



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Tecnologías de la Información

TRABAJO FIN DE GRADO

I-Racket: sistema para la mejora de técnica en deportes
de raqueta.

Álvaro Gómez De La Torre García del Rincón

Diciembre, 2019



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Tecnologías y Sistemas de la Información

Tecnologías de la Información

TRABAJO FIN DE GRADO

**I-Racket: sistema para la mejora de técnica en
deportes de raqueta.**

Autor: Álvaro Gómez De La Torre García del Rincón

Director: Javier Alonso Albusac Jiménez

Diciembre, 2019

Plantilla TFG

© Álvaro Gómez De La Torre García del Rincón, 2019

Este documento se distribuye con licencia Creative Commons Atribución Compartir Igual 4.0. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario: _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

En los últimos años el crecimiento y desarrollo de la tecnología en el deporte ha permitido desarrollar nuevas técnicas para monitorizar la actividad física en cualquier deporte. En los deportes de raqueta como el tenis existen diferentes dispositivos capaces de registrar datos relativos a los diferentes golpes que realizan los deportistas en sus entrenamientos como la velocidad empleada para ejecutar el golpe. Sin embargo, en el pádel aún no se ha encontrado ningún dispositivo mediante el cual se puedan registrar datos relativos al golpeo que realiza un deportista.

El presente Trabajo Fin de Grado (TFG) surge como una solución a esta problemática y tiene la intención de cubrir el vacío que existe en el mercado actual de herramientas para realizar una monitorización de golpes en el deporte de pádel. En este proyecto se plantea la construcción de un sistema formado por un dispositivo físico y un sistema *web* que permita monitorizar la ejecución de un golpeo de un usuario en un entrenamiento. El dispositivo físico será un prototipo *hardware* equipado con varios sensores que captan información acerca del movimiento que realiza el deportista a la hora de golpear a la pelota. Esta información es almacenada en un sistema *web* donde se mostrará al deportista el conjunto de estadísticas recogidas durante el entrenamiento, cuyos datos estarán centrados principalmente en el golpeo de la pelota. Dicho sistema *web* contará con técnicas avanzadas de visualización para que el deportista pueda interpretar fácilmente los datos recogidos. Además, el deportista podrá observar su evolución en cada tipo de golpeo y depurar su técnica para obtener un mejor rendimiento.

Por último, se diseñará una interfaz entre el deportista y el prototipo *hardware* buscando la mayor simplicidad posible en la interacción entre ambos. La interfaz estará basada principalmente en lenguaje gestual con el fin de reducir al mínimo la cantidad de elementos físicos incorporados en el dispositivo de medición.

Abstract

(... english version of the abstract ...)

Versión del resumen en inglés. En los trabajos cuyo idioma principal sea el inglés, el orden de Resumen y Abstract se invertirá.

AGRADECIMIENTOS

En primer lugar a mis padres Pedro y Clemen, por su gran esfuerzo realizado para ayudarme a labrar mi propio futuro y por su apoyo incondicional. A mi hermano J. Ramón por estar siempre a mi lado y ayudarme en los momentos más difíciles.

A María, por ser un apoyo fundamental y sacarme una sonrisa en los momentos más complicados.

Por último y no menos importante, a mi director de proyecto Javier Alonso Albusac Jiménez, por su implicación a lo largo del presente proyecto.

Álvaro Gómez De La Torre García del Rincón

ÍNDICE GENERAL

Índice de figuras	xv
Índice de tablas	xix
1. Introducción	1
1.1. Aparatos de monitorización en deportes de raqueta	3
1.2. Estructura del documento	5
2. Objetivos	7
2.1. Objetivo General	7
2.2. Objetivos específicos	7
3. Metodología	11
3.1. Arquitectura	11
3.1.1. Visión general de la arquitectura	11
3.1.2. Detalle de la arquitectura del sistema	13
3.2. Prototipo <i>hardware</i> para la captación de datos de entrenamiento	15
3.2.1. Diseño del dispositivo <i>hardware</i>	15
3.3. Sistema <i>web</i>	28
3.3.1. Arquitectura sistema <i>web</i>	28
3.3.2. Capa de persistencia: Almacenamiento de datos	45
3.4. Organización del proyecto en paquetes de trabajo	53
3.4.1. PT1. Diseño y desarrollo del dispositivo <i>hardware</i>	54
3.4.2. PT2. Diseño de interfaz entre el dispositivo de medición y el deportista	55
3.4.3. PT3. Diseño y desarrollo del sistema <i>web</i>	56
3.5. Metodología de trabajo	57
3.5.1. Iteraciones	58
3.6. Características <i>hardware</i> y <i>software</i> del desarrollo	62
3.6.1. Medios <i>hardware</i>	63
3.6.2. Medios <i>software</i>	63

4. Resultados	65
4.1. Desarrollo de un prototipo <i>hardware</i> equipado con varios sensores	65
4.2. Diseño y desarrollo de una interfaz de comunicación entre el dispositivo de medición y el deportista	69
4.3. Diseño de un sistema web para la visualización de la información recogida por el prototipo	73
4.3.1. Visualización de entrenamientos	74
4.3.2. Representación de golpesos	75
4.3.3. Comparación de golpesos y visualización de videotutoriales	77
4.3.4. Visualización de objetivos y puntuación	79
4.4. Competencias específicas de intensificación trabajadas durante este proyecto . .	81
5. Conclusiones	83
5.1. Trabajo futuro	85
A. Antecedentes	87
A.1. Microcontroladores	87
A.1.1. Raspberry Pi	87
A.1.2. Arduino	88
A.2. Desarrollo web	89
A.2.1. Lenguajes de desarrollo web	89
A.3. Gestor de base de datos	90
A.3.1. Sistemas gestores de bases de datos	91
A.4. Dispositivos de monitorización en deportes de raqueta	94
A.4.1. Smart tennis sensor	94
A.4.2. Raquetas inteligentes	95
A.4.3. Babolat Starter	95
A.4.4. Sensor Head Zepp Tennis	96
A.4.5. Babolat PIQ	97
A.4.6. Zepp Tennis	97
A.4.7. Babolat Play	97
A.5. Comparación de sistemas de monitorización en deportes de raqueta	99
Bibliografía	101

ÍNDICE DE FIGURAS

1.1. Smart tennis sensor.	4
1.2. Zepp tennis sensor.	4
1.3. Sensor Head zepp.	4
1.4. Babolat starter.	5
1.5. Babolat PIQ.	5
3.1. Esquema general del sistema propuesto.	12
3.2. Flujo de trabajo.	13
3.3. Arquitectura general.	14
3.4. Esquema conexión.	16
3.5. MPU6050.	17
3.6. Conexión bus I2C.	20
3.7. Módulo Micro SD.	22
3.8. Esquema de conexión módulo micro SD.	23
3.9. Posición golpe de derecha.	24
3.10. Posición golpe de globo.	25
3.11. Posición golpe de volea.	25
3.12. Posición de reposo	25
3.13. Diseño correa de fijación.	27
3.14. Correa de cuero adaptada al prototipo <i>hardware</i>	28
3.15. Esquema sistema <i>web</i>	29
3.16. Ejemplo de archivo <i>csv</i> generado por el prototipo <i>hardware</i>	30
3.17. Esquema para la importación de datos desde el prototipo <i>hardware</i> hacia el sistema <i>web</i> . Los datos registrados por el prototipo, se guardan en una tarjeta micro SD, el usuario carga estos datos en el sistema <i>web</i> (en la tabla auxiliar) y el procedimiento descrito anteriormente se encarga de distribuir la información.	31
3.18. Flujo función de autenticación.	34
3.19. Comparativa de la aceleración en el eje X de dos entrenamientos de globo.	36
3.20. Gráfica aceleración en el eje X de un golpe de derecha.	37

3.21. Gráfica aceleración en el eje Y de un golpe de derecha.	37
3.22. Gráfica aceleración en el eje Z de un golpe de derecha.	38
3.23. Gráfica ángulo en el eje X de un golpe de globo.	38
3.24. Gráfica ángulo en el eje Y de un golpe de globo.	38
3.25. Gráfica ángulo en el eje Z de un golpe de globo.	39
3.26. Comparación de vídeos entre usuario y profesional	40
3.27. Comparativa aceleración en el eje Y de un golpe de derecha.	40
3.28. Comparativa aceleración en el eje X de un golpe de derecha.	41
3.29. Videotutorial ofrecido a los usuario de un golpeo de globo.	42
3.30. Ejemplo de calendario de objetivos.	43
3.31. Ejemplo de listado de objetivos.	44
3.32. Secuencia de creación de objetivos.	44
3.33. Análisis de puntuación total que se puede obtener y de la puntuación restante de un usuario.	45
3.34. Servicios <i>REST</i> Oracle.	47
3.35. Comunicación <i>ORACLE APEX</i> con servicios <i>REST</i>	48
3.36. Modelo de datos relacional parte 1.	50
3.37. Modelo de datos relacional parte 2.	51
3.38. Modelo de datos relacional.	52
3.39. Diagrama de Gantt	54
3.40. Tablero <i>Trello</i>	58
3.41. Repositorio <i>git</i> del proyecto usado en <i>Github</i>	58
4.1. Datos de los sensores equipados en el prototipo <i>hardware</i>	65
4.2. Archivo guardado en modulo de tarjeta micro SD.	66
4.3. Ángulos empleados para realizar el test del eje Y.	68
4.4. Ángulos empleados para realizar el test del eje X.	69
4.5. Posiciones que debe adoptar el usuario para dar órdenes al prototipo <i>hardware</i> a través de la interfaz lógica desarrollada.	70
4.6. Ejemplo de cambio de movimiento al adoptar las posiciones anteriores para dar órdenes al dispositivo.	70
4.7. Login en el sistema <i>web</i>	74
4.8. Calendario de entrenamientos.	75
4.9. Listado de entrenamientos.	75
4.10. Gráfica aceleración en el eje X de un golpe de derecha.	76
4.11. Gráfica aceleración en el eje Y de un golpe de derecha.	76
4.12. Gráfica aceleración en el eje Z de un golpe de derecha.	76

4.13. Gráfica ángulo en el eje X de un golpe de globo.	76
4.14. Gráfica ángulo en el eje Y de un golpe de globo.	76
4.15. Gráfica ángulo en el eje Z de un golpe de globo.	77
4.16. Comparativa aceleración en el eje Y de un golpe de derecha.	77
4.17. Comparativa aceleración en el eje X de un golpe de derecha.	78
4.18. Comparación de vídeos entre usuario y profesional	78
4.19. Videotutorial ofrecido a los usuario de un golpeo de globo.	78
4.20. Listado de objetivos.	79
4.21. Listado de retos a conseguir nivel básico.	80
4.22. Listado de retos a conseguir nivel intermedio.	80
4.23. Listado de retos a conseguir nivel experto.	80
A.1. Ejemplo de Raspberry Pi.	88
A.2. Ejemplo de Arduino.	89
A.3. Estructura lógica base de datos Oracle	93
A.4. Estructura lógica de almacenamiento	93
A.5. Arquitectura Oracle.	94
A.6. Smart Tennis Sensor.	95
A.7. Babolat Starter.	96
A.8. Head Zepp Tennis.	96
A.9. Babolat PIQ.	97
A.10. Zepp Tennis.	97
A.11. App Babolat.	98

ÍNDICE DE TABLAS

3.1. Valores digitales MPU6050.	19
3.2. Valores digitales que mide el módulo <i>MPU6050</i>	20
3.3. Valores escalados velocidad angular.	20
3.4. Dispositivos de almacenamiento.	21
3.5. Conexión Módulo Tarjeta micro SD.	22
3.6. Resumen rango de valores para elección de golpeo	26
3.7. Utilización de <i>leds</i> y acción que indican.	27
3.8. Tabla resumen de los tipos de golpeo representados en el calendario de entrenamientos.	35
3.9. Descripción resumida de la primera iteración	59
3.10. Descripción resumida de la segunda iteración	59
3.11. Descripción resumida de la tercera iteración	60
3.12. Descripción resumida de la cuarta iteración	60
3.13. Descripción resumida de la quinta iteración	60
3.14. Descripción resumida de la sexta iteración	61
3.15. Descripción resumida de la séptima iteración	61
3.16. Descripción resumida de la octava iteración	61
3.17. Descripción resumida de la novena iteración	62
3.18. Descripción resumida de la décima iteración	62
3.19. Descripción resumida de la undécima segunda iteración	62
4.1. Ejemplo de grabación de aceleración de golpeo.	66
4.2. Ejemplo de grabación de ángulos de golpeo.	66
4.3. Valores test 1. Ángulos en el eje Y	67
4.4. Valores test 2. Ángulos en el eje X	68
4.5. Test 1 para medir el acierto de elección de golpeo.	71
4.6. Resumen de mediciones test 1.	71
4.7. Test 2 para medir el acierto de elección de golpeo.	72
4.8. Resumen de mediciones test 2.	72

4.9. Test 3 para medir el acierto de elección de golpeo.	72
4.10. Resumen de mediciones test 3.	73
4.11. Resumen de mediciones totales realizadas en los tres test.	73
A.1. Resumen de sistemas de monitorización en deportes de raqueta.	99

CAPÍTULO 1

INTRODUCCIÓN

El **deporte** ha evolucionado tecnológicamente en los últimos años en gran medida. Antes no se usaba nada de tecnología. Bastaba con tener una pelota para jugar al fútbol, al tenis, al pádel; un coche o una moto para participar en deportes de motor o simplemente tu propio cuerpo para competir en atletismo. A día de hoy, sin embargo, no se encuentra ninguna disciplina que no incorpore, en mayor o menor medida, los avances tecnológicos. Podemos encontrar todo tipo de innovaciones como por ejemplo en **equipamiento** (resistencia en los materiales con los que se diseña la ropa deportiva), **arbitraje** (ayuda para los árbitros en su tarea), **seguridad** (la tecnología busca prevenir las lesiones innovando en los dispositivos para la práctica de cada deporte), **recuperación** (electroestimulación), **entrenamiento** (*wearables* para hacer seguimiento de cada entrenamiento) con el fin de ayudar al deportista en su vida deportiva, para así obtener una mejora del rendimiento o prevenir lesiones [4].

Este tipo de tecnología se ha ido aplicando con el tiempo en diferentes deportes y disciplinas [7]. Con ello los deportes han ido creciendo y se han ido modernizando con el paso del tiempo. En el **fútbol**, se ha instaurado el VAR (vídeo arbitraje). En la **fórmula 1**, el HALO (arco protector) para los pilotos. En el **tenis** el ojo de halcón. En **atletismo** foto finish. En **baloncesto** cámara superlenta [5]. Además de esto, gracias a las nuevas tecnologías se ha conseguido monitorizar todo tipo de datos y estadísticas que reflejan el rendimiento de los deportistas. En deportes como el fútbol o atletismo podemos recoger datos y estadísticas mediante relojes inteligentes, chips inteligentes o GPS. En los deportes de motor como fórmula 1 o motociclismo, se lleva a cabo un proceso de telemetría avanzado que permite recoger las estadísticas mediante dispositivos instalados en los vehículos. Gracias a esto, se puede hacer una valoración de los datos y así poder mejorar el rendimiento en los diferentes deportes.

Las tecnologías no solo han venido para quedarse en el ámbito profesional, sino que también son de uso extendido en el deporte amateur. En la actualidad existen dispositivos de bajo coste, asequibles para la mayoría de economías familiares, que monitorizan parte de la actividad del deportista. Incluso, los móviles actuales vienen equipados con un conjunto de sensores que permiten desarrollar esta actividad. Estos dispositivos se centran principalmente en la medición de distancias, velocidad alcanzada, recorrido (localización en un mapa) o el ritmo cardiaco. Además, esta amplia gama de dispositivos suele estar complementada con aplicaciones *software* dedicadas, o bien, portales web genéricos donde es posible visualizar los datos recogidos. Algunos ejemplos son Strava¹, Endomondo², o Runkeeper³, entre otras.

En los deportes de raqueta existen varios dispositivos capaces de registrar datos acerca del golpeo sobre la pelota. Algunos de estos dispositivos se basan en recopilar información acerca

¹www.strava.com

²www.endomondo.com

³www.runkeeper.com

de la aceleración del golpeo y el punto de impacto sobre la pelota. En algunos de estos sistemas existen carencias que hacen que el sistema no sea completo del todo. Algunos de estos sistemas no estudian la posición de la muñeca a la hora de ejecutar el golpeo de la pelota y esto es factor importante a la hora de golpear a la pelota. Efectuar el golpeo adoptando un buen ángulo se puede traducir en un buen golpeo. Estos sistemas no permiten la elección del tipo de golpeo que se está efectuando, por lo que solo registran valores globales de golpeos. Es importante diferenciar el tipo de golpeo que se está realizando, ya que por ejemplo un golpe de *drive* no se efectúa con la misma aceleración que un golpe de globo o una dejada. Tampoco existen sistemas en los cuales se pueda comparar los golpes que realiza un jugador con los de su entrenador o los de un jugador profesional. Esta funcionalidad permitiría al jugador obtener un mejor *feedback* acerca del golpeo realizado y saber que parte tiene que mejorar.

En un deporte de raqueta como el pádel, no se encuentran dispositivos de esta clase, en los que se cumplan los requisitos expuestos anteriormente. En el pádel, hay varios factores importantes a la hora de realizar el golpeo como son la **aceleración** y el **ángulo** de ejecución. En este deporte **no existen dispositivos que se centren principalmente en la parte del golpeo ni software que se encargue de visualizar los datos relativos a éstos**. Estos golpes se pueden clasificar principalmente en: *drive*, revés, volea, bandeja, remate y dejada. Cualquiera de estos golpes se caracteriza por la posición de la muñeca y el codo, el ángulo que forman, así como la aceleración que se le imparte al golpeo. Mediante estos atributos, un sistema informático podría distinguir entre cada uno de estos golpes. Un sistema capaz de obtener información relativa a estos parámetros y que los visualizara de forma adecuada, permitiría analizar al propio deportista la técnica para poder mejorarla y, en consecuencia, mejorar su rendimiento.

Existen también otra clase de requisitos para que este tipo de tecnología pueda llegar a buen puerto en los deportes de raqueta. Se debe tratar de dispositivos con un peso reducido para no entorpecer el golpeo de la pelota. Además, el deportista durante su entrenamiento lleva a cabo una actividad realmente dinámica en la que apenas hay tiempos muertos. La interfaz entre el deportista y el dispositivo debe ser lo más amigable e intuitiva posible. De tal forma que el deportista pueda transmitir sus preferencias rápidamente sin tener que detenerse apenas. En definitiva, la tecnología incorporada debe entorpecer mínimamente el entrenamiento.

Entre los deportes de raqueta, el **pádel** es un deporte que se lleva practicando desde 1980. Este deporte antes no era tan conocido. En España se empezó a ver más sobre el año 1990, cuando todavía solo lo practicaba una pequeña parte de la población. A partir del año 2000, dicho deporte empezó a conocerse más, lo que conllevaría al crecimiento de éste. En concreto, este deporte es uno de los que mayor crecimiento presenta, llegando a ser un deporte que se encuentra entre los diez deportes que más se practican en nuestro país [19]. El último estudio estadístico sobre el pádel revela que 4,2 millones de personas practicaban pádel en el año 2015 (3 millones más que en 2010), siendo 2,7 millones hombres y 1,2 mujeres [1]. Esto supone que entre 2015 y 2018 se ha triplicado la cifra de participantes, lo que conlleva que el número de licencias federadas muestra un crecimiento del 75 % con respecto a 2010 [6].

En este contexto nace este proyecto, como una herramienta para poder monitorizar durante un entrenamiento los diferentes golpes del jugador y posteriormente poder hacer un análisis de estos. Con ello se puede contribuir a una valoración del entrenamiento del jugador, así como una mejora de su técnica.

De acuerdo a esta gran demanda por el pádel y la problemática expuesta anteriormente, en este proyecto se pretende desarrollar un sistema web donde se mostrará al deportista el conjunto de estadísticas recogidas durante el entrenamiento, cuyos datos estarán centrados principalmente en el golpeo de pelota. Dicho sistema web contará con técnicas avanzadas de visualización para

que el deportista pueda interpretar fácilmente los datos recogidos. Además, el deportista podrá observar su evolución en cada tipo de golpeo, comparar su diferentes golpeos con otros usuarios y depurar su técnica para obtener un mejor rendimiento.

Para obtener los datos relativos a cada entrenamiento de un jugador, se utilizará un prototipo *hardware* equipado con varios sensores, que permita recoger la información necesaria del entorno. Este prototipo consistirá en un microcontrolador *Arduino Uno* [2], equipado con sensores y dispositivos adecuados para la monitorización de un entrenamiento. Los sensores que se utilizarán en este proyecto son:

- **Sensor Acelerómetro y Giroscopio MPU6050.** Este sensor nos ayudará a recoger los valores relativos a la aceleración de cada golpe y al ángulo con el que se golpea a la pelota.
- **Modulo tarjeta micro SD.** Este dispositivo nos ayudará a guardar toda la información que recojamos con el prototipo, para un posterior volcado de esta en el portal web.

Por último, se hará uso de una interfaz entre el deportista y los dispositivos de medición buscando la mayor simplicidad posible en la interacción entre ambos. El jugador podrá indicarle a la interfaz que tipo de golpeo va a realizar en función de la posición de su muñeca. Esto hará que se reduzca al mínimo la cantidad de elementos físicos que llevará incorporado el prototipo. Lo que conlleva que el prototipo permita al deportista de forma sencilla, indicarle el golpe que va a realizar de forma rápida y sencilla sin realizar excesivas pausas en su entrenamiento.

Con esta propuesta se pretende cubrir el vacío existente en dispositivos para la evaluación de entrenamientos, así como el vacío de sistemas en los cuales un deportista pueda hacer un análisis de sus golpeos, visualizar estadísticas de sus entrenamientos y comparar sus golpeos con otros jugadores profesionales para ir depurando y mejorando su técnica de golpeo.

1.1. APARATOS DE MONITORIZACIÓN EN DEPORTES DE RAQUETA

La tecnología es un fenómeno que está en constante evolución y es que la tecnología no se aplica solo a un determinado tema o área específica. La evolución de la tecnología ha irrumpido en todos los campos controlados por el ser humano para modificarlos e implementar en ellos las ventajas que pueden llegar a suponer los constantes avances que trae consigo. El deporte no escapa a su influjo y se nutre de dichos avances tecnológicos para perfeccionarse, tanto a nivel de competición, a nivel usuario y a nivel reglamentario.

Podemos desatacar que en deportes de raqueta como el tenis existen dispositivos para registrar datos acerca del golpeo que se ejecuta sobre la pelota. Sin embargo, en otro deporte de raqueta como es el pádel, no se encuentran dispositivos capaces de evaluar la ejecución del golpeo realizado sobre la pelota. Por tanto, existe un vacío en el deporte de pádel referente a dispositivos de monitorización del golpeo de la pelota.

Algunos ejemplos de dispositivos de monitorización:

Smart tennis sensor. Se trata de un sensor que se coloca en el mango de la raqueta. Este dispositivo pesa 8 gramos y es capaz de medir datos relativos al golpeo de la pelota como la velocidad de la bola, zona de impacto de la pelota, velocidad de la raqueta a la hora de realizar el golpeo y número de golpes realizados.



Figura 1.1: Smart tennis sensor.

Zepp tennis sensor. Se trata de otro tipo de sensor que se coloca en el mango de la raqueta. Este sensor es capaz de medir la potencia con la que se realiza el golpeo, el tipo de golpeo que realizas y el tiempo de entrenamiento que has realizado.



Figura 1.2: Zepp tennis sensor.

Sensor Head Zepp Tennis. Este sensor cuenta con cuatro modos de utilización: juego, entrenamiento, competición y servicio 3D. El modo de juego permite registrar los tipos de golpeo, la velocidad de golpeo y punto dulce de golpeo. El modo de entrenamiento permite acceder a una amplia gama de entrenamientos creados por *Head*. El modo de competición permite retar a amigos o rivales y comparar su rendimiento. El modo Servicio 3D permite analizar la técnica de saque. Este sensor se puede acoplar a la mayoría de tapas de empuñadura y posee un peso de solo 7 gramos.



Figura 1.3: Sensor Head zepp.

Babolat starter. Pulsera que lleva incorporada una serie de sensores que ayudan a registrar valores relativos a la ejecución del golpeo.



Figura 1.4: Babolat starter.

Babolat PIQ. Sensor que se coloca en el cordaje de la raqueta. Este sensor permite medir la velocidad en cada instante del golpeo, número de golpes y el punto dulce en cada golpeo.



Figura 1.5: Babolat PIQ.

1.2. ESTRUCTURA DEL DOCUMENTO

El presente proyecto se compone de los siguientes capítulos:

Capítulo 1. Introducción

Se trata del capítulo actual. En él se plantea la problemática existente que se pretende resolver con este proyecto.

Capítulo 2. Objetivos

En este capítulo se presentan los objetivos que se deben satisfacer en el desarrollo del proyecto, para alcanzar el éxito del mismo.

Capítulo 3. Metodología de trabajo

En este capítulo se muestra la metodología de desarrollo que se ha llevado a cabo durante este proyecto, además de las distintas herramientas usadas para el mismo. Se presentarán también las distintas fases para la elaboración de este proyecto.

Capítulo 4. Resultados

Se mostrarán los resultados obtenidos, fruto de las pruebas realizadas con el prototipo desarrollado.

Capítulo 5. Conclusiones Es este capítulo se exponen las principales conclusiones extraídas tras la realización de este TFG.

CAPÍTULO 2

OBJETIVOS

En este capítulo se va a presentar el objetivo principal que se pretende alcanzar con la realización de este trabajo. Este objetivo a su vez se desglosa en varios subobjetivos, que son necesarios satisfacer para lograr el objetivo final.

2.1. OBJETIVO GENERAL

De acuerdo a la problemática expuesta anteriormente, nace este proyecto. Cuando un jugador está realizando un golpeo, no es consciente de si ese golpeo está bien efectuado o no. Por ello, nace la necesidad de saber si el golpeo se está realizando con la técnica correcta. Para llevar a cabo esta tarea, se necesita un dispositivo para la monitorización y el almacenamiento de información sobre el golpeo que se está realizando. Esta información será volcada en un sistema web, donde el deportista podrá hacer un análisis de sus golpes y valorar si se han efectuado con la técnica adecuada. Además, el jugador tendrá la posibilidad de comparar sus golpes con los golpes de jugadores profesionales.

En relación a la demanda con el pádel y el vacío existente de dispositivos de monitorización que cubran esta necesidad, surge el objetivo principal de este proyecto.

El objetivo principal de este proyecto es la **monitorización y estudio del golpeo de pelota en deportes de raqueta para una mejora de la técnica de golpeo.**

Para llevar a cabo este objetivo, se hará uso de un **prototipo hardware** dotado con varios sensores que nos ayudará a recoger información sobre el golpeo de la pelota. Gracias a esta información, el usuario podrá hacer un análisis posterior de los datos en un **portal web**, de tal manera que pueda interpretarlos y tener una información global acerca de la técnica de su golpeo. Así, podrá evaluar qué debe mejorar de cara al próximo entrenamiento e ir mejorando su técnica.

2.2. OBJETIVOS ESPECÍFICOS

1. **Diseño y desarrollo de un prototipo hardware equipado con varios sensores para la obtención de datos sobre el entrenamiento del deportista.**

Para realizar el análisis de un entrenamiento, se hará uso de un prototipo *hardware* dotado con varios sensores. Este dispositivo lo llevará incorporado el propio deportista para poder captar todas las fases del golpeo de la pelota: armado, golpeo y finalización.

Este prototipo estará constituido por un microcontrolador y un conjunto de sensores y dispositivos de almacenamiento. El microcontrolador se encargará exclusivamente de

recoger y almacenar la información de los sensores para una posterior interpretación por parte del deportista.

Algunas de las funciones que realizará este prototipo serán:

- Captación de información acerca de la posición de muñeca durante el golpeo sobre la pelota.
- Captación de información acerca de la aceleración sobre la pelota durante el golpeo.
- Almacenamiento de la información para una posterior interpretación en un sistema web.

2. **Diseño de una interfaz entre el dispositivo de medición y el deportista, para facilitar la interacción entre ambos.**

Para que el deportista tenga una buena interacción con el prototipo y un fácil manejo, es necesario diseñar una interfaz para interactuar entre ambos. La interacción entre el deportista y el prototipo de medición empleado, tiene que ser lo mas amigable y ágil posible, de tal forma que el deportista no tenga que parar su actividad. Se diseñará una interfaz basada en el lenguaje gestual, reduciendo al máximo posible el número de componentes de interacción de entrada como botones o similares, con el fin de reducir también al máximo el peso de los elementos que el deportista tiene que incorporar en su cuerpo.

La interfaz permitirá al deportista elegir entre varios tipos de golpeos. Para este proyecto destacaremos los tres golpes principales en el pádel. Estos golpes son:

- *Drive* de derecha.
- Globo.
- Volea.

Una vez que el deportista haya elegido alguno de los golpeos, el prototipo guardará toda la información acerca de ese golpe. Posteriormente se realizará un análisis de esta información en el sistema web.

3. **Diseño de un sistema web para la visualización de la información recogida por el prototipo, visualización de estadísticas, resultados y progreso.**

El usuario podrá volcar toda la información registrada por el prototipo en un sistema web. Para que la interfaz sea lo más amigable posible, el diseño de la interfaz estará basado en diversas técnicas de usabilidad web estándar, que mostrarán los datos de forma amigable al usuario.

El portal web mostrará información relativa a la aceleración y el ángulo con el que el deportista realiza el golpeo de la pelota. Estos dos factores son claves para la mejora de la técnica de golpeo del deportista.

Se mostrarán estadísticas acerca de los entrenamientos realizados por un deportista. Estos entrenamientos constarán de diferentes golpeos, de los cuales también se podrá obtener información. Acerca del golpeo, el usuario del sistema web podrá ver las fases y evolución de sus golpeos. Así podrá evaluar si su técnica ha sido la adecuada. Para una mejor interpretación, el usuario podrá comparar sus golpeos con la ejecución de un golpeo de un jugador profesional. Para llevar a cabo esta comparativa se realizarán mediante técnicas audiovisuales cómo gráficos y vídeos, así el usuario podrá ver de forma sencilla la comparativa.

El fin de este sistema web, es que el usuario pueda hacer un análisis de la información relativa a sus golpes y así poder ir mejorando su técnica de golpeo sobre la pelota.

Con lo expuesto anteriormente, el jugador podrá hacer uso de un dispositivo para recoger todos los datos relacionados con su golpeo. Estos datos podrán ser interpretados en un sistema web, dónde el jugador podrá ver la evolución de sus golpes. Así, podrá evaluar qué debe mejorar en un futuro para poder pulir su técnica y mejorar en la ejecución de sus golpes.

CAPÍTULO 3

METODOLOGÍA

En este capítulo se presenta la arquitectura del proyecto, qué componentes posee y cómo se han implementado esos componentes, indicado qué alternativas se han barajado y cómo se han solucionado los problemas que se han presentado a lo largo del desarrollo del proyecto. También se va a detallar la metodología que se ha empleado para el desarrollo del presente TFG, así como dar explicación del por qué se decidió usar dicha metodología. Se usará este capítulo para discutir también los medios *software* y *hardware* usados durante el proyecto.

3.1. ARQUITECTURA

En esta sección se va a presentar la arquitectura del sistema propuesto. *I-Racket* es el nombre que toma el presente proyecto, cuyo objetivo principal es la monitorización y estudio del golpeo de pelota en deportes de raqueta para una mejora de la técnica de golpeo. Además, también servirá para ver el progreso con el que va evolucionando un deportista de acuerdo a sus entrenamientos.

En esta sección, por tanto, se mostrarán los componentes que conforman la arquitectura y se desarrollarán las peculiaridades de cada uno de ellos, indicando qué problemática resuelven, por qué se ha implementado el componente de esa forma y qué otras alternativas se barajaron.

3.1.1. Visión general de la arquitectura

En la Figura 3.1 se puede observar el esquema general del proyecto completo, con un alto nivel de abstracción. A lo largo de esta sección se explicará el funcionamiento de cada uno de los componentes, así como el funcionamiento del proyecto en sí. En dicha imagen se pueden ver los componentes principales que conforman el sistema desarrollado:

- **Usuario.** Se trata del usuario - deportista que portará en su muñeca el dispositivo *hardware* para la captación de los datos acerca del golpeo de pelota.
- **Dispositivo *hardware*.** Dispositivo compuesto de sensores, encargado de los datos de golpeo.
- **Sistema web.** Sistema encargado de la representación de los datos obtenidos.

