



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Tremün: Plataforma para entrenamiento cognitivo aplicado al deporte

**AUTORES:** Delgado De Leon, Federica - Cortés, Ramiro

**DIRECTOR:** Fava, Laura Andrea

**CODIRECTOR:** Vilches Antao, Diego

**ASESOR PROFESIONAL:**

**CARRERA:** Licenciatura en Sistemas

### Resumen

*El rol de la neurociencia y la psicología cognitiva en la tarea de entender, predecir y potencialmente mejorar el desempeño en deportes de élite, es un área que ha recibido un creciente interés en los últimos años. Este enfoque de trabajo, bajo el nombre de "Entrenamiento Cognitivo", apunta a desarrollar un rango de habilidades como la atención y la toma de decisiones, a partir del entrenamiento de funciones cognitivas centrales. La oferta de dispositivos aplicables a este tipo de entrenamiento, sin embargo, es baja y de difícil acceso por su elevado costo. A partir de esta situación, se diseñó y desarrolló una plataforma de entrenamiento cognitivo aplicado al deporte con el objetivo de democratizar el acceso a dicha tecnología mas allá de la élite deportiva.*

### Palabras Clave

*Entrenamiento cognitivo - tecnología - deporte - psicología cognitiva.*

### Conclusiones

*Se construyó una plataforma de entrenamiento cognitivo aplicada al deporte para fomentar el desarrollo de las capacidades neurocognitivas de los deportistas, mediante la ejecución de distintos tipos de rutinas de entrenamiento. Tremün surge como una alternativa de calidad y bajo costo ante las opciones vigentes en el mercado en cuanto a tecnologías de entrenamiento cognitivo, con el objetivo de democratizar el acceso a este tipos de dispositivos mas allá del entrenamiento profesional de élite que hoy goza de la exclusividad del acceso a los mismos.*

### Trabajos Realizados

*Inicialmente se investigó acerca de la psicología cognitiva y del entrenamiento de las capacidades cognitivas aplicado al deporte. Se relevó el estado del arte en plataformas de entrenamiento cognitivo para identificar necesidades. Se analizaron diversas opciones para definir la propuesta tecnológica, y finalmente se desarrolló una plataforma orientada al entrenamiento cognitivo con una aplicación móvil multiplataforma y una controladora que se comunica con un conjunto de nodos. Además, se realizaron encuestas respecto al uso de Tremün y otras herramientas de entrenamiento cognitivo y se sacaron conclusiones.*

### Trabajos Futuros

*Se proponen diversas líneas de investigación y trabajos a futuro sobre la plataforma desarrollada. Entre estos se destacan la posibilidad de aumentar los tipos de rutinas de entrenamiento precargadas disponibles en la plataforma, la exportación de los datos recolectados en diversos formatos (.csv, .xlsx, .json), y la posibilidad de compartir rutinas de entrenamiento entre entrenadores y preparadores físicos. Por último, podría explorarse la integración con nuevos dispositivos para entrenamientos específicos por deporte.*

Fecha de la presentación: Octubre 2020

UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE GRADO

---

**Tremün: Plataforma para  
entrenamiento cognitivo aplicado al  
deporte**

---



*Autores:*  
Federica Delgado De Leon  
Ramiro Cortes

*Director:*  
Laura Andrea Fava  
*Co-Director:*  
Diego Vilches Antao

Octubre 2020

# Índice general

<b>1. Introducción</b>	<b>6</b>
1.1. Objetivo . . . . .	6
1.2. Motivación . . . . .	6
1.3. Resultados esperados . . . . .	9
1.4. Organización del documento . . . . .	9
<b>2. Tecnología y deporte</b>	<b>11</b>
2.1. Historia de la tecnología en el deporte . . . . .	11
2.2. Psicología cognitiva y deporte . . . . .	13
2.3. Entrenamiento Cognitivo . . . . .	15
2.3.1. Descripción . . . . .	15
2.3.2. Enfoques de EC . . . . .	15
2.3.3. Aplicaciones del EC . . . . .	16
2.3.3.1. Entrenamiento compensatorio . . . . .	16
2.3.3.2. Entrenamiento restaurativo . . . . .	16
2.3.3.3. Entrenamiento de perfeccionamiento . . . . .	17
<b>3. Estado del arte</b>	<b>18</b>
3.1. Búsqueda y selección de aplicaciones . . . . .	18
3.2. Led Trainer . . . . .	18
3.3. Fitlight . . . . .	21
3.4. Neural Trainer . . . . .	22
3.5. SmartGoals . . . . .	23
3.6. Análisis comparativo . . . . .	26
3.6.1. Funcionalidades . . . . .	26
3.6.2. Características técnicas . . . . .	26
3.7. Conclusión . . . . .	27
<b>4. Análisis de herramientas</b>	<b>28</b>
4.1. La aplicación móvil . . . . .	28
4.1.1. Desarrollo de aplicaciones móviles . . . . .	29

4.1.1.1.	La tecnología móvil . . . . .	29
4.1.1.2.	Sistemas operativos móviles . . . . .	30
4.1.1.3.	Enfoques de desarrollo y sus características . . . . .	30
4.1.2.	React Native . . . . .	32
4.1.2.1.	Características generales . . . . .	33
4.1.2.2.	Expo CLI vs React Native CLI . . . . .	34
4.1.3.	Persistencia de datos . . . . .	35
4.1.3.1.	Realm . . . . .	35
4.1.3.2.	Firebase . . . . .	36
4.1.3.3.	SQLite . . . . .	36
4.1.3.4.	PouchDB . . . . .	36
4.1.3.5.	Redux y Async Storage . . . . .	37
4.1.3.6.	Análisis comparativo . . . . .	37
4.2.	El controlador . . . . .	39
4.2.1.	Raspberry Pi . . . . .	39
4.2.2.	Golang vs Python . . . . .	39
4.2.3.	Go . . . . .	40
4.3.	Los nodos . . . . .	41
4.4.	La comunicación . . . . .	42
4.4.1.	Red Wi-Fi . . . . .	42
4.4.2.	El protocolo HTTP . . . . .	42
4.4.2.1.	REST . . . . .	43
4.4.2.2.	SSE vs WebSockets . . . . .	43
4.4.3.	Protocolo de comunicación entre el controlador y los nodos . . . . .	44
<b>5.</b>	<b>Diseño de la solución</b> . . . . .	<b>45</b>
5.1.	Vista general de la plataforma . . . . .	45
5.2.	La aplicación móvil . . . . .	47
5.2.1.	Descripción general . . . . .	47
5.2.2.	Login . . . . .	47
5.2.3.	Rutinas de entrenamiento . . . . .	48
5.2.4.	Rutina aleatoria . . . . .	48
5.2.4.1.	Descripción . . . . .	49
5.2.4.2.	Ejemplos de aplicación . . . . .	51
5.2.5.	Rutinas precargadas . . . . .	58
5.2.5.1.	Rutina selectiva . . . . .	58
5.2.5.2.	Rutina por equipos . . . . .	63
5.2.5.3.	Rutina de velocidad . . . . .	65
5.2.6.	Rutinas personalizadas . . . . .	69
5.3.	El controlador . . . . .	74
5.4.	Los nodos . . . . .	76

