

INFORME

SEDE: PAYSANDÚ

Proyecto final de la Carrera:



TECNÓLOGO INFORMÁTICO

Nombre del trabajo: Dispositivo de Entrenamiento de Arqueros de Handball (DEAH)

Autores: Dieter Preuss
Matias Cristaldo
Elías Bianchi
Tutores: Sonia Rocha
Marcelo Scotto

Fecha: 03/11/2023

Declaración de autoría

Los integrantes de este proyecto, Dieter Preuss, Elías Bianchi y Matías Cristaldo, dejan constancia mediante esta declaración que el presente trabajo de proyecto final de carrera es puramente de su autoría.

Todas las fuentes de información citadas o usadas se encuentran referenciadas en el documento y se incluyen en la sección de "Referencias Bibliográficas".



Firma:

Aclaración: Elías Bianchi

Cédula de identidad: 4.958.032-8



Firma:

Aclaración: Dieter Preuss

Cédula de identidad: 5.189.153-1



Firma:

Aclaración: Matías Cristaldo

Cédula de identidad: 5.178.720-1

Agradecimientos

Este equipo quiere agradecer al Técnico Deportivo de Handball Rodrigo Fender por sus aportes a la investigación sobre el deporte. A los docentes Jesus Guibert y Maximiliano Fernandez por sus recomendaciones sobre tecnologías de hardware, y al docente Bernardo Firpo por sus sugerencias respecto a tecnologías de software y constante apoyo. A los tutores Sonia Rocha y Marcelo Scotto por la disponibilidad que presentaron durante el transcurso del proyecto. Finalmente, se agradece a Matias Viera por brindarnos esta oportunidad de aplicar y ampliar los conocimientos adquiridos en el transcurso de la carrera.

Resumen de trabajo

Este proyecto surge a petición del representante de la Selección Uruguaya de Handball, Matias Viera, requiriendo una manera de facilitar el entrenamiento de tácticas de los arqueros en la Selección Uruguaya de Handball. El problema se establece en la carencia de una herramienta que ayude a recolectar información y datos en los entrenamientos en base a los tiempos de reacción de los arqueros. Así, se requiere un dispositivo capaz de analizar estas métricas y, en base a las mismas, idear, planificar e implementar nuevos métodos de entrenamiento que ayuden a potenciar las habilidades deportivas de un jugador clave en los equipos de handball, como es el arquero. Esta herramienta estaría acompañada de una aplicación móvil donde se visualizará y administrará la información obtenida.

Para la resolución del problema se presentaron y analizaron distintas propuestas que pudieran brindar una solución acorde a los requerimientos presentados. Estas propuestas fueron modeladas como distintos prototipos de una solución final. Al analizar ventajas y desventajas de cada uno, además de considerar los recursos disponibles para el proyecto, se eligió el prototipo que se definió como el más eficiente para cumplir los objetivos planteados. Este está compuesto por una aplicación móvil que permitirá al usuario crear rutinas (personalizadas o aleatorias), registrar los resultados de los jugadores y visualizar datos previamente registrados. Esta aplicación se conecta con un dispositivo electrónico que se encarga de mostrar las rutinas a los jugadores, utilizando LEDs, y de medir el tiempo de reacción de los mismos. Después de establecer el prototipo, se realizó una investigación exhaustiva sobre las tecnologías que se utilizarían. Estas investigaciones abarcaron el hardware del dispositivo electrónico, el software de la aplicación móvil y el tipo de base de datos que se implementaría.

Si bien el objetivo marcado fue el desarrollo de una solución que pudiera implementarse y llevarse a la práctica, no se logró un dispositivo final completamente ajustado para la experiencia de usuario. Sin embargo, el prototipo entregado cuenta con las funcionalidades propuestas en los objetivos, como recolección, visualización y administración de los datos. Por lo tanto, la mejora y adaptación del hardware para que pueda ser utilizado en entrenamientos reales es la principal optimización a futuro para convertir el prototipo en la solución buscada inicialmente.

A modo de conclusión, teniendo en cuenta el alcance del proyecto y los requerimientos

del cliente, se obtuvo un prototipo que cumple con la mayoría de los requerimientos del cliente, con un camino claro para extensiones y con potenciales funcionalidades a desarrollar a futuro.

Palabras clave: entrenamiento, handball, Arduino, ESP32, bluetooth, aplicación móvil, Android, React Native, Node.js, MongoDB.

Índice

1. Introducción	7
1.1. Objetivo general	9
1.2. Objetivos específicos	9
1.3. Resultados esperados	9
1.4. Estado del arte	10
1.5. Motivación	12
2. Descripción del problema	13
2.1. Justificación del problema	14
2.2. Beneficios de resolver el problema	14
2.3. Alcance	15
3. Métodos de solución	17
3.1. Investigación de hardware	17
3.1.1. Arduino	18
3.1.2. ESP 8266	18
3.1.3. ESP 32	19
3.1.4. Comparación de tecnologías	19
3.1.5. Conclusión del análisis	20
3.2. Investigación de conexiones	20
3.2.1. BLE (Bluetooth Low Energy)	21
3.2.2. Wi-Fi	23
3.2.3. MQTT (Message Queuing Telemetry Transport)	25
3.2.4. Comparación de redes Inalámbricas	26
3.2.5. Conclusión	27
3.3. Soluciones propuestas por el equipo	28
3.3.1. Propuesta N°1	29
3.3.2. Propuesta N°2	30
3.3.3. Propuesta N° 3	32
3.3.4. Propuesta N° 4	33
3.3.5. Conclusión	34
3.4. Prototipo	34
3.5. Investigación de software para aplicación móvil	35
3.5.1. Flutter	36
3.5.2. React Native	36
3.5.3. Comparación de tecnologías	38
3.5.4. Conclusión	39
3.6. Investigación de tecnología de almacenamiento	39
3.6.1. MySQL	40
3.6.2. PostgreSQL	41
3.6.3. MongoDB	42
3.6.4. Firebase	42
3.6.5. Conexión a base de datos	43
3.6.5.1. Node.js	43

3.6.5.2. Deno	44
3.6.5.3. Nim	44
3.6.6. Comparación de bases de datos	44
3.6.7. Conclusión	46
3.7. Almacenamiento local de dispositivo móvil	46
3.7.1. SQLite	46
3.7.2. AsyncStorage	47
3.7.3. Comparación de almacenamientos locales	47
3.7.4. Conclusión	47
3.8. Esquema de integración de tecnologías	48
4. Resultados	49
5. Conclusiones y Trabajo a futuro	51
Glosario de términos	53
Referencias bibliográficas	60
Anexo - Diagrama de ilustración de la base de datos	64

Lista de tablas y figuras

Tabla N°1 comparativa de hardware	20
Tabla N°2 comparativa de las tecnologías de conexiones	26
Diagrama N°1 ilustración de la propuesta N°1	29
Diagrama N°2 ilustración de la propuesta N°2	30
Diagrama N°3 ilustración de la propuesta N°3	32
Diagrama N°4 ilustración de la propuesta N°4	33
Diagrama N°5 Ilustración electrónica del primer prototipo	35
Tabla N°3 comparativa de las tecnologías de desarrollo de aplicación móvil	38
Tabla N°4 comparativa de las bases de datos	45
Tabla N°5 comparativa de las tecnologías de almacenamiento local	47
Esquema N°1 de integración de tecnologías	48

1. Introducción

El handball es un deporte que involucra a dos equipos y se distingue por el uso de las manos para transportar la pelota hacia una meta contraria donde se intenta anotar puntos. El campo de juego es rectangular y presenta una portería (arco) en cada extremo. Cada equipo tiene un portero (arquero) y seis jugadores de campo que intentan marcar goles en el arco contrario. Además dispone de siete suplentes para realizar cambios (Porto y Merino, 2022).

Según lo manifestado por Matías Viera (representante de la Selección Uruguay de Handball y entrenador de arqueros de la misma), este deporte dispone de una organización internacional denominada IHF (International Handball Federation), actualmente conformada por 209 federaciones nacionales de todo el mundo dedicadas a la regulación de las normas competitivas. Fue fundada en 1928 bajo el nombre de IAHF (International Amateur Handball Federation) y es reconocida desde 1936 por el Comité Olímpico Nacional, sede en Uruguay del Comité Olímpico Internacional (JJ. OO.). En el caso de Uruguay, la FUH (Federación Uruguay de Handball) fue fundada en el año 1921 y es la encargada de organizar torneos en las categorías de cadetes (mayores de 16 años), juveniles (mayores de 18 años), juniors (mayores de 23 años) y senior (entre los 33 años), torneos que dependen del número de inscripciones en cada categoría. La Federación se encarga también de fomentar la conformación de la selección nacional que representa al país en competiciones internacionales.

Viera también aportó que el handball en Uruguay es considerado deporte amateur debido a que sus jugadores no reciben una remuneración por practicarlo y su demanda es bastante menor en comparación con el fútbol. Además, debido a su baja popularidad y limitada exposición, su transmisión no es rentable para los principales medios de comunicación como la televisión o la radio.

En un equipo de handball el arquero es uno de los integrantes más importantes. Esta importancia radica en que, de acuerdo a lo manifestado por Viera, el resultado de un juego está directamente relacionado con la defensa, compuesta por los jugadores en la cancha y el porcentaje de eficacia del arquero. Entonces, ante iguales defensas, el punto de desequilibrio son los arqueros y sus habilidades técnico-tácticas, o sea la capacidad del jugador de percibir estímulos del juego y reaccionar de una manera sistemática.

Según lo comentado por el jugador y entrenador de handball Rodrigo Fender, el arquero, -en esa posición específica, con características diferentes al resto de jugadores-, cuenta con necesidades de mayor velocidad de reacción y reacción dirigida. Esto último se refiere a una reacción rápida que en determinado sentido evite un posible gol posicionándose en un lugar que obligue al contrincante a no realizar el lanzamiento o a forzar un lanzamiento hacia afuera del arco. Esto hace que el arquero desde un punto de vista físico tenga una gran exigencia, sumando que debe ser capaz de recibir impactos en varias partes del cuerpo a altas velocidades de pelotas que pesan alrededor de 1/2 kg. que no solamente se desplazan rápido sino con una inercia capaz de empujar al jugador. Por lo tanto, los entrenamientos de este tipo de jugador se enfocan en mejorar todas estas características, realizando por ejemplo ejercicios con pelotas pequeñas más rápidas que las comunes y más dolorosas para generar resistencia al dolor del impacto. Otro ejemplo de entrenamientos comunes en Uruguay es trabajar entre 2 goleros con pelotas con manijas o raquetas forradas colocadas en lugares donde se busca que el jugador realice una acción previa para llegar a posicionarse donde sean colocadas. Todo esto se puede sumar a ejercicios de rayuela y de piernas intentando atajar tiros al arco.

Actualmente, para realizar todos estos ejercicios el entrenador junto con los jugadores analizan jugadas grabadas de partidos para encontrar, según Rodrigo Fender, “zonas oscuras o grises” de los arqueros (referenciando a los lugares a los que estos no alcanzan a cubrir), obteniendo los lanzamientos realizados, desde qué lugares se realizaron, a qué zona del arco fueron dirigidos y cuántos de estos fueron goles o no, para determinar la cantidad de goles recibidos y en qué zonas del arco fueron realizados. Con esta información se busca preparar al jugador en esas zonas en específico.

En la presentación de la problemática, Matías Viera aportó un video proveniente de una práctica en Austria donde se podía ver una herramienta de entrenamiento que consistía en 4 sensores colocados en frente a un arco de handball al que un arquero debía reaccionar al encenderse la luz que correspondía (Handball Goalkeepertraining, 2019). La información obtenida demostraba ser de mucha utilidad para planificar y seleccionar los métodos de entrenamiento para obtener mejores resultados. Sin embargo, para la FUH conseguir una herramienta como esta resultaría un problema ya que debe ser traída desde el exterior y para su mantenimiento se dependería de servicios de reparación independientes del vendedor. Lo anterior no sólo implicaría un costo adicional sino también la posibilidad de que esta no se realice de manera eficiente y ocasione

potenciales daños extra a la herramienta, siendo que quien la repararía no la conocería en profundidad ni estaría familiarizado con un dispositivo desarrollado por otra empresa.

Entonces, la razón por la cual surgió la iniciativa de este proyecto fue buscar otras alternativas para poder contar con este tipo de dispositivos sin las desventajas planteadas.

1.1. Objetivo general

Crear un prototipo de solución tecnológica de fácil mantenimiento para medir la reacción de los arqueros de handball de la Selección Uruguaya durante su entrenamiento, contribuyendo a mejorar así su preparación física y táctica.

1.2. Objetivos específicos

- Diseñar una solución electrónica que permita captar y analizar los tiempos de reacción ante un estímulo al jugador, mediante la utilización de luces y sensores.
- Desarrollar una solución informática para dispositivos móviles amigable y capaz de permitir al usuario generar secuencias arbitrarias o aleatorias que servirán como instrucciones para las luces, indicándoles el tiempo y el orden de encendido.
- Permitir al usuario recolectar y visualizar los datos anteriormente registrados por la solución electrónica para generar una planificación de los entrenamientos.
- Implementar un esquema de base de datos que haga posible que la información se almacene para posterior acceso del usuario.

1.3. Resultados esperados

Se espera desarrollar un prototipo de solución informática que permita a los entrenadores de handball programar rutinas específicas para cada jugador de forma amigable, almacenar datos útiles de estas rutinas, -tales como tiempos de reacción y la asociación de los mismos con cada jugador-, y visualizar esta información en una aplicación móvil para complementar los entrenamientos de jugadores arqueros.

Además, esta aplicación se conectará con una solución electrónica, la cual funcionará con estímulos lumínicos y será la encargada traducir los datos capturados a una forma de fácil visualización y a su vez comprensible para los entrenadores, y enviarlas hacia la aplicación móvil para luego ser almacenadas en la base de datos. Esto trae como ventaja poder llevar un historial de las rutinas y ejercicios realizados por cada jugador,

observando de esta manera su progreso y evolución a través del tiempo, además de la realización de comparaciones entre los distintos arqueros para determinar quién está en un mejor estado físico.

1.4. Estado del arte

Actualmente existen diversas soluciones capaces de satisfacer la mayoría de los requerimientos vistos en el video presentado pero que no cumplen con todos lo que el cliente solicitó, según será explicado a continuación. Se destaca que este dispositivo funcionó como método de visualización de un posible prototipo, ya que estaba compuesto por cajas que contenían luces que se prendían y apagaban arbitrariamente pero no contaba con la funcionalidad de recolección de datos.

Según lo presentado por el cliente en el video y lo que espera de esta herramienta, la solución debe contar como mínimo con 4 dispositivos externos que permitan el registro de los tiempos de reacción. Estos tiempos serán guardados temporalmente en una aplicación móvil, la cual a su vez hará posible la creación de patrones de entrenamiento para los arqueros. Finalmente, esta aplicación almacenará la información en una base de datos externa al dispositivo móvil.

