



Disertación

Maestría en Ingeniería Informática - Computación Móvil

**SPORTMOVE
REGISTRO Y MONITORIZACIÓN DE
MOVIMIENTOS PARA LA PRÁCTICA
DEPORTIVA**

Wilson Fernando Yandún Torres

Leiria, septiembre del 2017



Disertación

Maestría en Ingeniería Informática - Computación Móvil

**SPORTMOVE
REGISTRO Y MONITORIZACIÓN DE
MOVIMIENTOS PARA LA PRÁCTICA
DEPORTIVA**

Wilson Fernando Yandún Torres

Disertación de Maestría realizada bajo la orientación del : Doctor. Luís Marcelino, Profesor de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria-Portugal, Doctor Pedro Assunção, Profesor de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria-Portugal y Doctor Diego Marcillo, Docente de la Universidad de Fuerzas Armadas ESPE-Ecuador.

Leiria, septiembre del 2017

Agradecimientos

Quiero agradecer a Dios por darme las fuerzas para continuar con este trabajo, a mis padres y hermanos por ser los pilares fundamentales en mi vida , a mis profesores que aportaron con su conocimiento para el desarrollo de esta investigación y a la SENESCYT que me brindó la oportunidad de obtener esta beca.

Resumen

En el mundo existen más de 4.6 millones de aplicaciones móviles disponibles para sistemas operativos iOS y Android, orientadas a facilitar las actividades cotidianas de los seres humanos. El problema radica, en la escasa o nula existencia de aplicaciones móviles que integren un algoritmo que mida el rendimiento de los deportistas en base a la precisión de sus movimientos.

La meta de esta investigación es explicar de manera detallada la propuesta, implementación, elaboración de pruebas y análisis de resultados de una aplicación móvil, enfocada en las necesidades de los usuarios deportivos (deportistas o entrenadores) que deseen evaluar, registrar y comparar el desempeño deportivo en base a la precisión de sus movimientos, empezando por la comparación de movimientos genéricos.

Para desarrollar este trabajo se realizaron pruebas con distintos algoritmos, para hallar el que mejor se adapte a la investigación (Para evaluar los movimientos se utilizó R con la librería Similarity Measures). De manera paralela se capturaron los movimientos de prueba, en un brazalete con un conjunto de sensores (Giroscopio, acelerómetro, sentido de orientación).

Todos los datos de la investigación y el registro de los movimientos fueron almacenados en una base de datos. (MySQL), para luego ser procesados por un software que realizó la integración de los datos, el sistema y el algoritmo mediante un framework de programación en el lenguaje PHP (Laravel).

El trabajo demuestra las diferentes características de cada algoritmo y las dificultades que presentan para adaptarse a las necesidades de la investigación. Encontrar el algoritmo adecuado, permitió desarrollar una interfaz móvil, que asegura una mejor experiencia de usuario.

Palabras claves: *Aplicaciones móviles, Algoritmos de comparación de movimientos, Acelerómetro, Giroscopio, Sentido de orientación.*

Abstract

In the world, there exist more than 4.6 million mobile available applications for operating systems iOS and Android, orientated to facilitating the daily activities of the human beings. The problem lies in the very fact in the scanty or void existence of mobile applications that integrate an algorithm that measures the yield of the sportsmen based on the precision of their movements.

The goal of this research is to explain the proposal, implementation, elaboration of proofs and analysis of results of a mobile application, focused in the needs of the sports users should want to evaluate, register and compare the sports performance based on the precision of their movements, beginning for the comparison of generic movements.

To develop this work proofs were realized with different algorithms, to find the best one that adapts to the investigation (evaluate the movements, we used R with Similarity Measures library). In a parallel way, the movements of proof were caught in a bracelet with a set of sensors. (Gyroscope, accelerometer, sense of orientation).

All the information of the investigation and the record of the movements were stored in a database (MySQL), to then to be processed by a software that realized the integration of the information, the system and the algorithm by framework in language PHP (Laravel).

The work demonstrates the different characteristics of each algorithm and the difficulties that they present to adapt to the needs of the investigation.

Find the pertinent algorithm, it allowed to develop an movil interface, which ensures a better user experience.

Key words: *Mobile applications, Motion comparison algorithms, Accelerometer, Gyroscope, Sense of Orientation.*

Tabla de contenido

Agradecimientos	iv
Resumen	v
Abstract	vi
Lista de figuras	ix
Lista de tablas	xi
Lista de Acrónimos	xii
1. Introducción	1
1.1 Encuadramiento	1
1.2 Motivación y objetivos	2
1.3 Metodología	2
1.4 Estructura del documento.....	3
2. Estado del arte	4
2.1 Biomecánica del Deporte Y Ejercicio	4
2.1.2 Análisis del movimiento humano	5
2.2 Herramientas de Hardware	5
2.2.1 Dispositivo móvil	5
2.3 Herramientas de Software	9
2.3.1 Android Studio.....	9
2.3.2 Laravel	9
2.3.3 Git	10
2.3.4 MySql.....	10
2.3.5 R: The R Project for Statistical Computing.....	10
2.4 Algoritmos de comparación de movimientos	11
2.5.1 DTW (Dynamic Time Warping)	12
2.5.2 Edit Distance on Real sequence.....	13
2.5.3 LCSS (Longest Common Subsequence).....	14
2.5.4 Distancia Frechet	16
2.6 Resumen.....	17
3. Propuesta	18
3.1 Estructura	18
4. Implementación	23

4.1 Test De Algoritmos Con Movimientos	23
4.1.1. Análisis Gráfico	23
4.1.2 Implementación con R.....	26
4.2 Implementación del Webservice	29
4.2.1 Base De Datos.....	30
4.2.1 Test de Algoritmos en Función del Webservice	31
4.3 Implementación de la Aplicación Móvil	41
4.3.1 INTERFAZ DEL USUARIO	42
4.3.2 Conexión con el Brazalete	46
4.4 Resultados	47
5. Conclusiones y Recomendaciones	50
5.1 Conclusiones.....	50
5.2 Recomendaciones	51
Bibliografía	52
Anexos.....	1
Anexo A. Análisis de movimientos con algoritmos DTW y Edit Distance	1
Anexo B. Resultados obtenidos con la utilización de LCSS y Frechet en 30 y 50 Movimientos.....	2
Anexo C. Resultados obtenidos con la utilización de LCSS y Frechet en 100 y 120 movimientos.....	4
Anexo D. Resultados obtenidos con la utilización de puntos máximos y mínimos en todos los algoritmos	6

Lista de figuras

Figura 1 Brazalete Myo. Elaboración propia.	7
Figura 2 Clasificación de Similarity Measures (Magdy, et al. 2015, p.614)	11
Figura 3 Dos series similares desfasadas en el tiempo (Magdy, et al. 2015, p.615)	12
Figura 4 Comparación de precisión entre algoritmos (Ding, H, et al. 2008, p 1548).....	14
Figura 5 Umbral en el algoritmo LCSS (Vlachos, et al. 2002, p.675).....	15
Figura 6 Distancia de Frechet (Magdy, et al. 2015, p.616).....	16
Figura 7 Características de algoritmos de comparación de movimientos (Lei Chen, et al. 2005, p493)	17
Figura 8 Propuesta de Arquitectura Inicial. Elaboración propia.....	18
Figura 9 Login de la aplicación. Elaboración propia	19
Figura 10 Listado de movimientos. Elaboración propia	20
Figura 11 Selección de resultados . Elaboración propia.....	21
Figura 12 Resultado con porcentaje de exactitud obtenido en la aplicación. Elaboración propia	22
Figura 13 Ejemplo de datos generados en CSV. Elaboración propia	24
Figura 14 Secuencias de movimientos utilizados para la comparación de movimientos. Elaboración propia	24
Figura 15 Gráfico comparativo de movimientos similares realizados por una persona. Elaboración propia	25
Figura 16 Comparación gráfica de dos movimientos similares en distintas personas en el sensor de orientación	25
Figura 17 Comparación gráfica de dos movimientos similares en distintas personas en el sensor de aceleración	26
Figura 18 Resultados de las pruebas realizadas con todos los resultados de un movimiento. Elaboración propia.....	29
Figura 19 Diagrama Lógico de la Base de Datos. Elaboración propia.....	30
Figura 20 Resultado de algoritmos DTW y Edit Distance. Elaboración propia	31

Figura 21 Arquitectura de la solución . Elaboración propia	41
Figura 22 Login con credenciales erróneas	42
Figura 23 Página principal. Elaboración propia	43
Figura 24 Interfaz de conexión del brazalete Myo	44
Figura 25 Ingreso de datos generales del movimiento. Elaboración propia	44
Figura 26 Pantalla de inicio de grabación de movimiento	45
Figura 27 Tiempo de gracia para el inicio de la grabación	46
Figura 28 Comparación gráfica de movimientos con cierto grado de similaridad	49

Lista de tablas

Tabla 1 Resultado de movimientos similares	32
Tabla 2 Resultados de un mismo movimiento realizado por personas distintas	32
Tabla 3 Resultados con movimientos distintos	32
Tabla 4 Resultados del algoritmo DTW en el sensor de aceleración.....	34
Tabla 5 Tiempos de respuesta de LCSS	35
Tabla 6 Resultados de algoritmo de Frechet para 30 y 50 entradas de datos	37
Tabla 7 Diferencia en porcentaje de resultados del algoritmo DTW	38
Tabla 8 Cálculo de margen de tolerancia de los resultados del algoritmo DTW.....	40
Tabla 9 Resultados de movimientos realizados	48

Lista de Acrónimos

CSV	Comma-Separated Values
DTW	Dynamic Time Warping
GPS	Global Positioning System
LCSS	Longest Common Subsequence
ORM	Object Relational Mapping
PC	Personal Computer
PDA	Personal Digital Assistant
PHP	Hypertext Preprocessor
SDK	Software Development Kit
SQL	Structured Query Language

1. Introducción

Este capítulo abordará el encuadramiento , explicando la situación actual de las herramientas tecnológicas en el área deportiva la motivación por la cual se realizó el mismo, seguido de la estructura del documento y la metodología con la que se realizó el mismo.

1.1 Encuadramiento

Este trabajo de investigación está motivado por la necesidad de los usuarios deportivos, de contar con una herramienta que esté al alcance de sus posibilidades; y, que sea capaz de facilitar sus entrenamientos y medir su avance o desempeño en el deporte.

Cabe mencionar que, aunque existen aplicaciones y dispositivos electrónicos wereables que miden el tiempo del entrenamiento, resistencia física, entre otros factores de rendimiento. Aún no se desarrollan las suficientes aplicaciones móviles, que brinden un panorama completo sobre el avance técnico del deportista basado en la precisión de sus movimientos.

Para solucionar esta necesidad, es necesario abordar datos que sustentan la importancia de las aplicaciones móviles en la vida cotidiana de los seres humanos y replantear la meta de los investigadores y programadores que debe radicar en desarrollar soluciones accesibles y competitivas que sean capaces de mejorar la calidad de vida de los usuarios:

Se debe resaltar que ahora el mercado de desarrollo tecnológico se enfrenta a consumidores más hiperconectados, en este 2017, se estimó que:

- Más del 50% de la población a nivel mundial cuenta con acceso a Internet. (We Are Social, 2017)
- Más del 4.917 billones de personas (66% de la población mundial) tienen acceso a teléfonos móviles. (We are Social, 2017)

- Las personas de entre 18 y 40 años con teléfonos inteligentes interactúan más de 185 minutos diarios con sus aplicaciones móviles. (Stratista, 2017)

Esta es la oportunidad que investigadores y programadores deben aprovechar para proponer respuestas tecnológicas innovadoras, que respondan a las necesidades de los usuarios.

Esto lo tienen claro los líderes tecnológicos, emprendedores e innovadores que han incrementado en 1 millón el número de aplicativos móviles, desde 2013 hasta marzo de 2017. Hasta el momento se calcula en existencia más de 2.8 millones de aplicaciones en Android y 2.2 millones en App Store de Apple (Statista, 2017) orientadas a facilitar las actividades cotidianas de los seres humanos, en: educación, salud, entretenimiento, deporte, etc.

1.2 Motivación y objetivos

En base a las nuevas necesidades de la sociedad, el objetivo principal de esta investigación es desarrollar una aplicación móvil, que permita la comparación de movimientos genéricos para la evaluar la similitud que se pueda encontrar entre ellos.

En dirección hacia otro público, la aplicación permitirá a los entrenadores registrar, evaluar y comparar los resultados de sus deportistas y plantear mejores estrategias de preparación física.

De manera específica, esta investigación tiene por objetivo democratizar el acceso a los usuarios a una aplicación deportiva con diseño interactivo, que posee otras funcionalidades fuera de las convencionales.

Otro de los motivos fue hallar un algoritmo que permita comparar los movimientos realizados durante los entrenamientos. Además la investigación tendrá que determinar el margen de tolerancia que acepta cada algoritmo utilizado en el proceso.

1.3 Metodología

Dentro del desarrollo de la investigación se tiene tres puntos principales:

1. La obtención de movimientos a través de sensores.

2. La comparación de los movimientos a través de una herramienta estadística
3. La interpretación de los resultados obtenidos de la comparación

Para llevar a cabo el primer punto se realizó la captura de movimientos que realizaba una persona mediante la utilización de un brazalete con un conjunto de sensores su la interfaz de programación , toda la información recolectada fue enviada a un dispositivo móvil para su posterior utilización.

Para el segundo punto se realizó pruebas en los movimientos con distintos algoritmos, obteniendo resultados que deben ser interpretados.

En el tercer punto se realizó una comparación de los resultados obtenidos con diferentes movimientos realizados por diferentes personas para obtener una tendencia.

Todos los datos de la investigación y el registro de los movimientos fueron almacenados en en una base de datos. para luego ser procesados por un software que realizó la integración de los datos, el sistema y el algoritmo.