<b>6. Pruebas de campo</b>	<b>78</b>
6.1. Introducción . . . . .	78
6.2. Conformación y análisis de la encuesta . . . . .	79
6.3. Análisis y conclusiones . . . . .	84
<b>7. Conclusiones y trabajos futuros</b>	<b>85</b>
7.1. Conclusiones . . . . .	85
7.2. Trabajos futuros . . . . .	87

# Índice de figuras

3.1. Inicio de la aplicación . . . . .	19
3.2. Información de módulos . . . . .	19
3.3. Opciones de entrenamientos . . . . .	20
3.4. Rutina rápida . . . . .	20
3.5. Rutina múltiple . . . . .	20
3.6. Rutinas predefinidas . . . . .	20
3.7. Configuración manual . . . . .	21
3.8. Resultados . . . . .	21
3.9. Configuración de rutinas . . . . .	24
3.10. Rutina por hits . . . . .	25
3.11. Rutina por tiempo . . . . .	25
3.12. Listado de resultados . . . . .	25
3.13. Detalle de resultados . . . . .	25
4.1. Nodo . . . . .	42
5.1. Vista general de la plataforma . . . . .	46
5.2. Pantalla de inicio de sesión . . . . .	48
5.3. Pantalla de configuración . . . . .	49
5.4. Pantalla de resultados . . . . .	50
5.5. Rutina Aleatoria. Ejemplo 1. Paso 1. . . . .	51
5.6. Rutina Aleatoria. Ejemplo 1. Paso 2. . . . .	52
5.7. Rutina Aleatoria. Ejemplo 1. Paso 3. . . . .	52
5.8. Rutina Aleatoria. Ejemplo 2. Paso 1. . . . .	53
5.9. Rutina Aleatoria. Ejemplo 2. Paso 2. . . . .	54
5.10. Rutina Aleatoria. Ejemplo 2. Paso 3. . . . .	54
5.11. Rutina Aleatoria. Ejemplo 3. Paso 1. . . . .	55
5.12. Rutina Aleatoria. Ejemplo 3. Paso 2. . . . .	56
5.13. Rutina Aleatoria. Ejemplo 4. Paso 1. . . . .	57
5.14. Rutina Aleatoria. Ejemplo 4. Paso 2. . . . .	57
5.15. Pantalla inicial. . . . .	58
5.16. Pantalla inicial. . . . .	58

5.17. Rutina Selectiva. Pantalla de resultados. . . . .	60
5.18. Rutina Selectiva. Ejemplo 1. Paso 1. . . . .	61
5.19. Rutina Selectiva. Ejemplo 1. Paso 2. . . . .	62
5.20. Rutina Selectiva. Ejemplo 1. Paso 3. . . . .	62
5.21. Pantalla de inicio. . . . .	63
5.22. Pantalla de inicio. . . . .	63
5.23. Rutina por equipos. Pantalla de resultados. . . . .	64
5.24. Rutina por equipos. Ejemplo 1. Paso 3. . . . .	65
5.25. Rutina por equipos. Ejemplo 1. Paso 3. . . . .	65
5.26. Rutina de Velocidad. Pantalla inicial. . . . .	67
5.27. Rutina de Velocidad. Pantalla de resultados. . . . .	68
5.28. Rutina de Velocidad. Ejemplo 1. Paso 1. . . . .	69
5.29. Listado de rutinas del usuario . . . . .	70
5.30. Formulario de alta . . . . .	71
5.31. Formulario de alta . . . . .	71
5.32. Formulario de alta . . . . .	71
5.33. Formulario de alta . . . . .	71
5.34. Formulario de alta . . . . .	72
5.35. Formulario de alta . . . . .	72
5.36. Listado de rutinas del usuario . . . . .	73
5.37. Detalle de la rutina . . . . .	74
5.38. Acciones disponibles . . . . .	74
6.1. Encuesta. Parte 1. . . . .	79
6.2. Encuesta. Parte 2. . . . .	80
6.3. Encuesta. Parte 3 . . . . .	81
6.4. Obstáculos para la utilización de dispositivos de entrenamiento cognitivo en Argentina . . . . .	82

# Capítulo 1

## Introducción

En este capítulo se presentará la motivación, los objetivos y los resultados esperados de la tesina de grado “Tremün: Plataforma para entrenamiento cognitivo aplicado al deporte”.

### 1.1. Objetivo

El objetivo principal de esta tesina es el desarrollo de una herramienta tecnológica con aplicación en el deporte, que estimule el entrenamiento físico y cognitivo de deportistas tanto individual como colectivo, cuyo funcionamiento se adapte a las condiciones particulares de los entornos de entrenamiento deportivo habituales (espacios grandes, abiertos, sin conectividad a internet). Partiendo de la utilización de esta herramienta, se espera explorar el potencial de la inclusión de la tecnología para entrenamiento de diferentes deportes con el objetivo de mejorar resultados y rendimientos en competencia, así como también prevenir y rehabilitar lesiones.

