



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE
ENTRENAMIENTO ASISTIDO
PARA JUGADORES DE
FÚTBOL BASADO EN IOT**

TESIS

QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECATRÓNICO

P R E S E N T A

ALFREDO HERRERA CORTÉS

DIRECTOR DE TESIS

M.A. LUIS YAIR BAUTISTA BLANCO



Ciudad Universitaria, Cd. Mx., 2022

AGRADECIMIENTOS

A mi familia que me apoyo durante el proceso de desarrollo de este trabajo, a mi mamá que me ha dado cariño, atención y amor durante toda mi vida lo cual me ha permitido desarrollarme adecuadamente; a mi hermana que siempre me ha dado motivación para continuar; y a mi padre que me ha apoyado en mis estudios. Finalmente, también agradecer a mi asesor, Yair Bautista, el apoyo, paciencia y comprensión que me ha prestado durante la realización de este trabajo.

ÍNDICE

1.	PLANTEAMIENTO DEL PROBLEMA	1
1.1.	JUSTIFICACIÓN	2
1.2.	OBJETIVO GENERAL	5
1.3.	ALCANCE	5
2.	ANTECEDENTES	7
2.1.	SISTEMAS DE MONITOREO POR VIDEO.....	8
2.2.	RASTREADORES	9
2.3.	SISTEMAS DE MONITOREO DE ENTRENAMIENTO A TRAVÉS DEL BALÓN.....	12
2.4.	ENTORNOS DE DESARROLLO DE HABILIDADES	13
3.	REQUERIMIENTOS Y ESPECIFICACIONES.....	16
3.1.	IDENTIFICACIÓN DE NECESIDADES	16
3.2.	ESPECIFICACIONES	21
4.	DISEÑO CONCEPTUAL	25
4.1.	GENERACION DE CONCEPTO	25
4.2.	DIAGRAMA DE BLOQUES.....	25
4.3.	GENERACIÓN, SELECCIÓN Y EVALUACIÓN DE CONCEPTOS	27
4.3.1.	SISTEMA DE DETECCIÓN DE POSICIÓN	27
4.3.2.	INTERFAZ DE USUARIO	31
4.3.3.	ELEMENTOS DEL CAMPO	33
4.3.4.	ALIMENTACIÓN	35
4.3.5.	TRANSMISIÓN DE INFORMACIÓN	36
4.3.6.	SISTEMA DE ALMACENAMIENTO	38
5.	DISEÑO DE DETALLE.....	40
5.1.	DETECCIÓN DE POSICIÓN.....	40
5.1.1.	CREACIÓN DE ACCESS POINT Y TOMA DE LECTURAS USANDO EL ESP32 ..	40

5.1.2.	VALIDACIÓN DE WIFI PARA DETECCIÓN DE POSICIÓN	41
5.2.	INSTALACIÓN DEL SERVIDOR.....	51
5.3.	RED	52
5.3.1.	CONEXIÓN WIFI	52
5.3.2.	TRANSMISIÓN DE DATOS	54
5.4.	BASE DE DATOS	58
5.5.	DISEÑO DE INTERFAZ WEB.....	61
5.5.1.	BOCETOS	61
5.5.2.	DISEÑO DE BACKEND	63
5.5.3.	DISEÑO DE FRONTEND.....	72
5.6.	CIRCUITOS ELECTRÓNICOS	85
5.6.1.	TRACKER.....	85
5.6.2.	ACCESS POINT	85
5.6.3.	PORTERÍAS	88
5.7.	DISEÑO FÍSICO DE LOS ELEMENTOS.....	89
5.8.	PROGRAMACIÓN DE DISPOSITIVOS	91
5.8.1.	TRACKER.....	92
5.8.2.	ACCESS POINT	93
5.8.3.	PORTERÍAS	93
6.	RESULTADOS.....	94
6.1.	SUMARIO	94
6.2.	PRUEBA	95
6.3.	ANÁLISIS DE RESULTADOS.....	101
7.	CONCLUSIONES	107
8.	TRABAJO A FUTURO	110
	REFERENCIAS.....	111

1. PLANTEAMIENTO DEL PROBLEMA

El fútbol se ha convertido en una actividad profesional, pasó de ser una actividad meramente lúdica, como lo era a principios del siglo XX, a ser una actividad lucrativa que mueve hoy en día cerca de 500,000 millones de dólares anuales alrededor del mundo [1]. Del mismo modo, es el deporte más popular en el mundo con cerca de 4 000 millones de seguidores [2]. La importancia de este deporte es tal, que permea seriamente en la política de varios países como Brasil donde el resultado del mundial del 2014 cambió el rumbo de las elecciones.[3]

Este mismo aumento en la popularidad y en los ingresos económicos planteó en los años 60's el reto a los equipos profesionales en mejorar su rendimiento y entrenamiento para ganar más torneos [4], pues de esta forma se ingresan más fondos y se consiguen grandes contratos monetarios de patrocinios. Por este motivo se buscó que los jugadores tomaran entrenamientos más intensivos, haciendo que los entrenamientos consumieran más tiempo a los jugadores y desempeñándose de tiempo completo en el fútbol, ya que antes de esto los futbolistas se dedicaban a otras actividades como el trabajo o estudiar una carrera universitaria tal como es el caso de Hugo Sánchez, que jugaba fútbol en la liga profesional mexicana mientras estudiaba [5].

A inicios de la década de los setenta, directores técnicos como Rinus Michels y Johan Cruyff introdujeron en su plantilla a entrenadores físicos, con el fin de proponer ejercicios que mejoraran la condición física de los jugadores. Los entrenamientos consistían en la realización de ejercicios de fuerza, resistencia y velocidad; en ellos rara vez se rodaba el balón. Estos entrenamientos evolucionaron en entrenamientos integrados fusionando carga física con acciones o situaciones que estuvieran presentes en el juego según el puesto del jugador, de manera que poco a poco se fueron introduciendo conceptos tácticos en los entrenamientos.[4]

La necesidad de llevar una recolección de datos más precisa durante el entrenamiento ha detonado la incursión del uso de la tecnología en ellos como: el seguimiento en las características físicas de cada jugador, uso de recursos multimedia para dejar claros conceptos con videos y animaciones, uso de realidad aumentada para entrenamiento de decisiones, etc [6]. Pero debido a su reciente implementación aún quedan varios aspectos por mejorar como lo son la interacción con el usuario, recolección y presentación de datos, evaluación de rendimiento y práctica de habilidades del jugador.

1.1. JUSTIFICACIÓN

La importancia de la preparación de los jugadores juega un papel crucial en las competiciones y partidos que estos disputan; tener a un equipo que no siga las instrucciones del entrenador o que se encuentren en mala condición física es algo impensable en el fútbol actual; un ejemplo claro de lo que pasa cuando un equipo no se encuentra físicamente preparado sucedió en el 2000 con el equipo español F.C Real sociedad que entró a disputar la Champions League, el torneo más importante de clubes de fútbol, esto implicó que jugara más partidos, pero no aguantó este ritmo de juegos y descendió. [4]

Es por ello por lo que se han analizado diferentes opciones de entrenamientos, se ha centrado la atención en observar cuales ofrecen mejoras en rendimiento físico y técnico y cuáles son los que permiten a los jugadores mejorar habilidades relacionadas al fútbol. El interés por los entrenamientos se puede ver reflejado directamente en el número de publicaciones científicas que abordan al fútbol como tema central, como se ve en Fig. 1, las cuales han crecido exponencialmente.

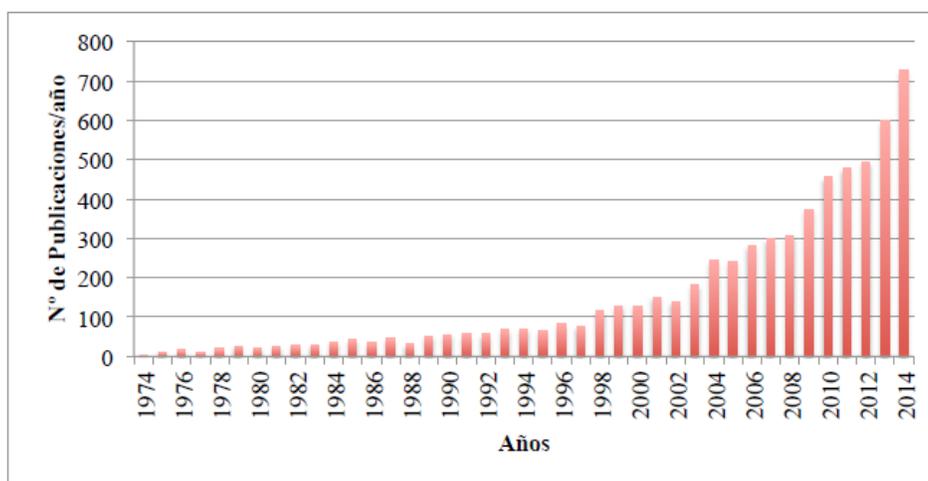


Fig. 1 Número de publicaciones científicas realizadas sobre el fútbol desde 1974 a 2014. [7]

De igual manera, otro factor que ha incentivado mejorar el entrenamiento y el análisis táctico de forma sustancial ha sido el número de goles anotados en cada partido, como se ve en la Fig. 2, el promedio de goles por partido en copas del mundo ha bajado en comparación del 1970 donde rondaba los 3 goles a 2006 donde rondo los 2.23 goles por partido, mientras que por otro lado el porcentaje de goles en jugadas a balón parado fue en aumento, esto significa que un solo gol

puede definir el encuentro de forma determinante [8], y es por esta misma razón que los entrenadores ponen especial énfasis en el entrenamiento táctico .

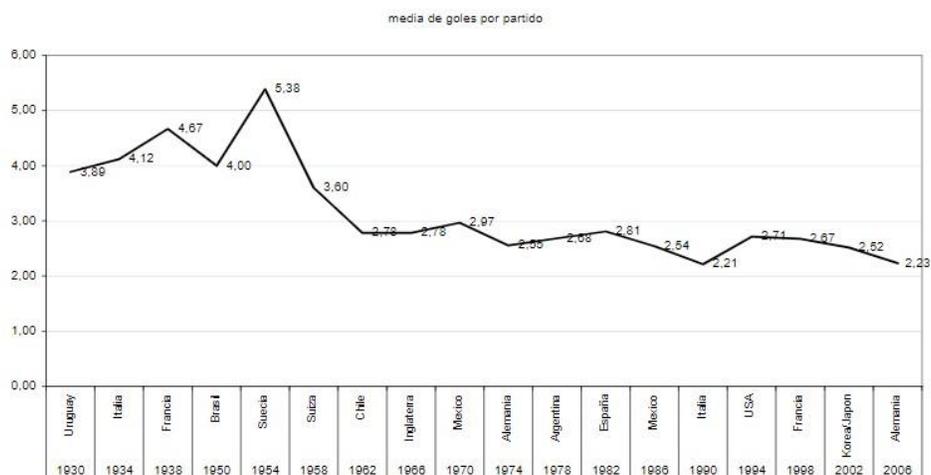


Fig. 2 Media de goles por partido durante los mundiales [9]

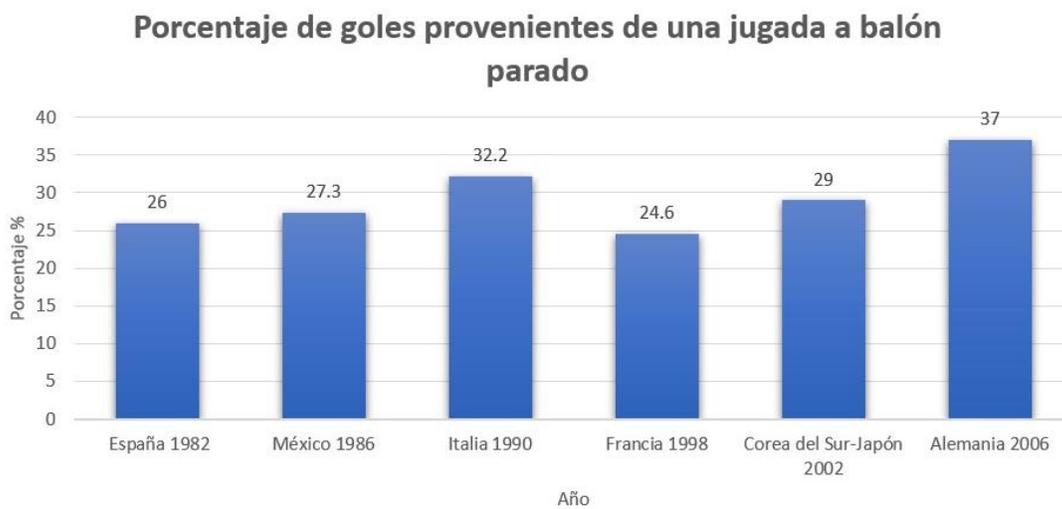


Fig. 3 Porcentaje de goles en jugadas a balón Parado en algunos mundiales de fútbol [10]

En los años setenta se hicieron los primeros esbozos de los entrenamientos, pues anteriormente los jugadores únicamente jugaban partidos libres, no había un control de ejercicios a seguir ni alguien que llevara un seguimiento del desarrollo de habilidades de un jugador; estos primeros esbozos contenían muy ligeros ejercicios de habilidades y se concentraban en trabajos físicos muy

generales y basados principalmente en el atletismo. Poco a poco fueron evolucionando y se introdujeron ejercicios físicos específicamente planeados para el fútbol, además de incluir segmentos tácticos donde se planeaban jugadas muy parecidas a situaciones que se podrían dar en el terreno de juego. Estos entrenamientos se vieron afectados por la tecnología y gradualmente se han propuesto varios sistemas que permiten mejorar el entrenamiento ya sea en la recolección de datos como en la práctica de habilidades deportivas. [4]

La introducción de la tecnología en los entrenamientos fue llevada a cabo en los noventa con sistemas automatizados en recolección de datos, aunque no tomaría relevancia hasta a mediados de la década pasada donde varios entrenadores empezaron a prestar atención e introducirlos a sus equipos. De igual forma, se crearon sistemas que además de tomar datos logran un entorno completamente interactivo, generando dinamismo y experiencias más cercanas a las de los partidos, por ende mejorando las habilidades del jugador; los principales exponentes de estos sistemas son el Footbonaut [6] y el Benfica 360 Lab, pero estos solo suelen ser usados por grandes clubes europeos como el Borussia Dortmund[6], Chelsea o Benfica[11].

La relativa novedad de estos sistemas ha hecho que no se hayan explotado a profundidad; dejando varios aspectos por mejorar, siendo uno de los más relevantes la calidad de los datos entregados, pues en el caso de los sistemas de recolección de datos, los entrenadores opinan que estos datos no son relevantes al momento de evaluar el rendimiento real de un jugador; mientras que los sistemas, para práctica de habilidades, no todos los clubes tienen acceso a ellos ya sea por el costo o por no tener personal capacitado, pues en algunas ocasiones es necesario tener el personal para llevar un control sobre estos sistemas, como el caso del F.C Benfica que cuenta con un equipo de especialistas encargados de llevar el control.[11]



Fig. 4 Entrenamiento supervisado por un técnico en el Benfica Lab 360 [12]

1.2. OBJETIVO GENERAL

Un sistema que ayude a monitorear el desempeño físico y el desarrollo de habilidades de jugadores de fútbol, buscando potenciar los entrenamientos de fútbol para hacerlos más interactivos, se usará una interfaz que permita llevar un control acerca del ejercicio y la forma en la que el jugador lo realiza y se pueda monitorear en tiempo real, además de que la configuración se pueda realizar de una forma sencilla, también se llevará un historial del desempeño de los jugadores pudiendo generar un resumen general del jugador en cada ejercicio

1.3. ALCANCE

En este trabajo de tesis se mostrará el desarrollo de un sistema dedicado al mejoramiento de actividades deportivas en el fútbol, principalmente en recolección de datos y automatización de ejercicios analíticos para llevar un control cuantitativo. Se presentarán las necesidades y especificaciones necesarias de acuerdo con las actividades del entrenamiento de fútbol, además se realizará el diseño conceptual, así como la fabricación de los prototipos de dispositivos que permita la recolección de los datos; del mismo modo, una interfaz que permita llevar un control sobre el entrenamiento, tanto el monitoreo de las actividades en tiempo real, como el almacenamiento para el posterior análisis por el cuerpo técnico.

Los pasos para seguir en la realización de este proyecto están basados en el *Cuaderno de Gestión de Tecnología: Innovación de Productos* [13]; en el cual se puede observar toda una metodología bien explicada para la fabricación de un producto. Esta metodología fue de gran ayuda pues sirvió para organizar mejor el trabajo en cuanto a desarrollo y tiempos. Los pasos que establece el cuaderno son:

- **Identificación de necesidad:** Establecer cuáles son los requerimientos del cliente, a partir de ello identificar cuáles son los elementos indispensables para el usuario. Finalmente se describe el producto.
- **Especificación del producto:** Definición de especificaciones, en esta etapa se pasa a una etapa cuantitativa, se asignan valores a los requerimientos previamente planteados para definir el producto y con base en ello plantear las metas.
- **Diseño conceptual:** En esta etapa se secciona el problema, de forma que es más sencillo organizar el trabajo. De igual forma en esta etapa se realiza búsqueda de información para generar alternativas de solución y seleccionar alguna.

- **Ingeniería de detalle:** En este paso se realiza el diseño metódico de cada una de las partes que compondrán al sistema, se verifican de igual manera aspectos de producción y se realiza la documentación necesaria de las piezas.
- **Prototipos y escalamientos:** Se realiza un prototipo sencillo para verificar de manera tangible el funcionamiento correcto del prototipo.

Los siguientes dos pasos pertenecen al método descrito en el cuaderno de gestión tecnológica pero no entran dentro de los alcances de este trabajo:

- **Producción:** En esta etapa se planea la producción, se realizan pruebas del proceso de producción de tal forma que cumpla con los estándares establecidos. De igual manera se documenta la información para el usuario.
- **Comercialización:** Considerar la comercialización en etapas previas del proceso. Se realiza una estructuración del proceso de comercialización. Aunque este paso no está dentro de los objetivos de este proyecto.

2. ANTECEDENTES

Como se ha mencionado en el capítulo anterior, el uso de tecnologías en el fútbol es reciente, la mayoría proviene de la década de los noventa o a mediados de la primera década de los dos mil donde se empezó a monitorear la actividad física de los jugadores y la medición de su desempeño de una forma cuantitativa y automática, pues antes solo dependía concretamente del juicio del entrenador y el desempeño físico de un jugador se estimaba de manera indirecta. El interés sobre monitorear la actividad física de los jugadores cobró mucha relevancia cuando se estudiaron las diversas formas de acomodar los ejercicios de fuerza y cargas, donde se encontró que una correcta distribución de ejercicios de cargas reduce el riesgo de lesión, además de acelerar el proceso de recuperación de un jugador lesionado.[14]

Siguiendo el mismo orden de importancia, en el fútbol moderno el conocer las habilidades del equipo rival y del propio se volvió una tarea esencial, es por ello por lo que en muchos equipos de élite se creó el departamento de análisis deportivo y scout, en dichos departamentos se tiene gente especializada para analizar jugadas de los equipos rivales, recolectar datos de jugadores y llegando en ocasiones hasta proponer fichajes; finalmente, toda ésta información se reporta al director técnico para que este tome un mejor criterio al momento de decidir la alineación [15], así que la recolección de datos se ha hecho imprescindible, por lo que se han implementado diversos métodos para la recolección de datos como:

- Uso de cámaras
- Sensores wearables o *trackers*
- Sensores embebidos en el balón

De la misma forma se ha puesto énfasis en mejorar las habilidades deportivas como la capacidad de reacción, potencia de tiro y precisión de pases; de este modo, se han creado sistemas o entornos dedicados a mejorar dichas habilidades, pero la creación de estos sistemas es reciente pues corresponde a inicios del año 2010, por lo que sólo algunos clubes de élite lo implementan en sus entrenamientos como el Borussia Dortmund con el Footbonout y el Benfica con el Benfica Lab.

En breve se describirán las características relevantes de cada uno de estos sistemas mencionando sus características, sus puntos fuertes y los puntos débiles.

2.1. SISTEMAS DE MONITOREO POR VIDEO

Uno de los primeros sistemas implementados fue el monitoreo del desempeño de jugadores a través de cámaras de video, como se ve en la Fig. 5. El método consistía en poner en el campo una o varias cámaras durante un partido y dejarlas grabando, estas cámaras se colocaban a una altura promedio de 15 m del suelo y cada una de estas cámaras solo podía realizar el seguimiento de un jugador, además de que se tenían que grabar muestras de control del jugador corriendo a diferentes velocidades para tener datos de calibración que funcionarían en la posterior etapa de análisis. Estas grabaciones eran reproducidas en televisores donde una persona codificaba las acciones de un jugador durante el partido y se registraba la duración de la actividad, finalmente se condensaban todos estos datos para poder obtener el total de duración, así como la frecuencia de cada una de las acciones realizadas por el jugador. Esta forma de tomar datos, aunque fácilmente reproducible, conllevaba una carga importante de trabajo tanto desde el punto de vista de colocación de las cámaras, hasta el tiempo que requería realizar los cálculos aproximados, estos factores hacen difícil su aplicación especialmente cuando se juega un partido por semana o más.[7]



Fig. 5 Ejemplo de monitoreo mediante cámaras [16]

Por lo que estos sistemas evolucionaron en programas informáticos que eran capaces de realizar el cálculo de la distancia recorrida por un jugador de una manera automática, ahorrando valioso tiempo. Ejemplos de estos programas es el sistema Trackperformance lanzado en el año 2004, este programa permitía al usuario seleccionar el seguimiento de un jugador en la toma de forma manual, usando una Tablet y un lápiz tipo *stylus*, se realizaban marcas en el campo como

referencia, éstas ayudaban al usuario a hacer el seguimiento de los jugadores, de este modo se lograba arrojar los datos en tiempo real, lo cual permitía hacer posible el análisis durante el entrenamiento; a pesar de estas mejoras este sistema se encontraba con la limitante de solo hacer seguimiento de un solo jugador. [7]

Fue con programas como AMISCO, que se logró la detección de varios jugadores en tiempo real para el análisis de un partido de forma completa, como ventaja permitía la detección de jugadores y el balón con una frecuencia de 25 Hz, permitiendo una recolección de datos precisa. Este sistema fue usado en la liga española y monitoreó a cada jugador en los estadios que contaban con este sistema, llegando a datos relevantes como que el equipo ganador de la Liga, el Real Madrid, era el que menos metros había corrido de la primera división [17]. Sin embargo, es importante tener en cuenta que este sistema es semiautomatizado, siendo necesaria la supervisión de un humano para comprobar que el seguimiento de jugadores se logra de manera correcta.[7]

Actualmente se encuentran varios programas como Dvideo, Datatrax, TRACAB, KlipDraw que usan algoritmos más sofisticados que permiten la detección del jugador automáticamente. KlipDraw en específico proporciona una interfaz cómoda para el usuario permitiendo no solo realizar el seguimiento y estimaciones del desempeño de un jugador, si no también crear anotaciones y animaciones muy parecidas a las que se harían con un editor de video.[18]

2.2. RASTREADORES

Entre las diferentes categorías, la más relevante y de las más usadas son los rastreadores o *trackers*. Como se vio en la sección anterior, el análisis de partidos y desempeño de jugadores usando cámaras resultó laborioso en un inicio, lo que planteó el uso de alternativas que fueran sencillas de llevar a cabo, por lo que se pensó en dispositivos electrónicos que estuvieran adheridos al jugador, colocados ya sea en la ropa o en alguna parte del cuerpo, con esto se podría transmitir la posición actual del jugador y se podría lograr una frecuencia de muestreo mayor a la obtenida a través del uso de cámaras.

Los primeros sistemas fueron diseñados usando el principio de triangulación, de esta forma se enviaba una señal de identificación emitida por el dispositivo adherido al jugador, *tracker*, y se mide el tiempo de retraso que pasó desde que fue enviado el paquete hasta que llegó a las antenas de control [7], un sistema que ejemplifica este tipo de rastreo es ZXY Sport Tracking, que usa receptores Cordis Eye de Radionor communications, que permite el rastreo del jugador con una precisión menor a un metro y con una frecuencia de 40 Hz [19], teniendo como limitantes la interferencia de las señales al haber muchos jugadores, además de que no se ha establecido su validez y reproducibilidad técnica para registrar movimientos durante partidos.[7]



Fig. 6 Uso ilustrativo del sistema ZXY Sport tracking [19]

La evolución de estos sistemas resultó en *trackers* con tecnologías de GPS. El desarrollo de la tecnología GPS permitió que se pudieran llevar de manera portátil y fueran integrados a los sistemas que se tenían para el rastreo de jugadores. El uso de esta tecnología se encuentra en su auge debido a la entrega de datos de rendimiento a los entrenadores y deportistas [7]. Su principio de funcionamiento es sencillo: el GPS entrega una señal con la posición del jugador la cual es transmitida a algún software, teniendo de esta manera datos en tiempo real de forma muy práctica, además de que, a través del software, se pueden almacenar fácilmente los datos.

Además del uso de GPS se han agregado otros sensores a estos sistemas para mejorar el análisis sobre perfiles de actividad y respuestas fisiológicas de los jugadores. Por ejemplo, el uso de

acelerómetros permite conocer la componente de aceleración en los tres planos y esta se asocia directamente al registro de carga física del jugador [7], teniendo de esta forma una idea más precisa de la fatiga a la que se ha expuesto el jugador durante el entrenamiento; además con los datos arrojados por el *tracker* es mucho más sencilla la realización de mapas de calor que muestran el espacio por donde se movió el jugador, de la misma manera, con algunos de estos dispositivos se mide la frecuencia cardíaca, así como el nivel de oxígeno durante el entrenamiento, permitiendo al cuerpo técnico conocer el estado en el que se encuentran sus jugadores y así tener un mejor criterio para evitar lesiones por sobrecarga. [20]



Fig. 7 chaleco con rastreador para monitoreo de la actividad física de los jugadores.[20]

Los primeros *trackers* GPS funcionaban a frecuencias de 1 Hz, pero actualmente estos sistemas pueden alcanzar hasta los 18 Hz [7]. La frecuencia es un factor para considerar, pues a frecuencias de 1 Hz en los *trackers*, se ha encontrado que se tienen pérdidas de información en cuanto a la estimación de distancia, principalmente cuando el jugador corre a velocidades mayores de 16 km/h y una distancia corta, por lo que se ha llegado a la conclusión que, para no tener pérdidas significativas en cuanto a datos relacionados a la velocidad, la frecuencia de muestreo debe de ser de 10 Hz [22].

Un sistema que ejemplifica esto es el Catapult de Player Tek, el cual es un *wearable* que se usa de la misma manera que los chalecos de la Fig. 7, tiene una frecuencia de 10 Hz para el monitoreo de posición e igualmente es capaz de mostrar datos inerciales como velocidad y aceleración con una frecuencia de 100 Hz. Este sistema viene en conjunto con software proporcionado por la misma empresa para el análisis de los datos y es usado por clubes de fútbol como Bayern Múnich, Chelsea, Olympique de Lyon y Real Madrid.[23]

La principal desventaja de los *trackers* GPS es que en ambientes cerrados pierden precisión, por lo que se han adaptado tecnologías GPS y se les ha añadido sistemas de triangulación para mejorar la precisión en ambientes cerrados; el ejemplo principal es el Wimu Pro, un sistema que funciona con señal de geolocalización GNSS y además integra antenas locales sobre el campo de juego para aumentar la precisión usando señales UWB, además de medir variables como la velocidad y distancia también proporciona variables fisiológicas como la frecuencia cardiaca y la saturación de oxígeno [24]. Este sistema actualmente está siendo usado por todos los equipos adheridos a primera división de la liga MX. [21].

2.3. SISTEMAS DE MONITOREO DE ENTRENAMIENTO A TRAVÉS DEL BALÓN

De igual manera que en los dos puntos anteriores se monitoreaba las actividades físicas de un jugador, también se han propuesto alternativas enfocadas en medir exclusivamente las habilidades del jugador a través del balón. Algunos de estos sistemas van muy ligados con el uso de teléfonos inteligentes, pues suelen ser donde el software se corre, ya que está pensado para que pueda ser usado por cualquier persona y no solo por cuerpos técnicos de clubes de fútbol. La utilidad de estos sistemas no se ha validado científicamente y por lo tanto no se tiene una idea clara del beneficio que podría o no traer al jugador.

Un ejemplo de estos es el sistema *Smart soccer ball* de *Dribbleup*, este es un sistema dirigido al público en general, principalmente a jugadores en etapa de formación. El uso de este sistema va ligado directamente con el uso de un teléfono inteligente pues es una aplicación y hace uso de la cámara de este. El funcionamiento es el siguiente: se coloca el teléfono inteligente a una distancia adecuada según el tipo de ejercicio y con el balón *Smart soccer ball* se monitorea que el ejercicio se haga de manera adecuada, internamente la aplicación, hace un monitoreo de la posición del balón y es este el que valida si el ejercicio se realizó de manera adecuada además de que guarda el tiempo en que se realizó la acción. Los ejercicios soportados son bastante limitados, pero aún

así logra medir de manera cuantitativa la habilidad de un jugador en un determinado ejercicio a través del número de repeticiones hechas y el tiempo que toma realizarlas [25]. Como se puede notar, este es un sistema muy parecido a los entrenamientos de posición realizados por cámara, solo que con las tecnologías actuales se puede lograr que cualquier persona pueda hacer uso de estos sistemas.

Otro ejemplo de estos sistemas es el balón *Smart Ball miCoach* de Adidas. Este balón cuenta internamente con sensores que permiten la medición de la velocidad con la que fue disparado, además de contar con una batería internamente para alimentar el sistema y un cargador inductivo para realizar la carga de forma inalámbrica [31]. Para hacer uso de este sistema era necesario el uso de un dispositivo con iOS para poder visualizar los datos [26]. Entre los datos que se despliegan está la velocidad alcanzada, así como la trayectoria tridimensional, a través de una animación, que siguió el balón. El concepto de este balón es bastante interesante pues permite llevar los registros del tiro de cada jugador y poder ver un progreso en su desempeño una actividad que antes solo se llevaba de manera cualitativa.



Fig. 8 Balón *Smart Ball miCoach* de Adidas y cargador inductivo. [26]

2.4. ENTORNOS DE DESARROLLO DE HABILIDADES

Estos sistemas son entornos completamente cerrados donde uno o dos jugadores realizan ejercicios tácticos, por lo general se tiene un círculo central donde se sitúa el jugador y se le lanzan

pases con un lanzador de balones automático, ya que el jugador tiene el balón se le muestra a donde tiene que lanzar el balón a través de una señal visual como luces de colores. Estos sistemas llegaron en esta década con el Footbonaut del Borussia Dortmund y el 360S del Benfica, ambos son clubes que se caracterizan por la venta de jugadores a clubes más grandes por cuantiosas sumas de dinero, por lo que el desarrollo de sus jugadores juega un papel crucial para su modelo de negocios.

El Footbonaut consiste en un entorno con 4 paredes en las cuales hay 72 paneles cuadrados, el jugador se coloca en el centro y puede recibir el balón desde alguna de las 4 paredes a través del lanzador de balones y en seguida se ilumina el panel a donde debe de hacer llegar el balón, como se ve en la Fig. 9, el objetivo de este sistema es ayudar al control de pases y a mejorar las habilidades espaciales del jugador. Es importante mencionar que esta herramienta se desarrolló a finales del 2012 cuando el Borussia Dortmund estaba en su momento de máximo esplendor, bajo la dirección de Jürgen Klopp y tenía jugadores como Robert Lewandowski y Mario Götze [27]. Algunas de las desventajas que tiene es que en la realidad sobre el terreno de juego el jugador está en constante movimiento por diferentes zonas del campo y por otro lado el costo de la construcción del equipo es elevado y lo hace difícil de replicar en otros clubes.

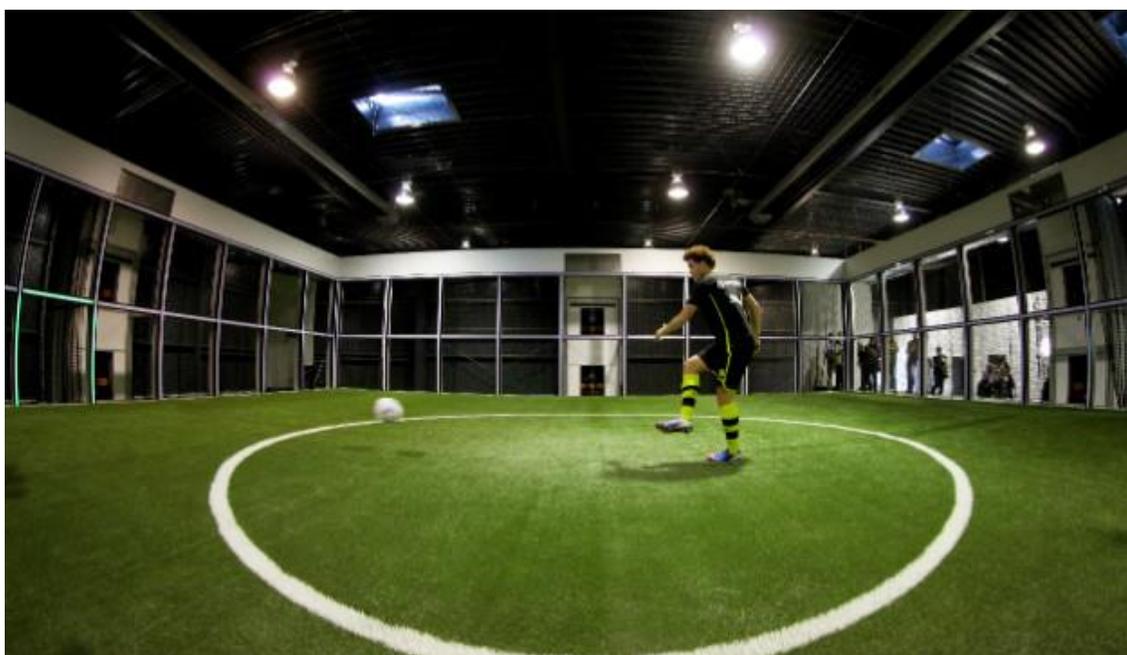


Fig. 9 Footbonaut y modo de operación [28]

De una forma muy parecida funciona el 360S del Benfica Lab, con la diferencia de que las paredes se sustituyen por paneles de luces LED los cuales son capaces de desplegar imágenes o animaciones, brindando la oportunidad de proponer entrenamientos más complejos a diferencia del Footbonaut [29]; por ejemplo, en este se puede proponer que el jugador apunte a un objetivo que esta en movimiento y muestra en tiempo real si el jugador se equivocó o no, también se puede desplegar una imagen de una portería y mostrar al jugador puntos clave para anotar como se ve en la Fig. 10. De igual manera que el Footbonaut la principal desventaja es la complejidad y el coste que supone montar uno de estos equipos, no lo hace accesible a todos los clubes.