Al investigar antecedentes similares, se encontró la solución Light Training Free, una aplicación para móviles orientada al entrenamiento de reflejos que permite programar rutinas de forma sencilla y rápida que podrían ser adaptadas para preparar arqueros. Esta aplicación es una opción asequible debido a que es gratuita y no cuenta con otros dispositivos externos. Sin embargo, no satisface todos los requerimientos planteados y, adicionalmente, no cuenta con dispositivos externos que se conecten a la aplicación para realizar los entrenamientos (tales como sensores, luces, etc.). Lo que esta aplicación pretende es que el usuario reúna objetos cotidianos (cajas de cartón, pelotas) u objetos utilizados en ciertos tipos de entrenamiento (conos de colores, cuerdas, marcas, etc.), que se diferencien entre sí por su color. De esta forma, el usuario distribuiría arbitrariamente los objetos recolectados dentro del espacio de entrenamiento y la aplicación mostraría por pantalla un color por un tiempo determinado. Entonces, cada vez que la aplicación cambie el color objetivo en pantalla, el usuario deberá interactuar con dicho objeto y finalmente con la aplicación para que se muestre el siguiente objetivo de color. Así, al mostrar varios colores en ciertos patrones una persona sería capaz de entrenar sus reflejos (Light Training: Free, 2018). Sin embargo, el método que se utiliza puede ser considerado inferior a otras tecnologías debido a que es incapaz de registrar

tiempos y no se pueden personalizar los patrones. Otro inconveniente es que esta aplicación no ha sido actualizada desde 2018, lo que presenta potenciales problemas de seguridad o directamente la imposibilidad de funcionar en dispositivos móviles nuevos. Además, la aplicación cuenta con anuncios integrados lo que podría ser molesto a la hora de entrenar al tener que esperar a que finalice el anuncio para continuar el uso.

Otra de las tecnologías más utilizadas y muy recomendadas en el área de entrenamiento de reflejos es Lummic, un sistema de entrenamiento de reacción luminosa y auditiva que consta de pulsadores con luces, los cuales presentan ventosas o imanes que permiten adherirlos a cualquier superficie, y una aplicación móvil gratuita que se conecta mediante bluetooth a los pulsadores. La aplicación, disponible para iOS y Android, hace posible programar los pulsadores con rutinas de juego para niños o entrenamiento para personas mayores, deportes o fitness. A su vez, permite monitorizar el progreso y tiempos de cada nueva sesión, cuenta con patrones de entrenamiento predeterminados (más de 40 preinstalados) y posibilita al usuario editar varios de estos patrones (Lummic, s.f.). En cuanto a las desventajas de esta solución se tienen las siguientes: el idioma oficial de la aplicación es el inglés y las traducciones no son completamente acertadas. Asimismo, se encuentra disponible en el extranjero siendo necesaria la importación para su adquisición, lo que conlleva no solo un costo extra a la hora de su obtención, sino también la falta de un servicio oficial dentro del país que sea capaz de reparar la herramienta en caso de mal funcionamiento o de sufrir una avería.

Además de los productos mencionados anteriormente, también se investigaron los siguientes: Blazepod, FITLIGHT y ReactionX. Sin embargo, los que destacan en el mercado en cuanto a búsqueda y valoraciones son Lummic y Light Training Free. En su mayoría, estas herramientas carecen de la capacidad de recopilar datos debiéndose esto hacer manualmente, lo que los limita en la aplicación de entrenamientos efectivos debido a que no se registra la mejora en los jugadores o la falta de entrenamiento de los mismos. Aquellos productos que sí cuentan con la capacidad de recopilar datos provienen del exterior y para su mantenimiento se tendría que depender de servicios de reparación locales, los cuales serían independientes del vendedor. Estas tecnologías además son cerradas, lo que significa que debido a que su modelo de negocio es vender el producto o publicidad en la aplicación, no pueden ser modificadas para cumplir con las necesidades de la problemática y no servirían como una solución a nivel nacional para la FUH.

1.5. Motivación

La principal motivación del equipo es el aporte e impacto que puede generar esta solución en la práctica del handball de Uruguay que, como se estableció anteriormente, se caracteriza por ser un deporte amateur. Con este proyecto se ayudaría a que la selección uruguaya de handball pueda alcanzar mejores niveles de competitividad y resultados en base a la mejora de las capacidades deportivas de una figura clave en el equipo como lo es el arquero.

Como objetivo personal del equipo, se encuentra el aprendizaje de nuevas tecnologías tanto a nivel de software como de hardware y la investigación e implementación de un prototipo que pueda ser posteriormente llevado a una solución tecnológica completa para satisfacer la problemática presentada, así como el desafío académico que conlleva la realización de este proyecto.

2. Descripción del problema

El principal problema que se presenta por parte de la Selección Uruguaya de Handball es la carencia de herramientas que permitan medir y llevar un control de las aptitudes físicas (en este caso principalmente tiempos de reacción y reflejos, entre otras) de los arqueros. Según se estableció en el estudio del estado del arte, no hay ninguna opción presente en el mercado local e internacional que proporcione funcionalidades que permitan entrenar este tipo de aptitudes y que además sea capaz de llevar un control de las mismas, registrando los resultados obtenidos para su posterior análisis. Esto permitiría a los entrenadores evaluar métricas y realizar un plan de entrenamiento más específico, enfocado en las carencias que presente cada jugador en esta posición.

Para realizar estas prácticas que permitan entrenar a los jugadores de forma más eficiente y relevar las métricas se determinaron ciertos **requerimientos**:

- La solución debe ser un dispositivo electrónico que cuente con 4 o más dispositivos E/R que indiquen de forma visual cuál debe ser encendido y detecten si fueron activados (mediante pulsación o por proximidad), registrando el tiempo desde que se encienden hasta que son activados. Al finalizar la información se debe enviar a una aplicación móvil.
- La aplicación móvil debe poder ser utilizada en entornos con disponibilidad irregular de conectividad a internet y contar con la capacidad de conectarse a los sensores a través de una red local independiente, dado que la conexión a la misma no estará siempre garantizada.
- La aplicación móvil debe permitir generar secuencias que indicarán a los E/R el orden y el tiempo en el que se encenderán (que a pedido del cliente podrán ser de forma aleatoria o definida por el entrenador). Además, desde la aplicación se deben obtener los datos captados por los sensores, visualizando los mismos de forma amigable dentro de la aplicación y, cuando se establezca conexión a internet, los mismos se guardarán en una base de datos externa.
- La solución debe ser controlada a través de una aplicación para dispositivos móviles.
- La solución debe recolectar información que pueda ser utilizada para medir y generar métricas del progreso del jugador entrenado (como por ejemplo cuánto tiempo demora en reaccionar desde que se enciende una señal hasta que el jugador actúa).

2.1. Justificación del problema

La carencia de herramientas que permitan la recolección de métricas que surgen de los entrenamientos específicamente realizados por los arqueros se convierte en un problema. Para que la Selección pueda alcanzar mejores niveles de competitividad, es primordial que los arqueros mejoren individualmente, ya sea física y/o en conocimientos deportivos del handball, dado que el desempeño de los mismos en los partidos es directamente determinante en el resultado final del juego.

Según lo expresado por Matías Viera, el punto desequilibrante en la defensa de cualquier equipo de handball es el arquero. Una mejora en el rendimiento deportivo de este jugador significaría una disminución en la cantidad de goles concedidos en cada partido. Por regla general, en cualquier tipo de torneo, liga o competición, los equipos que consiguen mejores resultados son aquellos que conceden la menor cantidad de goles, siendo esto aún más importante, e incluso determinante, que contar con una ofensiva que convierta muchos goles. Aquí es donde el rol que cumple el arquero se vuelve clave.

Ahora bien, para mejorar los rendimientos y la competitividad de los arqueros se necesita emplear distintos métodos y herramientas. Métodos como entrenamientos para mejorar su aptitud física, posicionamientos en el juego y respuestas a determinadas situaciones de juego; herramientas como conos plásticos, cuerdas, pelotas de diferentes tamaños y otros tipos de elementos utilizados comúnmente en entrenamientos deportivos que se pueden conseguir en el mercado local. Sin embargo, si se pretende llevar un control sobre los tiempos de reacción a determinadas situaciones, realizar una lectura y análisis de métricas asociadas a un jugador específico, -que a su vez permita al entrenador generar nuevas estrategias y rutinas de entrenamiento que ayuden a mejorar y presenten nuevos desafíos-, no se cuenta con una herramienta o dispositivo que pueda cubrir las necesidades del cliente. Esto se debe a que las soluciones disponibles, investigadas anteriormente, no cuentan con funcionalidades que permitan registrar los datos que se necesitan, siendo solo dispositivos que pueden ser utilizados como ayuda para el entrenamiento de reflejos.

2.2. Beneficios de resolver el problema

Una herramienta tecnológica capaz de recolectar métricas en los entrenamientos, permitiendo al usuario organizar la información y relacionarla con los integrantes de la plantilla -en este caso haciendo énfasis en los arqueros-, posibilitaría a la selección

uruguay llevar un control más preciso de los rendimientos de cada jugador, así como comparar los mismos para determinar quién cuenta con un mejor desempeño a lo largo de los entrenamientos.

Adicionalmente, con la herramienta se generarían nuevas formas de estimular los reflejos de quienes se sometan a las pruebas y desafíos presentados por el dispositivo. Esto ampliaría la gama de opciones disponibles para los entrenadores a la hora de elegir ejercicios de entrenamiento, haciendo que los mismos no se conviertan en monótonos y manteniendo al jugador expectante por nuevos desafíos que lo obliguen a salir de su zona de confort. Esto derivaría potencialmente en el crecimiento de las habilidades deportivas y capacidades de reacción de una pieza clave en el equipo, siendo que un aumento en su nivel de competencia significaría un aumento en las probabilidades de que los remates al arco del equipo contrario no se conviertan en gol. Así, se incrementarían las posibilidades de que la Selección Uruguay sea más competitiva frente a selecciones de mayor jerarquía.

Debe mencionarse también que, si bien el objetivo final es alcanzar un prototipo funcional y escalable, la solución que se obtendría en los trabajos a futuro de este proyecto estaría pensada a medida de los requerimientos para solventar el problema, pudiéndose llevar a cabo un registro de las métricas y al mismo tiempo ofrecer nuevos métodos de entrenamiento y desafíos para los arqueros dentro de una misma solución.

2.3. Alcance

Teniendo ya definidos los objetivos y qué es lo que se busca solucionar, es necesario considerar los recursos con que se cuenta para el desarrollo de este proyecto. En ese sentido, se identificaron tres principales áreas necesarias para el funcionamiento de la herramienta: microcontrolador, aplicación móvil y API.

Considerando estos tres componentes y lo solicitado por el cliente, se busca que esta herramienta cuente con las siguientes características:

- Creación de rutinas (aleatorias o definidas por un entrenador).
- Listado de rutinas creadas.
- Registro de jugadores.
- Listado de jugadores.
- Conexión y desconexión entre Microcontrolador y Aplicación móvil.

- Visualización de estados de la herramienta (está conectada, se está ejecutando una rutina, etc.).
- Envío y recepción de datos entre los componentes.

En el componente Microcontrolador se encontrará todo lo relacionado con los cuatro dispositivos electrónicos que serán utilizados para la interacción directa del jugador que esté entrenando, tanto para indicarle cuándo y con cuál debe interactuar, como también para procesar la información y enviarla a través de una conexión inalámbrica a la aplicación móvil. Teniendo en cuenta el alcance y los recursos con los que el equipo cuenta para este proyecto (en este caso solo se tiene un microcontrolador y se necesitaría más tiempo para diseñar y crear todo lo relacionado a la protección de los componentes de hardware), se decidió por el desarrollo de un prototipo que pueda ser fácilmente ampliado hasta completar las funcionalidades restantes.

Por otra parte, se desarrollará la aplicación móvil para interactuar con el usuario (entrenador), donde se pueda administrar el Microcontrolador enviando las rutinas que deba ejecutar, mostrar la información de manera que resulte fácilmente comprensible y guardar esta información de forma tanto local como remota.

Por último, la API refiere a un sistema que actuará como una interfaz intermedia entre la aplicación móvil y el servidor de la base de datos. Esta tecnología permite ejecutar las modificaciones deseadas sobre la base de datos, manteniendo seguros estos datos en relación a cómo se accede a ellos y cómo se utilizan.

3. Métodos de solución

Tras determinar que las soluciones encontradas no se adaptan completamente a las necesidades del proyecto, y a partir del análisis de un ejemplo aportado por el cliente, se comienza a determinar el diseño de la solución.

Considerando los requerimientos del proyecto, la solución pensada consiste en dos partes: por un lado dispositivos emisores/receptores (E/R) y por otro una aplicación móvil. Teniendo en cuenta esto fue necesario investigar diferentes tipos de tecnologías que permitieran desarrollar lo planteado, abarcando hardware, conexiones, software para el desarrollo de la aplicación y bases de datos. Por lo tanto, se investigaron microcontroladores y cuál sería el más conveniente para usar en este proyecto, qué tipo de conexión tendría el microcontrolador con el celular hacia la aplicación móvil, en qué tecnología se desarrollaría la aplicación para celular para que pudiera ser utilizada en distintos tipos de dispositivos y en qué tecnología se desarrollaría la base de datos.

Luego de finalizada la investigación se desarrolló un prototipo que permitiese visualizar las características planteadas para una posible solución tecnológica al problema.

3.1. Investigación de hardware

Al momento de seleccionar el hardware a utilizar para el desarrollo de un primer prototipo, se realizó una investigación sobre microcontroladores programables que permitieran procesar mensajes recibidos y enviarlos a través de medios de comunicación inalámbrica. Además, se buscaba un microcontrolador que tuviera documentación suficiente y una comunidad activa para realizar consultas.

Considerando la amplia gama de microprocesadores en el mercado y que fue recomendado por el docente Jesús Guibert, se optó por investigar la línea de microprocesadores ESP de la empresa Espressif System, establecida en 2008 en Shanghái, China. Así, se descubrió que estos microprocesadores ESP cuentan con la capacidad de funcionar bajo la plataforma de Arduino (además de su propia plataforma Espressif IDF), que su potencia de procesamiento es mayor que cualquier microprocesador de Arduino y que cuenta con módulos de conexión inalámbrica integradas (Espressif, s.f.). Los microcontroladores de Espressif que se decidió investigar para la utilización de este proyecto fueron el microcontrolador ESP8266 y el

microcontrolador ESP32, debido a su popularidad entre la comunidad de desarrolladores y a sus características que serán expuestas en los siguientes subcapítulos.

3.1.1. Arduino

Arduino es una plataforma con licencia de libre uso de desarrollo para componentes electrónicos que ofrece componentes de bajo costo modulares y software, por lo cual el hardware es sencillo de modificar a fin de añadir nuevos módulos para funcionalidades futuras. Esta tecnología fue diseñada por la empresa Arduino CO. para la creación de proyectos electrónicos, buscando reducir el tiempo y esfuerzo que implica desarrollar un prototipo electrónico, con las características que los microcontroladores de Arduino y todo su ecosistema de módulos ofrece (Arduino, 2021).

Bajo esta plataforma existen diferentes alternativas para el desarrollo, por ejemplo los ofrecidos por la propia empresa Arduino, que abarcan una amplia gama de microcontroladores con diferentes propósitos, además de sus propios módulos contenidos en los llamados Kits de desarrollo.

Otras opciones de microcontroladores que utilizan la plataforma de Arduino (debido a su licencia de libre uso) presentan opciones más variadas en componentes y precios. Ejemplos de esto son los ESP, denominación de una familia de chips SoC. A continuación se presentarán los resultados de su investigación.

3.1.2. ESP 8266

El ESP 8266 es un microcontrolador programable con capacidades Wi-Fi de bajo costo (entre U\$S 8 y U\$S 12 actualmente) lanzado en el 2014. Cuenta con un stack TCP/IP completo y un núcleo lógico fabricado por Espressif Systems. Además, presenta una comunidad extensa y una gran cantidad de información para aprender a utilizarlo de forma rápida y sencilla.

Como punto principal de este microcontrolador se puede mencionar que cuenta con una considerable capacidad de procesamiento, soporte integrado para el uso de redes wifi y a su vez conlleva un bajo consumo de energía. Este hardware cuenta con un tamaño reducido (es decir que puede usarse para dispositivos con espacio libre limitado) y una suficiente cantidad de pines como para conectar hardware adicional como periféricos. Sin embargo, el ESP8266 carece de soporte para una amplia variedad de periféricos como

son la interfaz ethernet, sensores táctiles y sensores de temperatura (del Valle Hernández, 2017).