A través del mencionado desarrollo se busca también democratizar el acceso a nuevas tecnologías de entrenamiento deportivo, de calidad y a la altura de las posibilidades tecnológicas actuales, más allá del ámbito profesional y de élite (que hoy gozan del privilegio exclusivo del acceso a las mismas), alcanzando también el deporte amateur.

### 1.2. Motivación

A lo largo de la historia es posible identificar una tendencia invariable en los procesos de aprendizaje y entrenamiento en el deporte, abordados desde una perspectiva conductista que los basa en la repetición de movimientos con

el objetivo de mejorar las aptitudes técnicas del deportista mediante la mecanización[1].

Sin embargo, en los últimos años existen algunos sucesos que merecen ser destacados, y que posibilitaron una ramificación de los ya mencionados procesos de aprendizaje, no sólo modificando la perspectiva desde la que se afrontan, sino también poniendo el foco de entrenamiento en el desarrollo de nuevas cualidades que resultan también de fundamental importancia para la mejora del rendimiento.

En primer lugar, la aparición del Internet de las Cosas (IoT) y su aplicación en el ámbito deportivo dio lugar a un universo de posibilidades totalmente novedoso con respecto a las formas de entrenamiento (principalmente a nivel profesional y de élite) permitiendo, gracias al uso de diversos tipos de sensores (GPS, sensores de proximidad, acelerómetros), el acceso a nuevas herramientas de trabajo que disponibilizan mayor información sobre el rendimiento individual y colectivo.

Por otro lado, el incremento de investigaciones en el área de la neurociencia cognitiva se ha acelerado y expandido a diversas disciplinas, entre las que se encuentran por supuesto el deporte y la actividad física. Son muchos los estudios que afirman que, para desarrollarse en los más altos niveles de competencia, es requisito indispensable en los deportistas el desarrollo de funciones cognitivas tales como la atención, la toma de decisiones y la memoria de trabajo, principalmente en entornos de estrés.

Introducido el contexto en el cual se encuadra el trabajo que se propone, resulta pertinente responder dos interrogantes fundamentales para comprender la importancia del presente desarrollo: ¿qué es el entrenamiento cognitivo aplicado al deporte? y ¿por qué es importante disponer de una plataforma diseñada específicamente para dicho fin?

El entrenamiento cognitivo aplicado al deporte tiene como objetivo el desarrollo de un conjunto de habilidades como la atención y la toma de decisiones a través del entrenamiento enfocado en funciones cognitivas centrales. Es posible diferenciar dos acercamientos con respecto a la metodología de entrenamiento para el desarrollo cognitivo: el entrenamiento de contexto específico, donde las ejercitaciones se ejecutan sobre acciones o situaciones propias del deporte o competición en cuestión, como puede ser el entrenamiento con videos para mejorar la lectura posicional de un rival antes de ejecutar una acción; y el entrenamiento de contexto general, que afirma que el entrenamiento y desarrollo de una función cognitiva en un entorno general se traduce directamente a un resultado favorable en competencia al momento de utilizar la función cognitiva en cuestión.

Si se analiza la importancia del entrenamiento cognitivo en el deporte, es

posible encontrar que existe una necesidad real y creciente de nuevas herramientas disponibles para su aplicación.

Ante todo, parece evidente que esta demanda nace desde el campo deportivo, siendo los clubes y entrenadores los primeros interesados en contar con la posibilidad de desarrollar estas aptitudes en sus competidores. Si bien es cierto que existen algunas propuestas en la industria que apuntan a satisfacer este requerimiento (Fitlight [2], SmartGoals [3], Led Trainer [4]), la posibilidad de acceso suele estar restringida a los niveles de competición más alto, siendo únicamente alcanzados por este beneficio quienes cuentan con un gran poder adquisitivo para costear los gastos de adquisición y utilización de los productos. Creemos que resulta de fundamental importancia acortar la brecha entre los recursos disponibles en los más altos niveles de profesionalismo y aquellos accesibles para quienes se desempeñan en la esfera del deporte amateur, ya que creemos en este último como trascendental por su función social y su contribución a la integración.

No obstante, existe otro motivo que merece ser resaltado, y es la creciente demanda por parte de la academia de más y mejores mecanismos de entrenamiento cognitivo[5]. Si bien la importancia del desarrollo cognitivo está lejos de ser una novedad, sí lo es su aplicación al campo del entrenamiento deportivo. A pesar de que se trata de una disciplina joven con mucho por investigar, y que las opciones que existen en el mercado son escasas y con un elevado costo de adquisición, no es difícil imaginar un crecimiento en los estudios sobre el potencial de este tipo de entrenamientos a corto plazo.

Para llevar a cabo la propuesta, se tendrán en cuenta las demandas tecnológicas actuales, a fin de construir una herramienta que se adapte no sólo a las necesidades desde el punto de vista del entrenamiento, sino también a las posibilidades de los usuarios finales. Es importante aclarar que el contexto donde se utilizará esta aplicación, tiene características particulares que debemos considerar, como el hecho de ser utilizada en espacios abiertos y amplios, lo que implica falta de corriente eléctrica, carencia de conectividad wi-fi o incluso de cobertura de señal por parte de las empresas proveedoras. Se presenta entonces el desafío de planificar y ejecutar una solución tecnológica de bajo costo y fácil acceso para la población sin resignar que a su vez se encuentre a la altura de los requerimientos y restricciones que impone el medio.

Teniendo en cuenta todas las consideraciones mencionadas proponemos un desarrollo que posibilite la ejecución de rutinas diseñadas específicamente para recuperar y mejorar las capacidades cognitivas de los deportistas, que pueda utilizarse en diferentes escenarios de entrenamiento, y que ponga el foco en la importancia de lograr hacer llegar la herramienta a todos los niveles de competencia.

### 1.3. Resultados esperados

Culminado el proceso de desarrollo de la tesina, se espera:

- Alcanzar una plataforma integral de bajo costo que permita la creación y ejecución de rutinas de entrenamiento deportivo que colabore con el desarrollo cognitivo y neuromuscular de los deportistas.
- Democratizar el acceso a tecnologías y formas de entrenamiento que en la actualidad son accesibles únicamente por quienes forman parte de la élite deportiva.
- Realizar un conjunto de pruebas en diversos deportes e instituciones con la finalidad de evaluar el grado de aceptación alcanzado por el desarrollo, desde la perspectiva de los entrenadores.

