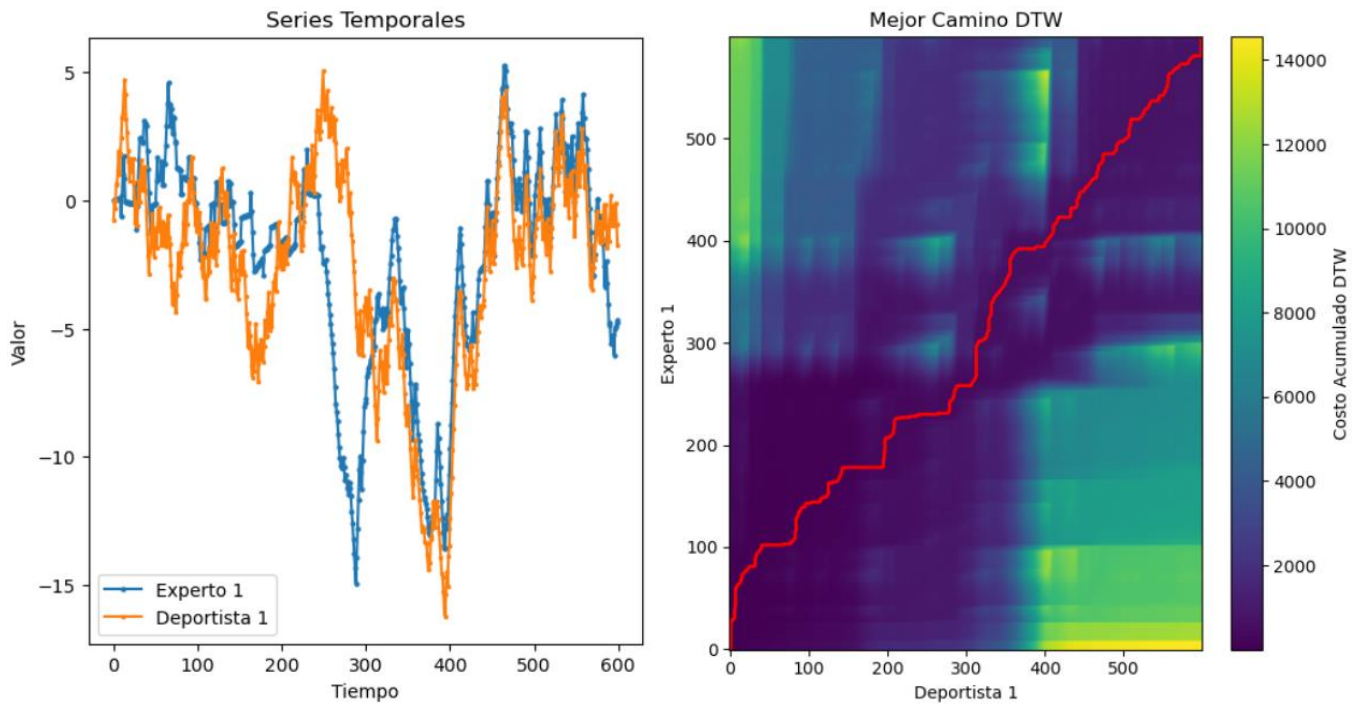


Figura 60. Series temporales sobre el eje X tiro penal a la derecha deportista 1. Fuente: Elaboración propia

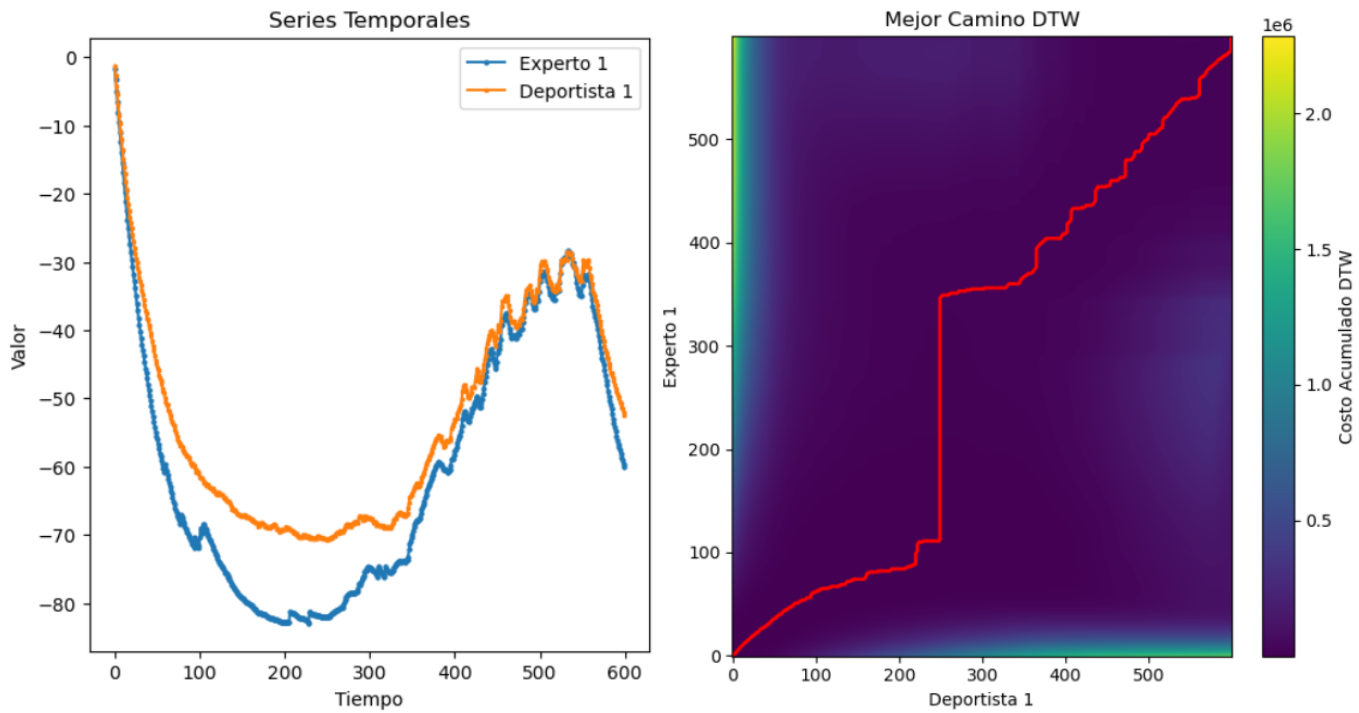


Figura 61. Series temporales sobre el eje Y tiro penal a la derecha deportista 1. Fuente: Elaboración propia

Al generar las comparaciones anteriormente mencionadas se generan dos valores, los cuales son las distancias:

$$x = 625.4726762011276.$$

y = 2227.2163563193.