3.1.3. ESP 32

ESP 32 es una denominación para un conjunto de microcontroladores lanzado al mercado en enero del 2016. Son programables, de bajo costo (entre U\$S 5 y U\$S 15 actualmente) y consumo de energía, contando con tecnologías Wi-Fi y Bluetooth de manera integrada.

Está provisto de un microcontrolador Tensilica Xtensa LX6 en los modelos simple y doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía. Fue creado y desarrollado por Espressif Systems y es fabricado por TSMC. Es un sucesor de otro microcontrolador, el ESP 8266.

Este microcontrolador cuenta con 40 pines de conexión lo que permite la utilización de muchos periféricos, -siendo compatible con muchos periféricos de ethernet, de audio, para tarjetas SD, etc.-, y cuenta con un módulo de hardware de generación de números aleatorios. Un inconveniente del hardware del ESP32 es que al ser tan reciente no todas sus funcionalidades se encuentran desarrolladas en el software de Arduino por lo que se tiene que recurrir a librerías para poder usarlas. Por otro lado, el ESP32 tiene un precio mayor a otros microcontroladores, como puede ser el ESP8266 (Espressif Systems, 2023).

3.1.4. Comparación de tecnologías

En base a los requerimientos del proyecto se propusieron los siguientes criterios para evaluar las tecnologías, considerando también las características, ventajas y desventajas de cada una: capacidad de procesamiento del microprocesador elegido (debe ser capaz de generar aleatoriedad, necesaria para las rutinas), conexión tanto por bluetooth como Wi-Fi para comunicarse con el dispositivo móvil, aplicación de protocolos seguros de manejos de datos, administración de la energía necesaria para alimentar todos los sistemas que se adicionen y espacio de memoria Ram como para ejecutar el código que analizará los inputs del jugador.

Criterio	ESP32	ESP8266
CPU	Xtensa LX6 de doble núcleo de 32 bits, operando a 160 o 240 MHz.	Risc de 32-bits y 80Mhz.
RAM	320 KB de RAM dinámica y 200 KB de RAM de instrucción.	96 KB de RAM dinámica y 64 KB de Ram de instrucción.
Conexiones inalámbricas	<ul style="list-style-type: none"> • Wi-Fi: 802.11 b/g/n. • Bluetooth: v4.2 BR/EDR y BLE. 	<ul style="list-style-type: none"> • Wi-Fi: 802.11 b/g/n.
Cantidad de pines	38 pines GPIO (Entradas/Salidas de propósito general).	16 pines GPIO (Entradas/Salidas de propósito general).
Seguridad	Soporta todas las características de seguridad estándar de IEEE 802.11, incluyendo WPA, WPA/WPA2 y WAPI.	Soporta todas las características de seguridad estándar de IEEE 802.11, incluyendo WPA, WPA/WPA2 y WAPI.
Administración de energía	Se puede alimentar a 3,3v (3.0-3,6v) o 5v, pero los pines GPIO trabajan a 3,3v.	Se alimenta con 3.3 V, pero incluye un regulador de tensión, lo que permite alimentarla por USB con 5V.

Tabla N°1 comparativa de hardware

3.1.5. Conclusión del análisis

El ESP 32 demuestra una capacidad de procesamiento mayor y una mayor cantidad de memoria RAM, una mayor cantidad de pines. Además, cuenta con la capacidad de generación de números pseudo-aleatorios, módulos de conexión wifi y bluetooth integrados, compatibilidad con protocolos de seguridad de datos y una compatibilidad con periféricos mayor que la del microcontrolador ESP8266, por lo que se decidió su utilización.

3.2. Investigación de conexiones

Para la solución electrónica será necesario disponer de un método por el cual los E/R sean capaces de comunicarse con la aplicación móvil de manera inalámbrica, usando esta conexión para seleccionar las secuencias y registrar los tiempos de los usuarios. Por lo tanto, fue necesario investigar si las tecnologías popularmente utilizadas serían compatibles. Las originalmente discutidas fueron el protocolo Bluetooth y las

comunicaciones utilizando una red Wi-Fi, lo cual implica utilizar HTTP Request, tecnologías consideradas debido a su velocidad de transferencia así como por el alcance y los tiempos del proyecto explicados en el capítulo 2.

3.2.1. BLE (Bluetooth Low Energy)

Bluetooth Low Energy hace referencia a la versión 4.0 del protocolo Bluetooth, tecnología de comunicación inalámbrica que puede ser usada para la transmisión de datos de corto alcance de un dispositivo digital a otro; funciona utilizando ondas de radio de banda de 2.4GHz para transmitir datos a través de una conexión segura. Es de destacar que en esta versión los desarrolladores lograron realizar comunicaciones inalámbricas con un consumo de energía considerablemente reducido en comparación con las versiones anteriores del producto.

La tecnología de Bluetooth Low Energy (BLE) trae consigo una serie de características que la hacen valiosa en diversos contextos de comunicación. Una de las más destacadas es su eficiencia energética: los dispositivos que utilizan BLE consumen energía sólo durante las comunicaciones activas, lo que permite mantener una conexión constante sin agotar rápidamente la batería. Sumado a esto, la mayoría de los dispositivos modernos son compatibles con BLE sin necesidad de configuraciones complejas, lo que acelera la adopción y el uso de la tecnología. Ofrece también comunicaciones a alta velocidad, agilizando la transferencia de datos entre dispositivos y tiene la capacidad de conectar varios dispositivos al mismo tiempo, lo que facilita la creación de redes interconectadas.

Por otra parte, esta tecnología se destaca por su capacidad para transmitir datos de manera eficiente utilizando un ancho de banda menor en comparación con otros métodos. Esto no sólo mejora la velocidad sino que también optimiza la utilización de la red. En términos de seguridad, BLE ofrece autenticación y cifrado robusto en sus comunicaciones, lo que garantiza que los datos transmitidos permanezcan privados y seguros.

No obstante, es importante tener en cuenta algunas limitaciones. Por un lado, el rango de alcance de BLE es moderado (alrededor de 10-20 metros), lo que podría ser una consideración en el marco de este proyecto en entornos de entrenamiento más grandes, dado que este rango puede limitar el área posible de ejercicio. Otra posible preocupación es la interferencia con tecnologías NFC, lo que podría producir un fallo en el

funcionamiento de los dispositivos que trabajen en el mismo radio de alcance. Sin embargo, esto es poco probable debido a que utilizan distintas frecuencias y protocolos.

En resumen, Bluetooth Low Energy (BLE) presenta un conjunto de características valiosas, como su eficiencia energética, velocidad de comunicación, facilidad de uso y seguridad en las transmisiones. Sin embargo, es importante considerar su alcance limitado y la posible incompatibilidad con ciertos dispositivos. Estas características definen la utilidad de BLE en diversas aplicaciones y entornos tecnológicos, cuyo ejemplo más relevante sería el de los relojes inteligentes usados por los corredores, capaces de registrar una cantidad considerable de datos de forma rápida, precisa y sin hacer uso excesivo de la batería (Bluetooth, 2023).

3.2.1.1. Cantidad de conexiones Bluetooth

Después de realizada la investigación sobre la tecnología Bluetooth se indagó si esta soportaría un potencial aumento de conexiones de E/R sobrepasando los 4 propuestos. Esto se tomó en cuenta debido a dos razones: en primer lugar porque algunas opciones de mercado (en este caso productos utilizados para el entrenamiento de reflejos) ofrecen entre 4, 6 y 8 dispositivos que emiten luces y detectan pulsaciones; en segundo lugar porque no se contaban con los conocimientos necesarios para asegurar que se pudiera aumentar la cantidad de conexiones. La investigación que se realizó en este punto fue focalizada a la cantidad de conexiones Bluetooth que se pueden tener simultáneamente en un dispositivo móvil, en este caso celulares, y que cada uno de los nodos conectados pudiera transmitir y recibir información desde el celular.

En este sentido, con la versión 5.0 de Bluetooth lanzada en 2016 el número de dispositivos que se pueden conectar simultáneamente al celular aumentó hasta un máximo de siete conexiones. Sin embargo, la cantidad de conexiones que permite cada dispositivo móvil varía, dependiendo no sólo de la versión de Bluetooth que posea, sino también de la versión del sistema operativo que se tenga instalada (Android, iOS, etc), así como de las prestaciones de hardware de cada dispositivo. En algunos casos, aunque se tengan las versiones de Bluetooth y del sistema operativo adecuadas, el dispositivo puede que acote el número de conexiones a una cantidad menor dependiendo de sus prestaciones. Tomando como ejemplo el caso de Android, recién a partir de la versión 10.0 de este sistema operativo es que se pueden realizar hasta 7 conexiones Bluetooth (OST Team, 2022).

A modo de conclusión se debe tener en consideración que, si bien es factible desarrollar una solución en la cual se conecten varios microcontroladores al celular por medio de Bluetooth, se entiende que el número de conexiones siempre dependerá del celular que el usuario posea. Por lo tanto, existen posibilidades de que ciertos usuarios no puedan utilizar esta solución al no contar con las prestaciones necesarias en sus equipos.

3.2.2. Wi-Fi

La tecnología Wi-Fi refiere a una red inalámbrica que utiliza señales de radiofrecuencia para conectar distintos dispositivos a una red, ya sea para comunicarse entre ellos o para acceder a internet. Debido a que es posible generar una intrared Wi-Fi independiente (red local) a conexiones externas si se utiliza el ESP32 (que cuenta con todo lo necesario incluido) o una tarjeta de red, se da la posibilidad de la utilización de conexiones Wi-Fi entre un microcontrolador y el dispositivo móvil, manteniendo así los requerimientos del proyecto.

Profundizando en la comunicación mediante el uso de HTTP Request a través de Wi-Fi, para entender este comando es necesario explicar el concepto de HTTP, sigla de *Hypertext Transfer Protocol*, lo que se traduce como Protocolo de Transferencia de Hipertextos. Este protocolo permite realizar peticiones de datos y recursos a lo que se conoce como un servidor, quien gestiona estas peticiones y da una respuesta hacia quien haya realizado dicha petición, conocido como cliente. Esto es lo que se denomina como el principio de cliente-servidor. HTTP Request se refiere, entonces, a un comando que permite enviar todo tipos de peticiones a una dirección web en específico y posteriormente procesar la respuesta obtenida por parte del servidor.

Otra opción diferente a HTTP Request muy utilizada en comunicación sincronizada y disponible en redes Wi-Fi, es la comunicación por medio de un *WebSocket*. Este concepto se refiere a un protocolo de red que, haciendo uso de TCP (Protocolo de Control de Transmisión, utilizado para transmitir datos en forma de “paquetes” a través de redes), establece de qué manera deben intercambiarse los datos. Las conexiones realizadas mediante TCP se producen entre dos puntos finales de comunicación, también llamados “sockets”. La comunicación a través de los WebSockets se realiza de manera bidireccional, enviando así datos desde ambas direcciones al mismo tiempo, lo que genera una ventaja con respecto a la velocidad con la que se accede a dichos paquetes de datos (Salazar, s.f.).

3.2.2.1. Investigación de compatibilidad de HTTP Request y microcontroladores

Para hacer uso de los protocolos HTTP y WebSockets en un chip basado en la plataforma de Arduino es necesario utilizar librerías que provean la lógica para las conexiones HTTP Request. Por lo tanto, se investigaron las librerías más relevantes considerando la amplia experiencia de la comunidad que podría llegar a resolver problemas que surjan en su uso.

3.2.2.1.1. WebServer.h

Es una librería de software libre desarrollada por Ivan Grokhotkov que permite el manejo de peticiones (HTTP Request) del tipo GET y POST (obtener datos del servidor mediante peticiones y enviar datos hacia un recurso para ser modificados). Ha sido especialmente modificada y optimizada para realizar aplicaciones utilizando el microcontrolador ESP32. Como ventajas de esta librería se cuentan que es muy simple de utilizar, genérica, de uso libre y bastante difundida en la comunidad (Grokhotkov, s.f.).

3.2.2.1.2. ESPAsyncWebServer

Es una librería de código abierto desarrollada por Ivan Kravets que soporta comunicaciones del tipo asincrónicas (pudiéndose manejar más de una conexión al mismo tiempo) a la vez que permite el uso de WebSockets. Esta librería fue creada específicamente para realizar aplicaciones utilizando microcontroladores del tipo ESP y cuenta con funcionalidades tales como velocidad de transferencia de datos mayor respecto a otras librerías, flexibilidad para aplicar en distintos contextos y procesador de plantillas de diseño simple permitiendo el uso de las mismas. Como desventaja, se encuentra que no siempre es compatible con otros tipos de microcontroladores, por lo que de utilizarse con otro modelo de microcontrolador no resultaría muy útil (Kravets, s.f.).

3.2.2.1.3. Conclusión del análisis

Al investigarse estas dos librerías que permiten hacer uso de conexiones HTTP (WebServer.h y ESPAsyncWebServer.h) se determinó que no existen grandes diferencias entre las mismas y que, debido a que ambas son compatibles con el tipo de microcontrolador programable a ser utilizado en el desarrollo de la solución bajo la plataforma de Arduino, cualquiera de estos protocolos podría ser implementado. Sin embargo se opta por ESPAsyncWebServer debido a la especificidad con el microcontrolador elegido ESP32.

3.2.3. MQTT (Message Queuing Telemetry Transport)

Durante la investigación de conexiones posibles se encontró la existencia del protocolo MQTT, que permite una transferencia de datos simple y fue específicamente creado para dispositivos con un bajo ancho de banda para sus comunicaciones. Por lo tanto, se decidió investigarlo como una posible tercera opción.

Este tipo de tecnología se puede encontrar asociada a dispositivos IoT (Internet of Things, por su traducción “el internet de las cosas”), que refiere a dispositivos cotidianos que puedan acceder a internet sin necesidad del accionar humano.

MQTT trabaja con un sistema de “publicado” y “suscripción”, en el cual dispositivos que funcionan como servidor MQTT publican mensajes sobre un tópico en específico y todos los dispositivos que funcionan como clientes MQTT y se encuentren suscritos a ese tópico reciben un mensaje. Los mensajes enviados pueden ser comandos, instrucciones o datos de lectura (recibidos por periféricos asociados a los dispositivos), mientras que los tópicos son cadenas de caracteres separadas por una barra inclinada (/), en la que cada barra divide los niveles de los tópicos. La aplicación de esta tecnología se puede ilustrar con el ejemplo de encender lámparas en una habitación de una casa mediante conexión a internet. Todas las lámparas deberían funcionar como clientes MQTT suscritas a un tópico “casa/lámpara” y un microcontrolador como servidor MQTT enviaría señales de encendido y apagado a ese tópico.

Otro punto a tener en cuenta sobre MQTT es que necesita de MQTT broker (sistema de Back-End que coordina los mensajes entre los diferentes clientes), encargado de administrar los mensajes enviados, es decir de recibir los mensajes, filtrarlos, decidir a quién va dirigido cada mensaje y por último publicar el mismo para todos los dispositivos suscritos (HiveMQ, s.f.).

Las principales ventajas que conlleva este sistema son una curva de aprendizaje baja y una alta confiabilidad en cuanto a la entrega de mensajes, debido que cuenta con un servicio que verifica el envío y recepción de mensajes. Según la comunidad de desarrolladores que utilizan este protocolo, éste prueba ser útil a la hora de recibir mensajes duplicados y es eficiente y de bajo consumo, por lo que no es necesaria una gran cantidad de energía o de procesamiento. Al comparar este sistema con el protocolo HTTP se tiene que puede ser más eficiente en cuanto a velocidad de transmisión y en

cuanto a uso de energía; además este sistema es más útil al momento de recibir y manejar mensajes duplicados.