Fig. 10 Funcionamiento del 360S de Benfica Lab [30]

3. REQUERIMIENTOS Y ESPECIFICACIONES

3.1. IDENTIFICACIÓN DE NECESIDADES

En el capítulo anterior se mostraron algunas de las principales tecnologías aplicadas al fútbol, su funcionamiento y algunas de las limitaciones que presentan, con esta información se puede hacer un listado de los requerimientos que debe de tener el sistema de este trabajo para cumplir con las necesidades esperadas por los potenciales usuarios finales, en este caso el equipo técnico de clubes de fútbol. A partir de la información recabada en el capítulo dos se identificó el siguiente grupo de necesidades:

- **N1. Despliegue de información en tiempo real:** Para los entrenadores es importante realizar el seguimiento de los jugadores en el momento, pues les permite verificar si el jugador está realizando de manera adecuada el ejercicio y corregirlos en caso de que no.
- **N2. Monitoreo de la distancia recorrida por los jugadores:** Este es uno de los principales usos que se les da a los trackers y permite prevenir lesiones. [14]
- **N3. Capacitación mínima:** Es necesario que el sistema lo pueda utilizar el público en general sin necesidad de una preparación especializada.
- **N4. Montaje rápido y transportable:** De la misma manera, como se vio en el capítulo anterior, sistemas como el footbonaut o el 360S tienen como desventaja que ocupan bastante espacio, además de una instrumentación compleja y no se pueden transportar.
- **N5. Costo accesible:** Por otro lado, un problema que aqueja a los clubes deportivos hacia la tecnificación es el alto coste que tienen estas tecnologías haciéndolo inaccesible para algunos clubes de la liga mexicana.
- **N6. Capacidad de almacenar datos:** Esta es una de las características básicas que hacen la mayoría de los dispositivos de rendimiento físico, pues permite llevar un historial del rendimiento del jugador.
- **N7. Identificación de datos relevantes:** Una de las principales quejas que se tiene sobre las tecnologías aplicadas al fútbol es que estos sistemas arrojan datos que en algunos casos no son relevantes o no aportan información nueva al equipo técnico.
- **N8. Ser lo menos invasivo posible con el jugador:** También es importante que el diseño del sistema le permita al jugador realizar sus movimientos con naturalidad, de manera que el sistema deberá realizar el monitoreo de la distancia recorrida sin afectar la movilidad del jugador.

- **N9. Interfaz amigable:** Muy relacionado con la N3, la interfaz deberá de contener elementos de diseño de las interfaces modernas de aplicaciones móviles como Netflix, Uber, etc. Para que al usuario le resulte muy fácil de usar y con elementos agradables visualmente.
- **N10. Transmisión de datos inalámbrica:** El sistema deberá de transmitir los datos de forma inalámbrica para evitar complicados cableados e interferir con el entorno del jugador y por lo tanto en su comportamiento en los ejercicios.
- **N11. Duración de batería adecuada:** La vida de la batería deberá de ser la adecuada para soportar la duración de las sesiones de entrenamiento, teniendo la autonomía necesaria para soportar al menos una sesión antes de necesitar ser recargada nuevamente.
- **N12. Alcance de transmisión:** El alcance de conexión de los elementos deberá de ser lo suficientemente amplio para poder cubrir un área de trabajo que normalmente ocupa un ejercicio.

Ahora se hará el ordenamiento de las necesidades ponderando la importancia de cada una, de esta forma se podrá dar una idea más clara de cuáles son necesidades esenciales para tratar y cuáles no. El resultado de este análisis se ve en la tabla 1, donde se muestran las necesidades por número y nombre además se ponderan del 1 al 5, donde el 1 significa que es poco relevante y el 5 denota que es una necesidad muy importante.

Tabla 1. Ponderación de la importancia de cada una de las necesidades

Id	Necesidad	Importancia (1-5)
1	Despliegue de la información en tiempo real	5
2	Monitoreo de la distancia recorrida por los jugadores	3
3	Capacitación mínima	3
4	Montaje rápido y transportable	4
5	Costo accesible	1
6	Capacidad de almacenar datos	4

7	Identificación de datos relevantes	5
8	Ser lo menos invasivo para el jugador	5
9	Interfaz amigable	4
10	Transmisión de datos inalámbrica	5
11	Duración de la batería	4
12	Alcance de transmisión	3

A continuación, se sustentará el por qué se le ha dado esa calificación a cada una de las necesidades dadas en la tabla 1:

N1.- Se le ha dado una calificación de 5 debido a la importancia que juega el conocer en tiempo real las acciones del jugador, siendo éste un requerimiento esencial para el entrenador, pues permite corregir las acciones de los jugadores, permite retroalimentar al jugador. Dar retroalimentación es un proceso esencial en la educación, pues permite acortar la diferencia entre el desempeño actual y el deseado [32], dar retroalimentación en tiempo real permite corregir más rápidamente las acciones no deseadas.

N2.- Por otro lado, se le asignó una calificación de 3 a la distancia recorrida por los jugadores, esto es debido a que es un parámetro importante, pero es menos relevante con respecto al objetivo principal del sistema, el cual es mejorar las habilidades de los jugadores en lugar de concentrarse en la medición de la distancia y la precisión de la medición.

N3.- En cuanto a esta necesidad se le asignó una calificación de 3 debido a que el usuario (cuerpo técnico) está acostumbrado a tomar capacitaciones para manejar los sistemas tecnológicos, por lo que, si bien se buscará que sea un sistema intuitivo, el objetivo de este trabajo no es llegar a un nivel óptimo que requiera cero capacitaciones, si no buscar un equilibrio entre automatizar la mayoría de las tareas y conocimiento del usuario para manejar de forma correcta el sistema.

N4.- La calificación de esta necesidad fue de 4, asignada debido al uso del terreno de entrenamiento para varias actividades y a que es necesario entrenar en varios lugares, por lo

que se debe de mover los elementos de entrenamiento y se debe evitar que sean elementos pesados y de grandes dimensiones, prefiriendo en su lugar que sean elementos que sean transportables entre una o dos personas para su montaje.

N5.- Esta necesidad no es relevante para este trabajo, pues el tema principal queda lejos de buscar reducir los costos del producto y el tema comercial, es por esta razón que se le ha dado una ponderación de 1.

N6.- Esta es una necesidad esperada por el usuario, la razón por la cual es el valor de 4, es debido a que el almacenamiento se diseñará pensando en guardar algunos de los datos más importantes como el tiempo promedio, mejor tiempo y algunos parámetros recientes, pero no el historial completo de los movimientos realizados por el jugador.

N7.- Una de las características más solicitadas por los entrenadores es que se filtren datos relevantes para los análisis, por lo tanto, se lleva una calificación de 5 pues es una característica que ayudará a los jugadores a tomar decisiones cruciales respecto a su desempeño.

N8.- Esta necesidad es algo primordial, pues el fin del sistema es permitir el desarrollo de habilidades que se pueden dar en el campo, por lo que el jugador no debe de percibir la presencia de los sensores pues podrían afectar su desempeño, por estos motivos se le asignó una calificación de 5 a esta actividad.

N9.- Una interfaz amigable con el usuario es algo relevante para el proyecto pues se busca como objetivo facilitar el trabajo del entrenador, se le denotó una calificación de 4 pues es relevante que la interfaz presente características de interfaces actuales, como ser responsiva y procurar tiempos de carga rápidos, pero buscando que su diseño sea enfocado más a la funcionalidad que a la estética.

N10. La transmisión inalámbrica debe de ser una prioridad, pues el uso de conexiones alámbricas limitaría la conexión de los dispositivos y también limitaría el entorno, por lo que se le asignó una calificación de 5 para denotar la importancia que tiene lograr una conexión inalámbrica.

N11. A este rubro se le colocó una calificación de 4, pues la duración de la batería es primordial, pero se puede ser flexible; debido principalmente a que los dispositivos sólo se

usan en ciertos intervalos de tiempo en lugar de todo un entrenamiento completo, por lo que puede permitirse que el dispositivo sea igual o ligeramente menor a los dispositivos *tracker* existentes en el mercado.

N12. El alcance de la conexión inalámbrica es un tema para considerar pues puede limitar la cantidad de ejercicios que se puedan hacer, por lo que se buscará que el alcance de la conexión sea una cantidad adecuada, pero debido a que la extensión de área de un ejercicio puede ser tan grande como el entrenador desee, se le colocó una calificación de 4 para buscar cumplir con la distancia que requieren la mayoría de los ejercicios, aunque algunos se queden fuera.

Viendo la tabla 1 podremos darnos cuenta de que algunas de las necesidades se escapan del objetivo principal de este trabajo, el cual es crear un sistema para el mejoramiento de las habilidades deportivas y monitoreo físico de los jugadores, por lo que se hará una tabla 2 de especificaciones donde se filtrarán las necesidades que se abordarán en este trabajo. La tabla dos está ordenada de la siguiente manera, en la primera columna se encuentra el número de la necesidad el cual es el mismo que en la tabla 1, en la segunda columna se encuentra el enunciado de la necesidad y en la tercera columna se encuentra el número que le corresponderá con la nueva numeración de especificación.

Tabla 2. Filtración de las necesidades en especificaciones

No. Necesidad	Necesidad	No. especificación
1	Despliegue de información en tiempo real	1
2	Monitoreo de la distancia recorrida por los jugadores	2
4	Montaje rápido y transportable	3
6	Capacidad de almacenar datos	4
7	Identificación de datos relevantes	5
8	Ser lo menos invasivo posible con el jugador	6

9	Interfaz Amigable	7
10	Transmisión de datos inalámbrica	8
11	Duración de batería	9
12	Alcance de transmisión	10

3.2. ESPECIFICACIONES

Con base en la filtración de las necesidades se procederá a realizar las especificaciones del prototipo, teniendo como especificación un parámetro cuantificable. La tabla 3 se encuentra distribuida de la siguiente manera: en la primera columna están las especificaciones, en la siguiente columna se encuentran las unidades, en la tercera columna se encuentra el valor propuesto, el cual, es el valor que se desea que tenga el prototipo, finalmente en la última columna se indica con qué especificación de la tabla 2 cumple; abreviando la necesidad a la cual se refiere con la letra “e” y el número de especificación, por ejemplo si nos queremos referir a la especificación uno lo pondremos como e1.

Tabla 3. Tabla de especificaciones y restricciones

Especificación	Unidades	Valor propuesto	Cumple con
Frecuencia	Hz	5	e2, e1
Peso de sensores para el campo	kg	7	e6, e3
Ancho para sensores del campo	cm	50	e6, e3
Alto para sensores del campo	cm	20	e6, e3

Largo para sensores del campo	cm	10	e6, e3
Peso de sensores de jugadores	g	90	e3
Ancho para sensores de jugadores	cm	1.2	e3
Alto para sensores de jugadores	cm	6.5	e3
Largo para sensores de jugadores	cm	4.3	e3
Duración de la batería para sensores de jugadores	Horas	4	e9
Capacidad de almacenamiento de la base de datos	MB	9	e4
Alcance del sistema de monitoreo	Metros	50x60	e10

A continuación, se justificará la razón por la cual se le ha dado ese valor propuesto a cada una de las especificaciones.

- Dimensiones, peso y batería para los sensores de jugadores.

Debido a que no hay muchos sistemas se tomará como referencia el Catapult y el Wimu Pro en cuanto a las dimensiones, peso y batería que deberá tener el sensor que llevarán los jugadores,

se tomaron estos dispositivos debido a que son 2 de los principales sistemas usados en el mercado por grandes clubes deportivos como el Barcelona o el Real Madrid. En cuanto al peso el rango de estos oscila entre los 110 g y 70 g por lo que el peso propuesto será el promedio de estas 2 medidas: 90 g; por otro lado, en cuanto a las dimensiones el Wimu Pro tiene 81x45x16 mm [33] y el Catapult tiene 48x40x6 mm [34] por lo que las dimensiones propuestas serán el promedio de ambas siendo finalmente 65x43x11 mm; por último la duración de la batería en ambos sistemas es de aproximadamente entre 5 y 3 horas, por lo que el valor propuesto será de 4 horas.

- Tasa de muestreo del sistema

Mientras que para la frecuencia de muestro se han encontrado estudios donde lo razonable es que esta frecuencia esté entre los 1 y los 10 Hz, en caso de usar la frecuencia de 1 Hz se pierde precisión al momento de estimar los desplazamientos por lo que la frecuencia que se recomienda usar son los 10 Hz para no tener pérdidas de información, aunque con 5 Hz se ha comprobado que es válido tomar el muestreo, ya que se tiene una estimación de la distancia cercana a la real, en especial cuando son recorridos lineales y distancias moderadas [22], es por esta razón y debido al tiempo que le toma hacer las lecturas y transmisión de datos que se eligió la frecuencia de 5 Hz.

- Dimensiones, peso y batería para los sensores en el terreno de juego (sensores de campo)

En cuanto a los sensores de campo es complicado tomar un punto de referencia si los comparamos con el Footonaut o el 360S, pues estos sistemas son salas de entrenamiento fijas; por lo que las medidas serán propuestas pensando que pueda ser transportado fácilmente, así que un punto de partida para la propuesta de medidas será que todo el sistema pueda entrar en la cajuela de un automóvil común, por lo que las medidas del conjunto del sistema no deben de sobrepasar los 0.290 m³ que son las medidas de la cajuela de un automóvil Aveo[35], además pensando tentativamente en que el sistema tendrá 8 sensores y pensando en formas cúbicas, las dimensiones de cada uno deberán de ser aproximadamente 33cm por lado. Mientras que el peso de los sensores de campo cumplirá con la NOM-006 en el apartado 8,5 inciso e, donde se establece que la carga máxima manual para un empleado debe ser 25 kg y para menores de 16 años debe de ser de 7 kg [36], por lo que se tomará este último valor como valor propuesto pues menores de edad podrían transportar el sistema.

- Tamaño de la base de datos

Para este fin se hizo un cálculo aproximado del tamaño que tendría la base de datos, tomando que el tiempo promedio de la duración de la batería será 4 horas, suponiendo que el entrenamiento se hace de forma continua se podría esperar una recolección de muestra de datos cada medio minuto, esperando que aproximadamente se tomen 20 datos por cada muestra, se tendrían 9600 datos al día, si tomamos estos datos como float con tamaño de 8 Bytes[37], el tamaño aproximado de muestra por día sería de 76,800 Bytes, suponiendo que se entrena con este sistema 4 veces a la semana y que la temporada de fútbol mexicano dura 6 meses (27 semanas), el tamaño mínimo de la base de datos debería ser de 8.3 MB.

- Alcance de transmisión.

Para el área de alcance se pensó en el área típica de entrenamientos, las cuales no suelen sobrepasar la mitad de la cancha, pues áreas de trabajo mayores causan mayor desgaste físico en el jugador; aunque pueden usar más espacio si el entrenador lo desea. Así que el área de entrenamiento, para este sistema, se planeará como 50x60 metros que es el equivalente aproximado a media cancha de fútbol, esto permitiría flexibilidad para entrenar varios ejercicios.

4. DISEÑO CONCEPTUAL

4.1. GENERACION DE CONCEPTO

Para iniciar este trabajo se plantea la generación del concepto, donde se buscará dar solución a las necesidades previamente planteadas en el capítulo anterior alineándolo con el objetivo planteado en el capítulo uno:

“Un sistema que ayude a monitorear el desempeño físico y el desarrollo de habilidades de jugadores de fútbol, buscando potenciar los entrenamientos de fútbol para hacerlos más interactivos, se usará una interfaz que permita llevar un control acerca del ejercicio y la forma en la que el jugador lo realiza y se pueda monitorear en tiempo real, además de que la configuración se pueda realizar de una forma sencilla, también se llevará un historial del desempeño de los jugadores pudiendo generar un resumen general del jugador en cada ejercicio”

4.2. DIAGRAMA DE BLOQUES

Como primer paso para el diseño conceptual será necesario bocetar el sistema como si fuera una caja negra para poder identificar las entradas y las salidas que tendrá éste, de esta forma se podrá crear una idea general de los subsistemas a necesitar.



Fig. 11 Diagrama de caja negra del sistema

En la Fig. 11 se puede ver el sistema en diagrama de caja negra, del lado izquierdo se encuentran las entradas que son:

- Energía: que se refiere a la alimentación eléctrica para suministrar energía a los elementos del sistema.

- Ejercicio a entrenar: la cual es una acción que debe seleccionar el usuario y ubicar los dispositivos para poder realizar el entrenamiento.
- Acciones de los jugadores: la cual hace referencia a los movimientos que realizarán los jugadores.

Mientras que del lado derecho se encuentra la salida esperada del sistema, la cual es un despliegue de información, con esto se hace referencia a estadísticas y datos del desempeño del jugador en cada ejercicio, así como indicadores que permitan la comparación entre jugadores.

Con este diagrama de caja negra se puede entonces empezar a bosquejar los subsistemas que tendrá el sistema para poder cumplir con el concepto anteriormente planteado y su interacción, para que a partir de las entradas se logre obtener la salida deseada.

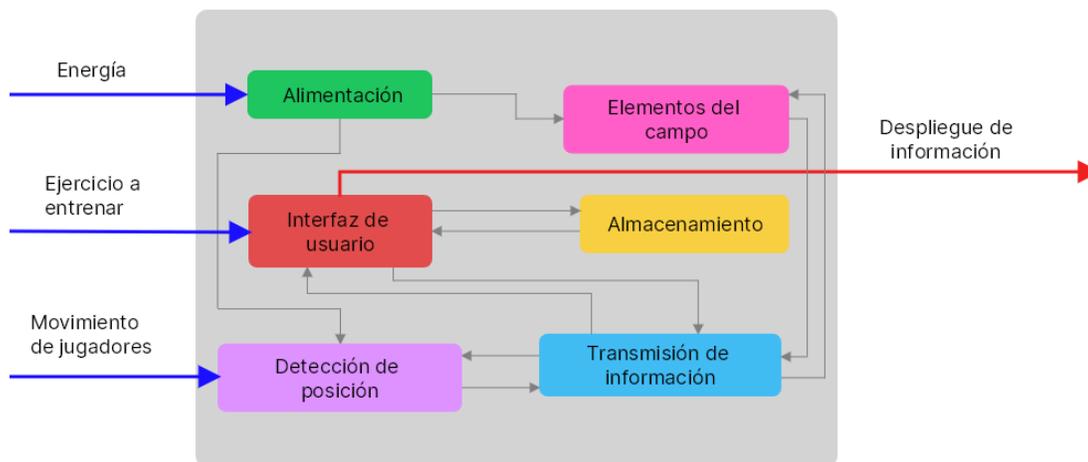


Fig. 12 Diagrama de caja negra con subsistemas

En la Fig. 12 se muestra el diagrama de caja negra, pero identificando los subsistemas que compondrán al sistema, a continuación, se detalla a qué se refiere cada uno de los subsistemas:

- S1 Sistema de detección de posición: Será el encargado de dar a conocer la posición del jugador en el campo de entrenamiento, para poder transmitir esta información a un siguiente subsistema.
- S2 Sistema de interfaz de usuario: es la encargada de procesar la información entrante del sistema de medición del jugador y coordinarlo con los dispositivos del campo para

generar interacción entre el jugador y éstos, también es la encargada de procesar los datos para mostrarlos en gráficas y generar las peticiones para guardarlos en el almacenamiento.

- S3 Almacenamiento: Este sistema es donde se guardarán los datos relevantes del ejercicio realizado por el jugador para su posterior revisión y para generar comparaciones entre jugadores.
- S4 Sistema de alimentación: este sistema es el encargado de proveer los parámetros de energía necesaria y adecuada para los elementos del campo y el sistema de detección de posición.
- S5 Elementos del campo: son dispositivos electrónicos que serán los encargados de tomar las mediciones de las acciones hechas por el jugador, así como mostrar las acciones que debe de realizar el jugador durante el ejercicio.
- S6 Sistema de intercomunicación: Es el encargado de realizar la comunicación entre la interfaz, el sistema de medición de posición y los dispositivos que se encuentren dentro del campo, pues como se puede ver en la Fig. 12, estos sistemas no se conectan de manera directa si no que es necesario un intermediario que gestione esta interacción.

4.3. GENERACIÓN, SELECCIÓN Y EVALUACIÓN DE CONCEPTOS

Con los subsistemas descritos se podrá empezar a analizar y proponer diversas tecnologías que permitan realizar las funcionalidades antes planteadas en la sección 4.2.

4.3.1. SISTEMA DE DETECCIÓN DE POSICIÓN

Como se planteó en los requerimientos, la detección de la posición es una necesidad primordial para el desarrollo de este trabajo, a esto se debe lo indispensable que es saber la posición del jugador para poder conocer qué acciones realizar en los dispositivos del campo dependiendo de la acción que haya tomado el jugador; por lo que se procedió a hacer una investigación para encontrar las posibles tecnologías que se podrían tomar en cuenta para el desarrollo de este trabajo.

Se encontró que la detección de la posición en tiempo real es un problema que se ha abordado en diferentes trabajos [38][39][40] y que se le conoce como RTLS (Real Time Localization System),

del mismo modo se encontró que se usaron principalmente las siguientes tecnologías para poder abordar esta tarea:

- Uso de un GPS y un acelerómetro
- Uso de triangulación por radiofrecuencia
- Uso de visión artificial

A continuación, se explicarán las características de cada uno de los sistemas para poder observar sus pros y contras y al final se decidirá cuál de las tecnologías usar a partir de estos parámetros.

GPS con acelerómetro

El Sistema de Posicionamiento Global (en inglés, Global Positioning System), originalmente Navstar GPS, es un sistema que permite posicionar cualquier objeto (una persona, un vehículo, etc.) sobre la Tierra con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo común son unos pocos metros [41]. El GPS en solitario es una opción de fácil implementación cuando se desea tener una precisión entre 2 a 10 metros en espacios abiertos, adicionalmente en trabajos como el de F. Vicente[40] se agrega una IMU (unidad de medición inercial) para mejorar la precisión del GPS, este procedimiento mejora significativamente la precisión. Por otro lado, experimentalmente se comprobó con una tarjeta de microcontrolador que permite una frecuencia de muestreo de hasta 10 Hz, siendo uno de los requerimientos que se buscan en el diseño. Algunas de sus desventajas son que, a pesar de hacer uso del acelerómetro, puede tener lecturas bastante erráticas en espacios cerrados por lo que es necesaria la implementación de una IMU y un filtro de Kalman discreto para el mejoramiento de las lecturas, lo cual implica generar un modelo del movimiento del jugador para lograr un buen filtrado.

Visión artificial

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real a través de una cámara con el fin de producir información numérica o simbólica para que puedan ser tratados por un equipo de cómputo [42]. Para el posicionamiento y seguimiento se han realizado algoritmos como el uso de caminos más cortos [39] y/o modelos probabilísticos [38], para este caso se instalarían 4 cámaras como en el trabajo de Jerome Berclaz [39] para poder ubicar al jugador pues con una sola cámara es complicado ubicar la profundidad del objeto y sobreposiciones entre ellos. Como ventaja tiene la frecuencia de muestreo y la precisión que ofrece en espacios cerrados.

Sistemas de radiofrecuencia

El monitoreo por radiofrecuencia es llevado a través de la medición de tiempos de retrasos en el envío de datos entre dispositivos o en la medición de la intensidad de señal (RSSI), en ambos casos es necesario al menos otros 3 dispositivos aparte del dispositivo a identificar para poder estimar la posición del objeto, esta es una de las principales formas de detección de posición en lugares cerrados donde el GPS no tiene buenas lecturas. Por ejemplo, en el trabajo de A. Bastida [22], se comprobó la eficacia de usar triangulación de posición por medio de radiofrecuencia UWB (ultra width band) y sobre el GPS, pues éste parecía tener una dificultad en intervalos en que el jugador estaba en una posición estática y de la misma forma se veía ligeramente afectado por la velocidad y el tipo de desplazamiento.

A continuación, se pondrá una tabla donde se ponderan las ventajas y desventajas de cada una de las opciones.

Tabla 4. Comparativa entre las diferentes características de las tecnologías para la detección de posición

Parámetro/Tecnología	GPS + Acelerómetro	Visión artificial	Radiofrecuencia
Frecuencia de muestreo [Hz]	10	20	10
Documentación	Media	Buena	Media
Rango Precisión [m]	0.3-100	1-2	2-10
Corriente máxima estimada	230mA	3000mA	920mA
Equipo de procesamiento	Microcontrolador	Equipo de cómputo especializado	Microcontrolador
Hardware requerido	Módulo GPS+IMU	Computadora, cámaras	Microcontroladores y antenas
Modelado necesario	Sí	No*	No
Posicionamiento	Absoluto	Relativo	Relativo

Como se puede ver en la tabla 4 la opción que parece ser más atractiva es la opción de monitoreo por medio de radiofrecuencias, pues como se puede observar la visión artificial, aunque posee una buena frecuencia de muestreo y un buen rango de precisión; debido al uso de una computadora para procesar la imagen necesita mayor cantidad de hardware complicando la movilidad, y como se explicó en el capítulo tres, se busca que el sistema pueda ser transportable y de rápido montaje. Por otro lado, el GPS y el acelerómetro, aunque poseen características como una baja corriente de uso, requiere modelado en el acelerómetro para obtener una correcta estimación de la posición, así que la opción que se eligió fue hacer la detección de posición mediante radiofrecuencia teniendo tal vez como inconveniente que es un mapeo de tipo relativo por lo tal será necesario una referencia a partir de otros dispositivos, pero que no interfiere con los requerimientos de este trabajo; a continuación se darán algunas de las tecnologías de radiofrecuencia que podrían ser opciones para llevar a cabo este fin.

Bluetooth es una especificación industrial para redes inalámbricas de área personal (WPAN), creado por Bluetooth Special Interest Group, Inc., que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. [43] La ventaja de este protocolo de comunicación es la fácil configuración de los módulos para transmitir información entre dos dispositivos principalmente en configuración maestro-esclavo; aunque hay diferentes clases y versiones de dispositivos, su alcance normalmente es de 10 metros.

Wifi es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con wifi (tales como ordenadores personales, teléfonos, televisores, videoconsolas, reproductores multimedia, etcétera) pueden conectarse entre sí o a Internet a través de un punto de acceso de red inalámbrica.[44]

En la siguiente tabla se expondrán las características técnicas de cada uno para poder observarlas y elegir cual es el más conveniente.

Tabla 5 Comparativa entre las diferentes características de las tecnologías de radiofrecuencia

Característica	Bluetooth	Wifi
Alcance base	10m	45m
Velocidad máxima de transmisión	52 Mbits/s	300Mbit/s
Seguridad	Baja	Alta
Banda de trabajo	2.4Ghz	2.4Ghz/5Ghz
Compatible con Microcontroladores	Sí	Sí
Documentación con microcontroladores	Media	Alta
Creación de red de trabajo	No	Sí

Al final, la elección que se usará para el monitoreo es a través del uso de señales de wifi, pues tiene un rango de alcance mayor y esto es un factor importante, tomando en cuenta un criterio del capítulo anterior donde se estimaba un área de entrenamiento de 50mx60m, la mitad de una cancha de fútbol, pues realmente muy pocos ejercicios de entrenamiento hacen uso completo del campo por lo que al hacer uso de tecnología bluetooth restringiríamos bastante el área de entrenamiento; por otra parte wifi posee una mayor documentación para su implementación en microcontroladores lo cual permitirá hacer el proceso de programación más ligero, pues en caso de surgir dudas en bibliotecas o librerías sólo será cuestión de consultarlas, de igual forma el tener una mayor velocidad en el envío de datos permitiría un monitoreo en tiempo real más fiel que es otra de las características planteadas en el capítulo tres que se buscan cumplir.

4.3.2. INTERFAZ DE USUARIO

La interfaz contempla un software con el que el usuario pueda interactuar con los elementos físicos y no gaste tiempo en realizar configuraciones técnicas y el tiempo en realizar la configuración sea mínimo. Como se planteó en los requerimientos, esta interfaz debe de ser sencilla de usar y

proporcionar información clara al usuario; así que para el desarrollo de la interfaz de usuario que cumpliera con esas características se consideraron las siguientes posibilidades:

- Interfaces para móviles: En esta posibilidad se piensa en el desarrollo sobre el sistema Android, pues este representa un 72.72% [45] del mercado actual de teléfonos celulares. La herramienta de programación que usa es Android Studio, este es el entorno de desarrollo integrado oficial para la plataforma, en 2013 reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.
- Interfaz para computadoras: En esta opción solo se agruparía la posibilidad de usar Visual Studio que permite el desarrollo de programas de software para Windows, se eligió Windows pues ocupa un 73.54% del mercado [46] lo cual permitiría que más personas hicieran uso de la interfaz. Microsoft Visual Studio es un entorno de desarrollo integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Monaco. [47]
- Interfaces web: Con esta posibilidad se referiría a diseñar una interfaz que se pueda desplegar en navegadores tanto de celulares como de computadoras, para este tipo de interfaces se usa HTML para crear el contenido de la página, CSS para dar el estilo y JavaScript como lenguaje de programación.

Tabla 6 Comparativa entre las diferentes características de interfaces graficas.

Característica	Móviles	Computadoras	Web
Documentación	Buena	Buena	Buena
Equipos compatibles	Móviles	Computadoras	Móviles y computadoras
Necesario uso de internet	No	No	Si
Software	Android Studio	Visual Studio	Navegador y editor de texto

Lenguaje de programación	Kotlin/Java	C#	JavaScript
Software de desarrollo libre	Sí	Sí	Sí

Como se muestra en la tabla 6 se puede ver la comparación entre los diferentes tipos de interfaces planteadas, en este caso es complicado decidir por alguna opción en específico, pues con las 3 opciones se pueden crear interfaces interactivas y atractivas visualmente; por lo que para poder cumplir la necesidad de diseño de movilidad se elegirá la opción de interfaz web porque permite el despliegue de información tanto en un móvil como en una computadora, lo cual permitirá al usuario poder gestionar los ejercicios desde su teléfono y posiblemente analizar los resultados desde su computadora, cumpliendo así con la necesidad 4, como desventaja que tiene esta opción es que es necesario el uso de internet o de la puesta en marcha de un servidor en la red local para el despliegue de la interfaz, esto no interfiere con las necesidades planteadas en el capítulo anterior pero es un punto a tener en cuenta para el diseño de detalle.

Debido a que desea que el sistema sea portátil, se decidió alojar en la nube el servidor encargado del despliegue de información, pues de este modo el usuario podría acceder fácilmente al servicio simplemente con tener una conexión a Internet. Actualmente existen principalmente dos alternativas para instalar servidores en la nube, Amazon Web Services y Microsoft Azure, ambos presentan características similares en cuanto a rendimiento y documentación, pero la ventaja de Amazon Web Services es que permite un año de uso de un servidor básico gratis, por lo que se usará esta opción para la parte de prototipado del proyecto.

4.3.3. ELEMENTOS DEL CAMPO

Para generar la interacción entre el jugador y su entorno es necesario que éste tenga microcontroladores o algún elemento que ayude a administrar esta interacción. Por lo que en esta sección se compararán y se elegirá el microcontrolador que irá en estos elementos.

Como se planteó en las necesidades, el sistema debe evitar tener cableado, así que la forma de comunicación será inalámbrica, y tomando en cuenta que la detección de posición será realizada a través de wifi será deseable que el microcontrolador sea compatible o posea un módulo de Wifi integrado; por lo que a partir de esta necesidad se encontraron dos posibles opciones que son populares en las comunidades de *IoT* el ESP32 y el Arduino MKR1010. En la siguiente tabla se observan características relevantes de cada tecnología para este proyecto.

Tabla 7 Comparativa entre las opciones de microcontroladores

Características	ESP32	Arduino MKR1010
Velocidad de reloj	160MHz	48 MHz
Núcleos	2	2
Protocolo Wifi	802.11 b/g/n	802.11 b/g/n
Documentación relacionada al Wifi	Buena	Buena
Consumo de energía	260mA [48]	524mA [49]
Costo	8 dls	32 dls

Como se puede ver en la tabla 7 ambos son compatibles con los protocolos más populares de Wifi, pero en el caso de la batería observamos que el consumo energético del ESP32 es menor por lo cual ayudaría a la necesidad de diseño de ser portátil, pues por ende, la batería necesaria podría ser de menor capacidad y por lo tanto más ligera; por otro lado se observa que la velocidad de reloj del procesador es más rápida en el ESP32 por lo cual se esperará que ayude a la necesidad de despliegue de información en tiempo real lo cual se planteó en el capítulo anterior. Por último, podemos ver que el precio del Arduino MKR 1010 es tres veces mayor al del ESP32, lo cual es un detalle muy importante teniendo en cuenta que al menos para la detección de la posición se necesitarán 4 modelos de microcontrolador, tres referencias y uno que irá puesto sobre el jugador, y el precio entre ambos prototipados se vería seriamente afectado, así que por este motivo y los

expuestos anteriormente se elegirá la opción del ESP32 para ser el microcontrolador que estará en los elementos del campo.

4.3.4. ALIMENTACIÓN

Ya elegido el microcontrolador es necesario pensar en cómo se alimentarán estos elementos, pues siguiendo los requerimientos del proyecto esta alimentación deberá proporcionar suficiente energía eléctrica para poder mantener operando de forma autónoma por al menos 4 horas continuas el equipo. Así que se realizará la operación matemática de la energía necesaria de requerimiento de corriente según la hoja de datos del ESP32 la cual puede llegar a tener consumos máximos de hasta 240mA[48] por lo que se realiza la operación matemática:

$$\text{Corriente eléctrica} * \text{tiempo} = \text{Carga eléctrica}$$

$$240mA * 4 \text{ horas} = 960 mAh$$

Siendo entonces 960mAh la mínima capacidad que debería tener la fuente de alimentación para poder satisfacer la necesidad energética del sistema. Así que entre las posibles opciones que se encontraron de fuentes de alimentación de energía están las baterías: LiPo, NiMH, triple A. En la siguiente tabla comparativa se muestran algunos detalles técnicos de las opciones ya mencionadas.

Tabla 8 Comparativa entre los diferentes tipos de alimentación

Característica	LiPo	NiMH	AAA
Recargables	Sí	Sí	No
Dimensiones [cm]	5.2x3.4x0.5	4.4x4x1	4.4x4x1
Voltaje de salida	3.7V	3.6V	4.5V
Corriente[mA]	1000	600	600
Necesario arreglo de baterías para conseguir la alimentación necesaria	No	Sí	Sí

Inflamables	Sí	No	No
-------------	----	----	----

Cabe mencionar que para el desarrollo de la tabla 8 se pensó en pilas NiMH recargables cuya apariencia y dimensiones es exactamente igual a las pilas AAA convencionales, es por ello que comparten dimensiones en ambos casos. Por el lado de la comparación, observamos que el NiMH y las LiPo son baterías recargables, lo cual es un punto a favor, por otro lado observamos que en la parte de las dimensiones no hay gran diferencia entre estos modelos, aunque para poder obtener esos valores de voltaje en las pilas de NiMH y AAA es necesario generar un arreglo de baterías, cosa que en las baterías de LiPo no es necesario lo cual podría ayudar a cumplir la necesidad de diseño de ser lo menos invasivo con el jugador; es por este motivo y por el hecho que son recargables, que se elegirá esta opción de fuente de alimentación.