Para lograr una monitorización de la actividad del deportista en sus entrenamientos, es necesario tomar datos acerca de la posición de la muñeca del deportista, con el fin de poder evaluar su golpeo y poder mejorar la técnica de golpeo. El fin del microcontrolador es exclusivamente

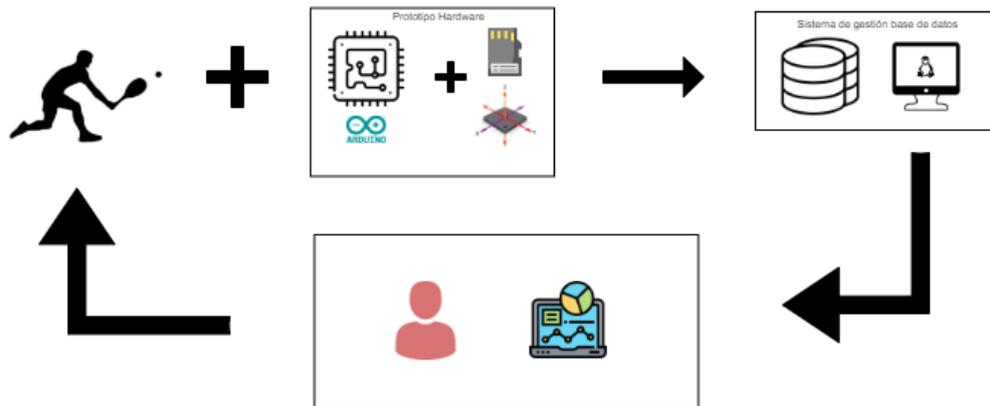


Figura 3.1: Esquema general del sistema propuesto.

la captación de datos para su posterior evaluación. Para hacer una captación de datos lo más eficiente posible, solo se capturan datos acerca del golpeo de la pelota y no de toda la actividad que se realiza en un entrenamiento. El fin de esto es captar datos acerca del golpeo sobre la pelota.

Para hacer un uso más interpretable del dispositivo, se ha diseñado una interfaz lo más amigable posible. Esta interfaz está capacitada para el reconocimiento del lenguaje gestual del usuario, gracias a esto se consigue una interfaz con el menor número de botones posibles. Así, se consigue una mayor simplicidad y menor peso en el dispositivo.

Una vez que se han registrado todos los datos a través del dispositivo *hardware*, estos son almacenados en una tarjeta micro SD. Dichos datos son cargados en el sistema *web* por el usuario. Este sistema *web* está conectado a una base de datos que registra todos los datos almacenados en el sistema, así como los datos que carga cada usuario acerca de sus actividad física y sus entrenamientos.

La evaluación de los datos es realizada por el sistema *web*. El sistema *web* será el encargado de mostrar la información recogida por el dispositivo *hardware*. Además, se encargará de mostrar la información de la manera más legible y amigable para que el usuario pueda hacer una buena interpretación sin excesivo esfuerzo.

De esta manera, el usuario es el encargado de llevar puesto el dispositivo *hardware*. Una vez acoplado el dispositivo en su muñeca, mediante el uso de lenguaje gestual, podrá decirle al dispositivo qué acción va a realizar éste. A continuación, el dispositivo empezará a recoger los datos a través de los sensores que lleva incorporado. Cuando el usuario haya finalizado su entrenamiento, podrá descargar los datos en el sistema web, donde podrá ver toda la información acerca de su entrenamiento.

En la Figura 3.2 se puede ver el flujo de trabajo que sigue este modelo de arquitectura descrita:

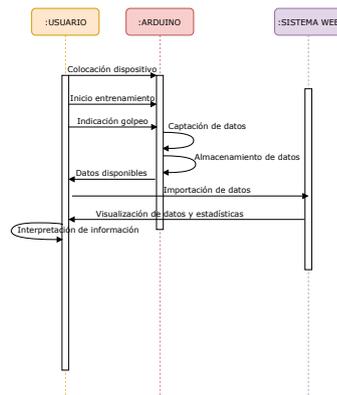


Figura 3.2: Flujo de trabajo.

3.1.2. Detalle de la arquitectura del sistema

Como se puede observar en la Figura 3.3, la arquitectura general se divide en tres niveles. En el nivel más bajo se sitúa el prototipo *hardware* cuya función principal es la comunicación con los distintos sensores y *leds* que conforman el dispositivo físico, para tomar datos que serán importantes en niveles posteriores. Para la captación de estos datos el usuario - deportista tendrá que indicar al prototipo qué tipo de golpeo va a realizar. Estas indicaciones se realizarán mediante lenguaje gestual, en el cual el usuario adoptará una posición determinada, indicando qué golpe va a efectuar. Para que el usuario tenga constancia del golpeo que está realizando, tendrá la ayuda de determinados *leds* de diferentes colores. Estos datos serán almacenados en un dispositivo de memoria externa. El prototipo contará con un módulo de tarjeta micro SD, en cual se guardarán todos los datos referentes al golpeo realizado por el usuario - deportista.

El usuario tendrá que interactuar con el prototipo *hardware*. Tendrá que indicarle que tipo de golpeo va a realizar en ese momento. Para ello, el usuario hará uso del lenguaje gestual para así poder interactuar con el prototipo de manera sencilla, lo que se traduce en una mejora del tiempo de interacción y una menor interrupción a la actividad física que realiza el deportista portador del prototipo.

En el siguiente nivel se encuadran la capa de lógica y la capa de persistencia. Los objetivos de este nivel son la importación de los datos que llegan del nivel previo y la visualización de los mismos de manera esquemática. Una vez que el prototipo ha registrado y guardado todos los datos, éstos son almacenados en el Sistema Gestor de Base de Datos (SGDB). Para que el sistema *web* pueda mostrar esos datos es necesario la comunicación con el SGDB. Este servidor se comunica con el sistema *web* mediante un protocolo *HTTP* utilizando servicios *REST*.

La función que tiene el sistema *web* es la visualización de los datos guardados en el servidor de base de datos de manera legible y amigable para el usuario. Este sistema *web* muestra todos los entrenamientos realizados por el usuario, así como las estadísticas propias de cada entrenamiento. El usuario, en cada entrenamiento, realiza distintos tipos de golpes, los cuales son analizados y mostrados al usuario de manera gráfica para su interpretación y valoración. El usuario cuenta con algunas recomendaciones acerca de los golpes, además puede comparar sus golpes con golpes de jugadores profesionales, para así analizar su golpeo y ver en qué puede mejorar. La

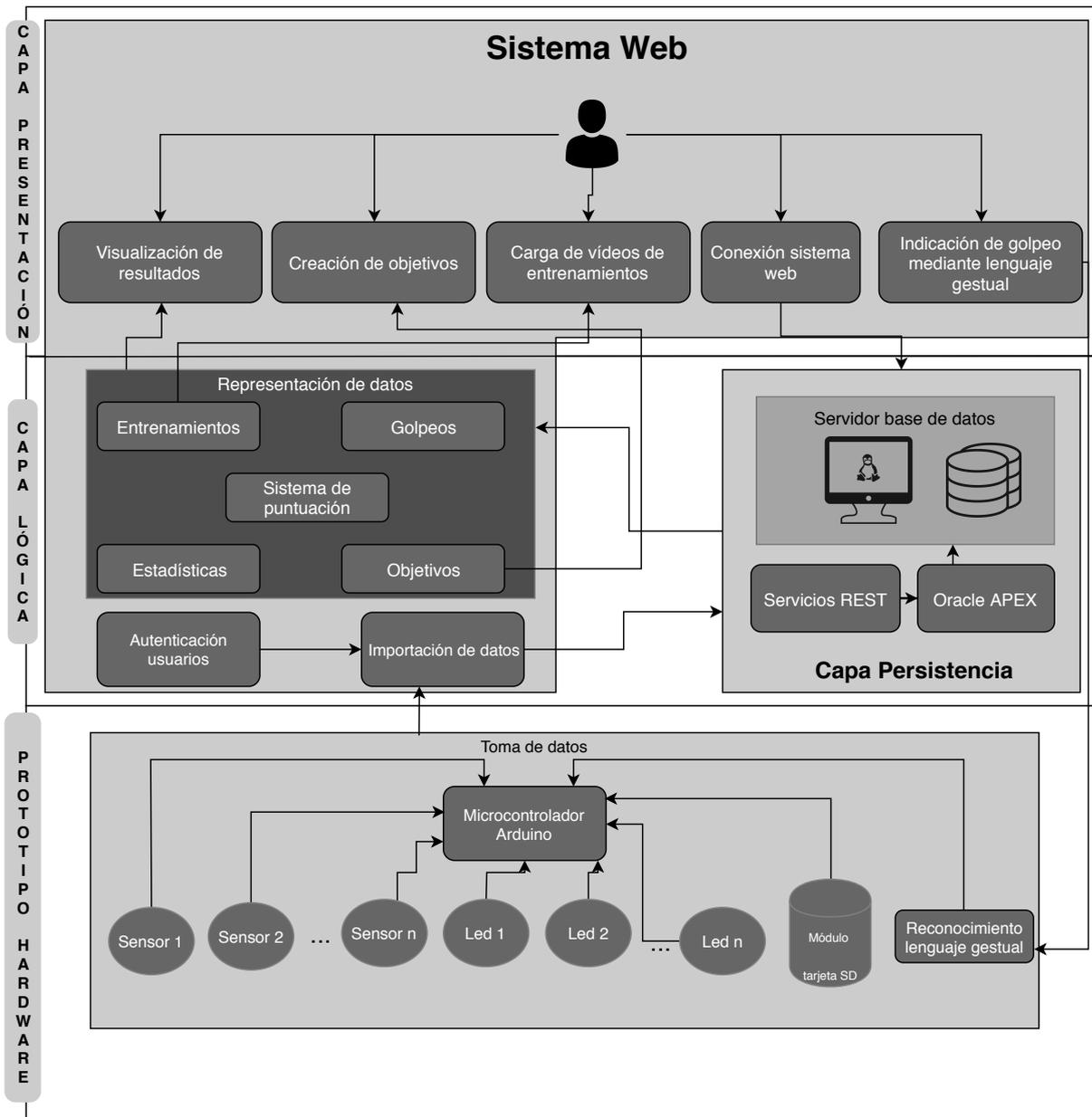


Figura 3.3: Arquitectura general.

aplicación cuenta con la visualización de videotutoriales, los cuales podrá usar el usuario para ver una demostración de cómo se realiza un golpeo.

El sistema *web* ofrece al usuario algunas recomendaciones acerca de sus golpesos y entrenamientos. El usuario puede visualizar en el sistema web si el golpeo realizado ha sido con demasiada aceleración o con muy poca aceleración, en función del tipo del golpeo que se esté evaluando. Además, el sistema cuenta con una funcionalidad basada en objetivos. Esta funcionalidad permite al usuario establecer una serie de objetivos para cumplir a lo largo de su uso con la aplicación y el prototipo. Esto hace que el usuario pueda ir mejorando sus golpesos y puliendo una técnica que irá mejorando con el cumplimiento de dichos objetivos.

El sistema *web* cuenta con un sistema de estadísticas generales, así como un sistema de puntuación. Un usuario puede tener una puntuación en función de los objetivos cumplidos y de los méritos que vaya acumulando. Un mayor número de golpesos, entrenamientos y cumplimiento de objetivos, se traduce en una mayor puntuación para el usuario.

Por último, se puede observar el tercer y último nivel de la arquitectura general, que se centra en mostrar información relevante a los usuarios del sistema *web*. El usuario puede visualizar y evaluar toda la información recogida por el prototipo *hardware*, así como interactuar con las distintas partes del sistema *web*. El usuario tiene la opción de cargar en el sistema *web* videos relativos a sus entrenamientos, permitiendo una comparativa de su golpeo con el de un profesional, de esta manera puede evaluar cómo ha realizado el golpeo y qué tiene que mejorar al respecto.

En las próximas secciones se hará una descripción técnica de cada uno de los componentes que forman parte de la arquitectura, así como aspectos del diseño de dichos componentes que se consideren necesarios.

3.2. PROTOTIPO *HARDWARE* PARA LA CAPTACIÓN DE DATOS DE ENTRENAMIENTO

En esta sección se presentará con detalle el diseño de un prototipo formado por un microcontrolador *Arduino*, diferentes sensores y actuadores. Con este prototipo se pretende realizar la captación de datos a partir de los sensores que lleva incorporado.

Para que el prototipo sea capaz de captar los datos de los sensores es necesario implementar las funcionalidades necesarias. Para poder implementar dichas funcionalidades, se ha hecho uso del lenguaje de programación *Arduino*. Este lenguaje está basado en los lenguajes C y C++. A lo largo de esta sección se explicará también algunos detalles importantes acerca de la implementación del dispositivo.

3.2.1. Diseño del dispositivo *hardware*

Para el desarrollo de este proyecto se ha usado una placa *Arduino Uno Rev3*¹. Los componentes utilizados para el desarrollo del prototipo son los siguientes:

- **Módulo MPU6050:** el *MPU6050* se utiliza para medir la aceleración y ángulo a la hora de realizar un golpeo de pelota.

¹<https://store.arduino.cc/arduino-uno-rev3>

- **Módulo Micro SD:** su función es almacenar los datos captados por el módulo *MPU6050*.
- **Leds:** ayudan al usuario a identificar qué está ocurriendo dentro del prototipo.

El esquema de conexión utilizado es el mostrado en la Figura 3.4:

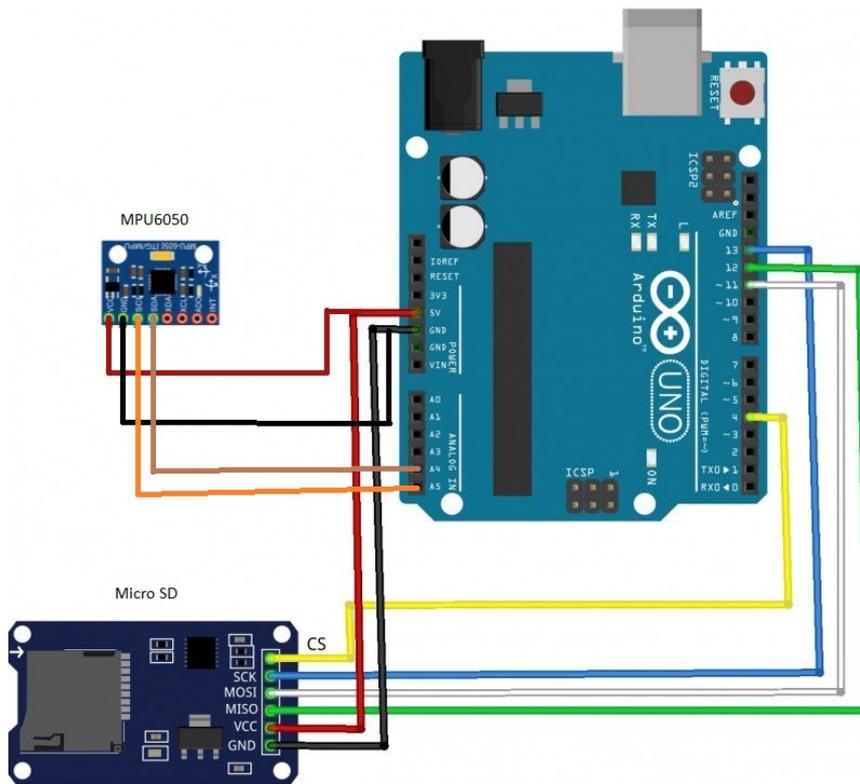


Figura 3.4: Esquema conexión.

En la Figura 3.4 se muestra un esquema general de conexiones entre el microcontrolador, los sensores y actuadores. En primer lugar, el *MPU6050* encargado de registrar la aceleración y ángulo de golpeo, utiliza los pines A4 y A5 para el *SCL* y *SDA* del dispositivo, además de las tomas de corriente (5V) y la toma de tierra (GND). El módulo de tarjeta micro *SD* utiliza más conexiones debido a que este dispositivo utiliza una arquitectura de tipo maestro-esclavo. Utilizará los pines 4, 11, 12, 13 correspondientes a las salidas del módulo *MISO*, *MOSI*, *SCK*, *CS*, además de las tomas de corriente (5V) y la toma de tierra (GND). Cada uno de estos pines serán explicados con más detenimiento en secciones posteriores.

Análisis de la problemática del golpeo

En el pádel hay muchos factores que intervienen en el golpeo de la pelota; altura, ángulo, velocidad, aceleración y posición de algunas partes del cuerpo. Este apartado se centrará en la aceleración y ángulo de golpeo sobre la pelota. Tanto la aceleración como el ángulo de golpeo sobre la pelota, son muy importantes en el pádel, ya que estos factores determinarán si el golpeo de la pelota ha sido realizado correctamente, lo que se traducirá en un golpe efectivo y ganador.

Cuando hablamos de ángulo sobre la pelota, puede ser ángulo del brazo, del codo, etc. Este proyecto se centrará y se hará especial hincapié en el ángulo que adquiere la muñeca a la hora de

realizar el golpeo. La muñeca es una parte fundamental del cuerpo que interviene en el golpeo de la pelota. Este ángulo que adquiere la muñeca es fundamental para realizar un buen golpeo sobre bola. El ángulo de muñeca junto con la combinación de una aceleración óptima, nos puede dar como resultado una buena ejecución del golpeo sobre bola.

Solución a la problemática del golpeo. Medición de velocidad y ángulo de rotación de la muñeca.

Para la medición, tanto de la aceleración como del ángulo de la muñeca, se disponían de varias opciones. La aceleración se puede medir con cualquier acelerómetro como es el modelo *ADXL345*, el cual nos ayuda a medir la aceleración en los tres ejes de coordenadas X, Y, Z.

A la hora de medir el ángulo de inclinación, se observó que podría realizarse con un giroscopio, el cual, nos mide el ángulo de rotación sobre la posición actual.

Para hacer una simplificación de la interfaz del prototipo a diseñar se ha optado por el uso del módulo *MPU6050*. Dicho módulo incorpora un acelerómetro y un giroscopio en entre sus componentes. Esto hace que tengamos un solo dispositivo en vez de dos, lo que ayuda también al usuario, ya que el prototipo no carga con tantos dispositivos, lo que se traduce en menor peso y mayor movilidad a la hora del golpeo.



Figura 3.5: MPU6050.

El sensor *MPU6050* es el encargado de recopilar datos acerca de la aceleración y posición de la muñeca del deportista. Este módulo contiene todo lo necesario para medir el movimiento con 6 grados de libertad, combinando un giroscopio de tres ejes y un acelerómetro de tres ejes en un mismo chip.

Para hacer uso de este sensor, se ha hecho uso de una librería de código libre². Con este módulo es muy posible que exista un desnivel en sus componentes (X,Y,Z). Para solucionar este problema, es necesario hacer una calibración del módulo, para que las medidas sean fiables. Para llevar a cabo esta calibración, se añaden unos *offset* a cada componente, tanto en la aceleración, como en la velocidad angular.

Para llevar acabo esta acción se ha creado un algoritmo en el cual se lleva a cabo lo descrito anteriormente. El algoritmo básicamente está modificando constantemente los *offsets* intentando eliminar el error con la medida real que deseamos. En este caso los *offsets* serán: Aceleración X=0, Aceleración Y=0, Aceleración Z=1. Giroscopio X=0, Giroscopio Y=0, Giroscopio Z=0.

¹ // **Leer las aceleraciones y velocidades angulares**

²<https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>

```

2  sensor.getAcceleration(&ax, &ay, &az);
3  sensor.getRotation(&gx, &gy, &gz);
4
5  // Filtrar las lecturas
6  f_ax = f_ax - (f_ax >>5)+ax;
7  p_ax = f_ax >>5;
8
9  f_ay = f_ay - (f_ay >>5)+ay;
10 p_ay = f_ay >>5;
11
12 f_az = f_az - (f_az >>5)+az;
13 p_az = f_az >>5;
14
15 f_gx = f_gx - (f_gx >>3)+gx;
16 p_gx = f_gx >>3;
17
18 f_gy = f_gy - (f_gy >>3)+gy;
19 p_gy = f_gy >>3;
20
21 f_gz = f_gz - (f_gz >>3)+gz;
22 p_gz = f_gz >>3;
23
24 //Cada 100 lecturas corregir el offset
25 if (counter==100){
26     //Calibrar el acelerometro a 1g en el eje z (ajustar el offset)
27     if (p_ax >0) ax_o--;
28     else {ax_o++;}
29     if (p_ay >0) ay_o--;
30     else {ay_o++;}
31     if (p_az -16384 >0) az_o--;
32     else {az_o++;}
33
34     sensor.setXAccelOffset(ax_o);
35     sensor.setYAccelOffset(ay_o);
36     sensor.setZAccelOffset(az_o);
37
38     //Calibrar el giroscopio a 0°/s en todos los ejes (ajustar el offset)
39     if (p_gx >0) gx_o--;
40     else {gx_o++;}
41     if (p_gy >0) gy_o--;
42     else {gy_o++;}
43     if (p_gz >0) gz_o--;
44     else {gz_o++;}
45
46     sensor.setXGyroOffset(gx_o);
47     sensor.setYGyroOffset(gy_o);
48     sensor.setZGyroOffset(gz_o);
49
50     counter=0;
51 }

```

Inicialmente se pone el dispositivo en posición horizontal. Hay que evitar moverlo para no interferir en su calibración. Después de comenzar con la calibración el algoritmo realiza lecturas tanto del acelerómetro como del giroscopio. Cada 100 lecturas se comprueba si los valores son cercanos a los valores que se desea obtener. Cuando se obtengan los valores deseados, tendremos el dispositivo *MPU6050* calibrado.

Los valores aproximados que debe tener el modulo una vez calibrado son los representados en la Tabla 3.1.

		Valores digitales de salida		
		Reposo	Máximo	Mínimo
Aceleración	Eje X	0g	32767g	-32768g
	Eje Y	0g	32767g	-32768g
	Eje Z	1g=16384g	32767g	-32768g
Velocidad angular	Eje X	0g	32767g	-32768g
	Eje Y	0g	32767g	-32768g
	Eje Z	0g	32767g	-32768g

Tabla 3.1: Valores digitales MPU6050.

Como se observa en la Tabla 3.1 se ha utilizado el valor de la fuerza g para la medición de la aceleración. La fuerza g es una medida de aceleración. Esta medida está basada en la aceleración que produciría la gravedad de la Tierra en un objeto cualquiera. Una aceleración de 1 g es generalmente considerado como igual a la gravedad estándar. La fuerza g para un objeto es de 0 g en cualquier ambiente sin gravedad, como el que se experimenta en el interior de una nave o habitáculo en caída libre.

Almacenamiento de los datos de entrenamiento

Todos los datos generados durante la ejecución en Arduino, se pierden una vez que se desconecta y hay que volver a cargar el *sketch*³. Esto es un problema ya que se perderían los datos de calibración cada vez que se apague el dispositivo, teniendo el usuario que dejar el dispositivo en una superficie plana cada vez que quiera empezar a usarlo. Para solucionar esto, se ha optado por guardar los datos de la calibración realizada en la memoria *EEPROM* que lleva integrada el microcontrolador Arduino. De esta manera, solo hace falta calibrar el dispositivo una sola vez, ya que los datos de calibración modifican el módulo con el último offset registrado.

Con esto conseguimos que el usuario no tenga que calibrar el dispositivo cada vez que quiera usarlo, así evitamos situaciones que puedan entorpecer el desarrollo de la actividad física del deportista.

Escalado de valores velocidad angular

A la hora de utilizar por primera vez el módulo *MPU6050* los valores que muestra no son representativos, ya que los datos tanto del acelerómetro como del giroscopio se muestran entre varios rangos, siendo entre -2g a 2g para el acelerómetro y entre $-250^\circ/\text{sec}$ y $250^\circ/\text{sec}$ para el giroscopio.

Como se muestra en la Tabla 3.2 los valores son poco interpretables. Para solucionar esto, se lleva a cabo un escalado de valores, para mostrarlos de manera más legible. Para poder ver el ángulo de inclinación, se deben escalar los valores que capta el giroscopio aplicando las siguientes fórmulas:

$$\Theta_x = \tan^{-1} \left(\frac{g_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (3.1)$$

³Programa que contiene el algoritmo programado.

		Valores digitales de la salida del sensor		
		En reposo	Máximo	Mínimo
Aceleración	EJE X	{16384 +- 8} = 1g	32767 = 2g	-32767 = -2g
	EJE Y	{0 +- 8} = 0g	32767 = 2g	-32767 = -2g
	EJE Z	{0 +- 8} = 0g	32767 = 2g	-32767 = -2g
Velocidad angular	EJE X	{0 +- 1} = 0deg/s	35767 = 250deg/s	-35767 = -250deg/s
	EJE Y	{0 +- 1} = 0deg/s	35767 = 250deg/s	-35767 = -250deg/s
	EJE Z	{0 +- 1} = 0deg/s	35767 = 250deg/s	-35767 = -250deg/s

Tabla 3.2: Valores digitales que mide el módulo *MPU6050*.

$$\Theta_x = \tan^{-1} \left(\frac{g_y}{\sqrt{a_x^2 + a_z^2}} \right) \quad (3.2)$$

$$\Theta_x = \tan^{-1} \left(\frac{g_z}{\sqrt{a_y^2 + a_x^2}} \right) \quad (3.3)$$

Una vez aplicadas las dos fórmulas, los datos quedarían de la siguiente manera:

		Valores digitales de salida		
		Reposo	Máximo	Mínimo
Velocidad angular	Eje X	0g	90°	-90°
	Eje Y	0g	90°	-90°
	Eje Z	0g	90°	-90°

Tabla 3.3: Valores escalados velocidad angular.

Bus *I2C*

El módulo *MPU6050* utiliza comunicación *I2C*. *I2C* es un bus que actúa en base a un protocolo de comunicación serial. El bus *I2C* requiere únicamente dos cables para su funcionamiento, uno para la señal de reloj (CLK) y otro para el envío de datos (SDA), lo cual es una ventaja frente al bus *SPI*. En la Figura 3.6 se observa la conexión al bus *I2C*. La placa *Arduino* tiene dos pines reservados para la conexión de este bus, que son el pin *A4* y *A5*.

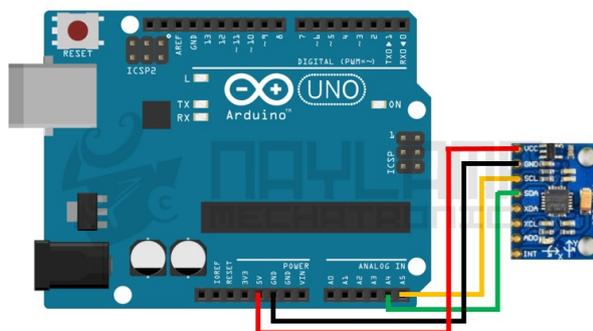


Figura 3.6: Conexión bus *I2C*.

En el bus cada dispositivo dispone de una dirección, que se emplea para acceder a los diferentes módulos (en caso de tener varios) de forma individual. Esta dirección puede ser fijada por *hardware* (en cuyo caso, frecuentemente, se pueden modificar los últimos 3 bits mediante *jumpers* o interruptores) o totalmente por *software*. El bus I2C tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro inicia la comunicación con los esclavos, y puede mandar o recibir datos de los esclavos. Los esclavos no pueden iniciar la comunicación (el maestro tiene que preguntarles), ni hablar entre si directamente.

Otra gran ventaja de este bus es que no necesita resistencias en su circuito. El protocolo I2C prevé resistencia de *Pull-UP* de las líneas Vcc. Como el protocolo I2C utiliza la librería *WIRE*⁴, que activa las resistencias internas. Estas resistencias tienen un valor de entre 20-30 KOhmios.

El bus I2C es síncrono. El maestro proporciona una señal de reloj, que mantiene sincronizados a todos los dispositivos del bus de esta manera, se elimina la necesidad de que cada dispositivo tenga su propio reloj.

En el presente proyecto el bus I2C es usado para la comunicación con la placa de Arduino. Este bus proporciona dos grandes ventajas. La primera ventaja es que requiere pocos cables, lo que facilita el manejo del prototipo por parte del usuario - deportista al tener menos complementos en el prototipo *hardware*. La segunda ventaja es la velocidad de envío y recepción de datos, lo que ayuda a procesar datos sin pérdida de información mientras el usuario está realizando la práctica deportiva.

Módulo tarjeta micro SD

Una vez captados los datos con el sensor *MPU6050* es necesario su almacenamiento o el traspaso de los mismos para poder ser visualizados e interpretados por el usuario en el sistema web. Para el almacenamiento de estos datos existían varias posibilidades. En la Tabla 3.4 se muestra un resumen de las posibilidades que se valoraban para poder almacenar los datos recogidos por los sensores:

Dispositivo	Tamaño físico	Velocidad lectura/escritura
Disco duro externo	Grande	640MB/s
Módulo tarjeta SD	Medio	90MB/s
Módulo tarjeta micro SD	Pequeño	90MB/s

Tabla 3.4: Dispositivos de almacenamiento.

Valorando las diferentes opciones que se tenían para almacenar los datos registrados por los sensores que incorpora el prototipo, se ha optado por la opción de utilizar el módulo de tarjeta micro SD. Su pequeño tamaño permite acompañar al prototipo en un diseño con el menor número de componentes posibles y con un tamaño reducido, para así, poder elaborar un prototipo en el que entorpezca lo menos posible el desarrollo de la actividad deportiva del deportista.

El módulo de **tarjeta micro SD** es el encargado de guardar todos los datos recogidos por el módulo *MPU6050*. Este módulo utiliza una librería de código libre⁵ que el propio lenguaje de programación Arduino lleva integrado. En el módulo micro SD solo se registran los datos referentes al golpeo de la pelota, para así obtener una información mas detallada y poder monitorizar el golpeo con mas exactitud.

⁴<https://www.arduino.cc/en/Reference/Wire>

⁵<https://www.arduino.cc/en/Reference/SD>



Figura 3.7: Módulo Micro SD.

El módulo de tarjeta micro SD utiliza el siguiente sistema de conexión:

Módulo micro SD	Pin arduino
GND	GND
+3.3V	No conectado
+5V	+5V
CS	4
MOSI	11
SCK	13
MISO	12

Tabla 3.5: Conexión Módulo Tarjeta micro SD.

Bus *SPI*

El módulo de tarjeta micro SD utilizada para comunicarse con el Arduino el bus *SPI*. El bus *SPI* tiene una **arquitectura de tipo maestro-esclavo**. El dispositivo maestro (*master*) puede iniciar la comunicación con uno o varios dispositivos esclavos (*slave*), y enviar o recibir datos de ellos. Los dispositivos esclavos, no pueden iniciar la comunicación, ni intercambiar datos entre ellos directamente.

En el bus *SPI* la comunicación de datos entre maestros y esclavo se realiza en dos líneas independientes, una del maestro a los esclavos, y otra de los esclavos al maestro, lo que implica que la comunicación es **Full Duplex**, es decir, el maestro puede enviar y recibir datos simultáneamente.

Debido a la gran cantidad de datos que se generan cuando el deportista está realizando su actividad deportiva, es necesario que en todo momento el maestro (placa Arduino) esté conectado con el esclavo (módulo micro SD), para registrar datos cuando sea necesario. Esto ayudará a obtener datos sin pérdida de información, lo que va a proporcionar unos datos más reales para así a la hora de hacer la representación de estos datos y mostrárselos al usuario, pueda tener una visión de su golpeo lo más real posible. Además de esto, el bus *SPI* proporciona una alta velocidad de transmisión (8 Mhz), lo que ayuda a guardar toda la información sin pérdidas.

Por tanto, como se puede apreciar en la Figura 3.8, el bus *SPI* requiere un mínimo de tres líneas de conexión para su correcto funcionamiento:

- **MOSI** (*Master-out,slave-in*) para la comunicación del maestro al esclavo.
- **MISO** (*Master-in,slave-out*) para la comunicación del esclavo al maestro.
- **SCK** (*Clock*) señal de reloj enviada por el maestro.



Figura 3.8: Esquema de conexión módulo micro SD.

- **SS** (*Slave Select*) línea adicional para cada dispositivo esclavo conectado (en caso de que tuviéramos otro módulo o dispositivo con el mismo bus).

feedback usuario

Una vez que tenemos el módulo *MPU6050* para registrar todos los datos relativos a la aceleración y el ángulo de rotación y el módulo micro SD para guardar todos estos datos de manera persistente, se tiene que identificar en la interfaz del prototipo que acción se está realizando en cada momento. El usuario - deportista portador del prototipo *hardware* deberá saber en todo momento que acción está realizando.

Cómo solución a esta problemática se han valorado diferentes opciones. En primer lugar se valoró la utilización de dispositivos acústicos, que mediante diferentes sonidos, indicarán al usuario qué acción está realizando. Otra opción que se valoró es la utilización de *leds*, para indicar mediante su señal luminosa, qué se acción esta realizando el usuario.

Finalmente, en base al cumplimiento del objetivo de diseñar una interfaz con el menos número de componentes y que sea lo más interpretable para el usuario, se hará uso de *leds*. Estos *leds* indicarán al usuario qué acción está realizando y qué está pasando por dentro de prototipo. Para distinguir cada una de las acciones, se hará uso de *leds* de diferentes colores. La funcionalidad de los diferentes *leds* será expuesta con más detalle en la siguiente sección.

Interfaz gestual de comunicación entre deportista y dispositivo *hardware*

Análisis problemática comunicación entre deportista y dispositivo

El usuario debe de dar ordenes al prototipo mediante la interfaz que se ha diseñado para indicarle qué tipo de golpeo va a realizar. Esta acción debe de hacerse de manera rápida y legible para el usuario, con el objetivo de que entorpezca lo menos posible su actividad física.

Se han valorado diferentes opciones para llevar a cabo esta problemática. En primer lugar consideró la opción de utilizar un reconocimiento por voz, mediante el cual el usuario indicaba al prototipo el tipo de golpeo que va a realizar. Esta opción fue descartada debido al gran tamaño de este dispositivo, ya que se compone de un módulo más un micrófono, el cual al estar el deportista fatigado por la práctica deportiva, sería menos eficiente. Otra opción que se contempló para resolver esta problemática es la utilización de un módulo de reconocimiento de gestos. Este módulo se basa en el reconocimiento de direcciones, es decir, el módulo lleva incorporado un sensor, el cual, reconoce cuando pasas la mano o cualquier parte del cuerpo por encima de sensor, la dirección que lleva, así se podrían traducir las ordenes de *Derecha*, *izquierda*, *etc* en diferentes tipos de golpeo. Esta era una buena opción que se podría haber utilizado en este proyecto, pero

surge el problema de que se añade un dispositivo más a la interfaz que se está diseñando. Al trabajar con un modelo de *Arduino* pequeño, no se dispone de demasiadas conexiones para poder conectar todos los módulos que se deseen en el prototipo, es por ello que se ha buscado la simplificación de la interfaz, para que cuente con el menor número de componentes posibles, reduzcan el peso de dicha interfaz y que ayuden a la interacción del usuario con el prototipo.