### 1.4. Organización del documento

Esta tesina está organizada en capítulos con los siguientes contenidos:

- Capítulo 2 - Tecnología y deporte: Se realiza un breve repaso histórico de la relación entre tecnología y deporte, y se analiza el vínculo entre la psicología cognitiva y el deporte, que dio lugar al surgimiento del entrenamiento cognitivo.
- Capítulo 3 - Estado del arte: Se analizan distintas herramientas similares a la planteada en esta tesina y se comparan bajo distintos criterios.
- Capítulo 4 - Análisis de herramientas: Se describen las tecnologías analizadas para el desarrollo de la plataforma, para su utilización en los distintos componentes de hardware y software.
- Capítulo 5 - Diseño de implementación: Se describe el diseño de la solución partiendo del análisis realizado en el capítulo anterior, que involucra:
  - Implementación de una controladora sobre una Raspberry Pi
  - Aplicación móvil
  - Dispositivos con luces LED y sensor de proximidad
  - Integración con el sistema Machi
- Capítulo 6 - Pruebas de campo: Se describen las pruebas iniciales realizadas con la plataforma y se muestran los resultados relevados mediante encuestas.

- Capítulo 7 - Conclusiones y trabajos futuros: Se detallan las conclusiones obtenidas luego del proceso de desarrollo y se proponen futuras líneas de investigación.

# Capítulo 2

## Tecnología y deporte

### 2.1. Historia de la tecnología en el deporte

La tecnología y el deporte han estado articuladas durante muchísimos años en formas muy diversas, siendo este vínculo particularmente indispensable en competiciones de alto rendimiento. La aplicación de dispositivos tecnológicos a las circunstancias de entrenamiento, suele ser uno de los puntos más evidentes de esta relación, especialmente a partir del surgimiento de sofisticados dispositivos de medición de variables físicas como es el caso de los dispositivos con Sistema de Posicionamiento Global o GPS. La tecnología deportiva sin embargo no solo antecede a estos dispositivos, sino que entra en contacto con el deporte más allá del entrenamiento en sí.

Desde los avances en equipamiento técnico para mejorar la transmisión de los espectáculos deportivos, pasando por la modificación de reglamentos a través de la inclusión de la cámaras y sensores (como es el caso del Ojo de Halcon en el tenis), hasta la reducción en los plazos de recuperación de lesiones o los avances en indumentaria y calzado, la tecnología ha inundado el universo deportivo generando cambios estructurales en sus prácticas y sus protagonistas (no sólo deportistas sino también médicos, psicólogos, entrenadores, preparadores físicos y dirigentes). Es posible categorizar, en seis grandes grupos, distintos tipos de aplicación de la tecnología en deporte[6]:

- Self-Technologies: Esta clase de tecnologías tiene la capacidad de alterar (en ocasiones permanentemente) el desempeño físico o psicológico del deportista. Las sustancias prohibidas para potenciar el desempeño son el caso más emblemático de este grupo, en el que también destacan los procedimientos quirúrgicos, las prótesis o la ingeniería genética.
- Landscape Technologies: Este grupo de tecnologías engloba a aquellas

que impactan el entorno o ambiente en el que se desarrolla la práctica deportiva. Dentro del mismo podemos observar distintos niveles de impacto en la práctica deportiva en sí misma y en los atletas que la practican, ya que abarca desde pantallas gigantes para mejorar la experiencia de los espectadores hasta césped sintético en el campo de juego, o distintas superficies para la practica de tenis o automovilismo.

- **Implement Technologies:** Estas tecnologías son aquellas relacionadas a la indumentaria o cualquier otro artefacto que involucre un contacto directo con los atletas. Calzados de alta tecnología, raquetas, cañas de pesca, pelotas, trajes de baño, cascos o palos de golf son algunos ejemplos destacables.
- **Rehabilitive Technologies:** Esta categoría hace referencia al desarrollo de sustancias y procedimientos que acompañan al atleta en procesos de rehabilitación de lesiones moderadas o severas. Incluye también la medicina utilizada por atletas sanos para poder tolerar o sobrellevar el exigente ritmo de entrenamiento de los entornos de alto rendimiento.
- **Movement Technologies:** Se refiere a aquellos dispositivos y procedimientos que están diseñados para evaluar la forma y la eficiencia del cuerpo de un atleta, siendo quizás el análisis de video uno de los casos más representativos (aunque existen instrumentos mucho más sofisticados que proveen información computarizada de la biomecánica del atleta). Por otro lado, dentro de esta categoría también se agrupan otras tecnologías de monitoreo del movimiento utilizadas para afinar la aplicación de los reglamentos, como es el caso de las cámaras ubicadas sobre la línea del arco de fútbol para determinar si la pelota cruza la línea de gol.
- **Database Technologies:** Este grupo incluye todas las innovaciones relacionadas a la computación que permiten, a través de la recopilación de datos, conocer en detalle toda la información necesaria sobre un atleta. Tecnologías como el GPS o monitores de ritmo cardiaco permiten monitorear continuamente el progreso de importantes parámetros fisiológicos y de performance.

Con el foco puesto en la actualidad, existen dos fenómenos que han revolucionado el mundo del deporte en los últimos años. Por un lado, con los dispositivos de GPS como su máximo exponente, la irrupción del IoT multiplicó los mecanismos de recolección de información poniendo a disposición un complejo y valioso universo de datos sobre los atletas. Por el otro, en el marco de la comúnmente denominada “década del cerebro”, los avances en materia de

neurociencia pusieron en el centro de la escena el concepto de entrenamiento cognitivo, dando inicio a una nueva rama de la industria tecnológica deportiva: el desarrollo de instrumentos de entrenamiento, medición y procesamiento de capacidades neurocognitivas. A continuación se profundizará este último fenómeno, relacionado con la psicología y el entrenamiento cognitivo, por ser tema central de nuestra tesina.

