



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología - ESIT

Máster Universitario en Internet De Las Cosas / INTERNET OF THINGS (IoT)

Desarrollo de un indicador de árbitro con tecnología IoT para transmitir conteos en tiempo real

Trabajo fin de estudio presentado por:	Rivero Pérez, Angie Tatuska
Tipo de trabajo:	Planteamiento de un proyecto de Internet de las Cosas
Director/a:	García García, Oscar
Fecha:	05/02/2022

Resumen

La introducción de la tecnología en el deporte ha permitido mejoras en su gestión, en el entendimiento, en la supervisión y monitorización de cada una de las disciplinas y participantes, permitiendo la fluidez, competitividad y precisión dentro de las diferentes disciplinas deportivas. En esta línea, el proyecto actual propone una nueva solución a nivel de transmisión de datos deportivos en tiempo real, proporcionando tanto a los jueces como espectadores una nueva herramienta que recopila de manera rápida y efectiva los datos para las estadísticas de deportes de la familia de beisbol.

La solución finalmente se basa en el diseño y desarrollo de un dispositivo electrónico inalámbrico, desarrollado sobre una placa de desarrollo NodeMCU y la integración de tecnologías como ThingsSpeak, Arduino, MQTT y Andoid para la transmisión y visualización de los datos, logrando un prototipo funcional y práctico que cumple con el objetivo de transmisión de conteo.

Palabras clave: IoT, Deporte, Béisbol, ThingsSpeak, NodeMCU.

Abstract

The introduction of technology in sports has allowed improvements in the management, understanding, supervision and monitoring of each of the disciplines and participants, making the practice of the different sports disciplines more entertaining, competitive and fair. In this line, the current project proposes a new solution at the level of sports data transmission in real time, providing both judges and spectators a new tool that quickly and effectively collects data for sports statistics of the baseball family.

The solution is finally based on the design and development of a wireless electronic device developed on a NodeMCU development board and the integration of technologies such as ThingsSpeak, Arduino, MQTT and Andoid for the transmission and visualization of data, achieving a functional and practical prototype that meets the objective of transmission of counting.

Keywords: IoT, Sport, Baseball, ThingsSpeak, NodeMCU.

Índice de contenidos

1.	Introducción	10
1.1.1.	Motivación	11
1.2.	Planteamiento del Problema	11
1.3.	Estructura.....	12
2.	Contexto y estado del arte	13
2.1.	Descripción general del contexto del proyecto	13
2.1.1.	IoT en el deporte	13
2.1.1.	IoT aplicado en el beisbol	16
2.1.2.	Actualidad de la Federación Española de Beisbol y Softbol.....	19
2.2.	Proyectos relacionados.....	20
2.3.	Tecnologías relacionadas	23
2.3.1.	Componentes tecnológicos de una solución IoT.....	23
2.3.1.	Software para el Desarrollo de un prototipo	29
2.4.	Conclusiones sobre el estado del arte	30
3.	Descripción general de la contribución del TFE	32
3.1.	Objetivos	32
3.2.	Metodología del trabajo	32
4.	Desarrollo específico de la contribución.....	35
4.1.	Análisis y definición de requisitos técnicos	35
4.2.	Diseño del prototipo a nivel de Hardware y Software	42
4.2.1.	Diseño hardware	42
4.2.1.	Diseño y desarrollo software (Micro-controlador)	44
4.2.1.	Diseño e implementación de plataforma IoT	48
4.2.2.	Diseño y desarrollo software (APP Móvil).....	51

4.3. Desarrollo e integración de la propuesta	55
4.4. Ejecución de pruebas de validación.....	59
4.5. Evolutivos y propuesta de mejoras.....	63
5. Conclusiones y Trabajos futuros	65
Referencias	67
Anexo A.....	72
Anexo B.....	78

Índice de figuras

Figura 1. Aplicación de Internet de las Cosas (IoT) y sus tecnologías adyacentes en la gestión deportiva. Fuente: (Carlos Galera-Zarco, 2015)	14
Figura 2. Arbitro conectado. Fuente: (Bueno, 2017).....	16
Figura 3. Estudio de métricas desde aplicaciones estadísticas y sensores conectados. Fuente: (Nestor, 2017).....	18
Figura 4. Definición de zonas de lanzamientos programadas para lectura de PitchFX. Fuente: (Gonzalez, 2015).....	18
Figura 5. Interfaz de Web de Estadísticas de RFEBS. Fuente: (RFEBS, 2022).....	19
Figura 6. Digital baseball and softball umpire game status indicator. Fuente: (Clark Hills, 2007).....	20
Figura 7. Arquitectura IoT del fútbol basada en el estándar M2M de la ETSI. Fuente: (Ikram, Alshehri, & Hussain, 2015).	21
Figura 8. Esquema de conexión de tecnologías usadas en Real-time Monitoring System in High Jump Sport. Fuente: (Muhammad Faris & Afandi, 2019).	22
Figura 9. Placa de desarrollo NodeMCU. Fuente: (Pariha, 2019).....	25
Figura 10. Placa de Desarrollo WeMos D1 mini. Fuente: (Gascón Bononad, 2019).....	26
Figura 11. Metodología definida para la ejecución del proyecto. Fuente: Elaboración Propia	33
Figura 12. Planificación general de la ejecución del proyecto. Fuente: Elaboración propia. ...	34
Figura 13. Esquema general de la propuesta. Fuente: Elaboración propia.	35
Figura 14. Diseño esquemático Pinout de la placa AZ-Delivery D1 mini. Fuente: (Az-Delivery, 2019).....	38
Figura 15. Conexión de pulsador en configuración Pull-Down. Fuente: (5 Hertz Electrónica, 2022).....	43
Figura 16. Diseño de circuito y conexión de componentes electrónicos. Fuente: Elaboración propia.....	44

Figura 17. Esquema de estructura del programa sobre el microcontrolador. Fuente: Elaboración propia.....	45
Figura 18. Diagrama de flujo de función para iniciar y comprobar conexión de Wi-Fi. Fuente: Elaboración propia.....	47
Figura 19. Diagrama de flujo de funciones lógicas para el conteo de Strike y Ball. Fuente: Elaboración propia.....	47
Figura 20. Diagrama de flujo de funciones lógicas para el conteo de Outs e Inning. Fuente: Elaboración propia.....	48
Figura 21. Configuración de Canales en <i>ThingSpeak</i> . Fuente: Elaboración propia.....	49
Figura 22. Claves de escritura y lectura para canales creados en <i>ThingSpeak</i> . Fuente: Elaboración propia.....	50
Figura 23. <i>Widgets</i> creado para la visualización digital de los canales en <i>ThingSpeak</i> . Fuente: Elaboración propia.....	50
Figura 24. API Requests de campos y canales creados en <i>ThingSpeak</i> . Fuente: Elaboración propia.....	51
Figura 25. Diseño Front-end de SCREEN 1 (pantalla de inicio) de la APP Móvil. Fuente: Elaboración propia.....	53
Figura 26. Desarrollo Back-end de SCREEN 1 (pantalla de inicio) de la APP Móvil. Fuente: Elaboración propia.....	53
Figura 27. Diseño Front-end de SCREEN 2 (pantalla de juego) de la APP Móvil. Fuente: Elaboración propia.....	54
Figura 28. Desarrollo Back-end de SCREEN 2 (pantalla de juego-Bloque1) de la APP Móvil. Fuente: Elaboración propia.	55
Figura 29. Desarrollo Back-end de SCREEN 2 (pantalla de juego-Bloque2) de la APP Móvil. Fuente: Elaboración propia.	55
Figura 30. Componentes para el montaje y construcción del prototipo. Fuente: Elaboración propia.....	56
Figura 31. Montaje en protoboard de prototipo de indicador. Fuente: Elaboración propia. .	56

Figura 32. Integración y validación del prototipo en protoboard. Fuente: Elaboración propia.
57

Figura 33. Instalación de APP móvil desde QR. Fuente: Elaboración propia.57

Figura 34. Primera versión de APP Móvil vista desde un dispositivo Android. Fuente:
 Elaboración propia.....58

Figura 35. Pruebas de comunicación con la plataforma de *ThingSpeak* y puerto serial (1Outs).
 Fuente: Elaboración propia.60

Figura 36. Pruebas de comunicación con la plataforma de *ThingSpeak* y puerto serial (2Outs).
 Fuente: Elaboración propia.60

Figura 37. Prueba de comunicación con *ThingSpeak* con 4 Canales. Fuente: Elaboración
 propia.....61

Figura 38. Ejecución de prueba Final del dispositivo. Fuente: Elaboración propia.....62

Figura 39. Resultado de pruebas de tiempo de publicación y visualización de datos. Fuente:
 Elaboración propia.....62

Figura 40. Extracto de Datos exportados en formato JSON para comprobar la escritura desde
 la APP móvil. Fuente: Elaboración propia.63

Figura 41. Diseño de PCB del circuito y conexión de componentes electrónicos. Fuente:
 Elaboración propia.....78

Figura 42. Imagen de Visor 3D del diseño electrónico. Fuente: Elaboración propia.....78

Índice de tablas

Tabla 1. Comparación de Características de distintas placas de desarrollo. Fuente: (Arduino y solo Arduino, 2017)	26
Tabla 2. Definición Funcional de característica del dispositivo. Fuente: Elaboración propia..	37
Tabla 3. Resumen de datasheet de la placa de AZ-Delivery D1 mini. Fuente (Az-Delivery, 2019).....	37
Tabla 4. Definición de componentes por tipo, modelo y/o valor. Fuente: Elaboración propia.	38
Tabla 5. Definición de funciones y librerías necesarias para el programa del microcontrolador. Fuente: Elaboración propia.....	39
Tabla 6. Condiciones lógicas para funcionalidad básica del indicador. Fuente: Elaboración propia.....	40
Tabla 7. Definición de características Requisitos de la Plataforma ThingSpeak. Fuente: Elaboración propia.....	41
Tabla 8. Definición de pines como entradas y salidas digitales según configuración de fabricante. Fuente: Elaboración propia.....	43
Tabla 9. Definición de componentes y características básicas para el funcionamiento básico de la APP Móvil. Fuente: Elaboración propia.	52

1. Introducción

En las disciplinas deportivas de la familia del béisbol (softball, críquet, kickingball, etc.), la recopilación de datos para las estadísticas de los juegos son de mucho interés para los jugadores, entrenadores, dirigentes y hasta espectadores de estos deportes. Por ello, la precisión y registro de los mismos juegan un papel relevante.

Para el registro de los datos durante el juego, existe una figura encargada llamado anotador, quien describe el juego a través de símbolos y abreviaturas, sin un estándar aprobado, que permite posteriormente la adquisición de datos para la estadística de juego (Baes, 2017).

Además de las anotaciones oficiales que recopila el anotador, el juez o árbitro utiliza una herramienta manual (umpire indicador) para realizar el conteo y clasificación de las decisiones que va tomando a lo largo del juego, y que en los torneos no televisados, sólo es visible en tiempo real por él. El umpire indicador es un dispositivo compuesto por ruedas mecánicas y numeración, que permite llevar las cuentas requeridas en los deportes de la familia de beisbol, este conteo suele estar compuesto por al menos tres variables: outs, balls y strike, y puede llevar el conteo de otras variables cómo: carreras por equipo e inning en juego.

El *umpire indicador*, tiene como finalidad mantener al árbitro del juego actualizado en todo momento sobre el conteo de las variables de interés del juego, pero este conteo no es visible por jugadores y/o espectadores, con la excepción de aquellos eventos o competiciones que disponen de recursos informativos digitales (televisión, tableros electrónicos o pizarras deportivas).

En esta línea, el uso de tecnologías aplicadas a deportes como el béisbol ha tenido una notable evolución en los últimos años, y son aprovechadas por organizaciones deportivas y competiciones de élite, permitiendo mejorar el análisis estadístico, el entretenimiento y otras aplicaciones que permiten potenciar el rendimiento de los atletas. Sin embargo, estas tecnologías requieren de inversiones importantes fuera del alcance de organizaciones y/o eventos de deportivos de menor envergadura o bajo nivel adquisitivo (Rodríguez, 2021).

1.1.1. Motivación

En la práctica de disciplinas deportivas se observan cada vez más el uso de tecnologías digitales, permitiendo mejorar la experiencia del espectador, del deportista y del juez o árbitro. Sin embargo, la implementación de estas tecnologías por lo general requiere de una inversión económica importante, lo que imposibilita a eventos de niveles no profesionales a implantar estas herramientas.

En esta línea, el proyecto nace de la necesidad de optimizar los métodos utilizados para el seguimiento y la recopilación de datos durante la práctica de deportes como el béisbol y sóftbol, dado que requieren de un esfuerzo manual y repetitivo por parte del cuerpo arbitral de estas disciplinas para que se desarrolle debidamente según el reglamento que las rige.

Los datos de seguimiento y recopilación en deportes como el béisbol son parte del mismo, y su obtención incluye una serie de procesos que requieren de una constante monitorización de las decisiones de un árbitro por parte de un anotador (persona que realiza una bitácora del juego, a través de una simbología reglada) (Herrera, 2012).

Conocer estos datos en tiempo real, permite al espectador estar informado de la situación de juego sin necesidad de interrumpir la práctica ni entretener al árbitro consultando los mismos. El conteo que suele llevar el árbitro principal, a pesar de ser de mucho interés para todos los involucrados en el evento, solo es visible en tiempo real por él, mediante la ayuda de un dispositivo mecánico portátil que permite contabilizar las decisiones del juego.

En este sentido, la propuesta busca aprovechar el principio utilizado por el dispositivo mecánico (*umpire indicador*), usado por el árbitro, para transmitir al resto de espectadores los datos registrados tras cada decisión del juego, a través de una solución basada en tecnología IoT y presentada a través de una aplicación móvil.

1.2. Planteamiento del Problema

El problema que se pretende solventar con la ejecución del proyecto, se basa principalmente en desarrollar una vía de comunicación directa y óptima para transmitir el conteo de lanzamientos (balls, strike, outs e inning) que suelen llevar los umpires o árbitros a través de un indicador, hacia el anotador oficial del juego y el espectador. De esta manera además de registrar los datos se optimizan los tiempos empleados para realizar esta recopilación y se vuelve transparente ante el público.

La propuesta parte del concepto funcional de un “*Umpire indicator* o *Umpire Assistant*”, para cumplir una doble función, la primera consiste en registrar las distintas cuentas del resultado del juego y, la segunda, transmitir los datos del marcador a una APP Móvil donde los distintos usuarios puedan consultar la información.

1.3. Estructura

Para solventar el problema planteado, la iniciativa requerirá del diseño y desarrollo un dispositivo, que combine por un lado un diseño electrónico de un dispositivo portable, transmitir y registrar los datos del evento deportivo en tiempo real para ser usado con fines estadísticos o sólo informativos.

Para ello, el trabajo se ha estructurado en 6 secciones que describen la propuesta:

- **Introducción:** Esta primera sección permite proporcionar una visión general del contexto motivación y la estructura del trabajo.
- **Contexto y estado del arte:** En esta sección se realiza un estudio del arte de la iniciativa, haciendo una revisión bibliográfica de las tecnologías, proyecto y aplicaciones actuales dentro del contexto del proyecto, permitiendo entender la actualidad de la problemática y la importancia de la solución.
- **Descripción general de la contribución:** En esta sección se describen los objetivos perseguidos por el proyecto y la metodología que se llevará a cabo.
- **Desarrollo específico de la contribución:** En esta sección se ejecutan todas las fases técnicas necesarias para construir y validar la solución planteada.
- **Conclusiones y trabajos futuros de la contribución:** En esta sección se reflexionará sobre el cumplimiento de los objetivos, y se propondrá mejoras detectadas para un futuro, que sirva como canales de estudio e información de interés para la comunidad interesada.
- **Referencias:** En esta última sección se aportan las bibliografías usadas, con el fin dar crédito a las fuentes y a su vez establecer la validez y confiabilidad del trabajo.

2. Contexto y estado del arte

2.1. Descripción general del contexto del proyecto

El paradigma de IoT representa una nueva evolución de Internet, dando un gran salto en su capacidad de recopilar, analizar y distribuir datos convirtiéndola en información, conocimiento, y en última instancia, sabiduría (Evans, 2011). Este paradigma en la actualidad se encuentra en un auge cada vez más expandido que aprovecha las opciones de comunicar datos a través de protocolos y el procesamiento que pueden ofrecer las plataformas cloud.

En este sentido se define IoT (Internet of things/Internet de las cosas) como una combinación de sensores y actuadores que son capaces de proporcionar y recibir información digitaliza y colocarla en redes bidireccionales capaces de transmitir todos los datos para ser utilizados por una gran cantidad de diferentes servicios y usuarios finales (Perera, 2014).

Con los años, el IoT ha pasado de ser un paradigma, a una capacidad fundamental para la creación de un negocio digital (Hung, 2017), y con esta afirmación se puede reflexionar sobre los beneficios que se pueden adquirir mediante la ejecución de soluciones aplicada a cualquier sector, como lo es el mercado deportivo.

2.1.1. IoT en el deporte

El deporte, como ámbito fundamental en la salud y el ocio de la sociedad no es ajeno a la realidad de la evolución de las tecnologías (Carlos Galera-Zarco, 2015). La aplicación de tecnologías IoT en el deporte ha ido creciendo a lo largo de los años, permitiendo que el desarrollo de estos, se realice cada vez más competitivo, aprovechando cualquier detalle que proporcionan los datos digitales y que pueda marcar una gran diferencia (Laura Fava, 2018).

En este sentido el uso de dispositivos inalámbricos, la variedad de datos que pueden obtenerse a partir de ellos y la velocidad a la que se pueden procesarse permiten una nueva etapa donde los deportistas, entrenadores y técnicos, pueden aprovechar estas nuevas tecnologías para mejorar su habilidad, entrenamiento y resultados (Laura Fava, 2018).

En la actualidad, según (Martinez, 2020) se puede se puede concluir, que el uso de tecnologías más utilizadas en el deporte se engloban principalmente en:

- Los dispositivos IoT o wearables.
- Las aplicaciones móviles.
- Machine learning.

Éstas son las tecnologías más usadas en la actualidad, debido principalmente porque el dispositivo móvil y las aplicaciones que corren sobre este, aportan una nueva forma de visualizar el progreso que los usuarios tienen con respecto a sus actividades físicas, esos datos son percibidos en su mayoría por los dispositivos IoT o los sensores del dispositivo mismo (Martinez, 2020).

Estas tecnologías pueden desencadenar una variedad de aplicaciones en el deporte, como se muestra en la Figura 1.

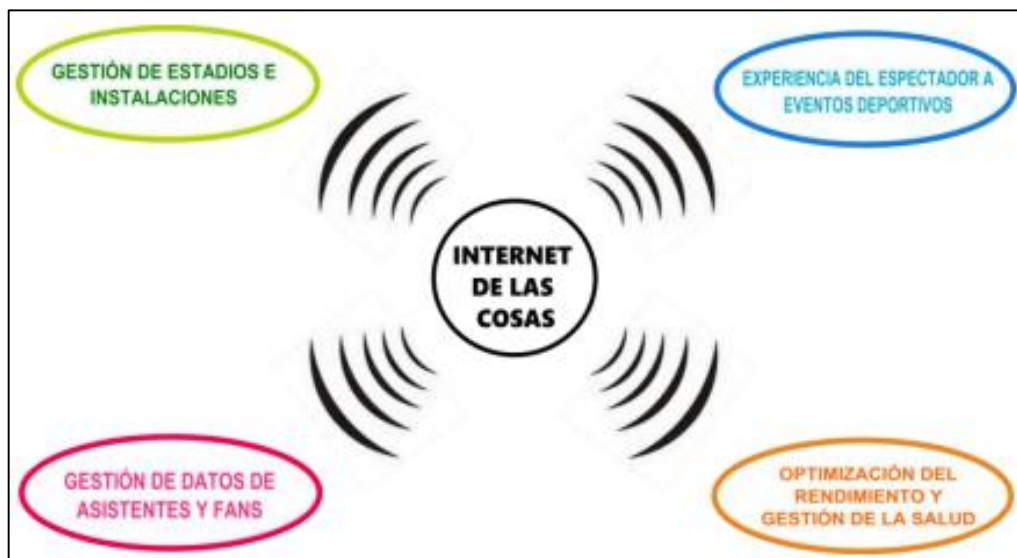


Figura 1. Aplicación de Internet de las Cosas (IoT) y sus tecnologías adyacentes en la gestión deportiva. Fuente: (Carlos Galera-Zarco, 2015)

Para la CSD (Consejo Superior de Deportes) las tecnologías en el deporte también tienen una importante aplicación en el conocimiento de los rivales, en el apoyo en entrenamientos, en el conocimiento de los jugadores del equipo y en el arbitraje y estadísticas (Gallardo, 2007). Este último, y objeto de la presente propuesta ha tenido un importante auge en los últimos años con la aplicación de tecnologías IoT, que permiten la toma de decisiones basadas en la lectura de transductores en diversos juegos de primer nivel y en tiempo real.