1.4 Estructura del documento

De manera general los capítulos de esta investigación explican:

En el Capítulo 2 se detalla de manera profunda la terminología especializada que se utilizó durante la investigación, con el fin de que el lector pueda familiarizarse con la jerga propia de la programación y alcance mejores niveles de comprensión acerca del trabajo expuesto, el Capítulo 3 detalla la arquitectura propuesta para el proyecto de una manera generalizada, el Capítulo 4 nos muestra la implementación que se realizó de la aplicación, los pasos que se siguieron para definir el mejor algoritmo para realizar la comparación de los movimientos y los resultados de la aplicación de la implementación de los algoritmos, mediante algunas pruebas realizadas. Por último en el Capítulo 5 tenemos las conclusiones y recomendaciones sobre el trabajo realizado.

2. Estado del arte

En este capítulo se dará a conocer una definición de los conceptos utilizados para realizar esta investigación, como primer punto principal se toma en cuenta la Biomecánica del Deporte y del Ejercicio, para poder entender un poco como funciona los movimientos realizados por los deportistas y como podemos representar los mismos a través de un método cuantitativo, luego se habla de los dispositivos que se usaron en la aplicación con sus respectivos sensores y por último se da una visión general de los algoritmos utilizados.

2.1 Biomecánica del Deporte Y Ejercicio

La biomecánica es un “conjunto de conocimientos interdisciplinarios que se apoyan en el uso de otras ciencias biomédicas, los conocimientos de la mecánica y distintas tecnologías para estudiar el comportamiento de los sistemas biológicos y en particular, del cuerpo humano; y, segundo: resolver los problemas que le provocan las distintas condiciones a las que puede verse sometido” (Vera, 1994)

Ésta, es de vital importancia para la actividad física porque, detrás de una actividad deportiva hay aspectos mecánicos que explican la importancia que tiene realizar un movimiento correctamente.

Si conocemos estos aspectos de la biomecánica y podemos replicarlos de una manera correcta, podríamos obtener estos beneficios:

- Optimización en las actividades deportivas
- Mejora del rendimiento en el deportista
- Prevención de originadas por la inadecuada postura del cuerpo.

2.1.2 Análisis del movimiento humano

Es importante mencionar que en la Biomecánica deportiva y de ejercicio se describen los movimientos a través de procedimientos cuantitativos y cualitativos.

2.1.2.1 Procedimiento cuantitativo

Este procedimiento se realiza en términos numéricos para describir los movimientos y así desvanecer la descripción subjetiva. Para poder realizar una correcta apreciación cuantitativa del movimiento debemos utilizar un instrumento de medida que debe poseer una constancia y precisión suficientes, de forma que cada magnitud de la característica, cualidad o fenómeno que se mide reciba la misma o semejante valoración en todos los casos en los que sea utilizado el instrumento. De esto depende la confianza que podemos conceder a los datos, es decir, la fiabilidad de la medición.

2.1.2.1 Procedimiento cualitativo

Describe el movimiento en términos no numéricos. En el procedimiento cualitativo se realiza desde análisis numéricos o cuantitativos, que es el que lo valida. Por cuanto el procedimiento cualitativo se caracteriza por ser sustentado en la subjetividad, lo que conlleva el riesgo de ser rechazado o confirmado por un análisis cuantitativo.

2.2 Herramientas de Hardware

2.2.1 Dispositivo móvil

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con

memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablet PCs. (Alonso, 2011)

Hoy en día existe un grupo de dispositivos móviles denominados smartphones, que son teléfonos que poseen más prestaciones que un teléfono normal, la característica principal de estos dispositivos es que permiten la instalación de aplicaciones, las mismas que son desarrolladas para cumplir ciertas funciones que un teléfono normal no podría hacer. Para complementar a estos dispositivos, existen otros aparatos denominados wereable (vestible). Como definición tenemos que: Wereable hace referencia al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo interactuando continuamente con el usuario y con otros dispositivos con la finalidad de realizar alguna función específica. (Corti, 2016)

Estos dispositivos pueden ser: relojes, pulseras o collares; que sirven para registrar de manera continua las actividades físicas de las personas, esto permite obtener datos que facilitan la toma de decisiones acerca de la rutina de entrenamiento del deportista. La diferencia entre cada uno de estos dispositivos radica en los diferentes sensores que posee dependiendo de la actividad que vaya a realizar y la posición del cuerpo en la es usado. Si tomamos en cuenta la posición del cuerpo nos encontramos con gafas, relojes, pulseras, brazaletes y anillos inteligentes.

Dentro de los brazaletes que existen en el mercado nos encontramos con Myo mismo que se muestra en la Figura 1, un dispositivo wereable que permite la interacción con el usuario de una manera sencilla. Posee una variedad de aplicaciones en el mercado, su interfaz de programación permite desarrollar aplicaciones móviles, que al utilizar sus sensores, cumple con funciones como: utilizarlo como mouse, mover objetos, cambiar canciones, entre otros.

Existe una amplia comunidad que desarrolla día a día nuevas aplicaciones para este dispositivo, que ha tenido gran acogida por su facilidad de integración con los dispositivos móviles, debido a su interfaz de programación multiplataforma.



Figura 1 Brazaletes Myo. Elaboración propia.

En relación con lo expuesto y al compararlo con otros brazaletes, estos son diseñados para cumplir una función en específico, como por ejemplo los cuantificadores de actividad como el Xiaomi MiBand 2 o Up Move que solo realizan la función de enviar datos al dispositivo móvil sobre la actividad física realizada por un usuario, el brazaletes Myo puede ser utilizado en diferentes áreas como Realidad Virtual, Robótica, entre otras, esto gracias a la mencionada interfaz de programación y los sensores que posee este dispositivo.

La importancia de los smartphones y los de este tipo de dispositivos wereables con el deporte se basa en la incorporación de los sensores que están dentro de estos aparatos electrónicos que a través de aplicaciones que tienen los smartphones son capaces de verificar: movimientos, pasos, coordinación, ritmo cardiaco, calorías quemadas, etc. Esto permite obtener resultados precisos y óptimos para desarrollar las habilidades de los deportistas. En los deportes de precisión los dispositivos pueden ayudarnos a fijar el movimiento correcto que debe practicar la persona con el fin de garantizar un mejor desarrollo deportivo.

2.2.1.1 Sensores

Los dispositivos móviles utilizan una gran variedad de sensores, que son utilizados para mejorar la experiencia del usuario al momento de utilizar un dispositivo móvil o recolectar información de acciones en concreto para luego ser utilizadas por una aplicación.

2.2.1.1.1 Acelerómetro

Es un sensor electrónico que permite transformar un movimiento (que se activa al realizar una aceleración sobre el mismo) a una respuesta eléctrica que puede ser interpretado por un dispositivo electrónico.. (Lucía Elasri, 2014)

El sensor del acelerómetro mide un movimiento de forma lineal, con este podemos medir las variaciones de que se aplican sobre un dispositivo. En dispositivos móviles es muy utilizado para realizar movimientos en especial en juegos, donde se puede utilizar el dispositivo como mando de un auto, simulando el volante del mismo. Cuando una persona realiza un movimiento sobre un dispositivo móvil, el sensor detecta la magnitud de la aceleración que generó este movimiento y hacia qué dirección se lo ejecuta, enviando esta información a la aplicación para que realice la acción del movimiento.

Una de las limitaciones de este sensor es que no puede saber la rotación que tienen un elemento, por lo que es necesario siempre complementarlo con otro sensor para conocer la posición de un dispositivo con respecto a un sistema de coordenadas.

2.2.1.1.2 Giroscopio

El giroscopio es un dispositivo que posee un disco de giro libre que puede realizar giros a alta velocidad angular, todo esto alrededor de un eje instantáneo que siempre pasa a través de un punto fijo; es decir, a medida que este dispositivo gira, el disco sólido se mantiene sobre el mismo eje, indicando así únicamente la fuerza de gravedad en el centro . (Scarborough, J. 1958) . A partir de esta definición se construyó el sensor del giroscopio, que permite ver la variación de la rotación de un dispositivo, con la determinación de un punto fijo y la variación de rotación que existe alrededor del mismo.

Dentro de los smartphones, el giroscopio traduce el movimiento en todos los ejes del teléfono en respuestas eléctricas.

Un giroscopio incorporado dentro de los teléfonos permite:

- Crear panoramas e imágenes esféricas en 360°
- Realizar fotos 3D
- Crear aplicaciones de Realidad Aumentada y Realidad Virtual.
- Potenciar y complementar la funcionalidad del acelerómetro.

Una de las ventajas del giroscopio es que reacciona con rapidez y precisión a los cambios, pero va acumulando errores con el paso del tiempo.

2.2.1.1.3 Sentido de Orientación

Es el sensor capaz de dar respuestas eléctricas en función de la posición del dispositivo y su orientación en el espacio. Este es un sensor de tipo lógico que combina la funcionalidad del acelerómetro y el sensor magnetómetro (sensor que permite medir los campos magnéticos). Suelen ser representados por cuaterniones, que son una extensión de los números reales, mismos que poseen 4 dimensiones para poder notarlos. Las tres primeras dimensiones corresponden a los componentes X, Y y Z del vector del cuaternio, mientras que la cuarta es el componente escalar de la misma.

2.3 Herramientas de Software

2.3.1 Android Studio

Entorno de desarrollo integrado que permite la programación en Android de manera nativa. Posee un sistema de construcción basado en Gradle. Gradle permite al desarrollador aplicar distintas configuraciones del mismo código para producir diferentes versiones del código de una aplicación. Gradle favorece a la reutilización de código y la integración en un servidor en construcción. (Horeense B, 2014)

2.3.2 Laravel

Framework de programación de lenguaje PHP que permite la creación aplicaciones y servicios orientados a la web. La característica principal de este framework es que permite el desarrollo de una forma simple, a través del uso de paquetes que permiten realizar operaciones de forma sencilla.

Mediante el uso de Eloquent ORM permite la comunicación con la base de datos de una manera fácil, proveyendo así un mecanismo para agilizar las consultas realizadas a través de la base de datos.

2.3.3 Git

Software que permite el control de versionamiento de código , este registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante (Chacon, S. & Straub, B, 2014). La diferencia con otros programas de versionamiento radica en que Git guarda el estado completo del proyecto cuando se registra un cambio, mientras que los demás programas guardan solo el estado de los archivos modificados. Mediante Git se puede llevar un control de todos los cambios realizados en un proyecto de una manera eficaz.

2.3.4 MySql

Sistema de Gestión de Base de Datos, es código abierto y se basa en el lenguaje SQL. Dentro de los gestores de base de Datos, este ofrece un delos mejores escenarios en varias formas, alta estabilidad, alta seguridad y un excelente soporte como muestra la Figura 1 (Suehring S, 2002).

2.3.5 R: The R Project for Statistical Computing

Sistema de procesamiento estadístico de muy alta calidad que está disponible de forma libre para la mayoría de plataformas informáticas. Al ser una plataforma de código abierto posee una comunidad que se dedica al desarrollo de nuevas funcionalidades y librerías, una de estas librerías es Similarity Measures desarrollada por Kevin Toohey. Esta librería posee 4 tipos de algoritmos para la comparación de movimientos: Longest Common Subsequence (LCSS), Frechet Distance, Edit Distance y Dynamic Time Warping (DTW).

2.4 Algoritmos de comparación de movimientos

Hoy en día, existen varios estudios relacionados con la comparación de matrices que varían en relación al tiempo, para la verificación de las similitudes de valores tomados de distintas investigaciones.

Las medidas de similitud de movimiento existentes pueden clasificarse en dos clases, como se muestra en la Figura 2 (Magdy, Sakr, Mostafa, & El-Bahnasy, 2015)

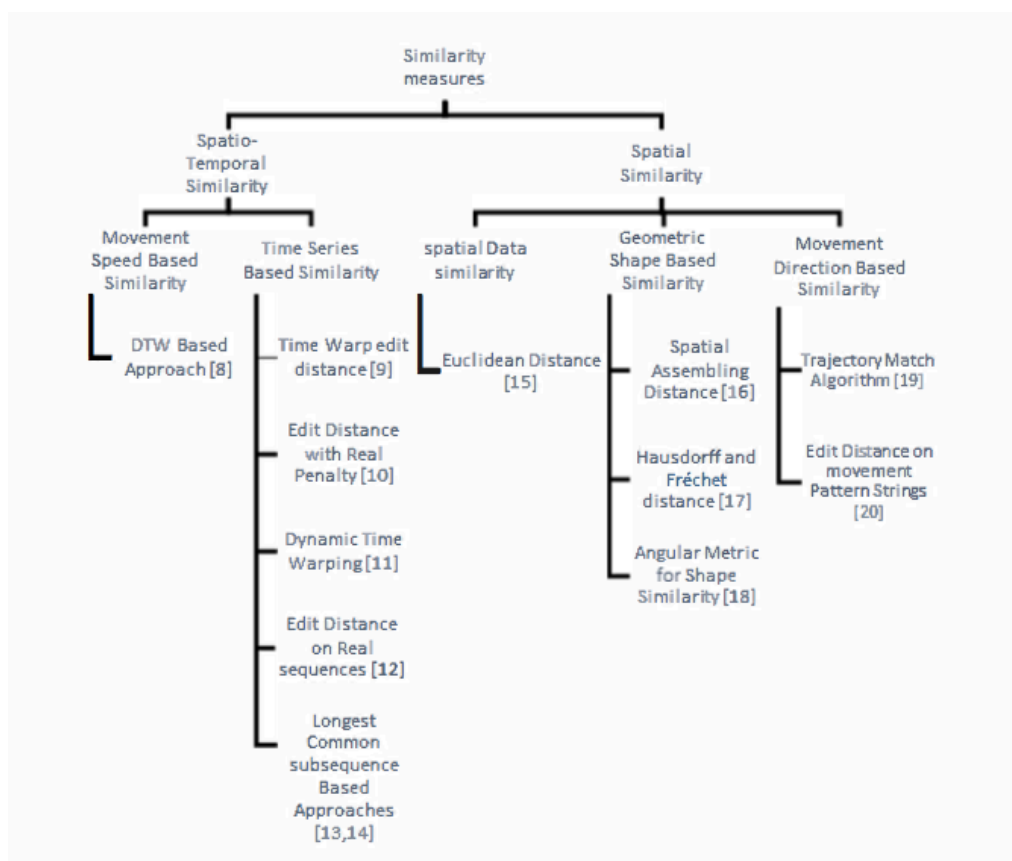


Figura 2 Clasificación de Similarity Measures (Magdy, et al. 2015, p.614)

Podemos ver que dependiendo del tipo de comparación que se vaya a realizar, se puede utilizar cierto tipo de algoritmo. Los algoritmos dedicados a la similitud espacial se centra en buscar similitudes entre las formas que poseen los movimientos a estudiar, mientras que la similitud espacio-temporal se toma en cuenta las similitudes tanto en el tiempo como en los datos del

movimiento generado.