El inconveniente que presenta este broker es que debe estar encendido continuamente para mantener la comunicación activa entre los dispositivos, por lo que usualmente se debe hacer uso de una placa Raspberry Pi para que hostee esta parte de la comunicación o instalar este broker en la nube, lo que requeriría el uso de un servicio de hosteo que debe encargarse de mantener este servicio activo en la nube. Además, este sistema no cuenta con una comunicación bidireccional integrada, por lo que no habría manera de saber si el mensaje enviado fue recibido correctamente por el/los dispositivos objetivos.

Finalmente, esta tecnología cuenta con baja velocidad de transmisión y latencia en comunicaciones, debido a que los mensajes deben pasar por la nube para llegar al dispositivo objetivo, y carece de seguridad integrada, lo cual puede ser una desventaja, sobre todo teniendo en cuenta el posible contenido de los mensajes (descubrearduino.com, s.f.).

3.2.4. Comparación de redes Inalámbricas

Luego de analizar las tecnologías presentadas, y teniendo en cuenta los requerimientos del proyecto, se analizarán las características, ventajas y desventajas de cada una. Para eso se tendrán en cuenta los siguientes criterios: el límite (tamaño) de los datos enviados por cada tecnología en cada comunicación entre emisor/receptor, la velocidad de transmisión de dichos datos, el rango de alcance máximo para establecer comunicaciones entre los nodos y la seguridad de los datos enviados por cada uno.

Características	BLE	Wi-Fi (HTTP Request)	Wi-Fi (WebSocket)	MQTT
Límite de envío de datos	64 Bytes.	Método GET: 2 KB. Método POST: Sin límites (depende del entorno en el que se esté trabajando).	8192 Bytes.	256 MB.
Velocidad de transmisión de	2 Mb/s.	Depende de la velocidad definida	Depende de la velocidad	70/80 Mb/s.

Características	BLE	Wi-Fi (HTTP Request)	Wi-Fi (WebSocket)	MQTT
datos		por el proveedor del servicio. Normalmente en Uruguay es de 250/40 Mb/s.	definida por el proveedor del servicio. Normalmente en Uruguay es de 250/40 Mb/s. Si se usa el protocolo UDP, se puede obtener una mejora en la velocidad respecto al uso de TCP.	
Alcance	<100 metros.	20/150 metros.	20/150 metros.	Al hacer uso de Wi-Fi para sus conexiones hereda el rango de dicha red.
Seguridad de los datos	Cifrado de 128 bits y autenticación. Es necesario un emparejamiento de los dispositivos la primera vez que se realiza una conexión entre ellos.	Se pueden cifrar los datos mediante WEP, WPA, WPA2 o WPA3. Requiere la contraseña de la red para acceder a los datos.	Se pueden cifrar los datos mediante WEP, WPA, WPA2 o WPA3. Requiere la contraseña de la red para acceder a los datos.	Mediante el protocolo SSL MQTT identifica y autentica a sus clientes. Además los servidores de MQTT proporcionan métodos de autorización para administrar aspectos del servidor.

Tabla N°2 comparativa de las tecnologías de conexiones

3.2.5. Conclusión

Se decidió por la utilización del protocolo Bluetooth (BLE) para la conexión entre el dispositivo y la aplicación móvil. Al momento de comunicarse el microcontrolador con la aplicación móvil y viceversa, la tecnología Bluetooth ofrece una mayor eficiencia en

cuanto a costo de energía permitiendo una mayor autonomía y, si bien las demás tecnologías ofrecen una mayor velocidad en la transferencia de datos, en este contexto no es relevante ya que por las características de la información enviada, la velocidad ofrecida por BLE es suficiente. Además, el factor de que en los espacios de entrenamiento de la selección no se cuenta con una conexión a Wi-Fi influye directamente en la decisión de implementar BLE (siendo que MQTT requiere una conexión a Wi-Fi), ya que de lo contrario se tendría que generar una red Wi-Fi desde el microcontrolador y esto requeriría el consumo de más recursos por parte del chip, algo que no es necesario utilizando la tecnología de BLE.

3.3. Soluciones propuestas por el equipo

Tomando como base el ejemplo de dispositivo electrónico proporcionado por el cliente, el equipo planteó varias posibles soluciones electrónicas teniendo en cuenta la cantidad de microcontroladores a utilizar y las posibles comunicaciones entre estos controladores y los E/R, que podrían ser cableadas o inalámbricas.

Estas posibles soluciones cumplen los requerimientos descritos en el capítulo anterior y cada propuesta resuelve de una distinta forma posible la distribución, los componentes y las conexiones a utilizar. Una vez planteadas, se realizó este ejercicio de analizarlas para contemplar las distintas ventajas y desventajas presentes en cada una de ellas. Por ejemplo, las propuestas que cuentan con más de un microprocesador presentarían una mejor distribución de los datos a procesar (no sobrecargando a uno en particular). Lo mismo sucede con las distintas maneras de conectar los componentes entre ellos, ya que por ejemplo las opciones que se conectan mediante cables poseerán una mayor velocidad de comunicación de datos pero carecen de la flexibilidad de distribución en el espacio que proveen las opciones inalámbricas.

El funcionamiento general de la solución es el siguiente: desde la aplicación del dispositivo móvil conectado al microprocesador vía Bluetooth se enviarán los datos de las secuencias a ejecutarse, que serán interpretadas por el microcontrolador, el cual ordenará a los LED que se enciendan siguiendo dicha secuencia. A su vez, esta aplicación estará conectada a una base de datos en la cual se relacionarán los datos capturados por los sensores del microprocesador (una vez se inicie el entrenamiento) y serán vinculados con uno de los jugadores previamente registrados en dicha base. Cuando se tenga conexión a internet, la aplicación guardará la información para un posterior análisis, pudiendo ser visualizada desde la interfaz de la misma aplicación móvil.

3.3.1. Propuesta N°1

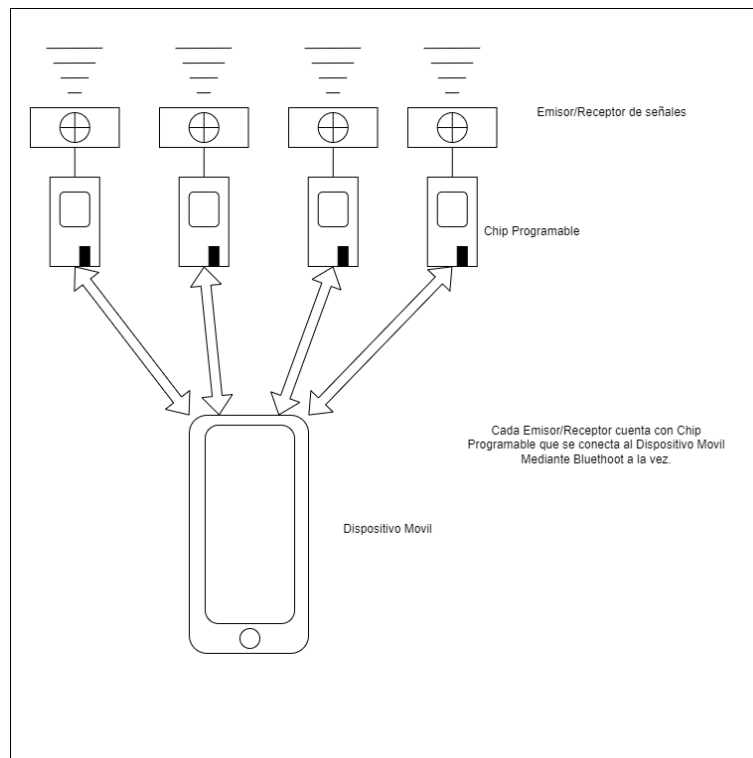


Diagrama N°1 - Ilustración de la propuesta N°1

En esta solución se hace uso de 4 o más E/R que consisten en un emisor LED que indicaría cuál es el receptor que estará esperando ser activado, un receptor que se encargará de detectar una interacción, una fuente de energía (batería AAA) y un microcontrolador programable por cada uno de los E/R.

En este caso se utilizará el microcontrolador que estará pendiente de si uno de los botones fue pulsado en el momento correcto (cuando se enciende su luz) y para la conexión a la aplicación móvil que le indicará si encender la luces o no. El entrenador generará el patrón (tanto personalizado como aleatorio) en la aplicación móvil, que además procesaría los datos que reciba de cada microcontrolador, presentaría la información de forma amigable y la almacenaría en una base de datos online.

En el caso de que el patrón sea personalizado, la aplicación le pediría al entrenador el orden de los LEDs a encender y el tiempo de duración del encendido. Si el patrón es aleatorio, los valores introducidos por el entrenador serían generados automáticamente por el software.

La principal ventaja que conlleva esta propuesta es el tipo de conexión entre los E/R y el celular, ya que se trata de una conexión directa. Esto implica que la aplicación se comunicaría directamente a los microcontroladores lo que podría suponer una mayor velocidad de comunicación.

Sin embargo se encuentran desventajas. Por un lado, por cada E/R que se agregue será necesario adquirir un microcontrolador programable extra de los que ya se tienen. Esto podría resolverse consiguiendo una opción más económica que un arduino común (Arduino Uno, que actualmente estaría rondando entre los U\$S 25 y los U\$S 30 en las tiendas más relevantes en el Uruguay que ofrecen esta tecnología).

Por otra parte, el funcionamiento del sistema sería dependiente del dispositivo móvil del usuario, quien realizaría el procesamiento lógico. Esto en algunos casos puede resultar en demasiada demanda de procesamiento para dispositivos móviles más antiguos y causar un funcionamiento más lento del sistema ya que debe utilizar más recursos.

3.3.2. Propuesta N°2

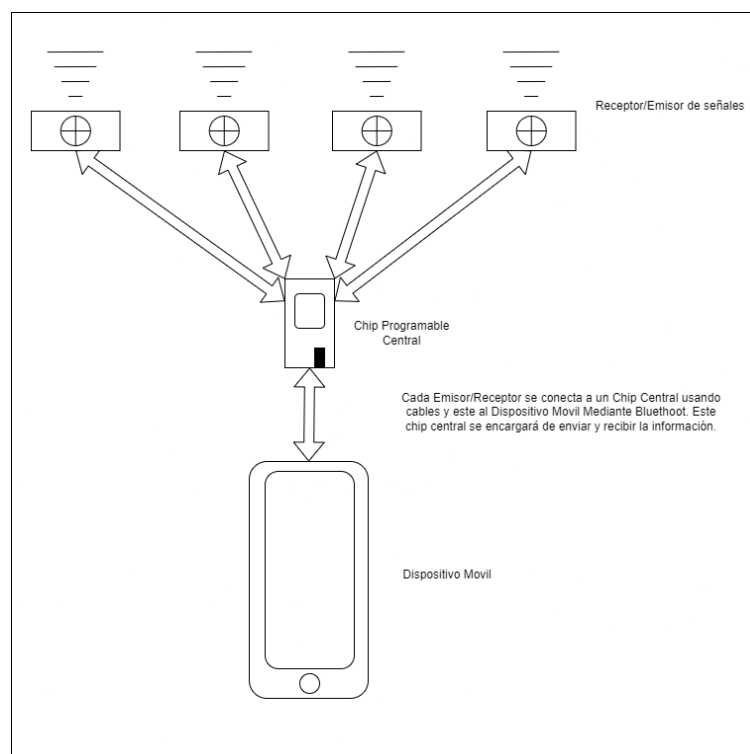


Diagrama N°2 - Ilustración de la propuesta N°2

En esta solución se hace uso de 4 o más E/R que consisten en un emisor LED, que indicaría cuál es el receptor que estará esperando ser activado, y un receptor, que se

encargará de detectar una interacción. Además, una fuente de energía (batería AAA) y un microcontrolador programable que actuará como cerebro central de los 4 E/R y estará encargado de controlarlos y proporcionar una conexión a la aplicación móvil. La conexión con los E/R debe ser cableada debido a que estos no cuentan con un microcontrolador que les proporcione conexión inalámbrica.

Como punto principal de esta solución se puede mencionar que, al contar con un microcontrolador central, este se encargaría del procesamiento lógico de las operaciones no dependiendo del dispositivo móvil del usuario, el cual sólo enviaría las órdenes al microcontrolador central. Al necesitar solamente un microcontrolador programable, la cantidad máxima de E/R sería menor que para el resto de las propuestas debido a la limitada cantidad de puertos de conexión del microcontrolador necesarios para conectar más E/R. Sin embargo, representaría un costo considerablemente más económico que las otras propuestas debido a utilizar un único microcontrolador y no uno por cada E/R.

Por otro lado, la conexión con los E/R debe ser física (cableado) debido a que los E/R no cuentan con microcontrolador que procese los impulsos eléctricos necesarios para su funcionamiento. Esto sería incómodo para su uso e implicaría un límite de distancia máximo a los que se podrían separar los E/R, el cual depende del largo del cable deseado. Esto conlleva un problema a la hora de almacenar el sistema, debido a la complejidad de guardarlo y conservarlo adecuadamente para evitar daños en el cable.

3.3.3. Propuesta N° 3

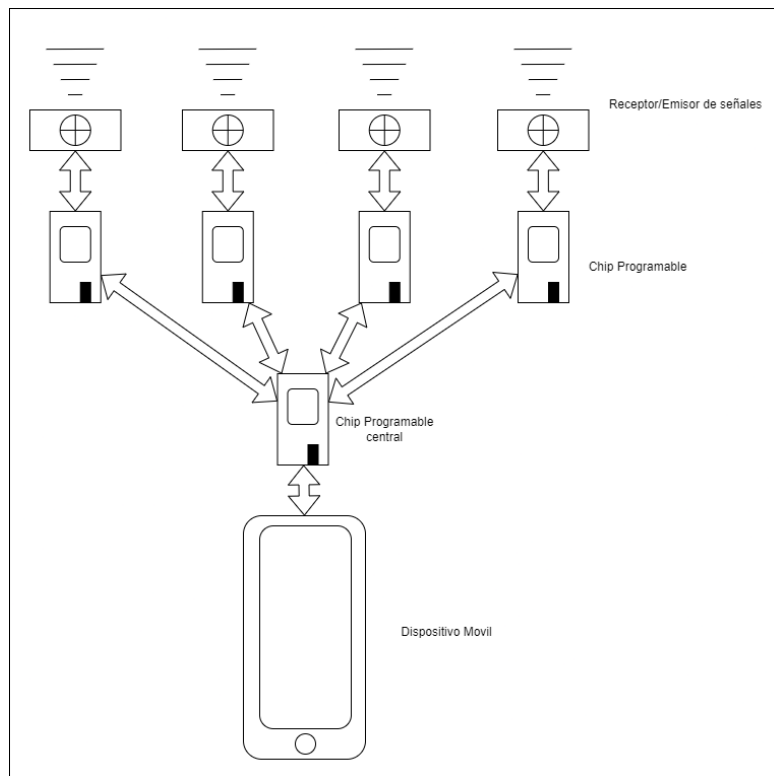


Diagrama N°3 - Ilustración de la propuesta N°3

En esta solución se hace uso de 4 o más E/R, que consisten en un emisor LED que indicaría cuál es el receptor que estará esperando ser activado y un receptor que se encargará de detectar una interacción. En este caso, se utilizará una fuente de energía y un microcontrolador programable por cada E/R encargado de proporcionar una conexión a un microcontrolador programable central que los controla y que a su vez se conectará a una aplicación móvil.

La principal ventaja de esta propuesta es contar con un microcontrolador central (nodo central) que se encargará del procesamiento lógico de las operaciones no dependiendo del dispositivo móvil del entrenador, el cual solo enviaría las órdenes al nodo central. A su vez, como cada E/R cuenta con su propio microcontrolador programable se puede usar un método inalámbrico para conectarlos al nodo central.