Para los espectadores y participantes de las distintas disciplinas deportivas, el arbitraje no siempre desempeña una buena actuación de los distintos deportes, y esto debido

principalmente a la premisa de que el error humano es una realidad innegable y, en muchos casos, causante de enormes injusticias, por lo que los sistemas de IoT aplicados al deporte pretenden agregar al juego un grado más de confianza (Malo, 2021).

Finalmente, bajo en contexto del presente proyecto, se destaca que el uso de IoT para recopilar y analizar datos que generan las estadísticas utilizada en los distintos deportes, siendo esta una de las disciplinas más utilizadas por profesionales de diferentes ámbitos como entrenadores, video analistas, cazatalentos, directores deportivos, médicos, preparadores físicos, nutricionistas o incluso los propios que buscan transformar los mismos en información de interés para espectadores, entrenadores y deportistas, mediante una perspectiva objetiva y cuantitativa (Martínez, 2021).

A nivel de arbitrajes, una gran variedad de deportes han ido evolucionando y mejorando sus métodos de arbitraje, añadiendo tecnología que ayuda a los jueces a la toma de decisiones de manera más precisa. En el fútbol particularmente, se habla del “Árbitro conectado” debido a la indumentaria y accesorios que utiliza que permiten la comunicación externa y la visualización de resultados con distintos mecanismos analógicos, dispositivos de comunicación y tecnología IoT, como el caso del “Goal Control” basado en el “Hawk-Eye” (Bueno, 2017).

Actualmente una de las tecnologías IoT con más aplicación en el mundo deportivo es el Hawk-Eye, cuyo origen se remonta a 2001, cuando el británico Paul Hawkins desarrolló una tecnología para reconocer la posición de las pelotas en el cricket, y hasta la actualidad se utiliza en el tenis, fútbol, beisbol y rugby (Bueno, 2017).

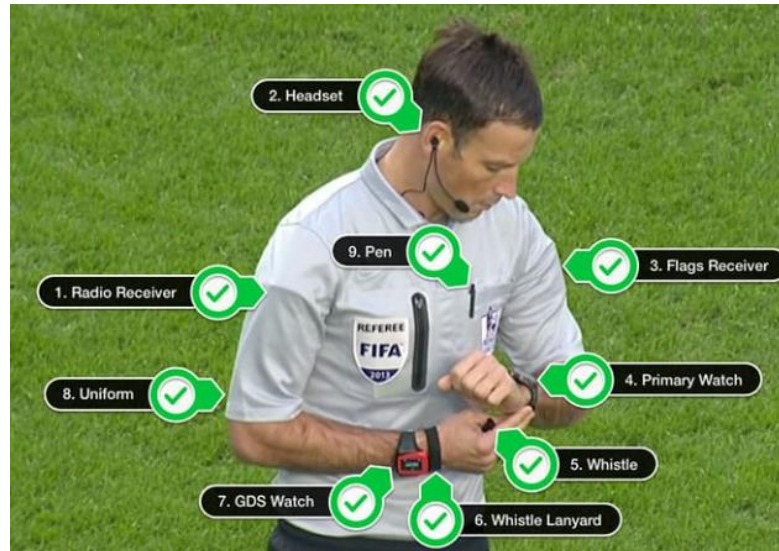


Figura 2. Arbitro conectado. Fuente: (Bueno, 2017)

Entre las tecnologías más trascendentes desarrolladas en los últimos tiempos basadas en IoT en el deporte se resaltan las siguientes:

- Atletismo: "ReacTime", también conocido como sensor de partida falsa, el sistema captura los tiempos de reacción de los deportistas y se utiliza para detectar cuando los atletas comienzan antes de que se les dé la señal. La tecnología está compuesta por sensores instalados en los bloques de partida de cada pista y por un centro de control.
- Natación: Paneles sensibles, estos dispositivos están situados en ambos extremos de la piscina y señalan el orden de llegada de los nadadores. El deportista detiene él mismo el cronómetro mediante un movimiento de presión, lo que produce que se hundan algunos milímetros.
- Tenis: "Hawk-Eye", al igual que en el Rugby, consta de varias cámaras grabando una acción. Se reproduce tridimensionalmente la trayectoria exacta de la pelota sobre la cancha en cada jugada y se pueden realizar repeticiones en tiempo real ante puntos conflictivos (Emol, 2015).

2.1.1. IoT aplicado en el beisbol

En los últimos años, para el beisbol se han desarrollado tecnologías con sensores que miden de manera sencilla la velocidad inicial de una pelota bateada, la velocidad del bate y otras métricas, facilitando a los jugadores la evaluación de su rendimiento y la comprensión de sus

estados de forma, por lo que es un claro ejemplo de la aplicación de IoT en el beisbol, para beneficiar a los jugadores (Zou, 2019).

Asimismo, el beisbol es una disciplina deportiva que desde sus inicios ha utilizado la estadística como parte del juego. Como deporte colectivo es un deporte sumamente complejo que ha favorecido la extracción de una gran cantidad de datos de diversa índole relacionados con el juego, los cuales hacen del béisbol uno de los deportes más completos en cuanto a estadísticas se refiere (César Soto Valero, 2015).

Hablar de datos y estadísticas en el beisbol es hablar de sabermetría, que ha sido definida como el análisis empírico del juego de béisbol a través del estudio de la evidencia objetiva obtenida, específicamente usando técnicas estadísticas, cuyo fin es medir de manera eficaz las actividades que se suscitan en el terreno de juego (César Soto Valero, 2015).

Por ello, la sabermetría ha cambiado significativamente la concepción del juego que se tenía décadas atrás. En la actualidad, los conceptos y principios saberométricos constituyen referencia obligada para todos aquellos que deseen realizar análisis estadísticos precisos a partir de la gran cantidad de datos que se generan continuamente (César Soto Valero, 2015).

Teniendo esto en cuenta, se puede destacar la importancia de la recopilación de datos en el beisbol y como las tecnologías de IoT permiten recopilar, procesar y visualizar los mismos para el interés y aplicación de los distintos interesados (César Soto Valero, 2015).

En la actualidad del béisbol profesional (competiciones como la MLB (Major League Baseball)), se hace uso cada día más de tecnologías e innovaciones para entrenamiento de jugadores y equipos, como es el caso del Blast Motion, que consiste en un dispositivo conectado, que mediante sensores recoge datos y métricas para el estudio del movimiento de los bateadores (Nestor, 2017).

En este sentido, existen dispositivos interesantes como sensores en los bates para mediciones durante las prácticas, que generan datos importantes y que al combinarlos con los videos, el jugador cuenta con herramientas analíticas para la propuesta de mejoras en la mecánica del movimiento u otros datos para mejorar su rendimiento (Nestor, 2017).



Figura 3. Estudio de métricas desde aplicaciones estadísticas y sensores conectados. Fuente: (Nestor, 2017)

En la misma línea, los equipos ya no se limitan a hacer uso de tecnologías de vídeo para ver las jugadas que ocurren en el campo, ahora se mide una mayor cantidad de métricas a través de sensores instalados en estadios, como es el caso del dispositivo PitchFX, que sigue la física de cada lanzamiento y permite a los equipos el estudio de lanzadores (Gonzalez, 2015).

Para el béisbol existe también la tecnología TrackMan que permite medir otros aspectos físicos del juego, como cuán rápido se golpea una pelota, cuál es su velocidad de giro al golpearla y lo fuerte que esta pueda pegar en el campo (Gonzalez, 2015).

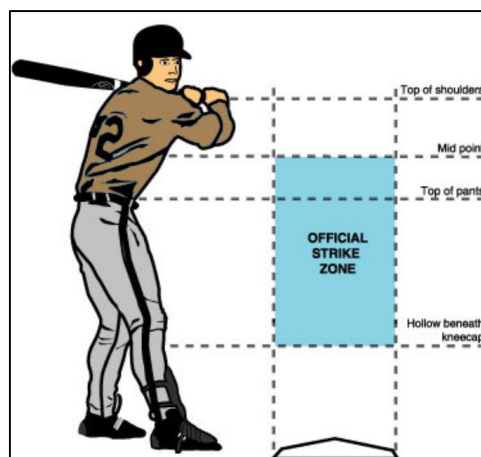


Figura 4. Definición de zonas de lanzamientos programadas para lectura de PitchFX. Fuente: (Gonzalez, 2015).

2.1.2. Actualidad de la Federación Española de Beisbol y Softbol

Es de interés mencionar que este deporte fue introducido en España por el periodista Agustín Peris, concretamente en Cataluña, creándose en 1929 la Federación Catalana de béisbol y en 1944 la Real Federación Española de Béisbol y Softbol (RFEBS) (Pernas, 2021).

Por tanto en España, el beisbol tiene una historia interesante con un crecimiento importante en los últimos años 10 años, donde no sólo se ha extendido a nivel de competición sino a nivel de datos estadístico. Contando en la actualidad con un colegio nacional de anotadores con registro de torneos desde 2008 (Pernas, 2021).

En la actualidad, los recursos digitales y/o tecnológica que cuenta la federación española y por tanto, todos los actores del deporte (árbitros, anotadores, entrenadores y jugadores) son APPs móviles y webs con licencias que permiten introducir la simbología de las anotaciones oficiales, para realizar la respectivas estadísticas de manera semi-automática, pero contando con el procedimiento manual del anotador y sin registrar datos como el conteo de picheos totales y/o por zonas , por lo que se pierde una cantidad de datos que brindan información de interés para la evaluación de los lanzadores (RFEBS, 2022).

BÉISBOL 2021		SÓFBOL 2021	
LIGA NACIONAL DIVISIÓN DE HONOR	11/04 - 15/08	LIGA NACIONAL DIVISIÓN DE HONOR	27/03 - 10/10
LIGA NACIONAL PRIMERA DIVISIÓN	06/03 - 25/07	CAMPEONATO DE ESPAÑA SUB-18	17/06 - 20/06
CAMPEONATO DE ESPAÑA SUB-15 (SELECCIONES AUTONÓMICAS)	20/05 - 23/05	CAMPEONATO DE ESPAÑA PRIMERA DIVISIÓN	30/07 - 01/08
CAMPEONATO DE ESPAÑA SUB-13 (SELECCIONES AUTONÓMICAS)	21/05 - 23/05	CAMPEONATO DE ESPAÑA SUB-15	23/09 - 26/09
CAMPEONATO DE ESPAÑA SUB-18	17/06 - 20/06	COPA S.M. LA REINA	16/10 - 17/10
SERIE NACIONAL SUB-18	17/06 - 20/06	CAMPEONATO DE ESPAÑA SUB-13	21/10 - 24/10
COPA DE SU MAJESTAD EL REY	24/07 - 25/07	CAMPEONATO DE ESPAÑA SUB-13 (SELECCIONES AUTONÓMICAS)	05/11 - 07/11
CAMPEONATO DE ESPAÑA SUB-13	16/09 - 19/09	CAMPEONATO DE ESPAÑA SUB-18 (SELECCIONES AUTONÓMICAS)	05/11 - 07/11
CAMPEONATO DE ESPAÑA SUB-15	23/09 - 26/09	CAMPEONATO DE ESPAÑA MODIFICADO MIXTO	12/11 - 14/11
SERIE NACIONAL SUB-15	23/09 - 26/09		
LIGA NACIONAL PRIMERA DIVISIÓN FASE FINAL	30/09 - 03/10		
CAMPEONATO DE ESPAÑA SUB-11	09/10 - 12/10	WBSC - EUROPEAN MEN'S SOFTBALL MASTERS CUP	27/10 - 30/10
CAMPEONATO DE ESPAÑA SUB-21	21/10 - 24/10	WBSC - EUROPEAN WOMEN'S SOFTBALL MASTERS CUP	27/10 - 30/10

Figura 5. Interfaz de Web de Estadísticas de RFEBS. Fuente: (RFEBS, 2022).

2.2. Proyectos relacionados

Para contextualizar la propuesta del proyecto, pueden referenciarse proyectos relacionados con el IoT en el deporte o dispositivos electrónico utilizados para el béisbol, como es el caso de la patente estadounidense “Digital baseball and softball umpire game status indicator - US20070030128A1”, se trata de un dispositivo inalámbrico para ayudar a los árbitros a realizar un seguimiento del estado y las reglas del juego de béisbol y softbol con el conteo de entradas e indicadores asociados para bolas, strikes, outs y carreras (Clark Hills, 2007).

En la solución que plantea el autor, se propone una evolución digital en relación con el tradicional indicador mecánico de árbitro, logrando mejoras de visualización e incorporando nuevas funcionalidades como el control de tiempo. A pesar de no incorporar tecnologías IoT, la solución utiliza componentes electrónicos como LED, LCD, pulsadores, batería y unidad de control, que permiten la adecuación de un sistema totalmente analógico a uno digital (Clark Hills, 2007).

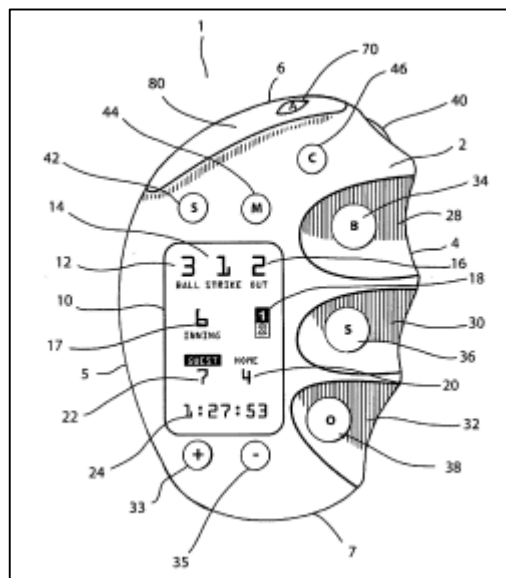


Figura 6. Digital baseball and softball umpire game status indicator. Fuente: (Clark Hills, 2007).

Por su parte, existen proyectos como “Architecture of an IoT-based System for Football Supervision” donde se aplica una arquitectura basada en IoT y aplicada al deporte, y cuyo objetivo ha sido incorporar dispositivos y tecnologías IoT para monitorear la salud de los futbolistas y reducir la ocurrencia de condiciones de salud adversas (Ikram, Alshehri, & Hussain, 2015).

El proyecto “Architecture of an IoT-based System for Football Supervision”, se expone como debe componerse una arquitectura IoT aplicada al deporte basada en el estándar M2M de la ETSI (Instituto Europeo de Estándares de Telecomunicación) y con 3 principales capas (capa de percepción, capa de red, capa de aplicación en la nube) como la mostrada en la Figura 7, siendo ésta la más implementada para la monitorización en diferentes deportes como voleibol y baloncesto (Ikram, Alshehri, & Hussain, 2015).

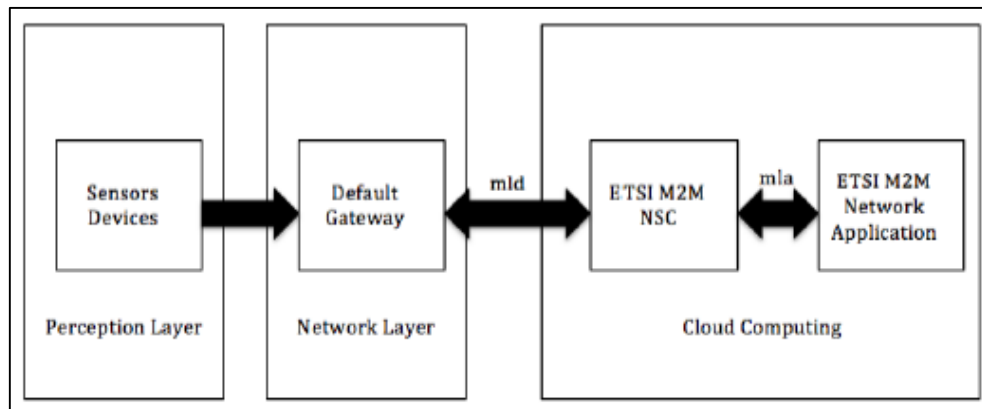


Figura 7. Arquitectura IoT del fútbol basada en el estándar M2M de la ETSI. Fuente: (Ikram, Alshehri, & Hussain, 2015).

Del trabajo de Ikram, Alshehri, y Hussain, se destaca el uso de redes área inalámbrica corporal (WBAN) y la implementación del protocolo CoAP para enviar y recibir paquetes de datos en la capa más baja de la arquitectura que proponen. Dentro de la propuesta de capa media, y actuando como estación base del sistema, el trabajo incorpora la tecnología ZigBee que permite una comunicación inalámbrica de bajo consumo integrable con todo el sistema. (Ikram, Alshehri, & Hussain, 2015).

Otro trabajo de investigación y desarrollo de la temática del proyecto, es el trabajo titulado “Internet of Things (IoT)-based Solution for Real-time Monitoring System in High Jump Sport”, donde se estudia y propone el desarrollo de una solución basada en IoT para un sistema de monitoreo en tiempo real en el deporte de salto de altura (Muhammad Faris & Afandi, 2019).

En el proyecto “Internet of Things (IoT)-based Solution for Real-time Monitoring System in High Jump Sport”, plantea una solución dentro del ámbito deportivo, desplegada a través del software Mosquitto, el uso de protocolo MQTT para el transporte de los mensajes y la plataforma OpenHAB (Muhammad Faris & Afandi, 2019).

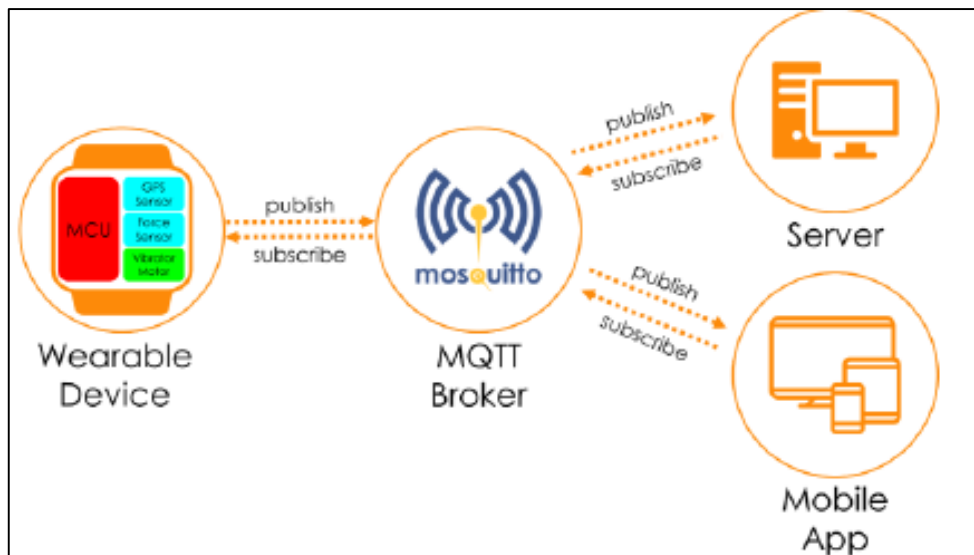


Figura 8. Esquema de conexión de tecnologías usadas en Real-time Monitoring System in High Jump Sport. Fuente: (Muhammad Faris & Afandi, 2019).

Cabe destacar que en el proyecto de Muhammad y Afandi manifiesta una propuesta experimental que permite brindar una mayor comprensión de la metodología de salto de altura de un atleta, permitiendo a la aplicación móvil y dispositivo, jugar un papel importante reduciendo el coste de comprar hardware como cámaras de alta velocidad y entrenadores que guían al atleta de forma remota desde los archivos de registro creados por los propios atletas (Muhammad Faris & Afandi, 2019).

De manera similar, el proyecto “Architecture of an IoT-based System for Cricket Supervision”, propone una solución bajo las tecnologías de IoT para la supervisión o monitorización de los jugadores de cricket con la finalidad de prevenir lesiones leves o mortales (Geethalakshmi, Fathimath , & Sinan Marikattay, 2019)

En esta última propuestas se resaltar el uso de tecnologías como dispositivos de detección (RFID), dispositivos de telecomunicación (Zigbee y WBAN) and Cloud Computing estableciendo con ello según la tipología de aplicación, una solución adaptada a la necesidad de la detección de movimientos irregulares para la prevención de lesiones (Geethalakshmi, Fathimath , & Sinan Marikattay, 2019).