Si deseamos realizar la comparación de movimientos generados a través de los sensores de un dispositivo móvil, uno de los principales desafíos que se presentan es la aplicación de los algoritmos en la programación de la aplicación propiamente dicha, en este sentido se debe tomar en cuenta las limitaciones de hardware que poseen estos dispositivos para realizar cálculos.

Una de las opciones que se propone es el uso de un procesador estadístico como es R, mismo que puede realizar la comparación de los movimientos generados a través de la librería Similarity Measures. Esta librería posee cuatro algoritmos de comparación, en esta investigación se hablará de los mismos y de sus aplicaciones.

2.5.1 DTW (Dynamic Time Warping)

Es la técnica utilizada para la comparación de matrices que se encuentran desfasadas a lo largo del tiempo, mediante este algoritmo se desplaza a los vectores que serán comparados, para alinearlos de acuerdo a los patrones que se tenga en los mismos.

Como se muestra en la Figura 3 la trayectoria superior es similar a la inferior con ciertos desfases, las líneas que van desde una trayectoria a otra representan la distancia mínima entre estos segmentos a pesar de que se encuentren desfasados.



Figura 3 Dos series similares desfasadas en el tiempo (Magdy, et al. 2015, p.615)

Una de las aplicaciones de este algoritmo de comparación es el Reconocimiento de Voz, en este ámbito, uno de los mayores problemas cuando se comparan patrones de lenguaje es la

variación en la tasas de conversión de la voz, por lo que se necesita un algoritmo que ayude a normalizar los datos para poder ser procesados con exactitud. Aquí se utiliza el DTW para proporcionar una medida más realista entre la similitud de patrones ya que proporciona un método de alineación entre dos patrones distintos (Madiseti, V. ,2009).

Este algoritmo también es herramienta utilizado en Minería de Datos (Data Mining) para diferentes problemas como la clasificación , agrupación y detección de anomalías. Berndt y Clifford introdujeron el concepto de DTW a la comunidad de minería de datos. Ellos mostraron cómo usar la programación dinámica para calcular la distancia DTW y demostraron su aplicación como una medida de similitud de series temporales. (Berndt, D & Clifford, J., 1994).

Uno de sus principales defectos es el tiempo de respuesta que tienen este algoritmo en la comparación de matrices de gran tamaño, para lo cual ha habido propuestas de mejorar. Una de ellas es usar una técnica denominada FastMap para indexar series de tiempo con la medida obtenida por el algoritmo DTW, sin embargo la misma puede obtener falsos negativos (Yi, B., Jagadish, H. , & Faloutsos, C.,1998)

2.5.2 Edit Distance on Real sequence

Este algoritmo se basa en el proceso Edit Distance, utilizado para la comparación en cadenas de caracteres.

El cálculo se realiza mediante pesos que se dan por cada cambio que se debe realizar en una de las matrices para que sean iguales. Por ejemplo en el estudio mencionado se dice que si tenemos dos cadenas de caracteres una A y una B, EditDistance (A,B) es el número de operaciones (inserción, eliminación o reemplazo) que se necesitan para que las dos cadenas sean similares.

En el caso de la comparación de dos vectores, si tomamos en cuenta un punto (X_i, Y_i) y un punto (X_j, Y_j) , y la distancia Euclidiana que hay entre los dos como D , un valor ϵ que representa la distancia máxima permitida para indicar que las trayectorias son similares y R que es el resultado final, si $D \leq \epsilon$ entonces $R=0$ casi contrario $R=1$. Al sumar todos los

resultados obtendremos un entero, que representara la similitud entre estas dos vectores, entre menor sea el resultado , mayor grado de similitud existirá en dicha comparación.

Edit Distance on Real sequence es efectivo en trayectorias que poseen un poco de ruido, y en comparación con algoritmos como LCSS o DTW poseen el mismo costo de computación (Lei Chen, M & Oria, V., 2005)

En la Figura 4 se muestra la comparación que tiene la eficacia del algoritmo Edit Distance en comparación con la distancia Euclidiana y el algoritmo DTW.

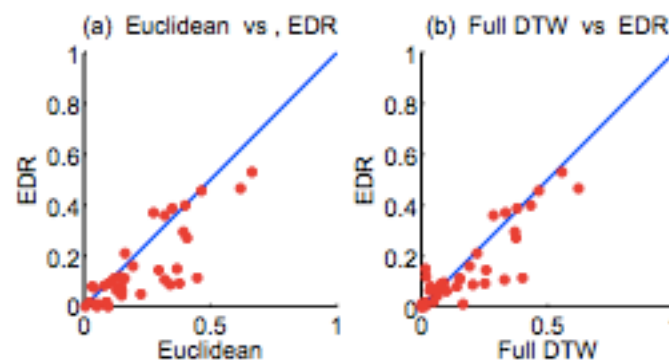


Figura 4 Comparación de precisión entre algoritmos (Ding, H, et al. 2008, p 1548)

En está investigación se llegó a la conclusión de que a pesar que se cree que el algoritmo EDR es superior en eficacia al DTW, esto no es verdad en todos los casos, pues los experimentos realizados en este estudio demuestran lo contrario.

2.5.3 LCSS (Longest Common Subsequence)

Es una variación del algoritmo Edit Distance, es un algoritmo que permite la comparación de dos matrices de diferentes dimensiones. Lo que intenta este algoritmo es ampliar las dos matrices que se desean comparar para buscar similitudes entre sí. Haciendo referencia a lo expuesto anteriormente en Edit Distance, este algoritmo también usa la distancia mas corta entre dos puntos, sin embargo también toma puntos que se encuentran dentro de un umbral, definido al momento de realizar el proceso del calculo, como se puede observar en la Figura 5,

si tomamos en cuenta el umbral en color plomo, permite que los dos vectores se asemejen un poco mas y su comparación sea un poco mas eficiente cuando existe desplazamientos en el tiempo.

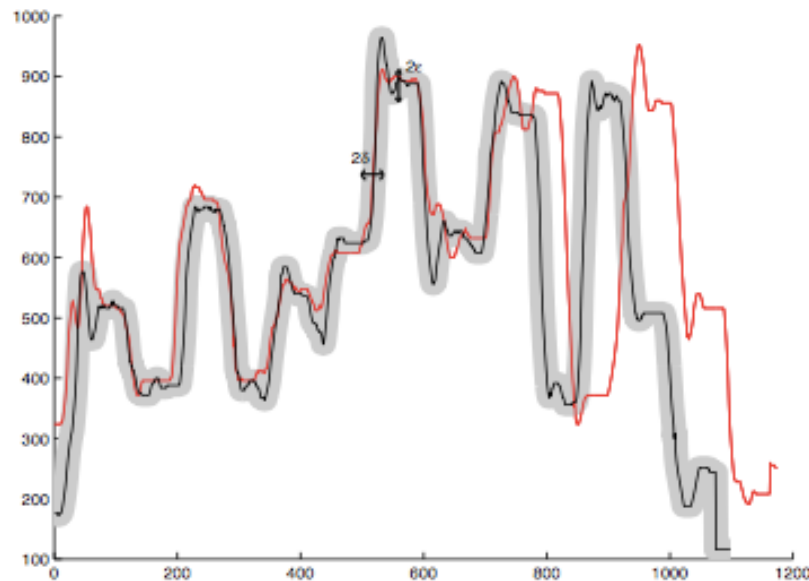


Figura 5 Umbral en el algoritmo LCSS (Vlachos, et al. 2002, p.675)

Las ventajas de este algoritmo son:

- 1) Algunos elementos no se toman en cuenta a diferencia de la distancia Euclidiana y el algoritmo DTW donde todos los puntos deben estar emparejados.
- 2) El modelo LCSS permite un cálculo aproximado más eficiente.

Una de las aplicaciones que se dio a este algoritmo es en el ámbito de las Bases de Datos de Trayectorias, que consisten en objetos que cambian de posición en todo el tiempo (por ejemplo el movimiento de un vehículo). Uno de los problemas que presenta este algoritmo tiene el inconveniente de que penaliza los puntos que estaban marginalmente fuera de la región de coincidencia asignándoles un valor de similitud cero (Nikos, P., et al. 2007).

Sin embargo se ha buscado mejorar el algoritmo a través una función de distancia no métrica basada en la LCSS en conjunción con una función de concordancia sigmoideas (Vlachos, M., et al, 2002).

2.5.4 Distancia Frechet

Es una medida de similitud entre curvas que tiene en cuenta la ubicación y ordenación de los puntos a lo largo de las curvas. Se le puede definir de la siguiente manera : “Un hombre está paseando a un perro con una correa: el hombre puede moverse en una curva, el perro en el otro; Ambos pueden variar su velocidad, pero el retroceso no está permitido. ¿Cuál es la longitud de la correa más corta pero que sea suficiente para atravesar ambas curvas?” En la Figura 6 se representa este movimiento, las líneas de color verde representarían la cuerda que separa al animal de su dueño y representarían la distancia de Frechet. Tomando en cuenta esta definición estamos hablando de dos espacios en las cuales se busca la distancia mínima entre los mismos a través de esta distancia.



Figura 6 Distancia de Frechet (Magdy, et al. 2015, p.616)

LeiChen,M. y Vincent Oria en su documento “Robust and Fast Similarity Search for Moving Object Trajectories” nos muestra una figura que hace referencia a tres de los cuatro algoritmos presentados, en el cual nos hace notar las ventajas que posee cada uno en diferencia del otro y el tiempo de cómputo que representa cada uno para realizar el cálculo tal y como se muestra en la Figura 7

	Definition	Local Time Shifting	Noise	Metric	Computation Cost
$Eu(R, S) = \sqrt{\sum_{i=1}^n dist(r_i, s_i)}$	$dist(r_i, s_i) = (r_{i,x} - s_{i,x})^2 + (r_{i,y} - s_{i,y})^2$ (1)			✓	$O(n)$
$DTW(R, S) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \text{ or } n = 0 \\ dist(r_1, s_1) + \min\{DTW(Rest(R), Rest(S)), \\ DTW(Rest(R), S), DTW(R, Rest(S))\} & \text{otherwise} \end{cases}$ (2)		✓			$O(n^2)$
$ERP(R, S) = \begin{cases} \sum_{i=1}^n dist(s_i, g) & \text{if } m = 0 \\ \sum_{i=1}^m dist(r_i, g) & \text{if } n = 0 \\ \min\{ERP(Rest(R), Rest(S)) + dist(t_1, s_1), \\ ERP(Rest(R), S) + dist(r_1, g), \\ ERP(R, Rest(S)) + dist(s_1, g)\} & \text{otherwise} \end{cases}$ (3)		✓		✓	$O(n^2)$
$LCSS(R, S) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ LCSS(Rest(R), Rest(S)) + 1 & \text{if } r_{1,x} - s_{1,x} \leq \epsilon \\ & \& r_{1,y} - s_{1,y} \leq \epsilon \\ \max\{LCSS(Rest(R), S), LCSS(R, Rest(S))\} & \text{otherwise} \end{cases}$ (4)		✓	✓		$O(n^2)$

Figura 7 Características de algoritmos de comparación de movimientos (Lei Chen, et al. 2005, p493)

Aquí se puede apreciar que según este estudio, el algoritmo LCSS permite un mejor cálculo en relación a movimientos donde exista ruido. Este tipo de diferencias pueden hacer un algoritmo mejor que otro, en diferentes situaciones. Sin embargo algunos pueden ser complementados de acuerdo al análisis de resultados del otro.

2.6 Resumen

La biomecánica ayuda a entender cómo funcionan los movimientos realizados por los deportistas para poderlos realizar de una manera correcta. Gracias al análisis de los mismos de manera cuantitativa se puede representar un movimiento en una serie de números, este es el caso de los valores que se obtienen a través de los sensores de los dispositivos móviles, quienes son los encargados de transformar los movimientos en datos que pueden ser procesados por diversas herramientas matemáticas, mismas que mediante algoritmos de comparación de matrices como DTW, LCSS, Edit Distance on Real Sequences o Frechet, permiten realizar un análisis de similitud de movimientos a través de la interpretación de los resultados que se obtienen con la utilización de los mismos

3. Propuesta

Para plantear la solución se analizó los antecedentes discutidos en el capítulo anterior y se planteó una arquitectura que pueda utilizar mejores recursos para el cómputo de los resultados.

La aplicación que se plantea, tiene dos perfiles:

- La persona que realiza el movimiento correcto al cual denominaremos entrenador
- La persona que intenta imitar el movimiento que se lo llamará aprendiz.

El entrenador tendrá la capacidad de realizar la grabación de movimientos a los cuales se les tomará como “correctos”, mientras que el aprendiz deberá escoger el movimiento que desea replicar para verificar la similitud con el movimiento escogido. Cada uno de estos movimientos es traducido en valores representados por coordenadas para los distintos sensores.