No obstante, la desventaja más grande es que cada E/R necesitaría un microcontrolador extra y una fuente de alimentación lo que, comparado con propuestas anteriores, resultaría más costoso. Sin embargo, debido a que estos dispositivos no superarían los

U\$S 50 a precio actual, el costo seguiría sin superar las opciones del mercado (cuyo precio varía entre € 250 y € 460 aproximadamente).

3.3.4. Propuesta N° 4

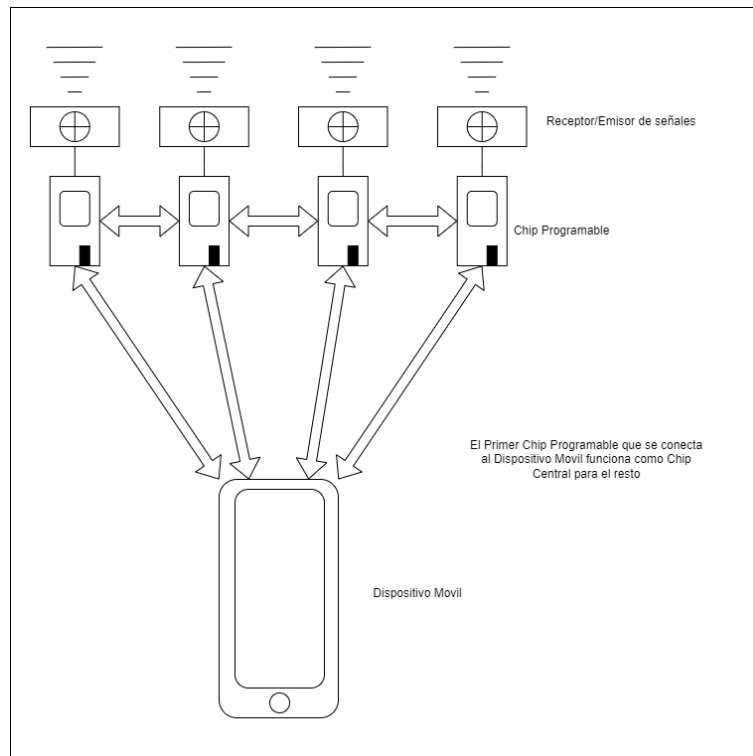


Diagrama N°4 - Ilustración de la propuesta N°4

En esta solución se hace uso de 4 o más E/R que consisten en un emisor LED, que indicaría cuál es el receptor que estaría esperando ser activado, y un receptor, que se encargará de detectar una interacción. Asimismo, una fuente de energía y un microcontrolador programable por cada E/R encargado de controlarlo, que proporciona una conexión a una aplicación móvil. En esta propuesta el primer microcontrolador programable que se conecte al dispositivo móvil servirá como nodo central y los demás se conectarán a él.

Esta solución cuenta con la ventaja de que cada microcontrolador programable es capaz de actuar como un nodo central. Al contar con un nodo central, éste se encargará del procesamiento lógico de las operaciones sin depender del dispositivo móvil del entrenador, el cual sólo enviará al nodo central las secuencias de entrenamiento. Sin embargo, se requiere incluir toda la lógica necesaria para la conexión entre microcontroladores, lo que implica que deben ser capaces de reconocer cuál es el central (el primero en encenderse).

3.3.5. Conclusión

Dadas todas las opciones, la solución más cercana a los criterios establecidos al principio de este capítulo es la propuesta N°3. Esta utiliza una configuración de microcontrolador programable central, lo que significa que habrá un microcontrolador que se encargue de toda la lógica y la conexión con la aplicación móvil, mientras que el resto será responsable de ofrecer una conexión inalámbrica entre los E/R y el nodo central.

Esta estructura es más eficiente que la propuesta N°1, debido a que cada uno de los microcontroladores tendrían que conectarse individualmente a la aplicación para saber cuál sería el siguiente paso. A su vez, la propuesta N°3 es más flexible que la propuesta N°2 debido a que la misma haría uso de una conexión física (cableada). Por otro lado, la propuesta número N°4 presenta las mismas ventajas que la propuesta número N°3 pero implicaría un gran incremento en la complejidad de la lógica utilizada y podría afectar el rendimiento de todo el sistema.

En conclusión, la propuesta N°3 hace un uso más eficiente de las tecnologías utilizadas y es más flexible al momento de su uso. Sin embargo, para un primer prototipo se desarrollará la propuesta N°2 la cual se modificará hasta llegar a las especificaciones definidas en la propuesta N°3. Esto debido a que los materiales que el equipo fue capaz de obtener al momento de realizar el prototipo (concretamente la cantidad de microcontroladores) no eran suficientes para implementar la solución N°3. Además, con la propuesta N°2 quedando operativa se cubrirán todas las funcionalidades requeridas en el proyecto. Esto hace que luego sólo haga falta añadir los demás microcontroladores propuestos en la solución N°3 sin tener que realizar grandes modificaciones en el código.

3.4. Prototipo

El funcionamiento general del prototipo seleccionado es el siguiente: desde el equipo electrónico conectado (laptop, computadora de escritorio, etc.), el usuario envía una secuencia con el orden en el que los LEDs se deben encender (definido o random) y con el tamaño máximo de la secuencia ya definido. Esto refiere a la cantidad máxima de LEDs que se pueden encender para completar el patrón. Una vez el microcontrolador recibe esa lista (con el orden y el tiempo de cada led), encenderá las luces leds en ese orden esperando el tiempo definido para cada una. Capturará el tiempo desde que se encendió el led hasta que se pulse el botón que le corresponde y lo guardará en un lista nueva que al finalizar se enviará de vuelta al equipo electrónico.

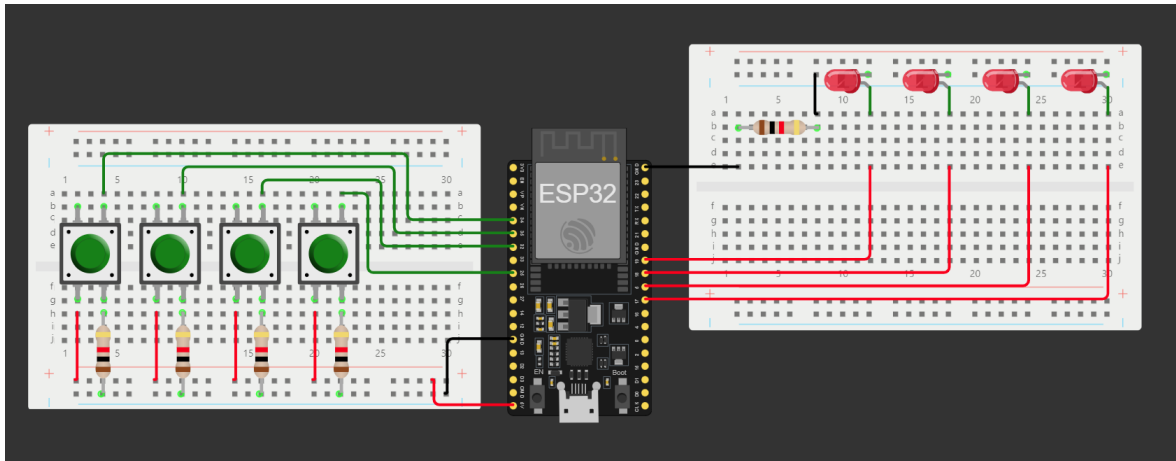


Diagrama N°5 - Ilustración electrónica del primer prototipo

3.5. Investigación de software para aplicación móvil

Para tomar la decisión de qué tecnología usar para el desarrollo de la aplicación móvil necesaria para controlar los dispositivos y llevar un registro de los datos obtenidos, se consideraron los frameworks más populares (React Native y Flutter) debido a que cuentan con grandes comunidades a su alrededor, por lo que es posible encontrar soluciones a distintos problemas comunes que pueden surgir en el desarrollo de una aplicación.

Además, ambos ofrecen la posibilidad de un desarrollo único para las distintas plataformas en las que se podría utilizar la aplicación, como páginas web, aplicaciones de escritorio, aplicaciones para dispositivos Android y para dispositivos iOS. Por lo tanto, estos frameworks son considerados multiplataforma.

Los criterios utilizados para determinar qué framework emplear fueron el rendimiento del software durante la etapa de desarrollo de una aplicación, la experiencia del usuario luego de que la aplicación ya esté en su etapa de producción y la facilidad de desarrollo dentro de la aplicación, lo cual puede depender del conocimiento que se tenga sobre el framework. A su vez, se consideró que sería necesario que el software a utilizar cuente con una comunidad activa de desarrolladores y una documentación actualizada, lo cual puede facilitar considerablemente al momento de trabajar sobre el framework. Finalmente, se necesitaría que permita que la aplicación sea mantenible a largo plazo.

3.5.1. Flutter

Según Google (Google LLC, s.f.), Flutter es un framework multiplataforma desarrollado por la empresa Google usando el lenguaje Dart, diseñado para ser eficiente y orientado a ser más amigable en el desarrollo de interfaces de usuarios (UI).

Esta herramienta muestra un mejor rendimiento que la otra opción analizada (React Native) ya que funciona dentro de su propio motor gráfico llamado Skia en el lenguaje C++. Esto le permite mantener una presentación similar sin importar la plataforma en la que se ejecute la aplicación y sin que se vea afectada por cambios de versión entre una misma plataforma, ya que no necesita utilizar los componentes nativos cada vez que se requiere representarlo en pantalla. Esto se debe a que cuando la aplicación está funcionando, las instrucciones ya están preparadas para ser ejecutadas por el hardware. Esto le da un mejor rendimiento, aunque para que esto funcione el motor gráfico se debe mantener en ejecución en todo momento, lo que genera un uso mayor de memoria que una aplicación nativa.

Este framework utiliza los Paradigmas Reactivos, que son fundamentales en el diseño y desarrollo de sistemas que deben manejar la concurrencia, la interacción con el usuario y la respuesta a eventos en tiempo real, y permite la utilización de paquetes externos (más conocidos como dependencias externas) que se gestionan desde la plataforma pub.dev. Esto agiliza al momento de programar ya que no será necesario desarrollar características que ya están publicadas.

A pesar de ser más nuevo que React Native cuenta con una amplia documentación en su página oficial que proporciona asistencia a personas con pocos conocimientos. Sin embargo, el uso de Flutter representa un gran incremento del tamaño de la aplicación a desarrollar debido a su gran kit de herramientas. Al mismo tiempo, aunque Flutter fue diseñado para ser utilizado tanto por iOS como por Android, debido a que Android y Flutter son parte de la misma empresa (Google), Android es más propenso a recibir actualizaciones y el framework suele tener mejor funcionamiento dentro de este sistema operativo. Por último, también es importante destacar el hecho de que Dart cuenta con una baja popularidad aun con las ventajas que presenta.

3.5.2. React Native

Según O'Reilly (2022), React Native es una opción popular para desarrollar aplicaciones móviles debido a su uso de JavaScript, un lenguaje dinámico e interpretado ampliamente

conocido. Aunque esto ofrece ventajas en cuanto a su popularidad y la posibilidad de utilizar TypeScript para agregar orden y tipos de datos, también se traduce en un rendimiento más lento.

El funcionamiento interno de React Native implica el uso de una serialización y deserialización sobre un puente entre la capa JavaScript y la capa nativa, donde se decide qué componente debe dibujarse y se pasa la información en formato JSON. Sin embargo, este proceso se convierte en un cuello de botella debido al uso de un único hilo. En otras palabras, este proceso conlleva un enlentecimiento del funcionamiento de la aplicación por el funcionamiento interno del framework. Para ejecutar JavaScript en React Native se emplea un motor llamado Hermes, una implementación open source optimizada para esta plataforma. Sin embargo, aunque el uso de JavaScript tiene sus ventajas en términos de popularidad, se convierte en una desventaja cuando se busca un alto rendimiento debido a serialización y deserialización.

React Native utiliza componentes nativos de la plataforma, lo que permite que una aplicación se vea como una aplicación Android en la plataforma Android y como una aplicación iOS en la plataforma iOS. Además, este framework cuenta con una variedad de opciones disponibles para el despliegue de la aplicación y es especialmente útil si se busca utilizar la misma aplicación en múltiples plataformas, incluida la web.

La capa nativa a la que React Native debe comunicarse está escrita en C++, lo que proporciona una base sólida para el desarrollo de aplicaciones móviles, aunque si se necesita algo más específico será necesario escribirlo en C++. Sin embargo, si bien este framework fue creado con las plataformas iOS y Android como objetivo, no siempre se logra el mismo resultado en ambas y la UI puede verse afectada por los cambios de versión dentro de una misma plataforma.

Por otra parte, la documentación de React Native puede no ser tan extensa y algunas partes pueden estar desactualizadas, por lo que el curso de un proyecto utilizando este framework depende en gran medida del esfuerzo voluntario y el apoyo de la comunidad para mantenerse vigente en temas de seguridad y de estabilidad que se van actualizando en el tiempo.

Finalmente, aunque existe una amplia gama de bibliotecas disponibles en npm no todas las librerías de JavaScript funcionan correctamente en React Native, lo que puede

generar dificultades en el proceso de desarrollo utilizando estas dependencias externas. Además, la comunidad mantiene la mayoría de estas librerías, lo que significa que no recibe actualizaciones oficiales y que es necesario depender de terceros.

3.5.3. Comparación de tecnologías

A continuación se compararon las tecnologías seleccionadas bajo el criterio definido:

Criterio	Flutter	React Native
Rendimiento	Cuenta con un motor gráfico propio encargado de imprimir componentes como píxeles, por lo que su lenguaje no es interpretado y eso lo hace más eficiente.	Es lento debido a que necesita procesar componentes en Javascript a componentes nativos de cada plataforma.
Experiencia de usuario y diseño (performance)	Lenguaje más amigable en la creación de interfaces de usuario. Su motor gráfico permite mantener un estilo sin depender del SO.	Las UI de React Native tienen una apariencia similar a las de las aplicaciones nativas gracias al lenguaje JavaScript que interactúa a la perfección con el entorno nativo.
Facilidad de desarrollo	Utiliza paradigmas reactivos y ofrece la posibilidad de uso de librerías externas desde su plataforma de gestión.	Al usar JavaScript cuenta con una gran cantidad de bibliotecas disponibles y opciones a la hora de realizar un despliegue.
Comunidad y documentación	Documentación completa y actualizada. Comunidad activa, pero no tan popular.	Existe una gran comunidad alrededor de este framework debido a su simplicidad. Cuenta con documentación extensa.

Mantenimiento a largo plazo	Debido a su alta performance y utilización de Paradigmas Reactivos permite una mayor escalabilidad a largo plazo.	Utiliza componentes y un enfoque declarativo para generar interfaces de usuario, lo que facilita la creación de aplicaciones escalables y mantenibles.
------------------------------------	---	--

Tabla N°3 - Comparativa de las tecnologías de desarrollo de aplicación móvil

3.5.4. Conclusión

Flutter permite desarrollar aplicaciones de uso a largo plazo con un mejor rendimiento y escalabilidad que otras opciones. Por otro lado, React Native es una excelente opción para comenzar un proyecto si el objetivo es desarrollar un prototipo de baja escala. Sin embargo, para hacer una correcta utilización de Flutter es necesario tener conocimiento sobre el lenguaje de programación Dart. Este no es el caso para React Native debido a que utiliza JavaScript, un lenguaje de programación muy popular, en el que el equipo cuenta con una amplia experiencia.