Por otra parte, existen proyectos aplicados a la industria de la supervisión en el deporte aplicando tecnologías IoT, que pueden determinar la precisión de un movimiento, como lo conseguido por el proyecto “Determinación de fuerzas en la “pirueta derecha atrás exterior” obtenidas mediante IoT, como asistencia en tiempo real para arbitraje y futura mejora

ergonómica en plantilla del patinador profesional de patinaje artístico” desarrollado con tecnología IoT (Ramírez Galarza, 2022).

Asimismo, en la propuesta de supervisión en patinaje, la autora diseña un dispositivo de bajo costo para la obtención de datos de sensores de presión e IMU, capaz de transmitir mediante la aplicación de tecnologías IoT dichos valores al software IDE de Arduino para ser procesados y transformados en valores de presión y aceleraciones fáciles de interpretar al usuario fina, con lo que se logra determinar el importante alcance de estos dispositivos (Ramírez Galarza, 2022).

Las distintas propuestas y proyectos descritos permiten reflexionar sobre el uso de las tecnologías y protocolos usados en sistemas IoT dentro del deporte, asentando algunas bases para los requerimientos del presente proyecto, como la composición típica de arquitectura IoT, la importancia del uso de protocolos de bajo consumo y las herramientas de software libres actuales para el desarrollo de prototipos experimentales.

2.3. Tecnologías relacionadas

Con el fin de estudiar y comprender las tecnologías relacionadas con el proyecto, se han analizado y comparado herramientas que permitirán sentar las bases para la selección de los componentes necesarios para el diseño y desarrollo de la solución.

2.3.1. Componentes tecnológicos de una solución IoT

En este sentido, los componentes necesarios para crear soluciones IoT se pueden agrupar en 3 categorías: hardware, middleware y presentación, que se basan en varias tecnologías intercomunicadas que incluye dispositivos conectados, protocolos de comunicación, plataformas de nube, análisis de datos y aplicaciones para presentación e interacción de estos (Peris, 2021).

2.3.1.1. *Hardware*

Una solución IoT, tal y como lo indica (Peris, 2021), contiene en primer lugar el componente hardware, que se compone en primer lugar por una serie transductores, los cuales se definen como un dispositivo que convierte un tipo de energía en otra, y estos se pueden clasificar en dos tipos: los sensores y los actuadores (HBK company, 2021).

En esta línea, los transductores pueden ser sensores, los cuales son los encargados de realizar el contacto físico con el exterior en forma de energía, como pueden ser luz o fuerza, y las convierten en una salida de información digital.

En sentido inverso un transductor también puede ser un actuador, que también reciben una entrada y generan una salida, pero funcionan en sentido opuesto a un sensor. Un actuador por tanto es un transductor que recibe información y produce una salida, consistente en alguna forma de energía física (HBK company, 2021).

Continuando dentro del componente hardware de una solución IoT, se pueden encontrar los microcontroladores, definidos como una unidad de procesamiento de los sistemas (MCU), que permiten de manera sencilla probar conceptos o desarrollar soluciones de manera compacta, gracias a los distintos módulos funcionales y su bajo consumo energético (Pariha, 2019).

Tal y como indica (Aguayo, 2004), un microcontrolador es un dispositivo electrónico que tiene un sistema cerrado capaz de llevar a cabo procesos lógicos, estos procesos, funciones o acciones son programados en lenguaje ensamblador o lenguajes más sencillos a través de compiladores que suelen tener una estructura de elementos conformada principalmente por: procesador, memorias, puertas de entradas y salidas y reloj.

Para el proyecto, se han estudiado distintos microcontroladores o placas de desarrollo que permitan realizar el desarrollo funcional que se plantea, y que sea compatible con protocolos de comunicación inalámbricos, bien sea porque vengán incorporado, o porque se pueda integrar algún módulo para cumplir con esta finalidad (Pariha, 2019).

NodeMCU, es un ejemplo de firmware y kit de desarrollo de código abierto que permite programar el microcontrolador Wi-Fi ESP8266. Es una placa que proporciona un conjunto de características como un entorno de desarrollo integrado (IDE) y soporte incorporado para protocolos comunes como HTTP y MQTT, desarrollado por Espressif Systems (Pariha, 2019).

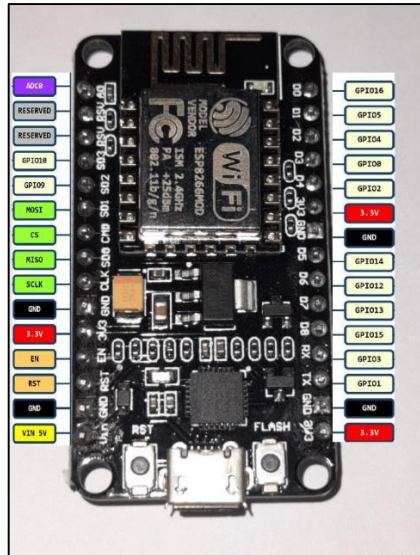


Figura 9. Placa de desarrollo NodeMCU. Fuente: (Pariha, 2019)

Cabe destacar que todas las versiones del NodeMCU se basan en los mismos módulos ESP-12 y ESP-12E, que incorporan el SoC ESP8266 y cuya diferencia principal radica en las funciones y número pines de pines (Gascón Bononad, 2019).

Otra de las placas de desarrollo usadas en prototipos de soluciones IoT, se encuentran las placas de desarrollo Arduino, que son una serie de tarjetas controladoras que utilizan el microcontrolador Atmel AVR. Estas placas vienen en varios tamaños y configuraciones, y se pueden utilizar para controlar una variedad de dispositivos y sensores además de permitir una variedad de protocolos que facilita junto con su respectivo IDE el diseño de prototipos de dispositivos IoT.

Para soluciones compactas, existen placas de desarrollo que proporcionan ventajas interesantes de versatilidad y operatividad como lo es la placa WeMos D1 mini, que es el competidor más directo a la NodeMCU y que presenta características similares a un coste inferior, el número de pines es menor, once frente a los diecisiete de la NodeMCU, aunque para aplicaciones como la que se pretende desarrollar en el presente proyecto, es una opción ideal, por su tamaño, versatilidad y precio (Gascón Bononad, 2019).

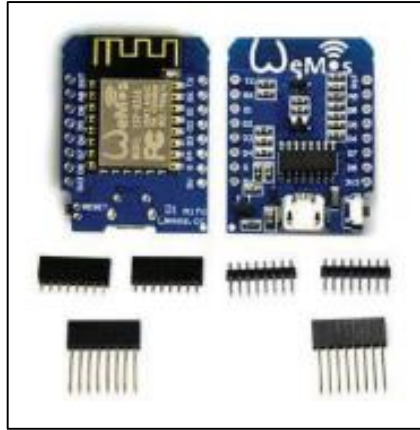


Figura 10. Placa de Desarrollo WeMos D1 mini. Fuente: (Gascón Bononad, 2019).

En resumen, se puede partir de algunas características o consideraciones relevantes para elegir un microcontrolador sobre otro, según el proyecto que se desea ejecutar, tales como la memoria, velocidad, interfaces, consumo y precio (IoT Consulting, 2019).

Por su parte, el microcontrolador como parte del componente hardware brinda el acceso o conectividad a la capa o componentes middleware, a través de comunicación integrada protocolos (Peris, 2021). En consecuencia, las características de conectividad forman parte de las consideraciones para elegir un microcontrolador o placa de desarrollo en cada proyecto.

Por ello, se han evaluado algunas características de los microcontroladores más populares en el mercado para sentar las bases de la selección de una placa de desarrollo, tal y como se muestra en la en la Tabla 1.

Tabla 1. Comparación de Características de distintas placas de desarrollo. Fuente: (Arduino y solo Arduino, 2017)

Placa	ESP32	Esp8266	Arduino UNO
Arquitectura	32 Bit	32 Bit	8 Bit
Frecuencia de C`PU	160 MHz	80 MHz	16 MHz
Wi-Fi	SI	SI	NO
Bluetooth	SI	NO	NO
Flash	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Buses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART

Finalmente, considerando las características de la tabla estudiada y el alcance del proyecto, se considera la placa basada en ESP8266 la más adecuada para el proyecto, debido a que cumple con los requerimientos mínimos para el desarrollo del dispositivo, como su módulo Wi-Fi integrado, GPIO, su bajo consumo, bajo precio y su reducido tamaño la versión propuesta con WeMos D1 mini.

2.3.1.2. Middleware

Como segundo componente principal de una solución IoT se encuentra el Middleware, que actúa como intermediario entre diferentes aplicaciones o sistemas, permitiendo que estos se comuniquen y compartan datos. En el contexto de IoT, el middleware se utiliza para conectar dispositivos IoT con aplicaciones y servicios en la nube proporcionando funciones como la recopilación de datos, la gestión de dispositivos, la seguridad y la escalabilidad (Amazon Web Services, Inc., 2023)

Los middleware también conocidos como plataformas IoT, son un paquete de software local o un servicio en la nube (plataforma IoT como servicio (PaaS) que monitoriza y administra dispositivos, proporcionando generalmente capacidades de operaciones de infraestructura (ciberseguridad.com).

Por su parte las plataformas Open-Source de IoT suelen incorporar modelos de descentralización para recopilar, analizar, visualizar y administrar dispositivos, permitiendo integraciones para la recopilación, procesamiento, conexión y control, sobre los dispositivos IoT (ciberseguridad.com).

En la actualidad, existen diversas plataformas de transmisión, observación y procesamiento de datos obtenidos mediante IoT y entre las más conocidas podemos conseguir Microsoft Azure IoT, AWS IoT e IoT Core de Google Cloud. En la línea de plataformas Open Source, se puede encontrar herramientas como *Node-Red*, *TinyOS*, *OpenHAB* y *ThingSpeak* (Muhammad Faris & Afandi, 2019).

OpenHAB por ejemplo, es conocido como un software de desarrollo de aplicaciones móviles y una multiplataforma basado en IoT de código abierto que utiliza lenguaje C. Utiliza solo una única interfaz de usuario para operar en todas las plataformas como web, Android, iOS y Windows. Los servicios en la nube de *OpenHAB* también ofrecen ventajas para que los desarrolladores utilicen esta plataforma. Brinda servicios gratuitos en la nube con datos

ilimitados que podrían almacenarse en su servidor en la nube (Muhammad Faris & Afandi, 2019).

Por su parte, otra plataforma de interés para el proyecto es *ThingSpeak*, el cual es un servicio de plataforma de análisis IoT que permite ver y analizar datos en vivo, en la nube y con la capacidad de ejecutar códigos MATLAB (Benyezza, Bouheddam, & Djellout, 2018).

Para evaluar la plataforma que más conviene para el desarrollo del proyecto, se han estudiado los principales protocolos aplicados en el IoT que permiten la comunicación de pocos datos sin necesidad ni una excesiva capacidad de transmisión y sencillo de implementar.

Algunos de los protocolos de comunicación que permiten el intercambio de datos entre dispositivos en el sector de IoT, se puede destacar el protocolo CoAP (Constrained Application Protocol) el cual es protocolo de la capa de aplicación que corre sobre UDP especializado en la transferencia de datos a través de la Web y que se destaca por permitir a los dispositivos con recursos limitados comunicarse (Rodríguez González, 2013).

Otro protocolo de interés es el MQTT (Message Queue Telemetry Transport), el cual es un protocolo del tipo publish/subscribe ligero y sencillo constituido sobre la pila TCP/IP, diseñado específicamente para redes y dispositivos de recursos limitados y con bajas tasas de transmisión por lo que se ha convertido en el estándar más utilizado en las comunicaciones IoT (Rodríguez González, 2013).

Por su parte, el protocolo HTTP (Hypertext Transfer Protocol) es el protocolo más difundido en la actualidad y es el empleado para cada transacción de datos que se hagan sobre la Internet, con una capacidad importante de ampliación, que además de utilizarse para transmitir documentos de hipertexto (HTML) se emplea para transmitir imágenes, videos, datos o diferentes tipos de contenido a los servidores y por tanto considerado pieza fundamental para el transporte de los servicios Web en esta nueva arquitectura de la IoT (Rodríguez González, 2013).

Considerando el poco volumen de datos que se desea gestionar en el proyecto, se puede concluir que cualquier plataforma IoT comentada, podría cumplir las funciones necesarias en el middleware, sin embargo, se considera que la plataforma *ThingSpeak* la más apropiada

para el desarrollo del proyecto, por sus ventajas y características en su versión gratuita tales como:

- API abierta.
- Recolección de datos en tiempo real.
- Simplicidad para configurar.
- Librerías compatibles con Arduino IDE.
- Visualización de datos.

2.3.1.3. Presentación

Finalmente, el componente de presentación en una solución IoT se refiere a la capacidad de un dispositivo o sistema IoT para presentar información de forma clara y fácil de entender para los usuarios finales, a las que se puede acceder en diferentes plataformas y que pueden ser diseñadas para diferentes aplicaciones (Peris, 2021).

En este sentido, algunas plataformas IoT del componente Middleware incluyen herramientas capaces de realizar la función de presentación o integrarse con tecnologías externas para implantar implementarla.

ThingSpeak, es una plataforma que proporciona una interfaz web para visualizar y presentar los datos en gráficos y tablas, lo que facilita la interpretación y el análisis. Además, *ThingSpeak* tiene una API abierta que permite a los desarrolladores conectar sus dispositivos y aplicaciones a la plataforma, siendo pro tanto un ejemplo de una herramienta con componentes Middleware y presentación (Soloelectrónicos.com, 2017).

2.3.1. Software para el Desarrollo de un prototipo

De la mano con las tecnologías y componentes necesarios para la creación de una solución IoT, es necesario conocer otras herramientas que permitirán el modelado a nivel de hardware, software y la integración de ambos. Por ello, se estudian algunas herramientas necesarias para crear un prototipo de manera eficiente, sencilla y económica.

Por ello, para el diseño electrónico, se estudia el software KiCad, que constituye un paquete de programas para la automatización del diseño electrónico (EDA) de código abierto, permitiendo con sus distintas herramientas la edición de esquemas edición de circuitos impresos y un visor de ficheros Gerber. También, aloja todas sus librerías en la plataforma

GitHub, donde se incluyen símbolos de esquemáticos, huellas de circuitos integrados, modelos 3D para posterior visualización de la placa e incluso plantillas de ejemplo (Gascón Bononad, 2019).

Para la programación sobre el microcontrolador, se usará Arduino IDE, el cual es un entorno de desarrollo integrado (IDE) con un lenguaje propio basado en C con el que poder generar los programas o sketch de proyectos de la plataforma Arduino, y compatible con otras placas como NodeMCU (Gascón Bononad, 2019).

Finalmente para el desarrollo de la aplicación móvil se hará uso de MIT APP Inventor, el cual es un entorno de programación visual e intuitiva que permite crear aplicaciones completamente funcionales para teléfonos Android, iPhones y tabletas Android/iOS. Su entorno además de amigable, facilita la programación a través de sus bloques estructurados para agilizar la programación (MIT App Inventor, 2013).

2.4. Conclusiones sobre el estado del arte

El IoT en el deporte, tal y como se puede evidenciar con las distintas fuentes consultadas, se encuentra desplegada de distintas formas y aplicaciones, que varían desde la gestión de eventos, optimización de rendimiento, mejora en el entretenimiento, supervisión en el cumplimiento de reglas y otras aplicaciones.

En esta línea, existen en la actualidad una gran variedad de dispositivos conectados que están día a día relacionados no sólo con el deporte profesional, sino en todos los niveles, con aplicaciones relacionadas con la supervisión de parámetros de salud, supervisión de progreso y dispositivos para el entrenamiento controlado.

Teniendo en cuenta que el proyecto está alineado en primer lugar a una solución IoT en el deporte y específicamente en el beisbol, se puede concluir que en las disciplinas relacionadas con el proyecto, las soluciones con tecnología IoT en el beisbol, se encuentran principalmente enfocadas a estudiar y mejorar el rendimiento de los jugadores.

Asimismo, en el beisbol las tecnologías aplicadas a nivel de control, recopilación de datos y supervisión del cumplimiento de reglas del juego se basan principalmente en la utilización de cámaras inteligentes, que por su costo y complejidad suelen aplicarse en eventos profesionales principalmente televisado, como los mundiales, olimpiadas y las grandes ligas estadounidense y japonesa.

Sin embargo, se ha detectado que en la actualidad, la federación española de béisbol a pesar que contar con distintos dispositivos electrónicos y digitales para recopilación de los datos estadísticos de la disciplina, carece de sistemas y/o dispositivos que automaticen la recopilación de datos del juego y/o que puedan transmitir los mismos en tiempo real a los espectadores.

En este sentido, y con el fin de proponer un salto sustancial a nivel nacional, se plantea una solución que permita, extraer de manera automática y en tiempo real, no sólo los datos del conteo de lanzamientos, sino que se pueda transmitir a través del indicador de umpire las decisiones que los jueces del juego toman en tiempo real.

3. Descripción general de la contribución del TFE

3.1. Objetivos

Objetivo general

El presente proyecto tiene como objetivo principal el Diseño y desarrollo de un indicador de árbitro con tecnología IoT para transmitir conteos de un juego de beisbol en tiempo real, buscando ofrecer una alternativa eficaz para la recopilación de datos de interés en los eventos deportivos no televisados.

Objetivos específicos

Para lograr el objetivo general de la propuesta se plantean los siguientes objetivos específicos:

- Analizar y definir los componentes electrónicos para la construcción del prototipo de indicador de árbitro.
- Diseñar y construir el prototipo del indicador, con los transductores adecuados para el entorno y las funcionalidades necesarias para registrar y transmitir los datos de manera inalámbrica.
- Programar la respectiva placa de desarrollo seleccionada, para establecer la lógica necesaria en el prototipo según las reglas del juego, la forma de registro y su comunicación.
- Desplegar una arquitectura IoT para lograr la disponibilidad de los datos en una plataforma en la nube.
- Diseñar y desarrollar una aplicación móvil que permita obtener los datos desde la plataforma IoT y visualizar los datos a través de ella.

3.2. Metodología del trabajo

La metodología que se propone en el presente proyecto es una metodología en cascada con 5 Fases Esta tecnología también conocida como Waterfall o Predictiva, es una metodología del marco tradicional que fue implementada por primera vez en 1970 por Winston W. Royce y, rápidamente, se empezó a adaptar a una gran variedad de industrias debido a su secuencia lógica y facilidad de implementación (Patino, 2020).

Cabe mencionar que entre las ventajas de la metodología en Cascada se encuentra su fácil implementación y administración, debido a que cada fase tiene finalidades específicas y cuenta con un proceso sencillo de revisión. Además, es más fácil realizar la planificación y programación debido a que los requisitos del proyecto se acuerdan en la primera fase, útil para proyectos de menor tamaño, permitiendo una fácil medición del progreso de los hitos (Patino, 2020).

Por ello, se establece para el presente proyecto los hitos bajo esta metodología indicados en la Figura 11.