Así, se obtiene un valor final igual a 2313.38 y con base a los rangos determinados previamente, esta evaluación tendría como resultado “Malo”.

Deportista 2: Remate de tiro penal al centro.

El procedimiento en este escenario es similar al caso anterior, solo varía la dirección del remate al centro, por lo cual el sistema cambia los datos de referencia. Una vez el deportista 2 realiza el movimiento, de los 600 registros generados por el deportista se obtienen las secuencias temporales en el eje X y eje Y, las cuales se van a comparar respecto a las secuencias temporales del deportista experto 1. A continuación se observa las gráficas de la comparación de las secuencias temporales, junto a la matriz de costos con el mejor camino:

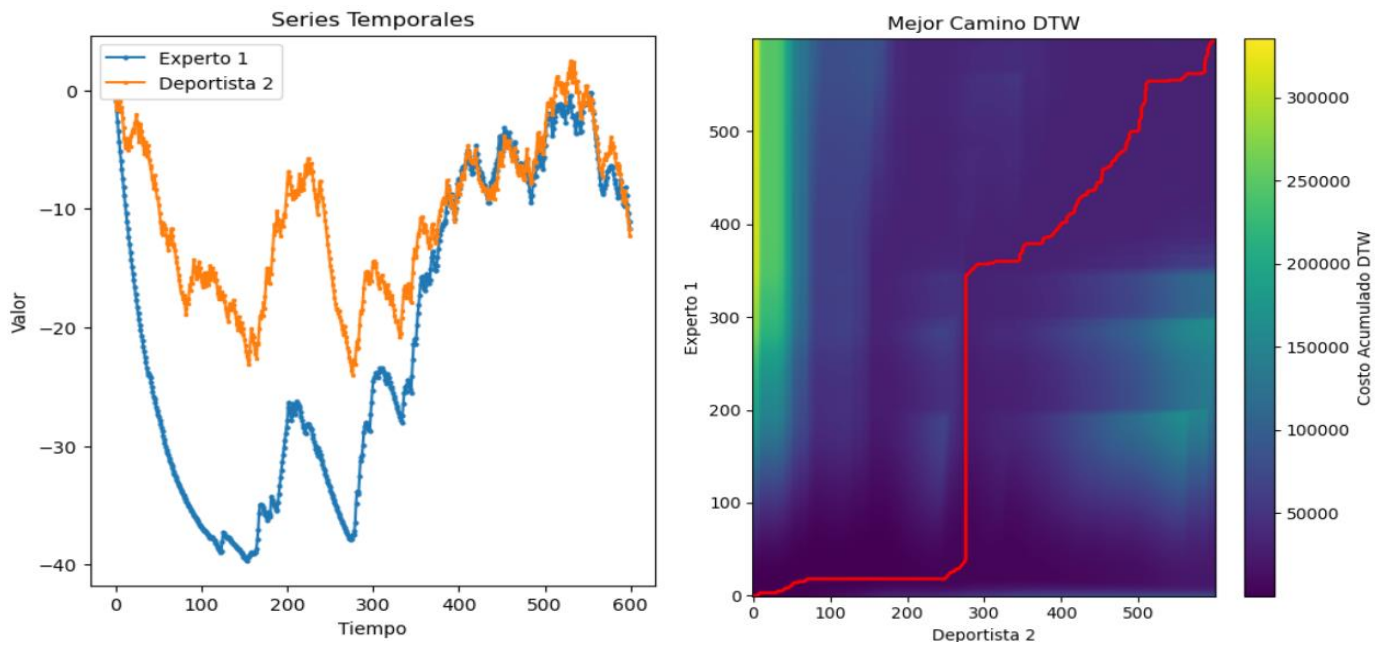


Figura 62. Series temporales sobre el eje X tiro penal al centro deportista 2. Fuente: Elaboración propia