4.3.5. TRANSMISIÓN DE INFORMACIÓN

La transmisión de la información desde el microcontrolador hasta la interfaz o base de datos juega un papel crucial, pues como se planteó en las necesidades ésta debe de ser inalámbrica para no interferir con los movimientos de los jugadores y debe de poderse realizar en un rango estimado de 50m x 60m para poder cubrir la mayoría de los ejercicios de entrenamiento en fútbol. Es por este motivo, y ya que la detección se está realizando sobre wifi, que la transmisión de datos deba de ser hecha mediante wifi. Por lo que entre las opciones de protocolos que son compatibles están los UTP/IP y entre este abanico se encontraron dos que son populares: el protocolo HTTP y el protocolo MQTT; a continuación, se hablará un poco más acerca de sus ventajas y sus desventajas.

El Protocolo de transferencia de hipertexto (en inglés, Hypertext Transfer Protocol, abreviado HTTP) es el protocolo de comunicación que permite las transferencias de información a través de archivos (XHTML, HTML) en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet *Engineering Task Force*, colaboración que culminó en 1999 con la publicación de una serie de RFC. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. [50]

MQTT (en inglés Message Queing Telemetry Transport) es un protocolo de mensajería estándar de OASIS para Internet de las cosas (*IoT*). Está diseñado como un transporte de mensajería de publicación/suscripción extremadamente liviano que es ideal para conectar dispositivos remotos con una huella de código pequeña y un ancho de banda de red mínimo. Hoy en día, MQTT se utiliza en una amplia variedad de industrias, como la automotriz, la fabricación, las telecomunicaciones, el petróleo y el gas, etc. Los clientes MQTT son muy pequeños, requieren recursos mínimos, por lo que pueden usarse en microcontroladores pequeños. Los encabezados de los mensajes MQTT son pequeños para optimizar el ancho de banda de la red. [51]

Tabla 9 Comparativa entre los diferentes protocolos de comunicación

Característica	MQTT	HTTP
Conexión	Full-duplex	Half-duplex
Tiempo promedio de mensaje	32 ms	768 ms
Optimizado para microcontroladores	Sí	No
Uso de un servidor	Sí	Sí
Control de llegada del mensaje	Sí	No

Finalmente, para la transmisión de datos se decidió emplear el protocolo de comunicación MQTT, el cual es un protocolo que en los últimos años ha ganado popularidad debido a su implementación en el internet de las cosas. Las ventajas que tiene este protocolo en comparación con HTTP, por ejemplo, es que está pensado para usarse en muchos dispositivos hasta microcontroladores, lo cual hace que se optimice el tamaño de los mensajes y no es necesario tener grandes cabeceras para el envío de mensajes cosa que si pasa en HTTP donde se usan estas cabeceras para tener más control sobre el despliegue y el envío de información pero que para este proyecto no es relevante, con esto se puede hacer uso de anchos de banda ligeros haciendo que la transmisión por protocolo MQTT sea hasta 25 veces más rápida que por HTTP [52] y con esto se podría ayudar a alcanzar un despliegue de datos en tiempo real, además de que con el protocolo MQTT la

interconexión y la transmisión de datos entre dispositivos se logra de una manera muy fácil mientras que con el protocolo HTTP se necesitaría crear un complejo diagrama de conexión para generar la forma en que la interfaz logre comunicarse con los dispositivos y viceversa; esto es debido a que MQTT es full-duplex lo cual hace sencillo esta tarea mientras que HTTP es half-duplex [53] lo cual hace complicado el compartir de información, pues principalmente HTTP no fue planeado como un protocolo para que múltiples dispositivos compartieran entre sí información, sino más bien como un protocolo para poder desplegar información en la web.

4.3.6. SISTEMA DE ALMACENAMIENTO

Como se desarrolló en las necesidades, es necesario que el sistema tenga una base de datos para almacenar la información relevante y para poder ser revisada posteriormente para su análisis. Justamente para este fin existen programas informáticos denominados DBMS (Data Base Management System) que se encargan de llevar un control de los datos mediante comandos para la creación, eliminación, búsqueda y reescritura de estos. Entre las opciones que se encontraron que podrían ayudar con esta tarea fue: MariaDB, Microsoft Access y el uso de un bloc de notas, este último no es un DBMS como tal, pero cuando se tiene un volumen bajo de datos y se requiere una implementación rápida puede ser una buena alternativa para cumplir el mismo fin de un DBMS. A continuación, se dará una breve descripción de cada uno.

- Microsoft Access: es un gestor de datos que utiliza los conceptos de bases de datos relacionales y pueden manejarse por medio de consultas e informes. Está adaptado para recopilar datos de otras utilidades como Excel, SharePoint, etc [54]
- MariaDB: Es una de las bases de datos relacionales de código abierto más populares. Está hecho por los desarrolladores originales de MySQL y se garantiza que seguirá siendo de código abierto. Es parte de la mayoría de las ofertas en la nube y es el predeterminado en la mayoría de las distribuciones de Linux.[55]
- Bloc de notas: Como tal no es un gestor de base de datos por sí solo, se tiene que hacer uso de algún lenguaje de programación para alcanzar este fin, a pesar de esto es funcional para algunos proyectos pues es de rápido prototipado y tiene como ventaja que sus

archivos pueden ser guardados en extensiones .txt o .csv, lo cual permite interacciones con múltiples paquetes de software y múltiples sistemas operativos.

Tabla 10 Comparativa entre las posibles opciones de bases de datos

Característica	MariaDB	Bloc de notas	Microsoft Access
Gestor de datos	Sí	No	Sí
Creación, Eliminación y Actualización de datos	Sí	Sí	Sí
Software Libre	Sí	Sí	No
Seguridad	Buena	Mala	Buena
Integración con Desarrollo Web	Alta	Regular	Buena*

Como se ve en la tabla 10 las opciones más viables parecen ser Microsoft Access y MariaDB pues ambos poseen una buena compatibilidad web para el tratamiento de datos y cosa para la cual el bloc de notas no está optimizado. Por otro lado, podemos ver que MariaDB es un programa de software libre, mientras Microsoft Access no, lo cual es un punto relevante pues por lo general cuando un software es libre se suele tener más documentación o encontrar más tutoriales de cómo realizar determinadas acciones; esto hace que al final se decida usar MariaDB como el gestor de base de datos.

5. DISEÑO DE DETALLE

5.1. DETECCIÓN DE POSICIÓN

Para detectar la posición, el concepto que se plantea usar es la triangulación por potencia de señal recibida, como se puede observar en la Fig. 13, se planea usar al menos tres emisores de señal wifi y un cuarto dispositivo, de este modo se podría hacer una lectura de potencias y a partir de esto lograr una estimación de la posición actual del jugador.



Fig. 13 Modelo de triangulación

Es por ello por lo que se validará qué tan efectivo sería el uso de este modelo, tomando en cuenta la precisión de la posición, así como el tiempo de muestreo empleando el ESP32. Para lograr que el ESP32 sirva como un emisor de señal wifi se necesitará programarlo en configuración de *access point*, siendo los *access point* dispositivos que son configurados en redes de tipo inalámbricas que son intermediarios entre una computadora y una red (Internet o local) [56], en este caso también servirá para la transmisión de datos; mientras que para las lecturas se necesitará programar el ESP32 para estar constantemente monitoreando la potencia de las señales de los *access point*.

5.1.1. CREACIÓN DE ACCESS POINT Y TOMA DE LECTURAS USANDO EL ESP32

Para lograr que un dispositivo ESP32 logre emitir señal wifi es necesario programarlo como un *access point*, para ello se revisó la documentación del ESP32 donde se encontró la biblioteca

Wifi.h y un ejemplo para poder lograr que este emita señal. Por otro lado, del mismo modo, se buscó un ejemplo de cómo escanear constantemente las señales wifi de los alrededores, de igual manera se encontró dentro de la biblioteca Wifi.h, donde se divide en 2 tipos, el escaneo de forma asíncrona y el escaneo de forma síncrona, siendo la diferencia que en el de forma asíncrona permite al ESP32 realizar otras acciones durante el escaneo mientras que de forma síncrona el ESP32 realiza el muestreo exclusivamente y no permite que ESP32 realice otras funciones sino hasta terminado el escaneo de redes, es por ello que se eligió la configuración asíncrona para el muestreo.

El siguiente paso fue cargar el código a dos de los ESP32 para comprobar el funcionamiento de los programas, encontrándose que el funcionamiento era correcto pues se muestreaban las intensidades de los dispositivos emisores de señal de los alrededores, pero el tiempo de muestreo oscilaba los 1.4 segundos, lo cual no cumplía con los requerimientos planteados en el capítulo 3. Se procedió a investigar si era posible bajar este tiempo de muestreo, se encontró que este efecto era causado debido a que la señal de wifi de 2.4Ghz se encuentra distribuida a través de 13 canales lo cual hace que el ESP32 tome muestras de cada canal, por lo que la solución que se encontró fue cambiar los parámetros pasados en la función scan para que solo escaneara en un canal y configurando la función SoftAP para hacer que el *access point* trabaje en el canal sobre el cual se está escaneando, con estos pasos se redujo el tiempo de muestreo hasta 70 ms (14.2 Hz) siendo el tiempo promedio 87 ms (11.49 Hz) los cuales ya cumplen con los 5 Hz de los requerimientos establecidos en el trabajo. Como una consecuencia de haber realizado esta configuración, las señales de wifi pueden sobreponerse y causar interferencia al estar sobre el mismo canal.

5.1.2. VALIDACIÓN DE WIFI PARA DETECCIÓN DE POSICIÓN

Antes de continuar cabe aclarar que se decidió cambiar del modelo ESP32 a un ESP8266 para la toma de lecturas debido a que este microcontrolador existe en versiones más compactas lo cual permitiría una mayor movilidad al momento de montarlo sobre un jugador, recordando que en los requerimientos del capítulo 3 se planteó que esta característica es primordial, además de que son dispositivos hermanos por lo que muchas de las bibliotecas del ESP32 tienen su equivalente en el ESP8266. En los *access point* se decidió seguir usando los ESP32 debido a sus características y a que en estos casos la diferencia de dimensiones no causa problemas.

Como primera prueba para revisar la viabilidad del uso del ESP32 se diseñó la siguiente prueba: se colocó un ESP32 en configuración de *access point* para emitir señal wifi y a una distancia de 1 a 8 metros se colocaron ESP8266 para tomar lecturas de la señal durante un minuto, después se graficó el promedio de cada uno de los puntos. En la gráfica de la Fig. 14, se pueden apreciar los datos crudos de una de estas muestras, se observa a simple vista como a mayor distancia existe una menor potencia de la señal recibida.

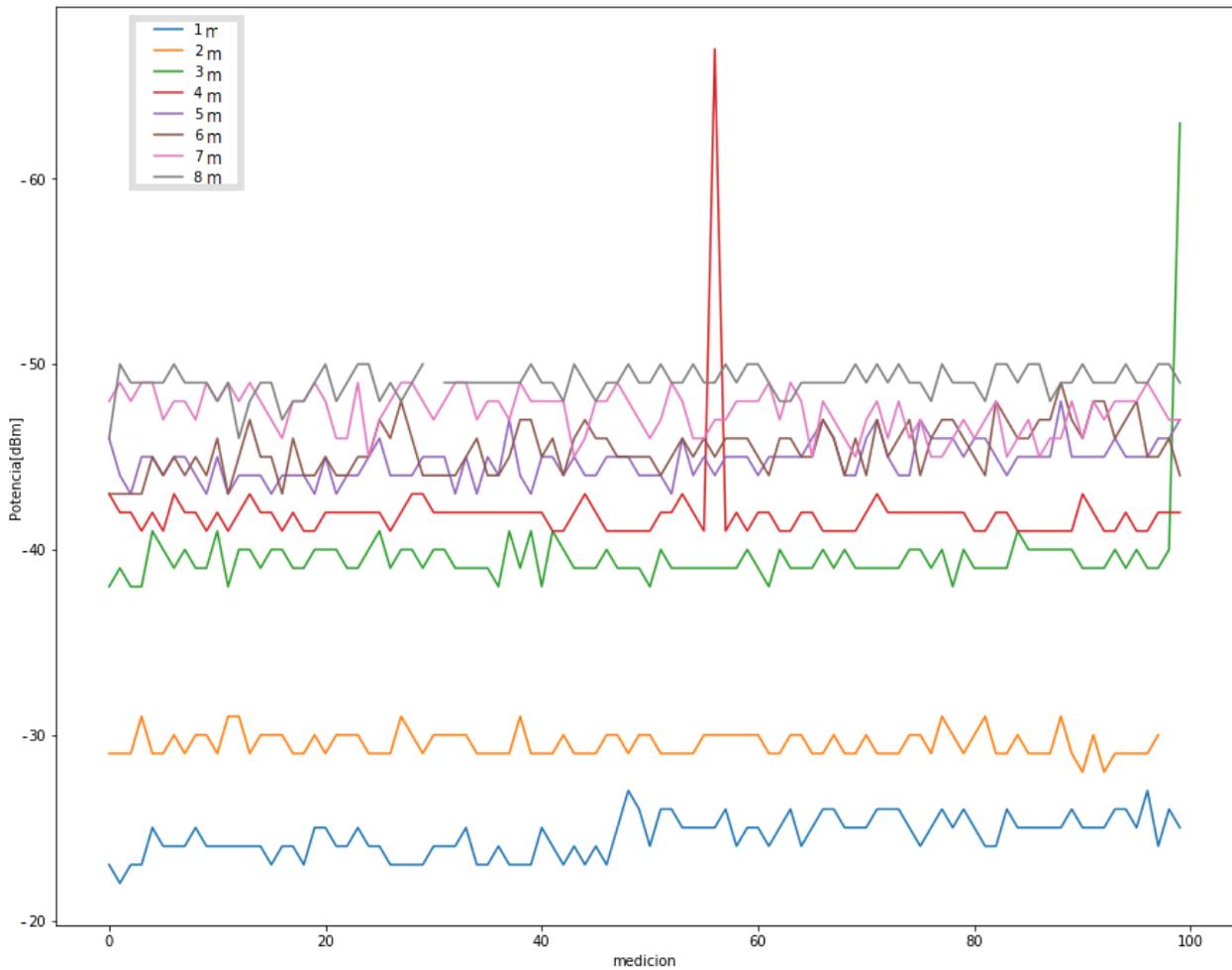


Fig. 14 Muestras recolectadas a distancias de 1 a 8 metros

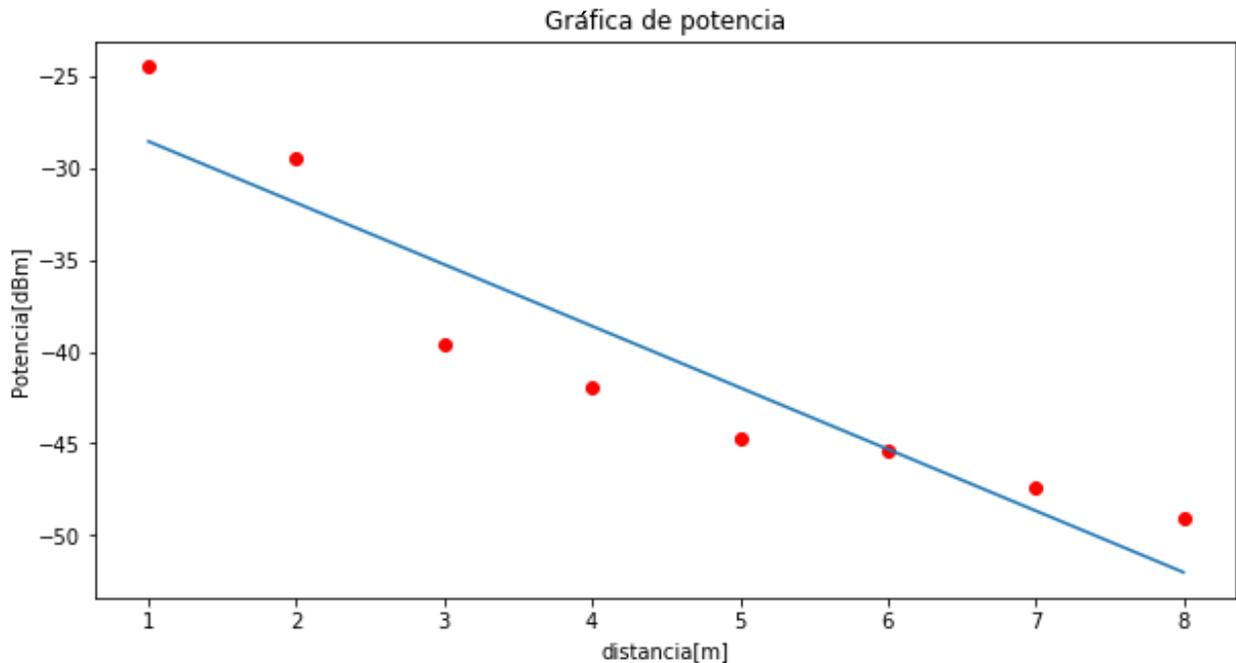


Fig. 15 Graficación de los promedios de las muestras y ajuste a una línea recta

En la gráfica de la Fig. 15, se ajustaron los puntos de los promedios a una línea recta y se obtuvo un coeficiente de determinación (r^2) de 0.86 lo cual indica una correlación fuerte entre ambas variables, de esta manera parece que el sistema puede alcanzar a distinguir entre las distancias correctamente aunque como se puede ver las variaciones después de los 3 metros son casi mínimas, lo cual cobra mucho sentido pues la propagación de ondas suele obedecer una relación inversamente proporcional de tipo exponencial ($\frac{1}{distancia^n}$) con lo cual la potencia de la señal baja drásticamente y a largas distancias se estanca en un valor.

Por otro lado, otra prueba que se hizo fue evaluar que la señal fuera simétrica, esto quiere decir que la señal se esparciera uniformemente en todas las direcciones, para ello se diseñó la siguiente prueba como se ve en la Fig.16, se colocó un ESP32 como *access point* en el centro y luego se colocaron dos ESP8266 a una distancia de 2 y 4 metros y otros dos ESP8266 a una distancia de -2 y -4 metros, de este modo se comprobaría si la señal se propagaba de la misma manera en ambos sentidos. Los resultados se pueden ver en la gráfica de la Fig.17 en este caso se puede observar que no tienen congruencia los datos, pues por ejemplo el comportamiento de la distancia de 2 metros y -2 metros es completamente diferente.



Fig. 16 Figura ilustrativa de cómo se colocaron los receptores de señal y el *Access point* para esta prueba

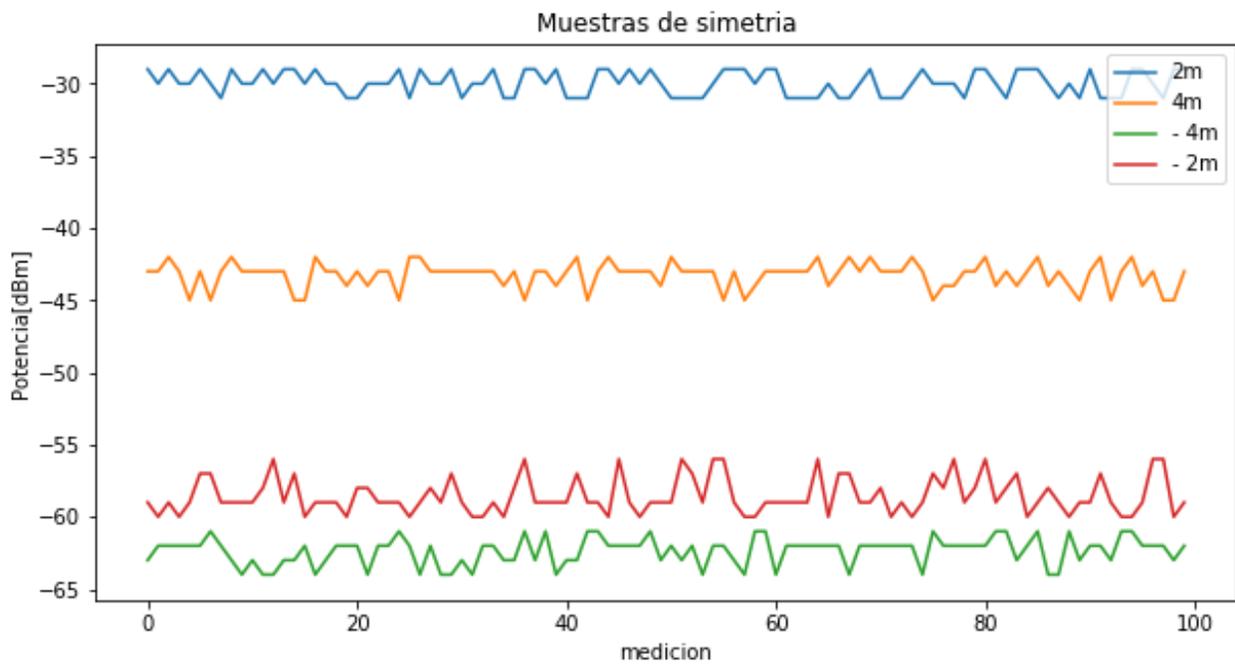


Fig. 17 Comparación de muestras a 2,4,-2 y -4 metros del access point

Por lo que en este caso se pensó que una de las posibles causas de este comportamiento podría ser la antena del ESP32, pues como se pudo ver en la gráfica, el comportamiento cambia drásticamente en la distancia de 2 y 4 metros teniendo en estos puntos lecturas de mayor valor,

por lo cual se decidió cambiar de modelo de ESP32, pues se estaba usando un modelo ESP32U y se decidió usar un modelo ESP32D, que permite añadir una antena de modem y esta permite emitir la señal a mayor potencia, aproximadamente a 3dBm más, y de la misma forma tiene un campo en la transmisión de señal más uniforme, lo cual fue comprobado, pues se repitió la prueba pasada con los mismos parámetros y los resultados se muestran en la gráfica de la Fig.18.

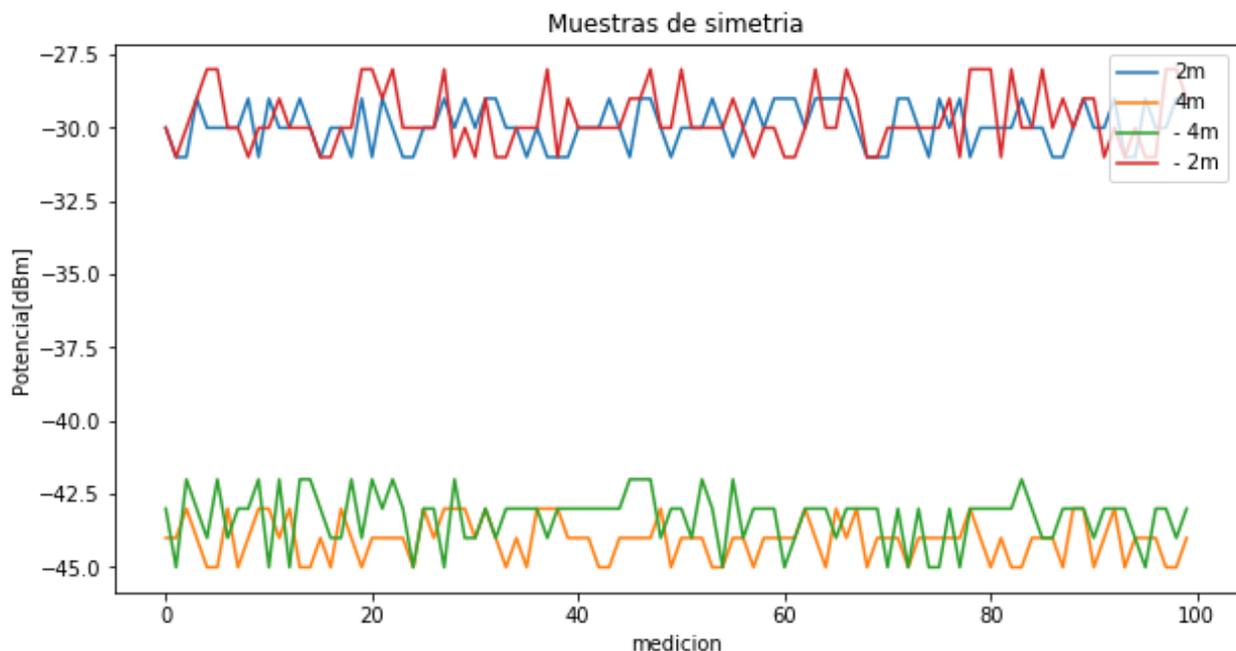


Fig. 18 Comparación de muestras a 2,4,-2 y -4 metros del access point usando una antena simétrica

Como se ve en la Fig.18, se puede apreciar que en esta ocasión los resultados a -2 y 2 metros fueron muy parecidos variando por 0.9dBm en promedio, mientras que a distancias de 4 a -4 metros se repite el mismo comportamiento variando solo por 1.14dBm en promedio. Con este ajuste se espera que se pueda realizar un mapeo consistente en todas las direcciones y por ende realizar una triangulación para conocer la posición del jugador.

Con las pruebas realizadas en los pasos anteriores todo parece apuntar que este sistema planeado de trilateración puede ser funcional, por lo que fue necesario comprobarlo en una prueba que simule su uso en un entrenamiento donde un jugador corre, para ello se ideó la siguiente prueba: se colocó un *access point* para emitir señal, luego se puso un ESP8266 en la mano de un sujeto de prueba, el cual se encontraba a 7 metros del *access point*, para que corriera a toda velocidad

hacia este y la señal iba a ser graficada, de este modo se iba a observar su comportamiento y caracterizarla. Los resultados se pueden ver en la gráfica de la

Fig. 19, donde se puede observar el comportamiento promedio de 80 pruebas, se ve que existe una clara tendencia de subida de intensidad de señal debido al acercamiento del sensor al *access point*.

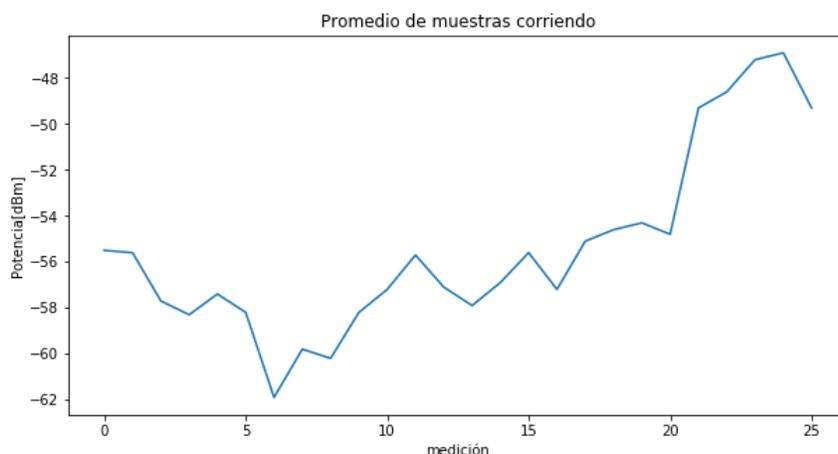


Fig. 19 Promedio de las muestras de señal corriendo

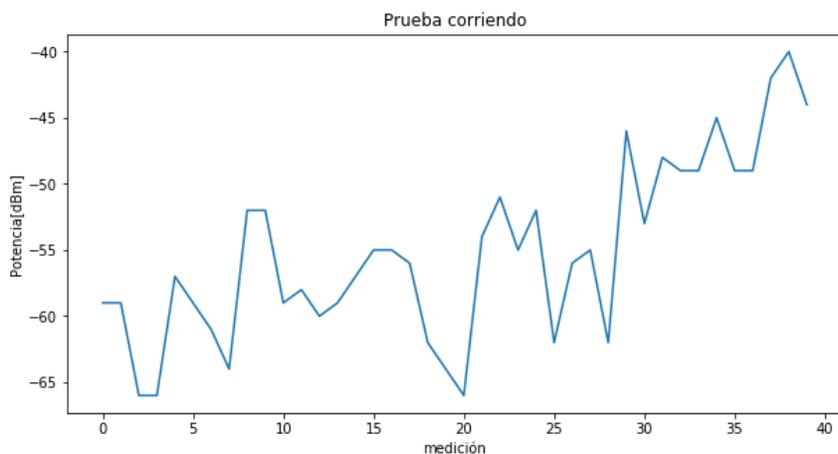


Fig. 20 Muestra de una prueba corriendo

Pero al observar una de estas pruebas por separado, se puede ver que existe ruido (Fig.20), mostrando drásticas subidas y bajadas al inicio del ejercicio, haciendo que por los picos no se pueda asegurar en qué punto se encuentra en el rango de los 4 a 7 metros lo cual hace pensar que la triangulación no será posible. Para buscar una de las posibles razones de la aparición de estos picos, se realizó la misma prueba, pero en esta ocasión con el sujeto de prueba caminando

para observar el comportamiento de la señal, pues se tuvo la sospecha de que una de las posibles causas por la cual fue tan ruidosa la señal es debido a que se encontraba el sujeto de prueba corriendo, lo cual causaba que la antena cambie de posición drásticamente y genere mediciones tan dispares. Una de estas muestras se encuentra en la Fig.21 donde se puede ver que se generan picos, pero son menos pronunciados y se puede observar tendencia a la subida más clara que en la prueba corriendo, lo cual parece confirmar esta idea.

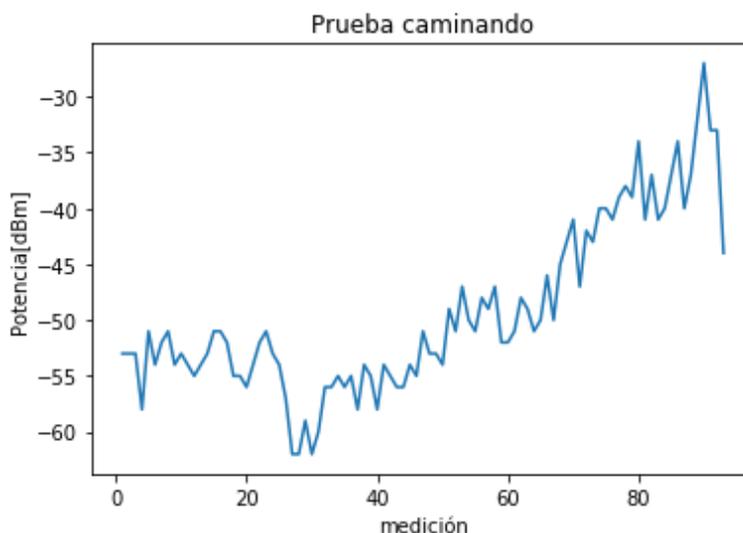


Fig. 21 Muestra de una prueba caminando

Con estos análisis se puede decir que el sistema no se puede usar como sistema de triangulación o no mientras se está corriendo, pues cuando se está caminando parece reducirse el ruido, pero si se puede usar el sistema como un sensor de proximidad, pues como se vio cuando el sujeto de prueba se encuentra cerca del *access point* la señal incrementa seriamente y es notorio un cambio, de este modo se optará por enfocarse en ejercicios donde no se necesite tomar una lectura minuciosa de la posición del jugador y se encamine con el objetivo de este trabajo de mejorar las habilidades técnicas del mismo. De esta forma se colocarán varios *access point* en puntos clave con los que el sistema sabrá que el jugador se encuentra en una determinada posición en lugar de poder mapear todo el espacio de trabajo.

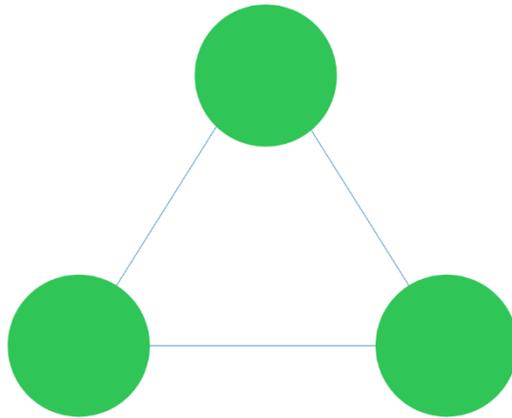


Fig. 22 Ejemplificación de la forma de trabajo del sistema de posición, en puntos clave se colocará un *access point* para poder establecer la posición del jugador, los círculos en verde representan el rango en que los *access point* lograrán detectar al jugador

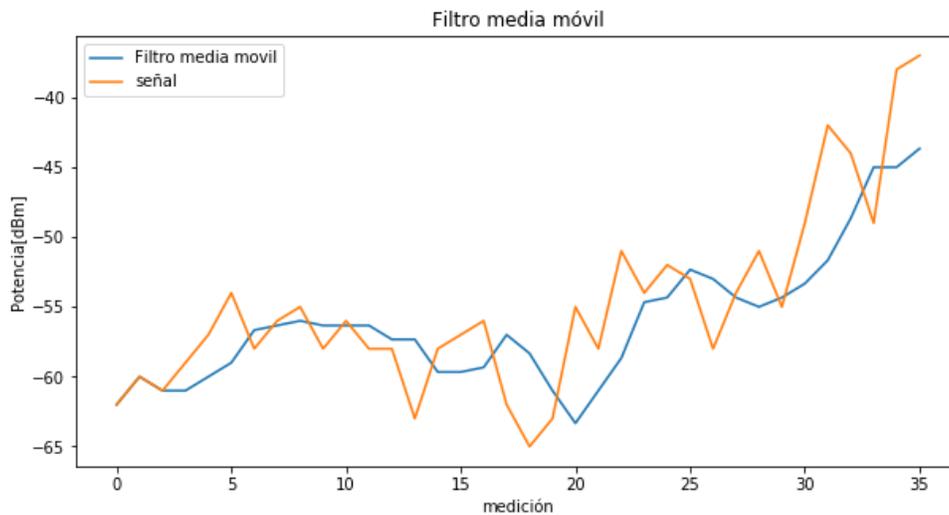


Fig. 23 Comparación entre la señal filtrada con media móvil y la original

Así que el siguiente paso fue filtrar la señal para volver más estable la detección de jugadores, por lo que se pensó implementar el filtro media móvil, el cual es un filtro que consiste en ir promediando los últimos valores y a partir de eso obtener el valor de la señal filtrada [57], para este caso se fueron promediando los últimos tres valores. Los resultados se pueden ver en la Fig. 23, donde se puede apreciar que el filtro logra atenuar de buena forma la señal, como desventaja de este filtro es que a pesar que logra atenuar las variaciones lo hace con un retraso, lo cual se puede ver en

el valor cercano a los -65dBm de la medición entre la 15 y 20 de la señal, pues se presenta un valle pronunciado, pero que en la señal filtrada se muestra hasta en la medida número 20, lo cual representa un retraso de 270 ms, este tiempo interfiere seriamente con el comportamiento esperado, pues se desea que el sistema permita la interacción con el jugador en tiempo real o lo más cercano a este, además de que el retraso total del sistema será aún mayor, debido a que la comunicación inalámbrica también sufrirá de retrasos , por lo que se buscó otra técnica de filtrado.

Se consideró la utilización del filtro de Kalman, pues tiene características deseables como la reducción del ruido cuando es causado por ruido blanco y la estimación del valor actual usando el dato anterior y el dato entrante[58], por lo que se procedió a su implementación y para ello fue necesario obtener la varianza de las señales, la cual fue tomada de las pruebas de las gráficas de la Fig.14 ,y se propuso manualmente la varianza asociada al proceso de tal forma que permitieran filtrar la señal, pero sin generar alisado, como condición inicial se estableció -57dBm, pues a pesar de que el *access point* se encuentre a 10 metros o más no supera los -75dBm, así que -57dBm es un valor inicial que ayuda a converger rápidamente la señal si se está lejos del *access point*.