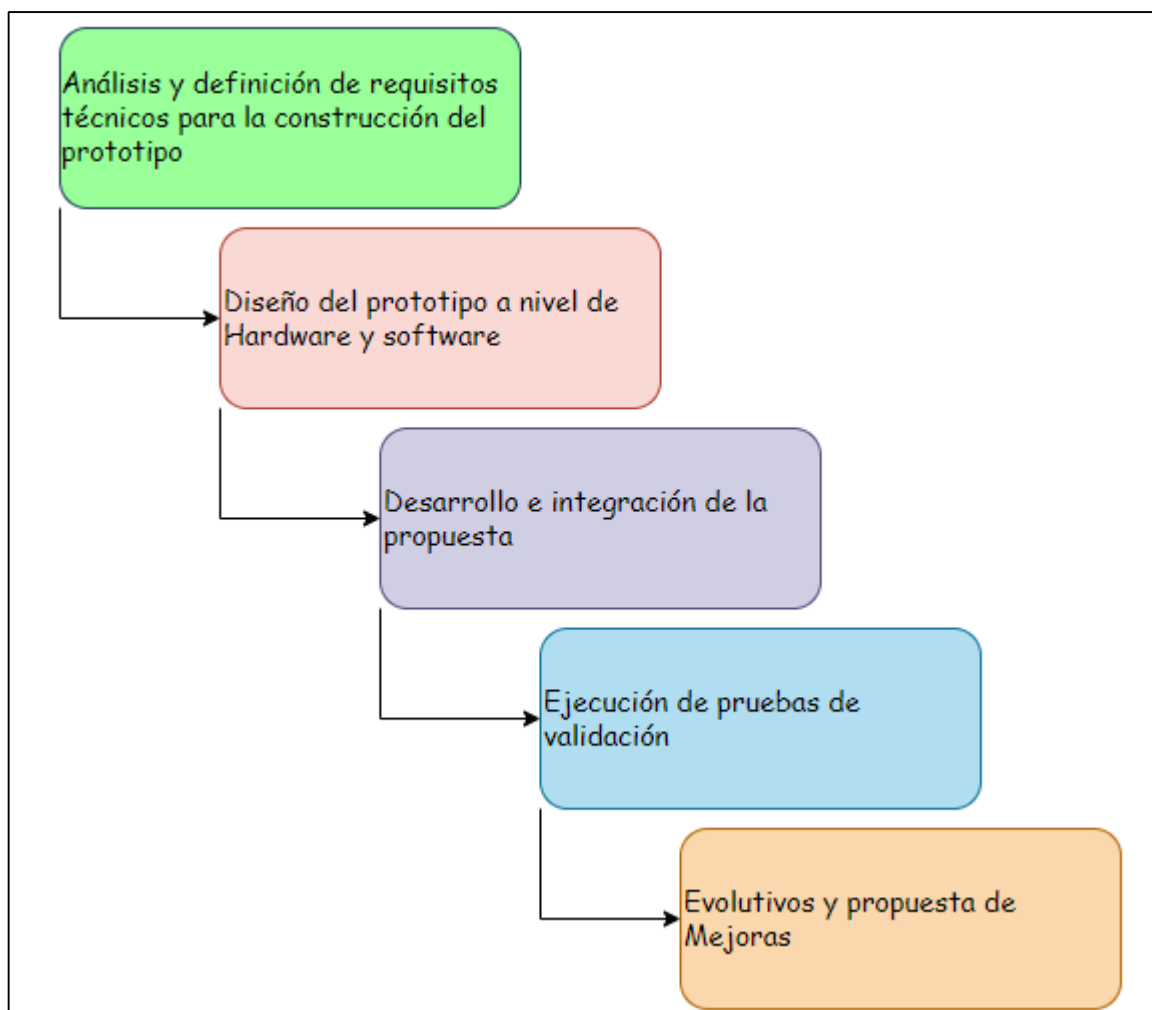


Figura 11. Metodología definida para la ejecución del proyecto. Fuente: Elaboración Propia

Planificación general

Con el fin de cumplir con los distintos objetivos definidos, se ha definido una planificación enmarcada en la Figura 12.

Actividades/Tareas	OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO			
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
Análisis y definición de requisitos técnicos para la construcción del prototipo																
Análisis y definición de requisitos funcionales deseados																
Estudio comparativo de tecnologías existentes en el mercado																
Selección de tecnologías																
Diseño del prototipo a nivel de Hardware y software																
Diseño de prototipo electrónico																
Selección y compra de componentes																
Diseño de lógica del embebido																
Diseño de arquitectura IoT a implementar																
Desarrollo e integración de la propuesta																
Construcción de prototipo electrónico																
Desarrollo de aplicación																
Integración con plataforma IOT																
Ejecución de pruebas de validación																
Ejecución de pruebas simuladas																
Ejecución de pruebas de campo																
Evolutivos y propuesta de Mejoras																
Análisis de mejoras y evolutivos																

Figura 12. Planificación general de la ejecución del proyecto. Fuente: Elaboración propia.

Con esta metodología y planificación se pretende abordar cada una de las fases e hitos establecidos con el fin de cumplir con los objetivos establecidos.

4. Desarrollo específico de la contribución

A continuación, se detallan las distintas tareas definidas en la metodología y planificación del proyecto para el desarrollo del indicador de árbitro con tecnología IoT, ejecutadas en los distintos bloques de tipología de trabajo.

4.1. Análisis y definición de requisitos técnicos

Partiendo de la descripción general del proyecto, y con la finalidad de desarrollar un indicador de árbitro con tecnología IoT para transmitir conteos en tiempo real durante un juego de béisbol o deportes similares, este dispositivo debe no sólo realizar las funciones básicas de un indicador convencional, sino que se diseña para que posea la capacidad de enviar los datos a un bróker, gestionarlos y visualizarlos a través de una aplicación móvil.

Se plantea en primer lugar un esquema general de la arquitectura de la propuesta, que tal y como se indica en la Figura 13, está compuesta por 3 capas, basada en el trabajo de Ikram, Alshehri y Hussain (2015) y Muhammad Faris y Afandi (2019) estudiados en la sección 2.2 del trabajo.

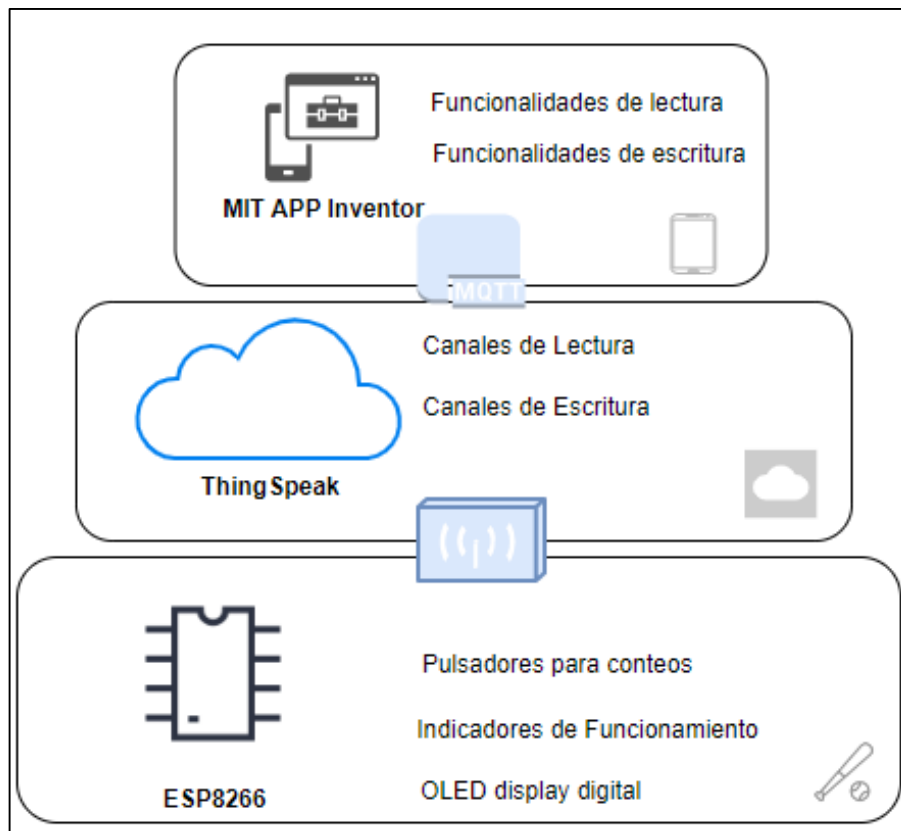


Figura 13. Esquema general de la propuesta. Fuente: Elaboración propia.

El esquema general de la propuesta consta de las siguientes capas:

- **Capa Hardware:** en esta primera capa se encontrarán todos los elementos y componentes físicos que permiten la interacción del usuario. Por tanto, en esta capa se encuentran los transductores y el microcontrolador que cumple la función de procesamiento de borde y pasarela para enviar los datos a la siguiente capa.
- **Capa Middleware:** en la segunda capa del esquema de la propuesta, se encuentra la plataforma cloud *ThingSpeak* que además de cumplir con la función de la obtención de los datos, también actúa como Broker MQTT para comunicarse con la siguiente capa.
- **Capa de presentación:** en la tercera capa, se propone la función de presentación a través de una aplicación móvil. Cabe destacar que *ThingSpeak* puede cumplir con la función de presentación, pero para darle una mayor versatilidad y simplicidad a la visualización de los datos se propone una APP móvil para ello.

Para el desarrollo de los componentes de la primera capa o capa hardware, se han definido los requerimientos funcionales de la solución, que están basados en el funcionamiento de un *umpire indicator* mecánico utilizado en la actualidad por los árbitros del béisbol y deportes similares.

En este sentido, el prototipo debe contener al menos tres pulsadores integrados en una única estructura, que permita contar los strikes, balls y outs, permitiendo mantener un registro preciso de la cuenta de estos indicadores esenciales del juego.

Asimismo se propone agregar un cuarto pulsador que permita numerar el inning en el que se encuentra el juego, permitiendo a los usuarios finales contar con este indicador de juego extra. Estableciendo así un total de 4 pulsadores como entradas en el prototipo.

Se propone también, utilizar leds que permita indicar al usuario el correcto funcionamiento del dispositivo, y una pantalla digital que pueda mostrar en vivo los estados de los indicadores, con el fin de que pueda confirmar el conteo actual de cada una de las variables del juego.

Estos requerimientos, se pueden resumir en la Tabla 2, donde se definen la función y característica de los componentes del dispositivo propuesto.

Tabla 2. Definición Funcional de característica del dispositivo. Fuente: Elaboración propia.

E/S	Descripción funcional	Componente
ENTRADA	Contador de Strike	Pulsador 1
ENTRADA	Contador de Bolas	Pulsador 2
ENTRADA	Contador de Outs	<i>Pulsador 3</i>
ENTRADA	Contador de Inning	Pulsador 4
SALIDA	Pantalla con Datos	<i>OLED 1</i>
SALIDA	Indicador de encendido	Led 1
SALIDA	Indicador de conexión	Led 2

Continuando en la capa de hardware, se estudian las características técnicas de la placa de desarrollo seleccionada para el prototipo que se propone, basada en el estudio inicial realizado en la sección 2.3.1.1 del presente trabajo.

En este sentido, es seleccionada y adquirida la placa de desarrollo D1 Mini basada en el chip ESP8266-12F, que cumplirá la función de unidad de procesamiento y control principal del dispositivo, realizando por tanto la lectura y escritura del dispositivo, y actuando como pasarela para la conectividad utilizando conexión inalámbrica Wi-Fi y la comunicación de datos a través de protocolo de comunicación MQTT con la siguiente capa del esquema del sistema propuesto.

Seguidamente se ha estudiado las conexiones disponibles y las recomendaciones del fabricante para valorar requerimientos técnicos como la alimentación necesaria, las entradas disponibles y protocolos compatibles por el mismo. En este sentido, las características más relevantes de la placa de desarrollo D1 Mini adquiridas mediante el datasheet del fabricante se resumen en la Tabla 3 (Az-Delivery, 2019).

Tabla 3. Resumen de datasheet de la placa de AZ-Delivery D1 mini. Fuente (Az-Delivery, 2019)

Parámetro	Valor
Conector USB	<i>Micro USB</i>
Voltaje de funcionamiento	<i>3,3 V</i>
Memoria flash	<i>4 MB</i>
E/S digitales	<i>11</i>
Entradas analógicas	<i>1</i>
Comunicaciones	<i>Serie, SPI, I2C y 1-Wire a través de bibliotecas de software</i>
Wi-Fi:	<i>802.11b/g/n incorporado</i>
Programación:	<i>Compatible con Arduino IDE y NodeMCU</i>

De manera similar, se ha estudiado el esquema de la placa proporcionado por su fabricante (Az-Delivery, 2019) y mostrado en la Figura 14, que permitirá decidir y seleccionar los pines adecuados durante el diseño de la propuesta que se describirá en la sección 4.2.1 del trabajo.

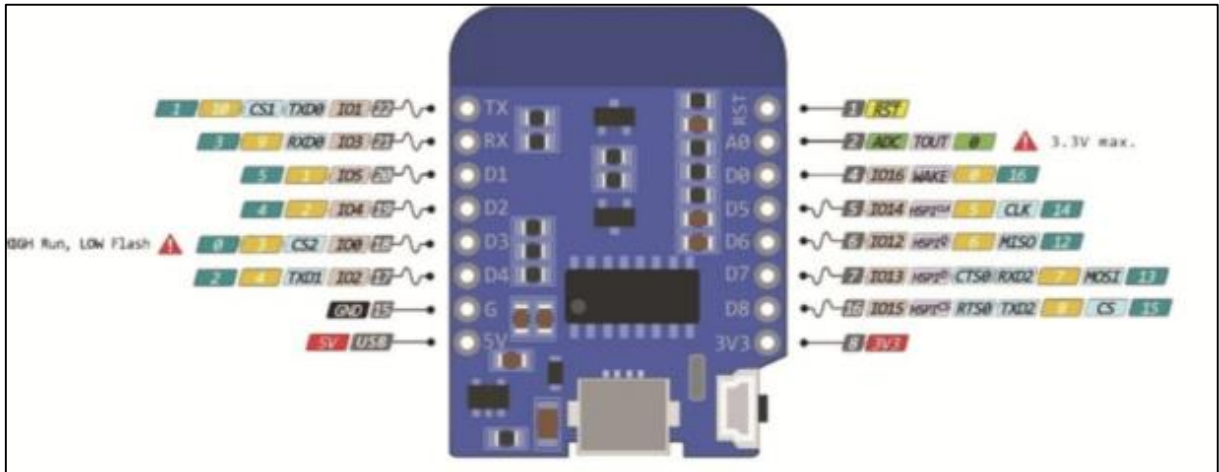


Figura 14. Diseño esquemático Pinout de la placa AZ-Delivery D1 mini. Fuente: (Az-Delivery, 2019)

Otras de las consideraciones a tener en cuenta para el diseño y desarrollo el prototipo, es el entorno y herramientas para la construcción y desarrollo del mismo. En este sentido, como se trata de un prototipo funcional de un sistema IoT se propone realizar el montaje o construcción del prototipo mediante un protoboard o placa de pruebas, siendo este un artefacto para realizar prototipados de circuito electrónicos antes de un ensamble final, permitiendo sin desperdicio de material ni recurso el rediseño y modificaciones sobre un circuito (Mora Daza, 2021).

Teniendo en cuenta lo mencionado, se han seleccionado componentes electrónicos *THT* (*through-hole technology*) compatibles con los requisitos funcionales previamente descritos y compatibles con la tecnología de agujeros pasante necesarios para la construcción sobre protoboard. Estos componentes se describen en la Tabla 4 indicando un total de 5 componentes electrónicos que interactuarán con el microcontrolador del sistema.

Tabla 4. Definición de componentes por tipo, modelo y/o valor. Fuente: Elaboración propia.

Componentes	Tipo/modelo/valor
Pulsadores	Pulsador Membrana
Led	<i>THT</i> /Rojo y verde
OLED	<i>SSD1306/64x48-I2C</i>
Fuente 3,3v/5v	<i>Modulo alimentación para protoboard HW-131/ 3.3 y 5V.</i>
Resistencias	<i>THT</i>

Una vez definido los requerimientos técnicos y funcionales a nivel de hardware, se pretende estudiar los requerimientos técnicos y funcionales necesarios a nivel de software para el funcionamiento del dispositivo propuesto, y las consideraciones necesarias para desarrollar y configurar éste.

En primer lugar, teniendo en cuenta que el microcontrolador finalmente seleccionado para el desarrollo del prototipo del dispositivo se trata de un AZ-Delivery D1 mini, se obtiene de su datasheet instrucciones para el desarrollo y configuración de la placa. En este sentido, el fabricante instruye y recomienda la instalación del controlador CH340 para la lectura de la placa en Windows, y el uso Arduino IDE para la programación del mismo, aprovechando las ventajas de librerías compatibles como esp8266 (Az-Delivery, 2019).

Tal y como lo recomienda en fabricante del microcontrolador, se propone la utilización del entorno de desarrollo Arduino IDE. Considerando esto, se estudian las funciones y librerías existentes que permitirán configurar y desarrollar el algoritmo que ejecutara el microcontrolador para realizar las distintas funciones mencionadas, en la descripción funcional.

En este sentido, además de librerías compatibles con la placa de desarrollo previamente mencionada, existen librerías que facilitan la ejecución de otras funciones como la conexión Wi-Fi, la comunicación con plataforma IoT y el control y actuación con las entradas y salidas propuestos para el dispositivo, algunas de estas librerías proporcionada por la comunidad de arduino se definen en la Tabla 5.

Tabla 5. Definición de funciones y librerías necesarias para el programa del microcontrolador. Fuente: Elaboración propia.

Funciones principales	Librería existente
Leer entradas digitales	<i>esp8266 by ESP8266 Community</i>
Controlar salidas digitales	<i>esp8266 by ESP8266 Community</i>
Escribir en pantalla OLED	<i>SPI.h; Wire.h; Adafruit_GFX.h; Adafruit_SSD1306.h</i>
Conectarse a Wifi	<i>ESP8266WiFi.h</i>
Establecer conexión con Broker	<i>ThingSpeak.h</i>
Lógica funcional	<i>N/A</i>

Además de las funciones proporcionadas por estas librerías, es necesario establecer los requisitos lógico adaptado al funcionamiento deseado de un indicador de árbitro.

Es decir, es necesario determinar algunas funciones y condiciones para modelar la lógica del programa del microcontrolador para seguir las reglas que el deporte dicta. Estas condiciones se basan en la lógica del juego de beisbol y en la responsabilidad del árbitro (Pardo Lozano, 2021):

- Cada bateador tiene hasta 3 strike (lanzamientos en zona buena) para batear.
- Cada bateador tiene hasta 4 balls (lanzamientos en zona mala) para batear.
- Para cambiar de inning o entrada es necesario realizar 3Outs.
- Un partido tiene 9 inning o entradas en total.

Estas condiciones, se definen a nivel de requerimiento de lógica del programa mediante 4 variables definidas en la Tabla 6 que surgen por las reglas del juego mencionadas y de las funciones básicas de un indicador.

Tabla 6. Condiciones lógicas para funcionalidad básica del indicador. Fuente: Elaboración propia.

Variable	Nombre	Condición 1	Condición 2
Conteo de Strike	<i>ContStrike</i>	<i>ContStrike +1 hasta <=3</i>	<i>ContStrike=0 Cuando ContOuts +1</i>
Conteo de bolas	<i>ContBall</i>	<i>ContBall +1 hasta <=4</i>	<i>ContBall=0 Cuando ContOuts +1</i>
Conteo de Outs	<i>ContOuts</i>	<i>ContOuts +1 hasta <=3</i>	<i>ContOuts=0 Cuando ContInning</i>
Conteo de Innings	<i>ContInning</i>	<i>ContInning +1 hasta <=13</i>	

Continuando con el esquema general de la propuesta, para la segunda capa del esquema general de la propuesta o middleware, se ha definido las características y requisitos técnicos para la configuración de la plataforma IoT *ThingSpeak* y necesarios para el intercambio de datos tanto con el microcontrolador como con la capa de presentación.

Por ello, teniendo en cuenta que existirán 4 distintos conteos (strike, balls, outs e inning) serán necesario en un canal, contar con 4 campos de tipo numérico que recoja estos datos. Y se propone adicionalmente 2 campos que permitan publicar texto, para proporcionar al usuario el registro de los equipos en juego.

Por tanto, se definen 6 campos en un canal para la plataforma *ThingSpeak* y de tipo público como se resumen en la Tabla 7, permitiendo así la lectura y escritura de las variables del prototipo, simplificando la publicación de los datos sin claves de acceso a estos canales.

Tabla 7. Definición de características Requisitos de la Plataforma *ThingSpeak*. Fuente: Elaboración propia.

Característica	Valor
Tipo de publicación	Pública
Número de campos en el Canal	6
Tipo de visualización	4 Numeric Display 2 Indiferentes

Finalmente, para la última capa, se han definido los requisitos y características necesarias para el diseño de la aplicación móvil. En este sentido la aplicación Android se desarrolla utilizando el entorno MIT APP Inventor estudiado previamente en el apartado 2.3.1.

El objetivo de crear una aplicación móvil es poder visualizar los datos en tiempo real del indicador, permitiendo la lectura accionada por el árbitro de forma remota. Por tanto, el diseño de la aplicación debe contener las siguientes características funcionales:

- Front-End: En este bloque deben existir al menos 2 interfaces, con un diseño sencillo e intuitivo, que permita por un lado la introducción y selección del juego objeto, y por otro lado debe permitir la visualización automática de los valores del indicador.
- Back-End: La primera interfaz debe contener una función que permita direccionar a la del respectivo juego seleccionado. Asimismo, en el programa debe existir la conexión a las API de lectura y escritura de *ThingSpeak*, y contar con un reloj que permita la actualización de los datos.