3.1 Estructura

Para poder realizar la comparación de similitud entre dos movimientos se propone la estructura que se muestra en la Figura 8

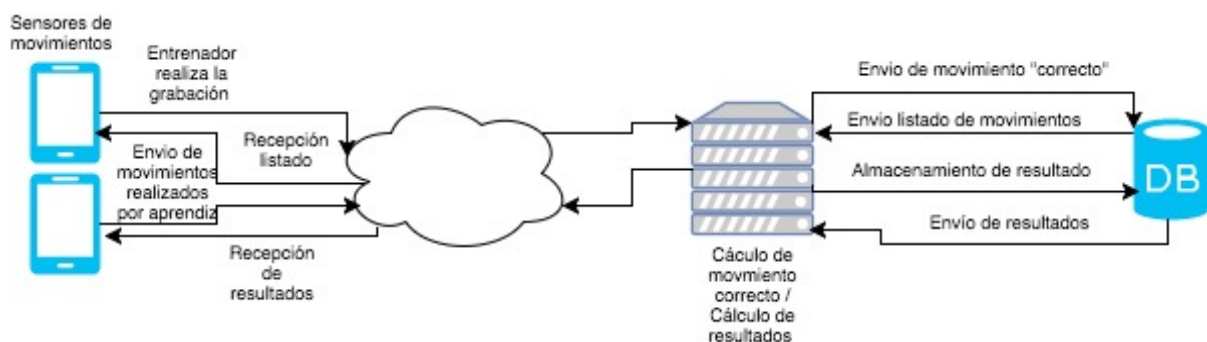


Figura 8 Propuesta de Arquitectura Inicial. Elaboración propia

En la Figura 8 se muestra un dispositivo móvil que puede ser un teléfono iPhone con sistema operativo iOS, que será el encargado de recibir la información de los sensores que cuantifican cada uno de los movimientos mediante una aplicación móvil desarrollada para este objetivo.

Debido al sistema operativo la aplicación puede ser desarrollada con Swift. Los movimientos son traducidos a coordenadas mediante el uso de los sensores del dispositivo, con los cuales tendremos valores en X, Y y Z en los sensores de aceleración y giroscopio, mientras que tendremos valores en X, Y, Z y W en el sentido de orientación del movimiento, estos datos son enviados a un servidor que será el encargado de realizar los cálculos matemáticos, la forma de envío será a través del consumo de un servicio web que tendrá un servidor que se encuentra en la nube. Este servidor tendrá una conexión a una base de datos que almacenará la información de cada uno de los movimientos enviados para un posterior procesamiento.

Antes de realizar cualquier movimiento, el usuario debe ingresar a la aplicación con un usuario y contraseña como muestra la Figura 9, el mismo que permitirá diferenciar un usuario entrenador de un aprendiz.



Figura 9 Login de la aplicación. Elaboración propia

Cada vez que un usuario entrenador quiera realizar la grabación de un movimiento, la aplicación le solicitará realizar este movimiento 3 veces, una vez que finalice los mismos, los datos recolectados serán enviados en bloque al webservice para intentar encontrar un movimiento promedio de cada uno de los sensores, el cual se tomará como un movimiento correcto. Cada usuario entrenador tendrá un listado de todos los movimientos que el ha

generado.

Con este tipo de arquitectura se intenta centralizar todos los movimientos recolectados por el dispositivo móvil, para que cualquier usuario pueda utilizar los mismos y realizar la comparación con sus propios movimientos.

Cada vez que un usuario quiera intentar reproducir un movimiento realizado por un entrenador, este deberá escoger el movimiento de un listado de opciones como muestra la Figura 10. Una vez seleccionado el movimiento, la aplicación nos indicará que podemos proceder con la grabación del movimiento. Cuando termina la grabación la aplicación enviará los datos del nuevo movimiento realizado, adicionando también la identificación del movimiento con el que se quiere realizar la comparación.

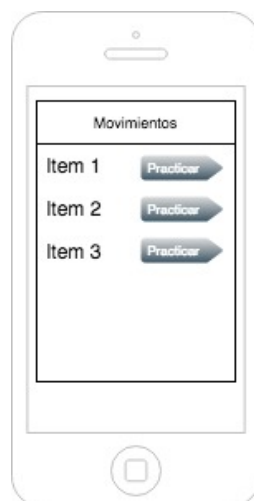


Figura 10 Listado de movimientos. Elaboración propia

El servidor será el encargado de realizar los cálculos matemáticos que permitirán determinar la similitud de los movimientos mediante una aplicación de procesamiento estadístico, todos los resultados serán almacenados en la base de datos para tener un registro sobre cada una de las comparaciones que realizó el usuario.

Cuando el servidor reciba los dos movimientos, realizará primero una inserción en la base de datos con el nuevo movimiento, para llevar un registro de los datos enviados por el usuario, y realizará una consulta para extraer los datos del movimiento a comparar. Estos datos deberán ser modificados para que la aplicación de procesamiento estadístico pueda leerlos y pueda procesarlos para arrojar un resultado, de esta forma el servidor deberá convertir los movimientos en matrices para que las mismas puedan ser leídas por la aplicación que procesará los algoritmos de similitud de trayectorias.

Una vez que esta aplicación estadística arroje un resultado, el servidor almacenará el mismo en la base de datos, para llevar un historial de los movimientos realizados por cada usuario, de manera que el mismo pueda llevar un control sobre sus avances realizados.

Una vez que el servidor termine de almacenar estos resultados, se enviará la respuesta al dispositivo móvil para poder mostrar los resultados al usuario final.

El usuario final realizará la consulta de los resultados de sus movimientos a través de la selección de los mismos dentro de un listado tal como muestra la Figura 11, la aplicación enviará la identificación del movimiento seleccionado al servidor, el servidor realizará la llamada a la base de datos con el id del movimiento, la respuesta será el resultado que fue almacenado en el cálculo de comparación, este resultado será un valor decimal que representará el porcentaje de similitud.

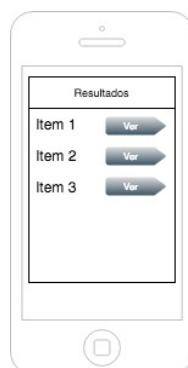


Figura 11 Selección de resultados . Elaboración propia

Una vez que el servidor obtiene este resultado será enviado a la aplicación para mostrarlo al usuario, este resultado será mostrado tanto de en forma de texto como en forma visual, tal como muestra la Figura 12

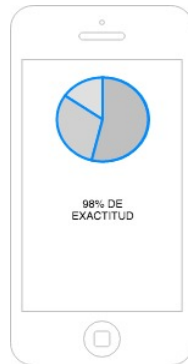


Figura 12 Resultado con porcentaje de exactitud obtenido en la aplicación. Elaboración propia

4. Implementación

Este capítulo describe como se realizó la implementación del proyecto, está dividido en tres partes principales que se muestran a continuación.

4.1 Test De Algoritmos Con Movimientos

Uno de los temas principales de esta investigación es la comparación de movimientos a través de algoritmos de comparación, para encontrar el que mejor se adapte a nuestras necesidades.

4.1.1. Análisis Gráfico

El procedimiento inicial que se hizo fue la conexión del dispositivo Myo a una aplicación móvil pequeña, que utiliza el SDK del dispositivo para poder tomar los valores de los 3 sensores con los que se está trabajando (Acelerómetro, Giroscopio y Orientación). Los valores rescatados de este procedimiento fueron almacenados en archivos de texto separados por comas (Csv) para poder realizar un análisis de los mismos.

En la Figura 13 se puede apreciar los datos que fueron tomados de un movimiento en particular, en el cual nos refleja:

- El tiempo en el que se realizó el movimiento
- Los 3 puntos del vector que representan los valores del acelerómetro
- Los 4 puntos que representan el cuaternion para representar la orientación
- Los 3 puntos del vector que representan a los valores tomados del giroscopio.

	A	B	C	D	E	F	G	H	I	J	K	L
1	10422387;	0.57855224609375;	0.461669921875;	0.64312744140625;	-0.196044921875;	0.9404296875;	0.244140625;	-0.06884765625;	-38.5;	-34.625;	-12.4375	
2	10422419;	0.57855224609375;	0.461669921875;	0.64312744140625;	-0.196044921875;	0.9404296875;	0.244140625;	-0.06884765625;	-38.5;	-34.625;	-12.4375	
3	10422427;	0.5828857421875;	0.45953369140625;	0.642333984375;	-0.1907958984375;	0.95751953125;	0.23388671875;	-0.1025390625;	-28.75;	-30.5;	-9.625	
4	10422430;	0.58642578125;	0.45745849609375;	0.64215087890625;	-0.18524169921875;	0.94775390625;	0.31005859375;	-0.0888671875;	-29.6875;	-27.3125;	-8.7	
5	10422476;	0.58978271484375;	0.4547119140625;	0.642822265625;	-0.17926025390625;	0.923828125;	0.35107421875;	-0.0751953125;	-34.625;	-25.0;	-6.0	
6	10422477;	0.59344482421875;	0.45111083984375;	0.64337158203125;	-0.1739501953125;	0.9169921875;	0.31396484375;	-0.09130859375;	-36.375;	-23.0;	-2.25	

Figura 13 Ejemplo de datos generados en CSV. Elaboración propia

Una vez que ya se pudo obtener valores de los sensores, se procedió a hacer un análisis gráfico-comparativo para comprobar la similitud entre dos movimientos realizados por la misma persona en distinto tiempo, la secuencia del movimiento se la puede visualizar en la Figura 14.



Figura 14 Secuencias de movimientos utilizados para la comparación de movimientos.

Elaboración propia

En la Figura 15 se representa una gráfica comparativa entre los dos movimientos en el sensor de orientación (Que es el más difícil de comparar por la naturaleza que tiene el mismo, puesto que un movimiento en el sentido de orientación puede ser representado de varias maneras), los puntos de la izquierda que simbolizan al primer movimiento son gráficamente similares a los puntos de la derecha en el sensor de orientación en todos sus ejes.

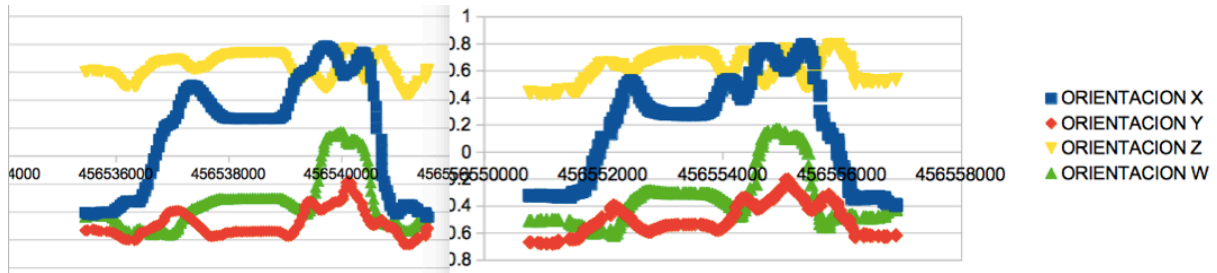


Figura 15 Gráfico comparativo de movimientos similares realizados por una persona.

Elaboración propia

Gráficamente se nota la similitud de las curvas, tomando en cuenta los puntos máximos y mínimos que se observan dentro de la imagen; sin embargo, para la aplicación es necesario entender los resultados que nos arrojan los algoritmos a acuerdo a los datos analizados.

De la misma manera que se hizo con un movimiento similar realizado por la misma persona, se realizó un mismo movimiento pero con personas distintas, de igual manera se tomó como referencia los valores obtenidos con el sensor de orientación, se realizó la gráfica de los datos y se obtuvo la Figura 16.

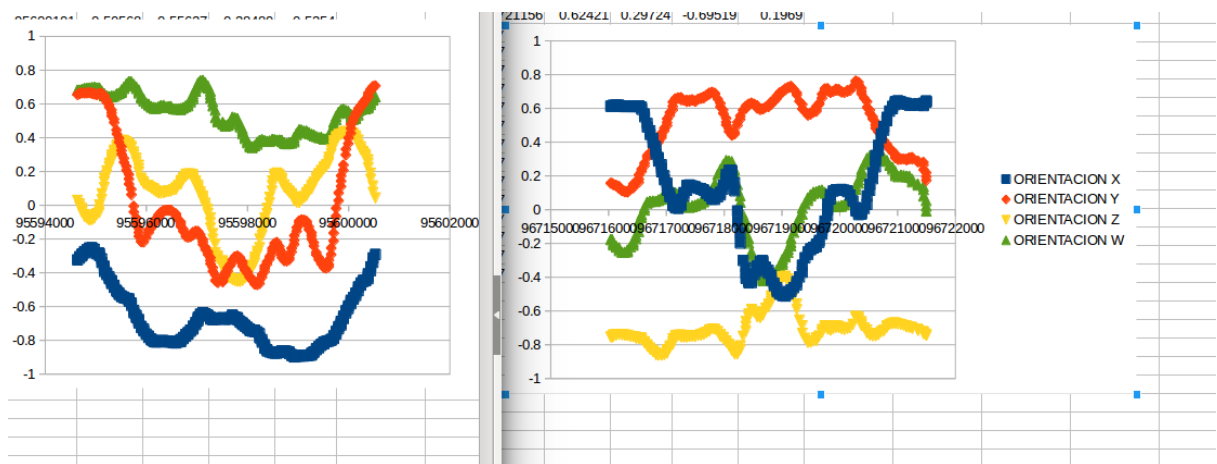


Figura 16 Comparación gráfica de dos movimientos similares en distintas personas en el sensor de orientación. Elaboración propia

Como se puede ver en la gráfica, no existe similitud entre los dos movimientos en relación al sensor de orientación a pesar de que los dos movimientos fueron iguales. Pero si hacemos la misma representación gráfica en el sensor del acelerómetro se obtuvo la Figura 17.