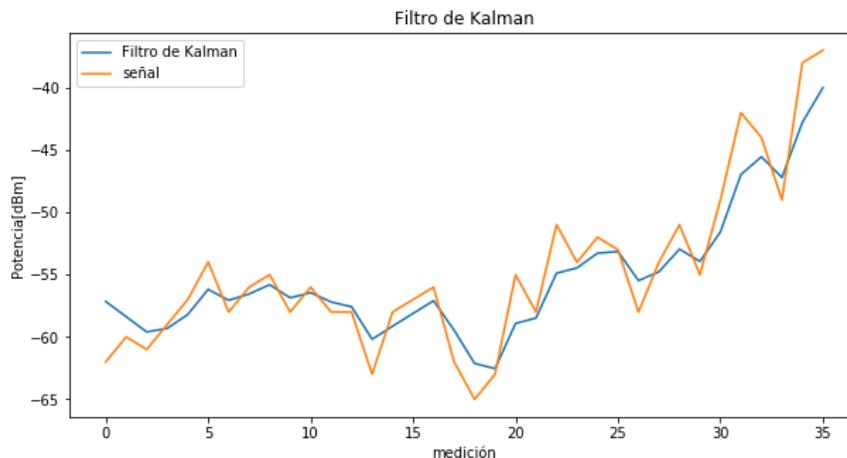


Fig. 24 Comparación entre la señal con filtro de Kalman y la original

Los resultados se observan en la Fig.24 donde se puede apreciar una diferencia con la gráfica de la Fig.23, pues usando un filtro de Kalman se nota que las variaciones leves son atenuadas (mediciones de la cinco a la 10) además podemos apreciar que se filtra en tiempo real, el mismo valle entre la medición 15 a la 20 se ve reflejado automáticamente lo cual se espera que ayude a cumplir los objetivos de este trabajo.

Finalmente para lograr el objetivo de detección en los *access point*, se decidió establecer un umbral sobre la señal filtrada para determinar que el jugador se encontraba a una determinada distancia, se realizaron cincuenta mediciones para determinar el valor de la potencia de señal cuando el jugador se encontraba a un metro de distancia mientras corría, de este modo se encontró que este valor oscilaba entre los -45 y -48 dBm, con lo cual se estableció la señal de umbral en los -47dBm; con esta medición se esperará identificar que el jugador esté cerca del *access point*. En las gráficas de la Fig.25, se puede apreciar como el valor usado como umbral (línea verde) separa la señal de forma apropiada pues se suele rebasar este valor solo cuando el jugador se encuentra a una distancia de un metro o menos del *access point*.

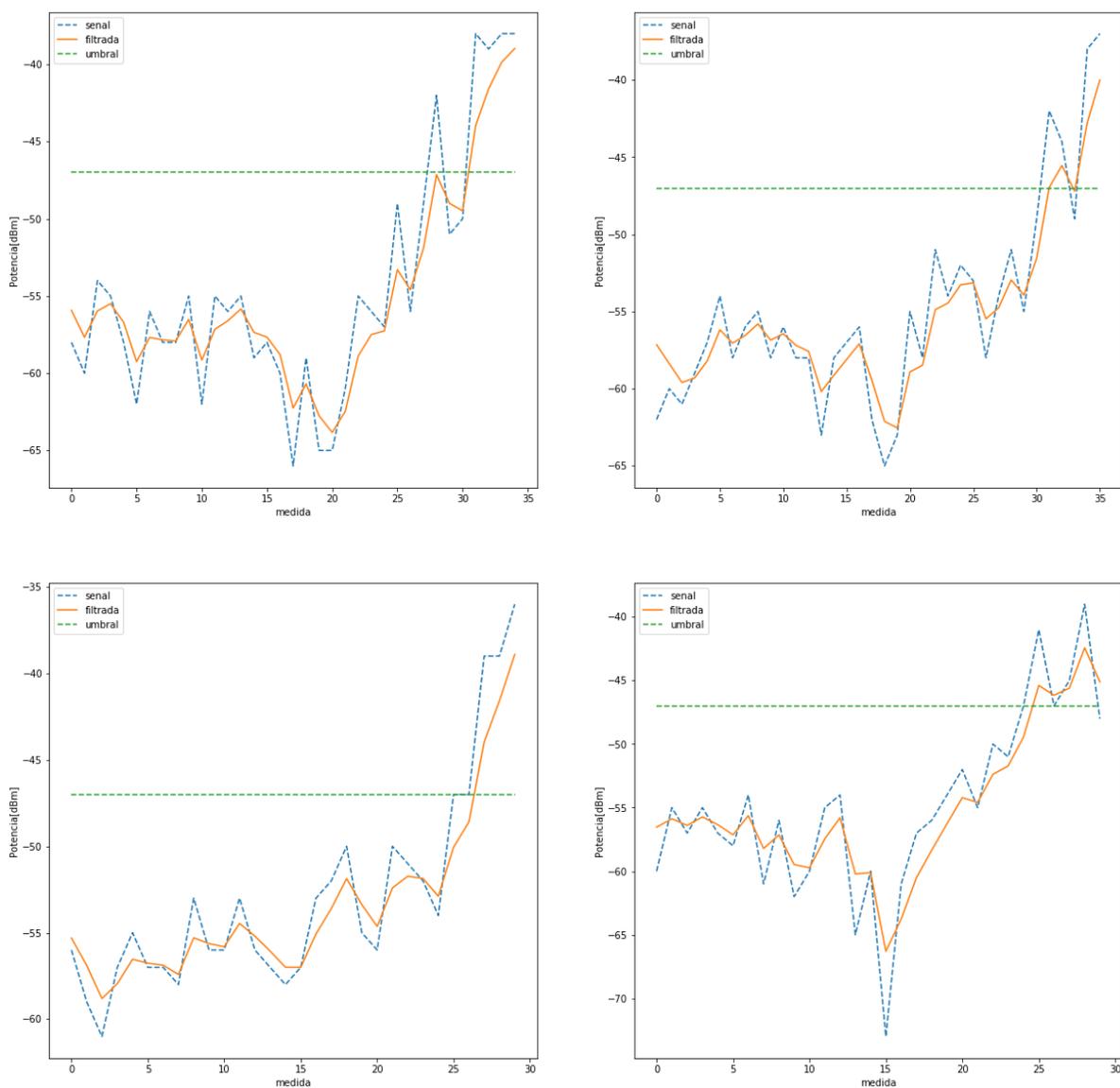


Fig. 25 Muestra de cuatro ejemplos de señales filtradas y con el umbral propuesto

5.2. INSTALACIÓN DEL SERVIDOR

Como se mencionó en el capítulo anterior la instalación del servidor se haría sobre la capa de servicios de Amazon Web Services, específicamente su servicio *Amazon Elastic Compute Cloud* (EC2) el cual es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores. [59]

Este servicio permite crear una instancia en la nube sobre la cual instalar un sistema operativo y de esta forma correr un servidor web. Se eligió instalar sobre esta instancia la versión de Ubuntu 18.04, pues los sistemas Linux son los predilectos para el montaje de servidores y tienen la característica de ser software libre; se eligió Ubuntu pues es una de las distribuciones más populares y que recibe actualizaciones constantemente, se le dio una memoria ROM de 30 GB y una memoria de 1GB de RAM. Amazon también tiene el servicio para obtener direcciones IP fijas llamado *elastic IP* de este modo se ligó una IP fija a la instancia para que se pueda acceder a través de esta con la misma dirección sin el problema que la instancia haya cambiado de dirección.

Ya con la instancia instalada fue necesario configurar la distribución de Ubuntu, se configuraron los puertos del firewall de Amazon para permitir usar el puerto 22 que es el encargado del protocolo SSH (Secure Shell), y para acceder a la instancia de forma remota en Windows se tuvieron que usar los programas putty y Git bash, desde ahí se realizó la configuración de la instalación de los programas. Para la instalación del programa que permite al servidor lograr la interacción y desplegar páginas web se usó Vesta Control Panel, el cual es un programa que instala sobre la instancia herramientas como Apache, MariaDB y PHP [60].

Después del paso anterior, para poder empezar a trabajar los archivos que se necesitan desplegar en el sitio web es necesario vincular el servidor con editores de texto, como Visual Code o Atom, de forma remota; para esto se utilizó la extensión “remote-ftp” de Atom que se comunica a través del protocolo FTP con el servidor, de esta forma se pueden editar los archivos desde una computadora remota sin necesidad de acceder por terminal a esta. Para lo anterior de forma muy parecida a la conexión SSH, se necesitó habilitar el puerto 21, que es el encargado de realizar la conexión FTP, y los puertos 12000 al 12100 para el FTP pasivo desde el firewall de Amazon.

Finalmente se consiguió un dominio www.entfut.ga y se ligó éste para que redireccionara a la instancia a través de la dirección IP, ésta era la razón principal por la cual se debía de tener una IP fija, de este modo al ingresar con la dirección IP de la instancia o con el nombre del dominio

Vesta Panel redirige a la página principal del sitio web. Como paso final para terminar la instalación se solicitó un certificado web SSL para evitar posibles alertas de precaución por parte de los navegadores al considerar el sitio web como inseguro, como ventaja de Vesta es que se puede solicitar un certificado gratuito desde su panel de control a *Let's encrypt*, de tal forma que cuando una computadora se conecte al servidor pueda conectarse por medio de protocolo https y ser considerada segura.

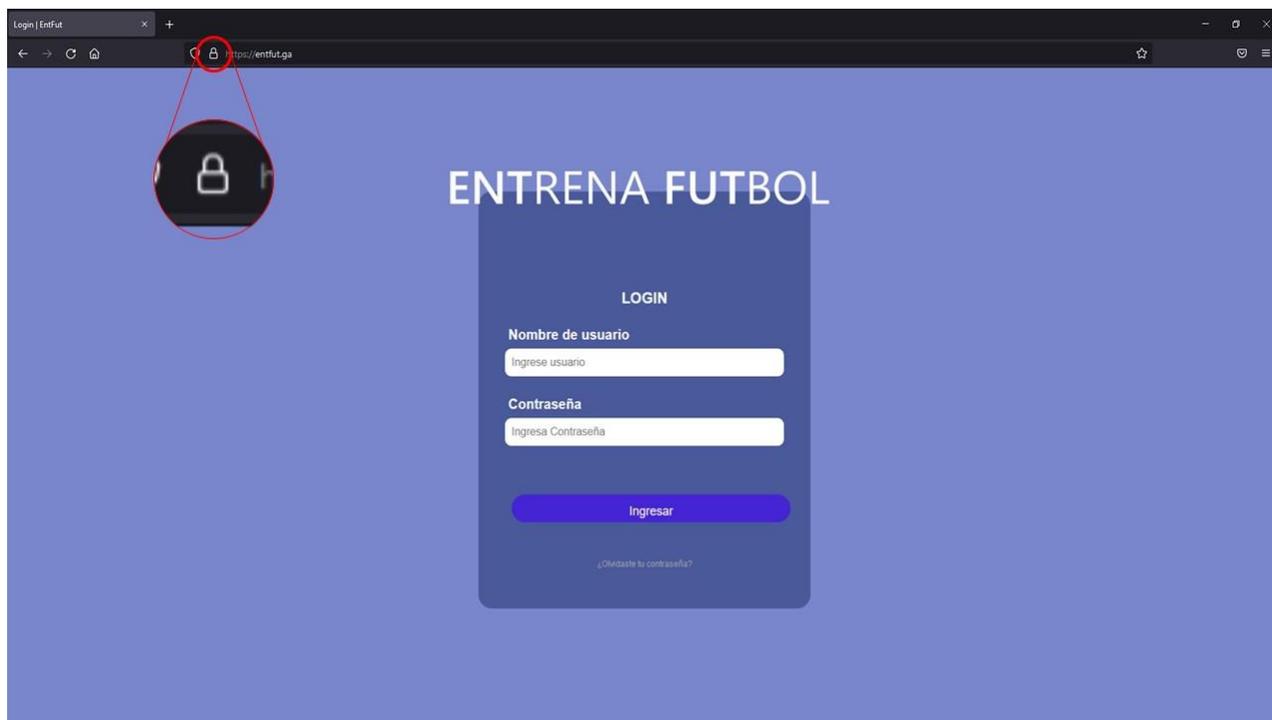


Fig. 26 Página principal del sitio web, se alcanza a apreciar el candado del certificado de seguridad

5.3. RED

5.3.1. CONEXIÓN WIFI

Para poder transmitir información es necesario primero poder conectar el ESP32/ESP8266 a una red wifi, para lo cual se tiene que usar la biblioteca Wifi.h para el ESP32 y WifiESP8266.h para el ESP8266, a pesar de esto las funciones y el procedimiento para conectarse son los mismos en ambos modelos. Para comprobar que se estuviera conectado a la red wifi se usó el comando PING desde una terminal de comandos en Windows de una computadora conectada a la misma red wifi, un ping comprueba el estado de la conexión con uno o varios equipos remotos por medio de los

paquetes de solicitud de eco y de respuesta de eco para determinar si un sistema IP específico es accesible en una red. [61]

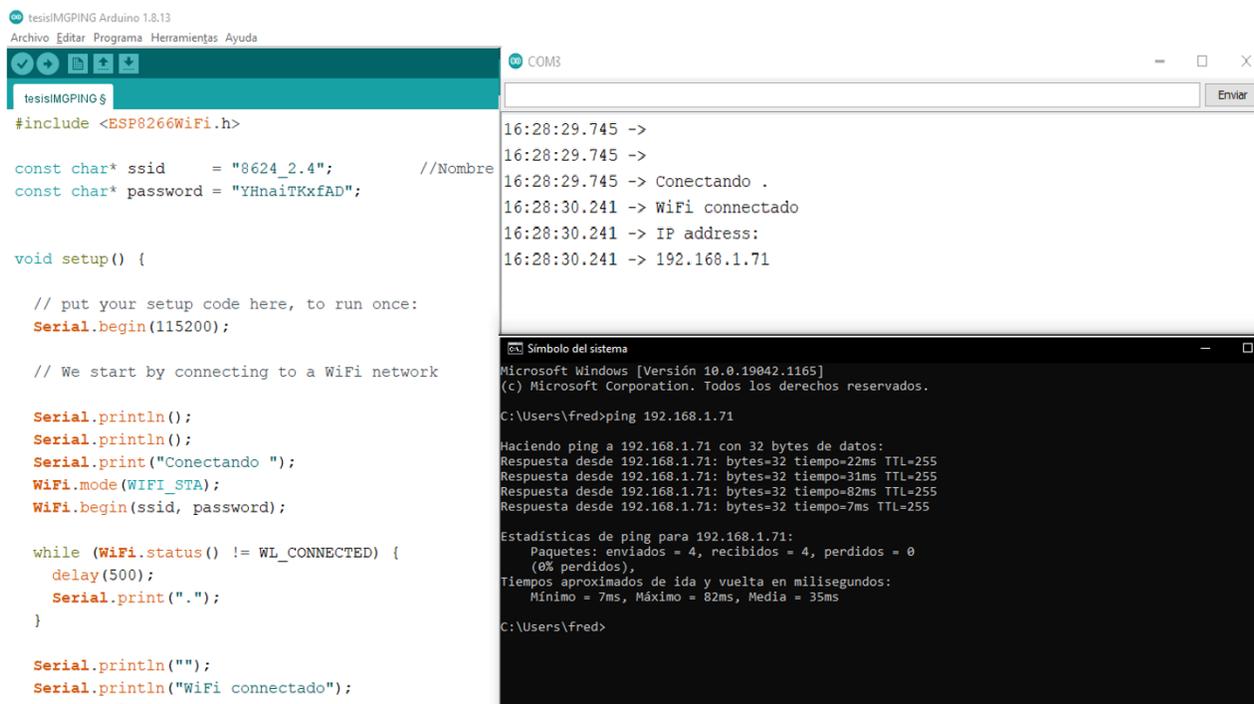


Fig. 27 Captura de pantalla donde se muestra el código usado, la salida del monitor serial con la dirección IP y la terminal de comandos

En la Fig. 27 se puede apreciar parte del código utilizado, así como la salida del puerto serial, como se ve en el monitor serial de Arduino aparece la validación de que se conectó a la red wifi correctamente además de que entrega la dirección IP asignada al ESP866, esta misma dirección IP sirve para ingresarla con el comando ping, obteniendo una respuesta satisfactoria como se ve en la terminal de comandos, de esta forma se validó que el ESP8266 se conectó correctamente a wifi.

Por otro lado para validar si el ESP8266 y el ESP32 se pudieran conectar correctamente a internet se realizó una prueba para que intentara conectarse al servidor web, para esto se utilizó la biblioteca ESP8266HTTP.h la cual permite realizar peticiones POST y GET por medio del protocolo HTTP y de este modo cargar la página web, se subió un código de ejemplo al ESP8266 para realizar una petición GET al servidor y cargar la página, mientras que por parte del servidor se programó en PHP una página web para responder un <h1>Bienvenido + \$user</h1> a peticiones GET. En la Fig. 28 se observa el aspecto de la página cuando se carga en un navegador web,

mientras que en el monitor serial se puede apreciar que la página web respondió con el código 200 lo cual indica que la conexión fue exitosa, e inmediatamente después se puede ver la respuesta del servidor la cual es igual a la que carga el navegador web, de esta forma se comprobó la conexión del ESP8266 a internet.

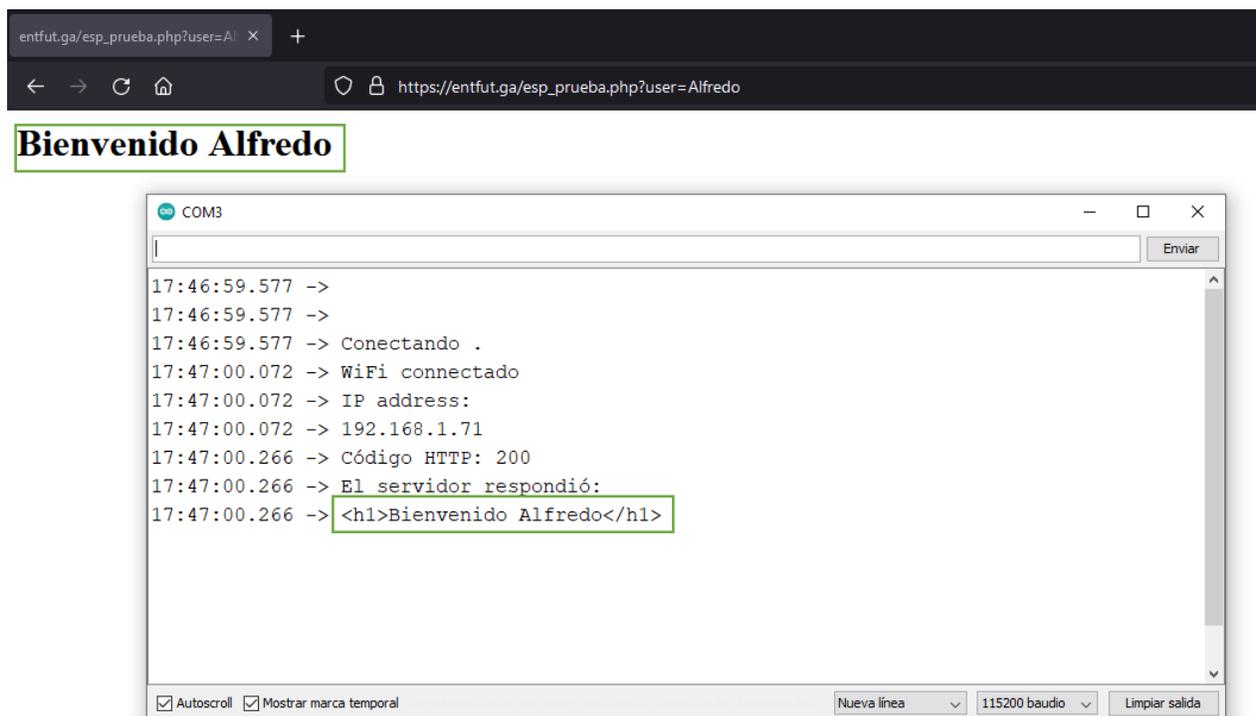


Fig. 28 Captura de pantalla donde se ve el puerto serial y el navegador web al cargar la página web.

5.3.2. TRANSMISIÓN DE DATOS

Para la transmisión de datos como se planteó en el capítulo anterior, se utilizó el *broker* EMQX [87] este es un programa de código abierto con amplia documentación que permite conectar dispositivos por el protocolo MQTT. Para entender el funcionamiento de una red MQTT será necesario hablar de sus dos principales componentes, el *broker* y los clientes; al igual será necesario hablar de las dos principales acciones: la publicación y la suscripción. A la acción de enviar un mensaje se le llama publicación, mientras que a la acción de estar esperando mensajes que se publiquen se le llama suscripción. El *broker* es un sistema encargado de llevar un control de las suscripciones y las publicaciones de los mensajes a los dispositivos, mientras que los clientes son todos aquellos dispositivos que están suscritos y/o publican mensajes.

Ya con todos estos conceptos se explicará un ejemplo de cómo funciona una red MQTT, la Fig. 29 ayudará a visualizarlo mejor, en este ejemplo se tiene un tema llamado *temperature* a los cuales está suscrito tanto el dispositivo móvil como un sistema de *backend*, así que ambos estarán esperando información que sea publicada en ese tema, del lado derecho se tiene un sensor de temperatura encargado de publicar la medición que haya tomado, por lo que cuando éste publique su medición en el tópico *temperature*, ésta llegará al *broker* el cual se encargará directamente de enviar esta medición tanto al dispositivo móvil como al *backend*. Esta topología de red tipo estrella hace sencilla la comunicación entre dispositivos, aligerando la carga de éstos, pues el *broker* es el encargado de distribuir e interconectar los mensajes y no los dispositivos. Es importante resaltar que a pesar de que en este ejemplo el sensor de temperatura solo está publicando también puede recibir información, solo bastará con suscribirlo a algún tema.

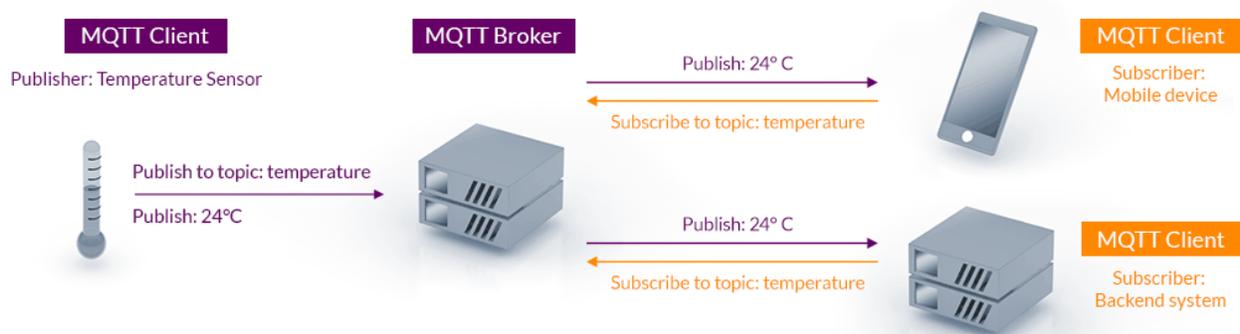


Fig. 29 Funcionamiento de una red MQTT [62]

Se procedió a la instalación del *broker*, para esto solo fue necesario descargar el archivo desde la terminal de comandos de la instancia y descomprimirlo en el servidor, solo que se realizó un cambio en EMQX, pues en la configuración del servidor de Vesta hace uso por default del puerto 8083 y el puerto MQTT de EMQX también, por lo que fue necesario configurarlo cambiando el puerto de comunicación MQTT al puerto 8093 para evitar conflictos.

Ya con la instalación del *broker* lo siguiente que se realizó fue hacer pruebas para validar la conexión entre una interfaz que corra en el lado del cliente y la conexión con un ESP32; así que se ideó una prueba de una aplicación tipo chat, de tal forma que se pudiera enviar mensajes entre la interfaz web y el ESP32, para ello fue necesario crear una pequeña interfaz con código HTML y Javascript; con el fin de lograr esto, se usó la biblioteca MQTT.js que es la principal biblioteca que se usa para la comunicación MQTT en Javascript, se añadió un *input* de texto para escribir, un botón para enviar y un lugar para ver los mensajes que llegaban y cuales habían sido enviados. Por otro lado, se programó el ESP32 y se utilizó la biblioteca PubSubClient.h, para esto fue necesario configurar a qué puerto conectarse, así como crear un *callback* el cual es el encargado de realizar la función de procesamiento de mensajes, del mismo modo se colocó en el código la instrucción de imprimir los mensajes enviados y los mensajes recibidos. Para lograr la interacción entre ambos sistemas se suscribió la interfaz web al ESP32 y el ESP32 a la interfaz web.

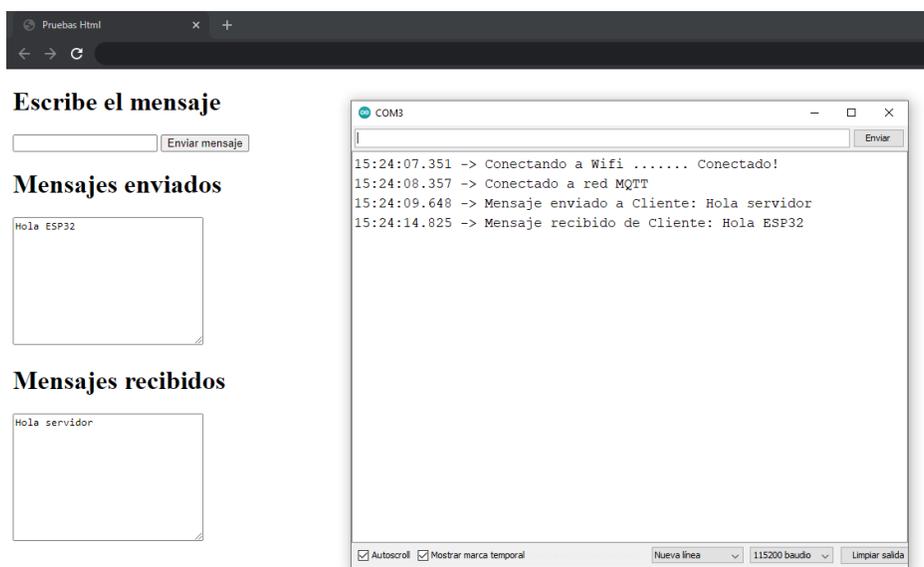


Fig. 30 Captura de pantalla del chat entre la página web y el ESP32

Como se ve en la Fig.30 esta prueba obtuvo los resultados esperados, pues la interfaz web logró comunicarse con el ESP32 y viceversa, ya que en ambos casos se recibió y se envió el mensaje de manera exitosa, por lo que se comprobó el correcto funcionamiento de la red MQTT en el envío de mensajes.

Luego de la prueba anterior, el siguiente paso fue comprobar si la bidireccionalidad de los datos sucedía en tiempo real o causaba algún retraso el estar transmitiendo un mensaje en ambas vías.

Con este motivo se creó una interfaz donde se grafican los datos de intensidad de señal y se crearon dos botones para poder prender y apagar un LED en el ESP32. Mientras que por el lado del ESP32 se programó para que enviara los datos de lectura de intensidad de señal a la mayor frecuencia posible, estar al pendiente en caso de prender o apagar el LED y también imprimir los mensajes enviados y recibidos vía monitor serial. De esta forma, se podría observar al prender y apagar el led y el ESP32 enviando datos, si esta bidireccionalidad causa algún *delay* en tiempo o si se ve reflejado automáticamente.

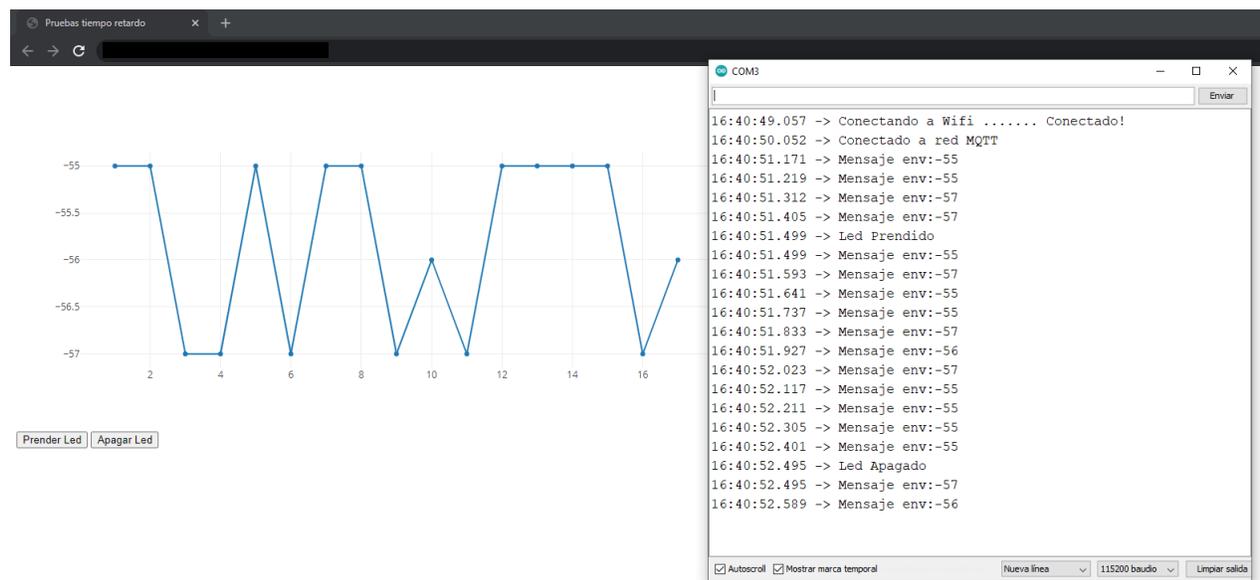


Fig. 31 Captura de pantalla de la comunicación en ambas vías entre el ESP32 y la interfaz web

Finalmente, al momento de hacer las pruebas se observó la respuesta bidireccional y se obtuvo que la interacción era casi en tiempo real, debido a la naturaleza del fenómeno que sucedía en instantes de segundo se decidió grabar con una cámara a 30fps para determinar cuánto era el tiempo de estos *delays*, se hicieron 30 pruebas y se determinó que el tiempo promedio de *delay* fue de 285ms, en ocasiones hubo algunos retrasos, pero estos son atribuidos principalmente a la conexión de internet, pues durante estas pruebas se usó una conexión de internet de 7Mbps y pings de hasta 500ms, lo cual pudo haber provocado estos retrasos. Por lo anterior se determinó que la interacción sucedía casi en tiempo real y este retraso de 285ms es un retraso casi imperceptible para el usuario.

5.4. BASE DE DATOS

En la parte de la base de datos se usó el gestor MariaDB, el cual fue instalado al momento de instalar Vesta Control Panel, para poder crear una base de datos solo fue necesario ingresar al panel de control de Vesta y generar una nueva base de datos ingresando nombre de usuario y contraseña a tener.

En la gestión de la base de datos de forma remota se usó HeidiSQL, el cual es un software gratuito. "Heidi" permite ver, editar y generar estructuras, pertenece a las herramientas más populares para MariaDB y MySQL en todo el mundo [63]. Con el fin de lograr el control remoto de la base de datos se hizo un procedimiento similar a la configuración FTP y SSH, se tuvo que habilitar el puerto 3306 en el firewall de Amazon. Para el acceso remoto se ingresa desde la interfaz de HeidiSQL, ingresando con el nombre de usuario y contraseña creados anteriormente.

El siguiente paso por dar es crear tablas para guardar datos, las tablas son los elementos más importantes dentro de una base de datos ya que se utilizan para almacenarlos en una aplicación. Una tabla está compuesta principalmente por filas y columnas [64]. En el desarrollo de la base de datos se planeó la creación de 3 tipos de tablas: una dedicada a la información de jugadores, otra dedicada a guardar los dispositivos y por último un tipo dedicado a guardar los datos de cada ejercicio. A continuación, se dará una breve explicación de lo que contiene cada una de las tablas, pero primero se explicará un concepto clave en las tablas: la llave primaria y la llave foránea; una *llave primaria* es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla, mientras que una *llave foránea* es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la llave primaria de otra tabla [65]. De esta forma con una llave primaria y una llave foránea se pueden relacionar dos tablas y observar cómo sus datos se interconectan.

En la tabla de JUGADORES se almacenó por fila la información de los jugadores y sus columnas fueron las siguientes:

- `id_jug`: Se usó un id(identificador) para relacionar de forma única los datos de un jugador, por lo que se eligió esta columna como llave primaria.
- `nombre_jug`: almacenará la cadena de texto del nombre del jugador.

- *tr_id_jug* asociado: de esta forma se asociará a un jugador con un dispositivo *tracker* en específico, se usará el id del *tracker* de la tabla de dispositivos, por lo que esta columna servirá de llave foránea.
- *tr_nom_jug*: contiene la cadena de texto del nombre del *tracker* asociado.
- *v_jug*: esta columna guardará el valor de la velocidad promedio del jugador
- *suma*: guarda la suma para calcular el promedio en próximas iteraciones.
- *i*: contiene el valor de cuantos ejercicios ha hecho el jugador, de este modo se divide entre la suma para poder obtener el valor de la velocidad promedio *v_jug*.

 id_jug	nombre_jug	v_jug	tr_id_jug	tr_nom_jug	suma	i
1	pedro perez	2.800	30	Tr20p	14	5
2	Juan Torres	3.200	24	Tr50p	0	0
3	Alejandro Juarez	8.400	7	Tr34	0	0
4	Pedro Flores	9.302	19	Tr230	0	0
5	Hugo Sanchez	22.500	21	Tr70	0	0
7	Pepe Juarez	2.104	40	Tracker	115.701	55
8	prueba	0.000	43	tr3450	0	0

Fig. 32 Representación gráfica de la tabla JUGADORES en el programa Heidi

Mientras que en la tabla de DISPOSITIVOS se almacenó por fila la información de los dispositivos y sus columnas fueron las siguientes:

- *id_con*: Se usó un id (identificador) para relacionar de forma única los dispositivos, se eligió esta columna como llave primaria.
- *dispositivo*: contiene la cadena de texto correspondiente al nombre del dispositivo.
- *tipo_con*: corresponde un número entero del 1 al 3 para indicar qué clase de dispositivo es, el uno indica que es un *tracker*, el dos indica una *portería* y el tres un *access point*.
- *Hora*: servirá para guardar la fecha en que se conectó por última vez al sistema el dispositivo, esta columna servirá para saber si el dispositivo se encuentra listo para su uso o aún no.
- *Entrada*: Es el tema con el cual el dispositivo esperará las instrucciones a recibir.