4.2. Diseño del prototipo a nivel de Hardware y Software

Una vez estudiado y analizado los distintos requerimientos funcionales y técnicos necesarios para el desarrollo del dispositivo que se propone en el presente trabajo, a continuación se proceden a realizar el respectivo diseño del dispositivo propuesto.

4.2.1. Diseño hardware

El diseño hardware de la solución está basado principalmente en los requerimientos previamente definidos, y considerando las características técnicas de cada uno de los componentes seleccionados.

En este sentido, con el objetivo de establecer los valores de cada componente resistivo para determinar la tensión y corriente soportada y aportada por el microcontrolador, por lo que se parte como tensión máxima de entrada y salida 3,3V, y con corrientes máxima de picos permitida de hasta 400mA.

Considerando estos datos, se calcula a continuación los valores ideales que deben poseer los componentes resistivos de la solución, siendo necesario para los Leds y pulsadores definidos previamente como entrada y salida:

Resistencia para el led rojo:

$$R_L = \frac{V_F - V_L}{I_L} = \frac{3,3v - 1,2v}{20mA} = 105\Omega$$

Resistencia para el led verde:

$$R_L = \frac{V_F - V_L}{I_L} = \frac{3,3v - 1,6v}{5mA} = 340\Omega$$

Resistencia para el pulsador: $8k\Omega - 10k\Omega$

De cara a definir el tipo de conexión de los pulsadores se ha estudiado las diferencias entre Pull-Up y Pull-Down.

En primer lugar se define Pull-Up la acción de elevar la tensión de entrada o salida que tiene un circuito lógico, y cuya configuración permite establecer en reposo una tensión suficiente para leer un 1 lógico, y al activarlo cambiar su tensión al 0 lógico (5 Hertz Electrónica, 2022).

Por otro lado, e le llama Pull Down a la acción de disminuir la tensión de entrada o salida que tiene un circuito lógico mientras está en reposo. Es decir se mantiene en 0 lógico en reposo, y al activarse se transforma en un 1 lógico (5 Hertz Electrónica, 2022).

En consecuencia, se determina que para el prototipo se debe hacer uso de la configuración de los pulsos como Pull Down como se muestra en la Figura 15, de manera que permitan contar por cada 1 lógico a la entrada del pin seleccionado, una vez se pulsa sobre éste.

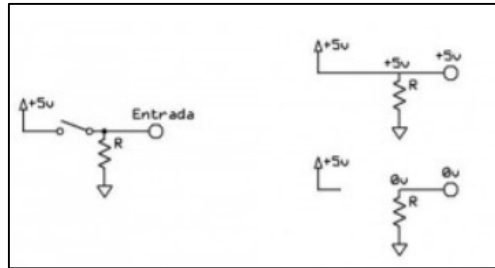


Figura 15. Conexión de pulsador en configuración Pull-Down. Fuente: (5 Hertz Electrónica, 2022).

Teniendo en cuenta que se tratan de entradas y salidas digitales, se definen los pines de conexión de la lectura de los pulsadores y leds, y cuya resistencia ya ha sido definida. Los pines finalmente seleccionados se describen en la Tabla 8, y han sido seleccionados basado en el datasheet del fabricante previamente estudiado (Az-Delivery, 2019)

Tabla 8. Definición de pines como entradas y salidas digitales según configuración de fabricante. Fuente: Elaboración propia

Componente	I/O	PIN
Led 1 Rojo	Salida	D3
Led 2 Verde	Salida	D4
Pulsador Strike	Entrada	D8
Pulsador balls	Entrada	D7
Pulsador Outs	Entrada	D6
Pulsador Innings	Entrada	D5

Adicionalmente, se ha estudiado el tipo de conexión necesaria para la incorporación de la pantalla OLED en la placa seleccionada. Teniendo en cuenta que la placa de desarrollo se basa en un chip ESP8266, se establece la necesidad de utilizar la interfaz I2C que para la placa corresponde a los pines D1 y D2.

Una vez se han seleccionado y definido los valores de los componentes y sus características de conexión, se ha diseñado el esquema del circuito con la conexión respectiva que se muestra en la Figura 16.

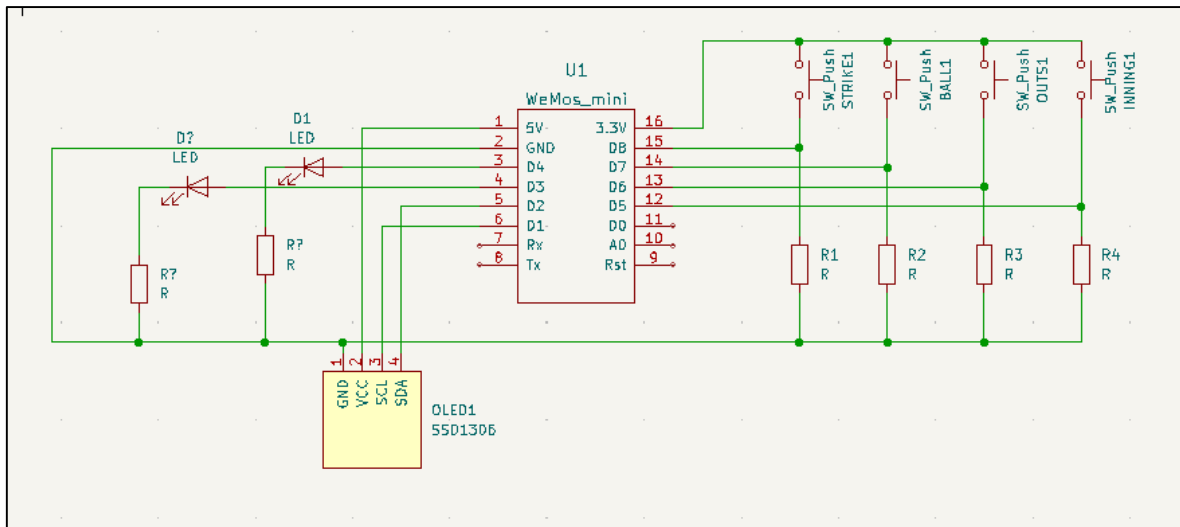


Figura 16. Diseño de circuito y conexión de componentes electrónicos. Fuente: Elaboración propia

4.2.1. Diseño y desarrollo software (Micro-controlador)

Tal y como se ha recomendado en el análisis de requerimiento, la programación del microcontrolador se realizará en el entorno de programación Arduino IDE, que además de ser compatible con la placa, es un entorno conocido que tiene la posibilidad de utilizar e instalar multitud de librerías que facilitan el desarrollo de la lógica del proyecto.

En esta línea, considerando los requerimientos técnicos y funcionales planteados previamente, se ha esquematizado el programa que se desea implementar sobre el microcontrolador, mostrado en la Figura 17, con la finalidad de estructurar a alto nivel las funciones necesarias para que éste simule el comportamiento del indicador de árbitro que se propone en el trabajo.

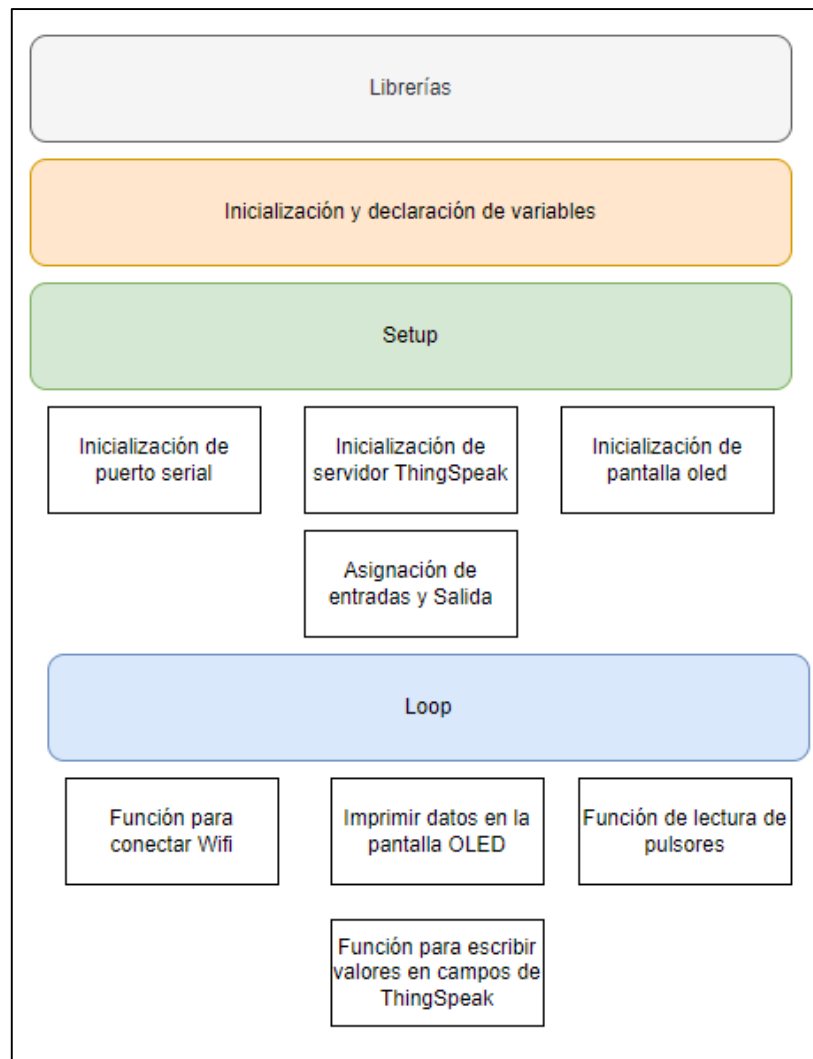


Figura 17. Esquema de estructura del programa sobre el microcontrolador. Fuente: Elaboración propia.

Tal y como se muestra en el esquema de estructura, el programa se dividirá en 4 principales bloques, librerías, Inicialización de variables, función de configuración (setup) y función principal (Loop).

El primer bloque del programa, se incorporan las distintas librerías que permiten facilitar la programación de los componentes del sistema, aprovechando las funcionalidades que brindan cada una de ellas simplificando el código. Por ello, y basado en los requisitos planteados, para el programa del indicador de árbitro se incorporan las siguientes librerías:

- ESP8266WiFi.h: Librería para la conectividad Wi-Fi de la placa seleccionada.
- ThingSpeak.h: Librería de *ThingSpeak*, que brinda funciones para publicación y suscripción de datos en los distintos canales de la plataforma.
- Wire.h: Librería para comunicación con dispositivos por bus I2C.

- `Adafruit_GFX.h` y `Adafruit_SSD1306.h`: Librería con las funciones para la escritura y configuración del OLED SSD1306.

Seguidamente, se encuentra el bloque de inicialización de variables, donde se declaran las variables que se utilizarán en el programa, algunas de ellas utilizadas para la lógica del programa, para la identificación de pines y otras como requerimiento de las distintas librerías. En este sentido podemos clasificar en 4 tipologías las declaraciones definidas en el programa:

- Variables para OLED, donde se definen las constantes y objetos para configurar la pantalla OLED SSD1306.
- Variables para *ThingSpeak*, donde se definen los datos del canal y API de la plataforma.
- Variables para conexión de Wi-Fi, donde se definen los datos de conexión de la red.
- Variables generales del programa, donde se definen la selección de pines, constantes y variables de conteos.

Continuando con el desglose de la estructura, se encuentra el bloque “setup”, que tal y como su nombre lo indica esta sección permite las configuraciones de la placa para el correcto funcionamiento del programa y el control del mismo. Dentro de esta función se encuentran:

- Asignación de pines: Donde se identifican los tipos de pines a utilizar y/o declarar su rol (Entrada/Salida).
- Inicialización y configuración: Se compone de una serie de funciones que permiten inicializar, configurar y comprobar el estado del servidor, puerto y pantalla.

Finalmente en el void loop, como función principal, se ejecutan las distintas líneas de lógica de funcionamiento, impresión, y comunicación de los datos, constituido principalmente en 4 funciones, que se encargan de la conectividad, la lógica de lectura, escritura y el envío de datos.

La primera función que se ejecuta en el programa, corresponde a la función de conexión a la red Wi-Fi, que aprovecha las clases brindada por la librería para comprobar la conectividad de la red, proponiendo el algoritmo regido por el diagrama de flujo presentado en la Figura 18.

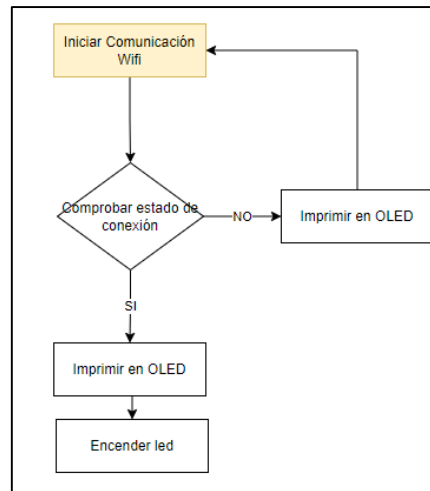


Figura 18. Diagrama de flujo de función para iniciar y comprobar conexión de Wi-Fi. Fuente: Elaboración propia.

Seguidamente, en el programa sigue una serie de secuencia lógica para la impresión y publicación de los datos, todo ello basado en los requisitos funcionales definidos que permita seguir la lógica y reglas del beisbol. Es decir, para cada variable del juego (strike, balls, outs e inning), que depende de un cambio de estado en la lectura de los respectivos pines digitales conectado a los pulsores, se ejecutará una serie de funciones dependiendo de las condiciones planteadas en los requerimientos.

De manera similar, las funciones y condiciones del conteo para balls y strike se asemejan en su lógica, que tal y como se observa en la Figura 19, cumplen con la misma secuencia diferenciada por la condición máxima del conteo.

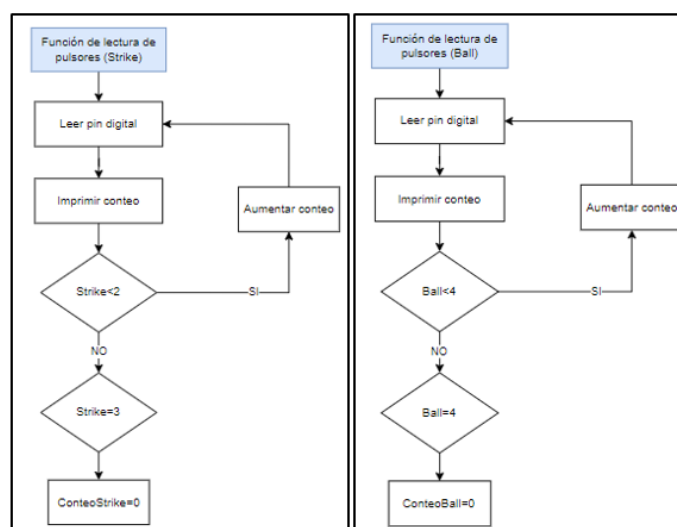


Figura 19. Diagrama de flujo de funciones lógicas para el conteo de Strike y Ball. Fuente: Elaboración propia.

De manera similar, se ha diseñado el flujo grama con la lógica y condiciones requeridas para realizar el conteo de outs e inning, que a diferencia de las funciones para el conteo de balls strike, estas influyen en la inicialización del resto de conteos cuando se cumple cierta condición establecida en el apartado de requisitos funcionales. En consecuencia ambas secuencias diseñadas se muestran en la Figura 20.

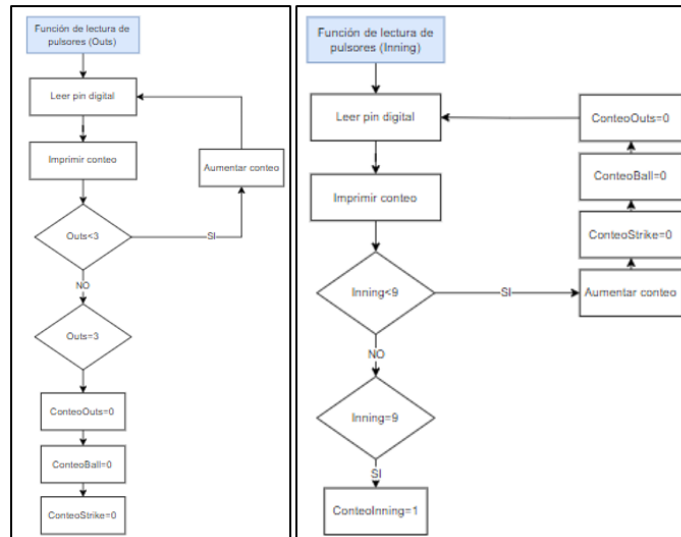


Figura 20. Diagrama de flujo de funciones lógicas para el conteo de Outs e Inning. Fuente: Elaboración propia.

Adicionalmente, dentro del loop el programa deberá enviar los datos actualizados a *ThingSpeak*, que bajo las ventajas que ofrece su librería se realiza cada cierto tiempo con una simple instrucción que permite actualizar el estado de los distintos conteo al canal que corresponda. Estas sentencias junto con el resto del programa se muestran en el ANEXO A del trabajo, proporcionando con ello el detalle del programa desarrollado.

4.2.1. Diseño e implementación de plataforma IoT

Tal y como se ha mencionado con anterioridad, la plataforma IoT con la que se trabaja en el proyecto es *ThingSpeak*, esta plataforma servirá de Broker para establecer la lectura y escritura de los datos de manera online, capturando por tanto los datos del dispositivo hardware y enviando los mismos a la aplicación móvil

Para iniciar, con la configuración e integración de la plataforma con el resto del sistema, se ha creado una cuenta en *ThingSpeak* utilizando un correo electrónico universitario, permitiendo seguidamente, realizar la configuración de los canales necesarios para el

proyecto. La cuenta gratuita creada, permite abrir un canal y enviar más de 2 millones de mensajes por día y con un envío de datos de al menos 15 segundos.

En este sentido, la plataforma cumple con los requisitos necesarios para creación del canal y los campos necesarios en el prototipo planteado. Por ello, se ha configurado tal y como se indica en la Tabla 7, el canal con los respectivos campos propuesto en de la plataforma.

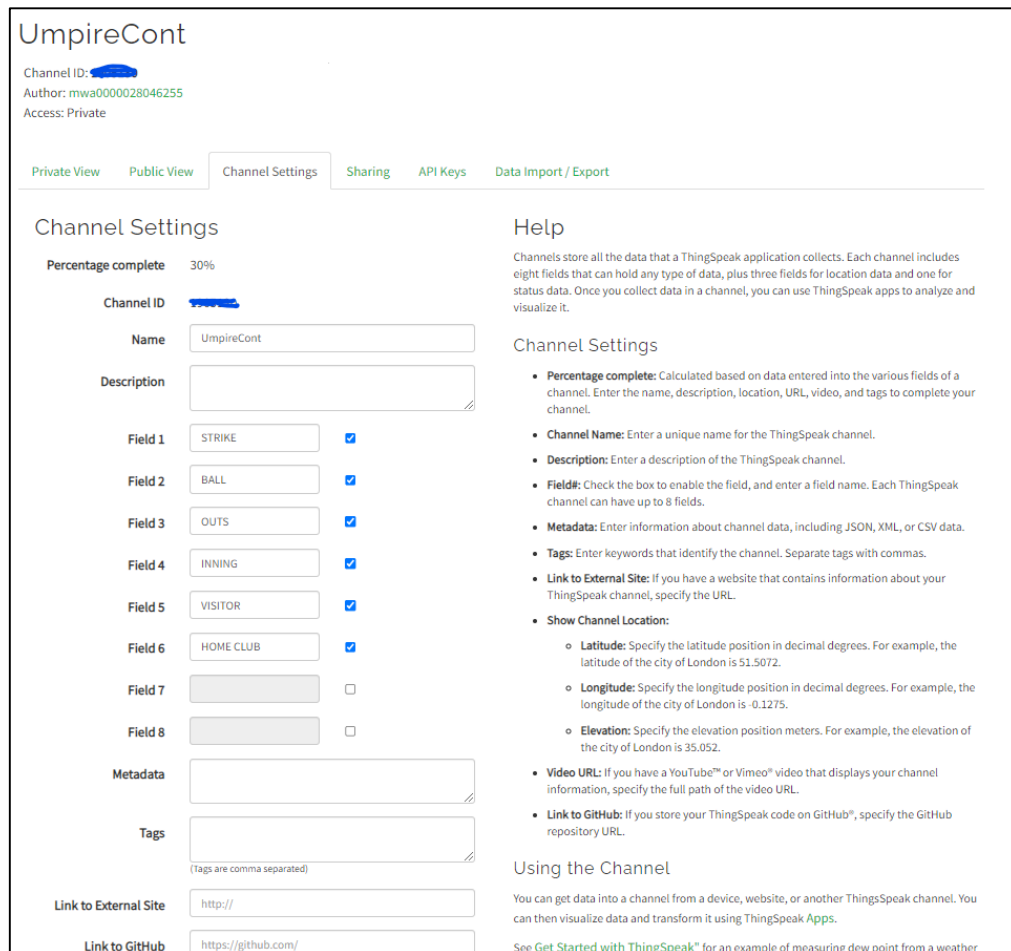


Figura 21. Configuración de Canales en *ThingSpeak*. Fuente: Elaboración propia.