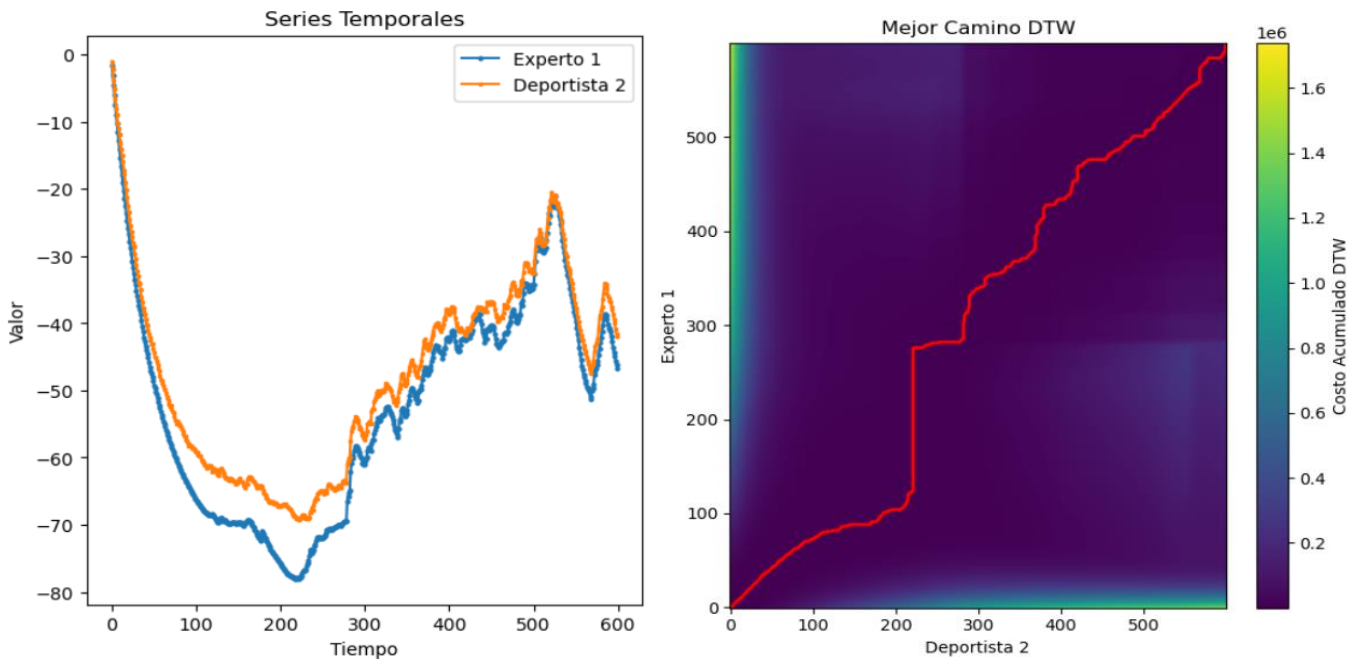


Figura 63. Series temporales sobre el eje Y tiro penal al centro deportista 2. Fuente: Elaboración propia

Al generar las comparaciones anteriormente mencionadas se generan dos valores, los cuales son las distancias:

$$x = 3603.079801998605$$

$$y = 835.6894955224415$$

Así, se obtiene un valor final igual a 3698.72 y con base a los rangos determinados previamente para el remate de tiro penal en la dirección de Centro, esta evaluación tiene como resultado “Malo”.

Deportista 3: Remate de tiro penal a la izquierda.

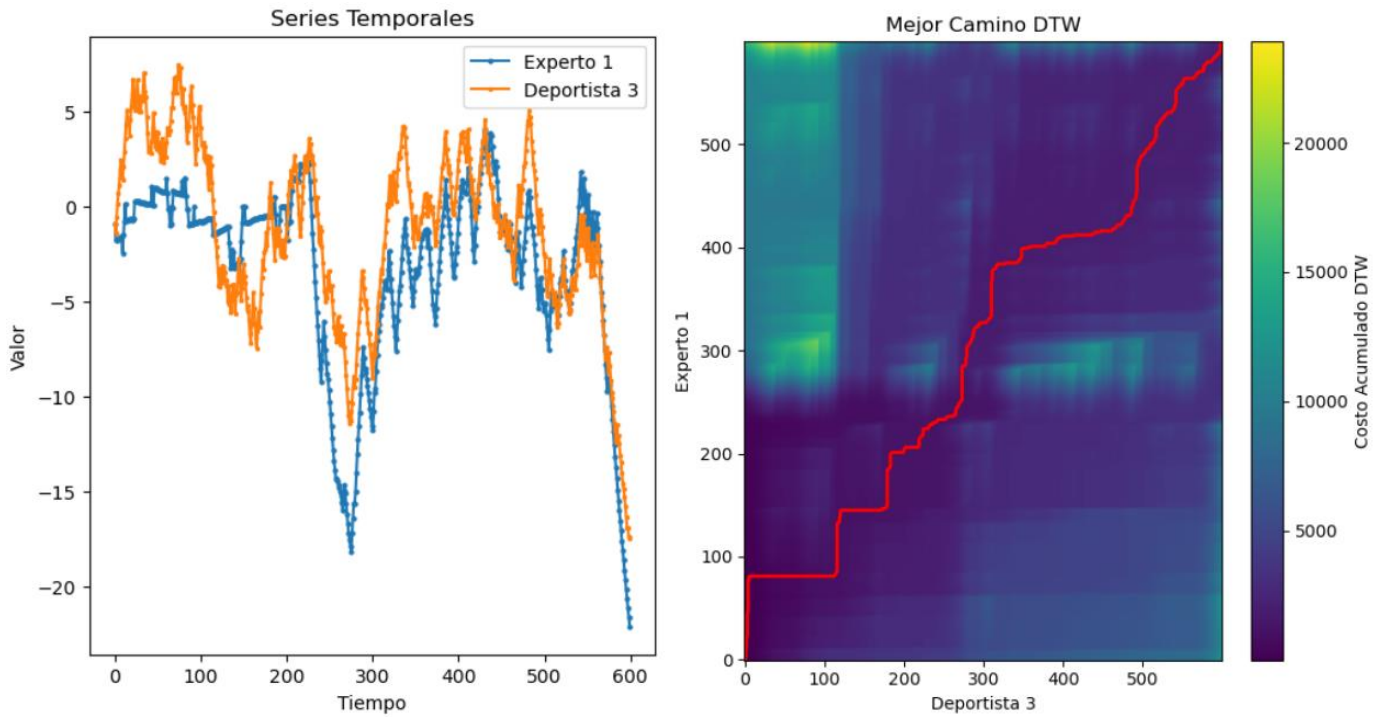


Figura 64. Series temporales sobre el eje X tiro penal a la izquierda deportista 3. Fuente: Elaboración propia