## 2.2. Psicología cognitiva y deporte

La psicología cognitiva fue definida por Solso en 1995 como el estudio científico de lo que piensa la mente, cuya representación más clara es el procesamiento de la información. Se centra en cómo atendemos y captamos la información sobre el mundo, cómo es guardada en la memoria por el cerebro, y cómo este conocimiento es usado para resolver problemas, pensar, y formular juicios.

La percepción, en este marco, es entendida como un proceso primario de elaboración de la información, siendo ésta un factor determinante en toda acción deportiva. Los actos perceptivos implican una relación de interacción física entre el medio y el sujeto a través de los sentidos. El canal visual es el más utilizado en el procesamiento de información de habilidades de carácter abierto, y por tanto el canal dominante en el proceso de optimización del rendimiento deportivo.

Dado que el ser humano tiene una capacidad limitada en cuanto al procesamiento de la información del entorno, el deportista se ve obligado a reducir y seleccionar solo aquellas partes relevantes del ambiente. La toma de decisiones es un factor determinante en el rendimiento, y la predicción es una de las herramientas más valiosas para afrontar con éxito la práctica deportiva.

Algunos deportes de balón son tan rápidos que obligan al jugador a valorar la trayectoria del balón y emitir la respuesta adecuada en muy poco tiempo. Además la precisión requerida es de unos pocos centímetros en el espacio y unos pocos milisegundos en el tiempo (Land, 2006). Cuando el medio es inestable, y el tiempo del que se dispone para responder es bajo, el deportista debe decidir con rapidez, siempre en función de los eventos externos. En estas condiciones cabe la posibilidad de que el tiempo de reacción, sumado al tiempo de movimiento, superen ese tiempo total del que el jugador dispone para desarrollar su acción de forma efectiva.

En este caso el deportista puede reducir el tiempo de dos formas no excluyentes: reduciendo la duración del movimiento o reduciendo el tiempo de reacción. Mientras que para el primer caso resulta esencial el entrenamiento físico, para la segunda opción el jugador necesita anticiparse, y encontrar índi-

ces que le permitan no cometer errores al reducir el tiempo de respuesta. A continuación se listan algunas de las variables neurocognitivas que deberían ser tenidas en cuenta a la hora de diagramar procesos de entrenamiento cognitivo aplicado al deporte[1]:

- **Atención sostenida:** los recursos atencionales que un deportista desplaza de manera consciente sobre una actividad. Los deportistas con dificultades en esta función presentan serios inconvenientes para realizar tareas que demanden monitoreo, vigilancia y altos niveles de alerta.
- **Memoria verbal:** es la capacidad de memorizar información de tipo conceptual, como la información que es suministrada por un entrenador en una clase magistral: palabras, frases, definición de conceptos e instrucciones verbales. Las dificultades que se suelen atribuir a la baja capacidad de esta función, se encuentran determinadas por el alto impacto que tienen en el rendimiento deportivo, ya que el bajo procesamiento, almacenamiento y evocación generan deficiencias en la incorporación y consolidación de nueva información.
- **Memoria Visoespacial:** es la capacidad de recordar la ubicación de un objeto en un espacio determinado, la distribución visual de los movimientos en la resolución de un problema motriz, las alteraciones en esta función repercute en el reconocimiento de personas o elementos en el campo de juego.
- **Velocidad de procesamiento:** es una función predominantemente de los lóbulos frontales, se define como la capacidad que un deportista tiene para procesar información de manera veloz y eficiente. Ciertos deportistas suelen tomarse más tiempo para dar una respuesta frente a una pregunta de un entrenador, lo cual implica alteraciones en el procesamiento de información y pérdida de tiempo en situaciones críticas.
- **Control inhibitorio:** es la capacidad que una persona tiene de inhibir respuestas (conductas), que no son las adecuadas, en favor de respuestas que resultarían ser más apropiadas. Entre los deportistas, es frecuente hallar sujetos que no logran un pleno control sobre su comportamiento y constantemente no suelen pensar en las consecuencias de sus actos o decisiones, lo que lleva a cometer errores por falta de reflexión en situaciones que requieren de mayor análisis.
- **Flexibilidad cognitiva:** se refiere a la habilidad para mirar objetos y eventos desde muchos puntos de vista, particularmente cuando se enfrenta a

un nuevo contexto. Los deportistas que tienen alteraciones en esta función, suelen presentar dificultades para entender variantes en la solución de un problema y entender los puntos de vista de otras personas.

- Auto-monitoreo: es la habilidad para monitorear y regular los propios comportamientos que se encuentran en ejecución. Los deportistas que son incapaces de automonitorearse, pueden no darse cuenta de los errores que cometen, o pueden ser capaces de identificarlos, pero no de corregirlos.

## **2.3. Entrenamiento Cognitivo**

### **2.3.1. Descripción**

El rol de la neurociencia y la psicología cognitiva en la tarea de entender, predecir y potencialmente mejorar el desempeño en deportes de elite, es un área que ha recibido un creciente interés en los últimos años. Este enfoque de trabajo, bajo el nombre de “Entrenamiento Cognitivo”, apunta a desarrollar un rango de habilidades como la atención y la toma de decisiones, a partir del entrenamiento de funciones cognitivas centrales.

### **2.3.2. Enfoques de EC**

A la hora de analizar el entrenamiento cognitivo (EC) es posible catalogarlo en dos grandes grupos: el EC de dominio general, que pone el foco en el desarrollo de funciones cognitivas centrales aplicables a una multitud de tareas, como por ejemplo el entrenamiento general de la memoria visoespacial; y el EC de contexto específico, que entrena una habilidad cognitiva aplicada a una tarea en un contexto específico, como puede ser el entrenamiento de la capacidad de anticipación de un arquero a la ejecución de un penal. La plataforma Tremün fue desarrollada teniendo como guía el primer enfoque.