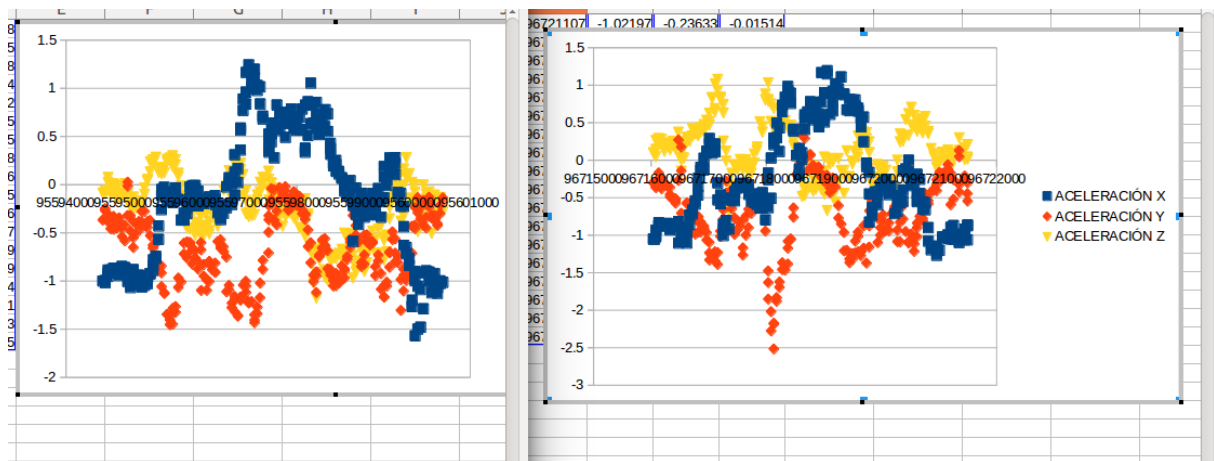


Figura 17 Comparación gráfica de dos movimientos similares en distintas personas en el sensor de aceleración. Elaboración propia

La gráfica nos muestra que si existe una tendencia a que el movimiento sea similar, en el eje X los puntos máximos y mínimos son muy parecidos y en el eje Z existe una similitud en las curvas pero con un cierto desfase.

4.1.2 Implementación con R

Después del análisis gráfico, se buscó la forma de integrar los algoritmos con la aplicación móvil, para dicha integración se utilizó R, que permite el uso de los algoritmos mencionados anteriormente, pertenecientes a la librería Similarity Measures.

La librería SimilarityMeasures posee varias funciones para realizar los cálculos, pero hablaremos de las cuatro principales que se utilizaron y los parámetros que utilizan las mismas para su funcionamiento.

Para poder comparar los movimientos generados anteriormente en CSV y poder utilizar los algoritmos de R, se instaló servidor local de R, en el cual se instaló de manera global la librería SimilarityMeasures.

Una vez en R se transformó los archivos CSV en matrices para que puedan ser interpretados por los algoritmos y se procedió a realizar pruebas de los diferentes algoritmos para poder analizar los resultados.

Para utilizar cada uno de los algoritmos hay que tomar los parámetros que acepta cada función, los mismos que se detallaran a continuación:

1) DTW(Traj1, Traj2, PointSpacing = -1)

- a) Traj1 : Representa la primera matriz que se va a utilizar para la comparación de movimientos.
- b) Traj2 : Representa la segunda matriz que se va a utilizar para la comparación de movimientos, no es necesario que las dos trayectorias tengan la misma longitud.
- c) PointSpacing : Un valor entero que representa la diferencia de índice máxima entre la trayectoria1 y la trayectoria2 permitida para realizar el cálculo. Un valor negativo establece el espaciado entre puntos ilimitado.

2) EditDistance(Traj1, Traj2, PointDistance = 20)

- a) Traj1 : Representa la primera matriz que se va a utilizar para la comparación de movimientos.
- b) Traj2 : Representa la segunda matriz que se va a utilizar para la comparación de movimientos, no es necesario que las dos trayectorias tengan la misma longitud.
- c) PointDistance : Es un valor flotante que representa la distancia máxima aceptada en cada dimensión para que un punto pueda ser considerado como equivalente

3) LCSS (traj1, traj2, pointSpacing=-1, pointDistance=20, ErrorMarg=2, ReturnTrans = FALSE)

- a) Traj1 : Representa la primera matriz que se va a utilizar para la comparación de movimientos.
- b) Traj2 : Representa la segunda matriz que se va a utilizar para la comparación de movimientos, no es necesario que las dos trayectorias tengan la misma longitud.

- c) PointSpacing : Un valor entero que representa la diferencia de índice máxima entre la trayectoria1 y la trayectoria2 permitida para realizar el cálculo. Un valor negativo establece el espaciado entre puntos ilimitado.
- d) PointDistance : Es un valor flotante que representa la distancia máxima aceptada en cada dimensión para que un punto pueda ser considerado como equivalente
- e) ErrorMarg: Es un flotante que nos indica el margen de error aceptado, tiene que ver con la precisión y velocidad del cálculo
- f) ReturnTrans: Si es true, permite que la mejor translation encontrada sea retornada

4) Frechet(Traj1, Traj2, testLeash=-1)

- a) Traj1 : Representa la primera matriz que se va a utilizar para la comparación de movimientos.
- b) Traj2 : Representa la segunda matriz que se va a utilizar para la comparación de movimientos, no es necesario que las dos trayectorias tengan la misma longitud.
- c) Teast Leash: Un valor que si es positivo, representa la correa entre las dos trayectorias que sirve para realizar el cálculo.

Para realizar pruebas, se enviaron al inicio datos con más de 250 registros de movimientos similares y distintos es decir en total 3 matrices de 250 registros en las 3 y 4 dimensiones, sin embargo y a pesar de tener un buen servidor en donde se encuentra alojado R, se obtuvo demora de incluso días para dar el resultado, en especial en los algoritmos de Frechet y LCSS.

En la Figura 18 nos muestra el resultado de una de las pruebas que se corrieron en R con los archivos generados , sin embargo como se mencionó anteriormente el tiempo de ejecución era muy largo.

```

218,] 0.953125000 0.216308594 -0.047851563
219,] 0.910156250 0.344238281 -0.123046875
220,] 0.904296875 0.413574219 -0.152832031
221,] 0.925781250 0.437988281 -0.170410156
222,] 0.884277344 0.390625000 -0.089355469
223,] 0.888671875 0.282226563 -0.120605469
224,] 0.882812500 0.312988281 -0.104980469
225,] 0.873046875 0.348632813 -0.104003906
226,] 0.878906250 0.391113281 -0.111328125
227,] 0.894042969 0.465332031 -0.143066406
228,] 0.858886719 0.430175781 -0.136718750
229,] 0.870117188 0.364746094 -0.136718750
230,] 0.893554688 0.349121094 -0.159179688
231,] 0.890136719 0.395996094 -0.162597656
232,] 0.872070313 0.401855469 -0.155761719
233,] 0.878417969 0.350585938 -0.145996094
234,] 0.884765625 0.388671875 -0.152832031
235,] 0.878417969 0.454101563 -0.124511719
236,] 0.885742188 0.430664063 -0.113281250
C
LCSS(temp1, temp2, 2, 2, 0.5)
1] 236

```

Figura 18 Resultados de las pruebas realizadas con todos los resultados de un movimiento.

Elaboración propia

Realizar este procedimiento de generación de CSV y ejecución de scripts en R tomaba mucho tiempo, por lo que se construyó un webservice para que ayude a la comparación de movimientos de manera dinámica y de igual manera, sirva como base para el desarrollo del webservice que va a utilizar la aplicación.

Esto permitió que las pruebas realizadas se puedan mejorar, y obtener así resultados en menor tiempo y con mayor eficacia.

4.2 Implementación del Webservice

El webservice fue desarrollado con un framework de PHP conocido como Laravel, en primera instancia lo que se realizó fue crear las denominadas migraciones, que permiten generar una base de datos de manera dinámica, de acuerdo a las necesidades que vayamos obteniendo.

4.2.1 Base De Datos

La base de datos generada para este proyecto fue variando de acuerdo a las necesidades que se fue obteniendo conforme transcurría la investigación, el resultado final se muestra en la Figura 19 , en donde se observa el diagrama lógico de la base de datos con sus respectivas relaciones.

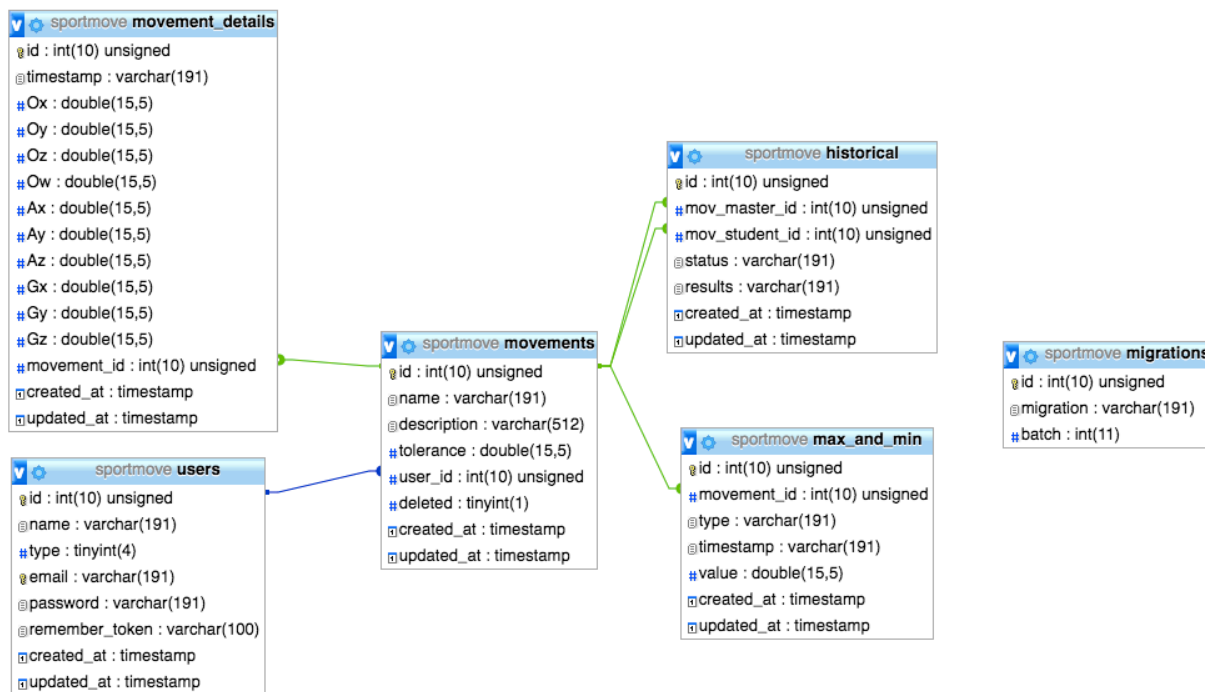


Figura 19 Diagrama Lógico de la Base de Datos. Elaboración propia

La base de datos fue estructurada de la siguiente manera de acuerdo al nombre de sus tablas y lo que almacena:

- Migrations: Es una tabla propia del framework Laravel, que guarda un historial de las migraciones que ha sido hechas con la herramienta.
- Users: Tabla que guarda información de los usuarios y perfiles que tiene cada uno
- Movements: Tabla que guarda la información general de un movimiento, que se enlaza con la tabla Users para poder conocer quien realizó dicho movimiento

- Movement_details: Tabla que almacena todos los valores de los sensores que generó cierto movimiento, se enlaza con la tabla Movements para conocer a que movimiento pertenece el detalle.
- Historical: Tabla que almacena el estado y el valor del resultado obtenido cuando finaliza la comparación de algoritmos
- Max_and_min: Tabla auxiliar que almacenó datos de los puntos máximo y mínimos de un movimiento

4.2.1 Test de Algoritmos en Función del Webservice

La función de comparación de movimientos del webservice recibe dos parámetros que representan el id de cada uno de los movimientos, así se pudo realizar una comparación más rápida para verificar los valores que retornan los algoritmos de comparación, es así por ejemplo si se desea realizar la comparación entre el mismo movimiento se realiza una petición como se muestra en la Figura 19



Figura 20 Resultado de algoritmos DTW y Edit Distance. Elaboración propia

Los valores que indican en el resultado obtenido son 0, pues se trata del mismo movimiento completamente, sin tener ningún tipo de variación; sin embargo, si realizamos la comparación con valores diferentes como muestra la Tabla 3 , empezamos a obtener los resultados que necesitamos evaluar.

Sensor \ Algoritmo	DTW	EditDistance
Acelerómetro	77.91666	65
Giroscopio	16646.63	209
Orientación	23.26363	42

Tabla 1 Resultado de movimientos similares

Sensor \ Algoritmo	DTW	EditDistance
Acelerómetro	107.8739	172
Giroscopio	18809.6	235
Orientación	77.83965	110

Tabla 2 Resultados de un mismo movimiento realizado por personas distintas

Sensor \ Algoritmo	DTW	EditDistance
Acelerómetro	314.4923	143
Giroscopio	27879.6	8
Orientación	297.6015	8

Tabla 3 Resultados con movimientos distintos

En la Tabla 1 se tiene resultados con dos movimientos similares realizados por la misma persona, mientras que en la Tabla 2 se muestran dos movimientos similares realizados por dos personas distintas y en la Tabla 3 observamos los resultados con movimientos distintos.

Como podemos ver las diferencias entre valores obtenidos con el acelerómetro en el algoritmo DTW de la Tabla 1 y 2 tienen una gran diferencia con el valor obtenido en la tabla 3, por lo que se podría establecer un valor de tolerancia en este algoritmo para indicar que el movimiento es similar a otro. Esta diferencia también se puede notar en el uso del Giroscopio, en el cual se nota la gran diferencia que existe entre los dos primeros valores de movimientos similares y la última dada.