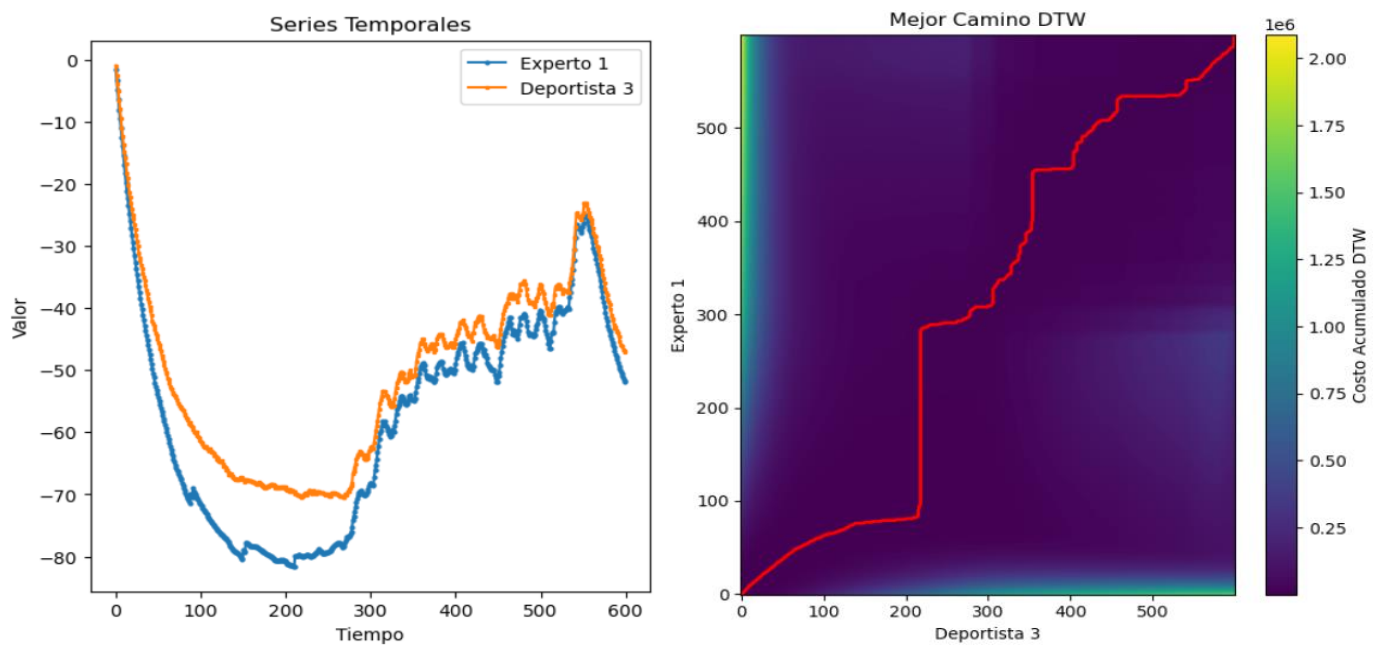


Figura 65. Series temporales sobre el eje Y tiro penal a la izquierda deportista 3. Fuente: Elaboración propia

Al generar las comparaciones anteriormente mencionadas se generan dos valores, los cuales son las distancias:

$$x = 933.8776109876159$$

$$y = 824.3538743525055$$

Así, se obtiene un valor final igual a 1245.67 y con base a los rangos determinados previamente para el remate de tiro penal en la dirección de la izquierda, esta evaluación tiene como resultado “Malo”.

Los tres escenarios descritos anteriormente son un caso de los tres que realizaron los deportistas: deportista 1, deportista 2 y deportista 3; a continuación de muestra el resultado de la ejecución de los otros dos remates de cada deportista según corresponda:

Tabla 11. Resultados de la ejecución de los dos remates restantes

Deportista	Dirección	Distancias	Valor Final	Evaluación
Deportista 1	Centro	$x = 2797.252431945452$ $y = 1205.701723443968$	3046.04	Malo
Deportista 1	Izquierda	$x = 1238.1481733505052$ $y = 2140.28858956786$	2472.62	Malo
Deportista 2	Derecha	$x = 793.9946345201173$ $y = 614.2768568789406$	1003.87	Bueno
Deportista 2	Izquierda	$x = 519.8625854473206$ $y = 612.2814119570116$	803.21	Bueno
Deportista 3	Derecha	$x = 512.5016375651251$ $y = 1004.751149600845$	1127.91	Regular
Deportista 3	Centro	$x = 1316.9099591467284$ $y = 1117.5154237040933$	1727.16	Malo

CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS.

7.1 CONCLUSIONES

A partir del trabajo realizado en esta investigación se puede concluir lo siguiente:

Conclusiones sobre el objetivo 1:

- La implementación exitosa del sistema requiere una integración efectiva de tecnologías de Internet de las Cosas (IoT) y Redes de Área Corporal Inalámbrica (WBAN). Esto implicó diseñar una arquitectura robusta la cual permitió la comunicación eficiente entre los dispositivos wearables y la infraestructura de IoT.
- La selección de sensores adecuados para capturar datos de movimientos del cuerpo humano es crucial. La precisión de estos sensores impacta directamente en la calidad de los datos recopilados, siendo fundamental para el análisis preciso de la coordinación óculo-manual u óculo-pédica.
- La usabilidad del sistema es un factor crítico. Una interfaz de usuario intuitiva facilitará la interacción con el sistema, permitiendo a los usuarios acceder y comprender fácilmente los datos recopilados sobre la coordinación óculo-manual u óculo-pédica.
- El diseño del prototipo debe ser escalable y modular para permitir futuras expansiones y actualizaciones. Esto asegurará que el sistema pueda adaptarse a cambios tecnológicos y requisitos adicionales en el futuro.

Conclusiones sobre el objetivo 2:

- La implementación exitosa de un algoritmo para evaluar gestos técnicos motrices implica un riguroso procesamiento de datos. Se requiere la aplicación de técnicas avanzadas de preprocesamiento y limpieza para garantizar la calidad de los datos antes de la evaluación.
- Es esencial identificar y seleccionar las características más relevantes de los datos recolectados para que el algoritmo pueda realizar evaluaciones precisas de los gestos técnicos. Esto puede incluir parámetros biomecánicos específicos, velocidades angulares, aceleraciones, entre otros.
- La elección de algoritmos de evaluación específicos para gestos técnicos motrices es crucial. Esto podría implicar la implementación de técnicas de aprendizaje automático, procesamiento de señales o análisis estadístico, dependiendo de la naturaleza de los gestos y los objetivos de la evaluación.
- Se debe realizar una exhaustiva validación del algoritmo implementado para garantizar su eficacia y precisión. Esto podría incluir pruebas utilizando conjuntos de datos de entrenamiento y prueba, así como la comparación de resultados con evaluaciones realizadas por expertos en el campo.