Como se puede observar en la Figura 21, se definen los campos con indicadores de los datos que recibirá, permitiendo aprovechar la interfaz de presentación que brinda *ThingSpeak* para visualizar y supervisar la comunicación de los datos.

Una vez creado los distintos campos en *ThingSpeak*, se han extraído los identificadores (Channel ID) y claves (API Keys) de cada canal, mostrados en la Figura 22, estos datos se utilizan para realizar la conexión y comunicación entre el microcontrolador y la plataforma.

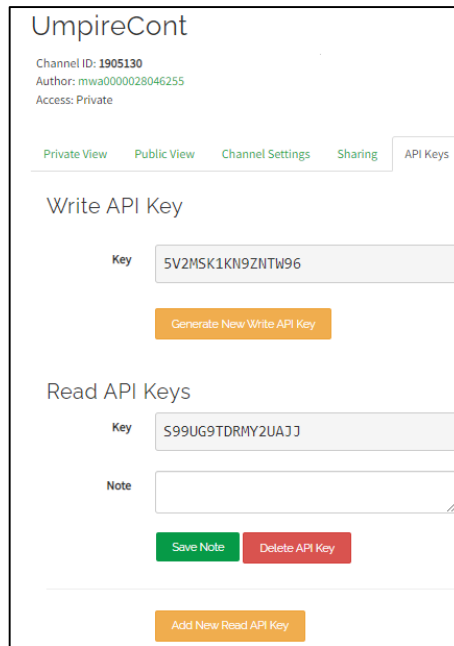


Figura 22. Claves de escritura y lectura para canales creados en *ThingSpeak*. Fuente: Elaboración propia.

Para visualizar los datos del contador enviados desde el microcontrolador desde *ThingSpeak*, teniendo en cuenta los requerimientos planteados, se han incluido displays utilizando “*Widgets*”, donde se representan los datos de cada variable del contador recibido en el canal, permitiendo supervisar el conteo recibido en tiempo real.

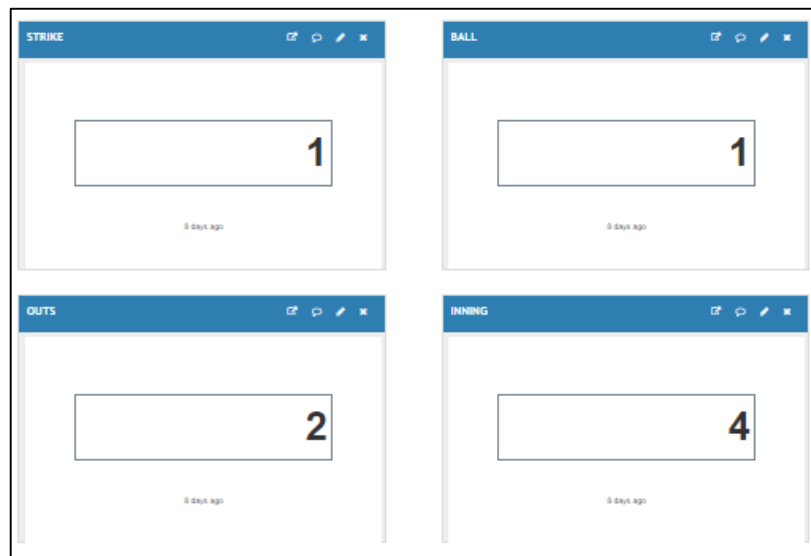


Figura 23. *Widgets* creado para la visualización digital de los canales en *ThingSpeak*. Fuente: Elaboración propia.

Finalmente, para la lectura y escritura de los datos desde la APP móvil, se extraen las claves API (read y write), que serán utilizadas en el programa del microcontrolador y back-end de la aplicación móvil para visualizar y registrar los datos respectivamente.

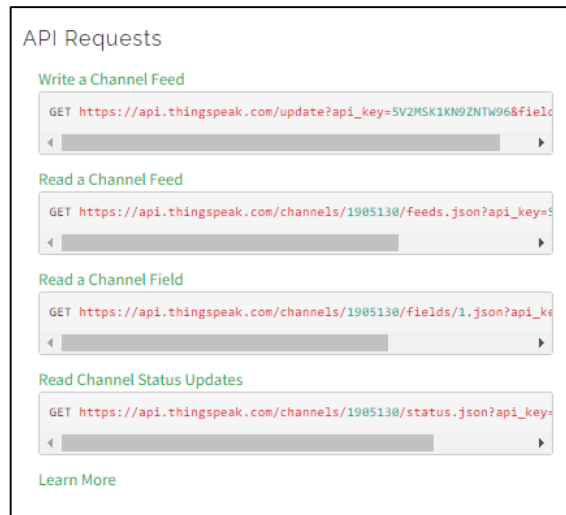


Figura 24. API Requests de campos y canales creados en *ThingSpeak*. Fuente: Elaboración propia.

4.2.2. Diseño y desarrollo software (APP Móvil)

Tal y como se ha indicado en el estudio de tecnologías y en el análisis de requisitos técnicos y funcionales, para el diseño y desarrollo de la APP Móvil, se hace uso de la plataforma de desarrollo MIT APP Inventor, que brinda una interfaz de programación intuitiva y sencilla, con dos tipos entornos (gráfico y lógico).

La plataforma, dispone de una serie de elementos y componentes básicos que pueden incorporarse el diseño como un objeto moldeable, algunos de estos componentes son: *Screen*, *Botón*, *Lienzo*, *CheckBox*, *Clock*, *Text Box*, *Slider*, *ListPicker*, *ListView*, etc.

Considerando los componentes que proporciona la herramienta y los requisitos de presentación necesarios en el prototipo y se propone la introducción de una serie de elementos y componentes necesarios para brindar las funcionalidades básicas del indicador, tal y como se muestra en Tabla 9.

Tabla 9. Definición de componentes y características básicas para el funcionamiento básico de la APP Móvil. Fuente: Elaboración propia.

Componentes	Sub-Componentes	Característica
Screen	<i>Screen1</i>	Interfaz principal para la introducción y selección del juego.
	<i>Screen2</i>	Interfaz de juego en vivo
Campos de Escritura	Text Box	Campos que permitan registrar los equipos participantes
Botón de registro	Button	Botón que permite el registro de equipos
Campos de Escritura	Web Viewer	Campo que permite la lectura de

Estos componentes, además de brindar la interfaz gráfica para la inserción y colocación de los mismos, vienen acompañado de una función objeto que podrá ser utilizada en el back-end de la aplicación.

Asimismo, con la finalidad de desarrollar una aplicación amigable con el usuario final, se plantea crear al menos 2 Screen o pantallas, que permitan por un lado dar la bienvenida al espectador proporcionándole las opciones de juego, y una segunda pantalla que muestre los datos del contero del juego seleccionado.

Por ello, en primer lugar se ha desarrollado la pantalla de inicio (SCREEN 1) que contendrá la aplicación móvil, siendo esta la encargada de mostrar una bienvenida a la aplicación y una serie de opciones para seleccionar el juego que desea monitorizar.

El desarrollo de esta interfaz ha consistido en la implementación de tareas de diseño Front-end y Back-end que se describen a continuación:

- Front-end: para la primera se ha realizado un diseño sencillo que consta de la incorporación de gráficos temáticos como el logo de la federación española de beisbol y componentes de tipo botón, como se observa en la Figura 25, que realizarán la acción de re-direccionar a la pantalla con los datos del juego seleccionado.



Figura 25. Diseño Front-end de SCREEN 1 (pantalla de inicio) de la APP Móvil. Fuente: Elaboración propia.

- Back-end: aprovechando las clases de los objetos creados por cada componente agregado en el Front-end de la pantalla, se han establecido las funciones de llamadas a otros Screen, tal y como se muestra en la Figura 26. Es decir, el programa redirigirá al usuario a la pantalla con los datos del juego seleccionado que desea visualizar o monitorizar.

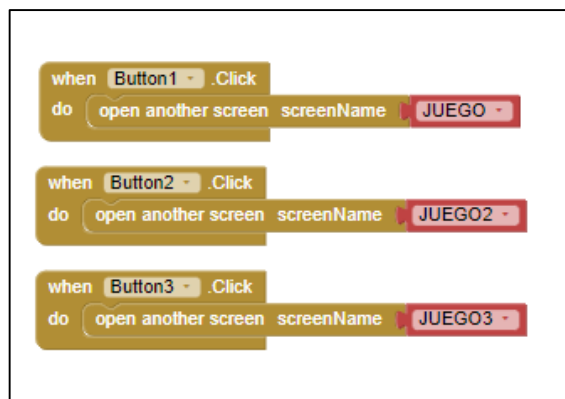


Figura 26. Desarrollo Back-end de SCREEN 1 (pantalla de inicio) de la APP Móvil. Fuente: Elaboración propia.

Por su parte, en lo que respecta a las pantallas o interfaces de juego (SCREEN 2-N), se han diseñado y desarrollado basada en una serie de componentes gráficos que permita visualizar los datos del indicador del juego en vivo, y además registrar los equipos que participan en el mismo.

De manera similar, se describen a continuación las consideraciones y características para la implementación del Front-end y Back-end de estas pantallas.

- Front-end: para las pantallas de juego los componentes del diseño incorporado, se basan en campos de textos, títulos, visualizadores web para la lectura, botón y arreglos para la ubicación de los componentes, organizados tal y como se muestra en la Figura 27. Permitiendo así dar una apariencia organizada de los datos y con ello una mejor experiencia al usuario.



Figura 27. Diseño Front-end de SCREEN 2 (pantalla de juego) de la APP Móvil. Fuente: Elaboración propia.

- Back-end: el desarrollo de las funciones en back-end de las pantallas de juego, han consistido en la implementación de 2 principales bloques de lógica. El primero permite la publicación de los campos de escrituras, que tal y como se muestra en la Figura 28 permite registrar en los respectivos campos de *ThingSpeak* una serie de datos de tipo texto que corresponden funcionalmente al registro de los equipos participantes del juego.

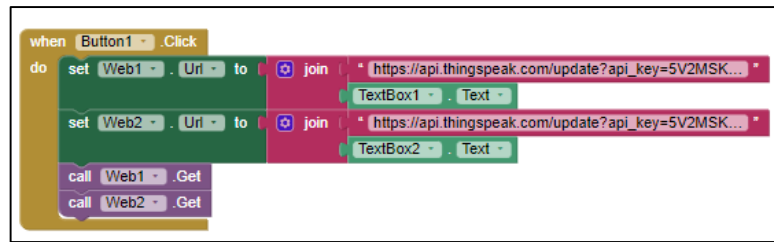


Figura 28. Desarrollo Back-end de SCREEN 2 (pantalla de juego-Bloque1) de la APP Móvil. Fuente: Elaboración propia.

El segundo bloque de las interfaces de juego, consta de funciones de presentación de los datos de cada widgets de *ThingSpeak*, a través de su dirección web, permitiendo con ello visualizar cada cambio en el canal y campo indicado de la plataforma, tal y como se muestra en la Figura 29.

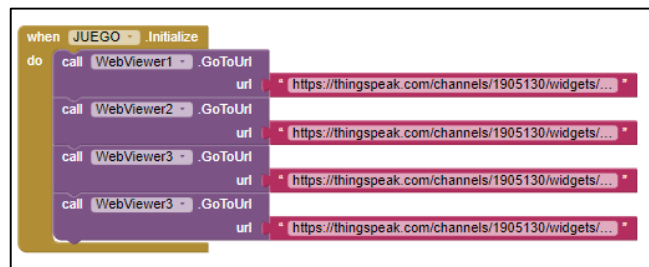


Figura 29. Desarrollo Back-end de SCREEN 2 (pantalla de juego-Bloque2) de la APP Móvil. Fuente: Elaboración propia.

4.3. Desarrollo e integración de la propuesta

El desarrollo e integración de la propuesta se basa en la construcción del prototipo a nivel de hardware, la compilación del programa del microcontrolador y la instalación de la APP móvil del indicador de árbitro desarrollado.

Para ello, en primer lugar se ha realizado el acondicionamiento del microcontrolador, teniendo en cuenta que el prototipo del proyecto se construirá sobre un protoboard, por lo que ha sido necesario la incorporación de pines macho para incorporarlo en el protoboard. Que tal y como se puede observar en la Figura 30 se trata de los componentes listados en los requisitos y diseño del circuito previo definidos en la Figura 16.

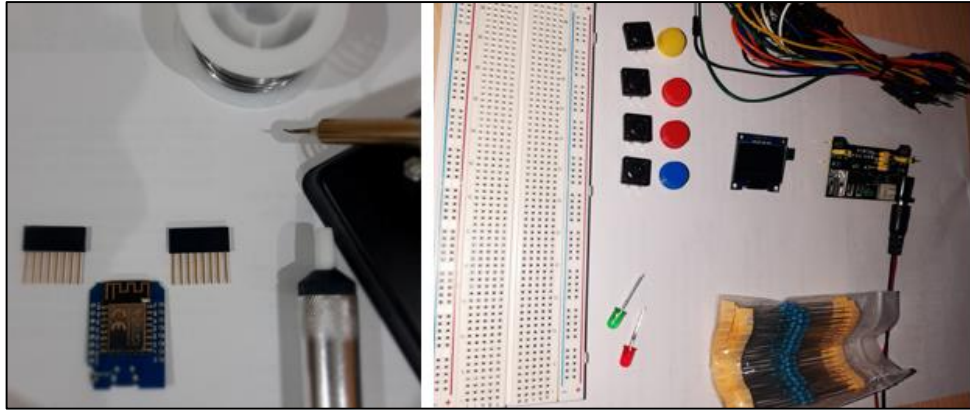


Figura 30. Componentes para el montaje y construcción del prototipo. Fuente: Elaboración propia.

Seguidamente, se ha realizado el respectivo montaje de los componentes, que tal y como se muestra en la Figura 31, se realiza tal y como lo establece el esquema del diseño hardware, y siguiendo la configuración de pista proporcionada por el protoboard.

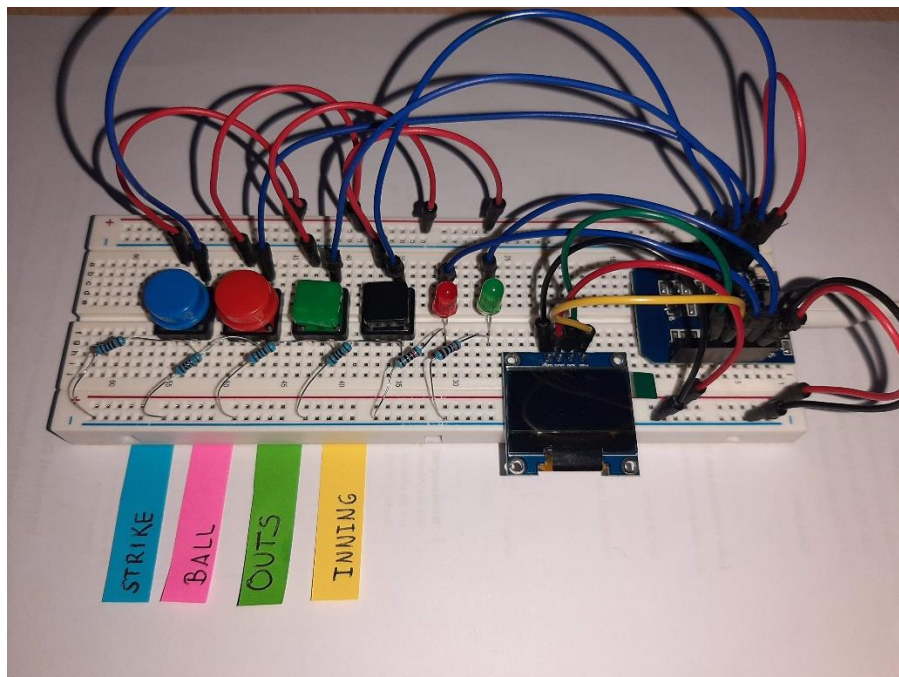


Figura 31. Montaje en protoboard de prototipo de indicador. Fuente: Elaboración propia.

Una vez se ha realizado el respectivo montaje, se ha cargado el programa y se ha comprobado el funcionamiento de cada uno de los componentes con distintas técnicas de medición y validación que se detallan en el siguiente apartado. (Figura 32)

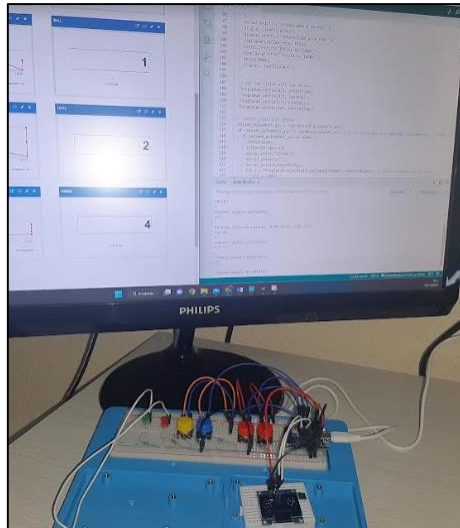


Figura 32. Integración y validación del prototipo en protoboard. Fuente: Elaboración propia.

Con respecto a la APP móvil, una vez desarrollada la aplicación, se procede a la instalación del archivo .apk generado por la plataforma de desarrollo MIT Inventor, que tal y como se muestra en la Figura 33, se realiza a partir de dos opciones de descarga, desde la lectura de un QR o la descarga del documento de tipo apk para la respectiva instalación.

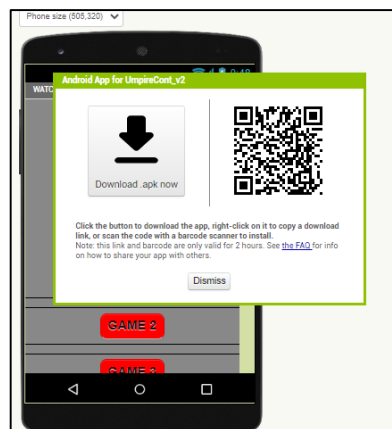


Figura 33. Instalación de APP móvil desde QR. Fuente: Elaboración propia.

Finalmente, la APP Móvil desarrollada adquiere la apariencia mostrada en la Figura 34, siendo esta una imagen capturada desde un dispositivo Android compatible para su instalación, y que comparando con la interfaz de desarrollo proporcionada por la plataforma de MIT Inventor, se trata de una réplica exacta de sus interfaces gráficas proporcionadas durante el diseño.



Figura 34. Primera versión de APP Móvil vista desde un dispositivo Android. Fuente: Elaboración propia.

4.4. Ejecución de pruebas de validación

Una vez desarrollado e integrado los distintos componentes del sistema y el dispositivo, se han realizado una serie de pruebas y validaciones que buscan validar el funcionamiento de las distintas partes y la validación del sistema en general para comprobar el cumplimiento de los objetivos propuestos.

Para ello, se realizaron una serie de pruebas de validación que van desde la comprobación del funcionamiento de la electrónica, la comprobación de conexión y comunicaciones del microcontrolador y el funcionamiento general en el envío de datos.

En primer lugar la comprobación de funcionamiento electrónico han consistido en ejecutar las siguientes pruebas:

- Pruebas de continuidad: durante el montaje de los componentes en el protoboard se ha comprobado incorporación de los componentes, mediante la identificación de continuidad entre las conexiones del protoboard mediante el uso de un multímetro con la funcionalidad de continuidad.
- Pruebas de funcionamiento: esta prueba ha consistido en ejecutar pequeños programas individuales de lectura y escritura sobre cada tipo de componente, con el objetivo de comprobar que responde de la manera esperada.