Se concluye, entonces, que sería más apropiada la utilización de React Native considerando las necesidades que se tienen en este proyecto de dependencias de conexiones, tipos de datos a ser manejados y utilización de componentes reutilizables, así como a su gran comunidad activa que comparte información relevante para el desarrollo.

3.6. Investigación de tecnología de almacenamiento

Uno de los requerimientos de este proyecto es almacenar las secuencias creadas, la información de cada jugador y los datos obtenidos de las sesiones de entrenamiento de cada uno. Dentro de la aplicación móvil guardar toda esta información no sería posible ya que se cuenta con espacio limitado dependiendo de la capacidad de cada dispositivo. A su vez se requiere que sea capaz de funcionar sin conexión a internet; por ende almacenar todos los datos directamente en una base de datos externa tampoco es una solución completa. Por lo tanto, se optó por desarrollar un sistema mixto que guardará de forma local la información más relevante (por ejemplo la lista de los jugadores, las 10 últimas rutinas creadas o los 5 últimos resultados de las sesiones y todo lo que se genere desde la última conexión). Una vez que el dispositivo móvil tenga conexión a internet deberá guardar remotamente toda esa información en una base de datos.

Las bases de datos (BD) hacen referencia a una colección de datos que guardan una relación entre sí y que se almacenan con el propósito de su futura utilización. Los Sistemas Gestores de Bases de Datos (SGBD) son los programas que facilitan dicho almacenamiento y el acceso a la información con vistas a su procesamiento.

Existen dos tipos de BD, los SQL (Structured Query Language) y los NoSQL (Not Only SQLes), teniendo un impacto directo en el rendimiento y desarrollo de cualquier proyecto dependiendo por cual se decida utilizar.

Las SQL, como su nombre lo indica, son BD que funcionan con un lenguaje de consulta estructurada que se caracterizan por tener un esquema determinado que define cómo son las tablas en las que se almacenan los datos, qué tipo de campos tienen y cómo se relacionan entre sí. Por otro lado, las NoSQL son BD no relacionales o sea que no cuentan con un identificador que relacione un conjunto de datos con otro. La información está organizada generalmente como documentos y no requieren que los datos estén estructurados para poder manipularlos. Por lo tanto, se investigaron las opciones más populares y robustas de cada tipo de BD: SQL y NoSQL.

Considerando las limitaciones de los dispositivos a utilizar y lo solicitado por el cliente se busca que la base de datos cuente con las siguientes características:

- Facilidad de uso.
- Tamaño reducido (que sea lo más eficiente posible para ahorrar recursos de procesamiento).
- Buen rendimiento (velocidad de consultas e inserciones lo suficientemente rápida como para no causar esperas en los entrenamientos).
- Escalabilidad.
- Capacidad de soportar un aumento considerable del volumen de datos ingresados o consultados en un corto tiempo.

3.6.1. MySQL

Según Robledano (2019) el motor de base de datos MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo una licencia de doble uso compuesta por la Licencia Pública General y la Licencia Comercial, otorgada por Oracle Corporation. Es ampliamente reconocida como la base de datos de código abierto más utilizada en el mundo debido a que es fácilmente accesible a los programadores para ser utilizada en sus proyectos.

Este motor de base de datos es ligero y puede ser usado en dispositivos de bajos recursos; a su vez, cuenta con una considerable velocidad de operaciones lo que garantiza un buen rendimiento de aplicaciones incluso en dichos dispositivos. Adicionalmente, es un programa multiplataforma con gran facilidad para la instalación y configuración.

Por otra parte, cuenta con limitaciones en cuanto a seguridad, especialmente si no se configura adecuadamente. También se cuenta con la desventaja de que varias de sus funcionalidades no están correctamente documentadas y la interfaz del usuario puede no ser intuitiva lo que dificulta el uso de esta herramienta.

3.6.2. PostgreSQL

Según la página oficial, PostgreSQL refiere a un potente sistema de base de datos objeto-relacional de código abierto con más de 35 años de desarrollo activo y una sólida reputación por su fiabilidad, robustez de características y rendimiento. Además, existe una gran cantidad de información que describe cómo instalar y utilizar PostgreSQL a través de su documentación oficial, mientras que la comunidad de código abierto proporciona diversos lugares útiles para familiarizarse con el lenguaje y descubrir cómo funciona (PostgreSQL, s.f.).

La principal ventaja de PostgreSQL es su gran escalabilidad, siendo capaz de ajustarse al número de procesadores y a la cantidad de memoria disponible de forma óptima, lo que resulta en una mayor cantidad de peticiones simultáneas a la base de datos. Además, esta herramienta presenta una gran estabilidad y confiabilidad debido a su uso de Hot-Standby (modo de espera activo). Aun así, los datos todavía se reflejan en tiempo real. Por tanto, ambos sistemas tienen datos idénticos. Finalmente, PostgreSQL cuenta con una amplia variedad de extensiones proporcionadas tanto por desarrolladores de la herramienta como por terceros. Estas extensiones existen para una multitud de lenguajes de programación como pueden ser Java, Python, C++, entre otros.

Sin embargo, PostgreSQL es lento a la hora de realizar cambios en bases de datos de bajo volumen debido a que está diseñado para ambientes de alto volumen. Adicionalmente, la sintaxis de algunos de sus comandos o sentencias puede no ser intuitiva si no se tiene conocimientos en lenguaje SQL.

3.6.3. MongoDB

Según la página oficial, MongoDB representa un sistema de gestión de bases de datos NoSQL caracterizado por su enfoque en documentos y su naturaleza de código abierto. Menciona que en contraste con las bases de datos relacionales convencionales que almacenan información en tablas, MongoDB almacena datos en estructuras BSON, las cuales se asemejan a la especificación JSON y presentan un esquema dinámico. Además afirma que este enfoque facilita y agiliza la integración de datos en aplicaciones específicas (MongoDB, s.f.).

MongoDB es una solución de bases de datos altamente apropiada para implementaciones en entornos de producción, destacándose por su rica variedad de capacidades y funcionalidades. Es ampliamente adoptada en la industria, con implementaciones relevantes en organizaciones como Electronic Arts, eBay, GOOGLE y SEGA.

La principal ventaja de este motor de base de datos es su gran integración con otras herramientas que fueron desarrolladas en el lenguaje JavaScript y la gran cantidad de documentación que cuenta para realizar esta integración. A su vez es de bajo coste por lo que puede ser utilizado en entornos de bajos recursos de procesamiento y al mismo tiempo fue creado considerando una gran velocidad de lectura y escritura. Esto la hace una herramienta ideal para aplicaciones web y móviles.

No obstante, esta herramienta cuenta con dificultades de compatibilidad en cuanto a consultas. Las consultas complejas pueden ser difíciles de realizar en este software, debido a la falta de soporte para JOIN y otras operaciones de consulta avanzadas. Además, MongoDB cuenta con una comunidad menor comparada con otras herramientas a pesar de su popularidad por lo que hace más difícil encontrar ayuda en línea.

3.6.4. Firebase

Según la página oficial (Google LLC, s.f.), Firebase Realtime Database es una base de datos alojada en la nube. Esta menciona que los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Además, afirma que cuando se compila aplicaciones multiplataforma con SDK de plataformas de Apple, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

Esta herramienta fue desarrollada considerando el uso de múltiples usuarios y por lo tanto cuenta con la gran ventaja de ser capaz de realizar almacenamiento en la nube y escalamiento automático. Esto puede disminuir el uso de recursos de los dispositivos y permitir una mayor velocidad en las aplicaciones que lo utilicen. Al mismo tiempo, Firebase fue diseñado para ser multiplataforma y funcionar sin dificultades en todo tipo de aplicaciones.

Por otro lado, a menos que la aplicación que utilice Firebase ejecute una base de datos actualizada por una gran cantidad de usuarios su uso no tendría sentido. Además, el formato de almacenamiento de esta herramienta es completamente diferente al de SQL y bases de datos NoSQL (Firebase usa JSON), por lo que no se puede migrar fácilmente de una base a otra.

3.6.5. Conexión a base de datos

Debido a que será necesario manejar un nivel de seguridad que garantice que los datos serán usados con un fin específico, la aplicación móvil no será la encargada de administrar la base de datos directamente sino que se comunicará indirectamente a través de una API. Esta será responsable de garantizar el correcto acceso a los datos mediante usuarios autorizados y limitando las posibles consultas a las estrictamente necesarias.

Al momento de decidir bajo qué lenguaje estaría funcionando la API se investigaron las siguientes tecnologías:

- Node.js: un entorno de ejecución JavaScript de código abierto y multiplataforma.
- Deno: un entorno de ejecución Javascript fuera del navegador que permite realizar aplicaciones en cualquier ámbito.
- Nim: un lenguaje de programación multiplataforma de código abierto compilado estáticamente inspirado en Pascal y Python.

3.6.5.1. Node.js

Node.js es un entorno de ejecución JavaScript de código abierto y multiplataforma que se utiliza para desarrollar aplicaciones escalables del lado del servidor y de red. Está basado en el motor de ejecución JavaScript, funciona de forma eficiente, ligera y es el software adecuado para aplicaciones de uso intensivo de datos y en tiempo real, como pueden ser chats en tiempo real, streaming de datos, tableros de control del sistema o APIs REST (Node.js, s.f).

3.6.5.2. Deno

Deno es un entorno de ejecución de JavaScript y TypeScript multiplataforma basado en el motor V8 de Chrome desarrollado por el creador de Node.js, Ryan Dahl. Esta tecnología surge para corregir una serie de defectos que en Node.js se habían acentuado con el paso del tiempo y el avance de las tecnologías. Este entorno ofrece una gran seguridad tanto al programador como al usuario, debido a que solo usa permisos absolutamente necesarios, a diferencia de Node.js que utiliza todos los permisos del sistema. Tampoco depende del manejador de paquetes NPM, mientras Node.js sí lo hace (Barragán, 2022).

3.6.5.3. Nim

Nim (o Nimrod) es un lenguaje de programación multiplataforma de código abierto lanzado en 2017. Esta tecnología es de tipo estático, presenta una gran velocidad en cuanto a tiempos de ejecución y además puede ser compilado y ejecutado en cualquier plataforma que cuente con un compilador de C (prácticamente cualquier plataforma cuenta con un compilador de este tipo). Esta tecnología muestra ser muy eficiente debido a que genera ejecutables nativos para el sistema operativo en el que se esté ejecutando, sin dependencias externas de librerías o máquinas virtuales. Finalmente, al estar escrito en C y ser de código abierto, asegura estabilidad y seguridad (hmong.es, s.f.).

Una vez consideradas las características de las tres tecnologías, se define que la API se desarrollará bajo el entorno de ejecución Node.js debido a ser la opción disponible más eficiente y rápida, considerando el alcance del proyecto y gracias a su sistema de empaquetado que permite la utilización de librerías específicas para lo que se busca desarrollar. A pesar de que, como fue comentado, Deno es un entorno pensado para mejorar y arreglar defectos encontrados en Node.js, el mismo es una tecnología relativamente nueva en comparación, además de contar con una comunidad más pequeña. Estos factores tienen como resultado varios aspectos a corregir y actualizar, y a la hora de buscar información y/o documentación que ayude a resolver problemas presentados en los desarrollos, Deno no cuenta con tanto respaldo como sí lo tiene Nodejs.

3.6.6. Comparación de bases de datos

Teniendo en cuenta los objetivos y el alcance de este proyecto, la base de datos a implementar debe almacenar datos de manera rápida dado que no siempre se tendrá acceso a internet; por lo tanto, el almacenamiento se realizará en cortos periodos de forma intermitente.

A su vez, las necesidades a tener en cuenta requieren que la base de datos cuente con un rendimiento estable, debe ser capaz de funcionar en entornos de bajos recursos de procesamiento, debe implicar una baja dificultad al momento de implementarla y contar con una capacidad considerable de escalabilidad de datos, algo imprescindible debido a la gran cantidad de datos que podría llegar a almacenar.

Criterio	MySQL	PostgreSQL	MongoDB	Firebase
Facilidad de uso	Muy conocida por la comunidad de desarrolladores . Cuenta con bastantes aportes de la comunidad para la solución de errores.	Buena reputación entre las empresas. Gran apoyo de la comunidad de desarrolladores	Simpleza en sus sistemas. Utiliza métodos fácilmente reconocibles.	Sistemas simplificados en la nube. Cuenta con la capacidad de escalabilidad automática.
Tipo de BD	SQL	SQL	NoSQL	NoSQL, nube
Tamaño	Motor ligero y adaptable, debe ser instalado localmente.	No muy liviano, no apto para sistemas limitados.	Pensado para aplicaciones web, puede utilizar la opción en la nube. Sin costo de recursos o implementar localmente	No se aloja localmente, sólo en servidores externos. No tiene costo de recursos.
Rendimiento	Considerable velocidad para consultas e inserciones de datos.	Equilibrado en las diferentes operaciones.	Excelente en velocidad de consultas, no tan eficiente en inserción de datos.	Dependiente de la conexión wifi y cantidad de datos.
Escalabilidad	Escalabilidad lenta y gradual.	Escalabilidad lenta y gradual.	Incremento veloz y sustancial.	Limitado por la conexión. Veloz.
Volumen de datos	Buena para BD que no tengan la necesidad de un crecimiento	Buena para BD que no tengan la necesidad de un crecimiento	Buena para BD que tengan la necesidad de un crecimiento rápido y con grandes	Buena para BD que tengan la necesidad de un crecimiento rápido y con grandes

	rápido de datos y consultas.	rápido de datos y consultas.	aumentos en poco tiempo.	aumentos en poco tiempo.
--	------------------------------	------------------------------	--------------------------	--------------------------

Tabla N°4 comparativa de las bases de datos

3.6.7. Conclusión

Debido a que este proyecto requiere una gran escalabilidad por el volumen de datos que es almacenado en cortos periodos de tiempo de forma intermitente, se concluyó que las bases de datos SQL no serían las más adecuadas, optándose por utilizar NoSQL.

La tecnología que se eligió para este desarrollo es MongoDB, debido a su facilidad de uso y complementación con el entorno de desarrollo Nodejs. Utilizar este gestor de bases de datos permitirá tener estructuras de datos dinámicas (esto refiere a la forma en la que se guardan los datos) e implicará menor tiempo de recuperación ante fallas. Además, su característica de basarse en un código abierto hace que la comunidad de MongoDB sea una de las más grandes y activas en comparación con el resto de bases de datos NoSQL.

3.7. Almacenamiento local de dispositivo móvil

Para el almacenamiento local de los datos en el celular fue necesario realizar una investigación sobre tecnologías de almacenamiento disponibles para React Native, ya que las tecnologías de bases de datos estudiadas están pensadas para ser instaladas en un servidor remoto y no en un dispositivo local (celular).

Además, partiendo de la base de que no todos los usuarios contarán con dispositivos móviles con gran capacidad de procesamiento y de recursos, se necesita una base de datos que sea capaz de trabajar de forma rápida y eficiente en un ambiente limitado y que pueda funcionar en una gran variedad de sistemas y dispositivos sin importar la capacidad de los mismos. Las tecnologías más utilizadas y recomendadas por la comunidad son SQLite y AsyncStorage.

3.7.1. SQLite

Según Rómmel (s.f.) SQLite es un software de código abierto y gratuito que proporciona una solución accesible y eficiente para el almacenamiento de datos en dispositivos integrados, como los microprocesadores, y en dispositivos con limitaciones de hardware, como los teléfonos celulares. SQLite cumple con el estándar SQL92 y también incluye extensiones que simplifican su implementación en cualquier entorno de desarrollo. Por lo tanto, SQLite es capaz de manejar consultas SQL tanto básicas como avanzadas.