Conclusiones sobre el objetivo 3:

- La elección de un gesto técnico específico, en este caso, el tiro libre penalti en el fútbol, demuestra la aplicabilidad y relevancia del sistema en un contexto real y específico. Este enfoque permite una validación más precisa y centrada en un escenario práctico.
- La validación del prototipo implica la recopilación de datos específicos relacionados con el gesto técnico de tiro libre penalti. Esto puede incluir datos biomecánicos, como ángulos y velocidades, así como información contextual relevante para la evaluación del rendimiento.
- Se deben establecer métricas claras y objetivas para evaluar el desempeño del sistema durante el caso de estudio. Estas métricas podrían abordar la precisión en la identificación y evaluación del gesto, la consistencia en diferentes ejecuciones y la capacidad para proporcionar retroalimentación útil.
- La validación del sistema debe incluir la comparación de sus resultados con evaluaciones realizadas por expertos en el gesto técnico específico. Esta comparación sirve como una medida de la eficacia del sistema y su capacidad para coincidir con las evaluaciones expertas.
- Se debe llevar a cabo una documentación detallada de todo el proceso de validación, incluyendo los procedimientos, los datos recopilados, las métricas utilizadas y los resultados obtenidos. Esta documentación respaldará la transparencia y la replicabilidad de la validación.

En resumen, la validación del prototipo en un caso de estudio específico como el tiro libre penalti implica una evaluación detallada del rendimiento del sistema en un contexto práctico. Los resultados y aprendizajes derivados de esta validación contribuirán a fortalecer la efectividad y aplicabilidad del sistema en el ámbito de la evaluación de gestos técnicos deportivos.

7.2 TRABAJOS FUTUROS

- Utilizar el sistema diseñado para evaluar diferentes movimientos con base a la coordinación óculo-manual u óculo-pédica de diferentes deportes.
- Agregar múltiples dispositivos *wearables* al deportista, para que así, se capturen más datos y se consiga caracterizar mejor el movimiento y obtener un mejor análisis del mismo.
- Se hace necesario indagar en realizar la captura de movimientos a partir de dispositivos *wearable* comerciales como smartphone o smartband.
- Optimizar el tiempo de respuesta del sistema al generar la evaluación para reducirlo a menos de 10 segundos.

REFERENCIAS

- [1] "QUEES.INFO," [ONLINE]. AVAILABLE: [HTTPS://WWW.QUEES.INFO/QUE-ES-WEARABLE.HTML](https://www.quees.info/que-es-wearable.html).
- [2] Y. MUNSHI AND M. A. ALAM, "DYNAMIC TIME WARPING (DTW) ALGORITHM IN SPEECH: A REVIEW," *INT. J. RES. ELECTRON. COMPUT.*, VOL. 6, NO. 1, P. 6, 18.
- [3] W. A. ESTUPIÑÁN CORREDOR AND C. A. AGUDELO VELÁSQUEZ, "EFECTOS DE UN PLAN DE ENTRENAMIENTO POR MODELAMIENTO, DIRIGIDO DE FORMA VIRTUAL, PARA MEJORAR LA POTENCIA DESALTO EN JÓVENES JUGADORAS DE FUTBOL DE SALÓN," 2021. [ONLINE]. AVAILABLE: [HTTPS://REVISTAS.UDEA.EDU.CO/INDEX.PHP/VIREF/ARTICLE/VIEW/347348/20806051](https://revistas.udea.edu.co/index.php/viref/article/view/347348/20806051).
- [4] J. ZHOU, "VIRTUAL REALITY SPORTS AUXILIARY TRAINING SYSTEM BASED ON EMBEDDED SYSTEM AND COMPUTER TECHNOLOGY," *MICROPROCESSORS AND MICROSYSTEMS*, VOL. 82, 2021.
- [5] D. V. P. LAURA FAVA, "XX WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN, TECNOLOGÍA APLICADA AL DEPORTE DE ALTO RENDIMIENTO," 2018. [ONLINE]. AVAILABLE: [HTTP://WICC2018.UNNE.EDU.AR/WICC2018LIBRODEACTAS.PDF](http://wicc2018.unne.edu.ar/wicc2018librodeactas.pdf).
- [6] A. BALDI, "ISLABIT.¿QUÉ ES WBAN, O WIRELESS BODY AREA NETWORK?," 14 NOVIEMBRE 2020. [ONLINE]. AVAILABLE: [HTTPS://WWW.ISLABIT.COM/109779/QUE-ES-WBAN-O-WIRELESS-BODY-AREA-NETWORK.HTML](https://www.islabit.com/109779/que-es-wban-o-wireless-body-area-network.html).
- [7] M. J. COBO, A. G. LÓPEZ, E. HERRERA AND F. HERRERA, "SCIMAT: A NEW SCIENCE MAPPING ANALYSIS SOFTWARE TOOL," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY*, VOL. 63, NO. 8, PP. 1609-1630, 2012.
- [8] AUTORES RESERVADOS, "1LIBRARY," 2023. [ONLINE]. AVAILABLE: [HTTPS://1LIBRARY.CO/ARTICLE/ARDUINO-IDE-ARQUITECTURA-HARDWARE-ESTUDIO-ESTADO-DEL-ARTE.ZGGRXJVZ](https://1library.co/article/arduino-ide-arquitectura-hardware-estudio-estado-del-arte.zggrxjvz).
- [9] AUTORES RESERVADOS, "WORDPRESS," 11 12 2016. [ONLINE]. AVAILABLE: [HTTPS://APRENDIENDOARDUINO.WORDPRESS.COM/2016/12/11/IDE-ARDUINO/](https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/).
- [10] Y. FERNANDEZ, "XATAKA," 23 SEPTIEMBRE 2022. [ONLINE]. AVAILABLE: [HTTPS://WWW.XATAKA.COM/BASICS/QUE-ARDUINO-COMO-FUNCIONA-QUE-PUEDES-HACER-UNO](https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno).
- [11] AUTORES RESERVADOS, "ARDUINO," 2023. [ONLINE]. AVAILABLE: [HTTPS://DOCS.ARDUINO.CC/SOFTWARE/IDE-V1/TUTORIALS/ARDUINO-IDE-V1-BASICS](https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics).