Seguidamente, se ha realizado la validación y comprobación de comunicación con el canal de *ThingSpeak*, que ha consistido en realizar dos tipos de pruebas que se describen a continuación:

- Pruebas de comunicación con *ThingSpeak* con un campo y un canal: esta prueba ha consistido en crear un primer programa que consiste en el envío de un único campo al canal de la plataforma *ThingSpeak*.

Para la ejecución de esta prueba el programa se ha simplificado para mostrar tanto en el monitor serial de Arduino IDE como en la plataforma de *ThingSpeak* el conteo de un único pulsador, indicando en el monitor serial si el cambio de la variable ha sido publicado de manera exitosa como en los ejemplos presentados con las Figuras (Figura 35 y Figura 36).

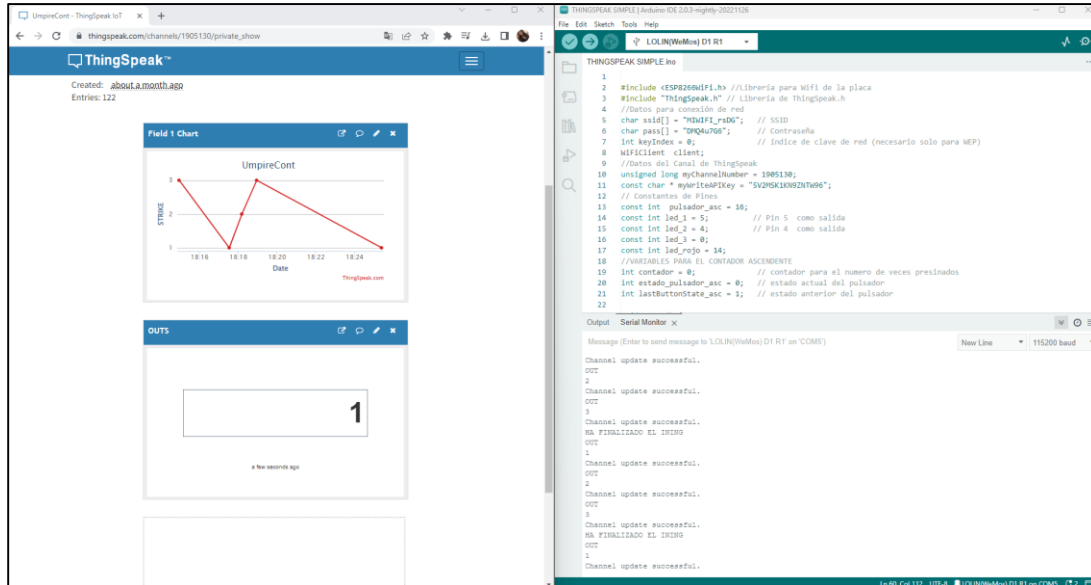


Figura 35. Pruebas de comunicación con la plataforma de *ThingSpeak* y puerto serial (1Outs). Fuente:

Elaboración propia.

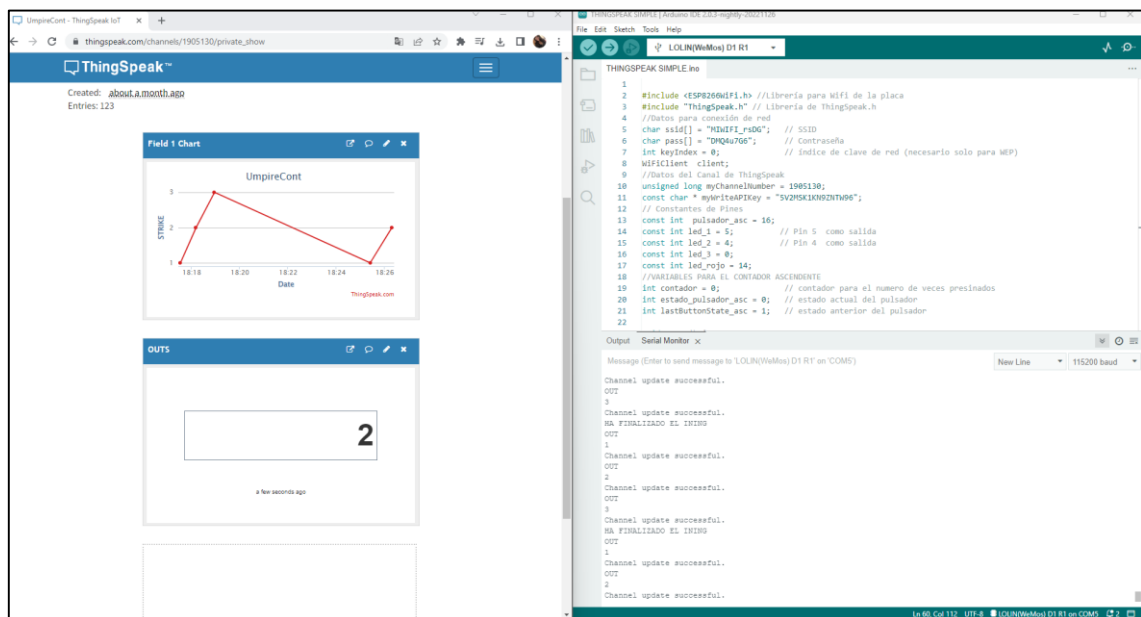


Figura 36. Pruebas de comunicación con la plataforma de *ThingSpeak* y puerto serial (2Outs). Fuente:

Elaboración propia.

- Pruebas de comunicación con *ThingSpeak* con cuatro campos y un canal: una vez superada la comprobación de la escritura del microcontrolador sobre el canal sobre un único campo, se ha implementado el programa que integra todos los canales necesarios para presentar las 4 variables del indicador, con el fin de percibir si se produce una variación en la comunicación cuando se envían más de un dato al canal de *ThingSpeak*.

En esta prueba, tal y como se muestra en la Figura 37, se ha demostrado la capacidad para la comunicación y publicación de múltiples datos de manera exitosa y sin sufrir pérdidas importantes de los datos.

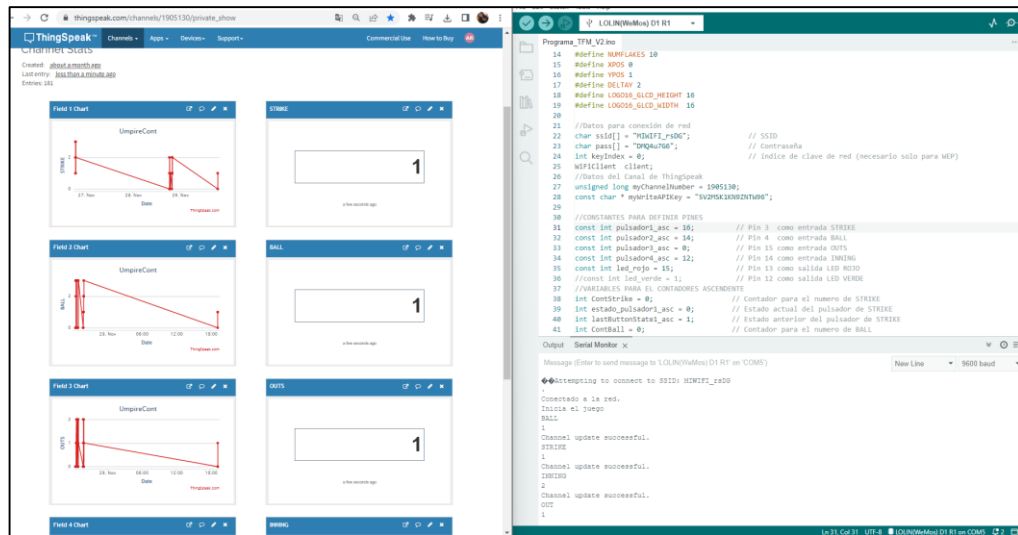


Figura 37. Prueba de comunicación con *ThingSpeak* con 4 Canales. Fuente: Elaboración propia.

Finalmente, se ha realizado la prueba de validación del sistema final, que ha consistido en comprobar la comunicación y los tiempos de respuesta desde que se acciona un pulsador, hasta que se publica en la aplicación móvil.

Esta prueba también es supervisada el envío de datos de cada pulsador, a través del monitor serial de arduino IDE, la plataforma de *ThingSpeak* y la aplicación móvil tal y como se muestra en la Figura 38. Para la prueba se ha confirmado la correcta lectura e impresión de los datos en la pantalla OLED y el tiempo de comunicación desde la ejecución del pulso hasta la publicación en la aplicación.

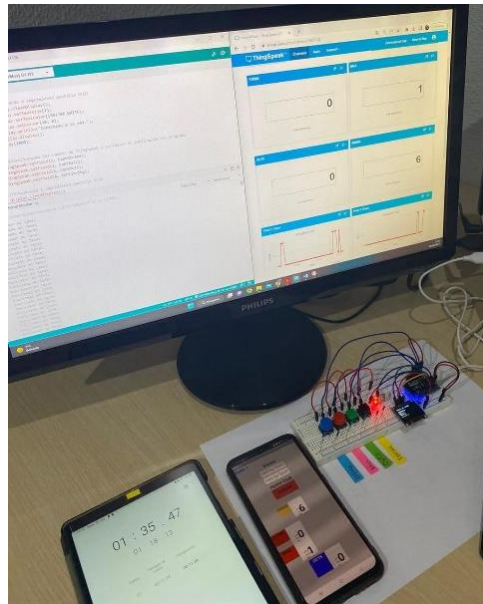


Figura 38. Ejecución de prueba Final del dispositivo. Fuente: Elaboración propia.

Finalmente se concluye que de una muestra de 50 pulsaciones para el conteo de las variables del juego se han entregado 100% de los datos. En cuanto al tiempo de publicación entre el pulso y la visualización en la APP Móvil y la aplicación, tal y como se muestra en la Figura 39, existe en promedio 15,52seg para la publicación en *ThingSpeak* y 20,35seg para la visualización en la aplicación móvil.

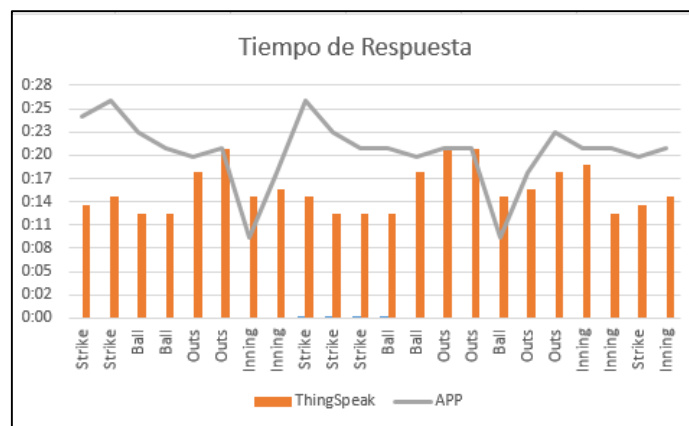


Figura 39. Resultado de pruebas de tiempo de publicación y visualización de datos. Fuente: Elaboración propia.

Además de la visualización de los datos a través de la APP móvil se ha comprobado a través de una extracción de archivo JSON de los canales 5 y 6 de la plataforma *ThingSpeak*, la correcta escritura o registro de los equipos participantes, comprobando así la posibilidad de hacer uso de estos datos para otras aplicaciones.

```
{
  "created_at": "2022-11-28T18:47:10Z", "entry_id": 162, "field6": null,
  {"created_at": "2022-11-28T18:47:46Z", "entry_id": 163, "field6": null},
  {"created_at": "2022-11-28T18:48:44Z", "entry_id": 164, "field6": null},
  {"created_at": "2022-11-28T18:49:08Z", "entry_id": 165, "field6": null},
  {"created_at": "2022-11-28T18:49:57Z", "entry_id": 167, "field6": null},
  {"created_at": "2022-11-28T18:50:30Z", "entry_id": 168, "field6": null},
  {"created_at": "2022-11-28T18:50:50Z", "entry_id": 169, "field6": null},
  {"created_at": "2022-11-28T18:51:24Z", "entry_id": 170, "field6": null},
  {"created_at": "2022-11-28T18:52:04Z", "entry_id": 172, "field6": null},
  {"created_at": "2022-11-28T18:52:32Z", "entry_id": 173, "field6": null},
  {"created_at": "2022-11-28T18:53:00Z", "entry_id": 174, "field6": null},
  {"created_at": "2022-11-28T18:53:18Z", "entry_id": 175, "field6": null},
  {"created_at": "2022-11-28T18:54:56Z", "entry_id": 177, "field6": null},
  {"created_at": "2022-11-29T18:06:46Z", "entry_id": 178, "field6": null},
  {"created_at": "2022-11-29T18:07:04Z", "entry_id": 179, "field6": null},
  {"created_at": "2022-11-29T18:07:39Z", "entry_id": 180, "field6": null},
  {"created_at": "2022-11-29T18:09:37Z", "entry_id": 182, "field6": null},
  {"created_at": "2022-11-29T18:10:50Z", "entry_id": 183, "field6": null},
  {"created_at": "2022-12-07T14:35:35Z", "entry_id": 184, "field6": "Vikings"},
  {"created_at": "2022-12-07T14:36:38Z", "entry_id": 185, "field6": "Vikings"}
}
```

Figura 40. Extracto de Datos exportados en formato JSON para comprobar la escritura desde la APP móvil. Fuente: Elaboración propia.

De las distintas pruebas se ha detectado cierta variación en los tiempos de publicación de los datos, sin embargo, sí se han entregado todos los datos generados por cada pulso. Por lo que si se considera el tiempo de publicación mínimo de la plataforma *ThingSpeak* se puede concluir que el sistema funcional tal y como se ha diseñado.

A pesar de ello, considerando que en la práctica del deporte el conteo de cada sentencia del árbitro (strike, ball, outs e inning) tiene un intervalo de al menos 15 segundo de promedio, según (Pardo Lozano, 2021), que determina el tiempo entre un lanzamiento y otro, se da por validado a nivel funcional el primer prototipo propuesto.

4.5. Evolutivos y propuesta de mejoras

Durante el desarrollo del proyecto, se han detectado una serie de mejoras que se podrían incluir en el prototipo del indicador que se detallan a continuación:

1. Para que el indicador sea funcional se debe adaptar el diseño del HW de manera que se pueda incorporar dentro de un mecanismo portátil y con un tamaño mucho mejor al propuesto en el proyecto.
2. De cara a mejorar los tiempos de transmisión de los datos con la plataforma, se pueden estudiar otras opciones de plataformas gratuitas que permitan la gestión de datos con una mayor periodicidad que 15 segundos.
3. A nivel de aplicación, es posible mejorar la apariencia de los indicadores de conteo, a través del uso de componentes de diseño que permitan guardar los datos en una variable interna y que se impriman en un formato adaptado a la aplicación. Para ello también será necesario el uso de un reloj interno para utilizarlo de referencia de cara a actualizar los datos.

4. Asimismo, se detecta la posibilidad de incorporar una funcionalidad que permita formatear de manera instantánea los datos a 0, sin necesidad de que se cumpla la lógica del juego para la inicialización, con el fin de darle la posibilidad al usuario de rectificar la decisión en el indicador.
5. Funcionalmente, se detecta la posibilidad de incorporar otras lógicas para guardar datos del juego, como el conteo de lanzamiento. Esto es posible, debido a que la sumatoria de todas las variables de ball y strike suman por cada inning el total de lanzamientos por cada uno de los lanzadores del juego.

5. Conclusiones y Trabajos futuros

El proyecto ha permitido desarrollar un prototipo de indicador de árbitro a través de tecnología IoT, transmitiendo el conteo de las variables de un juego de beisbol en tiempo real o cercano a este. Por tanto, el proyecto ha logrado cumplimentar cada uno de los objetivos propuesto, que han permitido guiar la consecución del objetivo principal del mismo:

- Se han analizado y definido los componentes electrónicos para la construcción del prototipo de indicador de árbitro.
- Se ha diseñado y construido el prototipo del indicador, con los transductores adecuados para el entorno y las funcionalidades necesarias para registrar y transmitir los datos de manera inalámbrica.
- Se ha programado y desarrollado sobre la respectiva placa de desarrollo seleccionada, para establecer la lógica necesaria en el prototipo según las reglas del juego, la forma de registro y su comunicación.
- Se ha desplegado una arquitectura IOT para lograr la disponibilidad de los datos en una plataforma en la nube.
- Y se ha diseñado y desarrollado una aplicación móvil para visualizar los datos transmitidos en tiempo real.

Durante la ejecución del proyecto y con los resultados conseguidos se destacan las siguientes conclusiones:

En primer lugar, cabe destacar la importancia del estudio previo de elementos y componentes para un sistema IoT, permitiendo en el presente proyecto adquirir componentes adaptado a la necesidad y a un bajo coste.

Por su parte, el uso de la placa de desarrollo AZ-Delivery D1 mini basado en el SoC ESP8266 además de sus ventajas físicas y técnicas, ha permitido desarrollar el programa del microcontrolador, debido a su compatibilidad con plataformas de desarrollo de código abierto, aprovechando distintas aportaciones de la comunidad a través de librerías y ejemplos existentes en la actualidad.

Asimismo, se puede concluir que *ThingSpeak* como plataforma IoT de la capa Middleware ofrece ventajas interesantes, que pueden aprovecharse para análisis avanzados de datos como estadísticas de sabermetría en una posible evolución del prototipo desarrollado. Sin embargo, aunque para el alcance del proyecto se cumple con los tiempos requeridos, se destaca que su tasa de transmisión de datos no son las más competitivas en el mercado.

En esta línea, el proyecto ha permitido conocer las distintas opciones para la comunicación de datos entre las 3 capas de la arquitectura propuesta, que para aplicaciones como la que se ha desarrollado, puede implementarse protocolos ligeros como MQTT que funcionan perfectamente para consumo bajo y bajo ancho de banda.

Finalmente, se determina que, a pesar de desarrollar la presente propuesta con un presupuesto asequible, de cara a una propuesta comercial, se debe considerar reducir este importe para competir con el producto mecánico existente (que es muy bajo), debido a que la principal motivación del proyecto ha sido la incorporación de esta tecnología en competiciones no televisadas, categorías no profesionales y por tanto con pocos recursos económicos.

Referencias

- 5 Hertz Electrónica. (2022). *Resistencias Pull-Up*. Obtenido de 5 Hertz Electrónica: https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=10
- Aguayo, P. (10 de Noviembre de 2004). *INTRODUCCIÓN AL MICROCONTROLADOR*. Obtenido de ACADEMIA: <http://usuarios.lycos.es/sfriswolker>
- Amazon Web Services, Inc. (2023). *¿Qué es el middleware?* Obtenido de AWS: <https://aws.amazon.com/es/what-is/middleware>
- Arduino y solo Arduino. (1 de Marzo de 2017). *¿Qué es un ESP32?* Obtenido de Arduino y solo Arduino: <https://soloarduino.blogspot.com/2017/03/que-es-un-esp32.html>
- Az-Delivery. (28 de Agosto de 2019). *Az-Delivery D1 mini ESP8266MOD 12-F*. Obtenido de AZ-Delivery : https://cdn.shopify.com/s/files/1/1509/1638/files/Betriebsanleitung-AZ-D1miniV1.2_2.pdf?v=1590603445
- Baes, F. (2017). *Simbologías Usadas*. Obtenido de Anotador: <http://anotador.mygamesonline.org/pag-simbolos.html>
- Benyezza, H., Bouheddam, M., & Djellout, K. (2018). Smart Irrigation System Based Thingspeak and Arduino. *2018 Internacional Conference on Applied Smart System (ICASS 2018)*, 24-25.
- Bueno, P. (03 de Febrero de 2017). *Cómo la tecnología ha cambiado (y cambiará) el fútbol*. Obtenido de alphr: <http://alphr.es/tecnologia/tecnologia-en-el-futbol-avances-cambian-deporte/>
- Carlos Galera-Zarco, M. O.-B. (2015). El Internet de las Cosas (IoT) en la gestión deportiva: principios, conceptos y aplicaciones. *Revista Intercontinental de Gestión deportiva*. 5 (2), 128-142.
- César Soto Valero, M. G. (2015). Sabermetría y nuevas tendencias en el análisis estadístico del juego de béisbol. *Federación Española de Asociaciones de Docentes de Educación Física (FEADEF)*. 28, 122-127.
- ciberseguridad.com. (s.f.). *ciberseguridad.com*. Obtenido de PRINCIPALES PLATAFORMAS IOT DE CÓDIGO ABIERTO QUE LAS EMPRESAS DEBEN CONOCER:

https://ciberseguridad.com/herramientas/plataformas-iot-codigo-abierto/#%C2%BFQue_son_las_plataformas_IoT

Clark Hills, D. V. (02 de Agosto de 2007). *Digital baseball and softball umpire game status indicator*. Obtenido de Google Patents: <https://patents.google.com/patent/US20070030128A1/en>

Emol. (22 de Septiembre de 2015). *Conoce las tecnologías aplicadas al arbitraje deportivo*. Obtenido de EMOL: <https://www.emol.com/noticias/Tecnologia/2015/09/22/751004/Conoce-las-tecnologias-aplicadas-al-arbitraje-deportivo.html>

Evans, D. (Abril de 2011). *How the Next Evolution of the Internet Is Changing Everything*. Obtenido de Cisco: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

Gallardo, M. A. (2007). *Tecnologías Aplicadas al Deporte de Alto Rendimiento 50*. Madrid: TADAR.