- Salida: Es el tema en el cual el dispositivo publique sus lecturas o interacciones dependiendo el tipo de dispositivo.

id_con	dispositivo	entrada	salida	hora	tipo_con
2	AP40	p1	p2	2021-02-10 00:22:17	3
5	P20	p2	p4	2021-02-10 00:24:06	2
7	TR1	p4	p7	2021-02-10 00:31:11	1
11	PP	p11	p14	2021-02-10 00:42:47	9
19	YAIR	H1	H2	2021-03-01 19:55:37	1
21	TR300	X1	X2	2021-04-05 16:12:04	1
22	AP300	F1	F2	2021-04-05 16:13:01	3
23	P300	G1	G2	2021-04-05 16:13:34	2

Fig. 33 Representación gráfica de la tabla DISPOSITIVOS en el programa Heidi

Finalmente, las tablas de EJERCICIOS se encargarán de guardar los datos característicos de cada ejercicio, de este modo cada una de las columnas contendrá información relevante como: el tiempo en completar alguna de las fases de interés del ejercicio, si se cumplió el objetivo de la prueba, el jugador que realizó el ejercicio y otros aspectos relacionados al ejercicio. Debido a que a cada ejercicio se le diseñará su propia tabla no se ahondará en este momento en las columnas de éstas.

5.5. DISEÑO DE INTERFAZ WEB

5.5.1. BOCETOS

Antes de empezar a programar la interfaz, el primer paso fue hacer un diseño inicial de ésta; para lo cual se utilizó la herramienta MarvelApp [66] la cual es una aplicación web que permite hacer bocetos de aplicaciones a través de sus pantallas y permite al diseñador crear ligas a otras pantallas al hacer clic sobre elementos, de este modo da la sensación de estar corriendo una aplicación funcional realmente.

En estos primeros acercamientos se diseñó de momento solo la interfaz móvil, para lo cual se comenzó experimentando con crear pantallas con la información y algunos colores de forma básica. Luego se agregaron algunos elementos que hacían parecer a la pantalla más agradable visualmente como agregar íconos y efectos como degradados o gradientes en las tarjetas de opciones. Finalmente, se decidió realizar un diseño más parecido a las aplicaciones actuales como Uber o Spotify donde las opciones son presentadas como tarjetas que abarcan casi todo el ancho con una imagen de fondo, pero con un ligero fondo oscuro para ensombrecer la imagen y causar un contraste con las letras blancas para facilitar la lectura.

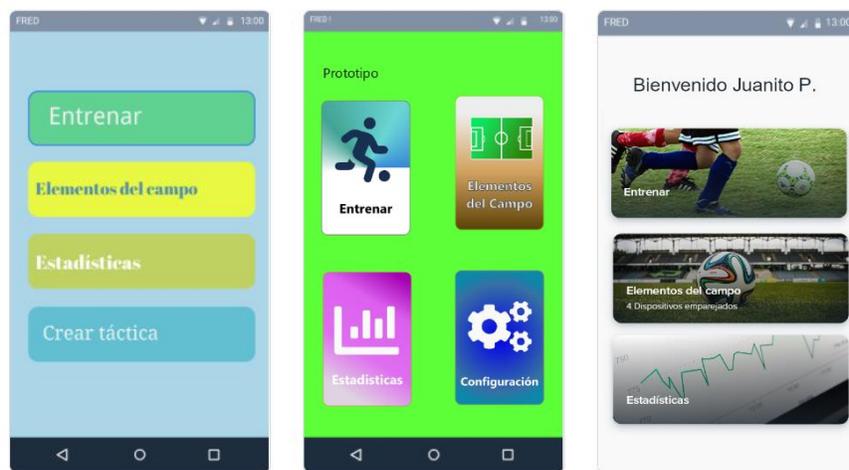


Fig. 34 Iteraciones de la interfaz, de izquierda a derecha, primera versión a más reciente

Los siguientes pasos fueron realizar el diseño de las otras pantallas con algunas funciones básicas planeadas como: revisar las estadísticas de algún jugador, revisar los dispositivos conectados, agregar un nuevo dispositivo. Todas éstas fueron hechas con el fin de revisar como se organizarían los elementos, qué orden deberían llevar y qué información poner para tener una buena experiencia de usuario y a partir de esto comenzar el diseño real de la aplicación en el servidor web.

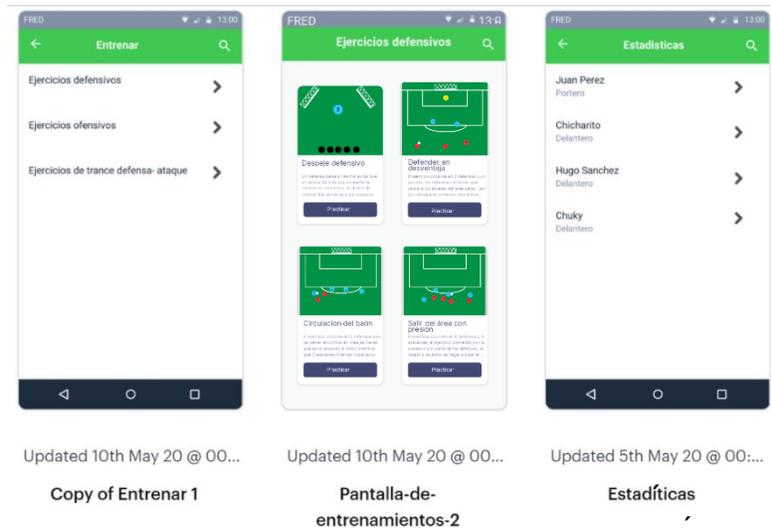


Fig. 35 Bocetos de otras pantallas diseñadas para la interfaz

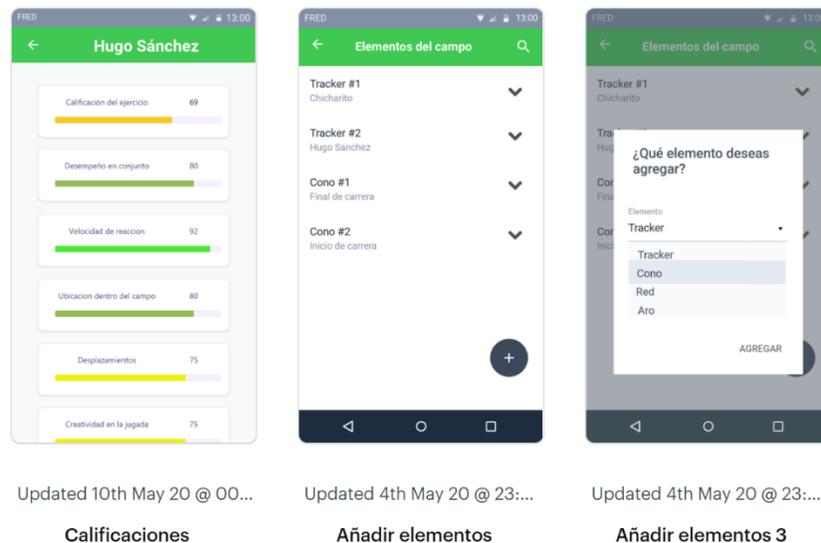


Fig. 36 Más bocetos de pantallas.

5.5.2. DISEÑO DE BACKEND

El desarrollo de interfaces web está constituido por dos componentes: *el backend* y *el frontend*. El *backend* se podría definir como todo aquello que carga de la base de datos sin que el usuario este plenamente consciente de ello, mientras que el *frontend* se refiere a todo aquello con lo que el usuario puede interactuar [67]. En este subcapítulo se muestra el desarrollo del *backend* para permitir que la interfaz corra adecuadamente.

En la programación del *backend* se decidió usar el lenguaje de programación PHP, este es un lenguaje pensado principalmente para los servidores, pues permite procesar el código HTML sin que el cliente sepa cómo se tiene distribuida la información del código con diferencia de JavaScript donde el cliente descarga el código y permite revelar información de la estructura de los datos. Para hacer uso de PHP es necesario llevar a cabo una instalación, pero ésta fue incluida durante la instalación de Vesta Panel, por lo que lo único que se procedió a hacer fue una prueba sobre el servidor para comprobar que esté corriendo de manera adecuada. Se programó un *script* en PHP para hacer una suma y desplegar un “Hola Mundo” y el resultado de la suma en el navegador, de esta forma se comprobaría que PHP estuviera funcionando adecuadamente sobre el servidor, el resultado se ve en la Fig.37; en esta figura se aprecia la principal característica de PHP, la cual es el preprocesamiento del código antes de enviarlo al cliente, en el rectángulo azul de la derecha se ve lo que recibe el cliente, mientras que en el rectángulo azul de la izquierda se ve el código fuente, como se observa del lado del cliente es invisible todas las operaciones realizadas del lado del servidor.

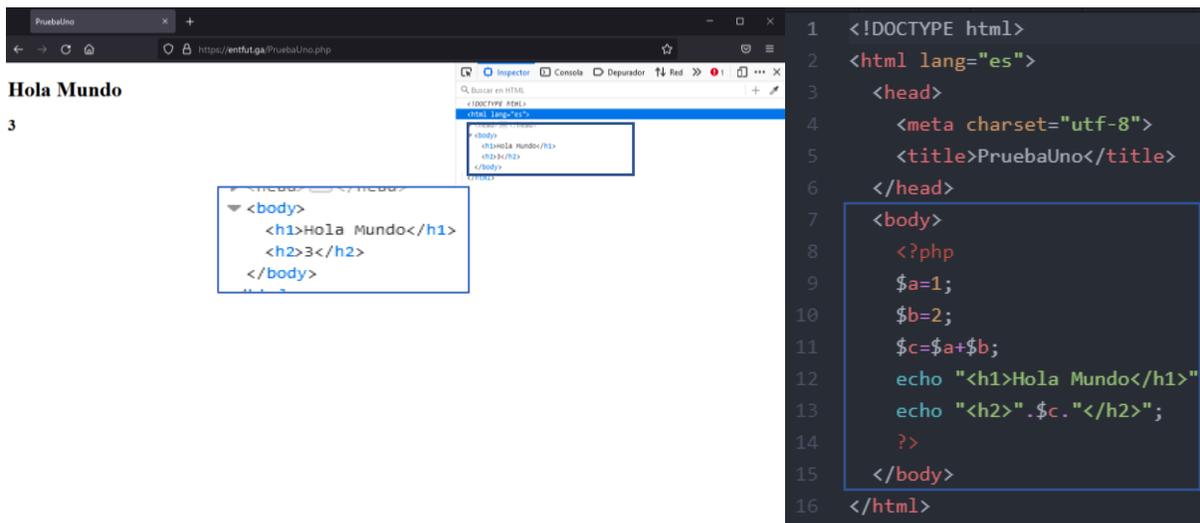


Fig. 37 Captura de pantalla del navegador y su código fuente

Durante el bocetaje de la interfaz se identificaron pantallas que iban a necesitar de la consulta a la base de datos como: obtener los datos de los jugadores, los datos del desempeño de cada jugador, las tablas de los dispositivos, etc. Para lograr traer estos datos al cliente desde la base, es necesario usar interfaces intermedias que los conecten; estas interfaces suelen conocerse como API, una API es un conjunto de funciones que permiten acceso a determinado software como acceso a características de bajo nivel o propietarias, detallando solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda, sin otorgar información acerca de cómo se lleva a cabo la tarea [68]. En los siguientes pasos se describirá en que consiste cada una de estas APIS.

API JUGADORES

Esta API será la encargada de realizar las cuatro operaciones básicas con la tabla JUGADORES en la base de datos, las cuales son: crear, leer, actualizar y eliminar. Para lograr lo anterior se hizo uso de los protocolos GET y POST, ambos protocolos permiten el envío de información, la cual será recibida por el servidor para saber cómo actuar. En el diagrama de flujo de la Fig. 38 se muestra cómo se estructuró la lógica de las diferentes posibilidades de petición, se decidió usar subrutinas para generar una visualización clara; a continuación, se explicará cada una de las subrutinas que componen al diagrama.

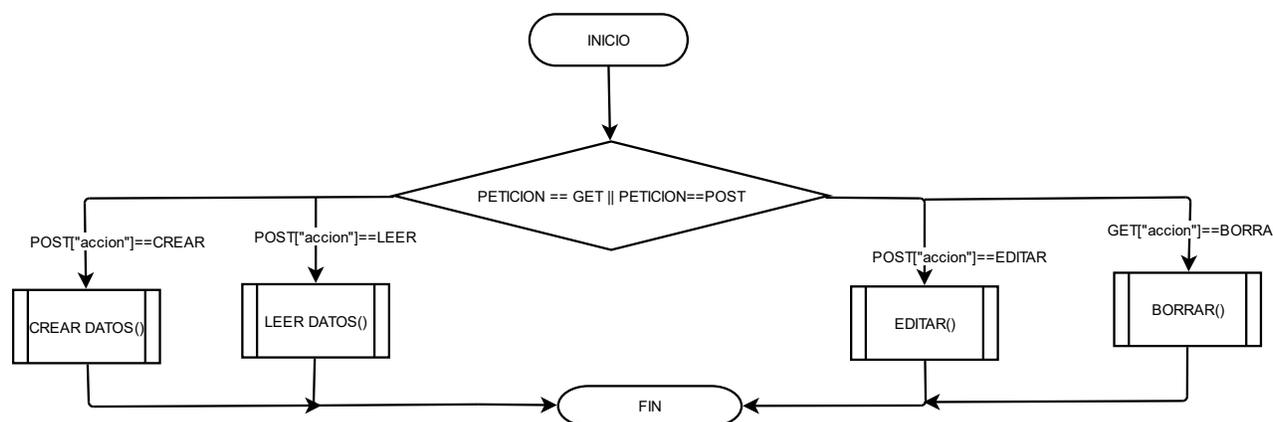


Fig. 38 Diagrama de Flujo de la API JUGADORES

- **CREAR DATOS:** Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo “acción” tiene como cadena de texto CREAR. Esta subrutina comienza con la

conexión a la base de datos, luego decodifica los datos enviados en el POST como el nombre del jugador y el *tracker* que se le asociará a este, si ya existe un registro con el nombre del jugador se lanza un mensaje de error, en caso de que sea el nombre de un jugador nuevo se guarda en la base de datos y se regresa un mensaje con un *ok*.

- LEER DATOS: Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo acción tiene como cadena de texto LEER. En este caso se conecta a la base de datos, luego realiza la consulta para traer todos los registros, al ser trabajado en PHP los datos traídos tienen forma de arreglo, por lo que para usarlos en el sistema de *frontend* es necesario codificarlos en formato JSON, finalmente la subrutina termina devolviendo un JSON con los datos.
- EDITAR: Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo acción tiene como cadena de texto EDITAR. Se conecta a la base de datos, luego se decodifican los otros datos enviados en la petición POST, de este modo se intenta ingresar la actualización del registro a la base de datos y en caso de que no se pueda realizar esta acción devuelve un mensaje de error, en caso de realizar la actualización correctamente se retorna un mensaje de *ok*.
- BORRAR: En este caso se manda a llamar a esta subrutina cuando la petición es de tipo GET y su campo acción tiene la cadena de texto BORRA, se hace por medio de un GET debido a una funcionalidad que se usará en la parte de *frontend* y al hacer uso de un GET hace más sencillo su implementación. El primer paso de esta subrutina es conectarse a la base de datos, se manda una petición a la base de datos para borrar el dato seleccionado y al igual que en EDITAR, en caso de realizar la acción correctamente se retorna un mensaje de *ok* y en el caso contrario se retorna un mensaje de error.

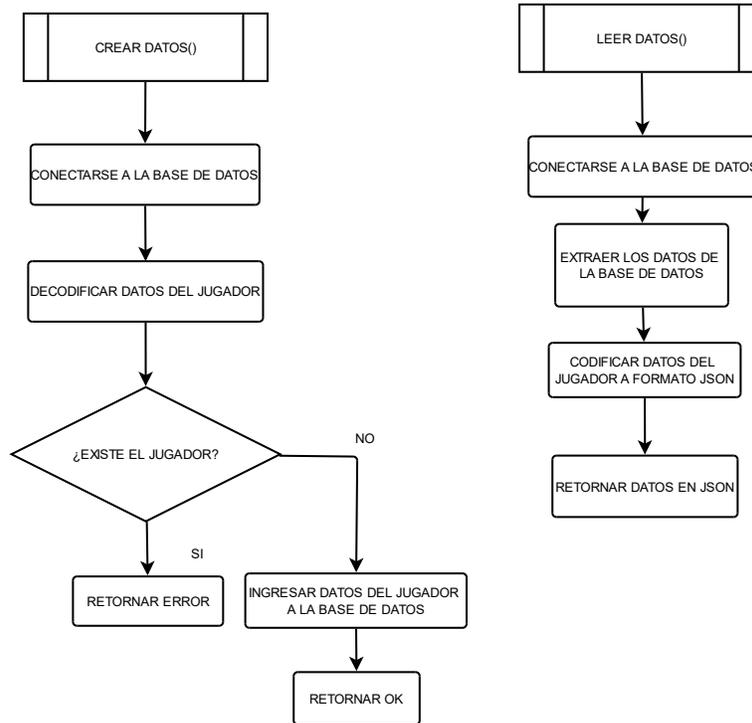


Fig. 39 Diagrama de flujo de las subrutinas crear datos (izquierda) y leer datos (derecha)

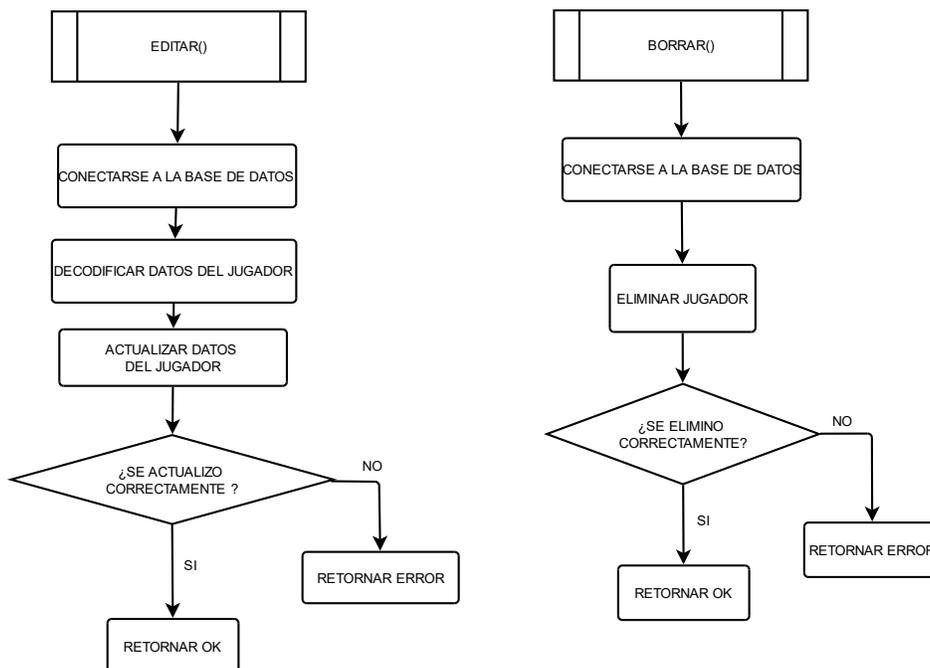


Fig. 40 Diagrama de flujo de las subrutinas editar (izquierda) y borrar (derecha)

API DISPOSITIVOS

Esta API es la encargada de gestionar y traer los datos desde la tabla DISPOSITIVOS, así que se estructuró la API muy parecido con la API JUGADORES. Esta API permite la interacción con la base de datos y crear las cuatro operaciones básicas (creación, lectura, actualización y eliminación), pero además de esto permite llevar un registro acerca de cuál fue la última hora de conexión. A continuación, se abordará la lógica con la que se estructuró cada una de las subrutinas.

- **CREAR DATOS:** Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo "acción" tiene como cadena de texto CREAR. Esta subrutina comienza con la conexión a la base de datos, luego decodifica los datos enviados en el POST como el nombre asignado al dispositivo y el tipo que es, si ya existe un registro con el nombre del dispositivo se lanza un mensaje de error, en caso de que sea un dispositivo nuevo se guarda en la base de datos y se regresa un mensaje con un *ok*.
- **LEER DATOS:** Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo acción tiene como cadena de texto LEER. En este caso se conecta a la base de datos, luego realiza la consulta para traer todos los registros, al igual que en la API jugadores estos deben de ser retornados en formato JSON para ser trabajado por el *frontend*.
- **EDITAR:** Se manda a llamar a esta subrutina cuando la petición es del tipo POST y su campo acción tiene como cadena de texto EDITAR. Se conecta a la base de datos, luego se decodifican los otros datos enviados en la petición POST, de este modo se intenta ingresar la actualización del registro a la base de datos y en caso de que no se pueda realizar esta acción devuelve un mensaje de error, en caso de realizar la actualización correctamente se retorna un mensaje de *ok*.
- **BORRAR:** En este caso se manda a llamar a esta subrutina cuando la petición es de tipo GET y su campo acción tiene la cadena de texto BORRA. El primer paso de esta subrutina es conectarse a la base de datos, se manda una petición para borrar el dato seleccionado y al igual que en EDITAR, en caso de realizar la acción correctamente se retorna un mensaje de *ok* y en el caso contrario se retorna un mensaje de error.
- **HORA:** Esta subrutina es mandada a llamar cuando la petición es del tipo GET y en su campo acción tiene la cadena de texto HORA. La subrutina comienza por obtener la hora actual, luego intenta actualizar la HORA del dispositivo seleccionado en la base de datos,

en caso de realizar la acción correctamente devuelve un ok y en caso contrario devuelve un mensaje de error.

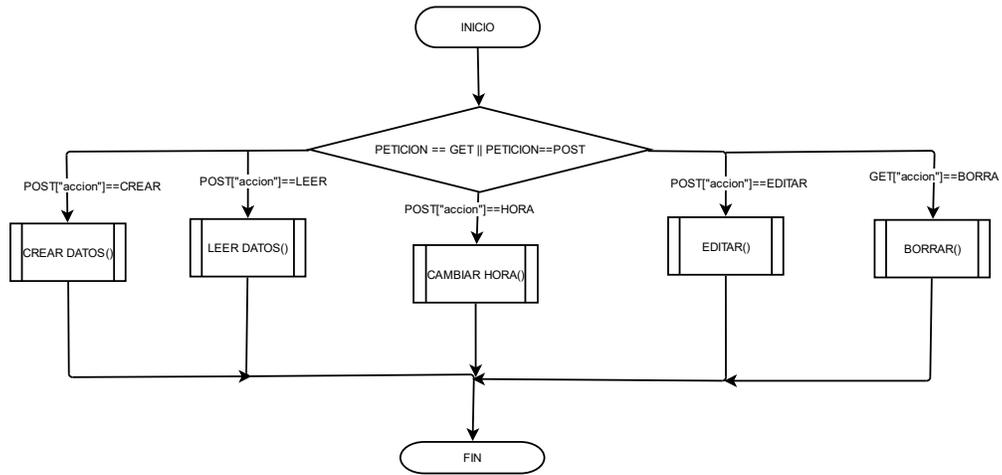


Fig. 41 .Diagrama de Flujo de la API DISPOSITIVOS

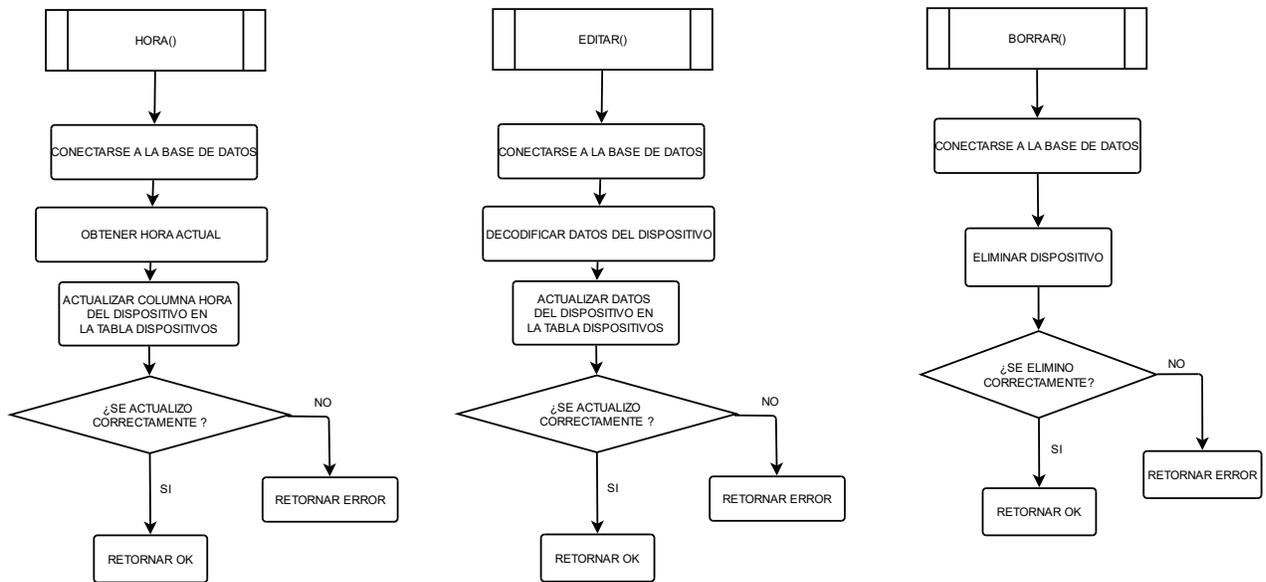


Fig. 42 Diagrama de flujo de las subrutinas Hora (izquierda), Editar (centro) y Borrar (derecha)

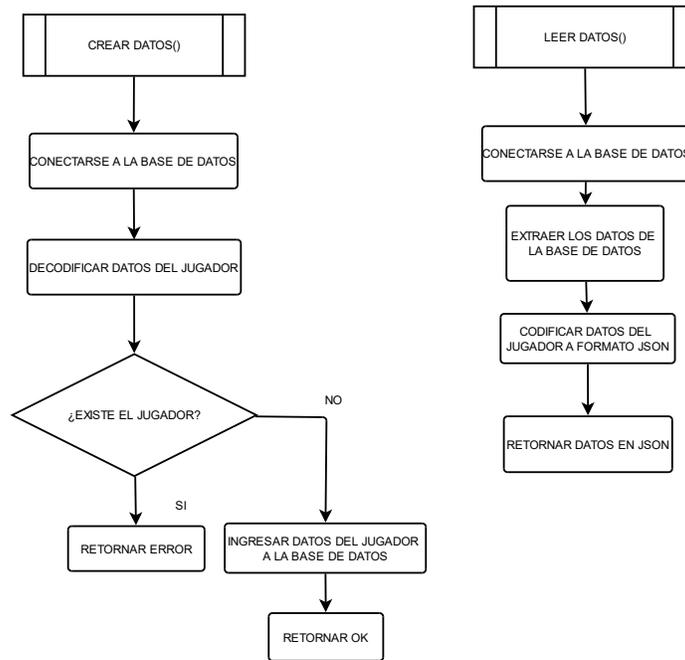


Fig. 43 Diagrama de flujo de las subrutinas crear datos (izquierda) y leer datos (derecha)

API RELACIÓN

Esta API trae la información de la vista JUNTAS, puede obtener la información de cómo se relacionan la tabla de jugadores y la tabla de dispositivos, de esta forma se podrá conocer cuál es el *tracker* que le corresponde a cada jugador al correr un ejercicio, así como la información de la entrada y la salida de su *tracker*. En la Fig.44, se puede ver el diagrama de flujo de esta API, el primer paso es validar que se reciba una petición POST y si esta contiene en su campo acción la cadena de texto EXTRAER, luego de esto se obtiene la información de jugador, su tracker, etc; finalmente se codifica en formato JSON y se retorna en este formato.

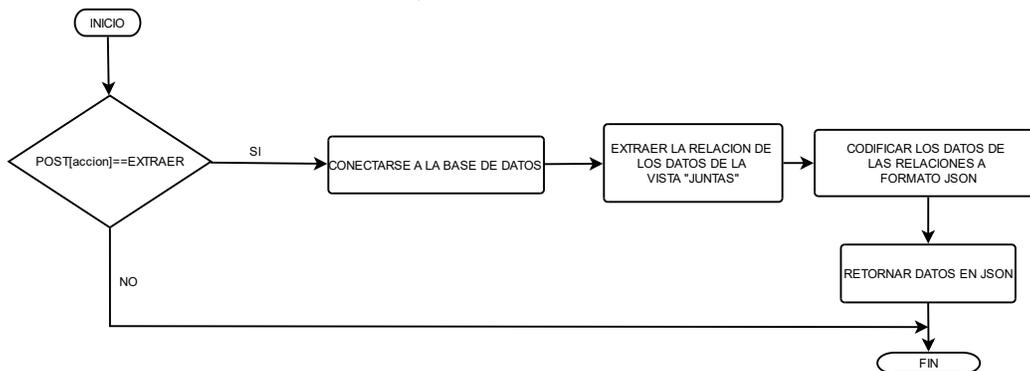


Fig. 44 Diagrama de flujo de la API RELACION

API EJERCICIOS

Esta API será la encargada de insertar los datos del jugador en la tabla correspondiente al ejercicio, así como extraer el porcentaje de acierto del jugador en ese ejercicio; para realizar ambas acciones lo que se hace es lo siguiente: primero verifica si la petición POST en su campo acción es igual a DATOS, si es así lo que hace es conectarse a la base de datos, luego decodifica los otros parámetros pasado al método POST como el tiempo y la efectividad, finalmente los ingresa en la correspondiente tabla del ejercicio; mientras que si la petición POST tiene en campo acción la palabra ACIERTO, este obtendrá todos los registros del jugador y calculará su porcentaje de efectividad.

API DATOS

En esta API se entregan las estadísticas de un determinado jugador como: velocidad promedio, porcentaje de acierto general, etc. Para lo anterior lo primero que hace es verificar que la petición que se reciba es de tipo POST con su campo acción igual a DATOS, siendo el primer paso la conexión a la base de datos, luego se decodifica cual es el jugador del cuál se quiere obtener la información y sobre qué ejercicio es el que se requiere esta información, después se obtienen los registros del jugador en la tabla del ejercicio indicada, por lo que el siguiente paso es obtener el tiempo promedio, el mejor tiempo y otros datos generales del ejercicio, finalmente se agrupan los datos por fecha para obtener el porcentaje de acierto por día y tiempo promedio por día, finalmente se pasan todos estos datos a un JSON y se retorna al sistema de *frontend*.

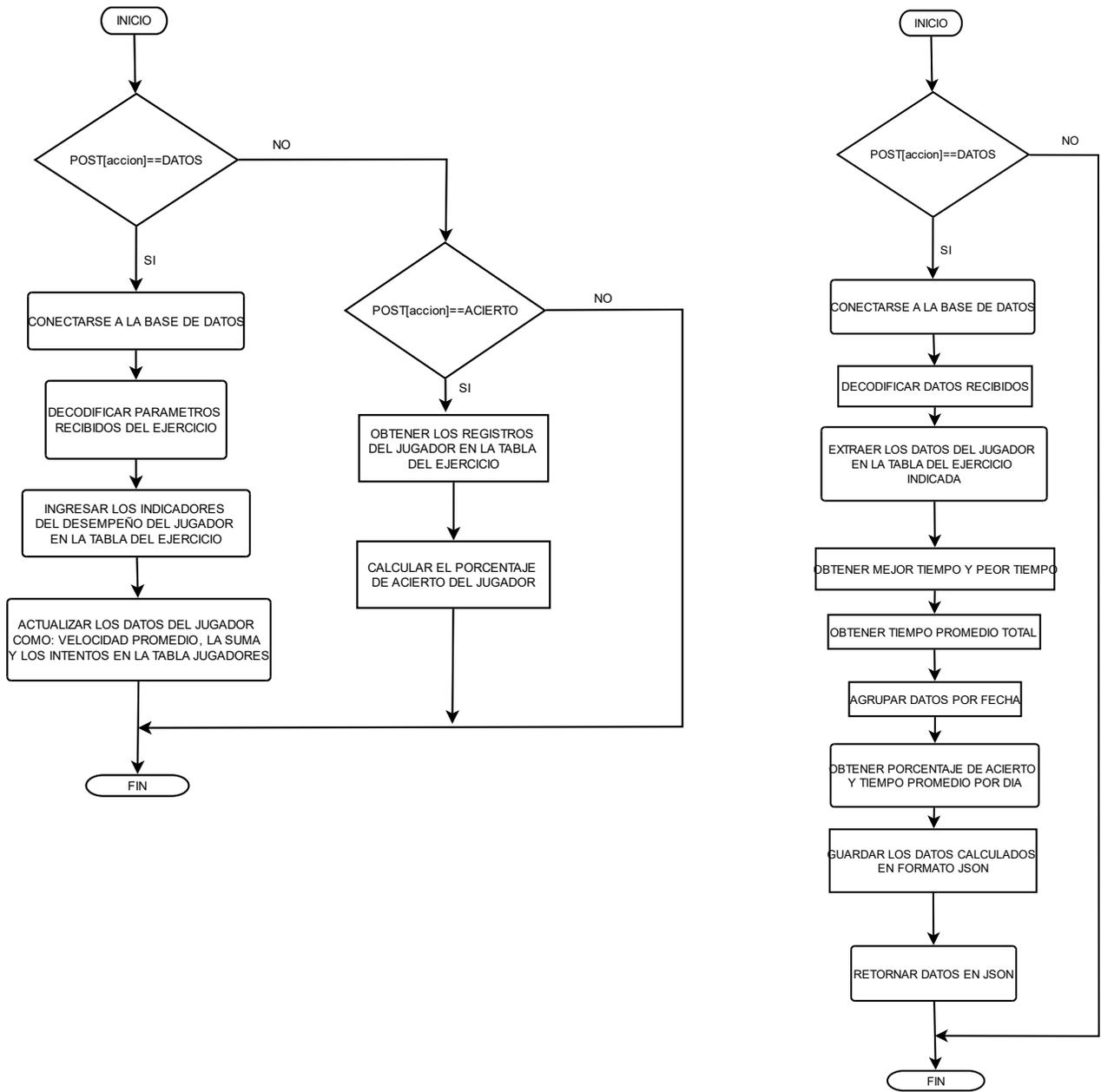


Fig. 45 Diagrama de flujo de la API EJERCICIO (izquierda) y API DATOS (derecha)

5.5.3. DISEÑO DE FRONTEND

En esta parte se abordará el diseño visual que se le terminó dando a la interfaz. Para el diseño de *frontend* es importante aclarar los tres lenguajes que son necesarios: HTML, CSS y Javascript, estos son las herramientas usadas universalmente para el desarrollo web y al igual que en el *backend* se usará PHP, pero sólo en partes puntuales, donde no se necesite retroalimentación del frontend, para obtener datos directos desde la base de datos. HTML es el lenguaje que le da la estructura a un sitio web, organiza la información a través de etiquetas para indicar a un navegador la forma en como mostrarla. CSS es el lenguaje para dar estilos y lograr mostrar la página de una manera más estética visualmente hablando. Finalmente, Javascript es el lenguaje de programación que permite realizar acciones desde el lado del cliente.