- [12] MICROSOFT, "VISUALSTUDIO," 2023. [ONLINE]. AVAILABLE: [HTTPS://CODE.VISUALSTUDIO.COM/DOCS](https://code.visualstudio.com/docs).
- [13] F. FLORES, "OPENWEBINARS.NET," 22 JULIO 2022. [ONLINE]. AVAILABLE: [HTTPS://OPENWEBINARS.NET/BLOG/QUE-ES-VISUAL-STUDIO-CODE-Y-QUE-VENTAJAS-OFRECE/#:~:TEXT=VISUAL%20STUDIO%20CODE%20ES%20UN,Y%20CONECTAR%20CON%20OTROS%20SERVICIOS](https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/#:~:TEXT=VISUAL%20STUDIO%20CODE%20ES%20UN,Y%20CONECTAR%20CON%20OTROS%20SERVICIOS).
- [14] IONOS CLOUD S.L.U., "IONOS," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.IONOS.ES/DIGITALGUIDE/PAGINAS-WEB/DESARROLLO-WEB/JUPYTER-NOTEBOOK/](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/).
- [15] A. BRUGUÉS, "PROGRAMAENPYTHON," 2021. [ONLINE]. AVAILABLE: [HTTPS://WWW.PROGRAMAENPYTHON.COM/MISCELANEA/TUTORIAL-DE-JUPYTER-NOTEBOOK/](https://www.programaenpython.com/miscelanea/tutorial-de-jupyter-notebook/).
- [16] AUTORES RESERVADOS, "DEVELOPER.ANDROID," 2023. [ONLINE]. AVAILABLE: [HTTPS://DEVELOPER.ANDROID.COM/STUDIO/INTRO?HL=ES-419](https://developer.android.com/studio/intro?hl=es-419).
- [17] JETBRAINS, "JETBRAINS," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.JETBRAINS.COM/IDEA/](https://www.jetbrains.com/idea/).
- [18] MOZILLA CORPORATION'S, "DEVELOPER.MOZILLA," 2023. [ONLINE]. AVAILABLE: [HTTPS://DEVELOPER.MOZILLA.ORG/ES/DOCS/WEB/HTTP/OVERVIEW](https://developer.mozilla.org/es/docs/web/http/overview).
- [19] AUTORES RESERVADOS, "NEO.LCC.UMA.ES," 2020. [ONLINE]. AVAILABLE: [HTTPS://NEO.LCC.UMA.ES/EVIRTUAL/CDD/TUTORIAL/APLICACION/HTTP.HTML](https://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html).
- [20] ARIMETRICS, "ARIMETRICS," 2022. [ONLINE]. AVAILABLE: [HTTPS://WWW.ARIMETRICS.COM/GLOSARIO-DIGITAL/WEARABLE](https://www.arimetrics.com/glosario-digital/wearable).
- [21] T. MOES, "SOFTWARELAB," 2023. [ONLINE]. AVAILABLE: [HTTPS://SOFTWARELAB.ORG/ES/BLOG/QUE-ES-WIFI/#:~:TEXT=EL%20WIFI%20ES%20UN%20MECANISMO,SIN%20LA%20NECESIDAD%20DE%20CABLES](https://softwarelab.org/es/blog/que-es-wifi/#:~:TEXT=EL%20WIFI%20ES%20UN%20MECANISMO,SIN%20LA%20NECESIDAD%20DE%20CABLES).
- [22] L. DEL VALLE, "PROGRAMARFACIL," 2023. [ONLINE]. AVAILABLE: [HTTPS://PROGRAMARFACIL.COM/ESP8266/ESP32/](https://programarfacil.com/esp8266/esp32/).
- [23] A. ROBLDANO, "OPENWEBINARS," 22 JULIO 2019. [ONLINE]. AVAILABLE: [HTTPS://OPENWEBINARS.NET/BLOG/QUE-ES-CPP/](https://openwebinars.net/blog/que-es-cpp/).
- [24] F. CAMACHO, "BLOG.GFT," 16 MARZO 2023. [ONLINE]. AVAILABLE: [HTTPS://BLOG.GFT.COM/ES/2023/03/16/EL-PAPEL-DEL-LENGUAJE-C-EN-SISTEMAS-EMBEBIDOS-PARA-LA-INDUSTRIA/](https://blog.gft.com/es/2023/03/16/el-papel-del-lenguaje-c-en-sistemas-embbebidos-para-la-industria/).