Solución comunicación entre deportista y dispositivo

Finalmente, con el objetivo de incorporar el menor número de dispositivos a la interfaz, se opta por el uso de **lenguaje gestual**. El usuario tendrá que hacer uno de movimientos específicos para dar órdenes al prototipo *hardware*. Esto facilitará al usuario portador del dispositivo la interacción con el prototipo y un fácil manejo. Así, se podrá diseñar una interfaz lo mas amigable y ágil posible, de tal forma que el deportista no tenga que parar su actividad para configurar el dispositivo o el modo de juego que quiere realizar.

A la hora de realizar el lenguaje gestual, el usuario tendrá cuatro opciones donde elegir. La primera opción será para decir al prototipo qué tipo de golpeo va a realizar. Dentro de esta opción el usuario podrá elegir entre tres tipos de golpesos. Para este proyecto se han seleccionado los tres golpes principales del pádel (*drive*, globo y volea).

Una vez que el usuario ha elegido el golpe que va a realizar, necesitará ponerse en posición de espera para recibir la pelota, así indicará al prototipo que está listo para realizar el golpeo.

A continuación se muestra un modelo de como serían las diferentes posiciones que tiene que adoptar el usuario para indicarle mediante gestos al prototipo, cuál es la acción que va a realizar:

En la Figura 3.9 se puede observar como es la posición que debe adoptar el usuario para indicarle al prototipo que va a realizar un golpe de *Drive*. Para adoptar esta posición el usuario debe estirar el brazo portador del prototipo y colocar el prototipo de manera alineada con el brazo.



Figura 3.9: Posición golpe de derecha.

La Figura 3.10 muestra la posición que debe adoptar el usuario para indicar al prototipo que va a realizar un golpe de globo. La posición es similar a la anterior, solo que esta vez, el prototipo se debe inclinar 90° hacia la derecha con respecto a la posición anterior.

La Figura 3.11 muestra la posición que debe adoptar el usuario para indicar que va a realizar un golpe de volea. Esta posición es similar a las anteriores, solo que esta vez, la posición que debe adoptar el prototipo es con un giro de 180° con respecto a la posición de derecha.



Figura 3.10: Posición golpe de globo.



Figura 3.11: Posición golpe de volea.

Por último, el jugador deberá indicarle al prototipo que está listo para realizar el golpeo de la pelota. Para ello, el jugador deberá colocarse en posición de espera, para recibir la bola y realizar el golpeo. La Figura 3.12 muestra como sería la posición.



Figura 3.12: Posición de reposo

Para la elección del tipo del golpeo, como se ha expuesto anteriormente, el usuario debe

adoptar una posición determinada para la elección de cada golpe. Cada usuario es diferente, y no todos adoptarán la misma posición exactamente en cada elección de su golpeo. Para resolver esto se ha estructurado la elección del golpeo haciendo uso tanto del giroscopio como del acelerómetro, en unos rangos de valores, tanto de los ejes del giroscopio como del acelerómetro. El siguiente código muestra como se ha implementado esta funcionalidad:

```

1  if ((angX<anguloXmaximoD && angX>-anguloXminimoD) && (angY <-
    <- <anguloYmaximoD && angY >-anguloYMinimoD) && <-
    <- (ax>acelXminimoD && ax <acelXmaximoD) && (ay <-
    <- >-acelYminimoD && ay<acelYmaximoD) && (az>acelZminimoD <-
    <- && az<acelZmaximoD)) {
2      tipo_golpe=1;
3      Serial.println(F("Derecha"));
4  } else if ((angX<anguloXmaximoG && angX>anguloXminimoG) && <-
    <- (angY <anguloYmaximoG && angY >-anguloYminimoG) && <-
    <- (ax>acelXminimoG && ax <acelXmaximoG) && (ay <-
    <- >-acelYminimoG && ay<acelYmaximoG) && (az>acelZminimoG <-
    <- && az<acelZmaximoG)) {
5      tipo_golpe=2;
6      Serial.println(F("Globo"));
7  } else if ((angX<anguloXmaximoV && angX>anguloXminimoV) && <-
    <- (angY <anguloYmaximoV && angY >-anguloYminimoV) && <-
    <- (ax>acelXminimoV && ax <acelXmaximoV) && (ay <-
    <- >-acelYminimoV && ay<acelYmaximoV) && (az>-acelZminimoV <-
    <- && az<-acelZmaximoV)) {
8      tipo_golpe=2;
9      Serial.println(F("Volea"));
10 }

```

En función del ángulo que adopte el usuario y la aceleración que tenga en ese momento, se seleccionará un golpeo u otro. En la Tabla 3.6 se observa el resumen del rango de valores utilizados en esta funcionalidad. Se ha utilizado un rango de valores amplio, debido a que cada usuario es diferente, y no siempre se va a adoptar exactamente la misma posición. Para asegurar el correcto funcionamiento de esta funcionalidad, se han realizado una serie de test en los que se comprueba que dicha funcionalidad funciona correctamente.

Golpeo	Valores ángulo X	Valores ángulo Y	Valores aceleración X	Valores aceleración Y	Valores aceleración Z
Drive	[+15°,-5°+]	[+5°,+20°]	[-7000g,+6000g]	[-5000g,+2000g]	[+13000g,+18000g]
Globo	[+90°,+40°]	[+5°,-25°]	[+11000g,+19000g]	[-9000g,+1000g]	[+0g,+8000g]
Volea	[+10°,+50°]	[-15°,+15°]	[+3000g,+13000g]	[-7500g,+4000g]	[-20000g,-10000g]

Tabla 3.6: Resumen rango de valores para elección de golpeo

Actuadores prototipo *hardware*

Una vez que el usuario ha dado las ordenes que desea al prototipo, el usuario debe saber en todo momento qué es lo que está realizando el prototipo, es decir, el usuario debe saber en qué tipo de golpeo está el prototipo en ese momento para realizar el golpeo de adecuado, o simplemente para poder cambiar el tipo de golpeo. Además, el usuario deberá saber cuándo el prototipo está preparado para registrar los datos de su golpeo, para que así el jugador pueda registrar correctamente su golpeo.

La solución a esta problemática pasa por el uso de determinados *leds* que ayudan al usuario a identificar en qué tipo de golpeo se encuentra, si el prototipo está preparado para recibir datos o

si el prototipo tiene algún problema en el registro de datos. En la Tabla 3.7 se muestra un resumen en relación a los *leds* utilizados y la información que aportan acerca de la actividad que se está realizando.

Led	Acción que indica
Verde	Listo para ejecutar golpeo
Azul	Modo golpe de derecha
Amarillo	Modo golpe de globo
Blanco	Modo golpe de volea
Rojo parpadeante	Problema al registrar datos

Tabla 3.7: Utilización de *leds* y acción que indican.

Fijación prototipo *hardware*

Una vez montado y diseñado la interfaz del prototipo, dicho dispositivo tiene que ser acoplado a la muñeca del deportista para poder registrar los datos referentes a su golpeo. En solución a esta problemática se ha optado por el diseño y construcción de una correa de cuero. Dicha correa cuenta con varias medidas, para que sea ajustable a diferentes tamaños de muñeca, así el prototipo podrá ser utilizado por varios usuarios - deportistas.

El diseño de esta correa se muestra en la Figura 3.13. Esta correa tiene 8 cm de ancho y 20 cm de largo:

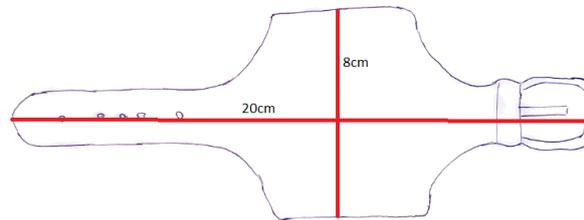


Figura 3.13: Diseño correa de fijación.

En la Figura 3.14, se puede observar como queda finalmente en la muñeca del deportista. Para garantizar una sujeción adecuada del prototipo *hardware* a la correa se ha fijado el dispositivo mediante cuatro tornillos a la superficie de la correa. Con esto garantizamos que el dispositivo no se mueva y tenga la estabilidad necesaria para registrar datos sobre el golpeo que se realiza.

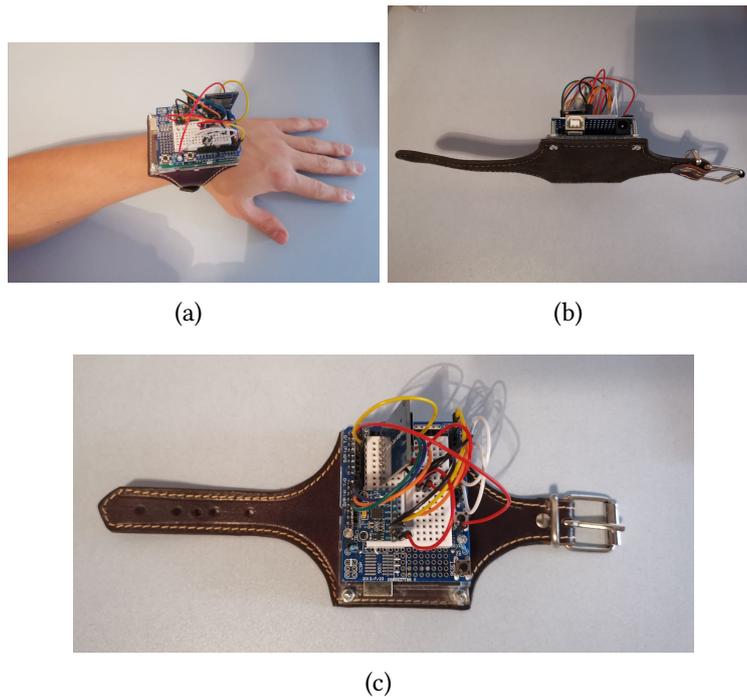


Figura 3.14: Correa de cuero adaptada al prototipo *hardware*.

3.3. SISTEMA WEB

En esta sección se presentará con detalle el sistema *web* para la monitorización de entrenamientos y mejora progresiva de la técnica de golpeo. Este sistema *web* será el encargado de la visualización de los datos recogidos por el prototipo *hardware*. En este sistema se mostrarán los diferentes golpes que ha realizado el usuario en cada uno de sus entrenamientos, así como estadísticas generales que ayuden al usuario a ir mejorando y puliendo su técnica. Los golpes de cada entrenamiento, podrán ser comparados con golpes modelo, golpes que han sido realizados por jugadores profesionales. Además, el sistema *web* contará con funcionalidades basadas en objetivos, que ayuden al deportista a establecer las metas a conseguir conforme va realizando entrenamientos.

El sistema *web* ha sido desarrollado con una tecnología basada en *JavaScript* llamada **ORACLE APEX**⁶. **ORACLE APEX** es una herramienta de desarrollo de páginas *web*, diseñada a partir del *software* de base de datos *Oracle*.

3.3.1. Arquitectura sistema *web*

El sistema *web* desarrollado en el presente proyecto sigue un diseño multicapa, el cual cuenta con varios módulos que encapsulan las diversas funcionalidades a desarrollar en el sistema. El sistema *web* contará con técnicas basadas en gráficos y vídeos, para una fácil interpretación del usuario. En la Figura 3.15 se muestra el esquema general de las funcionalidades que ofrece el sistema *web*. Dicho sistema se ha organizado en cinco módulos funcionales. En cada uno de estos módulos se desarrollan diferentes funcionalidades que servirán para ir obteniendo versiones más funcionales y para conseguir el objetivo propuesto de diseñar y desarrollar un sistema *web* que

⁶<https://apex.oracle.com/es/>

sirva para que el usuario pueda mejorar su técnica de golpeo sobre la pelota.

Lo módulos implementados en el sistema *web* se describen a continuación:

- **Módulo 1. Importación de datos.** Primer módulo en el que se desarrolla el proceso de importación de datos, desde el prototipo *hardware* al sistema *web*.
- **Módulo 2. Autenticación de usuarios.** Se desarrolla la funcionalidad de autenticación de usuarios para establecer seguridad sobre los datos que sostiene el sistema *web*.
- **Módulo 3. Representación de entrenamientos.** En este módulo se desarrollan dos funcionalidades que servirán para representar los datos obtenidos del prototipo *hardware*.
- **Módulo 4. Comparación de golpes.** Se desarrollan tres funcionalidades que servirán al usuario para comparar sus golpes realizados con los de otros usuarios, con la ayuda de datos, gráficas y vídeos.
- **Módulo 5. Evolución y mejora de técnica de golpeo.** En el último módulo planteado, se ha desarrollado dos funcionalidades para ofrecer al usuario un seguimiento de las mejoras que va obteniendo con el paso de los entrenamientos. Una de las funcionalidades estará basada en objetivo. El usuario tendrá la posibilidad de marcarse unos objetivos a completar. La segunda funcionalidad, estará basada en retos, en la que el usuario tendrá que ir superando unos retos estipulados por el sistema.

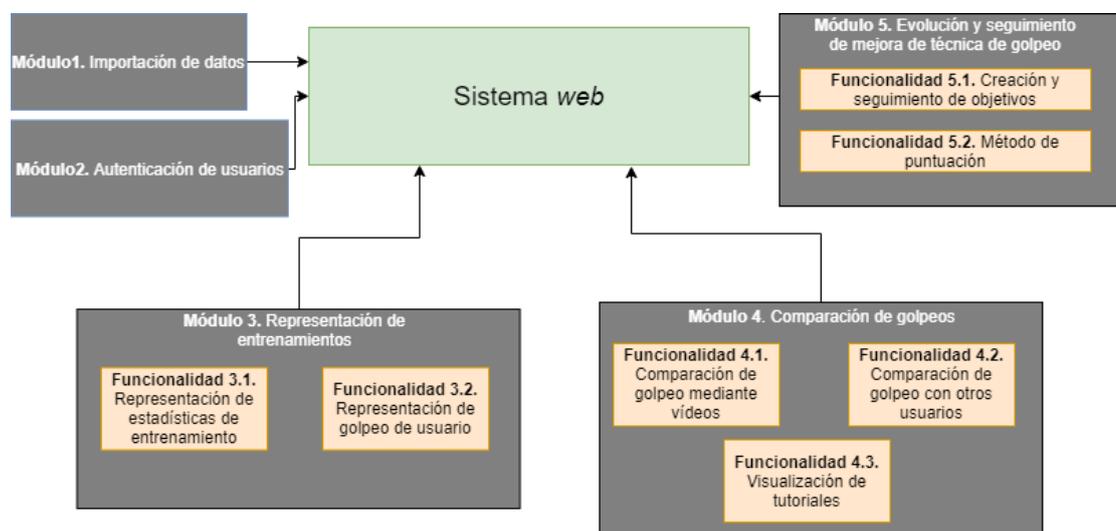


Figura 3.15: Esquema sistema *web*.

A continuación se explica con detalle los módulos anteriormente descritos:

Módulo 1. Importación de datos

Una vez registrados los datos por el prototipo *hardware* es necesario cargar estos datos en el sistema *web* para su posterior visualización. Los datos generados por el prototipo *hardware* son almacenados en un módulo de tarjeta micro SD, el cual registra estos datos en un archivo de datos .csv. Para cargar los datos en el sistema *web*, el propio deportista deberá descargar los archivos .csv del módulo de tarjeta micro SD y cargar esos archivos en el sistema *web*.

Para la carga de estos archivos se ha hecho uso de un *plugin* de Oracle APEX, el cual permite hacer cargas de datos masivas en el sistema *web*. Para realizar cargas de datos masiva, el *plugin* lee la información que se vaya a cargar de una tabla que contiene toda la información. Esta funcionalidad se ha desarrollado para repartir la información de forma consistente en el modelo relacional que se ha implementado.

Para realizar la carga y repartir toda la información se ha implementado una funcionalidad con el lenguaje *pl/sql* que se encarga de cargar estos datos en el sistema *web*, más concretamente en la base de datos que está asociada al sistema *web*. El procedimiento implementado para realizar esta carga es el siguiente:

```

DECLARE
cont1 number:=0;
cont2 number;
CURSOR aux_cursor
5 IS
6   select * from auxiliar;
7 BEGIN
8   FOR aux_rec in aux_cursor
9   LOOP
10  if cont1 < 1 then
11    cont2:=seq_entrenamiento.nextval;
12    insert into entrenamiento values (cont2, (select ←
        ↳ id_tipo_entrenamiento from tipo_entrenamiento where ←
        ↳ descripcion=aux_rec.tipoentrenamiento), ←
        ↳ to_date(aux_rec.FechaEntrenamiento, 'dd/mm/yyyy'), ←
        ↳ :ID_USUARIO);
13    cont1:=1;
14  end if;
15  insert into golpe values (SEQ_GOLPE.nextval, ←
        ↳ aux_rec.numerogolpe, ←
        ↳ aux_rec.AceleracionX, aux_rec.AceleracionY, ←
        ↳ aux_rec.AceleracionZ, ←
        ↳ aux_rec.AnguloX, aux_rec.AnguloY, ←
        ↳ to_date(aux_rec.fecha, 'dd/mm/yyyy_HH24:MI'), ←
        ↳ :ID_USUARIO, cont2);
16  END LOOP;
17 END;

```

NumeroGolpe	TipoEntrenam	AceleracionX	AceleracionY	Aceleracionz	AnguloX	AnguloY	FechaEntrenamiento
1	Derecha	13548	3072	6692	61.58	11.54	01/05/2019
1	Derecha	7900	-3308	9100	39.82	-15.57	01/05/2019
1	Derecha	14816	1680	6200	66.64	5.64	01/05/2019
2	Derecha	14440	3020	8120	58.99	10.43	01/05/2019
2	Derecha	9092	-3516	11392	37.38	-13.59	01/05/2019
2	Derecha	15304	2924	7672	61.65	9.84	01/05/2019
2	Derecha	15056	4212	7196	60.45	13.98	01/05/2019
3	Derecha	14796	2400	8388	59.79	8.17	01/05/2019
3	Derecha	4932	9196	11956	17.49	34.89	01/05/2019
3	Derecha	15040	1284	6104	67.60	4.54	01/05/2019
4	Derecha	14432	3756	8420	57.37	12.50	01/05/2019
4	Derecha	15000	25044	10128	29.30	53.91	01/05/2019
4	Derecha	17792	-996	5176	73.36	-3.40	01/05/2019
4	Derecha	13512	1852	10140	53.00	6.82	01/05/2019
4	Derecha	14144	2640	9508	55.30	9.07	01/05/2019
4	Derecha	13456	2740	10056	52.66	9.10	01/05/2019

Figura 3.16: Ejemplo de archivo *csv* generado por el prototipo *hardware*.

Para llevar a cabo la carga de datos se ha hecho uso de una tabla “Auxiliar” que se encarga de almacenar todos los datos importados al sistema *web*. Esta tabla sirve de “puente” antes de repartir

toda la información a las correspondientes tablas que conforman el esquema relacional de la base de datos. Una vez cargados los datos en esta tabla “Auxiliar”, se hace uso del procedimiento especificado anteriormente. Este procedimiento se encarga de repartir toda la información a las diferentes tablas que conforman el esquema relacional, para que la información esté organizada de forma consistente. En la Figura 3.17 se puede observar representado cómo funciona el proceso de carga:

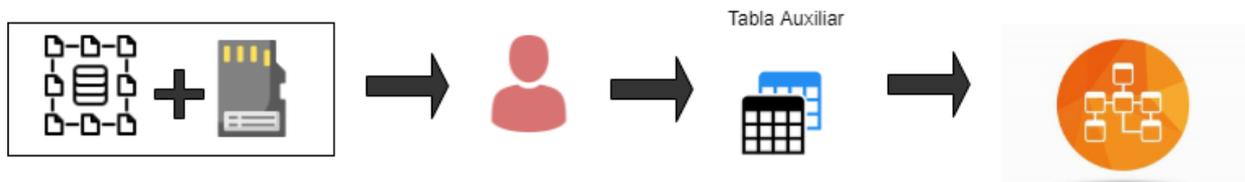


Figura 3.17: Esquema para la importación de datos desde el prototipo *hardware* hacia el sistema *web*. Los datos registrados por el prototipo, se guardan en una tarjeta micro SD, el usuario carga estos datos en el sistema *web* (en la tabla auxiliar) y el procedimiento descrito anteriormente se encarga de distribuir la información.

En base al modelo relacional de base de datos (se explicará en secciones posteriores) que se ha utilizado en el presente proyecto es necesario el uso de una tabla *Auxiliar* para poder repartir la información de forma correcta y escalable. En este modelo de datos se hace uso de restricciones y de relaciones entre tablas, por ello es importante organizar bien toda la información que registren los usuarios en el sistema *web*. Una opción sería almacenar toda la información en una tabla y no tener que repartirla en varias tablas, pero este método es muy poco escalable y eficiente. Por ello se opta por la utilización de un proceso intermedio en el cual se almacena la información en una única tabla y luego es repartida de forma organizada, cumpliendo con todas las restricciones del modelo de datos. Una vez repartida toda la información es sus respectivas tablas, la información que estaba en la tabla “Auxiliar” queda borrada. Este método se realiza para cumplir con todas las restricciones del modelo relacional y para organizar la información de manera correcta y escalable.

Módulo 2. Autenticación de usuarios

Una vez registrados los datos por el prototipo *hardware*, éstos son almacenados en el sistema *web* para su posterior procesamiento y representación de los mismos. Es importante mantener la seguridad y a privacidad de estos datos, ya que no todos los usuarios del sistema *web* pueden ver todos los datos que hay en él. Solo los usuarios administradores de la aplicación podrán ver toda la información que se almacena. Estos usuarios son los administradores, tanto del sistema *web*, como del sistema gestor de base de datos.

Con el objetivo de garantizar la seguridad y la privacidad de los datos que se almacenan en el sistema *web*, se ha implementado una funcionalidad de autenticación de usuarios. Para llevar a cabo el desarrollo de la siguiente funcionalidad, se ha creado un *PACKAGE* de seguridad que implementa el siguiente código:

```
CREATE OR REPLACE NONEDITIONABLE PACKAGE BODY ↵
  ↵ "ALVAROGOMEZ"."PKG_SECURITY" as
2
3
4function my_authentication (p_username in varchar2 , p_password ↵
  ↵ in varchar2 ) return boolean
```

```

51 s
6   l_user usuario.username%type := upper(p_username);
7   v_password usuario.password%type;
8   e_password usuario.password%type;
9   l_id usuario.id_usuario%type;
10 begin
11
12     e_password := encriptar_pwd(p_Password, p_username);
13
14     if p_username is null or p_Password is null then
15         Apex_Util.Set_Session_State('Please enter
16         ↪ username and password');
17         return false;
18     end if;
19     ----
20     BEGIN
21         select u.password
22         into
23         v_password
24         from usuario u
25
26         where u.username=p_username;
27     EXCEPTION
28         WHEN NO_DATA_FOUND then
29         Apex_Util.Set_Session_State('Usuario no
30         ↪ encontrado');
31         Return False;
32     end;
33
34     if v_password=e_password or v_password=p_Password then
35         return true;
36     end if;
37     Apex_Util.Set_Session_State('La contrasea
38     ↪ introducida no es correcta. ');
39
40     Return False;
41 end;
42
43 Function encriptar_pwd(p_password IN VARCHAR2, p_username IN ↪
44     ↪ VARCHAR2) return
45     ↪ VARCHAR2 is
46     salt varchar2(4000) := 'lkjhkjghGUOgljkbh1GLjKBH87978';
47     begin
48         return utl_raw.cast_to_raw(
49             dbms_obfuscation_toolkit.md5(
50                 input_string => salt || ↪
51                 ↪ p_password || upper
52                 ↪(p_username)
53             )
54         );
55     end;
56
57 Procedure Process_Login(p_User_Name Varchar2 ,p_Password ↪
58     ↪ Varchar2 ,p_App_Id Number) As
59
60     v_Result Boolean := False;

```

```

57
58     id_user number;
59 Begin
60     v_Result := my_authentication(p_User_Name , p_Password);
61     If v_Result = True Then
62         -- Redirect to Page 9 (Home Page).
63         SELECT ID_USUARIO into id_user from usuario where ↵
64             ↵ username = p_User_Name
65
66         apex_util.set_session_state('ID_USUARIO', id_user);
67         apex_util.set_session_state('USERNAME', ↵
68             ↵ lower(p_User_Name));
69
70         Wwv_Flow_Custom_Auth_Std.Post_Login(p_User_Name -- ↵
71             ↵ p_User_Name
72
73             ,p_Password -- ↵
74                 ↵ p_Password
75             ,v('APP_SESSION') -- ↵
76                 ↵ p_Session_Id
77             ,p_App_Id || ':1' -- ↵
78                 ↵ p_Flow_page
79             );
80     Else
81         Owa_Util.Redirect_Url('f?p=&APP_ID.:9999:&SESSION. ');
82     End If;
83 End;
84
85 end;

```

Para desarrollar esta funcionalidad se han implementado tres funciones principales. La primera de ellas es la función *my_authentication*. En dicha funcionalidad se pasan por argumentos el *username* y la *password*. Se comprueba que todos los campos, tanto el de usuario como el de contraseña no están vacíos; en ese caso se produciría un error. Si los campos se han rellenado correctamente, se llama a la siguiente función, *encriptar_pwd*. Esta función se encarga de recibir por argumentos el usuario y la contraseña que el usuario de la aplicación está poniendo y encripta el *password*. Se ha optado por la encriptación de el *password* para dar más seguridad al sistema *web*. Una vez que se han resuelto estas dos funciones, en la función principal *Process_Login*, donde comienza todo el procedimiento, se modifican las variables globales del entorno APEX, asignándoles las del usuario que se ha registrado correctamente y lo redirige a la página de inicio del sistema *web*.

En la Figura 3.18 se puede ver el flujo que se sigue para la autenticación del usuario.

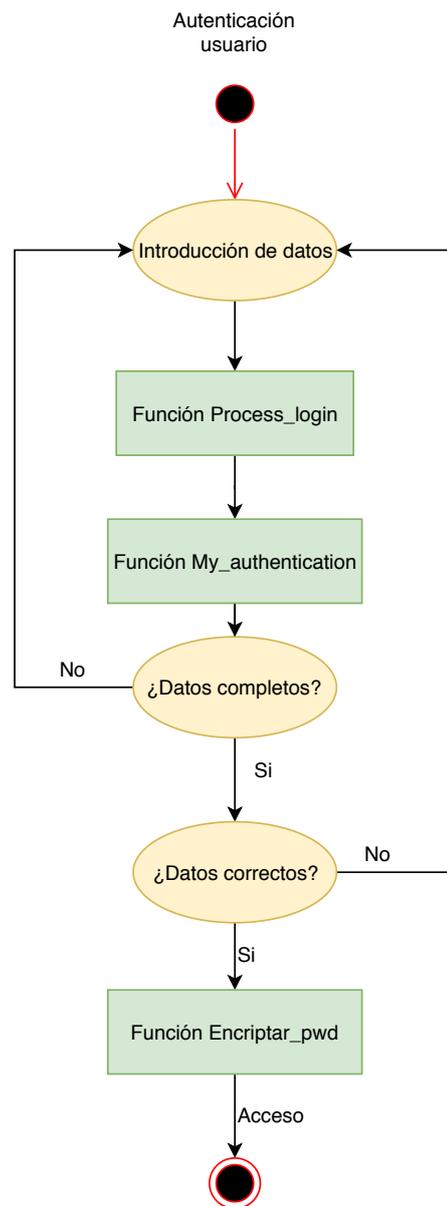


Figura 3.18: Flujo función de autenticación.

Módulo 3. Representación de entrenamientos

Una vez que el usuario ha podido autenticarse, éste tendrá acceso a todas las funcionalidades que ofrece el sistema *web*. Uno de los objetivos de este proyecto es hacer que la técnica de un deportista mejore con el uso del sistema que se está desarrollando en este proyecto. Para poder llegar a cumplir este objetivo es necesario guiar y ofrecer al usuario todas las facilidades posibles para conseguir sus metas, y ofrecerle una visualización correcta y lo más legible posible de toda la actividad deportiva realizada. Para ello, se ofrecerá al usuario la representación de los datos recogidos por el prototipo *hardware* empleado en este proyecto.

Como solución a esta problemática se han desarrollado dos funcionalidades principales que ayudarán al usuario a interpretar los datos recogidos por el prototipo *hardware*:

- **Representación de estadísticas de entrenamiento.**
- **Representación del golpeo del usuario.**

Funcionalidad 3.1. Representación de estadísticas de entrenamientos

Un entrenamiento de un deportista está compuesto por una serie de golpes. Estos golpes pueden ser de varios tipos. En el presente proyecto se tienen en cuenta los tres golpes principales del pádel, como son la derecha, el globo y la volea. En cada uno de estos golpes se miden una serie de datos que permiten posteriormente analizar el golpeo realizado. Para monitorizar los golpeos que realiza un deportista, se tienen en cuenta dos mediciones importantes a la hora de realizar el golpeo como son la aceleración durante el golpeo y el ángulo de la muñeca. Ambas mediciones se realizan en los tres ejes de coordenadas X,Y,Z.

La primera funcionalidad implementada en este módulo representará estadísticas relativas a cada entrenamiento realizado por el deportista.

En primer lugar se ofrece al usuario un calendario, donde podrá visualizar todos los entrenamientos realizados hasta la fecha. Se ofrece un calendario por colores, donde cada color representará un tipo de golpeo, quedando la clasificación de colores como se indica en la Tabla 3.8. De esta forma el usuario podrá valorar mediante el calendario la carga de trabajo y de entrenamientos que lleva realizados durante una semana o durante un mes. Esto será de ayuda para el deportista, porque una excesiva carga de trabajo sin el pertinente descanso puede acarrear lesiones deportivas, de esta manera el usuario podrá planificarse los entrenamientos y repartir la carga de trabajo según sus intereses.

Color	Golpe
Verde	Derecha
Rojo	Globo
Azul	Volea

Tabla 3.8: Tabla resumen de los tipos de golpeo representados en el calendario de entrenamientos.

El usuario podrá además visualizar estadísticas generales sobre el entrenamiento realizado. A continuación, se muestra un listado de las diferentes estadísticas que se mostrarán acerca de un entrenamiento:

- **Fecha de entrenamiento.** Fecha de ejecución del entrenamiento.
- **Tipo de entrenamiento.** Tipo de entrenamiento realizado. Entrenamiento de derecha, globo o volea.
- **Número de golpes.** Número de golpes totales realizados en el entrenamiento.
- **Estadísticas relativas a la aceleración de los golpes.**
 - Aceleración máxima X.
 - Aceleración máxima Y.
 - Aceleración máxima Z.

- Aceleración media X.
- Aceleración media Y.
- Aceleración media Z.

Como se ha dicho anteriormente, el usuario verá en el calendario los entrenamientos realizados hasta la fecha y podrá planificarse los entrenamientos posteriores para distribuir la carga de trabajo si así lo desea. También se ha incorporado una funcionalidad que permite comparar estadísticas de dos entrenamientos.

Con esto el usuario podrá determinar la carga de trabajo que se ha realizado en un entrenamiento u otro. Esto le ayudará a planificar también los siguientes entrenamientos, ya que si un entrenamiento ha tenido mucha carga de trabajo (se han realizado muchos golpes), el usuario decide no hacer un entrenamiento el siguiente día, sino que prefiere tomar descanso para prevenir lesiones. La funcionalidad ofrece, aparte de saber la carga de trabajo que se ha realizado, una comparativa de datos relativos a las estadísticas anteriormente descritas. Esta comparativa de estadísticas servirá al usuario a ver una diferencia entre las aceleraciones máximas, mínimas y las aceleraciones medias registradas en cada entrenamiento. Además de hacer una comparativa numérica, se hace uso de gráficos para representar esta comparativa. En la Figura 3.19 se puede ver un ejemplo de la comparación de la aceleración máxima en el eje X entre dos entrenamientos de globo. Para realizar esta comparación se han tomado como referencia los dos golpes que más aceleración han registrado en dos entrenamientos. Es decir, la barra roja representa el golpeo que más aceleración ha registrado en el entrenamiento 1, mientras que la barra verde, representa el golpeo que más aceleración ha registrado del entrenamiento 2. En el ejemplo de globo no es importante la aceleración máxima que se consigue a la hora de ejecutar el golpeo, ya que un golpe de globo no necesita demasiada aceleración, pero si necesario conocer su valor máximo porque una excesiva aceleración en un golpeo de este tipo no es correcta. Siempre se comparan entrenamientos del mismo tipo de golpeo, es decir, entrenamientos de globo con entrenamientos de globo, entrenamientos de derecha con entrenamientos de derecha y entrenamientos de volea con entrenamientos de volea. El eje X del gráfico (línea horizontal) representa el valor máximo de la aceleración registrado en cada uno de los dos entrenamientos, mientras que el eje Y del gráfico (línea vertical) representa la escala de valores de la aceleración desde 0 hasta 35000g.



Figura 3.19: Comparativa de la aceleración en el eje X de dos entrenamientos de globo.