Para el desarrollo de este trabajo se decidió usar el enfoque de *responsive design*, esto significa que se podría generar el sitio web para que se adapte tanto en computadoras como en smartphones, se decidió usar el enfoque *PC first* lo cual indica que primero se trabajará diseñando las pantallas para ser vistas en computadoras y luego se agregarán las reglas para que se adapte para pantallas de smartphones. Para la interacción de cuáles reglas se deben de aplicar en dispositivos móviles y cuáles en una pantalla de PC existen los llamados *mediaQueries*, los cuales son agregados en el código CSS y estos determinan, a partir del tamaño de pantalla, qué ciertos estilos se aplican en la pantalla web.

Otro aspecto necesario antes de comenzar cualquier proyecto web es que los navegadores aplican estilos por defecto a los elementos, en si estos estilos no suelen variar mucho pero sí lo suficiente para que algunos elementos en ocasiones se encimen y no se vean de la forma deseada, por lo cual es necesario aplicar hojas normalizadoras, en este caso se decidió usar “normalize” que es una hoja de estilos de CSS de código abierto que permite evitar los inconvenientes de los distintos estilos de los navegadores.[69]

Por otro lado, para generar un prototipado rápido se decidió usar Bootstrap, este es una biblioteca de código abierto lanzada por *Twitter* que contiene tanto estilos de CSS como de funciones de Javascript que permiten acceder a elementos como botones, títulos, márgenes, etc. De forma rápida, para esto solo es necesario ligar la biblioteca con una etiqueta “href” en el documento HTML de la página.

Las imágenes usadas en este proyecto fueron obtenidas de diferentes fuentes de imágenes libres, con la finalidad de evitar incurrir en faltas relacionadas a derechos de autor, las páginas usadas

fueron: freelmages[70], Pixabay[71] y Unsplash[72]. Del mismo modo se usaron íconos con licencias libres y estos se obtuvieron de fontawesome, una biblioteca con más de 1000 iconos para usarlos gratuitamente [73].



Fig. 46 Pantalla Principal, de lado izquierdo se puede ver la vista web con la opción de elementos del campo resaltada y de lado derecho la vista en un smartphone.

La pantalla principal se puede observar en la Fig.46 donde se puede ver el logo de la página en la parte superior izquierda, del mismo modo se puede observar en el lado superior derecho de la vista en computadora se encuentra el nombre del usuario, además si se hace clic sobre éste, se despliega el menú, las opciones de cerrar sesión y configurar perfil; mientras que en la versión móvil este menú se encuentra escondido y se despliega al hacer clic en las tres barras que se encuentran en la esquina derecha; de la misma forma se puede apreciar el diseño en tarjetas para las opciones, estas tienen un fondo sombreado que permite a las imágenes generar un contraste con las letras blancas a las cuales también se les agregó el efecto de sombra, en la versión web estas tarjetas tienen un efecto de brillo al pasar el cursor sobre ellas, ayudando al usuario a saber sobre qué opción está posicionado y hacer más interactiva la navegación, objetivo que se buscaba en las especificaciones del proyecto.

Mientras que en la Fig.47 se puede observar la barra de menú que permite llevar a las diferentes pantallas cuando no se está en la pantalla principal. Este menú normalmente está comprimido en la parte izquierda de las pantallas y al pasar el cursor sobre la barra este se despliega y colorea

de color azul el fondo de la opción sobre la cual está posicionada el cursor, del mismo modo se puede observar este mismo menú, pero en su vista en tablets y en smartphones donde se tiene una menor pantalla, en estos casos se decidió ubicar el menú en la parte inferior de la pantalla y solo dejar los íconos con cada una de las diferentes pantallas del menú principal.



Fig. 47 Vista del menú en diferentes pantallas, menú comprimido en PC (izquierda), menú desplegado en PC (centro) y menú en tablets y dispositivos móviles (derecha).

PANTALLA DISPOSITIVOS

Del mismo modo en la Fig.48 se puede observar el diseño que se le dio a la pantalla de elementos de campo, en esta se traen los dispositivos desde la tabla DISPOSITIVOS de la base de datos, se dividieron éstos por categoría si era un *tracker*, una *portería* o un *access point*. En la tabla de cada una de las categorías, se colocó el nombre del dispositivo, cuál era su tema de entrada, su tema de salida, la última hora en la que fue comprobada la conexión y botones de acciones para cada dispositivo las cuales fueron: eliminar, editar y comprobar conexión. A continuación, se describirá la acción y programación de cada uno de estos botones.

- Eliminar: Este botón al ser accionado primero despliega un mensaje de confirmación de si se quiere eliminar el dispositivo, si se confirma que se desea eliminar el dispositivo entonces el cliente se conecta con la base de datos mediante AJAX y envía a la API DISPOSITIVOS una solicitud GET con su campo acción igual a ELIMINA, de este modo el sistema de *backend* se encarga de eliminar el dispositivo de la base de datos y si se completó con éxito la eliminación, se pone una notificación en color verde que el dispositivo fue eliminado correctamente y se quita el dispositivo del *frontend*, de lo contrario se muestra una notificación de color rojo informando que hubo un error al eliminar.

- **EDITAR:** Este botón redirecciona al usuario a la pantalla de EDITAR dispositivos pasando por método GET el ID que le corresponde en la base de datos al dispositivo para que complete automáticamente los campos de información y haga la labor de edición de una forma más fácil.
- **COMPROBAR CONEXIÓN:** Este botón requiere un proceso complejo, primero intenta conectarse con el dispositivo por su tema de configuración enviando un mensaje por MQTT, en caso de que el dispositivo le responda indica que se encuentra conectado por lo que el siguiente paso es registrarlo en la base de datos, entonces se conecta con la API DISPOSITIVOS enviando una petición POST con su campo acción igual a HORA, de este modo el sistema de *backend* se encarga de actualizar la hora de conexión en la base de datos, finalmente se muestra una notificación con un mensaje en color verde de que el dispositivo se encuentra conectado y se actualiza la hora en la pantalla de *frontend*; en caso de que el dispositivo no haya respondido se despliega una notificación en color rojo con un mensaje de “no respondió el dispositivo”, mientras que si el dispositivo sí respondió, pero hubo un problema al actualizar en la base de datos, aparecerá una notificación color rojo con un mensaje de “error al ingresar en la base de datos”.

DISPOSITIVOS CONECTADOS				
Trackers				
Dispositivo	Entrada	Salida	Última hora de de conexion	Acciones
TR1	p4	p7	2021-02-10 00:31:11	[Iconos]
YAIR	H1	H2	2021-03-01 19:55:37	[Iconos]
TR300	X1	X2	2021-04-05 16:12:04	[Iconos]
TR301	X2	X3	2021-04-05 16:14:09	[Iconos]
Tracker	medidas	potencia	2021-04-13 14:54:03	[Iconos]
tracker2	med	pot	2021-05-09 18:41:54	[Iconos]

Hotspot				
Dispositivo	Entrada	Salida	Última hora de de conexion	Acciones
AP40	p1	p2	2021-02-10 00:22:17	[Iconos]
AP300	F1	F2	2021-04-05 16:13:01	[Iconos]
AP-1	AP1	P1	2021-04-13 14:52:58	[Iconos]

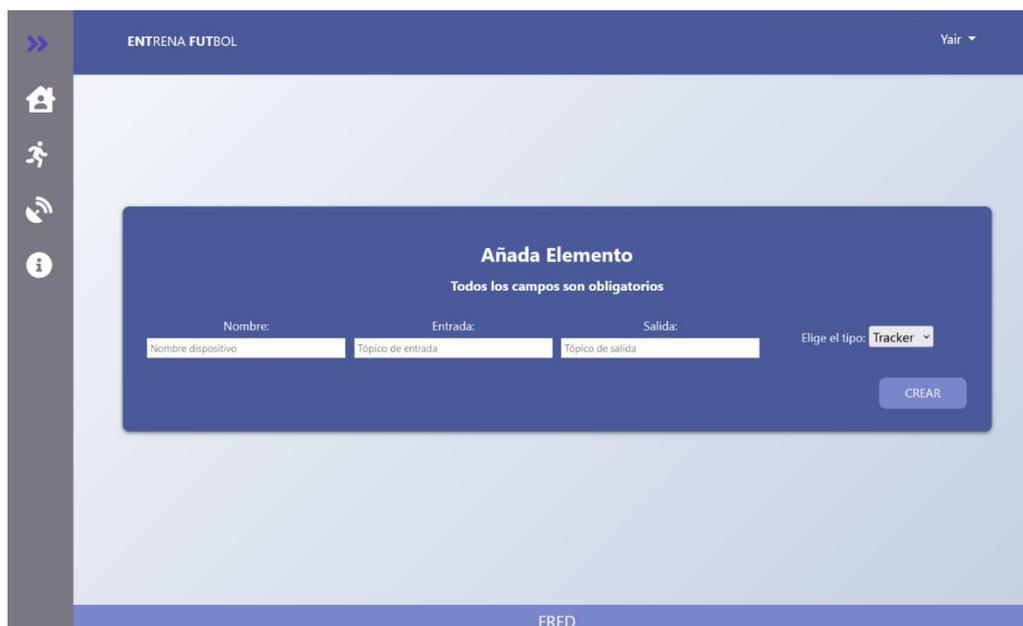
Fig. 48 Captura de la pantalla elementos del campo

Finalmente, en la parte inferior derecha de esta pantalla elementos del campo se encuentra un signo “más” el cual es para agregar un nuevo dispositivo en la base de datos, al hacer clic sobre este redirige a la misma página que al apretar un botón EDITAR, solo que con los campos vacíos

pues no se le pasa información por método GET, dando la apariencia que es una página distinta exclusivamente creada para añadir dispositivos, pero es una optimización del código editar.

PANTALLA CREAR ELEMENTO

Como se mencionó anteriormente se llega a esta pantalla cuando el usuario aprieta el signo de más en la pantalla de elementos del campo. Esta pantalla está compuesta por un formulario al centro, el cual debe de ser rellenado por el usuario con datos como: el nombre del dispositivo, la entrada y salida, además del tipo; cuando el usuario rellena todos estos campos y presiona el botón “crear” se intenta conectar con el dispositivo el cual debe haber sido preconfigurado desde la IDE de Arduino anteriormente con el nombre dado al dispositivo en el formulario, en caso de que se encuentre el dispositivo este responderá, lo cual significa que se encuentra conectado a la red y que realizó la configuración inicial, luego de esto el *frontend* enviará una petición POST en el campo acción igual a CREAR a la API DISPOSITIVOS, en caso de que se haya insertado correctamente a la base de datos se desplegará un mensaje color verde que el dispositivo fue creado correctamente; en caso de que no se haya podido comunicar con el dispositivo aparecerá una notificación en color rojo con el mensaje de que no se pudo comunicar con el dispositivo, por otro lado si no se pudo conectar a la base de datos aparecerá un mensaje con el texto “no se pudo conectar a la base de datos”.



The screenshot shows a web application interface for adding a new element. The page has a dark blue header with the text "ENTRENA FUTBOL" and a user name "Yair" with a dropdown arrow. A vertical sidebar on the left contains icons for home, a person, a signal tower, and an information icon. The main content area is light blue and features a central dark blue form titled "Añada Elemento" with the subtitle "Todos los campos son obligatorios". The form contains three input fields: "Nombre:" with the placeholder "Nombre dispositivo", "Entrada:" with the placeholder "Tópico de entrada", and "Salida:" with the placeholder "Tópico de salida". To the right of these fields is a dropdown menu labeled "Elige el tipo:" with "Tracker" selected. A "CREAR" button is located at the bottom right of the form. The footer of the page is dark blue and contains the text "FRED".

Fig. 49 Vista de la pantalla en PC de Añadir elemento

PANTALLA EDITAR ELEMENTO

Esta pantalla contiene varias similitudes con la pantalla crear elementos, se accede a esta pantalla cuando el usuario presiona el botón editar de alguno de los dispositivos en la pantalla de elementos de campo, de este modo se pasa información del dispositivo mediante el método GET para que se completen automáticamente los campos del formulario, con esto se evita que el usuario edite cada uno de los campos nuevamente y en su lugar sólo haga el cambio sobre el campo que necesita modificarse; otra diferencia con respecto a la pantalla de crear, es que cambia el texto del título pues dice “Edite elemento” en lugar de “Añada elemento”, además el texto del botón cambia, pues en lugar de decir “crear” dice “guardar”, estos cambios son posibles al realizar una validación del tipo “if” comprobando que se hayan pasado parámetros por el método GET . Del mismo modo que en la pantalla crear, es necesario verificar con el dispositivo y con la base de datos para comprobar que fue recibido por el dispositivo e ingresado el cambio en la base de datos. En la comprobación con el dispositivo no es solo necesario conectar con él, sino esperar una confirmación pues se podría cambiar el tema de entrada y de salida, así que para tener la seguridad de que estos cambios se realizaron correctamente se envía un mensaje a su tema “config” y en caso de realizar los cambios el dispositivo responde con un mensaje en el tema “config_gen” que contiene en el cuerpo del mensaje el nombre del dispositivo, de este modo se comprueba que los cambios se realizaron adecuadamente; el siguiente paso es realizar los cambios en la base de datos para ello se envía una petición POST a la API DISPOSITIVOS con la cadena de texto EDITAR en el campo acción y esta se encarga de realizar la actualización de los datos, si todos los pasos anteriores se realizaron correctamente entonces se mostrará una notificación en color verde indicando que se realizó correctamente la actualización de los datos. En caso de que no se haya podido comunicar con el dispositivo aparecerá una notificación en color rojo con el mensaje de que no se pudo comunicar con el dispositivo, mientras que si no se pudo conectar a la base de datos aparecerá un mensaje con el texto “no se pudo conectar a la base de datos”.

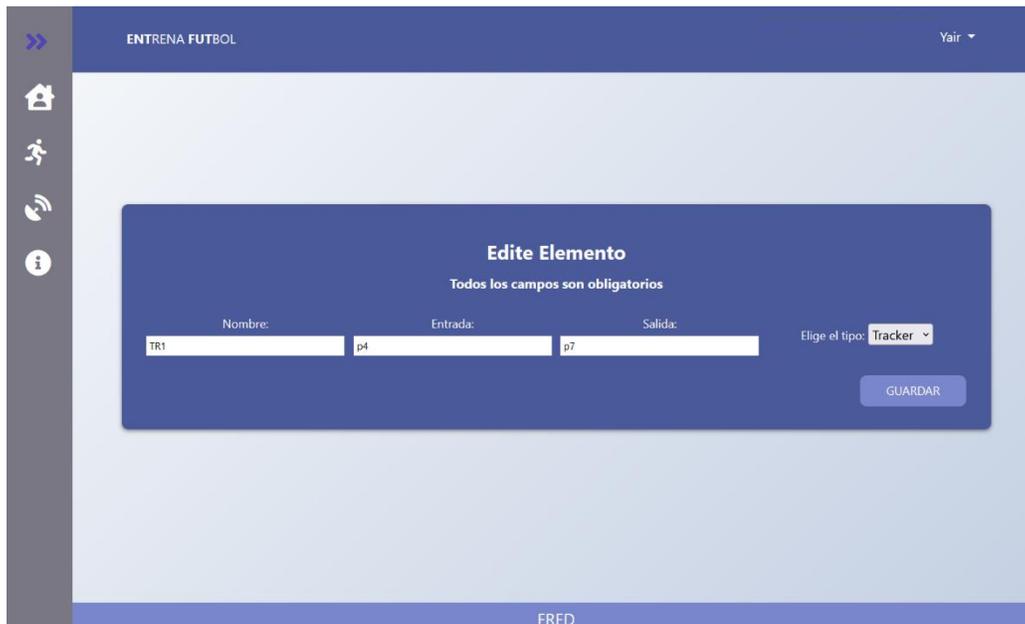


Fig. 50 Vista de la pantalla de “Editar elemento” como se observa, es casi idéntica a la pantalla “crear elemento”

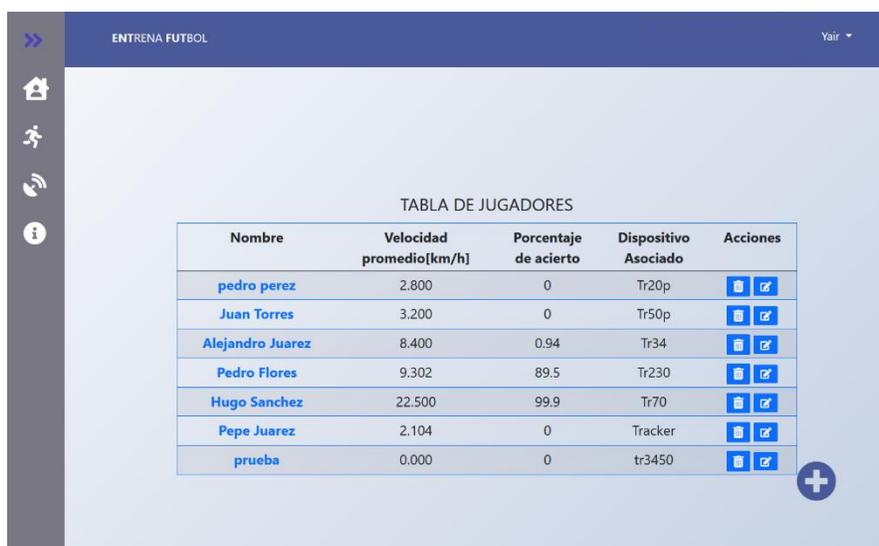
PANTALLA ESTADÍSTICAS

En esta pantalla se muestra toda la información de los jugadores, su velocidad promedio, su porcentaje de acierto promedio, así como su *tracker* asociado. Para poder desplegar la información se hace uso de la API JUNTAS enviándole una petición POST con su campo acción igual a DATOS, de este modo trae la información del nombre del jugador, así como de su *tracker* asociado, de la misma forma se hace uso de la API JUGADORES enviándole una petición POST con el campo acción igual a LEER y ésta trae la información acerca de los jugadores. Al igual que en la pantalla de elementos del campo, se pueden hacer las acciones de editar, crear y eliminar al jugador, pero además se puede ver más detalladamente información de las estadísticas del jugador al presionar su nombre, ya que redirige a la página estadísticas de jugadores.

En caso de que se presione la opción de eliminar jugador se pedirá al usuario una confirmación, si confirma que desea borrar al jugador, entonces se conecta con la API JUGADORES y se le envía una petición GET con el campo acción igual a BORRA con el ID del jugador, de este modo la API es la encargada de realizar la eliminación del registro del jugador en la base de datos, en caso de realizarse correctamente la acción, se mostrará una notificación en color verde con el texto

“Jugador eliminado correctamente”, de lo contrario se muestra una notificación en color rojo con el texto “Hubo un problema al eliminar jugador”.

La opción de editar (el icono del lápiz) redirige al usuario a la pantalla de editar jugador, donde se realizan estos cambios y se le pasa por medio del método GET los datos para hacer la edición más sencilla. Mientras que para acceder a la opción de crear jugador es necesario presionar al signo “más” que aparece en la esquina inferior derecha, este redirecciona a la página de editar jugador, solo que no se le pasan datos por el método GET, así que aparecerán todos los campos vacíos al momento de editar.



Nombre	Velocidad promedio[km/h]	Porcentaje de acierto	Dispositivo Asociado	Acciones
pedro perez	2.800	0	Tr20p	 
Juan Torres	3.200	0	Tr50p	 
Alejandro Juarez	8.400	0.94	Tr34	 
Pedro Flores	9.302	89.5	Tr230	 
Hugo Sanchez	22.500	99.9	Tr70	 
Pepe Juarez	2.104	0	Tracker	 
prueba	0.000	0	tr3450	 

Fig. 51 Vista de la pantalla de estadísticas

PANTALLA EDITAR JUGADOR/AÑADIR JUGADOR

Como se dijo anteriormente ambas pantallas son la misma sólo cambian en pequeños detalles, por ejemplo cambia el título en una se tiene “edite jugador”, mientras que en la otra se tiene “añada jugador” y cambia el texto del botón, pues en la pantalla de edite jugador el botón dice “guardar” mientras que en la opción de crear jugador lo que se tiene es el texto “crear”. El funcionamiento de estas pantallas es similar a las pantallas de editar/crear dispositivo pero con la diferencia que no realiza alguna validación con un dispositivo, si no que realiza únicamente la validación con la base de datos.

En el caso de “crear jugador” cuando el usuario presiona el botón de crear, se toman los datos del formulario y se conecta con la API JUGADORES, pasándosele una petición del tipo POST con el campo acción igual a CREAR de este modo se inserta en la base de datos, si todo lo anterior resultó correctamente, se mostrará un mensaje en verde con el texto “Jugador creado correctamente”; mientras que en caso de que haya habido un error se mostrará un mensaje en color rojo con el texto “Hubo un error al insertar en la base de datos”. Por otro lado, en la pantalla “editar jugador”, cuando el jugador hace clic en “guardar” este se conecta de igual forma con la API JUGADORES, pero en este caso envía la petición POST con el campo acción igual a EDITAR, entonces si se logró actualizar correctamente se mostrará un mensaje en color verde con el texto “Jugador editado correctamente”; mientras que si no se logró actualizar los datos mostrará un mensaje en color rojo con el texto “Hubo un error al insertar en la base de datos”.



Fig. 52 Vista de la pantalla de editar/crear jugador

PANTALLA ESTADÍSTICAS DEL JUGADOR

Se accede a esta pantalla únicamente cuando desde la pantalla de estadísticas se da clic sobre el nombre de un jugador, eso redirecciona al usuario hacia esta página donde se muestran las estadísticas de cada jugador, para esto al momento de dar clic sobre la pantalla Estadísticas se pasan parámetros por método GET para indicar cuál será el jugador del cual se traerá la información de cada ejercicio, para esto se contacta con la API datos, la cual recopila la información de cada ejercicio, los datos que trae y se muestran en el *frontend* son:

- El mejor tiempo del jugador en determinado ejercicio.
- Porcentaje de acierto: Este es el porcentaje de veces que el jugador realizó correctamente el ejercicio respecto al total de las veces que el jugador lo llevó a cabo.
- Tiempo promedio: Se calcula haciendo el promedio del tiempo total que le tomó al jugador realizar el ejercicio.
- Historial de tiempo promedio: Muestra en una gráfica ordenada cronológicamente por días el tiempo promedio del jugador en ese ejercicio.
- Historial de porcentaje de aciertos: Muestra en una gráfica ordenada cronológicamente el porcentaje de acierto de un jugador en un determinado día en ese ejercicio.
- Historial de aciertos: Muestra en una gráfica ordenada cronológicamente los intentos del jugador en determinado día en ese ejercicio.



Fig. 53 Vista de la pantalla de estadísticas del jugador en formato móvil

PANTALLA EJERCICIOS

Esta pantalla contiene los ejercicios disponibles a entrenar, como se ve en la Fig. 54, se puede observar que estos ejercicios son desplegados en tarjetas, en la parte superior se encuentra una imagen ilustrativa de los objetos y la disposición de éstos para poder realizar el ejercicio, las porterías se representan con cuadrados y los *access point* con círculos, justo debajo de esta imagen se encuentra una pequeña descripción del ejercicio y un botón que redirige a la página de éste.

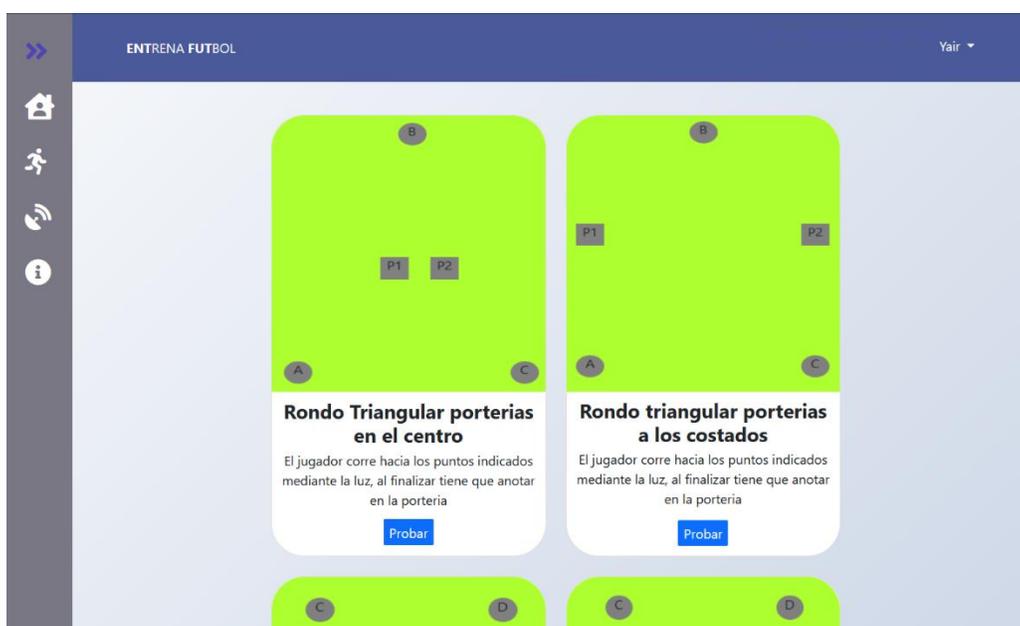


Fig. 54 Vista de la pantalla de ejercicios

PANTALLAS DE LOS EJERCICIOS

Estas pantallas varían un poco respecto a la dinámica de cada ejercicio, pero comparten varias características. La primera característica que comparten son que se tiene un engrane el cual al ser

presionado despliega un menú para configurar qué dispositivo se colocará en cada una de las posiciones.

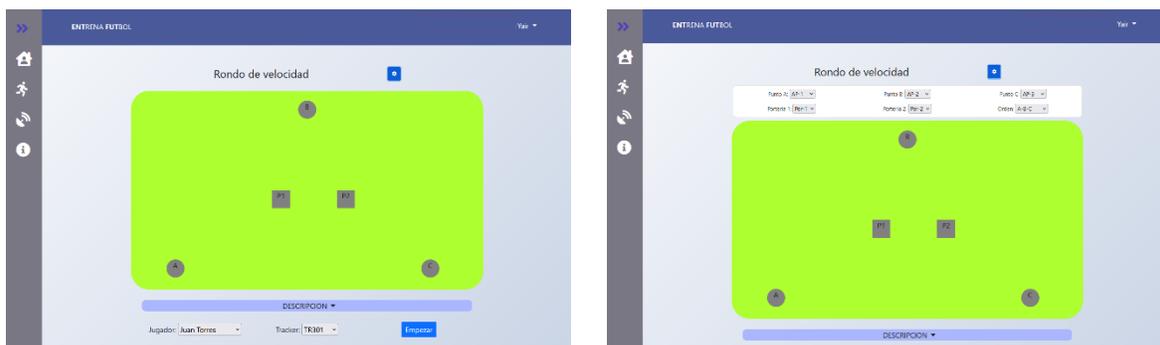


Fig. 55 Menú de configuración compacto(izquierda) y menú de configuración desplegado (derecha).

En la parte de abajo de esta página, se puede ver una pequeña descripción que igual se puede compactar o desplegar; del mismo modo se puede observar un menú desplegable donde se selecciona al jugador que se desea haga el ejercicio, automáticamente al seleccionar al jugador se seleccionará su *tracker*, evitando que el usuario tenga que realizar pasos adicionales aunque si así lo desea, el *tracker* lo puede seleccionar manualmente en caso de que el jugador esté entrenando con otro que no sea el que tenga ligado en la base de datos. Finalmente, cuando el usuario haya configurado el ejercicio podrá dar clic en empezar para iniciarlo, en la tabla de abajo, que se ve en la Fig.57, se puede observar el tiempo realizado por el jugador en cada una de las partes del ejercicio (columna de en medio) y se tienen los tiempos promedio del ejercicio de cada una de estas partes (columna derecha) en caso de que el tiempo hecho por el jugador resulte menor o igual que el tiempo promedio este se colocará en color verde con una flecha hacia arriba, indicando que fue un buen rendimiento, en caso de que el tiempo realizado por el jugador sea mayor al tiempo promedio este se colocará en rojo con una flecha hacia abajo mostrando que el desempeño del jugador no fue adecuado, en la última fila se tiene el porcentaje de acierto del jugador en el ejercicio.



Fig. 56 Pantalla del ejercicio uno con la descripción desplegada



Fig. 57 Pantalla del ejercicio uno luego de que se realizara el ejercicio.

5.6. CIRCUITOS ELECTRÓNICOS

Como se ha mencionado anteriormente, para el desarrollo de este trabajo se detectaron tres tipos de dispositivos a usar: *tracker*, *access point* y porterías, en esta sección se explicará cómo está compuesta la parte electrónica de cada uno de estos dispositivos. Para el diseño de los circuitos de forma digital, se usó Fritzing [88] y se usaron algunos componentes extraídos de GitHub [82][83][84][85][86].

5.6.1. TRACKER

Para el desarrollo del *tracker*, se utilizó un ESP8266-01 por tener una presentación más compacta con respecto al ESP8266-12E (su versión más común), aunque como se ve éste en la Fig.58, tiene como desventaja que sólo se tienen 4 pines de propósito general, detalle que no interfiere con las funciones que se tienen pensadas para el *tracker*. Se diseñó el circuito para que tuviera dos leds, uno de color verde para indicar si se encuentra encendido y conectado a una red wifi con conexión a internet; mientras que de igual modo se añadió un led de color rojo para indicar que el *tracker* se encuentra enviando mediciones de señal, a ambos leds se les agregó una resistencia de 330 ohm para regular la corriente y evitar quemar los leds. Además, se le agregó una batería de litio de 3.7V como fuente de alimentación y un switch para poder encender y apagar el *tracker*.

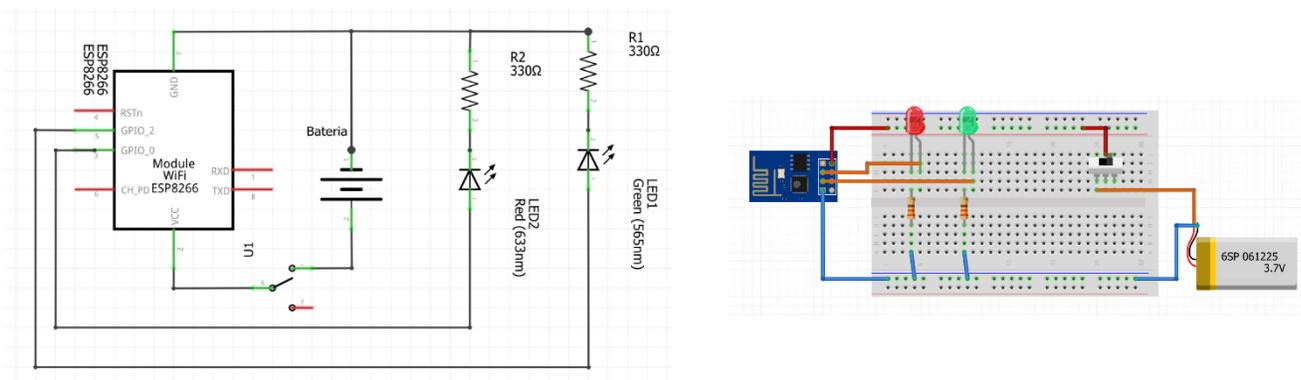


Fig. 58 Diagrama de conexiones esquemático (izquierda) y diagrama en protoboard (derecha) del circuito del tracker

5.6.2. ACCESS POINT

Para los *access point* se diseñó el circuito de la Fig.59, donde se puede apreciar que el controlador usado fue un ESP32, para ser más específico el modelo que se implementó fue un ESP32U, éste

tiene como característica que se le puede agregar una antena wifi externa para poder amplificar la potencia de la señal. De la misma forma, se agregó un switch para prender o apagar el dispositivo, para este caso como fuente de alimentación se usó una pila de Li-ion, con un voltaje de salida de 5V y hasta 2000mA. Finalmente se añadieron leds para que el jugador sepa que debe de realizar alguna acción con el *tracker* y un zumbador piezoeléctrico de 12V para señalar sonoramente en momentos en que no se alcancen a percibir las luces de los LEDs. Para esto se usó un transistor en configuración de switch, pues las salidas de corriente y voltaje del ESP32 no servirían para operar los componentes, así que también se tuvo que agregar un elevador de voltaje para obtener los 12V de alimentación para el zumbador; de este modo usando el transistor y el elevador de 12V, con un solo pin del ESP32 se podrían prender y apagar los componentes. Las operaciones que se hicieron para calcular el valor de la resistencia R4 para saturar al transistor, están basadas en la corriente de base como se muestra a continuación:

$$i_b = i_c / \beta_{min} \dots \dots \text{ec.1}$$

Debido a que se usaron 3 leds en configuración serie y cada uno de los leds usa alrededor de 20mA para funcionar y el zumbador piezoeléctrico trabaja a 25mA, entonces la corriente del colector (i_c) será igual a 85mA. La ganancia mínima del transistor (β_{min}) fue obtenida de su *datasheet* con valor de 75 unidades [74], por lo que en la ecuación 1 si se sustituyen los valores será igual a :

$$i_b = \frac{85mA}{75} = 1.13mA \dots \dots \text{ec.2}$$

Ya con el valor obtenido de la corriente necesaria en la base para saturar al transistor, será posible calcular el valor de R4, para ello es necesario aclarar que a pesar de que el ESP32 se alimenta con 5V las salidas de los pines digitales sólo ofrecen un voltaje (V_o) de 3.3V por lo que este valor será el utilizado en los cálculos, finalmente es necesario tomar en cuenta el voltaje base-emisor (V_{BE}) que es de 0.7 V, de este modo el cálculo de R4 quedaría:

$$R4 = \frac{V_o - V_{BE}}{i_b} = \frac{3.3V - 0.7V}{1.13mA} = 2300 \Omega \approx 2.2k\Omega \dots \dots \text{ec.3}$$

Debido a que R4 calculado sería 2,300 ohm y este no existe comercialmente, se usará el valor más cercano, sin que supere al valor obtenido, el cual es 2.2kΩ.

Mientras que para el cálculo de la resistencia R1, será necesario el uso de la ley de ohm, debido a que se usó LED ultrabrillantes color verde, el voltaje que consume cada uno de ellos es de 3.2 V y ya que son 3, consumirían 9.6V, por lo que el cálculo de R1 quedaría así:

$$R1 = \frac{\Delta V}{i} = \frac{12V - 9.6V}{60mA} = 40\Omega \dots \text{ec.4}$$

El resultado fue de 40 ohm, por lo que se optó por el valor comercial más cercano, el cual es 47 ohm.