A su vez, Rómmel (s.f.) agrega que la principal ventaja de SQLite es la capacidad de guardar la base de datos completa en un archivo ligero. Además, esta herramienta es capaz de funcionar completamente en memoria, es decir que guarda los datos directamente en la memoria RAM del dispositivo, permitiendo un acceso muy rápido a sus datos. Sin embargo, esta tecnología no es eficaz en casos de gran crecimiento de la base, no es personalizable y carece de funciones de seguridad y administración de usuarios.

3.7.2. AsyncStorage

Es una biblioteca que simplifica la utilización de SQLite para React Native, asemejando el funcionamiento del almacenamiento local de los navegadores web. Esto hace que tener experiencia utilizando React para desarrollo de páginas web, se convierta en una ventaja a la hora de trabajar con esta herramienta, por las grandes similitudes con dicha tecnología. Esta tecnología simplifica bastante la interacción con la base de datos, ya que utiliza las funciones `setItem` y `getItem` (semejante a lo que realizan los navegadores web), las cuales almacenan cadenas de texto enlazadas a una key que las identifica. Esto significa que no se necesitan utilizar directamente consultas de SQL (AsyncStorage, s.f.).

3.7.3. Comparación de almacenamientos locales

	SQLite	AsyncStorage
Almacenamiento y consulta de datos	Lenguaje de SQL	Simula el funcionamiento de LocalStorage web. <code>SetItem</code> y <code>GetItem</code> en forma de cadena de texto.
Capacidad necesaria	Poca capacidad	Poca capacidad
Ambientes de funcionamiento	Diseñado para funcionar en sistemas embebidos	Diseñado para funcionar en sistemas embebidos

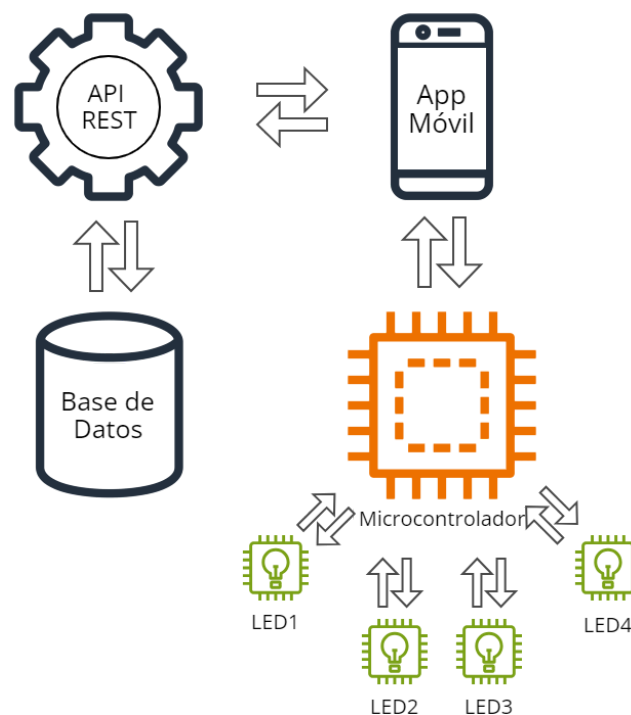
Tabla N°5 comparativa de las tecnologías de almacenamiento local

3.7.4. Conclusión

Considerando el contexto del proyecto, y siendo AsyncStorage una biblioteca que simplifica la utilización de SQLite para un entorno desarrollado en React Native (como es el caso de esta aplicación), el equipo se decide por su utilización. Además, se cuenta con experiencia en el uso de almacenamiento local desde los navegadores web, por lo que su uso no conlleva el aprendizaje de este tipo de funciones para su implementación.

3.8. Esquema de integración de tecnologías

Las diferentes tecnologías presentadas se integran en la solución de la siguiente manera. El usuario de la aplicación móvil (entrenador) seleccionará una rutina, la cual será enviada hacia el microcontrolador utilizando Bluetooth Low Energy (BLE). Luego, el jugador deberá completar la rutina interactuando con el sistema conectado al microcontrolador. Este sistema le dirá el orden en el cual debe presionar los botones utilizando LEDs vinculados a los mismos y, una vez presionados, registrará el tiempo de reacción. Al concluir la rutina los datos registrados serán enviados a la aplicación móvil para poder ser validada por el usuario, para luego ser reorganizados y guardados en el almacenamiento local hasta conseguir una conexión a internet. Finalmente, al conectarse por red Wi-Fi a internet los datos guardados localmente serán reenviados hacia la API REST y luego insertados en la base de datos, pudiendo ser visualizados desde la aplicación móvil.



Esquema N°1 - Integración de tecnologías.

4. Resultados

En esta sección se detallarán los resultados obtenidos una vez concluido el desarrollo del proyecto, los cuales se relacionan directamente con los objetivos previamente planteados. Como se detalló en la Introducción, se pretendía la implementación de un prototipo de solución tecnológica que permitiera medir el tiempo de reacción de los arqueros que realicen el entrenamiento con la herramienta. Dicho prototipo debía contar con las siguientes características: estar compuesta por una solución electrónica capaz de captar estímulos mediante sensores y una solución informática para dispositivos móviles que fuera amigable y que permitiera al usuario registrar rutinas y jugadores. Además, se debía posibilitar al usuario la visualización de los datos obtenidos por parte de la solución electrónica y, finalmente, implementar un esquema de base de datos donde se guardara toda la información. Por otra parte, los resultados esperados establecen que el prototipo desarrollado debía contar con una interfaz de uso amigable y sencilla, y que los datos obtenidos se guardaran de manera segura.

Al comparar lo anterior con los resultados obtenidos, se encuentra un punto que se entiende que faltó implementar para cumplir con la totalidad de lo planificado: el prototipo fue desarrollado utilizando botones de pulsación que, al ser presionados, envían una señal al microcontrolador indicando que fue realizada una acción a través de ellos. La planificación inicial detalla el uso de sensores para ser utilizados como dispositivos de entrada de la aplicación, cumpliendo la misma función que los botones pero con una mayor accesibilidad al no necesitar presionarlos sino sólo capturar una señal a una cierta distancia del dispositivo.

No obstante, el prototipo desarrollado es completamente funcional, solucionando la mayor parte de los problemas presentados y logrando establecer una conexión entre los distintos componentes del prototipo (leds y botones que reciben y envían señales hacia y por el usuario), pudiéndose conectar con el microcontrolador y recibir órdenes que serán enviadas desde la aplicación móvil, siendo procesadas por el microcontrolador y finalmente recibidas por los dispositivos E/R. Este circuito de conexiones también permite que los datos recolectados por el microcontrolador desde los dispositivos E/R puedan ser visualizados desde la aplicación móvil y, a su vez, que los mismos sean guardados localmente en el dispositivo y remotamente en una base de datos alojada en un servidor. Así, se genera un ecosistema estable entre distintos tipos de dispositivos

electrónico/informáticos y distintas tecnologías utilizadas para su programación y desarrollo.

De esta forma, quedan como principales mejoras a futuro el cambio de dispositivos de entrada (botones por sensores), además de una mejora y adaptación del hardware que ya se tiene para que se encuentre más protegido y a la vez accesible para realizar entrenamientos, para que así la solución pueda ser utilizada en entrenamientos reales por la Selección Uruguaya de Handball.

5. Conclusiones y Trabajo a futuro

La solución obtenida es una aplicación móvil que cuenta con distintas funcionalidades que le permiten registrar métricas sobre el rendimiento de los arqueros de handball, una vez los mismos interactúen con el dispositivo electrónico conectado. El dispositivo electrónico cuenta con 4 E/R los cuales, haciendo uso de sus LEDs, mostrarán distintos patrones para medir la velocidad de reacción del arquero. Esta herramienta se caracteriza por un bajo uso de recursos (en cuanto a requerimientos de software), por su portabilidad (siendo capaz de ejecutarse en diferentes plataformas), por ser trasladable (es un dispositivo que es capaz de ser transportado y utilizado en distintas localizaciones) y ser capaz de funcionar incluso sin conexión a internet.

Cabe destacar que, teniendo en cuenta el alcance del proyecto y los requerimientos del cliente, se obtuvo un prototipo que cumple con la mayor parte de lo solicitado y establece, además, un camino claro para extensiones y potenciales funcionalidades a desarrollar a futuro. Además, si bien se cumplen la totalidad de las funcionalidades solicitadas por el cliente (recolección de datos que se puedan visualizar, sistema integrado de E/R que capturen las métricas y aplicación móvil que permita administrar estos datos), la presentación y el armado no es suficiente para obtener una experiencia de usuario real, ya que se necesitarían pequeños contenedores que ayuden a proteger a los componentes, y que sus E/R puedan colocarse a más distancia del microcontrolador de lo que se encuentran en el momento.

En relación a los trabajos a futuro surgidos a partir de la investigación realizada sobre el dispositivo electrónico, se plantea la implementación de sensores ultrasónicos como una opción para mejorar el funcionamiento y la usabilidad del proyecto. Estos sensores emiten ondas ultrasónicas y son capaces de recibir la onda reflejada que retorna desde un objeto, lo que les permite medir la distancia con gran precisión. La implementación de estos sensores podría ofrecer una mayor flexibilidad en el uso de la herramienta, ya que sería más accesible mover las manos frente a los sensores para captar una señal en lugar de tener que presionar un botón. Asimismo, esta opción eliminaría factores que pueden ocasionar daños a la herramienta, ya que durante los entrenamientos la combinación de botones y la adrenalina podrían llevar a aplicar fuerza excesiva sobre éstos provocando desperfectos.

Por otro lado, se plantea la implementación de la propuesta N° 3 que hace uso de más microcontroladores, uno por cada E/R. Esto permitiría que los mismos funcionaran independientemente y se conectaran inalámbricamente al microcontrolador central, posibilitando así un mayor rango de alcance de conexión entre cada nodo. Como consecuencia de eso, se tendría la oportunidad de realizar rutinas más complejas y variadas dado que se podría reorganizar cada nodo por separado dentro del espacio de entrenamiento.

En cuanto a trabajos a futuro de la aplicación móvil se plantea implementar la conexión de la aplicación con la API, a fin de que sea posible almacenar los datos obtenidos tanto en forma local como remota. Por otra parte, se plantea implementar un sistema que permita exportar los datos registrados por la aplicación a un formato de documento, para facilitar la lectura y manejo de los datos obtenidos.

Glosario de términos

Almacenamiento local

Hace referencia a la manera en la que se guardan los datos generados por la aplicación. En este caso, la aplicación que trabaje con el método “local storage” almacenará los datos en la memoria del dispositivo que tenga instalada la aplicación o, en ciertos casos donde la aplicación trabaja con distintos dispositivos dentro de una red, los datos se almacenarán dentro de la infraestructura definida por la organización responsable.

Back-End

Término para referirse al área lógica de una aplicación. Es la arquitectura interna del software que asegura que los elementos del mismo funcionen de manera correcta. No es visible al usuario y no incluye ningún tipo de elemento gráfico.

Base de datos objeto-relacional

El modelo de base de datos objeto-relacional integra los conceptos de la tradicional base de datos relacional y los conceptos de paradigma de objetos que se utiliza en la programación orientada a objetos.

Base de datos relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

Binarios

Los binarios son archivos normales que albergan información legible por el sistema. Estos archivos binarios pueden consistir en ejecutables que instruyen al sistema para llevar a cabo determinadas tareas. Los comandos y las aplicaciones se encuentran almacenados en archivos binarios ejecutables. Para la transformación de texto ASCII en código binario se emplean programas de compilación especializados.

BSON

Es un formato de intercambio de datos usado principalmente para su almacenamiento y

transferencia en la base de datos MongoDB. Es una representación binaria de estructuras de datos y mapas.

Emisor/Receptor

(E/R) hace referencia al dispositivo con el que el jugador interactúa en este proyecto el cual está constituido por un conjunto de luces (emisores) y botones/sensores de proximidad (Receptores).

Entornos de Producción

Es un entorno operativo en el que una aplicación de software o el sistema es implementado y utilizado por los usuarios finales para realizar las tareas previstas.

Enfoque declarativo

Se refiere a una forma de expresar un problema o una solución en términos de lo que se desea lograr en lugar de cómo se desea lograrlo. Por lo tanto, se especifica el resultado deseado y no los pasos detallados para resolver un problema.

Ethernet

Este término hace referencia a la tecnología tradicional utilizada para conectar dispositivos en una red local o amplia por cable. Estos dispositivos se pueden comunicar mediante un protocolo de comunicación. Específicamente, Ethernet se describe cómo los dispositivos dentro de la red pueden formatear y transmitir datos para que otros dispositivos los puedan reconocer, recibir y procesar dicha información.

Front-End

Es la parte del desarrollo web dedicada al componente frontal de un sitio web, es decir del diseño, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos.

Funciones getItem y setItem

Estas funciones son utilizadas cuando se trabaja con almacenamiento local desde los navegadores web. La función “getItem” recibe como parámetro el valor de una clave y devuelve el contenido del valor almacenado en el dispositivo, que se encuentra bajo el nombre de esta clave. La función “setItem” hace lo contrario, recibiendo por parámetros el

nombre de la clave y a su vez el valor que se guardará bajo este nombre en el dispositivo, para posteriormente almacenar estos datos en el mismo.

Gamificación

Es una estrategia de enseñanza que lleva los elementos de los juegos al entorno educativo y profesional con el propósito de obtener resultados superiores. Esto puede incluir una mejora en la asimilación de ciertos conocimientos, el perfeccionamiento de habilidades específicas o la incentivación de acciones concretas, entre otros múltiples objetivos.

Hermes

Hermes es un motor JavaScript de código abierto optimizado para React Native. React Native utiliza Hermes de forma predeterminada y no se requiere configuración adicional para habilitarlo.

Hot-Standby (Modo de espera activo)

Es un método redundante en el que un sistema se ejecuta al mismo tiempo que un sistema primario idéntico. Ante la falla del sistema primario, el sistema de reserva activa se hace cargo inmediatamente, reemplazando el sistema primario. Sin embargo, los datos todavía se reflejan en tiempo real. Por lo tanto, ambos sistemas tienen datos idénticos. Esta técnica es utilizada especialmente cuando se quiere implementar Alta Disponibilidad en los Servidores de base de datos puesto que mediante esta técnica se puede mejorar los servidores en cuanto a disponibilidad y usabilidad.

Interfaz de Usuario (UI)

La interfaz de usuario es el medio que permite la comunicación entre un usuario y un dispositivo y comprende todos los puntos de contacto entre el usuario y el equipo.

Intrared

Intrared, más conocida como WLAN por sus siglas en inglés “red de área local inalámbrica”, se refiere a una red interna local de corto alcance.

I2C

I2C es un puerto y protocolo de comunicación serial que define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación: SDA y SCL. Este protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, alcanzando velocidades de 100, 400 y 1000 kbits/s.

I2S

I2S es un estándar eléctrico de bus serial utilizado para interconectar circuitos de audio digital. El bus I2S separa las señales de datos y de reloj, lo que hace que se produzcan menores cantidades de fluctuación de la señal que son recuperadas por el reloj de la señal de datos. Este bus consiste de 3 líneas: la línea de reloj de bit, la línea de reloj de palabra y al menos una línea de datos multiplexados.

JavaScript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web.

KEY (AsyncStorage)

Hace referencia a una cadena de caracteres que se utiliza como identificador único para almacenar y acceder a valores en AsyncStorage. En AsyncStorage se pueden guardar diferentes tipos de datos, como cadenas de texto, objetos JSON, números, etc. Cada valor guardado se asocia a una clave (key) única, que se utiliza como punto de acceso para recuperar o modificar ese valor.