Gascón Bononad, J. (Julio de 2019). *Diseño de una Placa Electrónica Modular para el Microcontrolador NodeMCU*. Obtenido de Universidad Politécnica de Valencia: https://riunet.upv.es/bitstream/handle/10251/125907/73226575A_TFG_1562923147379404345145595615760.pdf?sequence=2

Geethalakshmi, V., Fathimath , K., & Sinan Marikattay, M. (2019). Architecture of an IoT-based System for Cricket Supervision. *www.jetir.org (ISSN-2349-5162)*, Volumen 6, Número 5.

Gonzalez, M. (29 de Julio de 2015). *Un ordenador decide si hay strike o no en béisbol y nadie puede protestarle*. Obtenido de XATAKA: <https://www.xataka.com/otros/un-ordenador-decide-si-hay-strike-o-no-en-beisbol-y-nadie-puede-protestarle>

HBK company. (2021). *HBM*. Obtenido de Los transductores y el internet de las cosas: <https://www.hbm.com/es/7501/los-transductores-y-el-internet-de-las-cosas/#:~:text=Un%20actuador%20es%20un%20transductor,hace%20por%20efecto%20de%20actuadores.>

- Herrera, J. (28 de Junio de 2012). *Beisbolysftbol.com*. Obtenido de Anotación Béisbol y Softbol: <https://anotacionbeisbolysftbol.wordpress.com/2012/06/28/que-es-un-anotador-oficial/>
- Hung, M. (2017). Leading the IoT. *Gartner Insights on How to Lead*, 5-6.
- Ikram, M., Alshehri, M., & Hussain, F. (2015). Architecture of an IoT-based system for football supervision (IoT Football). *IEEE 2nd World Forum on Internet of Things (WF-IoT)* (págs. 69-74). Milan: IEEE.
- IoT Consulting. (6 de Diciembre de 2019). *Cómo elegir un microcontrolador para IoT*. Obtenido de IoT Consulting: <https://iotconsulting.tech/como-elegir-un-microcontrolador-para-iot/>
- Laura Fava, D. V. (2018). Tecnología aplicada al deporte de alto rendimiento. *RedUNCI - UNNE - ISBN 978-987-3619-27-4*, 864-868.
- Llamas, L. (21 de Febrero de 2019). *Protocolos de comunicación para IoT*. Obtenido de L. Llamas: <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- Malo, M. P. (2021). *Análisis de la influencia de las emociones en la toma de decisiones en árbitros de fútbol mediante un sistema prototipo de sensores*. Obtenido de Repositorio Universidad de Guayaquil: <http://repositorio.ug.edu.ec/handle/redug/52272>
- Martínez, D. P. (2021). *Aplicación de la estadística en el ámbito deportivo*. Obtenido de Repositorio de Universidad de A Coruña: <http://hdl.handle.net/2183/28698>
- Martinez, J. G. (2020). *Revisión sistemática de la literatura sobre el uso de la tecnología en el deporte*. Obtenido de Repositorio de Universidad Católica Santo Toribio de Mogrovejo: <http://hdl.handle.net/20.500.12423/3119>
- MIT App Inventor. (2013). *About Us*. Obtenido de MIT App Inventor: <https://appinventor.mit.edu/about-us>
- Mora Daza, M. A. (2021). *Aplicación móvil basada en Realidad Aumentada como apoyo para fortalecer el aprendizaje de la implementación de Circuitos Electrónicos Básicos en la Protoboard*. Obtenido de Repositorio de Universidad Distrital Francisco José de Caldas : <https://repository.udistrital.edu.co/handle/11349/29165>

- Muhammad Faris, R., & Afandi, A. (2019). Internet of Things (IoT)-based Solution for Real-time Monitoring System in High Jump Sport. *International Journal of Integrated Engineering*, 11(8), 197-205.
- Nestor, A. (18 de Septiembre de 2017). *10 innovaciones tecnológicas que están transformando el béisbol*. Obtenido de Néstor Altuve Blog-Transformación Digital con Impacto en Resultados: <https://nestoraltuve.com/2017/09/18/10-innovaciones-tecnologicas-que-estan-transformando-el-beisbol/>
- Pardo Lozano, L. I. (2021). *rfebs*. Obtenido de Reglas Oficiales de beisbol: <https://www.rfebs.es/reglamento-oficial-juego-beisbol/>
- Pariha, Y. S. (2019). Internet of Things and Nodemcu . *JETIR* , (ISSN-2349-5162).
- Patino, A. (7 de Agosto de 2020). *Compara Software*. Obtenido de Metodología Waterfall: Qué es | Ventajas y Desventajas: <https://blog.comparasoftware.com/metodologia-waterfall/>
- Perera, C. (2014). Sensing as a Service Model for Smart Cities Supported by Internet of Things. *Transactions on Emerging Telecommunications Technology*, 2.
- Peris, J. (25 de Noviembre de 2021). *Service Management Institute*. Obtenido de Taxonomía de los componentes de IoT: <https://news.itsmf.es/taxonomia-de-los-componentes-de-iot/>
- Pernas, J. (14 de Enero de 2021). *Real Federación Española de Béisbol y Sófbol*. Obtenido de <https://www.rfebs.es/>
- Proto Supplies. (2021). *Módulo WiFi ESP8266 D1 Mini V2 ESP-12F*. Obtenido de Proto Supplies: <https://protosupplies.com/product/esp8266-d1-mini-v2-esp-12f-wifi-module/>
- Ramírez Galarza, M. (Octubre de 2022). *Determinación de fuerzas en la “pirueta derecha atrás exterior” obtenidas mediante IOT, como asistencia en tiempo real para arbitraje y futura mejora ergonómica en plantilla del patinador profesional de patinaje artístico*. Obtenido de Universidad Internacional Sek: <http://repositorio.uisek.edu.ec/handle/123456789/4899>

- RFEBS. (2022). *Real Federación Española de Beisbol y Softbol*. Obtenido de Colegio Nacional de Anotadores : <https://www.rfebs.es/>
- Rodríguez González, D. (2013). Arquitectura y Gestión de la IoT. *Revista Telemática*. Vol. 12. No. 3. ISSN 1729-3804, 49-60.
- Rodríguez, E. (16 de Octubre de 2021). *Presente y futuro de los Avances Tecnológicos en el Béisbol*. Obtenido de HITBORICUAL: <https://www.hitboricua.net/post/presente-y-futuro-de-los-avances-tecnol%C3%B3gicos-en-el-b%C3%A9isbol>
- Soloelectrónicos.com. (23 de Noviembre de 2017). *APIs para IoT*. Obtenido de Soloelectrónicos: <https://soloelectronicos.com/2017/11/23/apis-para-iot/>
- Zou, L. H. (2019). Evaluation of a Virtual Reality-based Baseball Batting Training System Using Instantaneous Bat Swing Information. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces*, 1289-1290.

Anexo A

A continuación se expone las distintas secciones del código propuestos, iniciando con la definición de librerías y variables necesarias para el programa.

```
// Programa para Indicador de Umpire
//-----
-----Angie Rivero
//-----LIBRERIAS-----
#define __DEBUG__
#include <ESP8266WiFi.h> //Librería para Wifi de la placa
#include "ThingSpeak.h" //Librería de ThingSpeak.h
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Definir constantes
#define ANCHO_PANTALLA 128 // ancho pantalla OLED
#define ALTO_PANTALLA 64 // alto pantalla OLED

// Objeto de la clase Adafruit_SSD1306
Adafruit_SSD1306 display(ANCHO_PANTALLA, ALTO_PANTALLA, &Wire, -1);

//Datos para conexión de red
char ssid[] = "xxxxxxxxxxxx"; // SSID
char pass[] = "xxxxxxxxxxxx"; // Contraseña
int keyIndex = 0; // índice de clave de red (necesario
solo para WEP)
WiFiClient client;

//Datos del Canal de ThingSpeak
unsigned long myChannelNumber = 1905130;
const char * myWriteAPIKey = "5V2MSK1KN9ZNTW96";

//CONSTANTES PARA DEFINIR PINES
const int pulsador1_asc = 15; // Pin D8 como entrada STRIKE
const int pulsador2_asc = 13; // Pin D7 como entrada BALL
const int pulsador3_asc = 12; // Pin D6 como entrada OUTS
const int pulsador4_asc = 14; // Pin D5 como entrada INNING
const int led_rojo = 0; // Pin D3 como salida LED ROJO
const int led_verde = 2; // Pin D4 como salida LED VERDE

//VARIABLES PARA LOS CONTADORES ASCENDENTE
int ContStrike = 0; // Contador para el numero de STRIKE
int estado_pulsador1_asc = 0; // Estado actual del pulsador de STRIKE
```



```

int lastButtonState1_asc = 1;           // Estado anterior del pulsador de
STRIKE
int ContBall = 0;                       // Contador para el numero de BALL
int estado_pulsador2_asc = 0;          // Estado actual del pulsador de BALL
int lastButtonState2_asc = 1;          // Estado anterior del pulsador de BALL
int ContOuts = 0;                       // Contador para el numero de OUTS
int estado_pulsador3_asc = 0;          // Estado actual del pulsador de OUTS
int lastButtonState3_asc = 1;          // Estado anterior del pulsador de OUTS
int ContInning = 1;                     // Contador para el numero de INNING
int estado_pulsador4_asc = 0;          // Estado actual del pulsador de INNING
int lastButtonState4_asc = 1;          // Estado anterior del pulsador de
INNING

```

Seguidamente, se puede observar dentro de la función principal de configuración (void setup), la inicialización de los componentes, como la pantalla OLED, serial, conexión al servidor de Wi-Fi y la asignación de pines como entradas y/o salidas.

```

void setup() {
#ifdef __DEBUG__
  Serial.begin(9600);
  delay(100);
  Serial.println("Iniciando pantalla OLED");
#endif

  // Iniciar pantalla OLED en la dirección 0x3C
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
#ifdef __DEBUG__
    Serial.println("No se encuentra la pantalla OLED");
#endif
    while (true);
  }

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);           // Inicializar ThingSpeak

  //Asignación de entradas y salidas
  pinMode(pulsador1_asc, INPUT_PULLUP); //Resistencia de pullup interna,
  entrada
  pinMode(pulsador2_asc, INPUT_PULLUP); //Resistencia de pullup interna,
  entrada
  pinMode(pulsador3_asc, INPUT_PULLUP); //Resistencia de pullup interna,
  entrada
  pinMode(pulsador4_asc, INPUT_PULLUP); //Resistencia de pullup interna,
  entrada
  pinMode(led_rojo,OUTPUT);           //Pines como salida
  pinMode(led_verde,OUTPUT);         //Pines como salida

```

```

display.clearDisplay();           // Limpiar buffer
display.setTextSize(1);          // Tamaño del texto
display.setTextColor(SSD1306_WHITE); // Color del texto
display.setCursor(10, 0);        // Posición del texto
display.println("Conectando red"); // Escribir texto
display.display();               // Enviar a pantalla

digitalWrite(led_rojo, LOW);
digitalWrite(led_verde, LOW);
} // VOID SETUP

```

A continuación se muestra el código de la función principal (void loop), donde se realiza las 3 acciones principales del funcionamiento del dispositivo, la lógica necesaria para conteo de las variables, la comunicación y publicación de datos a *ThingSpeak*, y la impresión de los datos en tiempo real sobre la pantalla OLED.

```

void loop() {
//----- Conectar o reconectar el wifi :D-----
if(WiFi.status() != WL_CONNECTED){
  Serial.print("Intentando conectarse al SSID: ");
  Serial.println(ssid);
  while(WiFi.status() != WL_CONNECTED){
    WiFi.begin(ssid, pass);
    Serial.print(".");
    delay(5000);
  }
  //Indicadores de conexión
  Serial.println("\nConectado a la red.");
  digitalWrite(led_rojo, HIGH); //encender led rojo

  //Preparando e imprimiendo pantalla OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 0);
  display.println("Conectado a la red.");
  display.display();
  delay(2000);
}
// Seleccionando los campos de ThingSpeak y variables de publicación del
programa
ThingSpeak.setField(1, ContStrike);
ThingSpeak.setField(2, ContBall);
ThingSpeak.setField(3, ContOuts);
ThingSpeak.setField(4, ContInning);

```

```

//Preparando e imprimiendo pantalla OLED
display.clearDisplay();
display.setCursor(0,0);
display.println("Inicia el juego");
display.display();
display.setCursor(0,18);
display.println("STRIKE: ");
display.setCursor(42,18);
display.println(ContStrike);
display.setCursor(0,30);
display.println("BALL: ");
display.setCursor(30,30);
display.println(ContBall);
display.setCursor(0,42);
display.println("OUTS: ");
display.setCursor(30,42);
display.println(ContOuts);
display.setCursor(0,54);
display.println("INNING: ");
display.setCursor(42,54);
display.println(ContInning);
display.display();

//Publicando Datos en ThingSpeak
int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey); // Publico
Conteo en el canal
if(x == 200){
    Serial.println("Publicado en Canal"); //Revisando Pantalla serial para
identificar instante de envío
}
// Contar y Condiciones STRIKE
estado_pulsador1_asc = digitalRead(pulsador1_asc);
if (estado_pulsador1_asc != lastButtonState1_asc) { // si el estado fue
cambiado, incremente el conteo
    if (estado_pulsador1_asc == HIGH) {
        ContStrike++;
        if(ContStrike<=3){
            Serial.print("STRIKE");
            Serial.println(" ");
            Serial.println(ContStrike);
        }
        if (ContStrike==3) {
            delay(2000);
            ContStrike=0;
            Serial.print("Estas KO!");
            Serial.print('\n');
        }
    }
}
}

```

```

}
lastButtonState1_asc = estado_pulsador1_asc;

// Contar y Condiciones BALL
estado_pulsador2_asc = digitalRead(pulsador2_asc);
if (estado_pulsador2_asc != lastButtonState2_asc) { // si el estado fue
cambiado, incremente el conteo
  if (estado_pulsador2_asc == HIGH) {
    ContBall++;
    if(ContBall<=3){
      Serial.print("BALL");
      Serial.println(" ");
      Serial.println(ContBall);
    }
    if (ContBall==4) {
      delay(2000);
      ContBall=0;
      Serial.print("Tienes base por bola");
      Serial.print('\n');
    }
  }
}
lastButtonState2_asc = estado_pulsador2_asc;

// Contar y Condiciones OUTS
estado_pulsador3_asc = digitalRead(pulsador3_asc);
if (estado_pulsador3_asc != lastButtonState3_asc) { // si el estado fue
cambiado, incremente el conteo
  if (estado_pulsador3_asc == HIGH) {
    ContOuts++;
    ContStrike=0;
    ContBall=0;
    if(ContOuts<=3){
      Serial.print("OUT");
      Serial.println(" ");
      Serial.println(ContOuts);
    }
    if (ContOuts==3) {
      delay(2000);
      ContOuts=0;
      Serial.print("Se ha terminado el Inning!");
      Serial.print('\n');
    }
  }
}
lastButtonState3_asc = estado_pulsador3_asc;

// Contar y Condiciones INNING
estado_pulsador4_asc = digitalRead(pulsador4_asc);

```

```
if (estado_pulsador4_asc != lastButtonState4_asc) { // si el estado fue
cambiado, incremente el conteo
  if (estado_pulsador4_asc == HIGH) {
    ContInning++;
    if(ContInning<=9){
      Serial.print("INNING");
      Serial.println(" ");
      Serial.println(ContInning);
      ContOuts=0;
      ContBall=0;
      ContStrike=0;
    }

    if (ContInning==9) {
      delay(2000);
      Serial.print("Se terminó el juego");
      Serial.print('\n');
      display.clearDisplay();
      display.setCursor(5, 32);
      display.println("Se termino el juego");
      display.display();
      ContInning=0;
      int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    }
  }
}
lastButtonState4_asc = estado_pulsador4_asc;
} //lopp
```

Anexo B

De manera adicional, aunque no forme parte del alcance propuesto por el proyecto, se ha realizado un boceto inicial del prototipo diseñado mediante el software KiCad, mostrado en la Figura 41, con la finalidad de proponer una solución y propuesta integral de la solución.

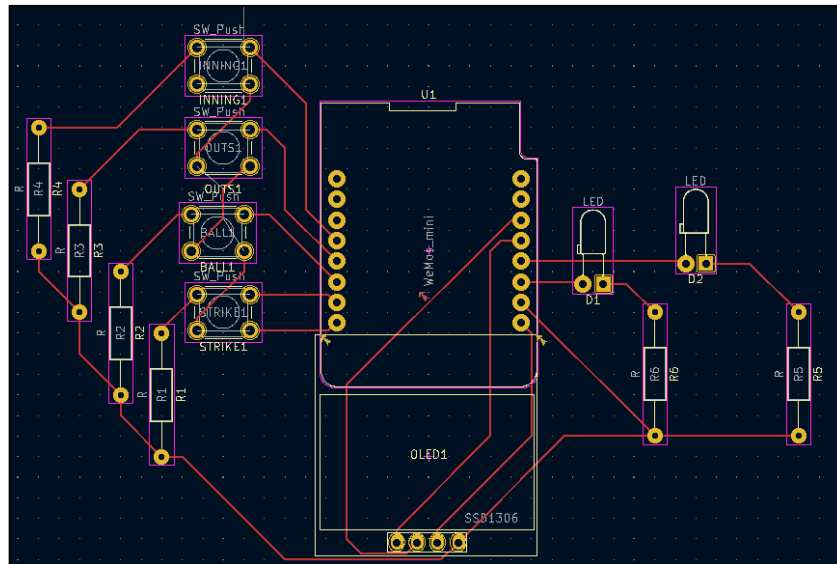


Figura 41. Diseño de PCB del circuito y conexión de componentes electrónicos. Fuente: Elaboración propia.

Asimismo, otra tarea adicional ejecutada ha sido la simulación 3D del diseño propuesto, que se muestra en la Figura 42, con el fin de obtener una imagen semi-realista de la electrónica del prototipo.

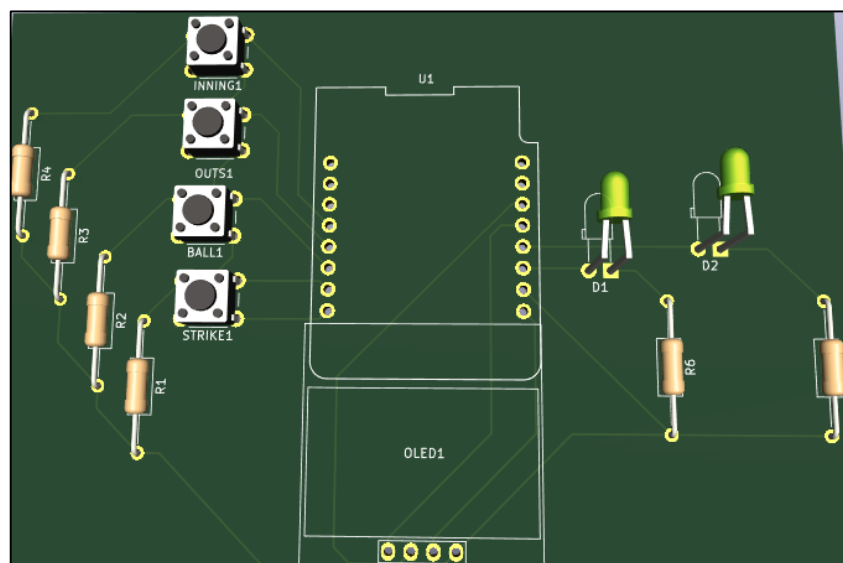


Figura 42. Imagen de Visor 3D del diseño electrónico. Fuente: Elaboración propia.