El entrenamiento cognitivo de dominio general, tiene su sustento científico en el concepto de “neuroplasticidad”, que afirma que el cerebro, de manera similar a un músculo, puede modificarse y adaptarse a nuevos desafíos, y que la exigencia dirigida a una región específica del mismo causará un crecimiento sostenido en cuanto a su tamaño o capacidad funcional. Las metas que traza este tipo de entrenamiento son: mejoras en la función cognitiva que fue entrenada; mejoras en otras funciones cognitivas cercanas o superpuestas; y fundamentalmente mejoras en el desarrollo de otras tareas en el mundo real que utilicen las funciones cognitivas entrenadas. De esta forma, se afirma que realizar actividades en contextos de entrenamiento que apunten al desarrollo

de una cierta función cognitiva se traduce en una mejora en la ejecución de tareas que dependan de dicha función en contextos de competencia.

### **2.3.3. Aplicaciones del EC**

Es posible realizar una clasificación más del EC, en función de cuál es el campo de aplicación sobre el que se desarrolla[7].

- Entrenamiento compensatorio: para solucionar o superar algún déficit cognitivo (asociado frecuentemente al Alzheimer o el deterioro cognitivo leve).
- Entrenamiento restaurativo: asociado a los procesos de rehabilitación.
- Entrenamiento de perfeccionamiento: utilizado para mejorar capacidades cognitivas.

#### **2.3.3.1. Entrenamiento compensatorio**

La aplicación del entrenamiento cognitivo compensatorio es un área de creciente interés, principalmente a lo largo de los últimos 15 años. De acuerdo a la “National Academies of Sciences, Engineering and Medicine”, el entrenamiento cognitivo representa uno de los tres acercamientos más prometedores para lidiar con los desafíos que presenta el envejecimiento cognitivo[8].

Investigaciones en relación a la aplicación de técnicas de entrenamiento cognitivo en pacientes con diagnóstico de esquizofrenia, muestran un efecto leve a moderado de mejoría[9]. Otros estudios muestran también efectos a largo plazo en la capacidad de razonamiento, la velocidad de procesamiento y la independencia para realizar actividades de la vida cotidiana en pacientes mayores a 65 años, para demorar o reducir el déficit cognitivo relacionado al envejecimiento[10]. Existen registros también de diversos trabajos en los que se observan mejoras a corto plazo en las habilidades cognitivas ejercitadas, e incluso con mejorías pequeñas pero significativas en otras habilidades cognitivas no ejercitadas pero relacionadas[10].

Existen también prometedores estudios en relación a, por ejemplo, trastornos por déficit de atención con hiperactividad, desordenes de ansiedad, trastornos del espectro autista y trastornos del estado de ánimo[11].

#### **2.3.3.2. Entrenamiento restaurativo**

El entrenamiento cognitivo aplicado a procesos restaurativos o de rehabilitación es avalado por numerosos estudios que demuestran diversos niveles de

mejorías en la recuperación de funciones cognitivas luego de procesos traumáticos.

Por citar un ejemplo, ante lesiones cerebrales traumáticas se han comprobado mejorías en el funcionamiento cognitivo tanto a nivel general como específico a raíz del entrenamiento, siendo destacables el dominio de la memoria verbal y un progreso en las funciones ejecutivas, dentro de las cuales se destaca una mejora en las Actividades Instrumentales de la Vida Cotidiana (manejo de finanzas personales, de compras, de comunicaciones telefónicas o correo, tareas de limpieza, entre otras)[12].

### **2.3.3.3. Entrenamiento de perfeccionamiento**

El entrenamiento de perfeccionamiento es el enfoque que enmarca aquellas aplicaciones del entrenamiento cognitivo en ámbitos deportivos. El entrenamiento cognitivo de perfeccionamiento se ha aplicado exitosamente en entornos de trabajo de contexto específico en muy diversas disciplinas, como es el caso del tenis[7], cricket[13], o béisbol[14].

Con respecto a su aplicación en el marco de un EC de dominio general, se trata de una premisa más reciente, sobre la cual las investigaciones aun no son extensas ni concluyentes, pero que debido a la estrecha relación entre cognición y deporte, y a los resultados positivos en otros ámbitos de aplicación del EC, resulta de gran interés tanto en el ámbito académico como en el deporte de elite[5].

# Capítulo 3

## Estado del arte

### 3.1. Búsqueda y selección de aplicaciones

Antes de comenzar con el diseño de la solución, investigamos algunas herramientas ya disponibles en el mercado, las analizamos y comparamos entre sí, con el objetivo de conocer cuáles eran las ventajas y desventajas de su implementación. A partir de esta investigación pudimos determinar las características deseadas para el desarrollo de nuestra plataforma, Tremün. A continuación se presenta un análisis en detalle sobre 4 aplicaciones seleccionadas que cumplen con algunas de estas características.

### 3.2. Led Trainer

Led Trainer[4] es un sistema de entrenamiento, evaluación y rehabilitación deportiva, que cuenta con módulos de luces led que se activan con diferentes colores con el objetivo de trabajar el entrenamiento cognitivo para diferentes deportes. Las características principales que se pueden observar a través de la aplicación móvil disponible en PlayStore<sup>1</sup> y AppStore<sup>2</sup> son:

- Conexión con los módulos a través de Bluetooth.
- Provee dos tipos de módulos, uno solo muestra colores y el otro puede ser configurado para que muestre distintas formas como letras, números, símbolos y flechas.
- Existen 6 colores disponibles.

---

<sup>1</sup>Tienda de aplicaciones de Google para dispositivos Android

<sup>2</sup>Tienda de aplicaciones de Apple para dispositivos iOS

- Se pueden conectar hasta 8 módulos simultáneamente.
- Tiene rutinas precargadas.
- Permite crear rutinas personalizadas.
- Permite guardar y exportar resultados.
- Disponible para Android y iOS.