De igual manera podemos observar que debido a la naturaleza de los valores que nos devuelve el sensor de Orientación, realizar una comparación entre los valores obtenidos por este sensor, es más compleja por las infinitas posibilidades con las que se puede representar el giro de un vector utilizando un cuaternion.

Se generó una tabla más completa que se encuentra en el Anexo A , en el cual podemos ver 3 tipos de movimientos comparados, realizados por la misma persona y por una persona distinta, que nos ayudará a ver el margen de error que poseen los algoritmos utilizados.

Separamos estos datos diferenciando solo los resultados del sensor de aceleración, puesto que es el sensor en el que se nota más similitud entre los valores dados, y de igual manera realizando solo el análisis con el algoritmo DTW, teniendo como resultados los mostrados en la Tabla 4.

		Aceleración
Movimiento	Tipo	DTW
Adelante Arriba Adelante	Misma Persona	99.29294
		79.7043
		80.67037
		89.06232
		67.85651
	Misma Persona Diferente Posición	80.98658
		89.06696

		85.26736
	Otra persona	152.0864
		132.5774
		122.9064
	Otra persona Diferente Posición	124.1609
		126.8174
149.05		
Bate Normal	Misma Persona	221.9777
		214.8365
		234.7064
	Otra persona	277.4121
		270.1237
		244.9816
Bate 45 grados	Misma Persona	211.0788
		228.3938
		209.9117
	Otra persona	270.1865
		278.5199
		274.7553

Tabla 4 Resultados del algoritmo DTW en el sensor de aceleración

En la primera columna tenemos el tipo de movimiento que se realizó, en la segunda quién realizó el movimiento y en qué forma lo realizó y por último en la tercera columna el resultado que se obtuvo. Estos movimientos se realizaron varias veces , por lo que se pueden ver variaciones entre los resultados obtenidos dentro de un mismo movimiento y realizado por la misma persona. De este tipo de variaciones se debe sacar una tolerancia que el algoritmo debe presentar para indicar si un movimiento es igual al otro.

Una vez finalizado este análisis se realizó diversas pruebas con el algoritmo LCSS y Frechet con un número inferior de datos por cada trayectoria, el primer análisis que se realizó fue el tiempo de respuesta tal como muestra la Tabla 5

TIEMPOS DE RESPUESTA	
NUMERO DE MOVIMIENTOS	TIEMPO(segundos)
20	3
40	7
60	15
80	65
100	132
120	314
140	460

Tabla 5 Tiempos de respuesta de LCSS

En esta tabla podemos ver el incremento en el tiempo de cómputo, que se va obteniendo conforme al número de movimientos que se ingresa en los scripts de comparación, como se muestra en el análisis de la Figura. El incremento es exponencial en relación al número de datos que se ingresan en el algoritmo, Por este motivo , el primer paso fue la separación de movimientos en segmentos de 30 y 50 datos, los resultados están en el Anexo B. En estos resultados se puede notar que el algoritmo LCSS presenta los mismos valores para movimientos similares y para movimientos opuestos, por lo que no ayudaría en el resultado de la comparación.

Para analizar si podemos utilizar el algoritmo de Fretchet en nuestro análisis de comparación, se tomó en cuenta el promedio de los movimientos correctos como base para una comparación y la diferencia que cada uno de los movimientos se colocó en porcentaje dando como resultado la Tabla 6.

Movimiento	Tipo	Frechet			
		30	Resultado(%)	50	Resultado(%)
Adelante Arriba Adelante	Misma Persona	0.2927011	-39	0.815296	23
		0.3271698	-32	0.6326642	-5
		0.5797557	21	0.6695048	1
		0.2933298	-39	0.415033	-38
		0.5706813	19	0.663161	0
	Misma Persona Diferente Posición	0.892544	86	0.7133788	7
		0.3625935	-24	0.6409482	-4
		0.520724	8	0.766617	15
	Otra persona	0.6881142	43	0.9467941	42
		1.630185	240	0.8607603	30
		1.308353	173	0.8694481	31
	Otra persona Diferente Posición	1.361481	184	1.062053	60
		1.546416	222	0.7952545	20
		1.202958	151	1.10199	66
	Bate Normal	Misma Persona	0.4341716	-10	0.8810095
0.6626212			38	0.9762198	47
0.9736206			103	1.280649	93
Otra persona		0.6896335	44	1.062917	60
		1.2011	150	1.283211	93
		1.905881	297	1.799293	171
Bate 45 grados	Misma Persona	0.4364846	-9	0.9223018	39
		0.7559804	58	1.27623	92

		0.9779294	104	1.409646	112
	Otra persona	1.600875	234	1.788796	169
		1.123144	134	1.321016	99
		1.813432	278	1.641312	147

Tabla 6 Resultados de algoritmo de Frechet para 30 y 50 entradas de datos

La Tabla 6 representa los resultados obtenidos de la utilización del algoritmo de Frechet en 30 y 50 datos de cada movimiento (se realizó esta división para ver resultados parciales, debido a que el tiempo de cálculo de un resultado total es demasiado extenso). Las columnas de resultado nos muestran la diferencia en porcentaje que se tiene en relación al promedio del resultado de un movimiento “correcto”.

Podemos observar en estos resultados que, a pesar de ser la misma persona realizando el mismo movimiento, existe un 39% de diferencia en la comparación de dos movimientos similares; este resultado disminuye, si se aumenta el número de movimientos analizados. Por otro lado, si tomamos en cuenta este resultado y el resultado de un movimiento distinto, que nos marca solo una diferencia negativa del 9% , se podría tomar erróneamente como similar un movimiento que no lo es. Por este motivo se descarta el algoritmo de Frechet en comparación de movimiento con número de datos pequeños (en este caso 30 datos seleccionados).

Si se realiza el mismo análisis en las pruebas realizadas con el algoritmo DTW tenemos los resultados mostrados en la Tabla 7.

Movimiento	Tipo	Aceleración	
		DTW	Resultado(%)
Adelante Arriba Adelante	Misma Persona	99.29294	18
		79.7043	-5
		80.67037	-4
		89.06232	6
		67.85651	-19

	Misma Persona Diferente Posición	80.98658	-4
		89.06696	6
		85.26736	2
	Otra persona	152.0864	81
		132.5774	58
		122.9064	46
	Otra persona Diferente Posición	124.1609	48
		126.8174	51
		149.05	77
Bate Normal	Misma Persona	221.9777	164
		214.8365	156
		234.7064	179
	Otra persona	277.4121	230
		270.1237	222
		244.9816	192
Bate 45 grados	Misma Persona	211.0788	151
		228.3938	172
		209.9117	150
	Otra persona	270.1865	222
		278.5199	232
		274.7553	227

Tabla 7 Diferencia en porcentaje de resultados del algoritmo DTW

Si tomamos en cuenta estos resultados obtenidos, vemos que la mayor diferencia entre movimientos similares es del 77% del valor original, mientras que la diferencia de movimientos similares va desde el 150%, es decir se podría tener el margen de tolerancia máximo de un valor aproximado del 80%. También tomaremos en cuenta que a pesar que la persona realice

el mismo movimiento en el mismo instante en la misma posición, existe variación entre los valores obtenidos, con un valor máximo obtenido del 18% sobre el valor promedio.

Los cálculos realizados anteriormente se realizaron con todos los datos generados con el dispositivo, sin embargo es importante realizar una comparación utilizando solo los puntos máximos y mínimos calculados en cada una de las curvas. Para esto se generó una tabla adicional para la generación de estos puntos por cada uno de los ejes de la aceleración, obteniendo los resultados detallados en el Anexo . En esta tabla podemos observar que en realidad usando pocos datos y con todos los algoritmos, no existe ningún tipo de tendencia en relación a los resultados obtenidos en ninguno de los algoritmos utilizados, por ejemplo el movimiento “Adelante Arriba Adelante ” con el uso de Frechet da un valor igual al movimiento del giro de un bate, a pesar de ser un movimiento completamente distinto.

En vista de estos resultados que no poseen ningún tipo de patrón, se decidió utilizar para el cálculo el algoritmo DTW, el cual nos da como resultado la menor distancia que existe entre dos trayectorias. Para un mejor análisis de los resultado de este algoritmo se realizó varias tablas para ver la diferencia de valores que existen entre cada uno de los resultados obtenido, como un ejemplo tenemos la Tabla 8 en la que se detallan algunos puntos.

Movimiento	Tipo	Resultado	Promedio	Diferencia con resultado	Promedio diferencia	Porcentaje de diferencia
Adelante Arriba Adelante	Misma Persona	99.29294	83.988417	15.304522	25.509986 79	23.297131 09
		79.7043		4.2841175		
		80.67037		3.3180475		
		89.06232		5.0739025		
		67.85651		16.131907		
	Misma Persona Diferente Posición	80.98658	3.0018375			
		89.06696	5.0785425			
		85.26736	1.2789425			

		152.0864		68.097982
	Otra persona	132.5774		48.588982
		122.9064		38.917982
	Otra persona	124.1609		40.172482
	Diferente	126.8174		42.828982
	Posición	149.05		65.061582
Bate Normal	Misma Persona	221.9777		
		214.8365		
		234.7064		
	Otra persona	277.4121		
		270.1237		
		244.9816		
Bate 45 grados	Misma Persona	211.0788		
		228.3938		
		209.9117		
	Otra persona	270.1865		
		278.5199		
		274.7553		

Tabla 8 Cálculo de margen de tolerancia de los resultados del algoritmo DTW

En la columna resultados tenemos el resultado obtenido de la utilización del algoritmo en comparación con los movimientos detallados en cada una de las filas, de este resultado se obtiene un promedio que es la siguiente columna, a este promedio se le hace la diferencia en relación a cada uno de los resultados, obteniendo la quinta columna. De esta diferencia se obtiene un promedio que representaría la tolerancia que se tiene sobre el promedio base del resultado del algoritmo.

Si representamos esta tolerancia en porcentaje, tenemos la ultima columna. De este proceso se obtuvo un promedio de aproximadamente un 26% de nivel de aceptación para indicar si un movimiento es similar a otro. Esto quiere decir que si se toma en cuenta el valor promedio que se tiene en los movimientos se puede añadir o restar un 26% a dicho valor y si el resultado de la aplicación del algoritmo DTW en relación promedio obtenido se encuentra dentro del 26% de este margen, se podría tomar en cuenta como un valor similar al movimiento, mientras que si la diferencia es mayor a dicho valor se le tomará en cuenta como un valor proporcionalmente similar de acuerdo al valor que se vaya representando.

De acuerdo al análisis anterior se realizó la programación del algoritmo de comparación de movimiento dentro del servicio construido con Laravel, en este servicio de comparación lo que se hace es comparar 3 movimientos similares desarrollados por una sola persona y se almacena en la base de datos el resultado promedio del análisis del algoritmo. Una vez que se almacena este valor se tiene ya el valor sobre la cual se va a comparar los márgenes de aceptación que mencionamos anteriormente, obteniendo los resultados para el análisis final.

4.3 Implementación de la Aplicación Móvil

Para poder realizar la comparación de movimientos final, se generó un esquema general, que se muestra en la Figura 21:



Figura 21 Arquitectura de la solución . Elaboración propia

En primera instancia la aplicación recibe la información de los sensores, mediante el dispositivo por parte del instructor, la aplicación envía esta información al webservice, que se encarga de almacenar la información en la base de datos junto a la información de la persona que realizó el movimiento.

Si es el caso de un usuario aprendiz, se realizará el mismo procedimiento anterior para almacenar el movimiento, la variación se produce en el proceso de almacenamiento, pues el webservice recibe las identificaciones de los movimientos que se van a comparar, crea dos archivos CSV con la información y envía esta información a R mediante un script. R realiza la comparación de los movimientos y arroja un resultado que es procesado de igual manera por el webservice para ser devuelto luego a la aplicación.

4.3.1 INTERFAZ DEL USUARIO

En primer lugar la aplicación va a pedir al usuario que ingrese su usuario y contraseña, tal como muestra la Figura 22

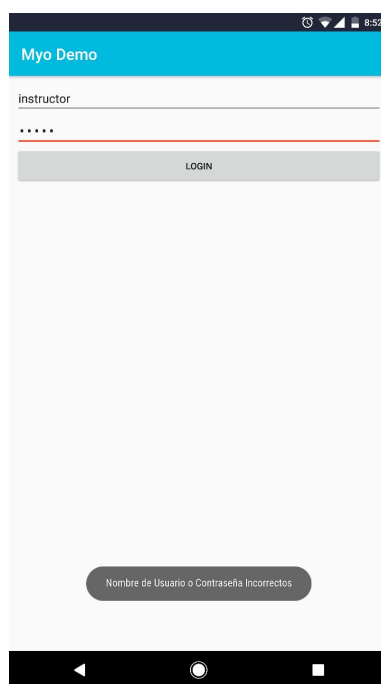


Figura 22 Login con credenciales erróneas. Elaboración propia

En el caso de que el usuario ingrese erróneamente las credenciales, la aplicación nos mostrará un error como se visualiza en la imagen. Si el ingreso es correcto, el sistema nos presentará las opciones de acuerdo al perfil que haya ingresado.

Una vez dentro de la aplicación, nos desplegará ya sea nuestra lista de movimientos realizados, o el listado de movimientos que podemos replicar, además en la parte inferior se tendrá un mensaje de alerta, en el que se menciona que se debe conectar al dispositivo. Es indispensable que el mismo se enlace con la aplicación para que pueda enviar los datos de sus sensores. Este mensaje puede ser observado en la Figura 23.