Por último, el usuario tiene la posibilidad de subir vídeos acerca de cada uno de los golpes realizados. Esto permite al usuario ver reflejado la ejecución de sus golpes, para una posterior comparación y análisis en funcionalidades que se explicarán en las siguientes secciones.

Funcionalidad 3.2. Representación del golpeo del usuario

Para proporcionar al usuario un *feedback* adecuado acerca de todos sus golpes realizados en el entrenamiento, se ha optado por la representación de cada uno de los golpes mediante gráficas lineales.

Estas gráficas representan la ejecución completa del golpeo, desde que se parte de la posición de reposo hasta que nuevamente se vuelve a la posición de inicio del golpeo. Para la representación del golpeo se han utilizado seis gráficas para ayudar al usuario a interpretar sus golpes de un entrenamiento. Estas gráficas representan datos sobre dos aspectos importantes a la hora de realizar un golpeo de pelota, como son la aceleración y el ángulo de muñeca. Mostrar estos datos al usuario le ayudará a mejorar estos aspectos con respecto a la ejecución de sus golpes, pudiendo así ir mejorando su técnica. Tres de estas cinco gráficas representan la aceleración empleada durante el golpeo desde su comienzo hasta la finalización del mismo. La aceleración de un golpeo se representa en los tres ejes de coordenadas, X, Y y Z como se puede observar en las Figuras 3.20, 3.21, 3.22.

Las otras tres gráficas se encargan de representar el ángulo de inclinación en los tres ejes de coordenadas. En las Figuras 3.23, 3.24 y 3.25 se pueden observar las gráficas empleadas para representar dicha funcionalidad.



Figura 3.20: Gráfica aceleración en el eje X de un golpeo de derecha.

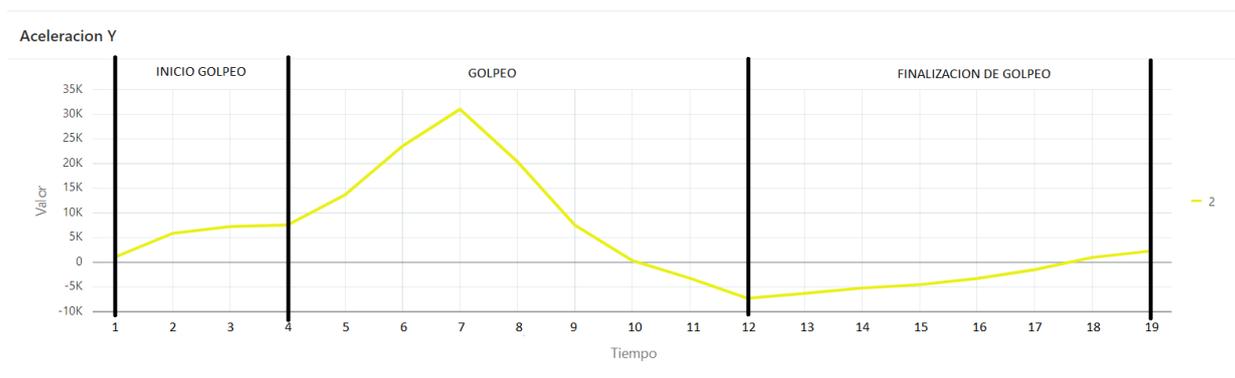


Figura 3.21: Gráfica aceleración en el eje Y de un golpeo de derecha.

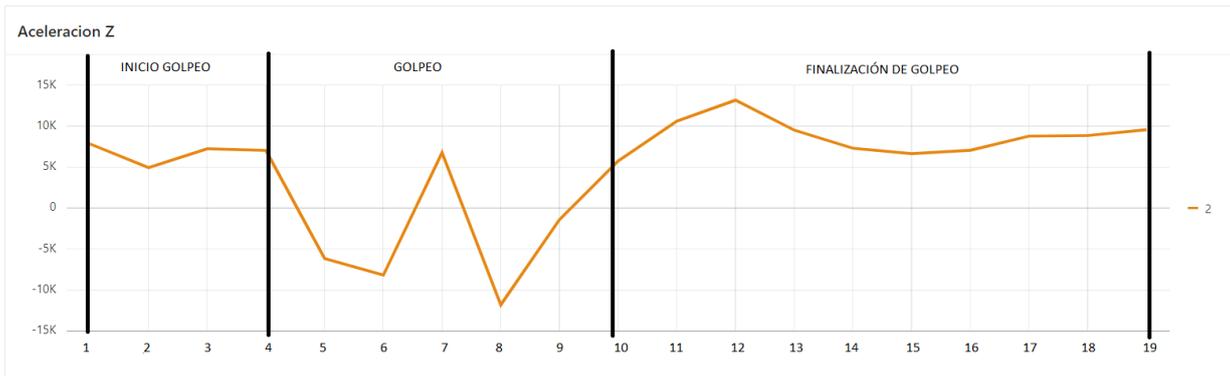


Figura 3.22: Gráfica aceleración en el eje Z de un golpe de derecha.

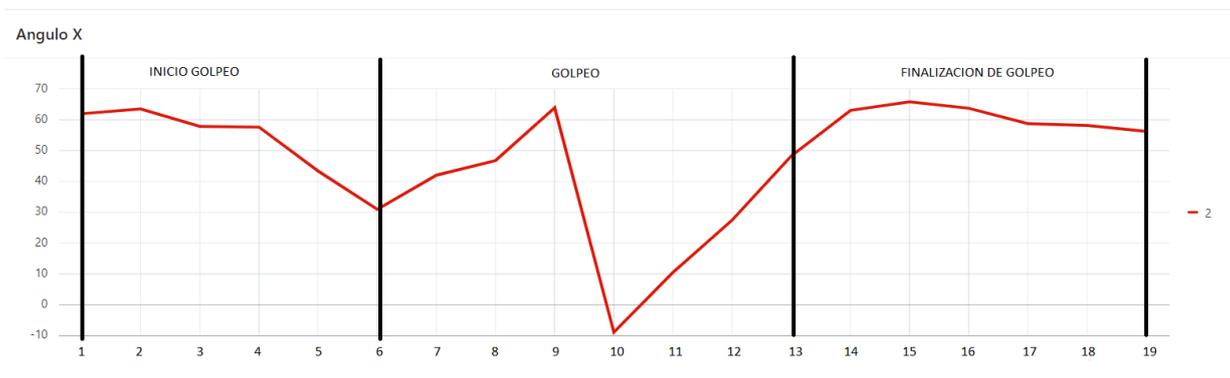


Figura 3.23: Gráfica ángulo en el eje X de un golpe de globo.

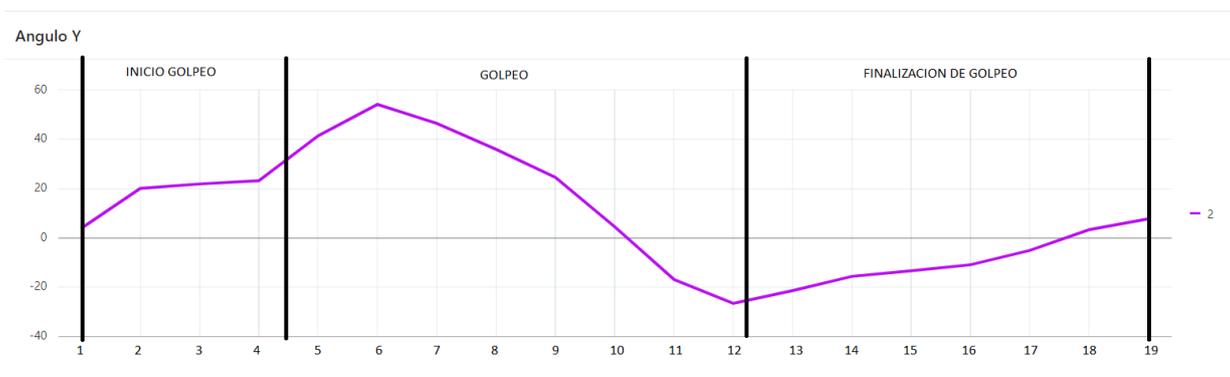


Figura 3.24: Gráfica ángulo en el eje Y de un golpe de globo.

En cada una de estas gráficas se representan los valores de la aceleración y del ángulo durante la realización de todo el golpeo. En las gráficas el eje X (horizontal) representa la línea temporal, es decir, todo el transcurso del golpeo y el eje Y (vertical) cada uno de los valores de la aceleración y del ángulo que va tomando en cada instante de tiempo. Como se observa en las gráficas, solo se representa un golpe para dar más claridad a la representación de los golpes, ya que si se representaran todos a la vez, dificultarían la interpretación por parte del usuario. No obstante, el usuario podrá ir cambiando la visualización de cada golpeo para analizar todos los efectuados en el entrenamiento.

En el pádel existen tres etapas importantes durante la ejecución de un golpeo. Estas etapas se corresponden con la posición inicial, la ejecución del golpeo y la terminación del mismo. En las gráficas se puede diferenciar estas tres etapas. Para distinguir estas partes se ha tomado como

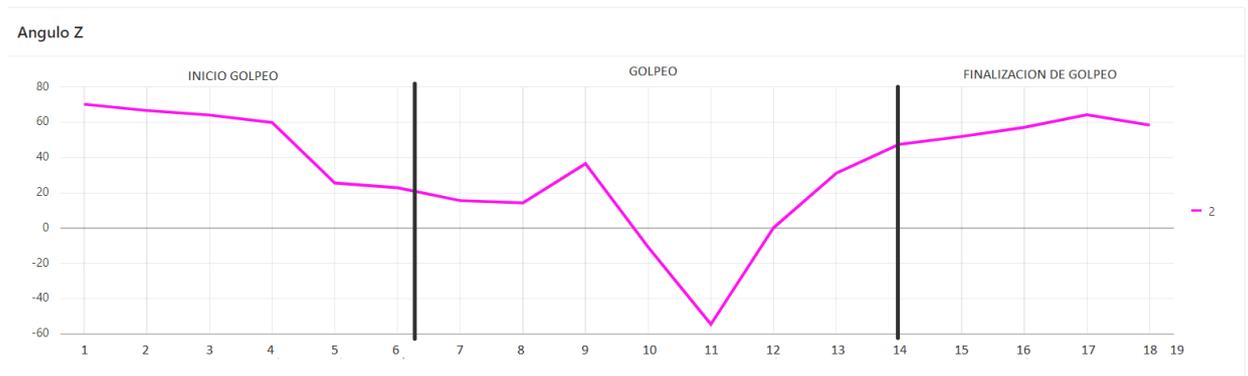


Figura 3.25: Gráfica ángulo en el eje Z de un golpe de globo.

referencia cuando hay un cambio brusco en la aceleración. En este deporte, cuando se acelera bruscamente el brazo es consecuencia de que se va a realizar el golpeo, por ello en la gráfica se representa esta diferencia de aceleración como la ejecución del golpeo. Como se puede observar cuando se finaliza el golpeo los valores vuelven a tomar los mismos valores que cuando se inició el golpeo, lo que es señal de finalización de golpeo.

Diferenciando estas tres partes en la gráfica el usuario podrá ser capaz de visualizar qué parte es la que debe mejorar y en qué parte es en la que mejor técnica emplea. Así el usuario podrá realizar un análisis más profundo sobre las gráficas aportadas.

Módulo 4. Comparación de golpes

Una vez realizada la representación de los golpes de manera gráfica, se ha optado por la implementación de dos funcionalidades de comparación de golpes para ofrecer al usuario una mejor interpretación de los datos recogidos referentes a sus golpes. Con esto el usuario podrá interpretar de mejor manera sus golpes y sacar las conclusiones pertinentes a cada golpeo que ha realizado.

A continuación se exponen las dos funcionalidades implementadas para la comparación de golpes:

- Comparación de golpeo mediante vídeos.
- Comparación de golpeo con otros usuarios.

Funcionalidad 4.1. Comparación de golpeo mediante vídeos

Como se indicaba en la sección anterior de representación de estadísticas de entrenamientos, se ha implementado una funcionalidad que permite al usuario subir vídeos de cada uno de sus golpes de un entrenamiento. Estos vídeos son utilizados posteriormente para la comparación con otro vídeo modelo. En este vídeo modelo se expone la ejecución del mismo tipo de golpeo que realiza el usuario, pero ejecutado por un jugador profesional. Con esto el usuario podrá realizar una comparativa de la ejecución de su golpeo con el de un jugador profesional, obteniendo una referencia real de cómo efectuar su golpeo en próximos entrenamientos.

En la Figura 3.26 se puede ver una representación de esta funcionalidad expuesta. Se puede observar a la izquierda el golpeo efectuado por uno de los jugadores que están en un entrenamiento y a la derecha el golpe efectuado por un jugador profesional.

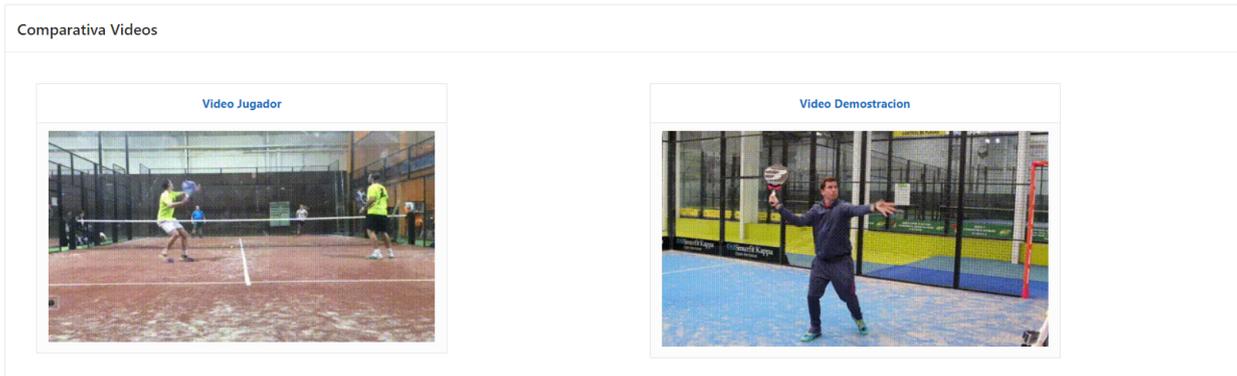


Figura 3.26: Comparación de vídeos entre usuario y profesional

4.2. Comparación de golpeo con otros usuarios

Anteriormente se ha expuesto cómo se representan los golpes de un entrenamiento de un usuario mediante gráficas. En las gráficas empleadas solo se representa un golpe realizado, para tener más claridad a la hora de interpretar las gráficas. El usuario podrá cambiar el golpeo que está visualizando en ese momento para hacer un análisis de todos los golpes realizados en un entrenamiento. Para obtener un mayor *feedback* y que el usuario pueda hacer un mejor análisis e interpretación de todos sus golpes, el sistema ofrece una funcionalidad de comparar los golpes del usuario con golpes realizados por jugadores profesionales del mundo del pádel. Esta comparativa se hace también mediante gráficas lineales, en las que se puede observar tanto el golpeo del usuario como el del jugador profesional. A la hora de comparar los golpes del usuario con un jugador profesional, se considera también a un monitor de pádel como jugador profesional. Además, el sistema que se está desarrollando puede ser utilizado por un monitor de pádel para realizar un seguimiento a sus alumnos y ver que progreso están realizando. En la Figura 3.27 se puede observar un ejemplo de esta comparación. En dicha figura, se puede observar que la aceleración empleada por el usuario (gráfica izquierda) en el eje Y, es bastante similar a la utilizada por el jugador profesional (gráfica derecha), con esto el usuario podrá interpretar que con respecto a este eje su golpeo se ha realizado correctamente. Sin embargo, se puede observar en la Figura 3.28 que la aceleración empleada en el eje X no es del todo correcta, con lo que el usuario deberá mejorar la aceleración en ese eje para mejorar su golpeo.



Figura 3.27: Comparativa aceleración en el eje Y de un golpe de derecha.

Si nos centramos en la Figura 3.28 donde se muestra la aceleración del eje X, las gráficas del usuario y del deportista son poco similares. Se puede observar que el usuario a la hora de realizar el golpeo, impone más aceleración a la hora de golpear la pelota. El jugador profesional, efectúa



Figura 3.28: Comparativa aceleración en el eje X de un golpe de derecha.

su golpeo con una aceleración máxima de 23000 g, mientras que el usuario realiza el golpeo con una aceleración máxima de 29000 g. Esto se puede traducir en una excesiva aceleración y por lo tanto en un golpeo mal efectuado. No siempre es necesario obtener la aceleración más alta a la hora de efectuar un golpeo, ya que en muchas ocasiones se busca la colocación y no la fuerza para efectuar un buen golpeo. Dependiendo de la situación de juegos, en ocasiones es preferible realizar un golpeo de control con una terminación más extendida que realizar un golpeo con una aceleración muy alta.

Sin embargo, podemos ver que la desaceleración tras el golpeo de la pelota y la finalización del golpeo si que tienen similitud, ya que en ambas partes se registran valores similares a la hora de finalizar el golpeo. El usuario efectúa correctamente la desaceleración hasta 0 g aproximadamente al igual que jugador profesional. Igualmente, el usuario vuelve a tener una aceleración en la finalización del golpeo hasta alcanzar los 17000 g que se aproximan a los 20000 g que registra el jugador profesional.

Tras este análisis, se podría determinar que el usuario debe mejorar la aceleración de su golpeo en el eje X ya que ha sido demasiada aceleración la que se ha empleado. Esto es debido a que a la hora de ejecutar el golpeo, el usuario ha imprimido demasiada fuerza para realizar el golpeo. Se puede observar que la desaceleración (la terminación del golpeo) si se ha efectuado correctamente, por lo que se puede determinar que se tiene que mejorar a la hora de golpear a la pelota. Además, si el jugador quiere seguir perfeccionando aún más, podría mejorar la aceleración de finalización y la finalización de golpeo para que se ajuste más a la del jugador profesional.

Funcionalidad 4.3. Visualización de tutoriales

En el presente proyecto se ofrece la posibilidad de visualizar vídeos de jugadores profesionales para tener como referencia un modelo de golpeo en el que fijarse. Hay jugadores que nunca han practicado este deporte y quieren comenzar a iniciarse en él. Otros ya saben lo que es este deporte, pero quieren repasar conceptos sobre cómo efectuar un golpeo. Para ello, se ha desarrollado una funcionalidad en el sistema *web* que permite la visualización de videotutoriales donde los jugadores profesionales enseñan y dan consejos sobre como efectuar un golpeo.

Estos vídeos serán reproducidos directamente en el sistema *web*, ya que están almacenados dentro del mismo. Para la elección de los videotutoriales se ha tenido en cuenta los tipos de golpes que se realizan en un entrenamiento, por ello, se dispone en el sistema *web* de videotutoriales de golpeo de derecha, globo y volea.

Esta funcionalidad permitirá a los usuarios que se inicien en el deporte, a tener una visión general de como efectuar un golpeo, así sabrán como comenzar en este deporte. Para los jugadores ya iniciados, esta funcionalidad les aportará algunos consejos sobre cómo efectuar y mejorar el

golpeo que están realizando para conseguir el objetivo de mejorar su técnica.

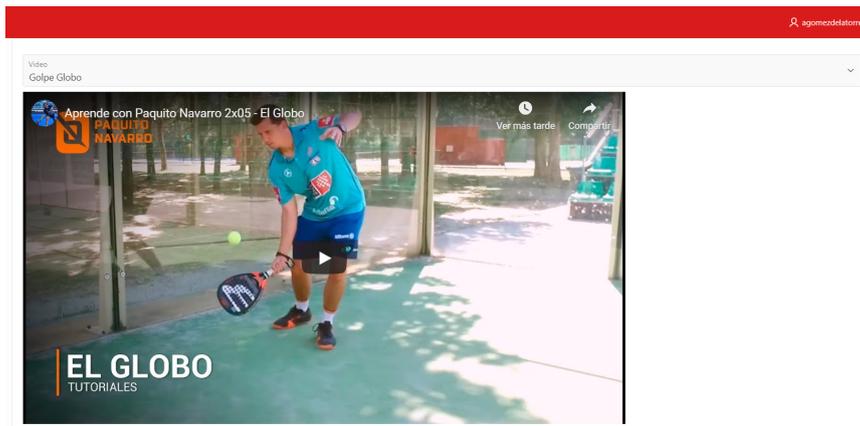


Figura 3.29: Videotutorial ofrecido a los usuarios de un golpeo de globo.

Módulo 5. Evolución y seguimiento de mejora de técnica de golpeo

El objetivo general de este proyecto es que el deportista vaya mejorando su técnica de golpeo con el uso de este sistema desarrollado en el presente proyecto. El deportista debe de ver una evolución y una mejoría de su técnica con el paso de los entrenamientos realizados y el uso de dicho sistema *web*. Para hacer un seguimiento y proponer algunas metas a cumplir para el deportista, se han implementado dos funcionalidades para la consecución de esta tarea.

Las dos tareas que se han implementado en este módulo sirven tanto para mejorar la técnica del deportista, como para motivar al deportista a seguir practicando el deporte y seguir usando el sistema que se está desarrollando en este proyecto. Es importante incentivar a un deportista a seguir practicando este deporte, para ello se proponen dos funcionalidades, una basada en objetivos y otra basada en la obtención de puntuación mediante retos. Con la funcionalidad de objetivos, haremos un seguimiento de la evolución de la técnica del deportista. El deportista creará objetivos en función de los datos y los resultados de sus entrenamientos. Estos objetivos se crearán para superar una determinada marca que estipule el usuario con respecto a la aceleración empleada o al ángulo que se toma en la realización del golpeo. Con la consecución de los objetivos, el usuario irá mejorando su técnica e incrementando su nivel de juego.

Con la funcionalidad de los retos, se pretende incentivar al usuario a seguir usando el sistema *web* y así seguir practicando este deporte. Los retos están diseñados como pequeñas metas que debe ir consiguiendo el usuario con el uso del sistema *web* y con la práctica de entrenamientos. A diferencia de los objetivos, los cuales se basan en la superación de metas con respecto a la aceleración y ángulo de golpeo, los retos se basan en la consecución de realizar X entrenamientos, superar X número de objetivos, realizar X número de golpes, etc. Estos retos solo se centran en estadísticas de juego, mientras que los objetivos se centran en la consecución de objetivos específicos de golpeo.

A continuación se explica en detalle las dos funcionalidades descritas.

La finalidad de los objetivos es la mejora de la técnica de golpeo, mientras que la finalidad de los retos es incentivar al usuario a seguir utilizando el sistema *web*.

La **primera** funcionalidad está basada en objetivos. Se plantea un método por el cual el deportista podrá establecer una serie de objetivos que desea cumplir a lo largo de sus entrenamientos.

Estos objetivos estarán centrados en la aceleración de los diferentes ejes y en ángulo empleado en el golpeo de la pelota. El deportista no definirá una fecha límite para cumplir el objetivo, solo se establecerá la fecha de creación de objetivo y se actualizará en el momento que el sistema registre que ese objetivo ha sido cumplido. Para que el objetivo pueda pasar de un estado de “pendiente” a “conseguido”, el sistema deberá registrar en algún golpeo de un entrenamiento posterior a la fecha de creación del objetivo, algún valor que indique que ese objetivo ha sido conseguido.

Se han representado todos los objetivos en un calendario para ofrecer al usuario un pequeño seguimiento de los objetivos que ha conseguido y los que le faltan por conseguir. En la Figura 3.30 se puede observar dicho calendario. Además de dicho calendario, se ofrece un listado de todos los objetivos, tanto los cumplidos como los que faltan por cumplir. En la Figura 3.31 se observa el listado de objetivos “pendientes” y “conseguidos” referentes a la aceleración en el eje X.

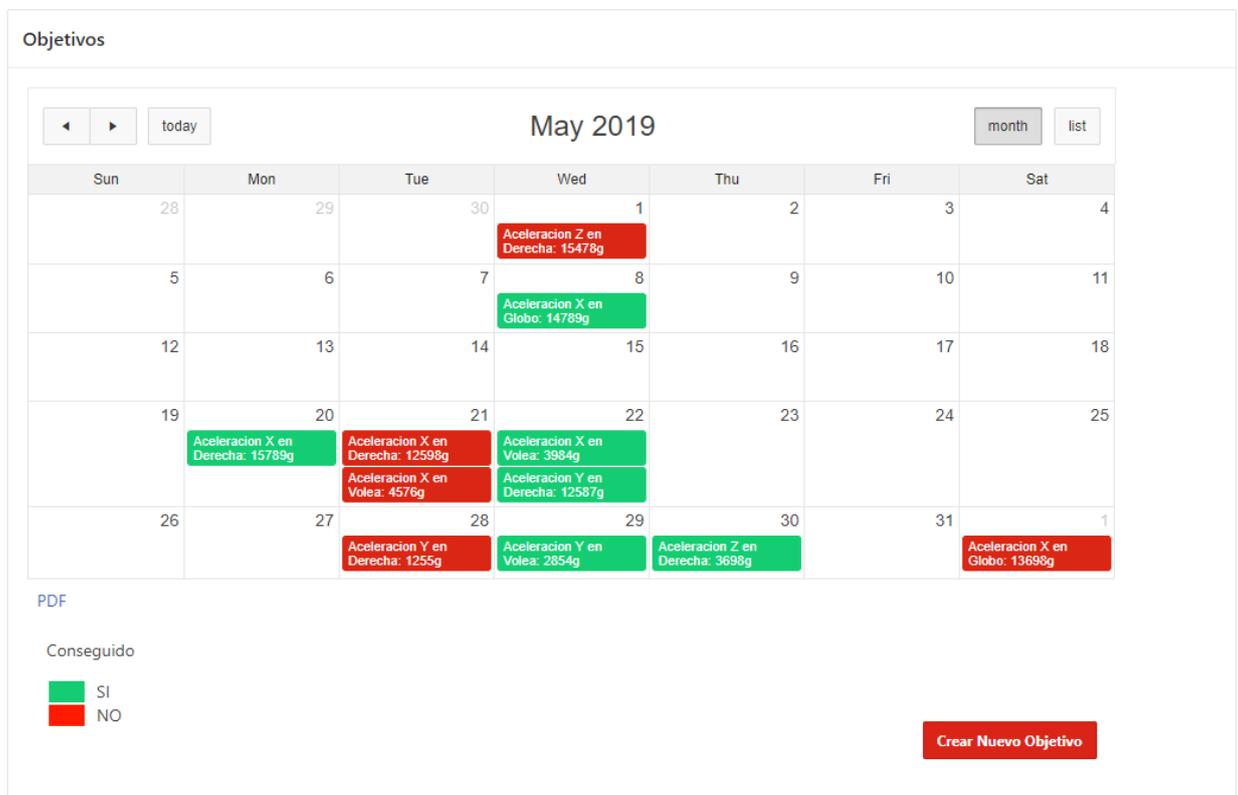


Figura 3.30: Ejemplo de calendario de objetivos.

Objetivos Derecha			
AG	Alvaro 21-AUG-19	No Conseguido	Aceleracion X 9999
AG	Alvaro 28-MAY-19	No Conseguido	Aceleracion Y 1255
AG	Alvaro 21-MAY-19	No Conseguido	Aceleracion X 12598
AG	Alvaro 01-MAY-19	No Conseguido	Aceleracion Z 15478
AG	Alvaro 01-JUN-19	Conseguido	Aceleracion Z 3698
AG	Alvaro 02-JUN-19	Conseguido	Aceleracion X 15789
AG	Alvaro 03-JUN-19	Conseguido	Aceleracion Y 12587

Figura 3.31: Ejemplo de listado de objetivos.

En el gráfico de la Figura 3.32 se resume toda la funcionalidad descrita anteriormente. El usuario visualiza los datos recogidos de su entrenamiento e interpreta dichos datos. Haciendo un análisis, como se ha explicado en la sección de visualización del golpeo, el usuario sabe que tiene que mejorar de cara al próximo entrenamiento. Para esto el usuario se establece un objetivo ha conseguir. Este objetivo es registrado por el sistema *web* con la fecha de creación de dicho objetivo. Una vez que el sistema *web* registra algún valor que supere el objetivo, este pasará a un estado de conseguido y se actualizará su fecha de obtención de dicho objetivo.

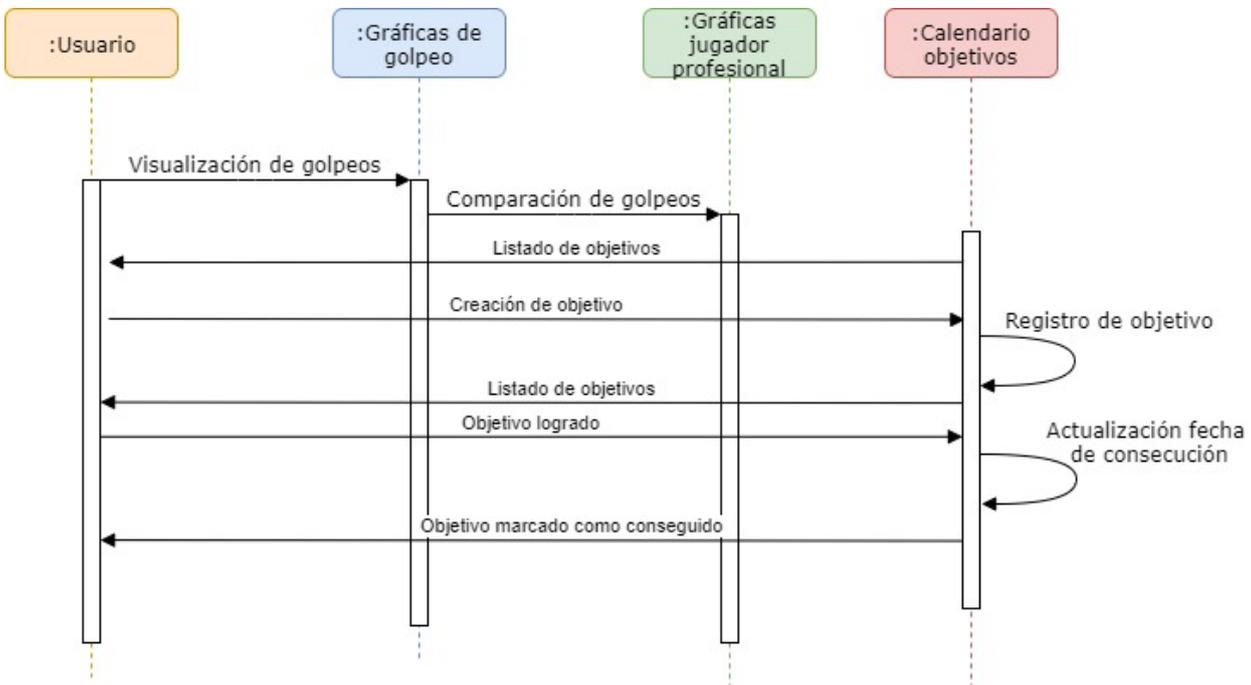


Figura 3.32: Secuencia de creación de objetivos.

La **segunda** funcionalidad ofrecida para el seguimiento y la evolución de la mejora de la técnica empleada en el golpeo, se basa en la obtención de puntuación. Para el desarrollo de esta funcionalidad se han establecido unos retos basados en puntuación. Dichos retos se han calificado según su nivel de dificultad, lo que conlleva la creación de tres niveles de dificultad donde se recogen estos retos. Los tres niveles que se han considerado para que el usuario supere y vaya incrementando su nivel son los siguientes:

- **Nivel 1.** Nivel básico.
- **Nivel 2.** Nivel intermedio.
- **Nivel 3.** Nivel experto.

El usuario tendrá que ir superando cada uno de estos retos para ir incrementando su nivel. Estos retos ayudarán al usuario a marcarse metas, lo que incitará al usuario a seguir entrenando y practicando sus golpesos, con lo que conseguirá una mejora de su técnica.

En la Figura 3.33 se recoge un resumen de los puntos totales que se pueden obtener en cada nivel y de los puntos restantes que le quedan por conseguir al usuario.

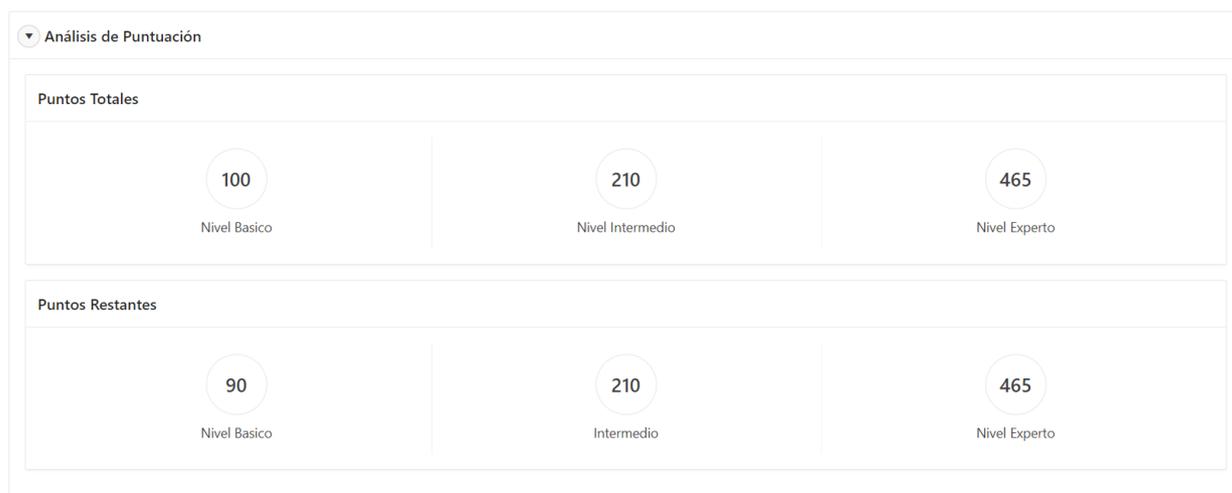


Figura 3.33: Análisis de puntuación total que se puede obtener y de la puntuación restante de un usuario.