LCD

La sigla LCD hace referencia a Liquid Crystal Display; su traducción al español sería Pantalla de Cristal Líquido. Por lo tanto, LCD es un tipo de pantalla caracterizada por ser plana y estar formada por píxeles que contienen moléculas de cristal líquido. Estos píxeles se ubican entre dos capas de cristales polarizados y son iluminados por una fuente de luz fija y constante. Las moléculas de cristal líquido permiten el paso de mayor o menor cantidad de luz según el estímulo eléctrico que reciben. Esto significa que una

pantalla de LCD muestra las imágenes a partir de esa retroiluminación que permite la combinación de los colores.

Memoria RAM

La Memoria de Acceso Aleatorio, por su traducción del inglés Random Access Memory, es la memoria de un computador o dispositivo que se encarga de almacenar la información que necesita un programa durante su ejecución. El término “aleatorio” en su nombre hace referencia a que los datos almacenados en dicho dispositivo son guardados en cualquier orden, es decir de manera aleatoria y no en secuencia.

Motor Gráfico

Es un software encargado de dibujar gráficos de una aplicación en la pantalla de un dispositivo digital.

NFC Technology

NFC, acrónimo de Near Field Communication, constituye una tecnología inalámbrica operante en la banda de frecuencia de 13.56 MHz. Esta tecnología se deriva de las etiquetas RFID, utilizadas comúnmente en medios de transporte y en sistemas de seguridad en establecimientos comerciales para pago de servicios.

NPM

NPM son las siglas de Node Package Manager. Consiste en una herramienta de línea de comandos que se usa para instalar y actualizar dependencias en proyectos Javascript o NodeJS.

Paradigmas reactivos

Los paradigmas reactivos representan enfoques de programación que se orientan hacia la construcción de sistemas fundamentados en eventos y respuestas ante estímulos externos. Dichos paradigmas se utilizan para abordar problemas de concurrencia, interacción y gestión de eventos en aplicaciones y sistemas distribuidos.

Pub.dev

Es un gestor de paquetes donde se pueden encontrar dependencias tanto para Dart como para Flutter. Esta plataforma cuenta con paquetes oficiales mantenidos por Google,

el equipo de Flutter y Flutter community.

Raspberry Pi

Es un dispositivo informático del tamaño de una tarjeta de crédito, caracterizado por su accesibilidad económica. Se conecta a un monitor del computador o televisor haciendo uso de un teclado y ratón estándar. Representa una herramienta compacta que brinda la posibilidad de explorar el ámbito de la informática y aprender a programar en lenguajes como Scratch o Python. Raspberry Pi es capaz de desempeñar todas las funciones que se esperan de una computadora de escritorio, incluyendo la navegación por la web, la reproducción de contenido en alta definición, la creación de hojas de cálculo, el procesamiento de textos y la ejecución de videojuegos.

Sensores de efecto Hall (Sensores Hall)

Están incluidos dentro del apartado de sensores de proximidad. Estos sensores detectan la presencia utilizando la magnitud del campo magnético creado por un objeto. Esto se basa en el principio de efecto Hall, utilizado para detectar la presencia e intensidad de un campo magnético.

Sensores táctiles capacitivos

Este tipo de sensores presenta un comportamiento similar a un pulsador o botón pero pueden ser activados con poca o nula presión. El principio de su funcionamiento es la medición de la variación de la capacitancia. Esto quiere decir que la placa sensora y el cuerpo humano actúan como condensadores y forman un sistema que almacena una carga de electricidad. Al reducirse las distancias, esta capacitancia aumenta y se almacena una carga eléctrica superior. Estas cargas pueden ser detectadas en la placa sensora y generar una señal digital cuando se supere un valor determinado. Estas señales pueden ser capturadas por chips Arduino.

Skia

Es una librería de gráficos 2D de código abierto que ofrece interfaces de programación de aplicaciones comunes que operan en una amplia gama de plataformas de hardware y software. Actúa como el motor gráfico principal para Google Chrome y ChromeOS, Android, Flutter, Mozilla Firefox y Firefox OS, además de ser utilizado en numerosos otros productos.

SoC

SoC (System on a Chip). Se trata de un chip/circuito integrado que contiene muchos componentes de una computadora, normalmente la CPU (a través de un microprocesador o microcontrolador), la memoria, los puertos de entrada/salida (E/S) y el almacenamiento secundario en una única placa.

SQL92

El American National Standards Institute (ANSI), liberó distintas versiones de estándares definidos para el lenguaje SQL. Su primera versión se liberó en el año 1986, y una segunda en el año 1989. Posteriormente, ANSI libera una actualización del estándar en el año 1992, la cuál es conocida por los nombres SQL92 y SQL2. Estas actualizaciones trajeron consigo nuevos comandos y capacidades para este lenguaje.

Tarjeta SD

Son tarjetas de memoria, es decir dispositivos de almacenamiento de datos que guardan los mismos en un tipo de memoria flash. Este tipo de memorias son no volátiles, es decir que conservan los datos incluso cuando no se les proporciona energía eléctrica. El término SD proviene de Secure Digital, lo que hace que este tipo de memorias extraíbles se adapten a los estándares de la Asociación Digital de Seguridad (lo que permite versatilidad y que se puedan utilizar en distintos dispositivos independientemente del fabricante).

UART (Universal Asynchronous Receiver/Transmitter)

Por su traducción al español, Receptor/Transmisor Asíncrono Universal. Este dispositivo define un protocolo o conjunto de reglas para intercambiar datos en serie entre dos dispositivos (un receptor y un emisor). La conexión mediante UART puede ser simplex (los datos son enviados sólo en una dirección), semidúplex (ambos nodos transmiten información, pero solo uno a la vez) o dúplex completo (ambos nodos pueden transmitir datos en simultáneo).

Referencias bibliográficas

- Álvarez Corredera, P. (2021). Nim: Un lenguaje de programación inspirado en Delphi y Python. Recuperado de <https://ciberninjas.com/nim/>
- Arduino. (s.f.). Acerca de Arduino. Recuperado de <https://www.arduino.cc/en/about#what-is-arduino>
- Arduino core for the ESP32. [GitHub]. s.f. Recuperado de <https://github.com/espressif/arduino-esp32/tree/master>
- AsyncStorage. [GitHub]. s.f. Recuperado de <https://github.com/react-native-async-storage/async-storage>
- Aws. (s.f.) ¿Qué es IoT? Recuperado de <https://aws.amazon.com/es/what-is/iot/#:~:text=El%20t%C3%A9rmino%20IoT%2C%20o%20Internet,como%20entre%20los%20propios%20dispositivos.>
- Barragán, A. (2022, marzo 1). Qué es Deno y qué lo hace diferente. [Publicación en blog]. Recuperado de <https://openwebinars.net/blog/que-es-deno-y-que-lo-hace-diferente/>
- Bello, E. (2022, mayo 16). Bases de datos NoSQL vs SQL: Qué son y en qué se diferencian [Publicación en blog] Recuperado de <https://www.iebschool.com/blog/bases-de-datos-nosql-vs-sql-big-data/>
- Blanch, A. (2021, abril 21). ¿Qué es Deno? [Publicación en blog]. Recuperado de <https://www.arsys.es/blog/deno#:~:text=Deno%20es%20una%20plataforma%20para,aplicaciones%20de%20backend%20en%20general.>
- Bluetooth. (2023). Acerca de Bluetooth. Recuperado de <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- Castro, F. (2022, enero 31). La selección uruguaya masculina de handball clasificó al Mundial al terminar cuarta en el Sur Centroamericano. *La Diaria*. (s.p.). Recuperado de <https://ladiaria.com.uy/deporte/articulo/2022/1/la-seleccion-uruguay-masculina-de-handbol-clasifico-al-mundial-al-terminar-cuarta-en-el-sur-centroamericano/>
- Corrales, J.A. (2019, agosto 2). Interfaz de usuario o UI: ¿qué es y cuáles son sus características? Recuperado de <https://rockcontent.com/es/blog/interfaz-de-usuario/>
- Dell (2021). ¿Qué es la memoria RAM)? Recuperado de <https://www.dell.com/support/kbdoc/es-uy/000148441/what-is-memory-ram>

Descubrearduino.com. (s.f.). MQTT, Qué es, ¿cómo se puede usar? y Cómo funciona. Recuperado de <https://descubrearduino.com/mqtt-que-es-como-se-puede-usar-y-como-funciona/>

Deymar, A. (2023). SQLite vs MySQL: ¿Cuál es la diferencia? Recuperado de <https://www.hostinger.es/tutoriales/sqlite-vs-mysql-cual-es-la-diferencia>

Diamond, P. (2020). Almacenamiento en la nube frente a servidores locales: Nuevo puntos a tener en cuenta. Recuperado de <https://www.microsoft.com/es-ar/microsoft-365/business-insights-ideas/resources/cloud-storage-vs-on-premises-servers#:~:text=El%20almacenamiento%20local%20significa%20que,o%20un%20socio%20de%20TI.>

Electrónica.Uy. (2018). Funcionalidades del ESP8266. Recuperado de <https://www.electronica.com.uy/robotica/modulos-comunicacion/m%C3%B3dulo-wi-fi-esp8266-esp-01-detalle.html>

Espressif. (s.f.). Acerca de la compañía detrás del ESP32 y el ESP8266. Recuperado de <https://www.espressif.com/en/company/about-espressif>

Espressif. (s.f.). ¿Qué es el ESP32? Recuperado de <https://www.espressif.com/en/products/socs/esp32>

Firebase. (s.f.). ¿Qué es y para qué sirve Firebase?. Recuperado de <https://firebase.google.com/docs?hl=es-419>

Flutter. (s.f.). ¿Qué es y cómo funciona internamente?. Recuperado de <https://docs.flutter.dev/resources/architectural-overview>

Grokhotkov, I. (2017). ESP8266 Arduino Core's documentation. Recuperado de <https://arduino-esp8266.readthedocs.io/en/latest/>

Guerra Carmenate, J. (s.f.). ¿Para qué sirve el ESP32? Recuperado de <https://programarfacil.com/esp8266/esp32/>

Handball Goalkeepertraining. (2019,08, 24). Handball Goalkeeper Training - REACTION LIGHTS + SLIDE. [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=qHKyrBMJTPs>

Holcombe J. (2023, agosto 21). Aprende Deno y Ve Más Allá de Node.js. [Publicación en blog]. Recuperado de <https://kinsta.com/es/blog/deno-js/>

IBM (2021). Tipos de archivos. Recuperado de <https://www.ibm.com/docs/es/aix/7.1?topic=files-types>

IHF. (s.f.). ¿Qué es la IHF? Recuperado de <https://www.ihf.info/about/congress>

IHF. (s.f.). ¿Quiénes conforman la IHF? Recuperado de <https://www.ihf.info/federations>

IHF. (s.f.). ¿Cuándo se fundó? Recuperado de <https://www.ihf.info/federations>

ljidola, G. (2022, diciembre 9). PostgreSQL, DynamoDB, & MongoDB: Choosing a Database for a Node.js App. [Publicación en blog] Recuperado de <https://www.bitovi.com/blog/postgresql-dynamodb-mongodb-choosing-a-database-for-a-node.js-app#:~:text=If%20you're%20looking%20for,below%2C%20especialmente%20in%20the%20Node.>

Ilimit (2020, abril 28). Bases de Datos SQL vs NOSQL: Diferencias, Ventajas y Mitos [Publicación en blog]. Recuperado de <https://www.imit.com/blog/base-de-datos-sql-nosql/#:~:text=Si%20las%20necesidades%20de%20procesamiento,utilizar%20bases%20de%20datos%20SQL.&text=Si%20el%20crecimiento%20de%20la,de%20datos%20no%20relacionales%20NoSQL.>

Kravets, I. (2022). ESPAsyncWebServer. [GitHub]. Recuperado de <https://github.com/me-no-dev/ESPAsyncWebServer>

Lummic. (s.f.). ¿Qué es? Recuperado de <https://lummic.com/es>

Lumic. (2023). Lummic - ¿Cómo funciona su aplicación móvil?. Recuperado de <https://play.google.com/store/apps/details?id=com.sys.lummic&pli=1>

Martins, J. (2022). Guía rápida para definir el alcance de un proyecto. Recuperado de <https://asana.com/es/resources/project-scope>

MQTT. (s.f.). MQTT Essentials. Recuperado de <https://www.hivemq.com/mqtt-essentials/>

MongoDB. (s.f.). Acerca de organizaciones que utilizan esta tecnología. Recuperado de <https://www.mongodb.com/who-uses-mongodb>

MongoDB. (s.f.). Qué es y cómo funciona MongoDB. Recuperado de <https://www.mongodb.com/docs/>

Nim (lenguaje de programación). (s.f.). En Hmong. Recuperado 25 oct., 2023, de [https://hmong.es/wiki/Nim_\(programming_language\)](https://hmong.es/wiki/Nim_(programming_language))

Node.js. (s.f.). Sobre Node.js. Recuperado de <https://nodejs.org/en/about>

Penalva, J. (2022). NFC: qué es y para qué sirve en este 2023. Recuperado de <https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve>

Pérez Porto, J. y Merino, M. (2022). Balonmano - Qué es, orígenes, definición y concepto. Recuperado de <https://definicion.de/balonmano/>

Pérez Porto, J. y Merino, M. (2022). WLAN - Qué es, ventajas, definición y concepto. Recuperado de <https://definicion.de/wlan/>

PostgreSQL. (s.f.). Qué es y por qué utilizar esta tecnología. Recuperado de <https://www.postgresql.org/about/>

React Native. (s.f.). ¿Qué es ? Recuperado de <https://reactnative.dev/>

- Redacción Runners World (2022). Entrena tu velocidad, reflejos (y memoria) con este sistema de luces de reacción. Recuperado de <https://www.runnersworld.com/es/training/a41553282/entrenamiento-velocidad-reflejos-memoria-luces-reaccion/>
- Robledano, A. (2019, setiembre 24). ¿Qué es MySQL? Características y ventajas [Publicación en blog]. Recuperado de <https://openwebinars.net/blog/que-es-mysql/>
- Rómmel, F. (s.f.). SQLite: La Base de Datos Embebida. Recuperado de <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- Rosas, L. (2018). Light Training: Free. Acerca de esta app. Recuperado de https://play.google.com/store/apps/details?id=com.lightraining.lightrainingfree&hl=es_419&gl=US&pli=1
- O'Reilly. (s.f.). What is React Native? Recuperado de <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>
- O'Reilly. (s.f.). The SQL Standard. Recuperado de <https://www.oreilly.com/library/view/sql-in-a/1565927443/ch01s03.html>
- OST Team. (2022). How Many Bluetooth Connections at Once for Devices? Recuperado de <https://us.soundcore.com/blogs/headphones/how-many-bluetooth-connections-at-once-across-devices>
- Salazar, J. (2016). Redes inalámbricas. TechPedia. Recuperado de https://upcommons.upc.edu/bitstream/handle/2117/100918/LM01_R_ES.pdf
- Trucoteca. (s.f.). ¿Quién inventó el lenguaje de programación Nim? Recuperado de <https://trucoteca.com/quien-invento-el-lenguaje-de-programacion-nim-2/>
- UNIR. (2021, 06, 17). NoSQL vs. SQL: características, diferencias y contextos de uso de estas tecnologías de SGBD. Recuperado de <https://www.unir.net/ingenieria/revista/nosql-vs-sql/>
- W3Schools. (s.f.). HTTP Request Methods. Recuperado de https://www.w3schools.com/tags/ref_httpmethods.asp

Anexo - Diagrama de ilustración de la base de datos

un **Usuario** que crea las **Secuencias** para los **Jugadores**, en cada secuencia se define el orden de los leds y el tiempo que va a estar encendido.
De cada jugador se va a guardar los **resultados** de cada sesion (el orden de los leds y cuanto tiempo demoró en pulsarlo)

Clases

- Usuario
- Jugador
- Secuencia
- Resultado