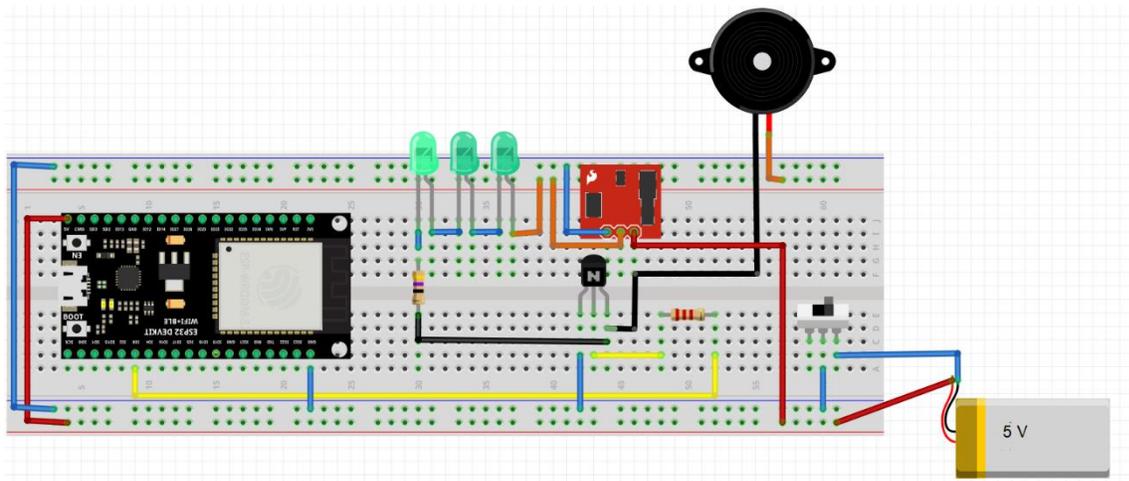


Fig. 59 Diagrama sobre una protoboard de un *access point*.

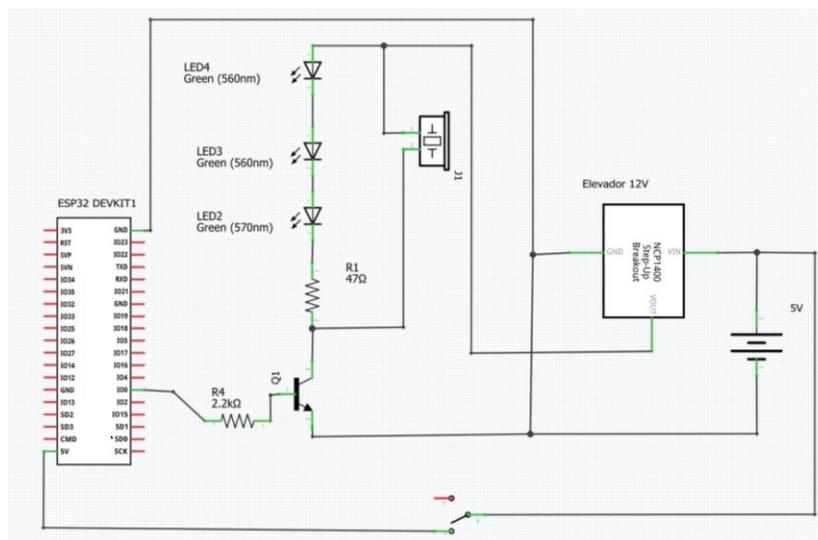


Fig. 60 Diagrama esquemático de un *access point*.

5.6.3. PORTERÍAS

En este circuito se usó el ESP826612-E, esto debido a que se usaron más de 4 pines GPIO causando que no se pueda usar el ESP8266-01. Por la parte de la alimentación se decidió usar baterías LiPo de 3.7V a 1200 mA, ya que al realizar las porterías con tubos de 2 pulgadas e ir la batería dentro de éste, el tamaño de la batería juega un papel crucial y esta batería es la de mayor capacidad que se ajusta a las dimensiones; el ultrasónico y las tiras LED trabajan a 5V, así fue necesario colocar un elevador de voltaje, para poder alimentar estos elementos.

Como se mencionó, se usó también un ultrasónico con el fin de sensar cuando un balón haya pasado por la portería; de la misma manera se añadieron dos tiras LEDs, el modelo añadido fue el NeoPixel WS128, esta tira tiene como característica que no es necesario tener que alambrar cada uno de los LEDs, si no sólo usando un *bus*, es capaz de transmitir el patrón con el que se quiera iluminarla, además si se tienen dos tiras, como en este caso, conectando en serie el *bus* de datos se pueden controlar ambas, del mismo modo se puede controlar el color con el que se quieran prender las luces, esta característica podría ser muy útil, en especial pensando en poder transmitir a través de colores diversas instrucciones, como iluminar de color verde si se realizó bien el ejercicio o color rojo en el caso contrario. Finalmente, se le añadió un *buzzer* de 5V, con la finalidad de ser una alternativa cuando por la luz exterior no se alcance a observar las tiras LED.

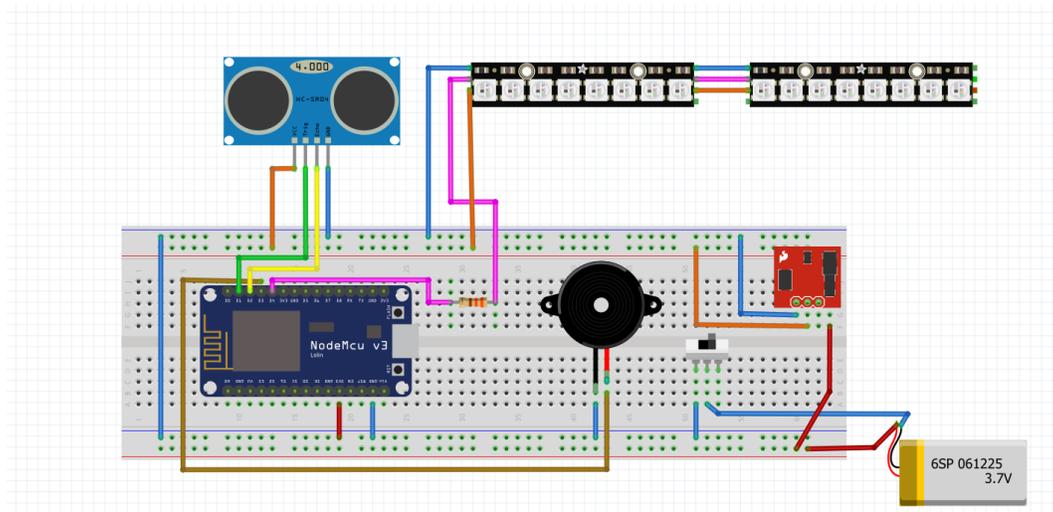


Fig. 61 Diagrama sobre una protoboard de una portería.

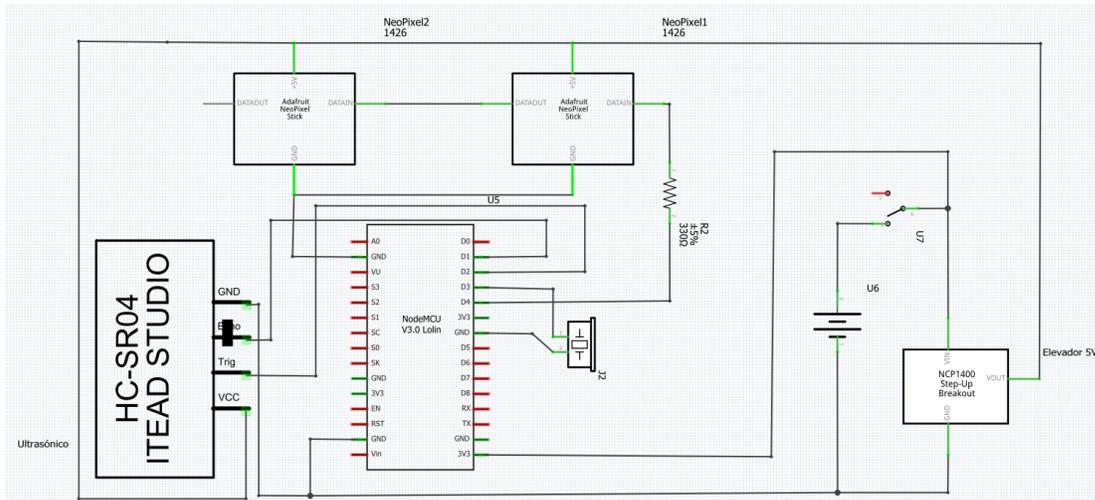


Fig. 62 Diagrama esquemático de una portería.

5.7. DISEÑO FÍSICO DE LOS ELEMENTOS

Para el *tracker* se decidió usar solamente el circuito tal como estaba en una PCB de soldadura, para hacer modificaciones fácilmente en caso de ser necesario. Pero en cuanto a la portería y los *access point* se decidió usar una estructura para mantenerlos asegurados, en especial a la portería se le hizo la estructura para tener la forma de portería, pues es un elemento con el que los jugadores se encuentran familiarizados y darle otra forma podría causar confusiones.

Para el diseño de la portería se propuso usar tubos y codos de PVC, debido a que es un material ligero y al estar hueco permite poner en su interior toda la circuitería, además que permitiría montarla y desmontarla fácilmente, lo cual era una característica buscada en los objetivos de diseño, para añadir estabilidad se colocó arena en el interior de los tubos de la base, de este modo no se moverían por el viento, se usó igualmente una “T” para fijar el sensor ultrasónico y pueda estar paralelo al tubo enfrente de el, al igual, se perforaron agujeros sobre el travesaño (el tubo largo frontal), para fijar las tiras LED. Finalmente, se propuso la altura de estas porterías como 40 cm, lo cual equivale a 1.8 veces el diámetro de un balón, de este modo se asegura que al colocar el sensor ultrasónico a la mitad de la altura pueda captar sin problemas el paso o no de un balón, cosa que no se podría asegurar en caso de ser 2 veces mayor al diámetro del balón, mientras que para elegir el ancho de la portería se buscó mantener la proporción áurea, lo que lo haría ver

estéticamente agradable, así que el ancho fue de 65 cm, de este modo recapitulando el material necesario, se usaron 6 codos, una “T” y 2.9m de tubo de PVC de dos pulgadas por cada una de las porterías.

En cuanto a los *access point*, se diseñó una estructura con forma de caja, se modelaron las medidas de la base como 17.5cmx13cm, para una protoboard y algunos otros componentes entraran en ella, en cuanto a la altura se le dio 4.5cm. En la tapa superior se colocaron algunos agujeros rectangulares con el fin de permitir la emisión del sonido por parte del *buzzer*, igualmente se colocó un agujero de 1cm de diámetro para que la antena wifi pueda salir y permanecer perpendicular con el suelo, de esta forma se garantiza que la señal se transmita uniformemente. Como material para el *access point*, se eligió el MDF, ya que es ligero y tiene la ventaja que se puede cortar con tolerancias de milímetros al usar el corte láser.

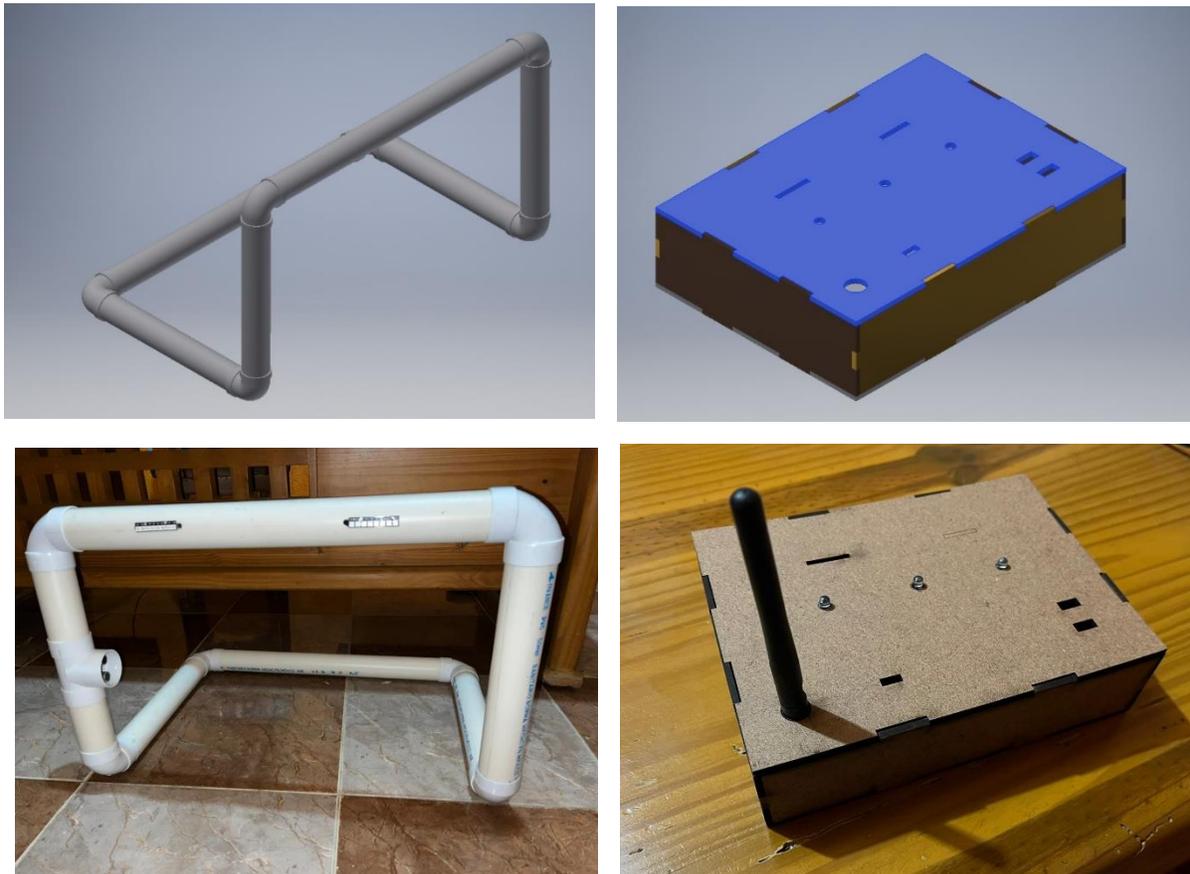


Fig. 63 En la parte superior se encuentra el CAD del boceto de la portería(izquierda) y el CAD del *access point* (derecha); mientras que, en la parte inferior, se encuentra el prototipo real de la portería(izquierda) y el prototipo real del *access point* (derecha)

5.8. PROGRAMACIÓN DE DISPOSITIVOS

En la programación de los dispositivos como se dijo anteriormente se usó la IDE de Arduino. Aunque cada uno de los dispositivos fue programado por separado se empezará con la explicación de las características que comparten en el código estos elementos y éstas son:

- Inicializar el puerto serial a 115200, para monitorear su comportamiento en caso de un error.
- Conectarse a internet y al servidor sobre el puerto configurado para MQTT.
- Declaración de los pines de salida y entrada.
- Declaración de los temas de salida y suscripción a los temas de entrada.
- Inicializar la memoria EEPROM para guardar los datos que sea necesario retomar por el dispositivo en caso de que se apague.
- Función “*callback*” encargada del manejo de los mensajes recibidos.

Justamente en la función “*callback*”, que es llamada desde el *loop* principal y que se encarga de revisar continuamente el estado de los mensajes de llegada, se colocaron las instrucciones para configurar/cambiar los dispositivos. Se procederá a explicar brevemente el funcionamiento de la función “*callback*”, primero se colocó una validación para comprobar que se haya recibido un mensaje, y en caso de que llegara un mensaje hiciera un filtrado por el nombre del tema del mensaje, así se sabría cuál es la acción para realizar por cada uno, las diferentes acciones de temas son las siguientes:

- NOMBRE DSP: el nombre del tema hace referencia al dispositivo, de este modo se asegura que ningún otro dispositivo debería recibir el mensaje, cuando se manda un mensaje sobre este tema es porque se necesita una validación para saber si el dispositivo se encuentra conectado a la red y funcionando, por lo que el dispositivo, después de recibir un mensaje sobre este tema, manda una respuesta al tema “*conf_gen*” con el nombre del dispositivo, de esta forma el cliente que mandó la petición podrá saber que el dispositivo se encuentra realmente en funcionamiento.
- CONF_NOMBRE DSP: el nombre del tema se forma concatenando la cadena de texto “*conf_*” más el nombre del dispositivo, por ejemplo, si el dispositivo se llama “*tracker2*” este tema se llamará “*conf_tracker2*”, de este modo se espera que el nombre de este tema sea único por lo que sólo el dispositivo debería recibir el mensaje. Los mensajes bajo este tema son mandados debido a que el usuario desea cambiar ya sea el nombre del dispositivo, el

tema de entrada y/o el tema de salida, por lo que el cuerpo del mensaje enviado está compuesto de tres elementos separados por comas (ej: Tracker1, H1,H2) así que el primer paso es separar estos elementos y asignarlos en variables, el primer elemento corresponde al nombre del dispositivo, el segundo al nombre del tema de entrada y el último al tema de salida, luego de esto se desuscribe a los temas anteriores y se suscribe a los nuevos, después guarda en la memoria EEPROM del dispositivo las actualizaciones para que queden respaldadas en caso de que se apague éste, finalmente se envía un mensaje al tema "conf_gen" con el nombre del dispositivo como cuerpo para indicar que las configuraciones se hicieron correctamente.

- ENTRADA: este tema puede compartirse con varios dispositivos, de esta forma con una sola señal se podrían activar diferentes dispositivos. En caso de que el mensaje llegue al tema de entrada del dispositivo, éste realizará acciones propias a su tipo, ya sea portería, *tracker* o *access*, más adelante se abordará qué es lo que realiza cada uno de los dispositivos cuando llega un mensaje sobre este tema.

5.8.1. TRACKER

Sus rasgos de programación que lo hacen distinto a los otros dispositivos es que en la parte del "setup" se prende un led solamente hasta que se ha asegurado que se está conectado a la red wifi; del mismo modo se creó una función llamada "obtenCanal" que se encarga de obtener automáticamente el canal sobre el cual se tomarán las lecturas wifi, pues como se vio en la sección 5.1.2 es necesario saber el canal sobre el cual se están transmitiendo las lecturas de wifi para hacer un escaneo más rápido y tener esta acción automatizada ayuda al usuario.

Para el desarrollo del *tracker* se implementó la lectura de potencia que se abordó en la sección 5.3.2 de esta forma cuando llega un mensaje a su tema de entrada con el cuerpo del mensaje igual a "on", el *tracker* prende un led para indicar que comenzó el proceso para enviar mensajes sobre sus lecturas de potencia al tema de salida, también se cambia el valor de una bandera a "true" para indicar al ciclo principal que es necesario tomar lecturas, este proceso sólo se detiene hasta que se recibe un mensaje de "off" sobre su tema de entrada, lo cual hace que se apague el led y hace que la bandera pase a "false" para indicar que deje de tomar lecturas.

5.8.2. ACCESS POINT

En la programación del *access point* lo que lo hace distinto a los otros elementos es el hecho que se configuró para la emisión de señal, esto se logró programándolo como cliente y estación, este proceso se abordó en la sección 5.1.1, de esta forma el *access point* se puede conectar a la red wifi con acceso a internet e independientemente de esto puede emitir señal wifi propia.

Cuando recibe un mensaje a su tema de entrada con el cuerpo del mensaje igual a “on”, se encarga de prender un pin digital, que es el que realiza la acción de saturar un transistor, este solo se apaga hasta que llega un mensaje al mismo tema sólo que con el cuerpo del mensaje igual a “off”.

5.8.3. PORTERÍAS

El circuito de las porterías hace uso de un sensor ultrasónico, por lo cual es necesario configurarlo en la parte del “setup” donde se inicializa usando dos pines (GPIO 5 y GPIO 4), del mismo modo se inicializa un *buzzer* como salida en el GPIO 0, finalmente en el GPIO 2 se inicializa el *bus* de comandos para las tiras LED.

Por otro lado cuando recibe un mensaje sobre su tema de entrada con el cuerpo del mensaje igual a “on” , se prende los pines encargados del *buzzer* y las tiras LED, pero también pone en “true” una bandera que indica que tome medidas del sensor ultrasónico, al igual que en el *access point* estos pines se desactivan cuando llega el mensaje “off” a su tema de entrada, pero también cuando el sensor toma una lectura menor al largo de la portería, lo cual indicaría que un balón la cruzó y por ende desactiva los pines del *buzzer*, se manda la instrucción de apagar las tiras LED y deja de tomar lectura el sensor ultrasónico.

6. RESULTADOS

6.1. SUMARIO

En este punto se tienen los siguientes recursos que van montados sobre el campo y son físicos, los cuales son: *porterías*, *access point* y *tracker*, estos elementos funcionan conectados a una red wifi donde se envían señales a través del protocolo MQTT. A continuación, se detallarán las funciones de cada uno:

- *Portería*: Estos dispositivos tienen la forma de una portería, están compuestos por tubos de PVC que son encargados de darle el sostén adecuado. Contienen además un microcontrolador que es el encargado de encender y apagar LEDs para indicar instrucciones a realizar por el jugador, además de esto también tienen un sensor ultrasónico que permite validar si ha cruzado un balón o no.
- *Access point*: es un elemento conformado por un microcontrolador y por una antena, la cual sirve para emitir señales wifi que son registradas por los *trackers*, al igual que en la portería, poseen LEDs que permiten indicar acciones a realizar por parte del jugador.
- *Tracker*: estos dispositivos contienen un microcontrolador encargado de registrar la distancia aproximada a un *access point* a través de la lectura de la potencia de señal. Están pensados para que el usuario los porte como un accesorio mientras realiza el ejercicio. Contiene un LED color verde para indicar que se encuentra conectado a la red wifi y uno rojo que indica que se está escaneando señal.

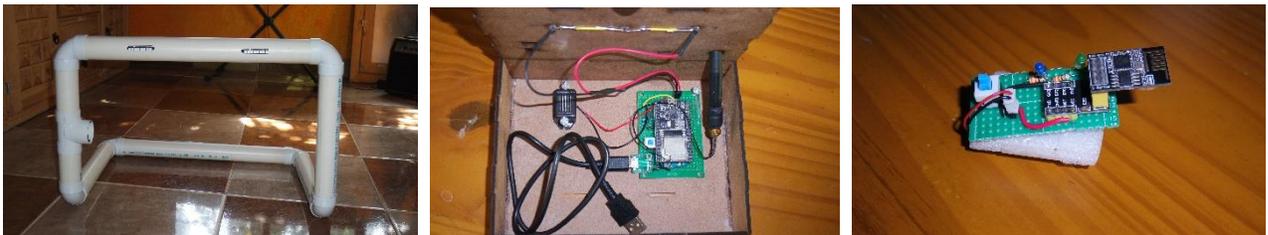


Fig. 64 A la izquierda el modelo de las porterías, al centro el modelo del *access point*, a la derecha dispositivo *tracker*.

De la misma forma se tienen elementos que no son tangibles como la interfaz y la base de datos. La interfaz se encarga de recibir los datos desde los dispositivos ubicados en el campo, esta

interfaz posee un menú el cual permite registrar jugadores, observar su seguimiento y seleccionar el ejercicio a entrenar, entre otras acciones. En cuanto a la base de datos, ésta se encuentra alojada en la nube donde se guardan los registros de cada jugador y toda la información relacionada con la interfaz.

6.2. PRUEBA

Para presentar los resultados obtenidos en el desarrollo de este trabajo se mostrará el accionar de un ejercicio, así se podrán observar la perspectiva del usuario y cómo interactúan todos los elementos antes descritos del sumario.

- Registro de dispositivos: Al ser la primera vez que un usuario ingresa en la interfaz, es necesario que este registre los dispositivos. Para llevar a cabo la tarea, desde la pantalla principal, se elige la opción de “elementos del campo”; luego se selecciona el “signo más”, éste redireccionará a la pantalla “CREAR DISPOSITIVO”, donde se selecciona el nombre del dispositivo, los temas de entrada y salida, así como el tipo de dispositivo que es; se repite este proceso con cada uno de los dispositivos a usar.

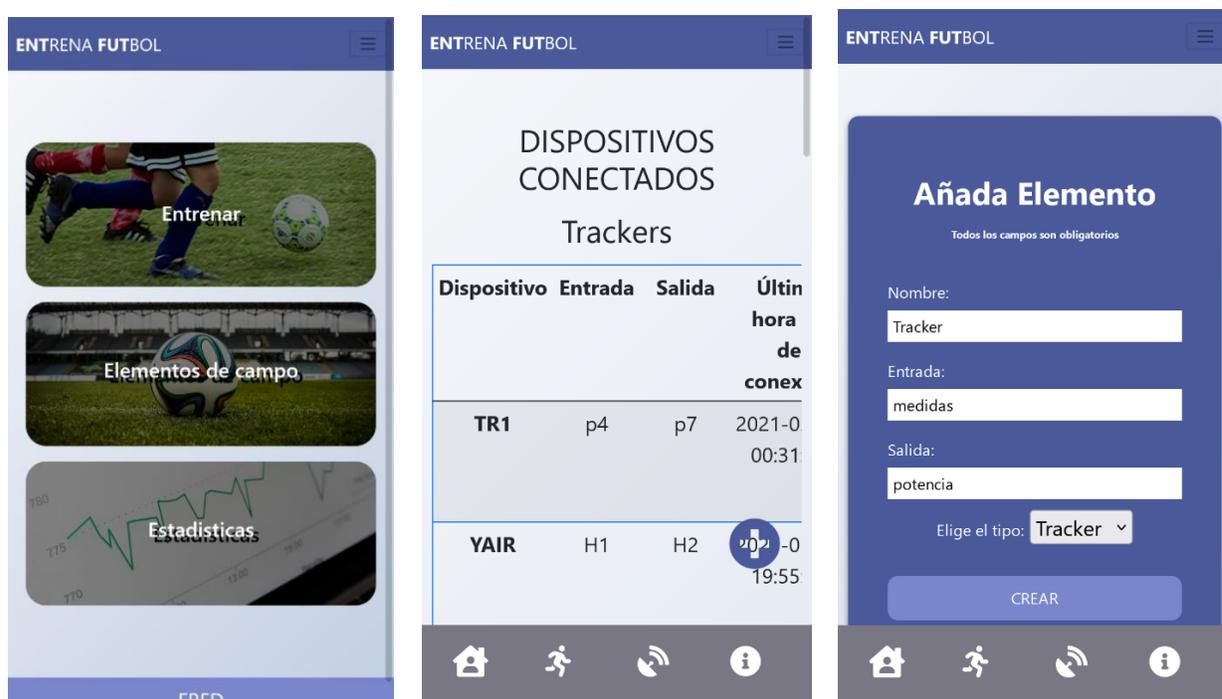


Fig. 65 Pasos a seguir para el registro de un dispositivo

- Encender todos los dispositivos y colocación de estos en puntos clave: Se colocan los dispositivos para el ejercicio que se quiera entrenar. Para comprobar que todos los

dispositivos efectivamente estén conectados a la red y listos para usarse, se puede verificar desde la interfaz, en la pantalla “DISPOSITIVOS”, simplemente haciendo clic sobre el ícono de “señal wifi” del dispositivo a comprobar.

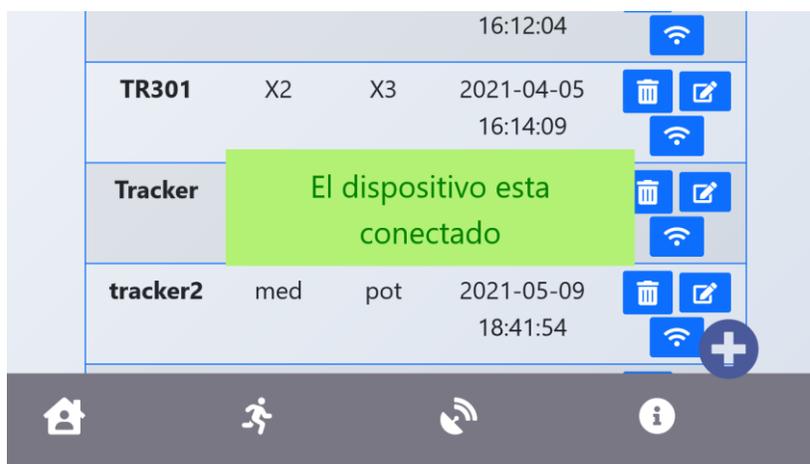


Fig. 66 Revisión de conexión con el dispositivo (pantalla en horizontal)

- Creación del jugador (Fig.67): Para este paso solo será necesario acceder a la pantalla “ESTADÍSTICAS” donde se seleccionará el “signo más” con el fin de agregar un nuevo jugador; luego, redireccionará a la pantalla “CREAR JUGADOR” donde se ingresará el nombre del jugador y el *tracker* con el cual se quiere asociar.
- Selección del ejercicio (Fig.67): En este paso, se puede acceder usando la barra lateral en caso de que se encuentre en diferentes pantallas o desde el menú principal se selecciona la opción ejercicios, este redireccionará a la pantalla “EJERCICIOS”. En esta prueba se seleccionará el ejercicio cuatro, ya que permite ver un mayor uso de elementos del campo.

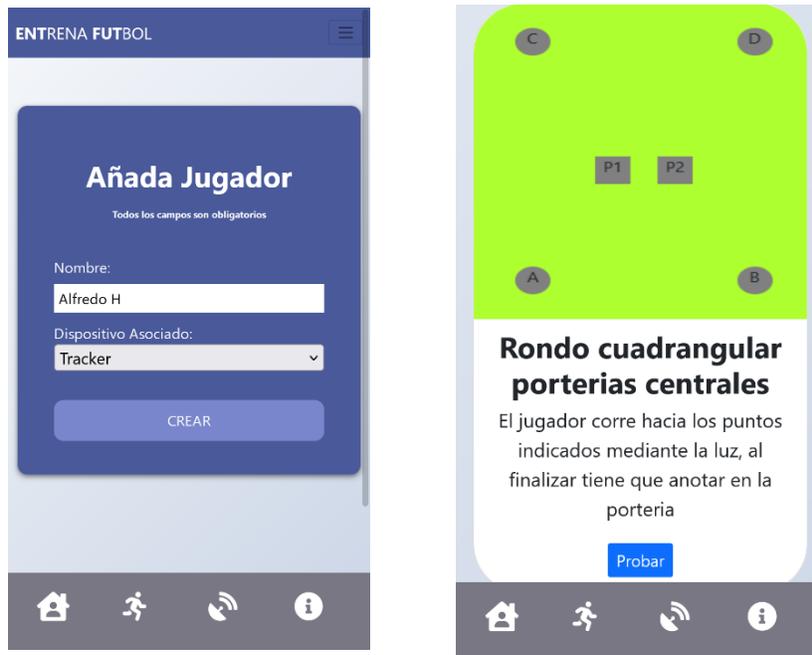


Fig. 67 Pantallas de la interfaz de lado izquierdo crear jugador y del lado derecho selección del ejercicio

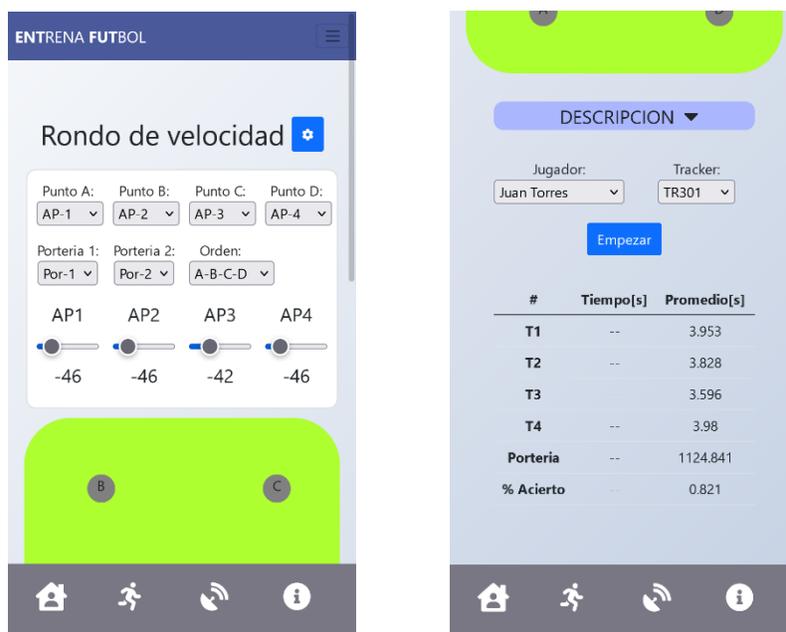


Fig. 68 Pasos para la configuración de los elementos del campo (izquierda) y selección del jugador (derecha)

- Configurar los elementos del campo y selección de jugador (Fig.68): En cada una de las pantallas de los ejercicios se tiene un ícono con un engrane, este permite cambiar o

configurar cuáles son los dispositivos para llevar a cabo el ejercicio; se selecciona cada uno de acuerdo con el dispositivo que está en el campo. El siguiente paso es seleccionar el jugador, lo cual se encuentra debajo de la descripción del ejercicio, al dar clic sobre su nombre se seleccionará automáticamente su *tracker* asociado, aunque se puede cambiar el *tracker* si así se desea.

- Inicio del ejercicio: Con todas las configuraciones realizadas, se puede dar clic sobre el botón “Iniciar”, con esto comenzarán a prenderse los *access point* para decirle al jugador a qué punto moverse; el objetivo en este ejercicio es que los *access point* se enciendan, y cuando el jugador haya llegado a ese *access point* se encienda el siguiente, hasta llegar al último donde tendrá que anotar un gol en alguna de las dos porterías. Todo esto transcurre mientras en la interfaz el usuario puede observar en tiempo real la parte del ejercicio en la que se encuentra el jugador, así como el tiempo transcurrido entre las diferentes fases.

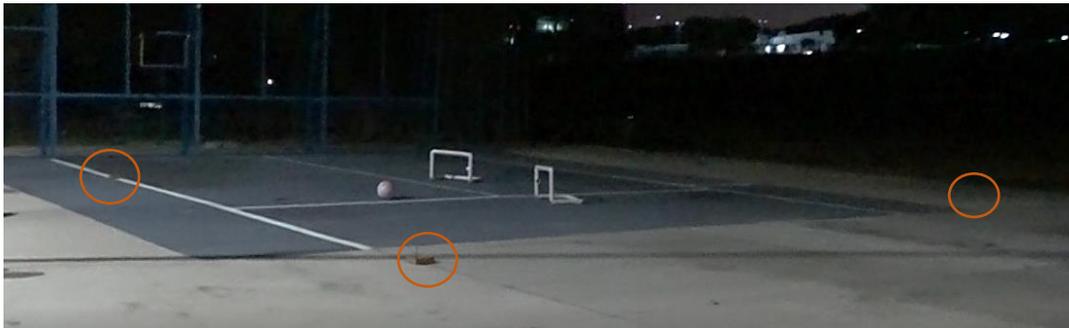


Fig. 69 Imagen de dispositivos colocados para el ejercicio



Fig. 70 Imágenes que muestran la detección en el punto A (izquierda) y el punto de detección B (derecha)



Fig. 71 Imágenes que muestran la detección en el punto C (izquierda) y el punto de detección D (derecha)

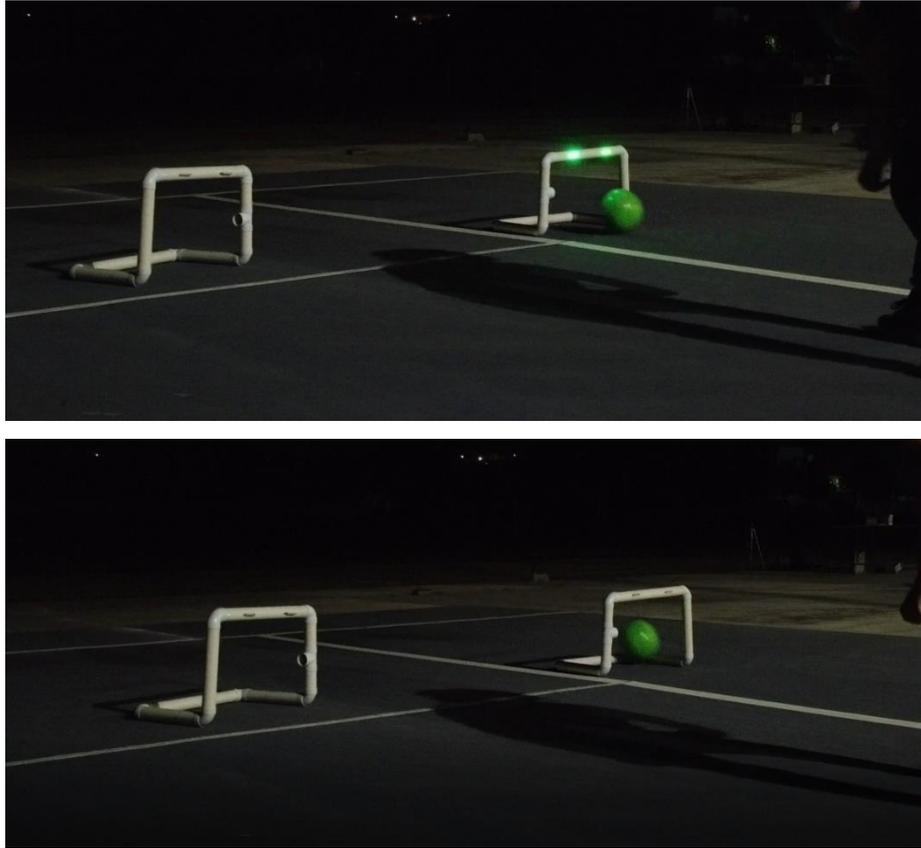


Fig. 72 Imágenes que muestran la detección del balón en las porterías

- Revisión del desempeño del jugador: Finalmente el usuario podrá revisar las estadísticas históricas del jugador en la pantalla “JUGADORES” donde al hacer clic sobre el nombre del jugador se accede a su historial. En el historial se pueden ver diversas gráficas del desempeño del jugador en cada ejercicio; al ser gráficas de líneas, se espera que se pueda tener una rápida clasificación por parte del usuario respecto a si el jugador ha mejorado en el progreso del ejercicio o ha empeorado su desempeño.