3.3.2. Capa de persistencia: Almacenamiento de datos

El sistema *web* está diseñado para que sea utilizado por varios usuarios al mismo tiempo. Esto implica que los usuarios querrán consultar y guardar sus datos al mismo tiempo. Para ello se ha diseñado y desarrollado una capa de persistencia donde los datos se guardan de forma organizada. Es necesario que el sistema disponga de esta capa de persistencia donde se pueda almacenar todos los datos que los usuarios deseen y que luego estén disponibles para su consulta. Para el desarrollo de esta capa de persistencia se ha hecho uso de un sistema gestor de base de datos.

Un Sistema Gestor de Base de Datos (SGBD) consiste en una colección de datos interrelacionados. El objetivo principal de un SGBD es proveer un entorno que sea conveniente y eficiente para usar recibiendo y almacenando información en una base de datos.

Los sistemas de base de datos están diseñados para manipular grandes cantidades de información. El sistema de base de datos debe asegurar la seguridad de la información almacenada, a pesar de que el sistema pueda tener algún problema o un acceso no autorizado. Si los datos están compartidos entre algunos usuarios, el sistema debe mantener la consistencia. Las operaciones que se ejecutan sobre la base de datos para así poder proporcionar al usuario un resultado en función de sus peticiones, se realiza de un modo eficiente y seguro.

El SGDB utilizado en este proyecto debe cumplir las siguientes funciones principales:

- **Definición de datos:** el SGDB ha de poder definir objetos de la base de datos a partir de definiciones en código fuente para transformarlas en una versión objeto.
- **Manipulación de datos:** el sistema debe dar respuesta a las peticiones de los usuarios, como eliminar, actualizar, solicitar datos, etc. El manejo de estos datos debe ser lo más eficiente posible, ya que la manipulación de datos continua por muchos usuario, se traduce en una alta solicitud de rendimiento.
- **Seguridad e integridad de los datos:** el SGDB debe de registrar todo tipo de uso de la base de datos, con cualquier tipo de petición debe aplicar la normativa establecida para cumplir con los requisitos de seguridad e integridad de los datos almacenados en la base de datos. A su vez, ha de garantizar su seguridad ante cualquier posible ataque y no permitir el acceso de los usuarios no deseados.
- **Recuperación y restauración de los datos:** ante un posible fallo, el sistema debe garantizar que sea posible recuperar y restaurar los datos.

Los criterios para elegir el SGDB adecuado deben tener en cuenta que el sistema está o pueda estar integrado con otro *software*, que sea escalable y adaptable a un posible crecimiento del sistema, que sea sostenible y rentable. Por otra parte, tiene gran importancia la seguridad de los datos almacenados, la funcionalidad que nos permite cumplir con los requisitos y necesidades, y la facilidad de uso para el usuario.

En base a estas pautas, en el presente proyecto se hace uso del SGDB de Oracle Database [14], concretamente la versión 18.4.0.0, una de las versiones mas recientes que existen de este *software*. Oracle Database es un sistema de gestión de base de datos relacional de tipo objeto-relacional, este tipo es una extensión de la base de datos relacional tradicional el cual tiene las características de la programación orientada a objetos.

Se considera Oracle como uno de los sistemas de bases de datos más completos. Sus características a destacar son:

- Soporte de transacciones.
- Garantiza una gran escalabilidad.
- Permite crear sistemas escalables.
- Es multiplataforma.

Para el desarrollo del sistema *web* se ha optado por la herramienta de *Oracle Application Express* (APEX). Esta herramienta es una plataforma de desarrollo que permite desarrollar aplicaciones seguras, escalables y que se pueden desplegar en cualquier lugar y ofrecen funciones de primera

clase. Esta herramienta ha sido diseñada a partir del *software* de base de datos Oracle. Esta herramienta nos permite desarrollar páginas-plataformas *web* de manera profesional y con una amplia gama de opciones.

Para el despliegue de la aplicación, se ha optado por el uso del protocolo cliente-servidor *REST*. Esto nos permite la creación de un servicio para el uso de todos los usuarios. En el caso de Oracle, la *API* que implementa este servicio se denomina *ORDS* (*Oracle REST Data Services*) [15].

Oracle REST Data Services (*ORDS*) es un servicio de datos que se utiliza para reemplazar el servidor *HTTP* de Oracle. Se basa en *Java EE* que proporciona un servicio *REST* y aumenta la capacidad de seguridad. Se puede implementar en *WebLogic*, *Tomcat*, *Glassfish* o en cualquier servidor.

En la Figura 3.34 se muestra el esquema general de como funciona el servicio *REST* en Oracle. Este servicio es desplegado en el servidor donde está montado el SGDB y es capaz de interactuar entre el usuario y la base de datos. El servicio *REST* da la capacidad de desplegar un sistema y que los usuarios puedan conectarse a él.

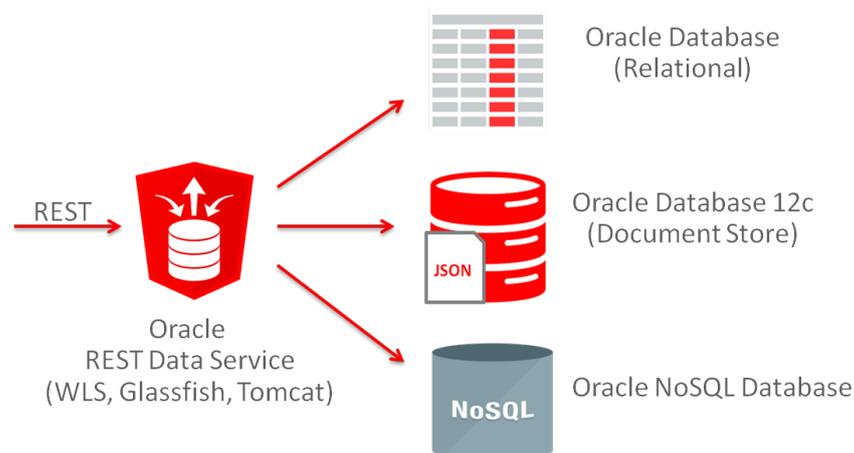


Figura 3.34: Servicios *REST* Oracle.

Para la interacción de los servicios *REST* con *ORACLE APEX*, hay dos formas de llevarlo a cabo:

- Método 1: configurar una puerta de enlace *pl/sql* que puede ejecutarse en *Oracle XML DB HTTP Server*, esto es un *HTTP* que está disponible cuando instala Oracle 11g o superior.
- Método 2: instalando *ORACLE APEX* en un servidor específico.

En el presente proyecto se ha optado por el método 2, ya que se usa *Oracle Linux Server* como servidor para la instalación de base de datos y la instalación de *ORACLE APEX*.

En la Figura 3.35 se observa la comunicación entre los servicios *REST* y la herramienta de desarrollo *ORACLE APEX*. En la imagen se puede observar como el usuario se conecta desde su ordenador personal al servicio *REST* y éste a su vez se comunica con *APEX*. Pero en realidad lo que está haciendo es comunicarse con la base de datos, ya que *APEX* es un *framework* de desarrollo para aplicaciones que está dentro del *software* de base de datos.

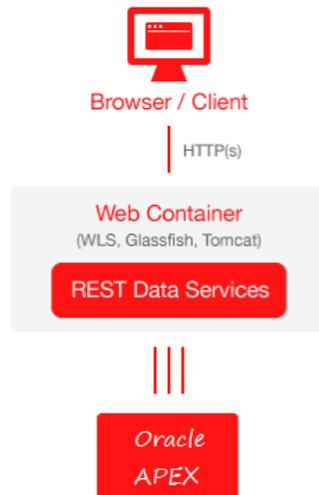


Figura 3.35: Comunicación ORACLE APEX con servicios REST.

Modelo relacional de datos

Para la organización de la información almacenada en la base de datos utilizada en el presente proyecto, se ha optado por el modelo de datos relacional mostrado en la Figura 3.38. Para una mayor apreciación del esquema relacional, se ha dividido en dos figuras más amplias, siendo éstas las Figuras 3.36 y 3.37.

La función principal de la base de datos es almacenar todos los datos registrados por el prototipo *hardware*. Estos datos giran entorno a un usuario ya que es el usuario quien realiza los entrenamientos y los golpesos, por ello se ha diseñado una tabla usuario donde se recogen todos los datos correspondientes al usuario que va a utilizar el sistema *web* y que va a realizar los entrenamientos. Los entrenamientos que realizan un usuario, solo son de un usuario, por ello se ha creado una tabla de entrenamientos para guardar los entrenamientos que realiza un usuario. Para distinguir y diferenciar qué entrenamientos tiene un usuario, se ha creado un campo en la tabla de entrenamientos llamado “usuario”. Este campo hace referencia al usuario de la tabla “USUARIO”.

Como se ha comentado en secciones anteriores, los entrenamientos pueden ser de diferentes tipos: de derecha, de globo o de volea. Para hacer una distinción de los tipos de entrenamientos que existen, se ha creado una tabla en la que se registran estos tipos. La tabla de entrenamientos hace referencia a esta tabla para saber de qué tipo es el entrenamiento que se ha guardado.

Un entrenamiento está constituido por los diferentes golpesos que se llevan a cabo en la actividad física que realiza el deportista. Estos golpesos también son de un tipo, pero un golpeo pertenece a un entrenamiento, por ello si estamos realizando un entrenamiento de volea, todos los golpesos que se realicen serán de volea, ya que el entrenamiento que estamos realizando es de volea. Lo mismo ocurre con los demás tipos de entrenamientos. Para guardar los diferentes golpesos que se realizan en un entrenamiento, se ha creado una tabla de golpesos, donde se guardan todos los golpesos de un entrenamiento. Esta tabla lleva un campo en la cual lleva asociado el “ID” de entrenamiento que se está realizando, es decir, lleva un identificador el identificador del entrenamiento para saber que ese golpeo pertenece a un entrenamiento.

Como se ha explicado en la secciones anteriores, el usuario puede subir vídeos de cualquier golpeo realizado en un entrenamiento. Para guardar los vídeos que el usuario desee subir sobre sus entrenamientos, se ha creado una tabla vídeos donde se almacena dicha información. Esta

tabla contiene un campo que hace referencia al entrenamiento sobre el que se quiere subir el vídeo. Además de ello cuenta con un campo para especificarle, si se desea, el golpe al que va asociado el vídeo.

El usuario tendrá la posibilidad de crear y conseguir objetivos en función de los datos que ha obtenido y la práctica que realice posteriormente para la consecución de éstos. Para almacenar esta información se ha creado una tabla objetivos, en la cual se almacenan todos los objetivos de un usuario. Esta tabla tiene un campo usuario que está asociado a la tabla de usuario, esto es para diferenciar que objetivos tiene cada usuario. Los objetivos pueden ser de varios tipos ya que se pueden crear para la aceleración en cualquiera de los ejes y para el ángulo en los ejes X, Y, Z. Para hacer esta diferencia se ha creado una tabla que registra el tipo de objetivo que se va a crear.

Otra funcionalidad implementada es la obtención de puntuación mediante la consecución de una serie de retos que se plantean en el sistema *web*. Para registrar los datos de esta funcionalidad se ha optado por la creación de tres tablas. La primera tabla de ellas establece el nivel de puntuación al que pertenece cada reto creado. Seguidamente se ha creado la tabla puntos, donde se registra el listado de todos los retos que el usuario debe ir superando para la obtención de puntos. Y, finalmente, se ha creado una tabla que registra la puntuación de cada usuario.

Tablas empleadas en el esquema:

- **AUXILIAR**: tabla utilizada para la carga de datos.
- **USUARIO**: tabla principal que registra los datos correspondientes a todos los usuarios.
- **ENTRENAMIENTO**: en ella se registran todos los entrenamientos de todos los usuarios del sistema.
- **TIPO_ENTRENAMIENTO**: registra los tipos de entrenamientos que se implementan en el presente proyecto.
- **GOLPE**: registra todos los golpes realizados en todos los entrenamientos de los usuarios.
- **VIDEO_TUTORIALES**: almacenan los videotutoriales que servirán de referencia a los usuarios.
- **VIDEOS**: esta tabla almacena los vídeos que los usuarios suben sobre sus entrenamientos.
- **OBJETIVO**: registra todos los objetivos de los usuarios.
- **TIPO_VALOR_OBJETIVO**: se almacenan la descripción de los tipos de objetivo que existen.
- **PUNTOS**: descripción de los diferentes retos a conseguir por el usuario.
- **NIVEL_PUNTUACION**: tipos de niveles que se han establecido para los retos.
- **PUNTUACION_USUARIO**: puntuación que tiene cada usuario.
- **IMAGENES**: almacenamiento de imágenes de perfil de los usuarios.
- **GOLPE_MODELO**: almacenamiento de los golpes efectuados por los jugadores profesionales.

El diseño de este modelo de datos se ha diseñado de forma escalable para la fácil recuperación de información por parte de los usuarios y para el mantenimiento futuro del sistema *web*.

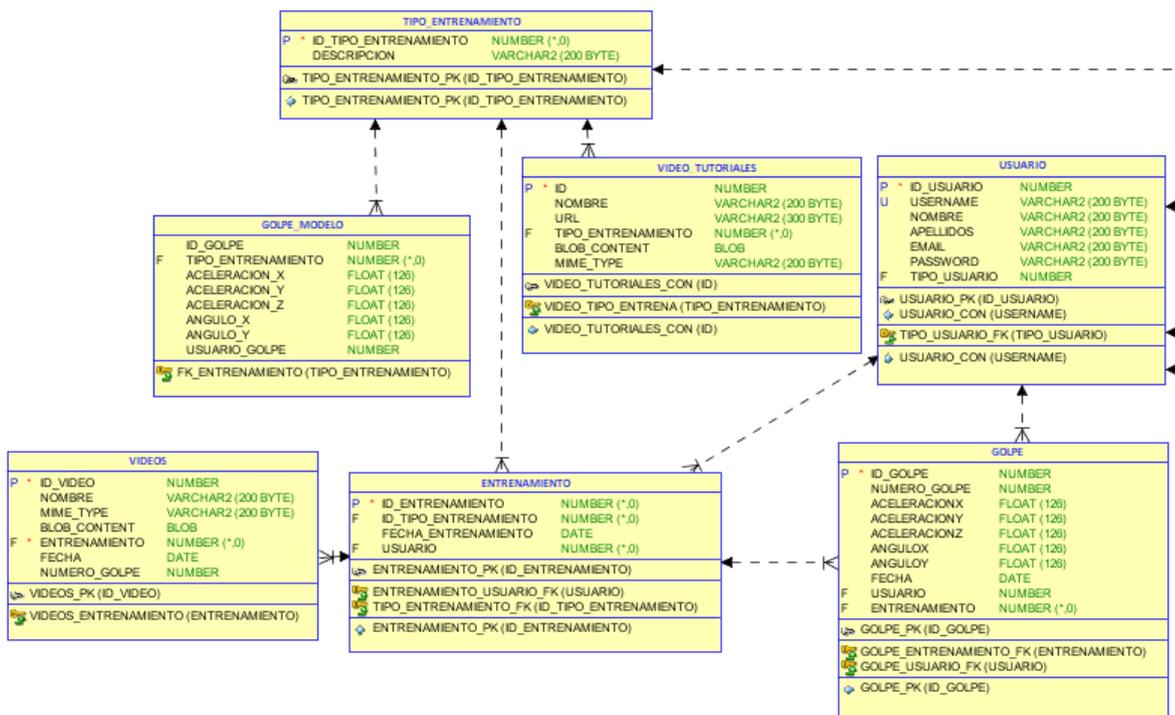


Figura 3.36: Modelo de datos relacional parte 1.

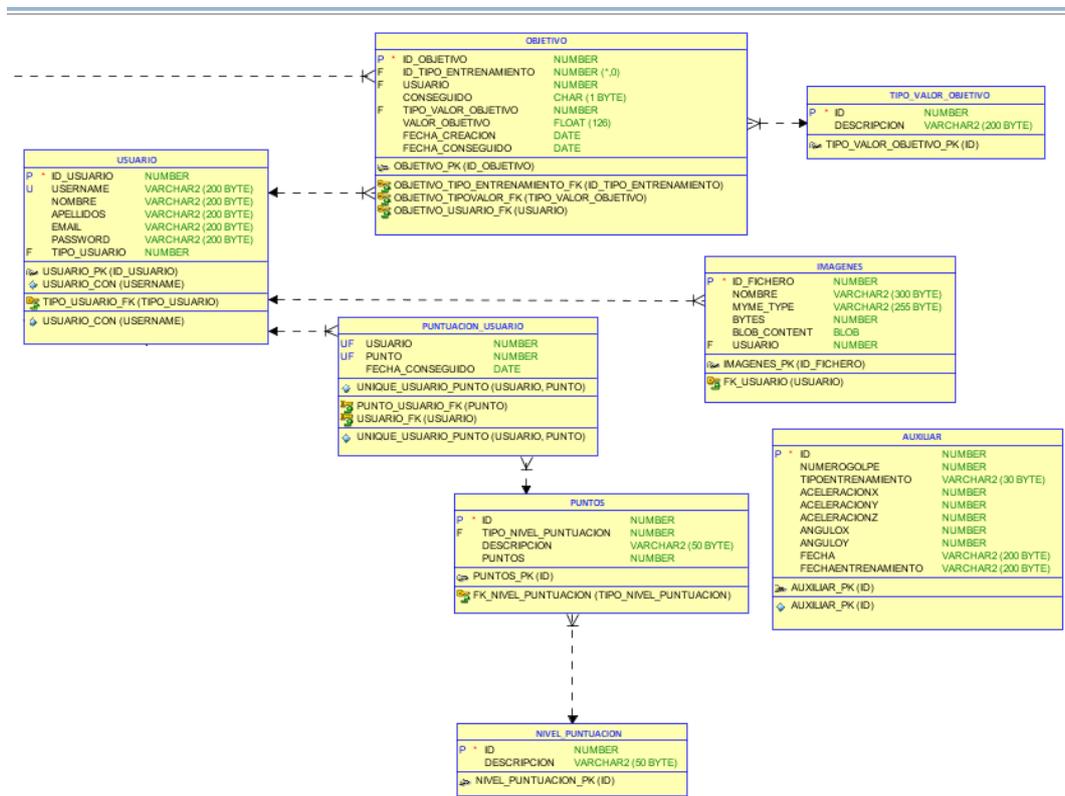


Figura 3.37: Modelo de datos relacional parte 2.

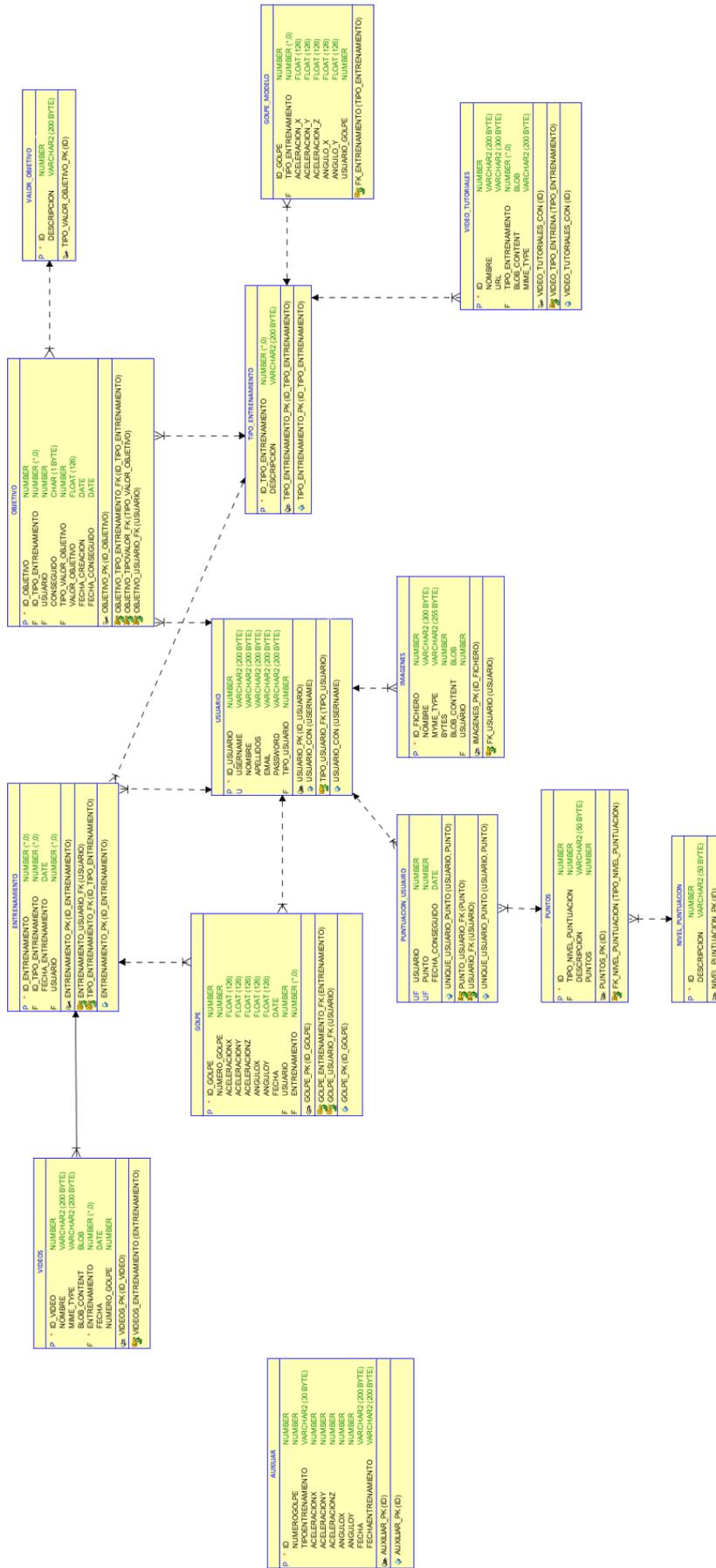


Figura 3.38: Modelo de datos relacional.

3.4. ORGANIZACIÓN DEL PROYECTO EN PAQUETES DE TRABAJO

Para la consecución del presente proyecto se definen un total de tres paquetes de trabajo (PT), cada uno de los cuales está dividido en una serie de Tareas (T). A continuación se exponen los paquetes de trabajo con sus respectivas tareas asociadas. A su vez, en el diagrama de Gantt de la Figura 3.39, se puede observar la descomposición temporal del proyecto en paquetes de trabajo y tareas.

- **PT1 - Diseño y desarrollo del prototipo hardware.**
 - T1.1. Estudio funcionamiento de placa arduino, sensores y módulos de almacenamiento.
 - T1.2. Implementación de ejemplos básicos y procesamiento de datos con sensores y módulos de almacenamiento.
 - T1.3. Integración de sensores y módulos en prototipo *hardware*.
 - T1.4. Pruebas y validación del prototipo *hardware*.
 - T1.5. Diseño de fijación del dispositivo a muñeca del deportista.
 - T1.6. Implantación y validación de correa de cuero.
- **PT2 - Diseño de interfaz entre el dispositivo de medición y el deportista.**
 - T2.1. Estudio y valoración de diferentes componentes para la introducción de ellos en el prototipo *hardware*.
 - T2.2. Diseño de la interfaz entre el usuario y el prototipo *hardware*.
 - T2.3. Implantación de componentes en la interfaz.
 - T2.4. Validación y pruebas de la interfaz con nuevos componentes.
- **PT3 - Diseño y desarrollo del sistema web.**
 - T3.1. Definición de funcionalidades del sistema web.
 - T3.2. Estudio de la tecnología a utilizar.
 - T3.3. Despliegue de tecnologías.
 - T3.4. Diseño y desarrollo de autenticación de usuarios.
 - T3.5. Diseño y desarrollo de importación de datos.
 - T3.6. Diseño y desarrollo de representación de entrenamientos.
 - T3.7. Diseño y desarrollo de comparación de golpes.
 - T3.8. Evolución y seguimiento de mejora de técnica de golpeo.
 - T3.9. Diseño y desarrollo de visualización de tutoriales.

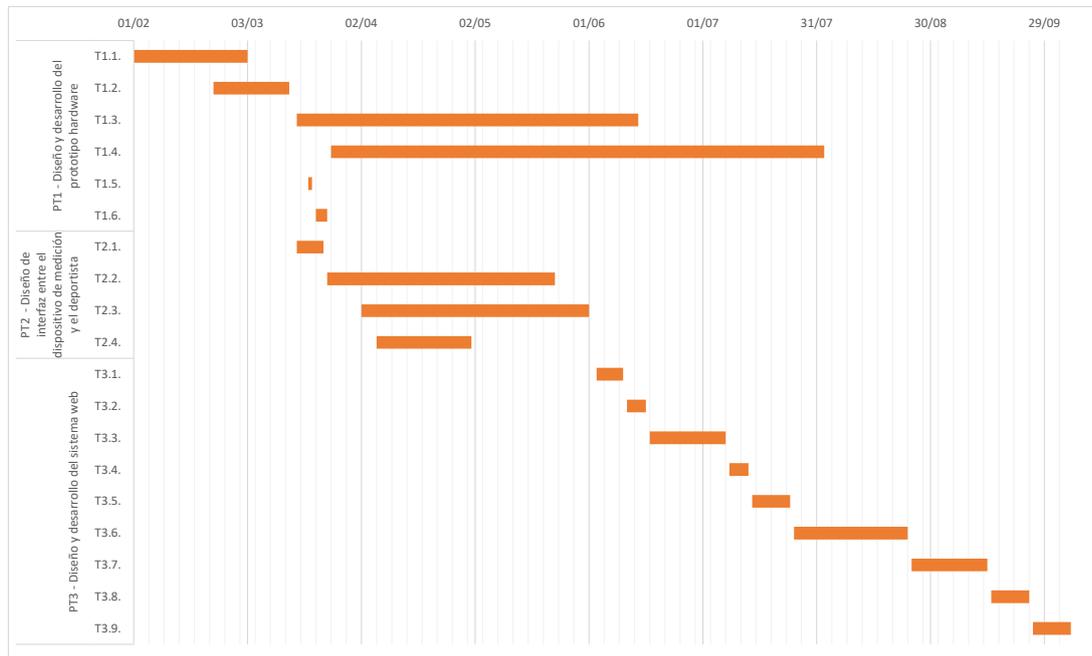


Figura 3.39: Diagrama de Gantt

3.4.1. PT1. Diseño y desarrollo del dispositivo *hardware*

Este paquete de trabajo engloba todo lo relacionado con el dispositivo físico, con el principal objetivo de obtener un prototipo *hardware* capaz de tomar los datos de los sensores necesarios, almacenarlos e importarlos en el sistema *web*. El trabajo en las tareas que conforman este paquete comenzó la primera semana de Febrero (semana 1) y comprendió hasta la última semana de Julio. Se obtuvo una versión del prototipo válida en la segunda semana de abril, pero semanas más tarde se diseñó la correa de cuero para la sujeción del dispositivo, la cual mejoró la estabilidad en la muñeca del deportista.

El PT1 se compone de las siguientes tareas:

- T1.1. Estudio funcionamiento placa Arduino, sensores y módulos de almacenamiento.** Esta tarea está diseñada para conocer de forma más profunda los sensores que se van a manejar en el proyecto. Es una tarea importante ya que conocer el funcionamiento de los sensores hace más sencilla la interpretación de los valores de los mismos, así como la identificación de un posible comportamiento anómalo en un momento dado.
- T1.2. Implementación de ejemplos básicos y procesamiento de datos con sensores y módulos de almacenamiento.** A lo largo de esta tarea se han realizado varios ejemplos de código en los que se utiliza los sensores de forma individual (captación de datos en el caso del sensor MPU6050 y pruebas de almacenamiento en el caso del módulo de tarjeta micro SD. En esta tarea se realizó la búsqueda de librerías existentes que gestionasen los sensores.

- **T1.3. Integración de sensores y módulos en prototipo *hardware*.** Durante esta tarea se han tomado todos los ejemplos individuales de uso de sensores de la tarea anterior y se ha creado un solo *sketch* que integraba todos los sensores y funcionalidades.
- **T1.4. Pruebas y validación del prototipo *hardware*.** Esta tarea sirvió para identificar los fallos del prototipo *hardware* y corregirlos, obteniendo una versión estable del mismo.
- **T1.5. Diseño de fijación del dispositivo a muñeca del deportista.** En esta tarea se ha diseñado una correa de cuero que se ajusta al deportista portador del prototipo *hardware*.
- **T1.6. Implantación y validación del sistema de fijación a muñeca del deportista.** En esta tarea se ha implantado la correa de cuero junto con el prototipo *hardware*. Una vez adaptado el prototipo a la correa, se han realizado pruebas de sujeción en un escenario real. Esto también ha ayudado a realizar pruebas más reales con el prototipo.

El resultado obtenido de PT1 es la creación de un prototipo *hardware* equipado con sus correspondientes sensores y módulos que sea capaz de tomar datos de los mismos y almacenarlos para una posterior interpretación.

3.4.2. PT2. Diseño de interfaz entre el dispositivo de medición y el deportista

Este paquete de trabajo tiene como objetivo el diseño de una interfaz entre el prototipo *hardware* y el deportista. Esta interfaz permitirá al usuario - deportista dar ordenes al prototipo y saber en cada momento qué acción está realizando, teniendo un constante *feedback*.

El PT2 está compuesto por las siguientes tareas:

- **T2.1. Estudio y valoración de los diferentes componentes para la introducción en el prototipo *hardware*.** Con el objetivo de desarrollar una interfaz lo más interpretable, con el menor peso posible y con el menor número elementos se ha realizado un estudio de diversos componentes que se podrían incluir en la interfaz, para cumplir el objetivo deseado.
- **T2.2. Diseño de la interfaz del prototipo *hardware*.** Una vez estudiado los componentes en la tarea anterior, en esta tarea se ha procedido a realizar el diseño de la interfaz con los componentes finales que van a ayudar al desarrollo de la interfaz.
- **T2.3. Implantación de componentes en el dispositivo.** Una vez realizado el estudio de los componentes que se podrían utilizar y desarrollado el diseño de la interfaz se procede a la implantación de los componentes en el prototipo *hardware*. Con el estudio realizado de los diferentes componentes que se podrían utilizar en la interfaz, y el diseño de la interfaz realizado, en esta tarea se procede a la implantación de dichos componentes en el prototipo *hardware*.
- **T2.4. Validación y pruebas de la interfaz con nuevos componentes.** En la última tarea del PT2, se han realizado pruebas con los nuevos componentes implantados en el prototipo y el diseño de la interfaz desarrollada. Las pruebas se han realizado en entornos reales para asegurar que el usuario interpreta la interfaz de manera intuitiva.

El resultado obtenido del PT2 es una interfaz entre el prototipo *hardware* y el usuario -deportista. A través de esta interfaz el usuario puede dar órdenes al prototipo sin que se pierda mucho tiempo en su práctica deportiva. Además, ayuda al deportista a interpretar de manera intuitiva las acciones que está realizando el prototipo.

3.4.3. PT3. Diseño y desarrollo del sistema web

Este paquete de trabajo tiene como responsabilidad el diseño y desarrollo de un sistema *web* que se encargue de mostrar la información al usuario portador del prototipo. De la misma manera, se encargará de guiar al usuario a la mejora de sus técnica en la práctica deportiva con diferentes funcionalidades.

El PT3 se compone de las siguientes tareas:

- **T3.1. Definición de funcionalidades del sistema *web*.** En la primera tarea del paquete PT3 se ha realizado un estudio de las diferentes funcionalidades que debe ofrecer el sistema *web*, definiendo así, las funcionalidades que ayudarán a la consecución del objetivo de realizar un sistema *web* en el que representar los datos registrados por el prototipo y ayudar al usuario a la mejora de su técnica.
- **T3.2. Estudio de la tecnología a utilizar.** En esta tarea se ha realizado un estudio de las tecnologías a utilizar para el desarrollo del sistema *web*, teniendo en cuenta las funcionalidades definidas en la anterior tarea.
- **T3.3. Despliegue de tecnologías.** En esta tarea se procede a la instalación y despliegue de las tecnologías valoradas y acordadas en la anterior tarea. Como sistema gestor de base de datos se utilizará la tecnología Oracle y como *framework* de desarrollo, Oracle APEX.
- **T3.4. Diseño y desarrollo de autenticación de usuarios.** Se procede a implementar la funcionalidad de autenticación de usuarios en el sistema *web*. Esta funcionalidad engloba tanto el registro nuevo como la autenticación para usuarios ya registrados.
- **T3.5 Diseño y desarrollo de importación de datos.** Una vez desarrollada la funcionalidad de autenticación de usuarios, se procede a la implementación de la funcionalidad de importar datos al sistema *web*. Esta funcionalidad permitirá al usuario importar todos los datos registrados por el prototipo *hardware*.
- **T3.6. Diseño y desarrollo de representación de entrenamientos.** Con la funcionalidad de importar datos al sistema *web*, en esta tarea se implementa la funcionalidad de representar dichos datos. En esta tarea se engloban varias funcionalidades, como la representación de los golpes y representación de estadísticas de cada entrenamiento.
- **T3.7. Diseño y desarrollo de comparación de golpes.** En esta tarea se implementan dos funcionalidades con las que el usuario puede comparar su golpeo realizado. El usuario puede comparar su golpeo de manera gráfica a través de representación de gráficos, o de manera visual a través de vídeos.
- **T3.8 Evolución y seguimiento de mejora de técnica de golpeo.** En esta tarea se implementa la funcionalidad basada en objetivos y basada en la obtención de puntuación, que permite al usuario ir mejorando su técnica de golpeo con el cumplimiento de dichos objetivos.