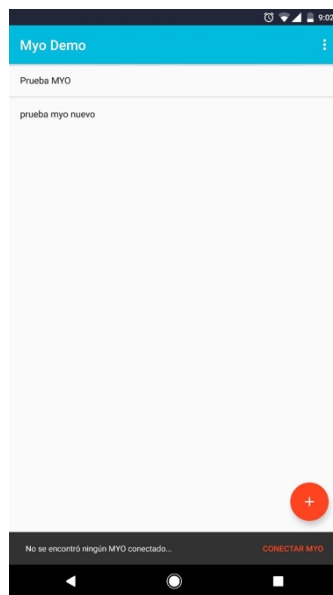


Figura 23 Página principal. Elaboración propia

Si se escoge la opción conectar Myo, se desplegará una pantalla que es propia de la interfaz de programación del dispositivo, en donde se debe escanear los dispositivos cercanos y escoger el brazalete Myo que se esté utilizando tal y como muestra la Figura 24.

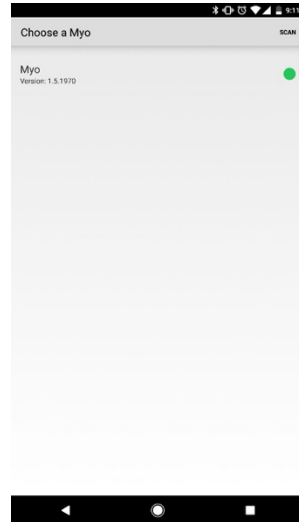


Figura 24 Interfaz de conexión del brazalete Myo. Elaboración propia

Una vez que se conecta el dispositivo, se debe ir a la ventana anterior para seleccionar el proceso que se quiera realizar. Si se ingresa con un usuario instructor, la aplicación permite generar un nuevo movimiento, para lo cual se debe escoger la opción con el signo “+” en la ventana principal. Al escoger esta opción nos desplegará una nueva ventana en la que se debe colocar la información del movimiento que se va a realizar, en este caso será el nombre del movimiento y una descripción en referencia al mismo, como se muestra en la Figura 25.

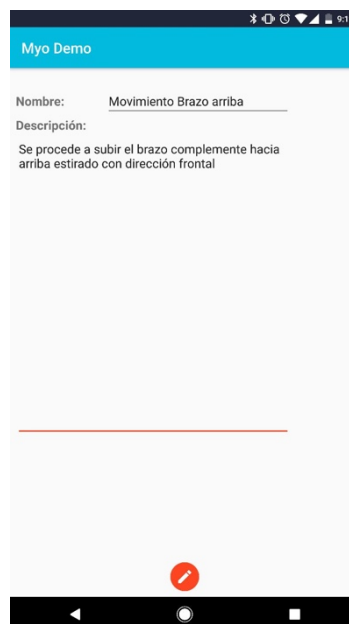


Figura 25 Ingreso de datos generales del movimiento. Elaboración propia

Una vez que se llene la información, se puede iniciar la grabación de los movimientos, antes de iniciar se desplegará una pantalla con la descripción del movimiento y en la parte inferior el botón que dará inicio a la grabación, tal y como se muestra en la Figura 26.

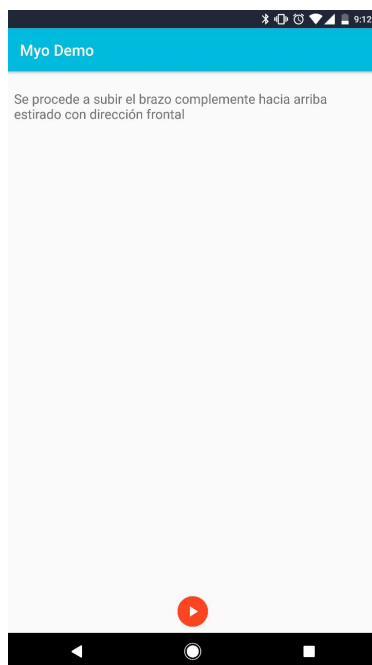


Figura 26 Pantalla de inicio de grabación de movimiento. Elaboración propia

Una vez seleccionado el botón de inicio, y si el usuario es un instructor, la aplicación solicitará la grabación del mismo movimiento 3 veces. Antes de realizar cada grabación se le indica al usuario en que tiempo iniciará la misma, esto para darle unos segundos para que se prepare para realizar el movimiento (5 segundos), Tal como se muestra en la Figura 26

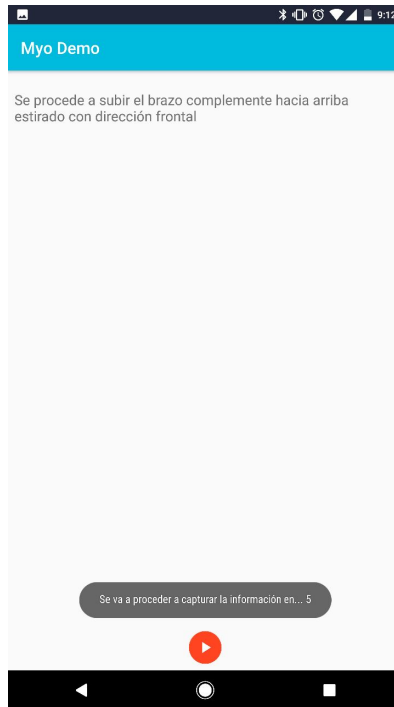


Figura 27 Tiempo de gracia para el inicio de la grabación. Elaboración propia

Todas estas imágenes son ejemplos de cómo fue desarrollada la interfaz de la aplicación, para que sea intuitiva y permita un fácil acceso a cada una de las opciones que posee la aplicación móvil desarrollada.

4.3.2 Conexión con el Brazaletes

Para realizar la conexión con el brazaletes y la aplicación, como se mencionó anteriormente, se utilizó el SDK que permite la interacción con este dispositivo. Aquí se utiliza una función especial propia del SDK que es `onCreate`. Esta función permite que se cree el proceso principal con su respectivo `Listener`, para que exista un servicio que esté siempre alerta a los cambios realizados en el brazaletes.

Por cada uno de los sensores se tiene una función que recibe datos siempre y cuando el brazaletes emita los mismos. Aquí nos encontramos con 3 funciones principales que son: `onOrientationData`, `onAccelerometerData` y `onGyroscopeData`. Cada una de las funciones recibe información siempre y cuando exista una variación en los valores que posee, y por cada

variación que se va presentando, se genera un objeto de tipo Sensor, mismo que es el encargado de almacenar los datos enviados de los 3 sensores.

4.4 Resultados

Una vez que se implementó el algoritmo de comparación de movimientos, con su respectivo margen de aceptación, se empezaron a realizar pruebas en relación a lo planteado en el anterior capítulo. Las pruebas fueron realizadas con movimientos similares a los que se hicieron en la etapa de análisis, se mantuvo las mismas condiciones en las que se realiza varias veces el mismo movimiento con las misma persona varias veces, en una misma posición y en una diferente, de igual forma se replicó el mismo movimiento con otra persona y en distintas posiciones. Para el resultado final se utilizó el algoritmo construido en el webservice de acuerdo al análisis del algoritmo DTW y el margen de tolerancia que se construyó y utilizando las matrices generadas por la recepción de los valores obtenidos a través de la utilización del acelerómetro del dispositivo Myo. Los resultados se muestran en la Tabla 9.

		Resultados
Adelante Arriba Adelante	Misma Persona	100%
		100%
		100%
		100%
		100%
	Misma Persona Diferente Posición	100%
		100%
		100%
	Otra persona	66%
		83.19%

		91.71%
	Otra persona Diferente Posición	90.61%
		84.88%
		88.26%
Bate Normal	Misma Persona	10.71%
		0%
		2.73%
	Otra persona	0%
		0%
		0%
Bate 45 grados	Misma Persona	14.02%
		0%
		15.05%
	Otra persona	0%
		0%
		0%

Tabla 9 Resultados de movimientos realizados

Se puede ver en esta tabla que en realidad los resultados que se obtienen tienen valores correctos en valores similares, sin embargo las variaciones que existen dependen mucho de la precisión con la que se realiza tanto el movimiento inicial, como el movimiento final. Si existe demasiada diferencia entre los intentos del movimiento inicial, representaría una variación completa en el nivel de tolerancia, pues en el cálculo que se desarrolló es de vital importancia este resultado. Para controlar estas diferencias entre los movimientos se realizó el cálculo del margen de tolerancia, pues los movimientos no van a ser exactamente iguales a pesar de que se intente replicar el mismo con exactitud, siempre habrá una pequeña diferencia entre un movimiento y otro.

Una manera de mejorar estos resultados sería realizar muchos más movimientos por parte del entrenador y no solo los planteados en este documento, así se podría sacar un mejor promedio en cuanto a los resultados obtenidos por el algoritmo DTW. Por ejemplo en la Tabla se muestra un resultado del 66% de similitud entre un movimiento realizado por otra persona. Para verificar que esta información es correcta y que en este movimiento existe una diferencia sobre el movimiento original, se realizó un gráfico de dispersión entre los dos movimientos, el mismo se encuentra en la Figura 27

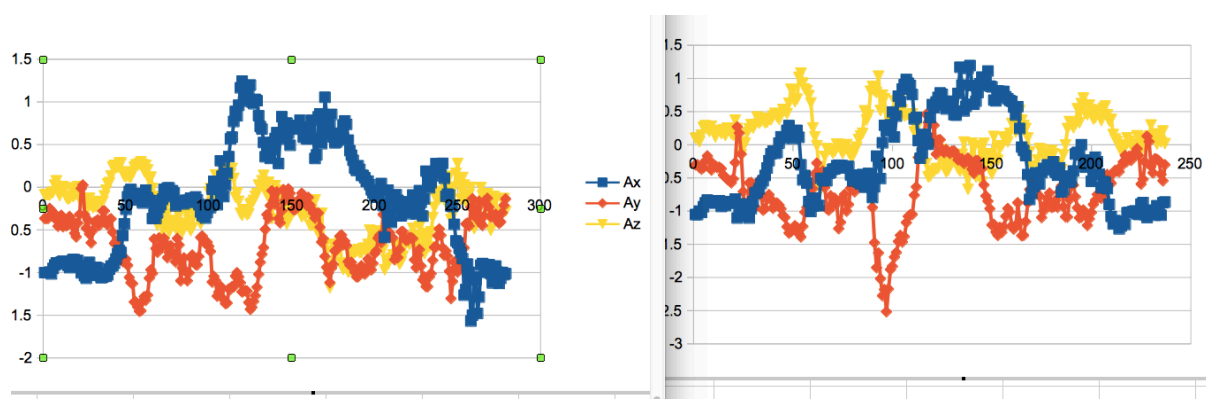


Figura 28 Comparación gráfica de movimientos con cierto grado de similaridad. Elaboración propia

En el lado izquierdo tenemos el movimiento correcto, mientras que en el lado derecho tenemos el movimiento incorrecto, como podemos ver en el eje Ay (tomate) existe una variación considerable en relación al primer movimiento, este punto de inflexión en la curva puede ser el delimitante que marcó que el movimiento no sea tan semejante como los otros realizados en la comparación.

5. Conclusiones y Recomendaciones

En base a los resultados obtenidos de la investigación se pudo llegar a la conclusión de los objetivos principales de la investigación.

5.1 Conclusiones

La aplicación desarrollada permite guardar los movimientos realizados por un entrenador, de manera que un deportista pueda utilizar después estas grabaciones para intentar simular los movimientos de su entrenador. Cada uno de estos intentos tiene como resultado un valor que nos indica el porcentaje de similitud que el deportista alcanzó sobre el movimiento base seleccionado, ayudando así al deportista a llevar un control sobre el entrenamiento de los movimientos que ha realizado

De acuerdo al análisis realizado de los algoritmos, se pudo notar que cada algoritmo tiene características propias que permiten realizar un mejor análisis de acuerdo a los datos que se obtiene, por ejemplo el algoritmo LCSS funciona correctamente cuando se tienen valores completamente similares, eso se pudo notar en el análisis de este algoritmo con el sensor del giroscopio en donde si se pueden encontrar valores completamente exactos entre los resultados obtenidos.

La interfaz del usuario facilita el acceso a las diversas opciones que posee la aplicación, para que el usuario pueda navegar sobre la misma de una manera intuitiva, además que separa por completo las opciones que posee cada uno de los perfiles, permitiendo así una mayor facilidad de uso

5.2 Recomendaciones

Una de las problemas para el análisis de los algoritmos descritos fue el tiempo de respuesta del servidor, esto debido a que R utilizaba en cada petición del servidor uno de los core del procesador, si mejoramos la interfaz gráfica con una tarjeta Nvidia existe una librería que permite una mejor distribución de los procesos en diferentes núcleos , por lo que se recomendaría realizar pruebas en un servidor con mejores características gráficas.

Bibliografía

Vera, P.(1994). La biomecánica deportiva, *Monografía de la Serie Deportes*, Málaga, pp. 361-369.

We Are social. (2017). Digital In 2017 Global Overview. Recuperado de: <https://www.slideshare.net/wearesocialsg/digital-in-2017-global-overview?ref=https://thenextweb.com/insights/2017/01/24/digital-trends-2017-report-internet/>

Stratista. (2017). Number of apps available in leading app stores as of march 2017. Recuperado de: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

Stratista. (2017). Number of available applications in the Google Play Store from December 2009 to June 2017. Recuperado de: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

Elasri, L (26/06/2014). Así funcionan las tripas de tu móvil: el acelerómetro, un sensor que te puede salvar la vida. Recuperado de: http://www.eldiario.es/hojaderouter/tecnologia/acelerometro-funciones-giroscopio-GPS-interior-magnetometro-sensor-sensor-de-humedad-sensor-de-temperatura-telefono-movil_0_275772515.html

Izquierdo, M. (2008). Biomecánica y bases neuromusculares de la actividad física y el deporte. Madrid: Médica Panamericana S.A.