Fig. 73 Pantalla de las gráficas acerca del rendimiento del jugador en el ejercicio tres.

6.3. ANÁLISIS DE RESULTADOS

Por la parte de los resultados se puede analizar el comportamiento del sistema físico como el de la interfaz.

Analizando la interfaz se logró que ésta se adaptara a diversos dispositivos desde smartphones hasta computadoras logrando una previsualización adecuada, ajustando los tamaños de los

elementos para este fin. Del mismo modo se logró generar una sencilla navegación por la interfaz haciendo que en la menor cantidad de interacciones se pueda cambiar de página o empezar un ejercicio. Al igual, en la parte dispositivos se puede apreciar una interacción fluida con los elementos físicos para comprobar su estado o hacer modificaciones de una forma simple para un usuario.

Revisando la interfaz y su interacción con los dispositivos físicos al momento de realizar un ejercicio, se logró una coordinación en cuanto a lo que aparecía en pantalla con respecto a los elementos físicos, pues se presentó en 10 muestras del ejercicio 4, un promedio de 100 ms de retraso de los objetos físicos con el de la interfaz, haciendo que se haya alcanzado uno de los objetivos planteados al inicio, el cual se centraba en la interacción en tiempo real; este retraso podría ser considerado tiempo real, pues roza los umbrales de percepción visual, donde debido a la persistencia de la visión, los estímulos visuales pueden alcanzar hasta los 250ms y percibirse como movimiento (entornos con poca luz) [75] por lo que un retraso de 100ms puede considerarse como poco perceptible.

En todo caso si se hace un análisis de los tiempos para estimar el efecto de los retrasos con relación a las mediciones reales, podríamos observar que este es pequeño, para esto se podría aplicar la fórmula de error relativo para poder cuantificar el retraso de éste u *offset* a las mediciones

$$e = \frac{|offset|}{promedio} * 100\% = \frac{0.100}{3.08} = 3.24\% \dots\dots ec.7$$

Al hacer las operaciones del error, se ve que es mínimo, pues son errores que representarían el 3.24% del valor total medido.

Por otro lado, los tiempos quedan registrados adecuadamente en la base de datos, pues éstos quedan guardados para volver a ser revisados en futuras ocasiones de manera permanente. Estos mismos datos son presentados al usuario a través de gráficas que facilitan su comprensión.

En la parte de los objetos físicos, como se mostró en la sección anterior, la respuesta del sistema en general da la información necesaria al jugador que se encuentra realizando el ejercicio, pues se logra que el usuario conozca a cuál de los dispositivos debe de ir por las indicaciones de la luz y sonido; aunque para lograr este nivel de interacción con el usuario fue necesario una calibración constante, es por este motivo que se añadieron en la interfaz *sliders* que permitían una calibración dinámica y que se fueron ajustando a prueba y error de acuerdo al jugador y al terreno; la hipótesis

que se tiene acerca de la causa de las variaciones y los umbrales van muy relacionadas con las pruebas hechas en la sección 5.1.2 y 5.3.2, pues la detección por la potencia de la señal, se vio que esta podría ser irregular y en ocasiones no comportarse de acuerdo a lo esperado, del mismo modo se sospecha acerca de un retraso en la recepción y envío de datos, prueba de ello son los retrasos o adelantos de hasta 0.5 segundos presentados en la sección 5.3.2, con lo cual posiblemente no sea una falla completamente en el sistema de detección de posición, sino más bien un retraso en la red que no permite que el sistema responda adecuadamente. Finalmente, las porterías, al igual que la interfaz, cumplieron su rol funcional, pues se comportaron de acuerdo con lo planeado detectando el balón en los momentos en el que éste pasaba por ellas y enviando la información correspondiente.

Tabla 11 Revisión de necesidades planteadas

No. Necesidad	Necesidad	Porcentaje de cumplimiento
1	Despliegue de información en tiempo real	96%
2	Monitoreo de la distancia recorrida por los jugadores	100%
4	Montaje rápido y transportable	85%
6	Capacidad de almacenar datos	100%
7	Identificación de datos relevantes	100%
8	Ser lo menos invasivo posible con el jugador	80%
9	Interfaz Amigable	100%
10	Transmisión de datos inalámbrica	100%
11	Duración de batería	100%
12	Alcance de transmisión	100%

Para cerrar este capítulo se hará una revisión de cada una de las necesidades planteadas en el capítulo tres, las cuales se ven en la tabla 11 donde se aprecian un identificador, la necesidad y un porcentaje con el cual se cumplió; a continuación, se expondrá el porqué de ese porcentaje:

1. Despliegue de información en tiempo real: Se decidió otorgarle un porcentaje del 95 por ciento, debido principalmente a los retrasos presentados y que fueron documentados tanto en el capítulo seis como en el cinco, en estos capítulos se observó que los retrasos eran provocados, principalmente, por la velocidad del internet entre el servidor y los diferentes dispositivos, siendo ésta una variable que no se pudo controlar para el desarrollo de este trabajo, pues depende de la infraestructura del proveedor del servicio. Ya que el error en retrasos fue cercano al 4%, se decidió al calificar, darle un 96% de calificación.

2. Monitoreo de la distancia recorrida: A este rubro se le colocó una calificación del 100%, pues el sistema logra tomar las distancias preconfiguradas desde la interfaz para guardar la distancia recorrida por el jugador, aunque esta forma de medición es bastante restrictiva pues el jugador puede correr más distancia que la ingresada en la interfaz, pero esto depende de la ruta que éste tome, y es algo en lo que se propone mejorar para el trabajo futuro.

4. Montaje rápido y transportable: Se le colocó un porcentaje del 85%, pues se cumplió casi en su totalidad este objetivo, las dimensiones de los dispositivos hacen que estos entren fácilmente en la cajuela de un auto sedan como se planteó en el capítulo 2; en cuanto al peso, cada uno de los *access point* pesa alrededor de 230gr, el *tracker* 61gr y las porterías 1kg, dándonos el total de casi 3 kg para el peso total del sistema, muy por debajo de los 7kg que se habían planteado en capítulo tres como tope máximo. La razón por la cual no se valoró con el 100% esta necesidad es debido a las porterías, ya que no se encuentran pegadas para facilitar su ensamble, esto hace que en ocasiones se desarme debido al impacto del balón, lo cual es solucionable fácilmente volviendo armar la portería, pero no es deseable, así que tomando en cuenta que existen 7 dispositivos en la realización de un ejercicio, las porterías corresponderían al 29% pero al funcionar parcialmente solo se penalizará este rubro con la mitad 15%.

6. Capacidad de almacenar datos: A esta necesidad se le asignó una calificación del 100% debido a que se observó en cada una de las pruebas que se quedaban guardados de manera correcta los datos del ejercicio, además de que la capacidad de almacenamiento supera los requerimientos planteados en el capítulo tres, por este motivo se sustenta su calificación del 100%.

7. Identificación de datos relevantes: En este rubro se le asignó una calificación del 100%, pues el sistema guarda datos relevantes como el tiempo, velocidad, acierto, y además muestra gráficas, medias y asigna colores para resaltar estos datos; estos indicadores son una ayuda para el usuario para comparar rápidamente el desempeño de los jugadores.

8. Ser lo menos invasivo con el jugador: Esta necesidad fue calificada con un 80%, pues el diseño del *tracker* es bastante portátil que casi permite el libre movimiento del jugador, pero es necesario llevarlo agarrado de la mano, con el puño cerrado, durante el ejercicio, lo cual podría llegar a afectar la libertad de movimiento. En cuanto a dimensiones se puede equiparar a otros dispositivos usados en los brazos como accesorios, tal es el caso de los relojes inteligentes como el apple watch con un peso de 51g y dimensiones de 45x38x11 mm[76], con lo cual haciendo una comparativa con éste, el peso del prototipo es de 61g lo cual lo hace un 20% más pesado, por lo tanto esa diferencia se le descuenta a la calificación, dando el resultado de 80%.

9. Interfaz amigable: En este rubro se colocó una calificación del 100% pues se creó una interfaz que es estéticamente familiar con el usuario, ya que muestra diferentes técnicas de desarrollo utilizadas en otras aplicaciones como Netflix, Spotify, Uber, etc. Haciendo que el usuario comprenda fácilmente como usarla, ya que está familiarizado con su diseño. Todo esto sin sacrificar funcionalidad, pues logra un buen acoplamiento con los elementos físicos.

10. Transmisión de datos inalámbrica: Le fue asignada una calificación de 100% de cumplimiento a esta necesidad, debido a que se logró este fin, pues todos los dispositivos se comunican correctamente de manera inalámbrica. El uso de una red wifi y el protocolo MQTT

dieron una excelente combinación, logrando comunicar todos los dispositivos de forma correcta para llevarlos en sincronía de acuerdo con las necesidades del ejercicio.

11. Duración de batería: Este rubro se cumplió de manera perfecta, pues cada uno de los dispositivos tiene una autonomía que le permite soportar las 4 horas de uso que se habían planteado como requerimiento, siendo que, en algunos dispositivos, como el *access point* y las porterías, sobrepasan este requerimiento; así que a este rubro se le dio una calificación del 100%.

12. Alcance de la transmisión: En cuanto a este rubro se le colocó una calificación del 100%, pues se logró un alcance deseado de 60 metros, permitiendo operar el sistema de forma adecuada, aunque no se comprobó su eficacia a mayor distancia, esta distancia cumple con los requerimientos planteados, así que se sustenta de manera adecuada su calificación.

7. CONCLUSIONES

Como conclusiones, en este trabajo se logró crear un sistema que puede llegar a tomar mediciones sobre ejercicios de jugadores de fútbol; este sistema, aunque con algunos puntos de mejora, es confiable para llegar a tomar mediciones para un análisis de datos, contiene una interfaz que permite observar el desempeño del jugador de una forma sencilla y los elementos físicos son fácilmente enlazables con la interfaz, por lo cual se cumplieron los objetivos de este trabajo.

Como condiciones a considerar para que el sistema funcione de manera adecuada son:

- Control de interferencias: Usar el sistema en un entorno abierto, pues en estos entornos no suelen existir otras señales wifi que interfieran con las de los diferentes *access point*. Del mismo modo como se vio en el desarrollo de todo el trabajo, la caracterización de la señal wifi es un proceso complicado que depende de muchas variables como posición de la antena, interferencias con otros objetos (hasta el propio cuerpo del jugador), por lo que un sistema de posicionamiento usando sólo señales wifi puede tener varios puntos de mejora, siendo por sus características, preferentemente usado como un sistema de proximidad más que para una detección precisa de un objeto.
- Control sobre latencias de la red, pues estas juegan un papel crucial, ya que pueden generar confusiones entre si el sistema está teniendo fallas y es necesario recalibrarlo o sólo se están presentando interferencias en la red.

Por otro lado, el uso del protocolo MQTT parece ser una opción adecuada y fácil de usar cuando se quieren enlazar sistemas que no necesiten gran información para interactuar entre sí, pues presenta bajas latencias y hace que la implementación tanto en computadoras como en microcontroladores sea sencilla de programar. Del mismo modo, se puede mostrar que una interfaz web puede ser funcional, a pesar de no estar instalada sobre la computadora como tal, pues como se vio, se pueden hacer varias acciones distintas y crear un diseño estético sin sacrificar funcionalidad.

Los conocimientos adquiridos y reforzados en el desarrollo de este sistema fueron varios como:

- La creación de interfaces web: para este punto se tuvo que aprender a usar el lenguaje HTML y CSS, además de adquirir conocimiento de JavaScript para la programación de la página web, del mismo modo se vieron varios tutoriales para poder diseñar de una manera estética la interfaz, para llegar a ser lo más familiar y responsiva con el usuario.

- Creación de un servidor y gestión de dominio: Este fue un conocimiento de la misma forma adquirido, pues para alojar la página web y que esta estuviera disponible en cualquier momento en cualquier parte del mundo, se tuvo que crear un servidor; en el desarrollo de este trabajo se hizo sobre Amazon Web Services y se configuró acorde a las necesidades del trabajo.
- Protocolos de comunicación *IoT*: Se adquirió este conocimiento debido a la necesidad de comunicar los dispositivos inalámbricamente, en este trabajo se ganó conocimientos de los protocolos HTTP y MQTT, a través de la lectura de su documentación y tutoriales.
- Control y gestión de base de datos: Este conocimiento fue muy ligado con el de creación de interfaces web, se profundizó sobre todo en las bases de datos SQL y en el lenguaje PHP para gestionarlas desde el servidor, así como el gestor HeidiSQL que permitía realizar cambios remotamente.
- Programación y diseño de los elementos físicos: Este fue un conocimiento reforzado, pues se obtuvieron conocimientos sólidos durante la carrera como la programación de microcontroladores, así como la electrónica necesaria y hasta la parte de usar software CAD para diseñar estos elementos, todos estos conocimientos fueron de gran ayuda al momento de crear todo este sistema.

Finalmente, el desarrollo de este trabajo deja latente que los datos pueden jugar un papel sumamente crucial en el deporte, pues permiten tomar mejores decisiones, a pesar de que en el fútbol apenas se están aplicando análisis de datos, en otros deportes como el baseball ya se ha aplicado varios años atrás, un ejemplo se puede ver en la película el juego de la fortuna (*Moneyball*)[77] del 2011, que está inspirada en hechos reales. Ésta narra la historia de un reclutador de baseball que para lograr obtener un equipo de ensueño hace uso de datos, esto muestra como la estadística y los datos pueden llevar a tomar mejores decisiones en los deportes. En el fútbol, en la última década, estalló una revolución del análisis de datos, bien por el estreno de *Moneyball* o por el alza de la popularidad de técnicas como el *machine learning*, que incentivó a varios clubes a tomar datos de sus jugadores para poder responder a preguntas como: ¿Cuáles son los datos que hacen a los jugadores buenos? ¿Qué métricas hay que obtener? ¿Cómo saber quién será el próximo Messi o Cristiano Ronaldo para rápidamente ficharlo? Y es por esto que clubes grandes como el Barcelona y Real Madrid tienen equipos de *scouting* exclusivamente dedicados a analizar los partidos y el desempeño de los jugadores, pero esto se ha llevado a otro nivel, en el 2019 salió una nota de cómo el Liverpool [78] había contratado a físicos de partículas

para realizar modelos matemáticos y así solo a través de estadísticos pudieran calificar a los jugadores y su rendimiento, sin siquiera tener que ver el partido, por otro lado a inicios del 2021 se dio a conocer que el Manchester City había contratado un grupo de astrofísicos con el mismo fin [79]. La importancia que se le está dando a los datos es tal que las páginas que ofrecen servicios de estadísticos se están popularizando cada vez más, tal es el caso de GolStat una empresa que se dedica a dar estadísticas de los equipos de la LigaMX [80], de la misma forma se tiene proyectado que para el mundial de Qatar 2022, se recolecten más de 15 000 datos por partido [81].

Todo esto refuerza el punto de la importancia que están teniendo los datos y su análisis en la era actual, pero como se ha visto estos análisis solo se están llevando a través de los grandes equipos que tienen un gran presupuesto para realizarlos, el sistema planteado en este trabajo permite disminuir la brecha entre las diferencias de los grandes clubes con los pequeños, pues permite tomar mediciones acerca del rendimiento de los jugadores y guardar estos datos para su futuro análisis a través de una opción más asequible, permitiendo una *democratización* del fútbol, mostrando de igual manera cómo la tecnología y la ingeniería, pueden ayudar a mejorar la eficiencia de la generación de información de los sistemas de múltiples industrias.

8. TRABAJO A FUTURO

El sistema logró cumplir con los objetivos centrales del trabajo, pero como todo proceso de diseño tiene oportunidades de mejora; a continuación, se expresan algunas de estas oportunidades, por ejemplo:

- Elegir una mejor tecnología para la detección de la posición del jugador: como se vio en el desarrollo de este trabajo, la tecnología wifi funcionaba como un estimador de la posición del jugador siendo en ocasiones bastante errática, pero si esta tecnología se cambia por otro método de radiofrecuencia, se podrían mejorar estos resultados para una estimación más precisa y por ende una mejora en las mediciones de las características del jugador.
- Añadir más ejercicios: Para este trabajo sólo se desarrollaron cuatro ejercicios para observar el funcionamiento de este prototipo, pero en realidad se podrían generar más ejercicios que desarrollen otras habilidades en el jugador de fútbol, en especial actividades grupales donde se pueda observar la interacción de varios jugadores con los dispositivos desarrollados.
- Implementar sensores de pulso cardiaco y oxímetros: como se vio en otros desarrollos del capítulo dos, estos sensores podrían dar más información al *coach* del estado en el que se encuentra el jugador al momento de hacer el ejercicio; de este modo, se podría conocer más acerca de cómo el estado físico de un jugador afecta su desempeño.
- Implementar Inteligencia Artificial para analizar los datos: con la información recogida por los sensores, se puede empezar a hacer un análisis exploratorio de los datos para poder definir cuáles son las principales características (como velocidad, inteligencia de pase, capacidad de reacción) que hacen a un jugador mejor que otro y con base en ello poder conocer cuando un jugador es más apto para una determinada posición en el terreno de juego.
- Mejoramiento de los colores: En tanto en la interfaz como en los dispositivos, se podría mejorar la elección de colores para lograr una mejor interacción con el usuario y los jugadores, se podrían hacer variar los colores de los LEDs de los elementos del campo en función de la luz del día para mejorar su visibilidad.
- Probar en un equipo de fútbol con varios integrantes el desempeño del sistema para observar posibles áreas de mejora y cómo responde el sistema ante diferentes sujetos de prueba, pues los resultados de las calibraciones podrían variar por la altura, peso y forma de correr del jugador.

REFERENCIAS

- [1] J.L Olmo Arriaga, “El gran negocio del fútbol” [Online] Available: <https://www.theeconomyjournal.com/texto-diario/mostrar/1525487/gran-negocio-futbol#:~:text=Seg%C3%BAn%20la%20consultora%20Deloitte%2C%20la,de%20todas%20las%20ligas%20europeas.>
- [2] El Plural, “Los deportes con más seguidores en el mundo” [Online] Available: https://www.elplural.com/sociedad/deportes-con-mas-seguidores-en-el-mundo_125954102
- [3] elDiario.es, B. Gutierrez, “Dilma Rouseff intenta huir de 7-1” [Online] Available: https://www.eldiario.es/internacional/mundial-brasil-dilma-rousseff-elecciones-futbol-politica_1_4766726.html
- [4] N. Barriga Bermejo. “Evolución del entrenamiento de fútbol en los últimos 50 años”. Trabajo de fin de grado. Facultad de ciencias de la actividad física y del deporte, Universidad del País Vasco, Álava, España 2015
- [5] Wikipedia, “Hugo Sanchez” [Online] Available: https://es.wikipedia.org/wiki/Hugo_S%C3%A1nchez
- [6] EXCELSIOR, “Siete nuevas tecnologías aplicadas al fútbol ” [Online] Available: <https://www.excelsior.com.mx/adrenalina/siete-nuevas-tecnologias-aplicadas-al-futbol/1327546>
- [7] V.E, Muñoz Fernandez-Arroyo. ”Cuantificación de las demandas físicas y respuestas fisiológicas en el fútbol juvenil durante entrenamientos de competición, y su relacion con el perfil condicional “. Tesis Doctoral. Departamento de actividad física y ciencias del deporte.Universidad de Castilla. Toledo, España, 2015
- [8] Maneiro D. R , “Análisis de las acciones a balón parado en el fútbol de alto rendimiento: saques de esquina y tiros libres indirectos. Un intento de variables explicativas”. Departamento de educación física e deportiva. Universidade da Coruña, Coruña, España, 2014
- [9] Castellano Paulis, “CONOCER EL PASADO DEL FÚTBOL PARA CAMBIAR SU FUTURO”, AM, vol. 2, n.º 1, pp. 39–53, 2009
- [10] Acar, M. F., Yapicioglu, B., Arikan, N., Yalcin, S., Ates, N., & Ergun, M. (2008). Analysis of goals scored in the 2006 World Cup. In *Science and football VI* (pp. 261-268). Routledge.

- [11] A. Sebastian. (5 de mayo 2017) "Football: A deep dive into the tech and data behind the best players in the world" [Online] Available: <https://arstechnica.com/science/2017/05/football-data-tech-best-players-in-the-world/>
- [12] Youtube, Discovery Canada, "Benfica Lab 360 Daily planet". [Online] Available: <https://www.youtube.com/watch?v=8vEBWHMAHGU>
- [13] V. Borja Ramirez and A.C Ramirez Reivich "Innovación de producto". Premio Nacional de tecnología, México, 2006
- [14] M. Aquino "Prevención de lesiones en jugadores juveniles de fútbol profesional cuantificando distancia total con tecnología GPS" Reporte de prácticas profesionales, Nuevo León, México 2019
- [15] NY Times, B. Schoenfeld, "How Data (and some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory". [Online] Available: <https://www.nytimes.com/2019/05/22/magazine/soccer-data-liverpool.html>
- [16] BenditoFutbol.com, "Brasil estrenara sistema de cámaras para evitar goles fantasma" [Online] Available: <https://www.benditofutbol.com/futbol-internacional/mundial-brasil-2014/brasil-tecnologia-mundial-goles-fantasmas.html>
- [17] YouTube, "Amisco a fondo Antena 3" [Online] Available: <https://www.youtube.com/watch?v=qA4SjGjvs8E>
- [18] KLIPDRAW, "Mucho más que una herramienta de dibujo". [Online] Available: <https://www.klipdraw.com/es/que-es.php>
- [19] Remote Presence, "Top-Four Real-Time Location Tracking Systems" [Online] Available: <https://remotepresence.org/top-four-real-time-location-tracking-systems>
- [20] MUNDODEPORTIVO, Pérez Ramos Jesús, "El chaleco negro que usa el Barca ya triunfa en todo el mundo" [Online]: Available: <https://www.mundodeportivo.com/fútbol/fc-barcelona/20181119/453038785272/el-chaleco-negro-que-usa-el-barca-ya-triunfa-en-todo-el-mundo.html>
- [21] SPORTTECHIE, "Wimu Pro Wearable Grows to All Levels of Elite Mexican Soccer" [Online] Available: <https://www.sporttechie.com/wimu-pro-wearable-elite-mexican-soccer-liga-mx>

- [22] A. Bastida, "Análisis de los sistemas de seguimiento de jugadores basados en GNSS y posicionamiento local para su aplicación al fútbol". Tesis doctoral. Departamento de actividad física y deporte. Universidad de Murcia, Murcia, España. 2019
- [23] CATAPULT, "Football" [Online] Available: <https://www.catapultsports.com/sports/football>
- [24] MARCA, G. Sánchez, "Wimu Pro: el chip prodigioso" [Online] Available: <https://www.marca.com/primeraplana/2018/12/15/5c1144d8e5fdea1a1b8b4615.html>
- [25] Dribbleup. [Online] Available: <https://dribbleup.com/products/smart-soccer-ball>
- [26] Xombit, "Justo a tiempo para el mundial llega miCoach, el balón inteligente de Adidas". [Online] Available: <https://xombit.com/2014/05/micoach-adida>
- [27] T. McGowan, "Football enters space age with 'Footbonaut' ". [Online] Available: <https://edition.cnn.com/2012/11/16/sport/football/dortmund-footbonaut-robot-football/index.html>
- [28] CNN en español, "El fútbol entra en la era espacial con el "Footbonaut" [Online] Available: <https://cnnespanol.cnn.com/2012/11/17/el-futbol-entra-en-la-era-espacial-con-el-footbonaut/>
- [29] Caño Football. [Online] Available: <https://www.canofootball.com/articles/look-inside-benficas-academy-developing-winning-mentality-individuals/>
- [30] AmericaTV, "Benfica y su revolucionario sistema de entrenamiento" [Online] Available: <https://www.americatv.com.pe/deportes/futbol-mundial/benfica-y-su-revolucionario-sistema-entrenamiento-n162760>
- [31] Clipset, "Adidas Smart Ball micoach, el balón de fútbol inteligente". [Online] Available: <https://clipset.com/adidas-smart-ball-micoach-el-balon-de-futbol-inteligente/>
- [32] Ramaprasad, A. (1983). "On the definition of feedback". Behavioral Science 28,p 4-13.
- [33] Bastida-Castillo A, Gómez-Carmona C, Hernández-Belmonte A, Pino-Ortega, J, "Validez y fiabilidad de un dispositivo inercial (Wimu pro) para el análisis del posicionamiento en balonmano" Ebm Recide Vol.14, Marzo, 2018
- [34] CATAPULT, "Football" [Online] Available: <https://www.catapultsports.com/products/clearsky-t6>

[35] Diario Motor, "Chevrolet Aveo". [Online] Available: <https://www.diariomotor.com/coche/chevrolet-aveo/>

[36] "Manejo y almacenamiento de materiales-Condicion de seguridad y salud en el trabajo". Norma Oficial Mexicana NOM-006-STPS-2014. Diario Oficial de la Federación, 11 de septiembre de 2014

[37] DBA dixit, "Tipos de datos numéricos en SQL server". [Online] Available: <http://dbadixit.com/tipos-datos-numericos-sql-server/>

[38] F. Fleuret, J. Berclaz, R. Lengagne and P. Fua, "Multicamera People Tracking with a Probabilistic Occupancy Map," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 267-282, Feb. 2008

[39] J. Berclaz, F. Fleuret, E. Turetken and P. Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pp. 1806-1819, Sept. 2011, doi: 10.1109/TPAMI.2011.21.

[40] F. Ramirez, "Filtrado de datos de posicionamiento global (GPS), Tesis de maestría, ingeniería en geociencias, Instituto tecnológico superior de Venustiano Carranza, Cd. Lazaro Cardenas, Puebla, Mexico, Julio 2018"

[41] Wikipedia, "GPS". [Online] Available: <https://es.wikipedia.org/wiki/GPS>

[42] Wikipedia, "Vision Artificial". [Online] Available: https://es.wikipedia.org/wiki/Visi%C3%B3n_artificial

[43] Wikipedia, "Bluetooth". [Online] Available: <https://es.wikipedia.org/wiki/Bluetooth>

[44] Wikipedia, "Wifi". [Online] Available: <https://es.wikipedia.org/wiki/Wifi>

[45] StatCounter, "Mobile Operating System Market Share Worldwide". [Online] Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>

[46] StatCounter, "Desktop Operating System Market Share Worldwide". [Online] Available: <https://gs.statcounter.com/os-market-share/desktop/worldwide>

[47] Wikipedia, "Microsoft Visual Studio". [Online] Available: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio

- [48] Last Minute Engineers, "Insight into ESP32 Sleep Modes & their consumption".[Online] Available: <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>
- [49] Arduino, "Powering MKR Wifi 1010 with batteries". [Online] Available: <https://www.arduino.cc/en/Guide/MKRWiFi1010/powering-with-batteries>
- [50] Wikipedia, "Protocolo de transferencia de Hipertexto". [Online] Available: https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto
- [51] MQTT, "MQTT: The standard of IoT messaging". [Online] Available: <https://mqtt.org/>
- [52] Flespi, "HTTP vs MQTT performance tests".[Online] Available: <https://flespi.com/blog/http-vs-mqtt-performance-tests>
- [53] BBits, "MQTT vs HTTP ¿Qué protocolo es mejor para IoT?". [Online] Available: <https://borrowbits.com/2020/04/mqtt-vs-http-que-protocolo-es-mejor-para-iot/>
- [54] Wikipedia, "Microsoft Access". [Online] Available: https://es.wikipedia.org/wiki/Microsoft_Access
- [55] Maria DB , "Maria DB". [Online] Available: <https://mariadb.org/>
- [56] SYSCOM, "WiFi Diseño de una Red Inalámbrica". [Online] Available: <https://soporte.syscom.mx/es/articles/3006495-wifi-diseno-de-una-red-inalambrica>
- [57] Luis Llamas, "Implementar un filtro media móvil rápido en Arduino". [Online] Available: <https://www.luisllamas.es/arduino-filtro-media-movil/>
- [58] Dorado, J., Ruíz, J. F., de Tiempo, G. D. M., & de Meteorología–IDEAM, C. S. (2013). "Implementación de filtros de kalman como método de ajuste a los modelos de pronóstico (GFS) de temperaturas máximas y mínima para algunas ciudades de Colombia". Group of Time and Climate Modeling. Subdirection of Meteorology. IDEAM.
- [59] AWS, "Amazon EC2". [Online] Available: <https://aws.amazon.com/es/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc>
- [60] VESTA,"Simple&Clever Hosting panel". [Online] Available: <https://vestacp.com/>

[61] Culturacion, "Cómo hacer ping a una dirección IP o página web". [Online] Available: <https://culturacion.com/como-hacer-ping-a-una-direccion-ip-o-pagina-web/>

[62] MQTT, "MQTT The standard for IoT Messaging". [Online] Available: <https://mqtt.org/>

[63] HeidiSQL, "HeidiSQL". [Online] Available: <https://www.heidisql.com/>

[64] Styde, "Como crear y eliminar tablas en MySQL/MariaDB". [Online] Available: <https://styde.net/crear-y-eliminar-tablas-en-mysql-mariadb/> IBM, "Claves primarias y claves foráneas". [Online] Available: <https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys>

[65] IBM, "Claves primarias y claves foráneas". [Online] Available: <https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys>

[66] MarvelApp, "Rapid prototyping, testing and hadoff for modern design teams". [Online] Available: <https://marvelapp.com/>

[67] Certus, "Conoce las principales diferencia entre fronted y backend". [Online] Available: <https://www.certus.edu.pe/blog/diferencias-front-end-back-end/>

[68] Wikipedia, "Web API". [Online] Available: https://es.wikipedia.org/wiki/Web_API

[69] Normalize.css, "Normalize.css". [Online] Available: <https://necolas.github.io/normalize.css/>

[70] Free Images, FreeImages.com/je1196, "Pelota Flotante". [Online] Available: <https://www.freeimages.com/es/photo/floating-soccer-ball-1509397>

[71] Pixabay, "fútbol pelota". [Online] Available: <https://pixabay.com/es/photos/f%c3%batbol-pelota-estadio-campo-488700/>

[72] Unsplash. [Online] Available: <https://unsplash.com/photos/lrRbSND5EUc>

[73] Font Awesome, "Take the hassle out of icons in your website". [Online] Available: <https://fontawesome.com/>

[74] ON Semiconductor, "P2N2222A". [Online] Available: <https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf>

[75] Wikipedia, “Persistencia de la vision”. [Online] Available: https://es.wikipedia.org/wiki/Persistencia_de_la_vision

[76] Apple, “Apple Watch Series 7 Especificaciones técnicas” [Online] available: https://support.apple.com/kb/SP860?locale=es_ES

[77] Wikipedia, “Moneyball”. [Online] Available: <https://es.wikipedia.org/wiki/Moneyball>

[78] El clarin, “Un modelo matemático fue clave para que el Liverpool decidiera la contratación de Jürgen Klopp”. [Online] Available: https://www.clarin.com/deportes/futbol-internacional/modelo-matematico-clave-liverpool-decidiera-contratacion-jurgen-klopp_0_YvL7QVfhe.html

[79] El tiempo, “¿Por qué el Manchester city contrato a un grupo de astrofísicos?”. [Online] Available: <https://www.eltiempo.com/vida/ciencia/manchester-city-por-que-el-equipo-contrato-a-un-grupo-de-cientificos-575446>

[80] GolStats. “GolStats”. [Online] Available: <https://golstats.com/>

[81] Forbes, “FIFA estrena el ‘big data’ del fútbol: analizarán 15000 datos por partido”. [Online] Available: <https://www.forbes.com.mx/mundo-fifa-estrena-sistema-de-analisis-de-partidos-que-ofrece-mas-de-15000-datos>

[82] GitHub, “ESP8266-01 WiFi Module.fzpz”. [Online] Available: https://github.com/ydonnelly/ESP8266_fritzing/blob/master/ESP8266-01%20WiFi%20Module.fzpz

[83] GitHub, “NodeMCUv3 Lolin.fzpz”. [Online] Available: <https://github.com/roman-miniailov/nodemcu-v3-fritzing/blob/master/NodeMCUv3%20Lolin.fzpz>

[84] GitHub, “MH-ET LIVE ESP32 DEVKIT 38 pins.fzpz”. [Online] Available: <https://github.com/Warlib1975/Fritzing-parts/blob/master/MH-ET%20LIVE%20ESP32%20DEVKIT%2038%20pins.fzpz>

[85] GitHub, “Adafruit NeoPixel Stick.fzpz”. [Online] Available: <https://github.com/adafruit/Fritzing-Library/blob/master/parts/Adafruit%20NeoPixel%20Stick.fzpz>

[86] GitHub, “HC-SR04.fzpz”. [Online] Available: <https://github.com/emilio2hd/arduino/blob/master/detector-movimiento-distancia/Fritzing%20My%20Parts/HC-SR04.fzpz>

[87] EMQX, "EMQX: Open-Source, Cloud Native MQTT Broker for IoT". [Online] Available: <https://www.emqx.io/>

[88] Fritzing, "Fritzing". [Online] Available: <https://fritzing.org/>