El resultado obtenido del PT3 es un sistema *web*, que sirve para visualizar los datos obtenidos por el prototipo *hardware* para ayudar al usuario a mejorar su técnica en deportes de raqueta.

3.5. METODOLOGÍA DE TRABAJO

Para el desarrollo del proyecto, se ha optado por seguir una metodología basada en un desarrollo iterativo e incremental [16] ya que esta tecnología se adaptaba perfectamente a este proyecto. Se ha optado por esta metodología debido a que ésta permite separar la complejidad del proyecto en varias iteraciones e ir obteniendo versiones más estables a medida que se cumplía con los objetivos de cada iteración. Además, el uso de esta metodología disminuye las probabilidades de riesgo de fallo en el sistema por su implementación mediante iteraciones.

Para llevarlo a cabo, se han seguido una secuencia de pasos no lineales haciendo que cada poco tiempo se tenga una versión operativa del prototipo final. En este TFG es muy importante esta característica, ya que era necesario tener una versión operativa del prototipo *hardware* usado antes de comenzar con otras fases como el desarrollo del sistema *web*. Las posteriores iteraciones se han usado para, prioritariamente, corregir los fallos de iteraciones anteriores. En nuevas iteraciones también se han añadido funcionalidades nuevas, aumentando la calidad del producto final. Al final de cada iteración se ha mantenido una reunión con el director del proyecto, que ha identificado los fallos en el trabajo realizado hasta ese momento, proponiendo mejoras sobre el trabajo ya desarrollado y nuevas funcionalidades que añadir.

El desarrollo del proyecto ha estado motivado por una metodología ágil [17]. Cada iteración se ha dividido en una lista de tareas, a las que se les ha dado una prioridad antes de comenzar la iteración. Estas tareas han sido de corta duración, motivadas por la entrega de *software* funcional y que aporta valor. La comunicación con el director del proyecto ha sido continua durante la ejecución de las tareas. Durante los primeros meses del proyecto se mantuvieron reuniones presenciales con el director del proyecto. Durante los meses posteriores las reuniones se mantuvieron a través de videoconferencias debido al cambio de domicilio y la situación laboral del estudiante. Se ha usado un método basado en un tablero *Kanban* para gestionar el tiempo y visualizar en todo momento qué tareas se han realizado y cuáles faltan por revisar. En la Figura 3.40 se puede ver un ejemplo del uso de este tablero.

También se ha usado *git* como sistema de control de versiones, para mantener un registro detallado de qué cambios se realizan en cada uno de los archivos del repositorio creado en este proyecto. El repositorio *git* se ha alojado en *Github*, como se puede observar en la Figura 3.41.

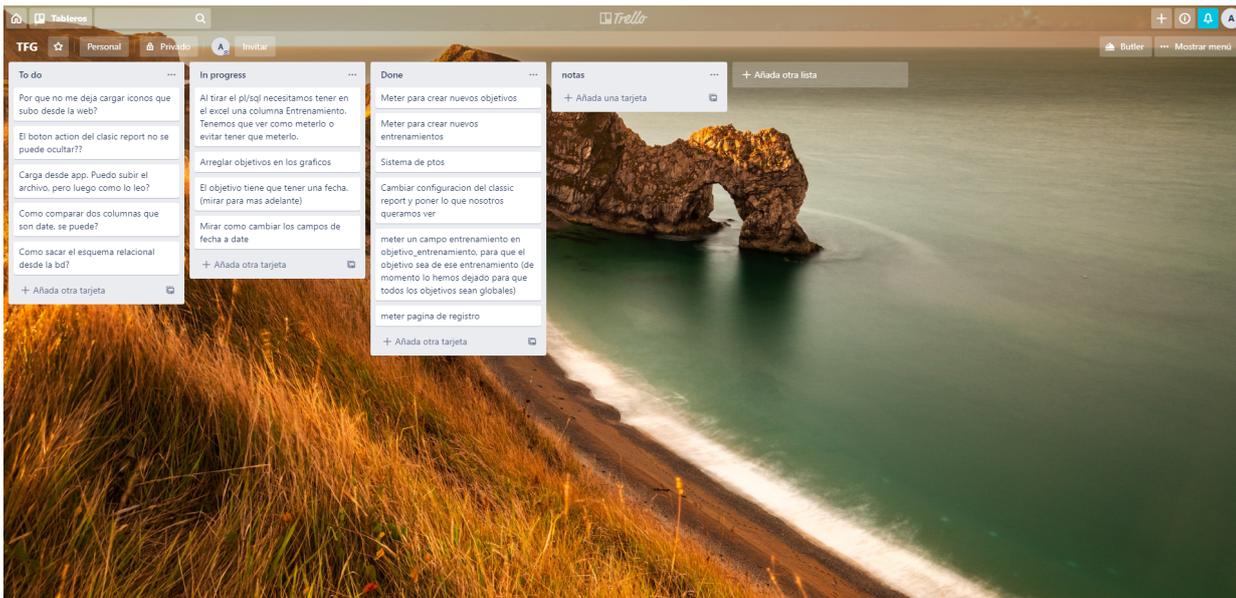


Figura 3.40: Tablero *Trello*.

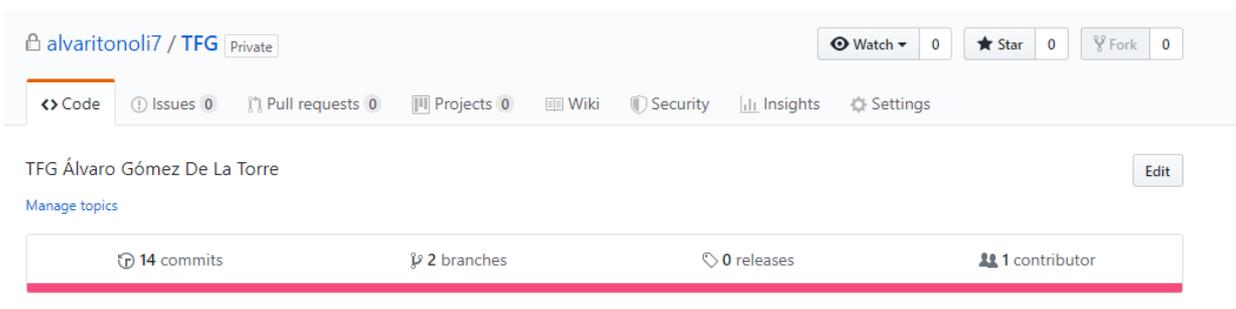


Figura 3.41: Repositorio *git* del proyecto usado en *GitHub*.

3.5.1. Iteraciones

Este proyecto está compuesto por un total de 12 iteraciones. Las primeras iteraciones se centran en el análisis del alcance del proyecto, la recogida de requisitos y la búsqueda de componentes necesarios para construir el prototipo *hardware*. A continuación las iteraciones se centran en la mejora del del prototipo *hardware* y en el diseño y desarrollo de la interfaz del prototipo. Además del diseño y desarrollo del sistema *web*. Las últimas iteraciones se centran en las pruebas del prototipo *hardware* y el sistema *web* para la mejora de ambos y, la redacción de la presente memoria. A continuación, se detallan cada una de las 12 iteraciones que se han realizado:

Iteración 1

En esta iteración se llevaron a cabo las tareas relacionadas con la especificación de requisitos. Una vez definidos los requisitos se definió el alcance del proyecto, para tener una idea de tiempo y los recursos que se emplearían. Durante esta iteración también se llevó a cabo la realización del anteproyecto.

Los requisitos que se obtuvieron fueron:

- El prototipo *hardware* será capaz de obtener datos se los sensores necesarios.

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
Documentación		Definición de requisitos	20h
		Definición del alcance del problema	
		Realización del anteproyecto	

Tabla 3.9: Descripción resumida de la primera iteración

- El prototipo *hardware* será capaz de registrar los datos en el módulo de memoria adecuado.
- La interfaz entre el prototipo y el usuario debe ser lo mas interpretable posible para ayudar al usuario.
- La interfaz deberá reflejar que golpe se está realizando en cada momento.
- La interfaz deberá permitir al usuario dar órdenes para indicar el tipo de golpeo que va a realizar.
- Se diseñará una correa de sujeción para el prototipo *hardware*.
- El sistema *web* será capaz de registrar los datos obtenidos por el prototipo *hardware*.
- El sistema *web* mantendrá un histórico de todos los entrenamientos realizados por el deportista.
- El sistema *web* hará uso de autenticación de usuarios para garantizar la información del sistema *web*.
- El sistema *web* deberá mostrar toda la información registrada en el prototipo *hardware* de manera legible para el usuario.
- El sistema *web* mostrará estadísticas generales y de cada entrenamiento.
- El sistema *web* proporcionara un sistema de puntuación para la motivación del deportista.
- El sistema *web* proporcionara un sistema de cumplimiento de objetivos.
- El sistema proporcionará ayuda al usuario sobre como realizar su golpe.
- El sistema *web* tendrá un diseño *responsive*, que se adaptará al tamaño de pantalla del dispositivo dónde se visualice.

Iteración 2

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT1	T1.1	Compra de componentes que conformarían el dispositivo físico	22h
	T1.2	Búsqueda de librerías para los sensores	
	T1.3		

Tabla 3.10: Descripción resumida de la segunda iteración

Durante esta iteración se realizó una búsqueda de todos los sensores que se necesitarían en el proyecto y se procedió a la compra de los mismos. Después de estudiar alternativas en lo que a los microcontroladores se refiere (*rapsberry Pi*, *STM32* o *Arduino*, finalmente se eligió *Arduino* por la rapidez de prototipación, la gran disponibilidad de librerías para muchos sensores

y las necesidades básicas de este proyecto que no requerían un microcontrolador de mayores prestaciones.

En esta iteración también se buscaron librerías existentes para los sensores que se iban a utilizar. En algunos casos fue necesario adaptar estas librerías para este proyecto en concreto.

Iteración 3

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT1	T1.3	Ensamblado de sensores en microcontrolador	40h
PT2	T2.1	Estudio de los componentes necesarios para la interfaz	

Tabla 3.11: Descripción resumida de la tercera iteración

Durante esta iteración se procedió a la integración de todos los sensores para generar un primer prototipo muy básico del dispositivo *hardware*. Además, se procedió a realizar un estudio de los componentes que se podían utilizar en la interfaz del prototipo. Con la integración de todos los sensores y un análisis de los componentes para la interfaz, se tenía como objetivo tener un primera versión muy básica del prototipo *hardware*.

Iteración 4

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT2	T2.2	Diseño de la interfaz del prototipo	30h
	T2.3	Implantación de componentes en la interfaz	

Tabla 3.12: Descripción resumida de la cuarta iteración

Durante la cuarta iteración se realizó el diseño de la interfaz entre el prototipo *hardware* y el deportista. Para hacer el diseño se tuvo que hacer un estudio y valoración previo para ver que componentes eran los más adecuados para ser utilizados en la interfaz. Una vez realizado el diseño de la interfaz, se procedió a la implantación de estos componentes en el prototipo *hardware*, quedando así una primera versión estable de la interfaz.

Iteración 5

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT1	T1.4	Versión más estable del prototipo	20h
PT2	T2.4	Nueva versión de la interfaz Mejoras en el prototipo y la interfaz	

Tabla 3.13: Descripción resumida de la quinta iteración

En la quinta iteración se procedió a realizar pruebas reales con el prototipo para ver su estabilidad. Se realizaron tanto pruebas con el prototipo *hardware* como con su interfaz. Esto sirvió para implantar y desarrollar algunas mejoras y corregir errores. Las pruebas se realizaron en un entorno real, con el deportista practicando la actividad física. Gracias a estas pruebas con el prototipo, se creó una versión más estable del dispositivo.

Iteración 6

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT1	T1.5 T1.6	Sujeción del prototipo en la muñeca del deportista	20h

Tabla 3.14: Descripción resumida de la sexta iteración

En la sexta iteración se analizó como se podría adherir el prototipo *hardware* a la muñeca del deportista. Se realizó un estudio de diversas posibilidades teniendo en cuenta que el método empleado para este objetivo tendría que entorpecer lo menos posible la actividad deportiva del portador del prototipo. Finalmente, se diseñó una correa de cuero a medida para el prototipo. Con esto se consiguió que el prototipo se pudiera adherir a la muñeca del deportista.

Iteración 7

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT3	T3.3	Instalación de tecnologías a utilizar Importación de datos	35h

Tabla 3.15: Descripción resumida de la séptima iteración

En la séptima iteración de implantaron las tecnologías que finalmente se han utilizado para el desarrollo del sistema *web*. Finalmente se ha optado por la utilización de Oracle como SGBD y Oracle Apex como *framework* de desarrollo del sistema *web*. Además, en esta iteración se ha implementado la primera funcionalidad del sistema *web*, la cual consiste en la importación de los datos registrados por el prototipo al sistema *web*.

Iteración 8

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT3	T3.4 T3.5	Autenticación de usuarios	30h

Tabla 3.16: Descripción resumida de la octava iteración

En la octava iteración del proyecto, se implementa la funcionalidad de autenticación de usuarios. Con esta funcionalidad dotamos al sistema *web* de seguridad sobre los datos que almacena.

Iteración 9

En la novena iteración se desarrollaron las funcionalidades de representación de entrenamientos y comparación de golpes. La representación de entrenamientos engloba varias subtarear, en las

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT3	T3.6 T3.7	Representación de entrenamientos Comparación de golpes	30h

Tabla 3.17: Descripción resumida de la novena iteración

cuales se representa todo lo relacionado con los entrenamientos, tanto estadísticas como golpes. La comparación de golpes, al igual que la representación de entrenamientos también engloba varias subtareas relacionadas con la comparación de golpes, ya que existen varias maneras de comparar los golpes que realiza un usuario.

Iteración 10

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT3	T3.8	Seguimiento de mejora de técnica Pruebas de funcionamiento	30h

Tabla 3.18: Descripción resumida de la décima iteración

En la iteración décima se implementa la última funcionalidad del presente proyecto. Esta funcionalidad consiste en el desarrollo de las funcionalidades adecuadas para el seguimiento de la mejora de técnica del deportista. Además, en esta última iteración se realizan pruebas de todo el sistema *web* para ver su robustez y tolerancia a fallos.

Iteración 11

Paquetes de trabajo	Tareas a las que afecta	Objetivos alcanzados	Tiempo estimado de realización
PT3	T3.8	Pruebas finales y funcionales del sistema <i>web</i>	15h

Tabla 3.19: Descripción resumida de la undécima segunda iteración

Por último, en la última iteración del presente proyecto se han realizado las últimas pruebas funcionales para verificar y evaluar el funcionamiento del sistema *web*. Estas últimas pruebas han servido para realizar los últimos retoques en el sistema *web*.

3.6. CARACTERÍSTICAS *HARDWARE* Y *SOFTWARE* DEL DESARROLLO

A lo largo de esta sección se van a exponer los componentes *hardware* y *software* que se han empleado para la consecución de este proyecto.

3.6.1. Medios *hardware*

Los componentes *hardware* que se han usado a lo largo de este proyecto son los siguientes:

- **Equipo de trabajo.** Ordenador portátil Hp pavilion, con las siguientes características principales:
 - Procesador Intel Core i7 octava generación.
 - Memoria RAM de 8GB de capacidad.
 - 1TB de almacenamiento HDD y 250GB de almacenamiento SSD.
 - Tarjeta gráfica GeForce GTX 1050.
- **Microcontrolador.** Arduino UNO REV3.
- **Acelerómetro y giroscopio.** Modelo MPU6050.
- **Módulo tarjeta micro SD.** Modelo FUT3002.
- **Leds indicadores.**

3.6.2. Medios *software*

A continuación se muestran las herramientas *software* usadas durante este proyecto:

Sistemas operativos

- **Windows 10.** Es el sistema operativo en el que se ha desarrollado el código fuente que gestiona los distintos sensores integrado en el prototipo *hardware*. Además, se ha usado este sistema operativo para escribir el documento de la presente memoria.
- **Oracle Linux 7.0.** Este sistema operativo se ha utilizado para el desarrollo del sistema gestor base de datos y para el desarrollo del sistema *web*.

Herramientas de desarrollo

- **Arduino IDE.** Entorno de desarrollo utilizado para definir todo el funcionamiento del prototipo *hardware*. Además, este entorno de desarrollo hace que la carga del *software* desarrollado para el microcontrolador *Arduino* se lleve a cabo de forma muy sencilla. Es la principal razón por la que se ha decidido usarlo en este proyecto.
- **Git.** Se trata del *software* de control de versiones usado a lo largo del proyecto. Se ha usado para gestionar los distintos cambios en el código fuente y en la documentación.
- **Trello.** Se trata de una herramienta para la administración de tareas. Esta herramienta ha permitido llevar un orden con las tareas a realizar en cada momento del proyecto.
- **Oracle APEX.** Herramienta que se ha usado para el desarrollo del sistema *web*.

Tecnologías usadas

- **Javascript** ha sido el lenguaje elegido para el desarrollo *web*.
- **C** ha sido usado durante el desarrollo del código para el microcontrolador Arduino. El código fuente de la solución que se carga en el microcontrolador no es exactamente C++ (es código Arduino), pero está basado en este lenguaje.

Documentación

- **Overleaf** ha sido el editor de texto *Latex* que se ha utilizado para la confección del presente documento.
- **Draw.io** se trata de una *web* en la que se pueden crear diagramas *online* de forma gratuita. Ha sido la principal herramienta de creación de imágenes y diagramas en este proyecto.

CAPÍTULO 4

RESULTADOS

En este capítulo se realizará un análisis para mostrar el cumplimiento de los objetivos específicos que se propusieron en el capítulo 2. Para la verificación de las distintas partes que componen este proyecto se han realizado unos test de prueba que ayudarán a verificar el correcto funcionamiento del presente proyecto y a la consecución del objetivo de ayudar al deportista a mejorar su técnica en deportes de raqueta.

4.1. DESARROLLO DE UN PROTOTIPO *HARDWARE* EQUIPADO CON VARIOS SENSORES

El primer objetivo específico de este proyecto era el diseño y desarrollo de un prototipo *hardware*, que equipase varios sensores necesarios para la obtención de datos acerca del golpeo que realiza un deportista en deportes de raqueta. En la Figura 4.1 se puede observar cómo el prototipo capta los datos de los sensores que equipa. Dichos datos son guardados en el módulo de tarjeta micro SD como se puede comprobar en la Figura 4.2.



```
COM7 (Arduino/Genuino Uno)
Iniciando SD ...Inicializacion exitosa de tarjeta SD
Archivo nuevo, Escribiendo encabezado(fila 1)
Sensor acelerometro iniciado correctamente
AngY: -1.12,AngY: 1.28
AcelX: -320,AcelY: 324,AcelZ: 15520
Derecha
CAMBIO
Grabando
AngY: -1.51,AngY: 1.54
AcelX: -380,AcelY: 344,AcelZ: 15464
Derecha
Grabando
AngY: -1.32,AngY: 1.25
AcelX: -420,AcelY: 344,AcelZ: 15424
Derecha
```

Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Figura 4.1: Datos de los sensores equipados en el prototipo *hardware*.

El objetivo del prototipo es captar datos sobre el golpeo que realiza un deportista. Para llevar a cabo esto de la forma más precisa es necesario evitar la pérdida de información cuando se está ejecutando el golpeo para obtener datos lo más precisos y completos posibles. Para ello los datos registrados por el prototipo se realizan cada 100 ms, así se puede evaluar todo el golpeo

64.1. DESARROLLO DE UN PROTOTIPO *HARDWARE* EQUIPADO CON VARIOS SENSORES

NumeroGolpe	Golpe	AceleracionX	AceleracionY	Aceleracionz	AnguloX	AnguloY	AnguloZ	FechaEntrenamiento
1	Globo	14688	3684	7248	61.17	12.59	25.50	22/09/2019
1	Globo	10552	8564	11500	36.39	28.49	39.79	22/09/2019
1	Globo	7540	14472	15412	19.64	40.17	43.39	22/09/2019
1	Globo	9044	7000	5508	45.35	33.68	25.86	22/09/2019
1	Globo	15736	2180	4444	72.64	8.04	15.90	22/09/2019
1	Globo	16256	3144	5096	69.84	10.54	17.17	22/09/2019
1	Globo	15328	3436	6952	63.00	11.60	24.19	22/09/2019
2	Globo	14276	2780	8496	58.00	9.46	30.08	22/09/2019
2	Globo	1620	13600	8312	5.74	58.26	30.76	22/09/2019
2	Globo	4696	1084	9164	27.67	5.73	62.47	22/09/2019
2	Globo	7712	836	13852	29.08	3.02	60.78	22/09/2019
2	Globo	15384	8708	3836	58.48	28.79	12.47	22/09/2019
2	Globo	14052	2524	8436	57.74	8.55	30.74	22/09/2019

Figura 4.2: Archivo guardado en modulo de tarjeta micro SD.

sin pérdida de información. En la Tabla 4.1 se muestra un ejemplo de como las aceleraciones en los tres ejes de coordenadas van cambiando progresivamente en cada instante de tiempo sin la pérdida de información. En la Tabla 4.2 se muestra la información de cómo los ángulos van cambiando de forma progresiva sin perder información.

Instante	Valor Aceleración X	Valor Aceleración Y	Valor Aceleración Z
1	15328g	3436g	6952g
2	14276g	2780g	8496g
3	14688g	3684g	7248g
4	10552g	8564g	11500g
5	7540g	14472g	15412g
6	7712g	836g	13852g
7	9044g	7000g	5508g
8	15736g	2180g	4444g
9	16256g	3144g	5096g
10	14276g	2780g	8496g

Tabla 4.1: Ejemplo de grabación de aceleración de golpeo.

Instante	Valor Ángulo X	Valor Ángulo Y	Valor Ángulo Z
1	58.00°	9.46°	30.08°
2	61.17°	12.59°	25.50°
3	36.39°	28.49°	39.79°
4	19.64°	40.17°	43.39°
5	29.08°	3.02°	60.78°
6	45.35°	33.68°	25.86°
7	72.64°	8.04°	15.90°
8	69.84°	10.54°	17.17°
9	63.00°	11.60°	24.19°
10	58.00°	9.46°	30.08°

Tabla 4.2: Ejemplo de grabación de ángulos de golpeo.

En las tablas anteriores se han registrado los datos durante un segundo completo. Como se puede observar entre los instantes 4 y 7 se produce un cambio en los valores registrados. Esto indica que el dispositivo es movido por el deportista y que está realizando el golpeo. Se ha optado por el registro de datos cada 100 ms. Con este intervalo de tiempo pequeño se asegura que no se pierde información sobre los datos de golpeo que se registran en el prototipo. El tiempo de golpeo

medio de un deportista oscila entorno a un segundo contemplando todas sus fases, por ello, con recoger datos cada 100 ms es suficiente para cubrir todas las fases del golpeo con precisión.

Para demostrar que los datos registrados por el prototipo son precisos, se han comparado estos ángulos con el ángulo realizado a través de un transportador de ángulos. De esta manera verificamos que se está registrando el ángulo correcto.

Para verificar esta medición se han realizado dos test en los cuales se hace una medición de los ángulos X e Y con respecto a un plano horizontal.

Test 1. Evaluación de la precisión en el eje Y

En el primer test se verificará la precisión de medición en el eje Y. Para la realización del primer test se han empleado tres ángulos de prueba para verificar si precisión. En la Tabla 4.3 se muestra un resumen de los ángulos utilizados y su resultado.

Angulo esperado	Ángulo registrado
45°	44.67°
45°	43.15°
45°	45.18°
30°	30.47°
30°	29.08°
30°	31.40°
60°	60.67°
60°	60.50°
60°	61.33°

Tabla 4.3: Valores test 1. Ángulos en el eje Y

En las Figura 4.3 se puede observar, con la ayuda de un transportador de ángulos, la inclinación exacta del prototipo *hardware*, concretamente la inclinación del módulo MPU6050 encargado de realizar estas mediciones.

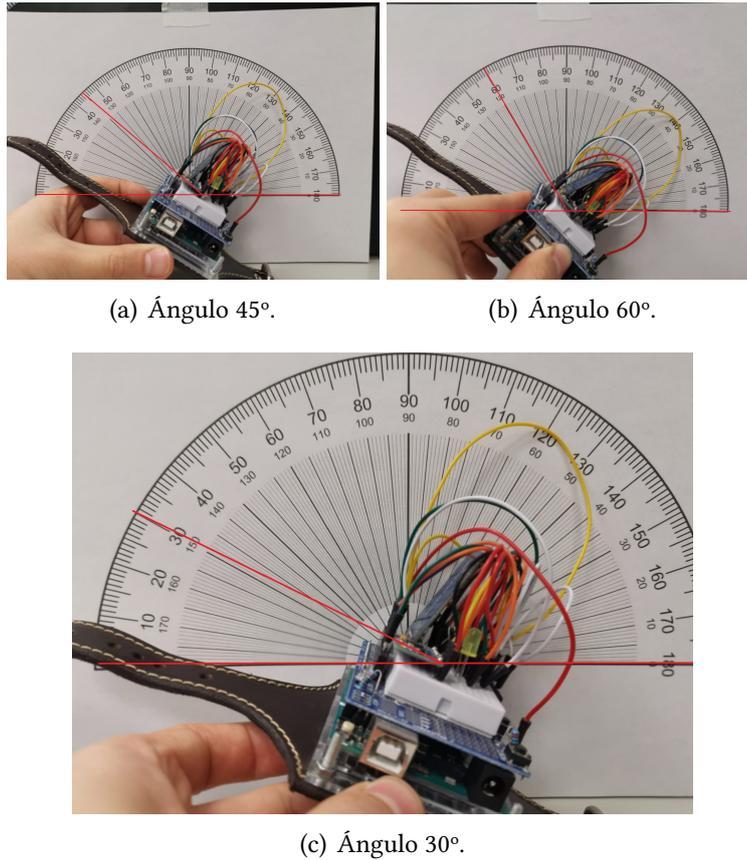


Figura 4.3: Ángulos empleados para realizar el test del eje Y.

Test 2 Evaluación de la precisión en el eje X

En el test 2 se ha realizado la comprobación de la precisión del eje X sobre un plano horizontal. Al igual que en el anterior test, en la Tabla 4.4 se muestran un resumen de los valores utilizados y esperados. En la Figura 4.4 se muestra la inclinación empleada para realizar el test.

Ángulo esperado	Ángulo registrado
45°	44.23°
45°	45.56°
45°	46.29°
30°	30.59°
30°	28.76°
30°	30.30°
60°	60.45°
60°	59.56°
60°	60.21°

Tabla 4.4: Valores test 2. Ángulos en el eje X

Tras estas pruebas se puede determinar que la medición de los ángulos es exacta con uno o dos grados de error. Esto ayuda a registrar todos los datos acerca del golpeo de manera correcta y sin errores.

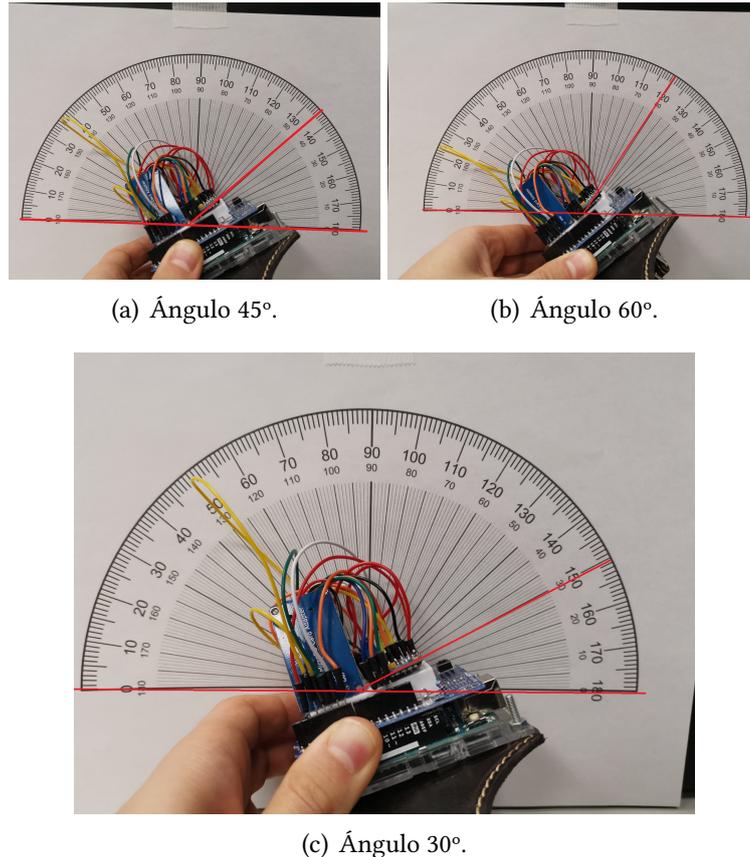


Figura 4.4: Ángulos empleados para realizar el test del eje X.

4.2. DISEÑO Y DESARROLLO DE UNA INTERFAZ DE COMUNICACIÓN ENTRE EL DISPOSITIVO DE MEDICIÓN Y EL DEPORTISTA

El segundo objetivo principal del presente proyecto era el diseño y desarrollo de una interfaz de comunicación entre el prototipo *hardware* y el deportista para facilitar la interacción entre ambos.

Para facilitar la interacción entre el dispositivo físico y el deportista se han incluido varios componentes indicadores que ayudan al usuario a saber qué acción se está realizando en el interior del dispositivo. Además se ha implementado una funcionalidad basada en lenguaje gestual que permite al deportista dar órdenes al prototipo *hardware*.

Para indicar al usuario qué acción se está realizando se ha hecho uso de unos *leds* de indicación luminosa. En la Tabla 3.7 se puede ver reflejado que acción indica cada *led*. En esta tabla se diferencia los diferentes tipos de golpes que se pueden realizar, si está ocurriendo algún problema en el prototipo *hardware* y cuándo está listo el prototipo para empezar a registrar datos.

Como se indicaba anteriormente se han implementado una funcionalidad basada en el lenguaje gestual que permite al usuario dar órdenes al dispositivo. En total se han considerado cuatro ordenes que el prototipo y la interfaz son capaces de captar en función de la aceleración y el ángulo que adopta la nuestra posición corporal. En la Figura 4.5 se puede observar qué posición hay que adoptar para dar cada una de estas cuatro órdenes al prototipo. Hay que tener en cuenta

que para la indicación de estas órdenes hay que mantener estas posiciones durante uno o dos segundos hasta que el dispositivo registra la orden.

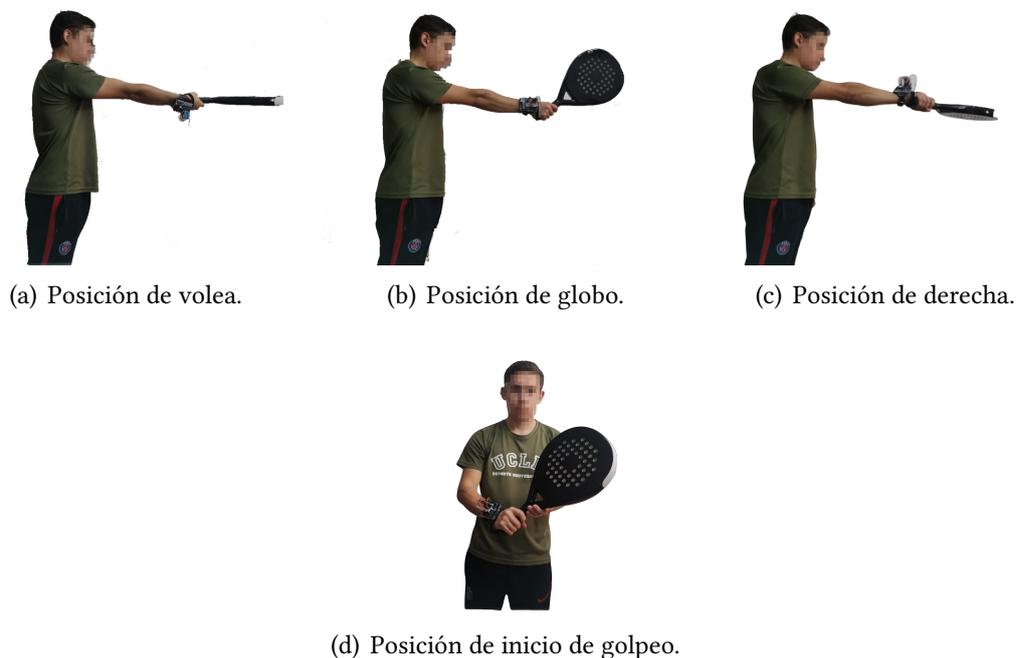


Figura 4.5: Posiciones que debe adoptar el usuario para dar órdenes al prototipo *hardware* a través de la interfaz lógica desarrollada.