- [25] AUTORES RESERVADOS, "AWS," AMAZON WEB SERVICES, 2023. [ONLINE]. AVAILABLE: [HTTPS://AWS.AMAZON.COM/ES/WHAT-IS/PYTHON/](https://aws.amazon.com/es/what-is/python/).
- [26] AUTORES RESERVADOS, "PYTHON," PYTHON SOFTWARE FOUNDATION, 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.PYTHON.ORG/](https://www.python.org/).
- [27] G. M. BELTRAN, "JAVA COMO LENGUAJE UNIVERSAL DE PROGRAMACIÓN," XIKUA, 8 JULIO 2016. [ONLINE]. AVAILABLE: [HTTPS://WWW.JAVA.COM/ES/DOWNLOAD/HELP/WHATIS_JAVA.HTML](https://www.java.com/es/download/help/whatis_java.html).
- [28] THE PHP GROUP, "PHP," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.PHP.NET/MANUAL/ES/INTRO-WHATIS.PHP](https://www.php.net/manual/es/intro-what-is.php).
- [29] AULADIG, "ROCO," 23 MARZO 2022. [ONLINE]. AVAILABLE: [HTTPS://AGENCIAROCO.COM/QUE-ES-UNA-LIBRERIA-DE-PROGRAMACION/](https://agenciaroco.com/que-es-una-libreria-de-programacion/).
- [30] A. HINCAPIE, "MEDIUM," 5 OCTUBRE 2021. [ONLINE]. AVAILABLE: [HTTPS://LAHINCAPIE.MEDIUM.COM/LIBRER%C3%ADAS-EST%C3%A1TICAS-Y-DIN%C3%A1MICAS-EN-C-578749F6E079](https://lahincapie.medium.com/librer%C3%ADAS-est%C3%A1ticas-y-din%C3%A1micas-en-c-578749f6e079).
- [31] INGMECAFENIX, "INGMECAFENIX," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.INGMECAFENIX.COM/AUTOMATIZACION/SENSORES/SENSOR-INERCIAL/](https://www.ingmecafenix.com/automatizacion/sensores/sensor-inercial/).
- [32] AUTORES RESERVADOS, "330OHMS," 27 SEPTIEMBRE 2017. [ONLINE]. AVAILABLE: [HTTPS://BLOG.330OHMS.COM/2017/09/27/SENSORES-INERCIALES-ACELEROMETROS-GIROSCOPIOS-E-IMUS/](https://blog.330ohms.com/2017/09/27/sensores-inerciales-acelerometros-giroscopios-e-imus/).
- [33] NAYLAMP MECHATRONICS SAC, "NAYLAMPMECHATRONICS," 2016. [ONLINE]. AVAILABLE: [HTTPS://NAYLAMPMECHATRONICS.COM/BLOG/45_TUTORIAL-MPU6050-ACELEROMETRO-Y-GIROSCOPIO.HTML#:~:TEXT=EL%20MPU6050%20ES%20UNA%20UNIDAD,%2C%20GONIOMETR%C3%ADA%2C%20ESTABILIZACI%C3%B3N%2C%20ETC](https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscoPIO.html#:~:TEXT=EL%20MPU6050%20ES%20UNA%20UNIDAD,%2C%20GONIOMETR%C3%ADA%2C%20ESTABILIZACI%C3%B3N%2C%20ETC).
- [34] EDITORIAL ETECÉ, "CONCEPTO," 15 JULIO 2021. [ONLINE]. AVAILABLE: [HTTPS://CONCEPTO.DE/BATERIA/](https://concepto.de/bateria/).
- [35] COPYRIGHT EQUIPOS Y LABORATORIO DE COLOMBIA, "ARTICULOS Y LABORATORIO," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.EQUIPOSYLABORATORIO.COM/PORTAL/ARTICULO-AMPLIADO/BATERIAS-\(ELECTRICIDAD\)](https://www.equiposylaboratorio.com/portal/articulo-ampliado/baterias-(electricidad)).
- [36] RECICLATUSPILAS, "TODO SOBRE PILAS BOTON," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.RECICLATUSPILAS.COM/TODO-LO-QUE-DEBES-SABER-ACERCA-DE-LAS-PILAS-DE-BOTON/#:~:TEXT=LA%20PRINCIPAL%20CARACTER%C3%ADSTICA%20DE%20ESTE,ENTRE%20%2D20%C2%BA%20Y%20%2B60%C2%BA](https://www.reciclatuspilas.com/todo-lo-que-debes-saber-acerca-de-las-pilas-de-boton/#:~:TEXT=LA%20PRINCIPAL%20CARACTER%C3%ADSTICA%20DE%20ESTE,ENTRE%20%2D20%C2%BA%20Y%20%2B60%C2%BA).