En las siguientes imágenes se pueden observar algunas de las funcionalidades que provee la aplicación móvil para personalizar y ejecutar las distintas rutinas:



Figura 3.1: Inicio de la aplicación

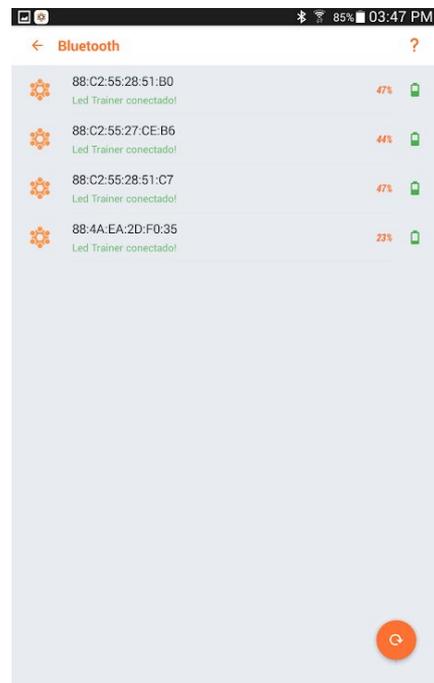


Figura 3.2: Información de módulos

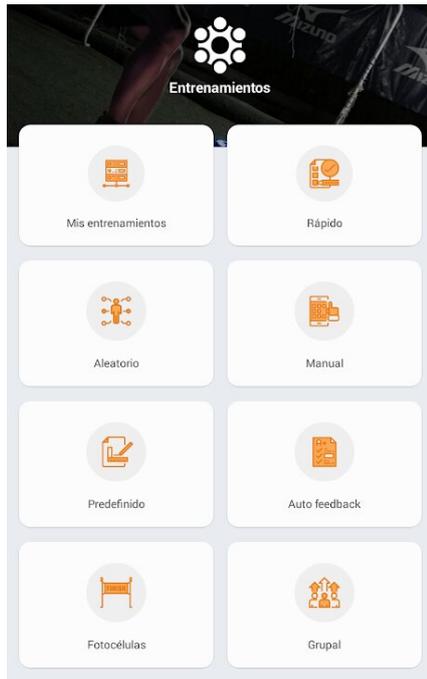


Figura 3.3: Opciones de entrenamientos



Figura 3.4: Rutina rápida



Figura 3.5: Rutina múltiple



Figura 3.6: Rutinas predefinidas

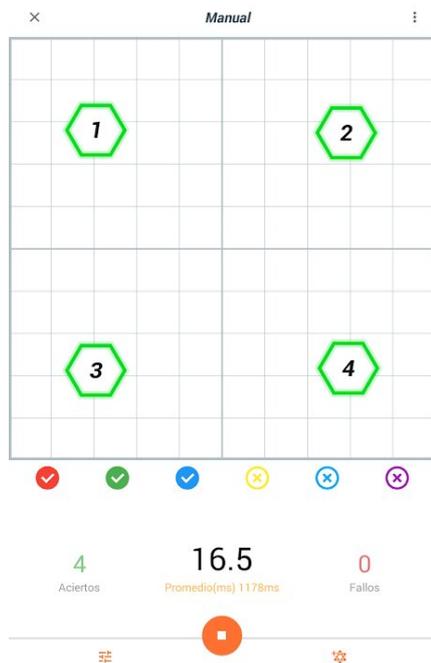


Figura 3.7: Configuración manual



Figura 3.8: Resultados

### 3.3. Fitlight

Fitlight[2] es un sistema de entrenamiento cognitivo compuesto por módulos de luces led y una tablet con el software para configurar los entrenamientos. Los módulos se encienden con diferentes colores y pueden ser apagados por movimiento o golpe, soportando distintas distancias e intensidades. El software viene instalado en la tablet y no esta disponible en las tiendas de aplicaciones móviles (PlayStore y AppStore), por lo que no proveen actualizaciones y no permiten utilizarlo en otros dispositivos. Esto es una gran desventaja ya que los dispositivos se renuevan rápidamente en el mercado y pueden presentar fallas del hardware al ser expuestos a condiciones de entrenamiento, como la lluvia y la humedad.

Sin embargo, este sistema es uno de los mas innovadores con respecto a la flexibilidad que le provee al usuario para programar entrenamientos. Permite diseñar rutinas por pasos indicando qué color se debe encender en cada nodo, permite configurar la intensidad de la luz y la forma en que se desea apagar (golpe o movimiento), incluye la prueba de velocidad por tramos, entre otras funcionalidades. Además, los módulos son muy robustos ya que no tienen botones ni puerto de carga.

En resumen, las características principales de Fitlight son:

- Conexión con los módulos a través de Bluetooth.
- Muestra solo colores, no muestra figuras.
- Tiene sensores que detectan el movimiento y distintas intensidades de golpe.
- Cuenta con rutinas precargadas.
- Permite crear rutinas personalizadas.
- Se pueden programar rutinas por grupos.
- No esta disponible en las tiendas de aplicaciones, solo se puede usar en la tablet provista por Fitlight.
- Permite guardar resultados.

Debido a que el software y el hardware es privado y de muy alto costo, no pudimos tener acceso a la aplicación para poder mostrar las distintas pantallas.

### **3.4. Neural Trainer**

Neural Trainer[15] es un sistema de entrenamiento muy similar a Fitlight ya que provee los nodos de luces led junto con una tablet donde se encuentra el software instalado para programar las rutinas de entrenamiento. La gran diferencia es que permite mostrar, además de colores, distintas formas como figuras geométricas y números, para mejorar la variedad de rutinas que se pueden programar. Sin embargo, el resto de las funcionalidades y características son muy similares a Fitlight y son las siguientes:

- Conexión con los nodos a través de Bluetooth.
- Muestra colores y figuras.
- Provee estímulos auditivos.
- Los nodos se apagan con movimiento.
- Cuenta con rutinas precargadas.
- Permite crear rutinas personalizadas.

- Se pueden programar rutinas por grupos.
- No esta disponible en las tiendas de aplicaciones, solo se puede usar en la tablet provista por Neural Trainer.
- Permite guardar resultados.

Al igual que lo sucedido con Fitlight, no tenemos acceso a la aplicación móvil para poder mostrar las funcionalidades del sistema a través de las pantallas de la aplicación.