Cunha, J., Cunha, B., Pereira, A., Xavier, W., Ferreira, N., & Meireles, L. (2010). VitalJacket R : A wearable wireless vital signs monitor for patients' mobility in cardiology and sports. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), Proc. 4th Int. ICST Conf. Pervasive Comput. Technol. Healthc*, pp. 12.

Scarborough, J. B. (1958). *The Gyroscope: Theory and Application*. Interscience Pub.

Camacho, D., Erazo, M., Mera, E., & Velasco, A. Control de un Brazo Robótico LYNX AL5D Empleando Electromiografía a Través de la Tecnología MYO ARMBAND.

Corti, S. (2016). Impacto de dispositivos" wearables" en el monitoreo de la salud.

Alonso, A. B., Artime, I. F., Rodríguez, M. Á., & Baniello, R. G. (2011). Dispositivos móviles. EPSIG Ing. Telecomunicación Universidad de Oviedo.

Hohensee, B. (2014). Introducción A Android Studio. Babelcube Inc.

Suehring, S. (2002). MySQL Bible, New York: Wiley Publishing, Inc.

Dessau, R. B., & Pipper, C. B. (2008). "R"--project for statistical computing. *Ugeskrift for laeger*, pp. 328-330.

Chacon, S., & Straub, B. (2014). Pro git. Apress.

Magdy, N., Sakr, M., Mostafa, T., & El-Bahnasy, K. (2015). Review on trajectory similarity measures. *Seventh International Conference*, December, pp. 613-619.

Madisetti, V. (2009). Video, Speech, and Audio Signal Processing and Associated Standards. New York : CRC Press.

Ratanamahatana, C. A., & Keogh, E. (2005). Three myths about dynamic time warping data mining. *Proceedings of the 2005 SIAM International Conference on Data Mining* , pp. 506-510.

Aach, J. & Church, G. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*(17), pp. 495-508.

Zhu, Y. & Shasha, D. (2003). Warping Indexes with Enve-lope Transforms for Query by Humming. *SIGMOD*, pp. 181-192.

Ratanamahatana, C.A. & Keogh, E. (2004). Making Time-series Classification More Accurate Using Learned Constraints. *SDM International conference*, pp. 11-22.

- Yi, B., Jagadish, H., & Faloutsos, C.** (1998). Efficient retrieval of similar time sequences under time warping. *14th International Conference on Data Engineering Proceedings*, pp. 201-208.
- Berndt, D. & Clifford, J.** (1994). Using Dynamic Time Warping to Find Patterns, *Workshop on Knowledge Discovery in Databases*, New York, April, pp. 360-370.
- Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., & Theodoridis, Y.** (2007). Similarity search in trajectory databases. *Temporal Representation and Reasoning, 14th International Symposium*, pp. 129-140.
- Lei Chen, M. & Oria, V.** (2005). Robust and Fast Similarity Search for Moving Object Trajectories, [*SIGMOD '05 Proceedings of the 2005 ACM SIGMOD international conference on Management of data*](#), New York, Junio, pp. 491-502.
- Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E.** (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, pp. 1542-1552.
- Alt, H. & Godau, M.** (1995). Computing the Frechet distance between two poligonal curves, *International Journal of Computational Geometry & Applications*, Vol 5, pp. 75-91.
- Vlachos, M., Gunopulos, D., & Kollios, G.** (2002). Robust similarity measures for mobile object trajectories. *13th International Workshop on Database and Expert Systems Applications*, pp. 721-726.
- Vlachos, H., Kollios G & Gunopulos, D.** (2002). Discovering Similar Multidimensional Trajectories, *Data Engineering, 2002. Proceedings. 18th International Conference.*, pp. 673-684.
- Bozkaya, T., Yazdani N. & Ozsoyoglu M** (1997). . Matching and Indexing Sequences of Different Lengths. ,*Proc.of the CIKM*, Las Vegas.
- Keogh, E., Chakrabarti K., Mehrotra S. & Pazzani M** (2001). . Matching and Indexing Sequences of Different Lengths. , pp- 151–162.

Anexos

Anexo A. Análisis de movimientos con algoritmos DTW y Edit Distance

Movimiento	Tipo	Aceleracion		Giroscopio		Orientación	
		DTW	Edit Distance	DTW	Edit Distance	DTW	Edit Distance
Adelante Arriba Adelante	Misma Persona	99.29294	210	23883.33	233	43.42613	215
		79.7043	33	18410.02	35	35.73213	47
		80.67037	47	20827.88	279	37.45966	145
		89.06232	138	25271.14	239	43.39935	201
		67.85651	105	16884.28	72	33.59446	121
	Misma Persona Diferente Posición	80.98658	51	21227.36	214	208.7716	57
		89.06696	111	17837.4	249	207.7234	135
		85.26736	151	17361.05	89	211.0388	74
	Otra persona	152.0864	109	23049.76	146	413.2772	42
		132.5774	178	24453.68	221	390.6281	78
		122.9064	139	26267.83	200	393.8804	89
	Otra persona Diferente Posición	124.1609	92	23663.18	249	521.2768	77
		126.8174	150	24908.07	260	513.6298	91
		149.05	74	27451.97	114	509.7446	47
Bate Normal	Misma Persona	221.9777	151	35984.21	193	389.0296	72

		214.8365	122	34932.39	243	374.9361	104
		234.7064	186	39214.54	187	355.4441	62
	Otra persona	277.4121	117	40083.8	279	489.8603	76
		270.1237	140	37706.12	277	476.271	105
Bate 45 grados	Misma Persona	244.9816	139	38929.65	129	470.5839	129
		211.0788	95	36494.3	95	251.8431	95
		228.3938	103	36823.86	269	243.4925	101
	Otra persona	209.9117	128	36181.82	262	237.5171	128
		270.1865	110	478.7376	103	42302.58	235
		278.5199	100	39340.77	125	487.2545	100
		274.7553	109	38988.62	271	485.3065	109

Anexo B. Resultados obtenidos con la utilización de LCSS y Frechet en 30 y 50 Movimientos

Movimiento	Tipo	Aceleración		Giroscopio				Orientación			
		Frechet		LCSS		Frechet		LCSS		Frechet	
		30	50	30	50	30	50	30	50	30	50
Adelante Arriba Adelante	Misma Persona	0.2927011	0.815296	10	12	113.9487	171.8708	30	50	0.1531569	0.3970479
		0.3271698	0.6326642	16	18	175.8179	175.8179	30	50	0.2495381	0.4317499
		0.5797557	0.6695048	10	12	170.8605	216.2799	30	50	0.3634479	0.5395499
		0.2933298	0.415033	11	12	147.7934	147.7089	30	50	0.1237708	0.1287612
		0.5706813	0.663161	13	13	193.092	215.7987	30	50	0.4173726	0.487174
	Misma Persona Diferente Posición	0.892544	0.7133788	9	10	185.1568	196.3536	30	50	0.8372589	0.8372589
		0.3625935	0.6409482	8	10	122.8541	172.8767	30	50	1.011997	1.011997
		0.520724	0.766617	11	15	93.42791	208.7747	30	50	0.8716496	0.8716496
	Otra persona	0.6881142	0.9467941	8	13	81.14139	137.3847	30	50	1.573986	1.573986
		1.630185	0.8607603	7	11	164.4366	185.3429	30	50	1.496573	1.528477
		1.308353	0.8694481	8	9	193.3402	180.9634	30	50	1.566798	1.564661
	Otra persona Diferente Posición	1.361481	1.062053	5	8	223.854	209.0709	30	50	1.944888	1.944854
		1.546416	0.7952545	6	6	211.607	175.331	30	50	1.902947	1.917963
1.202958		1.10199	3	5	200.7557	214.5189	30	50	1.898636	1.916844	
Bate Normal	Misma Persona	0.4341716	0.8810095	14	14	169.255	149.8385	30	50	1.601891	1.601891
		0.6626212	0.9762198	13	14	120.0947	246.4462	30	50	1.610068	1.610068
		0.9736206	1.280649	10	10	229.0032	200.0459	30	50	1.61157	1.61157
	Otra persona	0.6896335	1.062917	5	5	182.6246	181.7665	30	50	1.971354	1.968039
		1.2011	1.283211	6	8	203.078	193.6486	30	50	1.966947	1.966794
		1.905881	1.799293	4	4	327.2672	204.7246	30	50	1.960115	1.960115
Bate 45 grados	Misma Persona	0.4364846	0.9223018	10	10	186.2082	208.4453	30	50	1.083231	1.083231
		0.7559804	1.27623	8	8	202.6312	202.6165	30	50	1.086454	1.105792
		0.9779294	1.409646	7	7	324.7636	179.1585	30	50	1.089008	1.240308
	Otra persona	1.600875	1.788796	5	5	362.4493	303.9999	30	50	1.990222	1.990222
		1.123144	1.321016	5	5	201.1726	184.6915	30	50	1.99138	1.99138

		1.813432	1.641312	7	9	310.6566	178.8077	30	50	1.991306	1.991306
--	--	----------	----------	---	---	----------	----------	----	----	----------	----------

Anexo C. Resultados obtenidos con la utilización de LCSS y Frechet en 100 y 120 movimientos

		Aceleracion				Giroscopio				Orientación			
		LCSS		Frechet		LCSS		Frechet		LCSS		Frechet	
Movimiento	Tipo	100	120	100	120	100	120	100	120	100	120	100	120

Adelante Arriba Adelante	Misma Persona	100	120	0.9246878	1.013731	23	26	139.0744	196.7189	100	120	0.295812	0.549430
		100	120	1.481736	1.287934	32	37	199.02280	175.81790	100	120	0.419614	0.448100
		100	120	1.206484	1.273027	20	24	175.73550	175.73550	100	120	0.567563	0.476157
		100	120	1.481588	1.604172	21	23	159.12370	201.16020	100	120	0.494948	0.502014
		100	120	1.028019	1.247364	28	35	224.38700	159.77720	100	120	0.564319	0.454269
	Misma Persona Diferente Posición	100	120	1.356629	1.452403	30	34	252.52890	221.61480	100	120	0.856540	1.146825
		100	120	1.325726	1.344531	23	25	273.40540	207.24050	100	120	1.034545	1.054334
		100	120	1.557664	1.506678	25	26	238.86540	184.56810	100	120	1.016836	1.053083
	Otra persona	100	120	1.615458	1.533692	18	18	229.14280	303.80370	100	120	1.573986	1.585959
		100	120	1.491124	1.491124	19	19	185.33080	243.98830	100	120	1.495857	1.495857
		100	120	1.497072	1.551308	21	22	201.95560	281.43510	100	120	1.564613	1.564595
	Otra persona Diferente Posición	100	120	1.488302	1.474922	18	19	200.07520	336.09940	100	120	1.944846	1.944846
		100	120	1.462598	1.462594	15	16	175.33100	225.89680	100	120	1.902947	1.946755
100		120	1.494498	1.555201	13	15	329.96890	242.01520	100	120	1.898636	1.960055	
Bate Normal	Misma Persona	100	120	1.730212	1.542738	24	27	349.22840	253.00820	100	120	1.601891	1.601891
		100	120	1.590338	1.590338	19	20	252.27350	252.27350	100	120	1.610068	1.610068
		100	120	1.593107	1.593107	13	13	236.70480	236.70480	100	120	1.611570	1.639222
	Otra persona	100	120	1.695427	2.129445	13	13	305.63940	250.84610	100	120	1.968039	1.968039
		100	120	1.763805	1.926641	15	16	261.95230	272.94850	100	120	1.966780	1.966780
		100	120	1.600373	1.749130	9	11	258.01030	473.44260	100	120	1.960115	1.960115
Bate 45 grados	Misma Persona	100	120	1.471485	1.471485	17	17	279.08480	279.08480	100	120	1.181471	1.114964
		100	120	1.434880	1.442983	14	14	291.99450	291.99450	100	120	1.205243	1.172250
		100	120	1.617912	1.617765	11	13	331.22320	336.94140	100	120	1.140640	1.177659
	Otra persona	100	120	1.970829	1.970829	16	16	301.13110	301.13110	100	120	1.990222	1.990222
		100	120	1.651964	1.687046	16	16	228.75550	262.76930	100	120	1.991380	1.991380
		100	120	1.753566	1.753566	15	15	263.88620	263.88620	100	120	1.991306	1.991306

Anexo D. Resultados obtenidos con la utilización de puntos máximos y mínimos en todos los algoritmos

Movimiento	Tipo	Aceleracion X				Aceleracion Y				Aceleracion Z			
		DTW	Edit Distance	LCSS	Frechet	DTW	Edit Distance	LCSS	Frechet	DTW	Edit Distance	LCSS	Frechet

Adelante Arriba Adelante	Misma Persona	0.54102	2	3	0.24854	0.4795	1	3	0.2041	1.34912	6	3	0.41162
		0.36719	2	3	0.09424	0.29052	0	3	0.15918	3.63574	11	3	0.97754
	Misma Persona Diferente Posición	0.37695	2	3	0.09350 5	0.80176	1	3	0.49219	0.76709	2	3	0.39258
		0.37695	2	3	0.09350 5	2.02783	1	3	0.73535	0.48486	3	3	0.17139
	Otra persona	0.53173	3	2	0.22802	0.68897	2	2	0.41407	4.32715	9	2	1.00342
		1.77979	0	5	1.06884	8.12058	11	5	1.39013	0.01074	0	5	0.01074
	Otra persona Diferente Posición	0.52199	3	2	0.17969	0.79834	2	2	0.52344	0.02978	0	2	0.02978
		1.88575	0	5	0.93213	0.75	2	5	0.4751	2.67041	5	5	0.93945
Bate Normal	Misma Persona	0.28272	2	3	0.09350 5	0.82519	2	3	0.32861	0.17676	0	3	0.17676
		0.34863	2	3	0.09350 5	0.3457	0	3	0.17725	0.1626	2	3	0.10547