En la Figura 4.6 se puede observar los diferentes cambios que registra el prototipo cuando se adopta cada una de estas posiciones. Se puede observar que cuando se adopta la posición del golpeo de derecha 4.5 c), el dispositivo lo registra y activa el *led* azul. A continuación, el usuario decide realizar un golpe de volea 4.5 a), para ello, adopta la posición de golpe de volea, el dispositivo lo registra y activa el *led* blanco para hacer saber al usuario que está preparado para registrar un golpe de volea. Lo mismo ocurriría si el usuario decide realizar un golpe de globo.

```
Volea
Led blanco - > ON
Parado
AngY: 38.69,AngY: 51.12
AcelX: 11060,AcelY: 13740,AcelZ: 1080
Parado
AngY: 6.46,AngY: 23.52
AcelX: 1368,AcelY: 4560,AcelZ: 10304
Parado
AngY: -4.19,AngY: 2.00
AcelX: -1200,AcelY: 576,AcelZ: 15828
Derecha
Led azul - > ON
Parado
- - - - -
```

Figura 4.6: Ejemplo de cambio de movimiento al adoptar las posiciones anteriores para dar órdenes al dispositivo.

Para la verificación de estas funcionalidades se han realizado tres test con tres personas que practican el deporte de pádel. Cada persona realizará un total de 32 veces la elección de golpes

(ocho veces cada tipo de golpeo). Con esto se determinará el porcentaje de acierto que tiene la interfaz a la hora de interpretar el movimiento del deportista.

Test 1. Usuario 1

Prueba realizada con el primer usuario.

Golpeo elegido	Resultado	Valores aceleración X,Y,Z	Valores ángulos X,Y
Derecha	Acierto	5685g, 1962g, 12364g	5.63°, -3.96°
Derecha	Acierto	-3984g, -6520g, 16026g	7.69°, 4.23°
Derecha	Acierto	4865g, -3658g, 15874g	8.56°, 4.48°
Globo	Acierto	13598g, -2036g, 3698g	53.26°, -13.25°
Globo	Fallo	15698g, 8541g, 2368g	69.23°, 7.78°
Volea	Acierto	7568g, 8541g, 1258g	45.36°, -3.58°
Volea	Acierto	8568g, 6589g, -2369g	36.58°, -5.69°
Inicio golpeo	Acierto	15689g, 3987g, 7896g	70.65°, 7.54°
Inicio golpeo	Fallo	16413g, 2126g, 8178g	49.96°, 20.63°
Inicio golpeo	Acierto	14788g, 2584g, 8963g	50.25°, 3.25°

Tabla 4.5: Test 1 para medir el acierto de elección de golpeo.

En la Tabla 4.5 se observa un ejemplo de las mediciones que se han registrado a la hora de seleccionar el tipo de movimiento que se va a realizar. Se observa que estas diez mediciones, se han registrado dos fallos. El primero de ellos ha sido a la hora de elegir el golpe de globo en el cual el usuario ha utilizado un ángulo demasiado cerrado de 7.78° en el eje de la Y, con esto el prototipo no ha conseguido registrar el movimiento adecuadamente. El segundo fallo que se comprueba es a la hora de indicarle al prototipo que está listo para realizar un golpeo, en el cual como la vez anterior, el usuario usa un ángulo demasiado cerrado de 20.63° en el eje de la Y.

A continuación, en la Tabla 4.6 se muestra un resumen global de todas las pruebas realizada con este usuario determinando y su porcentaje de acierto y fallo en las 32 mediciones realizadas.

Golpeo elegido	Aciertos	Fallos	%Acierto
Derecha	7	1	87,5 %
Globo	7	1	87,5 %
Volea	8	0	100 %
Inicio golpeo	7	1	87,5 %
Total	29	3	90.6 %

Tabla 4.6: Resumen de mediciones test 1.

Test 2. Usuario 2

Prueba realizada con el segundo usuario.

En las mediciones de ejemplo expuestas anteriormente, el deportista tiene dos fallos a la hora de elegir su golpeo. El primer fallo, al elegir un golpe de derecha en el cual la aceleración en el eje X sobrepasa en valor estipulado de 6000 g. La segunda elección fallida es a la hora de indicar el inicio de golpeo, donde el usuario emplea un ángulo con demasiada inclinación en el eje X.

En la Tabla 4.8 se muestra un resumen de las 32 mediciones realizadas.

Golpeo elegido	Resultado	Valores aceleración X,Y,Z	Valores ángulos X,Y
Derecha	Acierto	4865g, -3658g, 15874g	5.63°, -3.96°
Derecha	Fallo	6064g, -5412g, 14239g	7.85°, 2.69°
Derecha	Acierto	5698g, -2368g, 14865g	9.25°, 3.96°
Globo	Acierto	13458g, -1287g, 3478g	57.57°, -10.36°
Globo	Acierto	15322g, -3458g, 1368g	62.34°, 9.48°
Volea	Acierto	6852g, 7425g, -1478g	43.36°, -5.58°
Volea	Acierto	8369g, 6138g, -2759g	33.48°, -5.69°
Inicio golpeo	Fallo	16187g, 2158g, 8476g	79.96°, 4.28°
Inicio golpeo	Acierto	15177g, 3254g, 6851g	49.96°, 4.63°
Inicio golpeo	Acierto	13698g, 4587g, 4783g	72.68°, 5.27°

Tabla 4.7: Test 2 para medir el acierto de elección de golpeo.

Golpeo elegido	Aciertos	Fallos	%Acierto
Derecha	6	2	75 %
Globo	8	0	100 %
Volea	7	1	87,5 %
Inicio golpeo	7	1	87,5 %
Total	28	4	87.5 %

Tabla 4.8: Resumen de mediciones test 2.

Test 3. Usuario 3

Prueba realizada con tercer usuario.

Golpeo elegido	Resultado	Valores aceleración X,Y,Z	Valores ángulos X,Y
Derecha	Acierto	2895g, -1258g, 14863g	4.69°, 4.36°
Derecha	Acierto	3698g, -3612g, 17634g	7.85°, 2.69°
Derecha	Acierto	5147g, -2596g, 15698g	5.36°, -1.96°
Globo	Acierto	13789g, -1652g, 2147g	49.24°, -7.69°
Globo	Fallo	8265g, -1587g, 2369g	56.24°, 10.36°
Volea	Acierto	5418g, 6325g, -2369g	49.25°, -4.69°
Volea	Acierto	8369g, 6138g, -2759g	39.36°, -1.69°
Inicio golpeo	Acierto	14597g, 2158g, 8476g	65.24°, 6.28°
Inicio golpeo	Acierto	15177g, 3254g, 6851g	49.96°, 5.24°
Inicio golpeo	Acierto	16541g, 2478g, 4478g	65.12°, 7.36°

Tabla 4.9: Test 3 para medir el acierto de elección de golpeo.

En las mediciones de ejemplo registradas con la ayuda del tercer usuario, se ha obtenido un porcentaje de acierto mayor que en los otros dos test. En la Tabla 4.10 se puede ver reflejado el resultado de este último test.

Golpeo elegido	Aciertos	Fallos	%Acierto
Derecha	7	1	87.5 %
Globo	7	1	87.5 %
Volea	8	0	100 %
Inicio golpeo	8	0	100 %
Total	30	3	93.75 %

Tabla 4.10: Resumen de mediciones test 3.

Resultados finales

Golpeo elegido	Aciertos	Fallos	%Acierto
Derecha	20	4	83.33 %
Globo	22	2	91.6 %
Volea	23	1	95.83 %
Inicio golpeo	22	2	91.6 %
Total	87	9	92.7 %

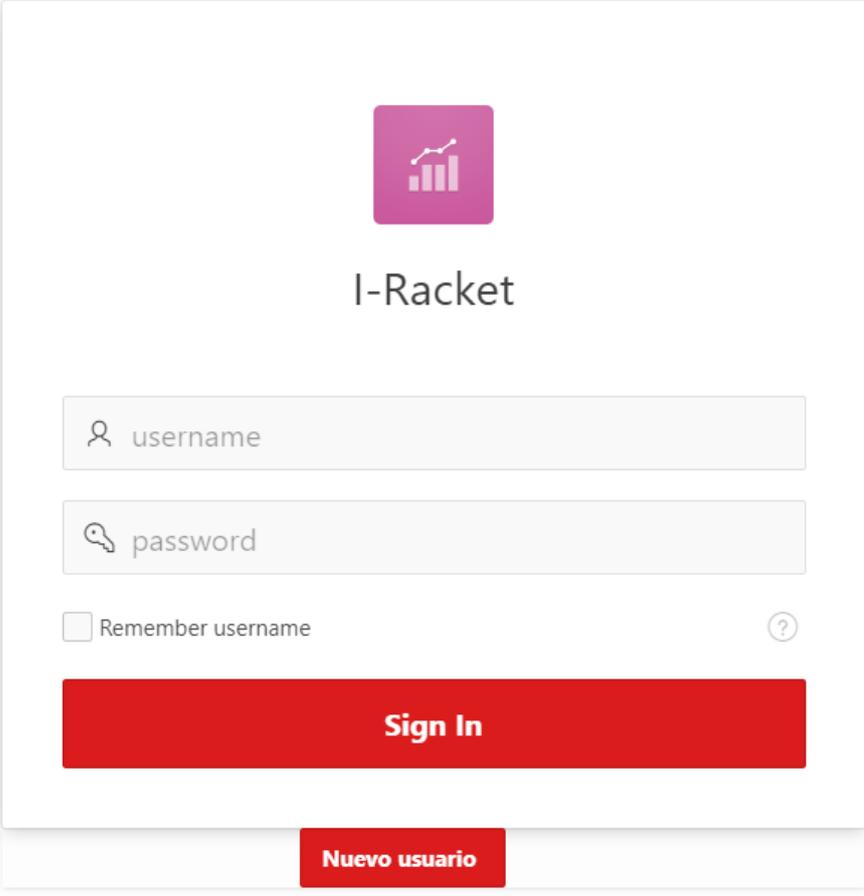
Tabla 4.11: Resumen de mediciones totales realizadas en los tres test.

Tras haber realizado los tres test con tres personas diferentes, se obtiene un porcentaje de acierto a la hora de elegir el golpeo del 92.7 %. Cabe destacar que se han tenido más fallos a la hora de la elección del golpe de derecha con respecto a la elección del resto de golpes. Esto se debe a que no todos los deportistas utilizan la misma empuñadura a la hora de agarrar la raqueta, por lo que es posible que en las primeras elecciones no se adopte la posición adecuada para elegir los diferentes golpes.

4.3. DISEÑO DE UN SISTEMA WEB PARA LA VISUALIZACIÓN DE LA INFORMACIÓN RECOGIDA POR EL PROTOTIPO

El último objetivo del presente proyecto era el diseño y desarrollo de un sistema *web* que permitiera la visualización de los datos recogidos por el prototipo *hardware*, visualización de estadísticas, resultados y progreso.

En la Figura 4.7 se puede observar la pantalla principal de acceso al sistema *web*. Para el desarrollo de este sistema se ha ido diseñando y creando diversas funcionalidades que ayudarán a la consecución del objetivo principal de obtener un sistema *web* que sirva para visualizar los datos recogidos por el prototipo, visualización de estadísticas y progreso.



The image shows a login form for a system named "I-Racket". At the top center is a purple square icon containing a white bar chart with an upward-trending line. Below the icon, the text "I-Racket" is displayed in a dark grey font. The form consists of two input fields: the first is labeled "username" with a person icon on the left, and the second is labeled "password" with a key icon on the left. Below these fields is a checkbox labeled "Remember username" and a small circular help icon with a question mark. A prominent red button with the text "Sign In" in white is centered below the form. At the bottom of the page, there is a red button with the text "Nuevo usuario" in white.

Figura 4.7: Login en el sistema *web*.

4.3.1. Visualización de entrenamientos

El usuario tiene la posibilidad de visualizar todos los entrenamientos que ha realizado a través de un calendario y un listado. El calendario muestra todos los entrenamientos realizados en todas las semanas y meses. Esto ayuda al usuario a planificar todos sus entrenamientos posteriores y repartir la carga de trabajo para poder prevenir lesiones. En la Figura 4.8 se muestra el ejemplo del calendario en el cual el usuario puede ver todos sus entrenamientos realizados indicando por colores cada tipo de entrenamiento.

Además del calendario, se ha optado por mostrar un listado histórico de todos los entrenamientos realizados como se observa en la Figura 4.9.

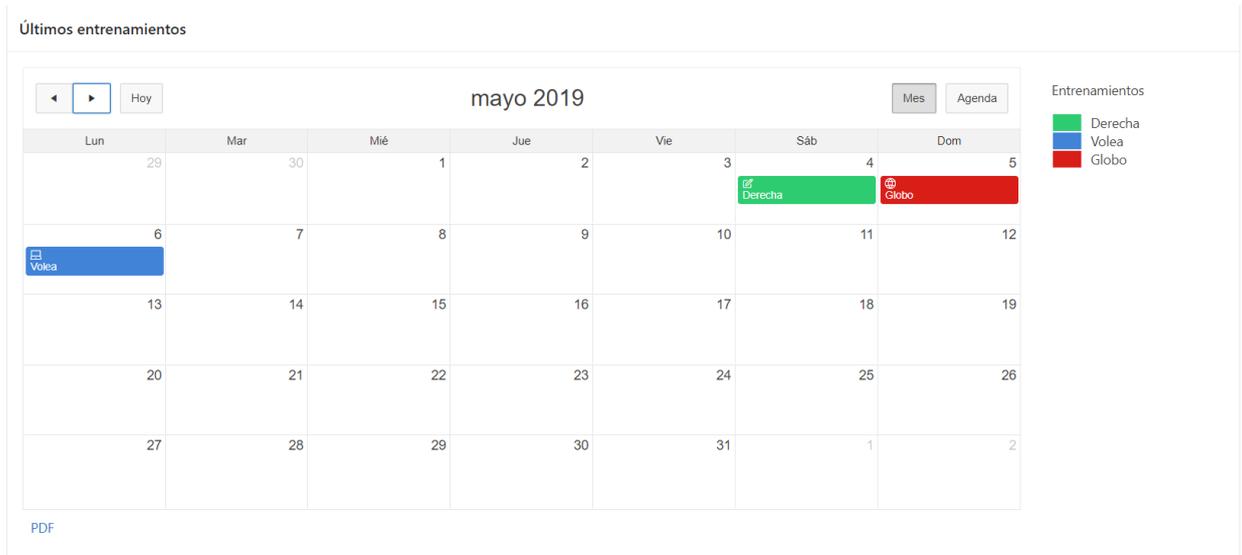


Figura 4.8: Calendario de entrenamientos.

Entrenamientos

	Entrenamiento	Fecha Entrenamiento
	Derecha	04/05/19
	Globo	05/05/19
	Volea	06/05/19
	Derecha	22/08/19
	Globo	22/08/19
	Volea	22/08/19

Figura 4.9: Listado de entrenamientos.

4.3.2. Representación de golpes

Como se ha expuesto en la sección 3.3.1, para la representación de los golpes que el usuario realiza en un entrenamiento se ha hecho uso de gráficas lineales que representan tanto la aceleración como el ángulo empleado durante todo el golpeo.

Se puede observar el cumplimiento de esta tarea en las Figuras 4.10, 4.11, 4.12 que representan la aceleración y en las Figuras 4.13, 4.14, 4.15 que representan el ángulo empleado durante el golpeo. Para ayudar al usuario a interpretar las gráficas, se han marcado tres puntos claves como son el inicio del golpeo, la ejecución del golpeo y la finalización del golpeo. Con esto el usuario podrá realizar un análisis acerca de sus golpesos.



Figura 4.10: Gráfica aceleración en el eje X de un golpe de derecha.

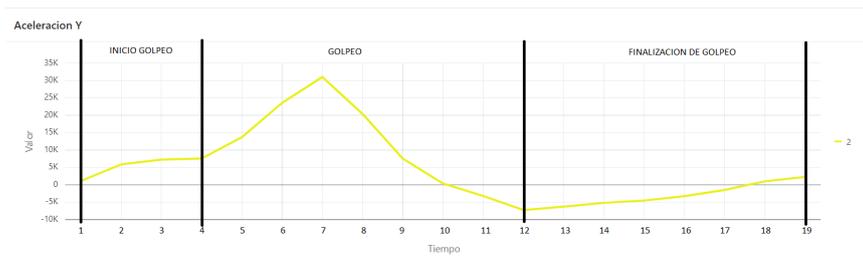


Figura 4.11: Gráfica aceleración en el eje Y de un golpe de derecha.

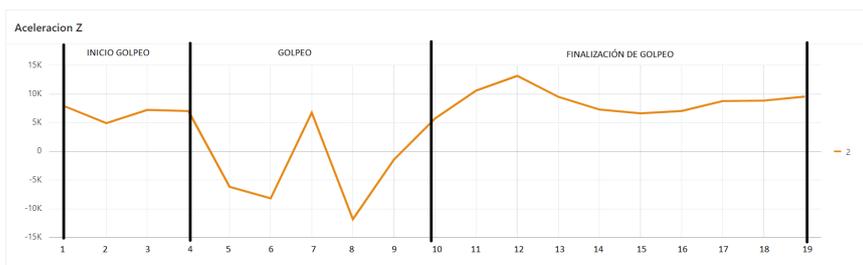


Figura 4.12: Gráfica aceleración en el eje Z de un golpe de derecha.

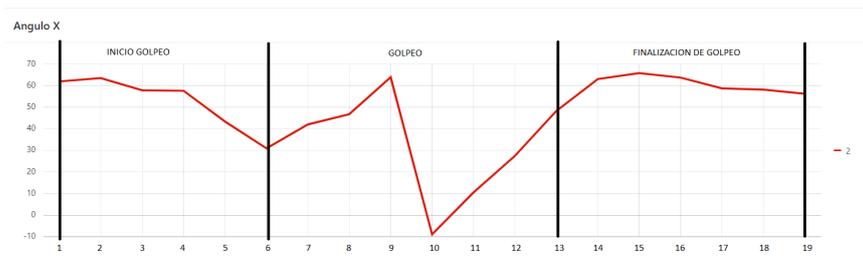


Figura 4.13: Gráfica ángulo en el eje X de un golpe de globo.

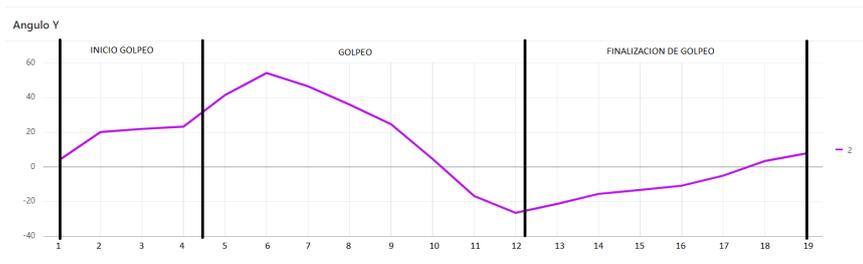


Figura 4.14: Gráfica ángulo en el eje Y de un golpe de globo.

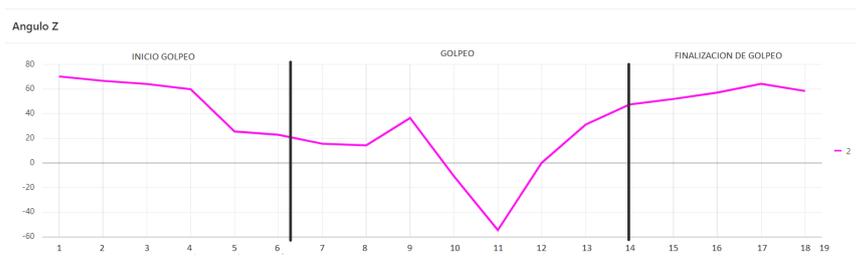


Figura 4.15: Gráfica ángulo en el eje Z de un golpe de globo.

4.3.3. Comparación de golpes y visualización de videotutoriales

Para dar un mayor *feedback* al usuario y ayudarlo a interpretar mejor las gráficas mostradas, éste tiene la posibilidad de comparar cada uno de sus golpes con los de un jugador profesional. En las Figuras 4.16 y 4.17 se puede observar como se realiza la comparación de un golpe de derecha. En la comparación se comparada los tres momentos principales a la hora de realizar un golpeo (inicio, ejecución y finalización).

Las gráficas mostradas a continuación no solo permiten ver los valores de aceleración adecuados en cada uno de los ejes, si no también, la relación temporal. Mediante las gráficas se puede observar también los intervalos de tiempo y su amplitud en los que se producen dichas aceleraciones. En otras palabras, control del tiempo de golpeo. Esto hace referencia a los distintos tipos de golpeo que se pueden ejecutar. Dependiendo de lo que se busque a la hora de ejecutar un golpeo, imprimiremos más aceleración sobre la pelota o menos. Si se busca un golpe fuerte y rápido, se incrementará la aceleración en un intervalo de tiempo muy reducido, mientras que si lo que se busca es un golpe flojo y colocado, se ejecutará el golpeo con una aceleración más prolongada y creciente a lo largo del tiempo.

En la Figura 4.16 se puede apreciar como el jugador profesional (golpe modelo situado a la derecha) alcanza una aceleración adecuada en el eje Y (en el movimiento de la pala de abajo arriba) en un menor tiempo que el jugador *amateur*, lo que permitirá imprimir mayor velocidad a la bola. Por otro lado, la Figura 4.17 muestra como el golpeo del jugador profesional y el *amateur* son bastante similares en el eje X.



Figura 4.16: Comparativa aceleración en el eje Y de un golpe de derecha.

Otra forma de comparar los golpes que realiza un usuario es mediante la comparación visual. La Figura 4.18 muestra como se comparan dos vídeos de una duración corta. De esta manera el usuario puede comparar su golpeo con el de un jugador profesional de manera visual.

Para los usuarios más principiantes se ha creado una funcionalidad que permite visualizar

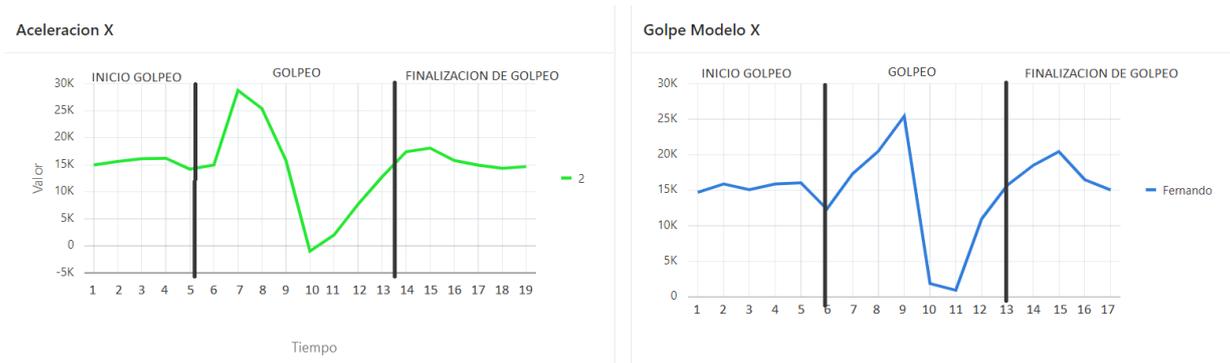


Figura 4.17: Comparativa aceleración en el eje X de un golpe de derecha.

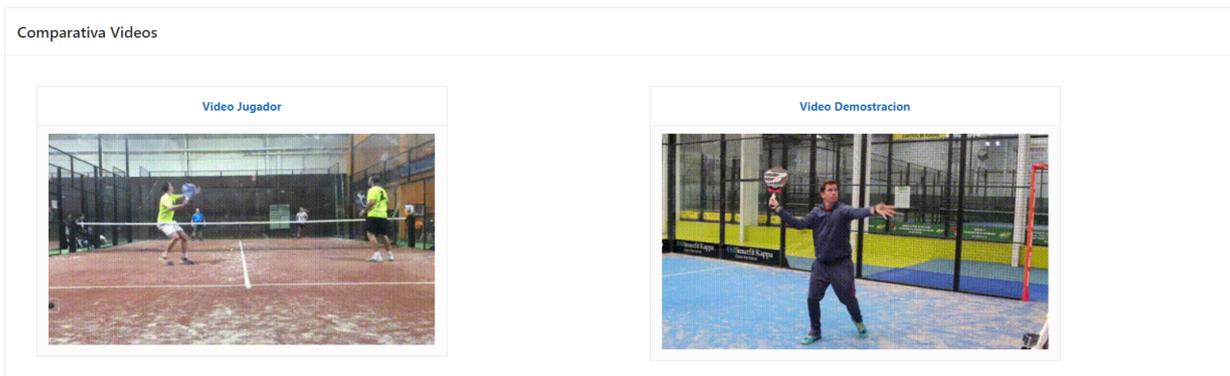


Figura 4.18: Comparación de vídeos entre usuario y profesional

un videotutorial acerca del tipo de golpeo que se quiere realizar, con esto el usuario tiene una idea básica sobre como realizar el golpeo. Para los jugadores más avanzados sirve para visualizar la perfección de la ejecución del golpeo y para seguir consejos acerca de cómo realizar el golpeo correctamente. En la Figura 4.19 se puede observar un ejemplo de visualización de un videotutorial.

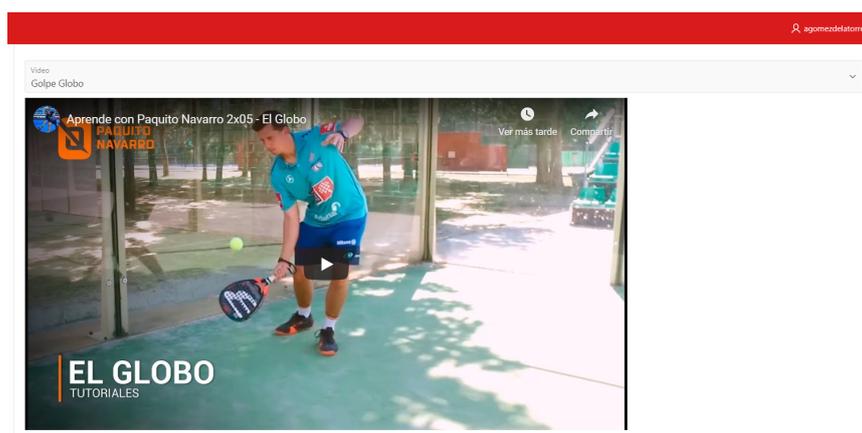


Figura 4.19: Videotutorial ofrecido a los usuario de un golpeo de globo.

Mediante la comparación entre las gráficas ofrecidas, la comparación audiovisual a través de vídeos y la visualización de videotutoriales, el usuario puede determinar qué necesita mejorar de cara a los próximos entrenamientos y así ir mejorando y puliendo su técnica con el paso de los entrenamientos.

Para que el usuario pueda ver un progreso de su actividad, el sistema muestra un histórico de entrenamientos y golpesos realizados, por lo que el usuario podrá comparar golpesos realizados en entrenamientos pasado y ver si ha mejorado su técnica con respecto a los golpesos que ha realizado recientemente.

4.3.4. Visualización de objetivos y puntuación

En función de las conclusiones que obtenga el usuario con la visualización de las diferentes gráficas y comparaciones de golpeo, éste tiene la posibilidad de crear objetivos a cumplir para ir mejorando su técnica poco a poco. En la Figura 3.30 se muestra un calendario con los objetivos que se han cumplido con su fecha de consecución y los objetivos que aún quedan por cumplir con su fecha de creación.

Además de este calendario se tiene la posibilidad de ver estos objetivos en función del tipo de golpeo y lo que se quiera mejorar. En la Figura 4.20 se muestra un resumen de los objetivos que faltan por cumplir y los que ya están cumplidos se muestran en forma de listado. Esta funcionalidad permite al usuario establecer metas a conseguir con el fin de ir mejorando su técnica poco a poco.

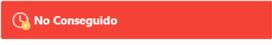
Objetivos Globo		
 Alvaro 20/06/19		Aceleracion Z 17863g
 Alvaro 01/06/19		Aceleracion X 13698g
 Alvaro 10/10/19		Aceleracion Y 14789g
 Alvaro 10/10/19		Aceleracion Z 13574g
 Alvaro 10/10/19		Aceleracion X 14789g
 Alvaro 10/10/19		Aceleracion Y 12781g

Figura 4.20: Listado de objetivos.

Para incentivar al usuario a seguir utilizando el sistema, se ha desarrollado un sistema de puntos basado en retos que el usuario tiene que ir consiguiendo con el uso del sistema *web* y el paso de los entrenamientos realizados. Estos puntos están clasificados en tres tipo de niveles, desde el nivel básico hasta el nivel experto. En la Figura 3.33 se muestra los puntos totales que se han de conseguir en cada nivel para obtener la mayor puntuación. A continuación, se exponen los retos estipulados para la consecución de la puntuación en las Figuras 4.21, 4.22, 4.23.



Figura 4.21: Listado de retos a conseguir nivel básico.



Figura 4.22: Listado de retos a conseguir nivel intermedio.



Figura 4.23: Listado de retos a conseguir nivel experto.

4.4. COMPETENCIAS ESPECÍFICAS DE INTENSIFICACIÓN TRABAJADAS DURANTE ESTE PROYECTO

A lo largo de este proyecto se han trabajado competencias específicas de la intensificación que he cursado: **Tecnologías de la información**. Estas competencias son las siguientes:

- **Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.** Es este proyecto se ha desarrollado un prototipo *hardware* con la ayuda de los componentes necesarios para obtener los datos relativos de un deportista a la hora de realizar un golpeo sobre la pelota. Se ha hecho un estudio de los diferentes módulos y dispositivos a utilizar para desarrollar un sistema con el menor coste posible y manteniendo la calidad deseada. Por otro lado, se ha realizado una selección y despliegue de una tecnología web adecuada para el desarrollo del sistema *web* que ha permitido la consecución de uno de los objetivos principales de este proyecto.
- **Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.** Se ha diseñado una interfaz entre el deportista y el dispositivo de medición basado principalmente en el reconocimiento de gestos. Con esto se ha buscado dotar al sistema de la mayor funcionalidad posible, con el menor número de componentes físicos. Se ha desarrollado una interfaz teniendo en cuenta dos requisitos importantes como son la facilidad de uso del dispositivo y un peso reducido que no dificulte la actividad física del deportista.
- **Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y de las comunicaciones.** En este proyecto no se ha estudiado las necesidades de una empresa como tal, pero sí se ha realizado un estudio en profundidad sobre las tecnologías existentes en los deportes de raqueta para comprender aún más sus necesidades. Con los medios *hardware* empleados y el *software* utilizado se ha ocupado parte de este vacío.
- **Capacidad de concebir sistemas, aplicaciones y servicios basados en las tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios iterativos y computación móvil.** Se ha desarrollado un sistema *web* en el cual el usuario puede volcar los datos registrados por el prototipo de medición. Este portal cuenta con una representación multimedia de estos datos para poder visualizar de manera clara y concisa las estadísticas de los entrenamientos y golpesos de un usuario y así poder evaluar su rendimiento. Tras esta evaluación el usuario puede hacer una valoración de su técnica y poder mejorarla posteriormente.

CAPÍTULO 5

CONCLUSIONES

En los últimos años las nuevas tecnologías se han integrado en diferentes deportes permitiendo una mayor profesionalización, tanto en el ámbito profesional como en el amateur. En gran medida, las nuevas tecnologías han permitido a los deportistas del mundo del deporte, entrenar mejor en el mismo tiempo. El pádel y otros deportes relacionados con la raqueta son un buen ejemplo de esto, ya que se los usuarios de estos deportes buscan mejorar cada vez más en su juego con el fin de poder alcanzar grandes metas y convertirse en grandes jugadores. El pádel es un deporte que está en constante crecimiento y cada vez más son los jugadores que practican este deporte. Con ello, el nivel de juego se va incrementando considerablemente, con lo que los jugadores que se inician en la práctica de este deporte buscan una mejora de la técnica lo más rápido posible para poder competir a niveles mayores. Este proyecto proporciona la monitorización individual acerca del golpeo sobre la pelota que realiza el deportista que no proporcionan otros dispositivos o aplicaciones actualmente en el mercado, por lo que el uso de la herramienta desarrollada a lo largo de este proyecto realmente mejora la técnica del deportista en su golpeo.

En este proyecto se ha diseñado un prototipo para la medición de datos y un sistema *web* para la representación y el análisis de los datos que registra el prototipo. De esta forma:

- El proyecto ayuda a monitorizar el golpeo que realiza el deportista sobre la pelota y tomar conciencia de todos los puntos débiles para reforzarlos.
- Y ayuda a mejorar la técnica de golpeo del deportista.

El prototipo *hardware* se encarga de tomar datos de los sensores que tiene conectados y, posteriormente, almacenarlos en un módulo de almacenamiento. Los datos que recoge el prototipo son los siguientes:

- Datos de aceleración lineal en los ejes de coordenadas X, Y y Z en el espacio.
- Ángulo utilizado en el golpeo de la pelota en los ejes de coordenadas X, Y y Z en el espacio.

La interfaz que se ha desarrollado en el prototipo *hardware* se encarga de facilitar la interacción entre el prototipo y el deportista que lo utiliza. Esta interfaz está basada en el lenguaje gestual y visual para facilitar la interpretación del deportista. El lenguaje gestual es utilizado para que el usuario pueda dar ordenes al prototipo sin entorpecer su actividad deportiva y de la manera más eficiente posible. A través, del uso de leds, esta interfaz ayuda al deportista a saber en cada momento qué acción está seleccionada en el prototipo. Además, se ha diseñado la interfaz con el menor número de componentes posible, lo que facilita la práctica deportiva cuando se porta el

dispositivo. Para conseguir una adherencia eficaz del prototipo a la muñeca del deportista, se ha diseñado una correa que se ajusta a la muñeca del deportista, permitiendo su fuerte sujeción.