- [37] VELAMP, "TIPO DE BATERÍAS," 2021. [ONLINE]. AVAILABLE: [HTTPS://WWW.VELAMP.COM/ES/ALCALINAS/1163-PILA-ALCALINA-LR6-AA-1-5V-MULTIPACK-DE-12.HTML](https://www.velamp.com/es/ALCALINAS/1163-PILA-ALCALINA-LR6-AA-1-5V-MULTIPACK-DE-12.HTML).
- [38] IBERDROLA, "LAS BATERÍAS DE ION DE LITIO, FUNDAMENTALES PARA EL ALMACENAMIENTO DE ENERGÍA," 2023. [ONLINE]. AVAILABLE: [HTTPS://WWW.IBERDROLA.COM/INNOVACION/BATERIAS-ION-LITIO](https://www.iberdrola.com/innovacion/baterias-ion-litio).
- [39] 1LIBRARY, "MÉTODO PARA ACELERAR EL DTW (FAST DTW)," 2020. [ONLINE]. AVAILABLE: [HTTPS://1LIBRARY.CO/ARTICLE/M%C3%A9TODO-ACELERAR-DTW-FAST-AUTOMATIZACI%C3%B3N-ADQUISICI%C3%B3N-AN%C3%A1LISIS-PAR%C3%A1-MET.QMWEWE5Z](https://1library.co/article/m%C3%A9todo-acelerar-dtw-fast-automatizaci%C3%B3n-adquisici%C3%B3n-an%C3%A1lisis-par%C3%A1-met.qmwewez).
- [40] JESUS, "CONOCE QUÉ ES XAMPP Y POR QUÉ DEBERÍAS USARLO EN TUS PROYECTOS," 25 ABRIL 2022. [ONLINE]. AVAILABLE: [HTTPS://WWW.DONGEE.COM/TUTORIALES/QUE-ES-XAMPP/](https://www.dongee.com/tutoriales/que-es-xampp/).
- [41] M. HÉCTOR R, "MONOGRAFIAS," 2020. [ONLINE]. AVAILABLE: [HTTPS://WWW.MONOGRAFIAS.COM/TRABAJOS94/PROCESAMIENTO-DIGITAL-SENALES-2/PROCESAMIENTO-DIGITAL-SENALES-2](https://www.monografias.com/trabajos94/procesamiento-digital-senales-2/procesamiento-digital-senales-2).
- [42] S. HERNÁNDEZ, "ANÁLISIS DE SERIES DE TIEMPO," 23 ENERO 2015. [ONLINE]. AVAILABLE: [HTTPS://WWW.CEPAL.ORG/SITES/DEFAULT/FILES/COURSES/FILES/01_1_CONOCIENDO_UNA_SERIE_DE_TIEMPO.PDF](https://www.cepal.org/sites/default/files/courses/files/01_1_conociendo_una_serie_de_tiempo.pdf).
- [43] M. BURBANO FERNANDEZ, J. SANDOVAL SERNA, Y. RIASCOS, M. MUÑOZ ORGANERO, M. THILAGARAJ, V. VENKATARAMAN, N. ARUNKUMAR AND G. RAMIREZ GONZALEZ, "ALGORITHM FOR THE COMPARISON OF HUMAN PERIODIC MOVEMENTS USING WEARABLE DEVICES," *JOURNAL OF HEALTHCARE ENGINEERING*, VOL. VOL. 2021, P. 18, 2021.
- [44] S. HIROAKI AND C. SEIBI, "DYNAMIC PROGRAMMING ALGORITHM OPTIMIZATION FOR SPOKEN WORD RECOGNITION," *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*, VOL. 26, NO. 1, P. 7, 1978.
- [45] Y. XU AND W. HU, "LOAD EVALUATION OF CAMPUS FOOTBALL MATCH BASED ON MICROPROCESSOR AND IOT WEARABLE EQUIPMENT," *MICROPROCESSORS AND MICROSYSTEMS*, VOL. 81, NO. 103778, 28 DICIEMBRE 2020.
- [46] R. LUO AND Y. J. LI, "AN INDOOR POSITIONING METHOD INTEGRATING WIFI AND WEARABLE INERTIAL NAVIGATION MODULE," MARZO 2022. [ONLINE]. AVAILABLE: [HTTPS://WWW-SCOPUS-COM.ACCESO.UNICAUCA.EDU.CO/RECORD/DISPLAY.URI?EID=2-S2.0-85130842037&ORIGIN=RESULTS LIST&SORT=PLF-F&SRC=S&ST1=WEARABLE+AND+WIFI&SID=04E4DD3071CE201AD2BECCF2FE46A7EB&SOT=B&SDT=B&SL=32&S=TITLE-ABS-KEY%28WEARABLE+AND+WIFI%29&RELPOS=5&CITECNT](https://www.scopus.com/acceso.unicauca.edu.co/record/display.uri?eid=2-s2.0-85130842037&origin=resultslist&sort=plf-f&src=s&st1=wearable+and+wifi&sid=04e4dd3071ce201ad2beccf2fe46a7eb&sot=b&sdt=b&sl=32&s=title-abs-key%28wearable+and+wifi%29&relpos=5&citecnt).

- [47] M. A. GAWAD, "REMOTE HEALTHCARE MONITORING USING WEARABLE IOT DEVICES AND CLOUD SERVICES," 30 MARZO 2022. [ONLINE]. AVAILABLE: [HTTPS://WWW-SCOPUS-COM.ACCESO.UNICAUCA.EDU.CO/RECORD/DISPLAY.URI?EID=2-S2.0-85130490735&ORIGIN=RESULTSLIST&SORT=PLF-F&SRC=S&ST1=WEARABLE+AND+WIFI&SID=04E4DD3071CE201AD2BECCF2FE46A7EB&SOT=B&SDT=B&SL=32&S=TITLE-ABS-KEY%28WEARABLE+AND+WIFI%29&RELPOS=18&CITECN](https://www-scopus-com.acceso.unicauca.edu.co/record/display.uri?eid=2-s2.0-85130490735&origin=resultslist&sort=plf-f&src=s&st1=wearable+and+wifi&sid=04e4dd3071ce201ad2beccf2fe46a7eb&sot=b&sdt=b&sl=32&s=title-abs-key%28wearable+and+wifi%29&relpos=18&citecn).
- [48] H. DENG, " REAL-TIME MONITORING OF ATHLETES' TRAINING DATA BASED ON," *MICROPROCESSORS AND MICROSYSTEMS*, VOL. 81, 2021.
- [49] K. D. DAHL, K. M. DUNFORD, S. A. WILSON, T. L. TURNBULL AND S. TASHMAN, "WEARABLE SENSOR VALIDATION OF SPORTS-RELATED MOVEMENTS FOR THE LOWER EXTREMITY AND," *SCIENCEDIRECT*, PP. 144-150, 2020.
- [50] I. ARDIYANTO, S. WIBIRAMA AND F. NURWANTO, "SLIDING VARIANCE AND DATA RANGE FOR LIGHTWEIGHT SPORTS ACTIVITY RECOGNITION WITH FUSION OF MODALITIES. JOURNAL OF KING SAUD UNIVERSITY," *COMPUTER AND INFORMATION SCIENCES*, PP. 775-783, 2020.
- [51] P. KRUCHTEN, "PLANOS ARQUITECT'ONICOS: EL MODELO DE "4+1" VISTAS DE LA," *ARQUITECTURA DEL SOFTWARE*, P. 16, 2020.
- [52] J. E. B. RAMÍREZ AND Y. F. T. CASTILLO, "APROXIMACIÓN METODOLÓGICA CON PROPÓSITO ACADÉMICO PARA EL DESARROLLO DE DISPOSITIVOS WEARABLE PARA EXTREMIDADES INFERIORES DEL CUERPO HUMANO," P. 107, 2018.
- [53] A. N. MARTELLOTTI, "SISTEMA DE CAPTURA DE MOVIMIENTO CON INTEL EDISON," *LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN*, P. 61, 2018.