El sistema *web* que se ha desarrollado se encarga de representar los datos registrados por el prototipo *hardware*. Para representar estos datos se han utilizado diversas técnicas de representación y se han implementado diversas funcionalidades para ayudar al usuario a mejorar su técnica. Este sistema permite la visualización de estadísticas acerca de cada entrenamiento y golpeo realizado, así como las estadísticas generales. El usuario dispone de una funcionalidad basada en objetivos. Esto motivará al usuario a seguir manteniendo la práctica deportiva y seguir mejorando con el cumplimiento de dichos objetivos. Además el usuario podrá comparar sus golpes con el de jugadores profesionales, para así obtener un *feedback* más real acerca de como está realizando el golpeo.

Tal como se muestra en el capítulo de Resultados (Capítulo 4) se ha demostrado el correcto funcionamiento de todos los módulos que conforman el sistema. Con ello, se concluye, que los objetivos propuestos inicialmente (definidos en el Capítulo 2) se han cumplido satisfactoriamente.

Actualmente existen sistemas de características similares en lo que a monitorización sobre el golpeo que realiza un jugador sobre la pelota se refiere. Este proyecto tiene una utilidad real en deportes de raqueta (en este proyecto se ha centrado la atención en el pádel porque en el mercado actual existe la demanda de un producto que ofrezca la funcionalidad propuesta).

Un punto clave a destacar en este proyecto es la novedad del mismo. En el presente proyecto, a diferencia de otros sistemas similares, se monitoriza el ángulo de golpeo a la hora de ejecutar éste, además de la aceleración. Con esto se puede observar y analizar con detalle la posición de la muñeca en cada tipo de golpeo. El proyecto permite comparar los entrenamientos realizados por un deportista con entrenamientos previos propios o con entrenamientos de otros usuarios. Además, esta comparación se puede realizar a través de vídeos en los cuales se muestra la ejecución de un deportista y se compara con la de un deportista profesional. El proyecto desarrollado cuenta con un sistema basado en objetivos y retos, con los que el deportista puede tener una idea clara de su nivel y progresión. La suma de todo ello facilita la mejora de la técnica del deportista.

También cabe destacar, su **multidisciplinaridad** ya que a lo largo de este proyecto se han trabajado distintas áreas ya que se ha construido un prototipo *hardware* basado en un microcontrolador y se ha equipado con varios sensores y módulos. También se ha diseñado y desarrollado una interfaz gráfica para que el usuario - deportista pueda interactuar con el prototipo *hardware*. Por último se ha desarrollado un sistema *web* siguiendo una arquitectura cliente-servidor.

A lo largo de este proyecto he aplicado y sobre todo afianzado conocimientos adquiridos a lo largo del Grado en Ingeniería Informática. Por ejemplo, he aplicado muchos conocimientos adquiridos en las asignaturas específicas de la intensificación de Tecnologías de la Información, como son los relacionados con el diseño de interfaces y sistemas *web*. También han sido útiles los conocimientos de asignaturas relacionadas con la Ingeniería del *Software* para plantear una forma de trabajo basada en un modelo iterativo e incremental. La asignatura de base de datos para el correcto desarrollo del SGDB implementado en este proyecto. La asignatura de programación de computadores para la implementación de la lógica del prototipo *hardware*. Por último, en este proyecto he adquirido nuevos conocimientos relacionados con los microcontroladores, el aprendizaje sobre un sistema gestor de base de datos nuevo para mi como es Oracle y el aprendizaje de una nueva herramienta de desarrollo de sistemas *web* como es Oracle APEX.

5.1. TRABAJO FUTURO

Las líneas de trabajo futuro más destacables para continuar con este proyecto son:

- Mejora del algoritmo de medición de golpeo. Se proponen dos formas de mejorar el algoritmo de captación de datos basado en la aceleración y ángulo de golpeo:
 - El algoritmo se podría mejorar de tal manera que el usuario no indicara al prototipo que golpeo se va a realizar. Esto consistía en hacer un análisis en tiempo real, para que el propio dispositivo detecte que golpeo se ha realizado en ese periodo de tiempo. Con esto se ahorraría aún más tiempo en las indicaciones que le tiene que hacer el deportista al dispositivo y entorpecería mucho menos la práctica del deportista.
 - Añadiendo otro prototipo *hardware* e incorporarlo en el código del deportista. Con esto se tendrían dos dispositivos *hardware*, que se comunicarían entre sí, para obtener datos muchos más concretos acerca de la posición de muñeca y codo que emplea el deportista a la hora de realizar el golpeo de la pelota.
- Diseño de una PCB para la disminución de cableado en el dispositivo. Esto disminuiría el peso del dispositivo y sería aún más manejable para el usuario.
- Minimizar el tamaño del dispositivo físico y ajustarlo aún más. En el prototipo desarrollado en este proyecto, se ha usado Arduino UNO como microcontrolador. Este microcontrolador tiene un tamaño un poco grande, por lo que sería mejor hacer uso de un microcontrolador más pequeño como sería el Arduino Pico o el Arduino Nano. De igual forma habría que buscar sensores más pequeños para integrarlos todos en la PCB.

ANEXO A

ANTECEDENTES

En este anexo se presentarán algunos conceptos básicos del contexto en el que se enmarca el proyecto.

Se hace un estudio de los sistemas similares que existen en la actualidad y que comparten algunos de los objetivos con la problemática que se quiere abordar en este TFG. En el mercado se pueden encontrar distintas aplicaciones y dispositivos que pueden tener una relación con el objetivo principal del presente proyecto, la monitorización para la mejora de la técnica de golpeo en deportes de raqueta.

A.1. MICROCONTROLADORES

Un microcontrolador [12] es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria, que por regla general incluye un microprocesador o *Central Processing Unit* (CPU), una cierta cantidad de memoria RAM y elementos de conversión de señal analógica a digital y viceversa.

Los microcontroladores se usan para ejecutar pequeñas tareas específicas y normalmente se encuentran ligados a otro dispositivo. Estas tareas pueden ser de control de algún dispositivo como cámara de vídeo, o de alguna funcionalidad concreta dentro de un teléfono móvil, por ejemplo.

A.1.1. Raspberry Pi

Raspberry Pi [8] es un microcontrolador lanzado al mercado por la fundación Raspberry Pi. Raspberry Pi puede ejecutar aplicaciones sobre un sistema operativo, al contrario que Arduino, que ejecuta directamente el *software* sobre el *hardware*.

Raspberry Pi es una placa computadora (SBC) de bajo coste. Se podría decir que es un ordenador de tamaño reducido. Raspberry Pi fue diseñada en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012.

El concepto de Raspberry Pi es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte el funcionamiento básico. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es que capaz de comportarse como tal.

Raspberry Pi tiene un microprocesador *ARM* y un juego de instrucciones *RISC* de 32 bits hasta la versión *Raspberry Pi3*. Desde esta versión en adelante usa un juego de instrucciones *RISC* de 64 bits. También posee un CPU y memoria *SDRAM*.

Entre las características más importantes de Raspberry Pi se destacan:

- Un Chipset Broadcom BCM2835, que contiene un procesador central (CPU).
- Un procesador gráfico (GPU) VideoCore IV.
- Un módulo de 512 MB de memoria RAM.

El principal lenguaje para programar una Raspberry es *Python*, aunque también se pueden usar otros lenguajes como C++, C o Java.



Figura A.1: Ejemplo de Raspberry Pi.

A.1.2. Arduino

Arduino [3] es una plataforma de *hardware* libre, basada en una placa con un microcontrolador y entorno de desarrollo *software*, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El *hardware* de *Arduino* consiste en una placa con un microcontrolador (generalmente Atmel AVR) con puertos de comunicación y puertos de entrada/salida. Por otro lado, el *software* de *Arduino* nos proporciona un software consistente en un entorno de desarrollo IDE que implementa el lenguaje de programación de *Arduino*. Este lenguaje está basado en los lenguajes de programación C y C++.

Las principales ventajas de usar *Arduino* son las siguientes:

- Facilidad de configuración. *Arduino* es un dispositivo *plug-and-play*, por lo que el tiempo de configuración es mínimo.
- Integración con gran cantidad de sensores. Muchos de los sensores utilizados en este proyecto, poseen librerías para interactuar con *Arduino*.
- La programación del *software* se realiza en un lenguaje basado en C y C++, lenguajes conocidos por el estudiante.



Figura A.2: Ejemplo de Arduino.

A.2. DESARROLLO WEB

La *World Wide Web* (www) es un sistema de interconexión de documentos de hipertexto, que pueden ser accedidos a través de Internet, mediante un navegador *web*. El propósito por el que la *web* nació, de manos de Tim Berners-Lee, fue hacer un sistema de comunicación en el CERN más efectivo. La *web* ha pasado por varios cambios, desde el año 1989 hasta hoy en día. A cada uno de estos cambios se les ha llamado versiones de la *web*.

La **web 1.0** hace referencia al primer estado de la *World Wide Web* (www). Durante este primer estado de la *web*, los diseños típicos de las páginas *web* eran páginas estáticas, debido a que la demanda de los usuarios de aquel entonces no requería una alta interacción con la *web*. La mayoría de los casos que podías encontrarte eran sitios *web* que su única función era mostrar información no variable separada en páginas *web* a las que se redirigía con una navegación tradicional. En cuanto al contenido multimedia más avanzado era la reproducción de imágenes en movimiento (GIF), en muchos casos la resolución no era variable, debido a que venía definida por el propio navegador *web* con el que se accedía. Las páginas eran creadas de forma fija y casi nunca se actualizaban, y la forma más usada para enviar información en formularios era mediante el correo electrónico, a pesar de todos los problemas de seguridad que esto podía provocar.

La **web 2.0** la entendemos como el estado que continúa a la *web* tradicional ó 1.0, cuya principal característica es que la participación de los usuarios deja de ser pasiva para convertirse en una participación activa, permitiendo contribuir en la red, creando y dando soporte a una sociedad virtualizada que facilita la comunicación y genera conocimiento y contenido. Esta segunda etapa, se caracteriza por la aparición masiva de *websites* con páginas *web* dinámicas, es decir, que su contenido puede cambiar de la manera frecuente.

Por último, la **web 3.0** la cual surge a partir del año 2016, se conoce como la «web ejecutable». La idea detrás de la *web* 3.0 es definir una estructura de datos y enlazarlos entre ellos para hacer más efectivo el descubrimiento, la automatización y la integración y reuso de los mismos en varias aplicaciones. A este marco de compartición y reuso de datos se le llama «web semántica». Esta *web* semántica, permite a las máquinas «entender» y responder a las peticiones humanas basándose en su significado.

A.2.1. Lenguajes de desarrollo web

HTML5

HTML (*Hypertext Markup Language*) es un lenguaje de marcado usado para estructurar y presentar el contenido para la *web* [9]. HTML es uno de los aspectos fundamentales para el funcionamiento de las diferentes páginas *web*. Con HTML5, los navegadores pueden saber cómo mostrar una

determinada página *web*, saber dónde están los elementos, dónde poner las imágenes y dónde ubicar el texto.

Con la creciente demanda de contenido interactivo en las páginas *web*, HTML5 introduce APIs para la creación de estas *webs*. Las APIs estandarizan tareas que tradicionalmente requerían de contenido propietario o programación personalizada para llevar a cabo dichas tareas. Las APIs más importantes de la especificación HTML5 tienen que ver con la manipulación de vídeo y audio, así como la sincronización multimedia y los subtítulos. Otras APIs importantes se usan para manipular el historial del navegador (avanzar, retroceder y manipular el contenido de dicho historial), para permitir que ciertos recursos de una *web* estén disponibles de forma offline y para añadir ciertos eventos como los drag and drop.

Las hojas de estilo o CSS fueron introducidas con el propósito de separar el diseño del contenido. CSS no existió hasta que pasaron unos años desde el nacimiento de HTML y desde entonces se ha convertido en la forma estándar de definir la capa de presentación de una página *web*. CSS3 se trata de una evolución natural de sus versiones anteriores, CSS2 y CSS1 e introduce efectos visuales, como sombras que dan la apariencia de un realce del elemento, esquinas redondeadas, gradientes y otros.

JavaScript

JavaScript es el lenguaje usado para la programación *web* [10]. Se trata de un lenguaje de alto nivel, interpretado, débilmente tipado (la declaración de una variable no exige la asociación de un tipo de datos) y orientado a objetos y complementa a HTML (que especifica el contenido de una *web*) y CSS (que establece la presentación de la *web*), definiendo el comportamiento de la *web*. El nombre de *JavaScript* nada tiene que ver con Java, siendo dos lenguajes completamente distintos.

Los usos más importantes de *JavaScript* son:

- Desarrollo de sitios *web* del lado del cliente.
- Desarrollo de tipo de aplicaciones gracias a la plataforma NodeJs.
- Desarrollo de aplicaciones para dispositivos móviles.
- Desarrollo de aplicaciones de escritorio para sistemas *Windows*, *Linux* y *Mac*.

A.3. GESTOR DE BASE DE DATOS

Una base de datos es una colección de datos. Podría, por ejemplo, contener información sobre una empresa en particular. Un sistema gestor de base de datos (SGDB) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a estos datos. El objetivo principal de un SGDB es proveer un entorno que sea conveniente y eficiente para usar recibiendo y almacenando información en una base de datos.

Los sistemas de base de datos están diseñados para manipular grandes cantidades de información. Por ello, deben garantizar la seguridad de la información almacenada, a pesar de que el sistema pueda tener algún problema o acceso no autorizado. Si los datos están compartidos entre algunos usuarios, el sistema debe mantener la consistencia.

Los criterios para elegir el SGDB adecuado deben tener en cuenta que el sistema esté o pueda estar integrado con otro *software*, que sea escalable y adaptarse a un posible crecimiento del sistema, que sea sostenible y rentable. Por otra parte, tiene gran importancia la seguridad de los datos almacenados, la funcionalidad que nos permite cumplir con los requisitos y necesidades y la facilidad de uso para el usuario.

A.3.1. Sistemas gestores de bases de datos

MySQL

MySQL [13] es un sistema encargado de la administración de bases de datos relacionales. Las bases de datos relacionales archivan los datos en diferentes tablas, en lugar de almacenar todos los datos en un único archivo. Permitiendo una mejora de la eficiente y flexibilidad. Entre las tablas se definen relaciones que permiten combinaciones de datos de diferentes tablas.

Está desarrollado por *Oracle Corporation* y se considera la base de datos de código abierto con mayor popularidad y la segunda más popular en general junto a Oracle y Microsoft SQL Server, en el sector del desarrollo web.

Como MySQL es un *software* de código abierto, significa que cualquier persona puede usarlo y modificarlo. El código fuente de MySQL lo puede descargar cualquier persona y usarlo de forma gratuita. El código fuente puede ser estudiado y modificado para que se ajuste a las necesidades del usuario.

Microsoft SQL Server

Es un sistema gestor de base de datos relacional desarrollado por Microsoft [20]. Como servidor de base de datos, es un producto *software* cuya principal función es almacenar y recibir datos según las peticiones hechas por otras aplicaciones que se estén ejecutando en el mismo ordenador o en otro ordenador conectado a la red.

Microsoft ha sacado al mercado decenas de ediciones de Microsoft SQL Server, cada una con el objetivo de llegar a diferentes audiencias para cargas de trabajo que van desde pequeñas aplicaciones de una sola máquina hasta grandes aplicaciones orientadas a Internet con muchos usuarios simultáneos.

Las principales características de SQL Server son:

- Permite transacciones.
- Se pueden crear procedimientos almacenados.
- Tiene una interfaz gráfica para la administración, en la que se pueden utilizar comando DDL y DML de manera gráfica.
- Da la posibilidad de trabajar en el modo cliente - servidor, almacenando información y todos los datos en el servidor y el cliente tiene acceso a la información.
- Se pueden administrar datos e información de otros servidores.

PostgreSQL

PostgreSQL [18], también conocido como Postgres, es un sistema gestor de bases de datos relacional orientado a objetos (ORDBMS) que le da un gran énfasis al cumplimiento de los estándares. Como servidor de bases de datos, su principal función es almacenar datos de forma segura y devolver estos datos cuando existan peticiones de otras aplicaciones *software* que los requieran. Puede manejar cargas de trabajo desde aplicaciones pequeñas hasta grandes aplicaciones orientadas a Internet con muchos usuarios conectados de manera concurrente. Para los servidores MacOS, PostgreSQL es el SGBD predeterminado aunque también esta disponible para Microsoft Windows y Linux.

Como otros muchos proyectos, PostgreSQL es de código abierto y su desarrollo no se lleva a cabo por una empresa o persona, sino que es realizado por una comunidad de desarrolladores que trabajan de forma desinteresada y libre. Dicha comunidad se llama PostgreSQL Global Development Group.

Oracle database

Oracle es una herramienta cliente-servidor para la gestión de Bases de datos relacional.

Una BD *Oracle* tiene una estructura física y una estructura lógica que se mantienen separadamente.

- La estructura **física** se corresponde a los ficheros del sistema operativo. Estos ficheros contienen diferente información y su uso es imprescindible para el buen funcionamiento de la BD.
 - **Fichero de datos (*Datafiles*):** Contienen toda la información de la BD.
 - **Ficheros *Redo Logs*:** Los ficheros *redo logs* guardan todos los cambios hechos en los datos y permiten volver a aplicarlos en caso de caída de la BD.
 - **Ficheros de control (*Control Files*):** Contienen toda la información acerca de la bd. Sin este fichero la BD no puede arrancar. Contiene información acerca de: nombre de la base de datos, nombres de *tablespaces*, nombre y localización de ficheros de control, nombre de secuencia del redo log, información de archivado e información de backup.
- La estructura **lógica** está formada por:
 - ***Tablespaces*:** Un *tablespace* contiene uno o mas ficheros de datos del sistema operativo. Dicho esto, se puede decir que de forma lógica, un *tablespace* almacena información.
 - **Objetos de un esquema:** Cuando se habla de objetos en la base de datos, hablamos de tablas, vistas, índices, vistas materializadas, etc.

Estructura lógica

Cómo se ha indicado anteriormente, una base de datos se divide en unidades de almacenamiento lógicas: *Tablespaces*.

Cada base de datos estará formada por uno o mas *tablespaces*. Al menos existe un *tablespace* **SYSTEM** que es el encargado del catálogo del sistema. Cada *tablespace* se corresponde con uno o más ficheros de datos (**datafiles**).

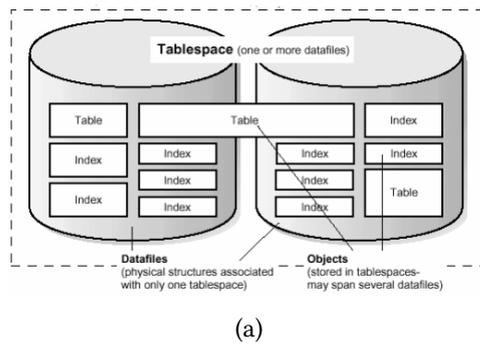


Figura A.3: Estructura lógica base de datos Oracle

Oracle define **esquema** como la colección de objetos y estructuras lógicas que corresponden directamente a los datos almacenados, y crea un nuevo esquema por cada usuario que crea objetos en la base de datos. Dicho esto, se podría definir esquema, como el conjunto que engloba usuario y objetos de ese usuario.

El control del uso del espacio del disco se obtiene mediante las **estructuras lógicas de almacenamiento**: bloque de datos, extensión y segmento.

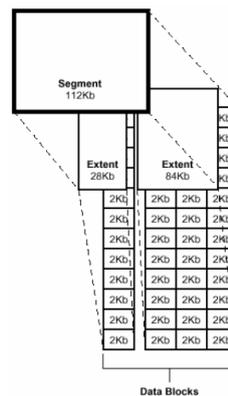


Figura A.4: Estructura lógica de almacenamiento

El nivel más pequeño de granularidad es el del **bloque de datos**: número específico de bytes contiguos de espacio físico en el disco. Tiene un tamaño mínimo de 2K. El siguiente nivel es el de **extensión**, que es un número específico de bloques de datos **contiguos** en el disco. Por último, el **segmento** es un conjunto de extensiones utilizadas para almacenar alguna estructura lógica.

Estructura física

Una base de datos tiene uno o mas ficheros de datos (**datafiles**). Estos ficheros son de tamaño fijo y se establecen en el momento en que se crea la base de datos o en el momento que se crean tablespaces. Los datos del fichero de datos son leídos cuando se necesitan y situados en una caché de memoria compartida llamada **SGA** (*System Global Area*), para que el próximo acceso a los mismos sea más rápido.

El conjunto de ficheros **redo logs** sirven para registrar todos los cambios (*insert, update, delete, create, alter o drop* sobre la base de datos y poder recuperarla ante un error.

Los **ficheros de control** almacenan información de la estructura física de la base de datos.

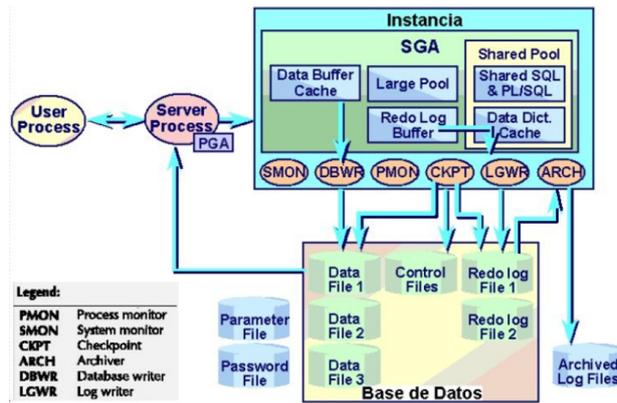


Figura A.5: Arquitectura Oracle.

En la Figura A.5 se puede observar un esquema general de una instancia de base de datos Oracle.

Los distintos elementos que forman parte del entorno de **memoria** Oracle son los siguientes:

- **SGA (System Global Area):** Zona principal de la memoria de Oracle. Está dividida en varias subareas desempeñando cada una de estas una tarea totalmente distinta:
 - Shared pool: Compuesta por la **library caché** y la **dictionary caché**.
 - Dictionary caché: Mantiene datos de sus propias tablas y vistas ya que accede constantemente a ellas al ejecutar cualquier sentencia.
- **PGA (Program Global Area):** Destinada a guardar información de los procesos de usuario y procesos de background que corren en una instancia de la base de datos y que a través de distintos procesos intercambian la información con la SGA.

A.4. DISPOSITIVOS DE MONITORIZACIÓN EN DEPORTES DE RAQUETA

A.4.1. Smart tennis sensor

Smart tennis sensor se trata de un sensor que se coloca en el mango de la raqueta. Este sensor pesa solo 8 gramos y tiene una protección antigolpes. El sensor cuenta con una batería que tiene una autonomía de 90 minutos, lo que permite completar un entrenamiento completo. Este sensor es capaz de recoger datos acerca de:

- Velocidad de la bola. El sensor registra la velocidad de la bola tras el golpeo. Este es un dato que podemos registrar en el saque de inicio, por ejemplo.
- Zona de impacto de la pelota. Algo muy importante a la hora de analizar por qué se falla un golpe. Esto ayuda también a crear un mapa de calor de la zona con la que se golpea a la bola según el tipo de golpe que se ha realizado.

- Velocidad de la raqueta. Algo fundamental para analizar los golpes que se han ejecutado según sean *drive* o revés, que son lo más habituales. El *Smart Sensor* identifica que tipo de golpeo se ha efectuado.
- Número de golpesos que se han realizado en una sesión.

Todos los datos recogidos por el sensor en una sesión son almacenados y ordenados en una app Adroid o IOs para poder ser consultados, de manera que se pueda llevar un diario de entrenamientos muy completo.



Figura A.6: Smart Tennis Sensor.

A.4.2. Raquetas inteligentes

En 2016 se instauró una nueva tecnología en las raquetas de tenis de los deportistas. Esta nueva tecnología fue desarrollada por Babolat, la cual consiste en la integración de una serie de sensores en la raqueta. Estos sensores están colocados en la empuñadura de la raqueta para registrar todos los datos técnicos de cada golpe que realiza el tenista. En la empuñadura de la raqueta se encuentran dos botones. En el primero, cuando se presiona el encendido, aparece una luz azul que permite jugar el partido. Cuando finaliza, se pulsa el segundo botón para conectar el *bluetooth* y sincronizar los datos con el dispositivo que contiene la aplicación que registra los datos. Al final del entrenamiento, la información puede descargarse en un *smartphone*, tableta u ordenador para ser analizada después y ver los errores y golpes correctos que se han llevado a cabo. Este tipo de sensores implantados en la propia raqueta son capaces de medir la aceleración de golpeo y la velocidad de la bola.

La Federación Internacional de Tenis prohibió el uso de este tipo de tecnologías para el análisis de juego, pero en enero de 2016 se decidió cambiar esta normativa y permitir el uso de dispositivos inteligentes, como la «raqueta inteligente».

A.4.3. Babolat Starter

Babolat Starter consiste en una pulsera inteligente que se coloca en la muñeca. Esta pulsera es capaz de registrar datos acerca del juego que realiza un deportista. La pulsera recoge datos como la aceleración con la que se ejecuta el golpeo. Todos los datos que registra esta pulsera pueden ser descargados en una app para móvil. En esta aplicación se pueden visualizar todas las estadísticas referentes al entrenamiento realizado.

La pulsera Babolat cuenta con una batería que tiene una autonomía aproximada de 90 minutos. Además de registrar valores acerca de la aceleración, la pulsera es capaz de registrar las pulsaciones por minuto del deportista que está realizando la actividad física.



Figura A.7: Babolat Starter.

A.4.4. Sensor Head Zepp Tennis

Este dispositivo se ha diseñado para registrar y analizar los movimientos en la pista de un deportista con el fin de ayudarle a mejorar en su rendimiento. Este sensor cuenta con cuatro modos de utilización:

- Juego, mediante el cual permite registrar los tipos de golpe, la velocidad de golpe, los efectos y el punto dulce.
- Entrenamiento, mediante el cual se puede acceder a una amplia gama de entrenamientos creados por Head para ayudar al deportista a mejorar su juego.
- Competición, que permite retar a rivales o amigos y comparar el rendimientos de éstos con el del deportista.
- Servicio 3D, a través del cual se puede analizar la técnica empleada en el saque.

Este sensor se puede acoplar a la mayoría de empuñaduras de raqueta. Cuenta con un peso de solo 7 gramos y es fácil de instalar en la raqueta.



Figura A.8: Head Zepp Tennis.

A.4.5. Babolat PIQ

Este sensor se considera un *wearable* muy ligero y portátil, que proporciona al jugador de tenis información de su juego a través de varios parámetros. Este sensor también permite recibir información a tiempo real en el dispositivo móvil conectado. El sensor permite registrar los datos específicos de los golpes de *drive*, potencia, velocidad de la muñeca y velocidad de servicio. El dispositivo cuenta con una memoria *flash* que registra aproximadamente 6000 golpes, un procesador ARM M4, un módulo *bluetooth* para el traspaso de información y una batería con una autonomía de 2 horas.



Figura A.9: Babolat PIQ.

A.4.6. Zepp Tennis

Este sensor se coloca en el mango de la raqueta de tenis. El dispositivo registra las estadísticas de pista y prácticas. Permite comparar con otros jugadores para obtener, de forma instantánea, un conocimiento más profundo de tu rendimiento. El sensor usa una tecnología Smart Capture que permite grabar clips de cada golpe. El dispositivo permite registrar hasta 1000 puntos y enviar los datos al *smartphone* del deportista para su posterior análisis. El sensor evalúa las métricas de rendimiento clave para cada golpe, incluida la velocidad de la pelota, la velocidad de la muñeca y el punto dulce. Además, cuenta con funcionalidades que permiten registrar las calorías quemadas en un entrenamiento.



Figura A.10: Zepp Tennis.

A.4.7. Babolat Play

Babolat Play es una aplicación en la cual se puede ver y analizar toda la actividad deportiva realizada por el deportista. Esta aplicación cuenta con varias funcionalidades para visualizar

todos los datos relativos a un entrenamiento. Las diferentes funcionalidades que ofrece son:

- Visualizar tiempo de juego empleado en el entrenamiento.
- Visualizar estadísticas sobre golpes de derecha y revés.
- Visualizar estadísticas sobre servicios.
- Visualizar estadísticas sobre la potencia empleada en cada golpeo.
- Visualizar número de golpes realizados en un entrenamiento.
- Visualizar el mayor número de golpes realizados en un intercambio de pelota.
- Visualizar velocidad de servicio en km/h.
- Visualización del punto dulce de la raqueta.



Figura A.11: App Babolat.

A.5. COMPARACIÓN DE SISTEMAS DE MONITORIZACIÓN EN DEPORTES DE RAQUETA

En la Tabla A.1 se resumen las diferentes funcionalidades que ofrecen los sistemas que se han expuesto anteriormente frente al sistema desarrollado en el presente proyecto. Se observa que la mayoría de los sistemas que existen en el mercado permiten registrar la aceleración y la velocidad de la pelota. En el sistema desarrollado en este proyecto se ha contemplado además, la medición del ángulo con el que se ejecuta el golpeo, ya que es un dato importante y que puede ayudar al usuario a mejorar la técnica de su golpeo. Se observa que solo uno de los sistemas propuestos es capaz de hacer una distinción sobre el tipo de golpeo que se está realizando. En el sistema desarrollado en el presente proyecto se ha optado por la distinción de los tres golpes principales del pádel como son el *drive*, globo y volea. Cabe destacar la funcionalidad desarrollada de comparación de golpes, mediante la cual el usuario puede comparar sus golpes con los de un jugador profesional o con los suyos propios para ir visualizando su progresión. Por último, a diferencia de los demás sistemas, se ha desarrollado una funcionalidad basada en objetivos y retos a través de la cual el usuario puede medir su nivel y progresión de su técnica.

Sistema de medición	Aceleración	Ángulo	Detección de punto dulce	Distinción de golpes	Comparación de golpes	Velocidad de la bola
Smart tennis sensor	Si	No	Si	No	No	Si
Raquetas inteligentes	Si	No	No	No	No	Si
Babolat starter	Si	No	No	No	No	No
Sensor head zepp tennis	Si	No	No	Si	No	Si
Babolat PIQ	Si	No	No	No	No	Si
Zepp tennis	Si	No	SI	No	No	Si
I-Racket	Si	Si	No	Si	Si	No

Tabla A.1: Resumen de sistemas de monitorización en deportes de raqueta.

BIBLIOGRAFÍA

- [1] C. Amieba y J.J Salinero. «Overview of paddle competition and its physiological demands». En: *AGON International Journal of Sport Sciences* (2013), págs. 60-67.
- [2] *Arduino Uno Rev3*. <https://store.arduino.cc/arduino-uno-rev3>. 2019.
- [3] Óscar Torrente Artero. «El mundo GENUINO-ARDUINO». En: 2017.
- [4] M.A. Campos Vázquez. «Monitorización de respuestas físicas y fisiológicas al entrenamiento y la competición en fútbol». En: *Universidad Pablo de Olavide* (2015).
- [5] *Cómo se ha transformado el deporte con la tecnología*. <http://www.impulsodigital.elmundo.es/sociedad-inteligente/como-se-ha-transformado-el-deporte-con-la-tecnologia>. 2018.
- [6] J. Courel Ibáñez y col. «Evolution of padel in Spain according to practitioners gender and age». En: *Facultad de ciencias del deporte, federación española de pádel* (mar. de 2016).
- [7] *Deporte y tecnología: 4 avances que lo cambiaron todo*. <https://www.redbull.com/es-es/tecnologia-avances-deporte>. 2018.
- [8] Gareth Halfacree Eben Upton. «Raspberry Pi. Guía de usuario». En: 2016.
- [9] Juan A. Recio García. «HTML5, CSS3 y JQUERY». En: 2016.
- [10] Mircea Diaconescu Gerd Wagner. «Web Applications with Javascript or Java: Volume 2». En: 2019.
- [11] R. Hervás Lucas y A.I. Molina Diaz. «Soporte al diseño: Principios, estándares y guías de estilo». En: *Apuntes de interacción persona-ordenador II*, 2016.
- [12] Calos Ruiz Zamarreño Jesús M. Corres Sanz Cándido Bariáin. «Programación de Microcontroladores PIC en Lenguaje C». En: 2017.
- [13] *MySQL*. <https://www.mysql.com/>. 2019.
- [14] *Oracle Database*. <https://www.oracle.com/es/index.html>. 2019.
- [15] *Ords Oracle*. <https://www.oracle.com/es/database/technologies/appdev/rest.html>. 2019.
- [16] M. Polo, M. Rodríguez e I. Caballero. «Metodologías de desarrollo ágiles: Scrum». En: *Apuntes de ingeniería del software II*, 2016, págs. 6-12.
- [17] M. Polo, M. Rodríguez e I. Caballero. «Procesos de Ingeniería del Software». En: *Apuntes de ingeniería del software II*, 2016, págs. 21-22.
- [18] *Postgresql*. <https://www.postgresql.org/>. 2019.
- [19] B.J. Sánchez-Alcaraz, J. Cañas y J. Courel Ibáñez. «Analysis of scientific research in padel». En: *AGON International Journal of Sport Sciences* (2015), págs. 44-54.
- [20] *SQL Server*. <https://www.microsoft.com/es-es/sql-server/>. 2019.