### 3.5. SmartGoals

SmartGoals[3] es una propuesta diferente ya que los nodos de luces led funcionan de a pares, y son apagados cuando detectan un movimiento entre ellos. A partir de este cambio se pueden proponer otros tipos de entrenamiento, como por ejemplo en deportes con pelota se pueden desactivar los nodos pasando la pelota entre el arco que se forma entre ellos. A pesar de que parezca un sistema de entrenamiento diferente, su aplicación es muy similar ya que se puede usar para entrenamiento deportivo y rehabilitación.

A continuación se enumeran algunas características de este sistema de entrenamiento:

- Conexión con los nodos a través de Bluetooth.
- Los nodos funcionan de a pares y solo muestran dos colores: naranja y azul.
- No provee estímulos auditivos.
- Los nodos se apagan al detectar movimiento entre cada par.
- Cuenta con una gran cantidad de rutinas precargadas divididas por deporte.
- Disponible para Android y IOS.
- Permite guardar resultados.

En las siguientes imágenes se puede observar cómo se configuran los entrenamientos, seleccionando cantidad de nodos y tiempos de respuesta. SmartGoals propone dos tipos de entrenamientos configurables: uno en el que se indica la cantidad de veces que se deben apagar los nodos por color, donde el resultado es el tiempo que se demora en lograr el objetivo; y otro, en el que

se configura la cantidad de tiempo durante el cual los nodos deben encenderse y apagarse, y, en este caso, se contabiliza la cantidad de veces que se logran apagar los nodos por color.



Figura 3.9: Configuración de rutinas

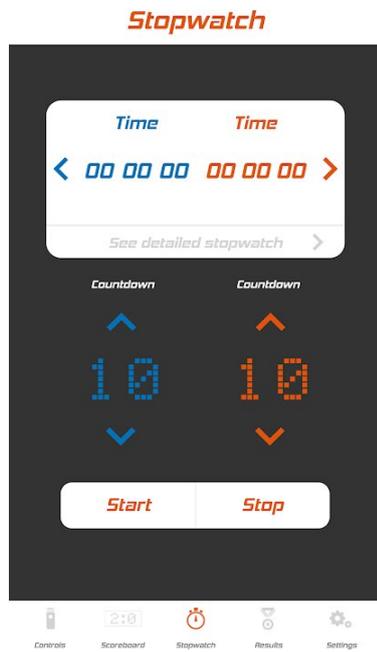


Figura 3.10: Rutina por hits

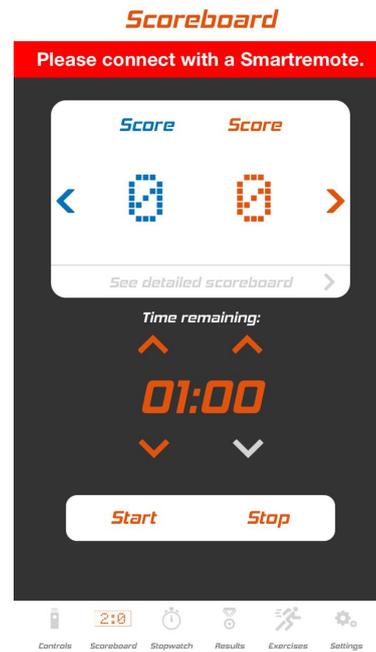


Figura 3.11: Rutina por tiempo

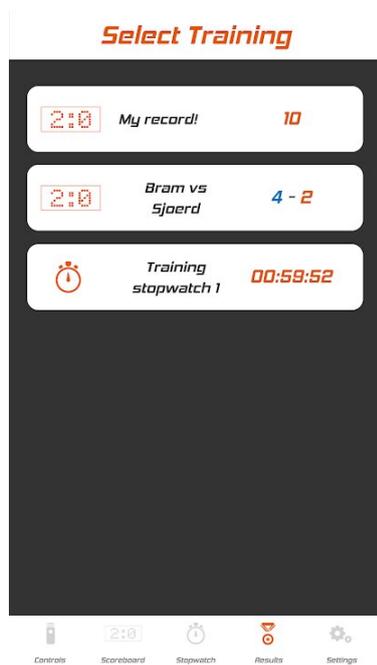


Figura 3.12: Listado de resultados



Figura 3.13: Detalle de resultados

## 3.6. Análisis comparativo

A partir del estudio de las distintas alternativas de sistemas de entrenamiento cognitivo, se realizó un análisis comparativo teniendo en cuenta las principales características funcionales y técnicas, para luego poder tomar una decisión respecto al diseño de la solución.

### 3.6.1. Funcionalidades

En primer lugar, observamos las funcionalidades principales de cada plataforma y las comparamos entre sí. En la tabla 3.1 se puede observar que todos los sistemas permiten crear rutinas personalizadas, pero también tienen un conjunto de entrenamientos configurados para una ejecución más rápida. También se puede apreciar que las plataformas difieren en cuanto a cantidad de colores y formas disponibles para configurar en los nodos.

	<b>Led Trainer</b>	<b>FitLight</b>	<b>Neural Trainer</b>	<b>SmartGoals</b>
Rutinas preconfiguradas	Sí	Sí	Sí	Sí
Rutinas personalizadas	Sí	Sí	Sí	Sí
Rutina aleatoria	Sí	Sí	Sí	No
Colores disponibles	Sí (6)	Sí	Sí	Sí (2)
Formas disponibles	Sí	No	Sí	No
Test de velocidad	No	Sí	Sí	Sí

Tabla 3.1: Funcionalidades

### 3.6.2. Características técnicas

En segundo lugar, se analizaron las características técnicas de los sistemas de entrenamiento, donde se puede observar la forma de conexión entre el controlador y los nodos, la disponibilidad de la aplicación móvil en las tiendas de distribución, el idioma, entre otros.

	<b>Led Trainer</b>	<b>FitLight</b>	<b>Neural Trainer</b>	<b>SmartGoals</b>
Conexión con nodos	Bluetooth	Bluetooth	Bluetooth	Bluetooth
Disponible en Google Play	Sí	No	No	Sí
Disponible en App Store	Sí	No	No	Sí
Disponible en Español	Sí	No	Sí	No
Autonomía de los nodos	12 hs	4 hs	3 hs	8 hs
Rango de utilización	15 metros	50 metros	30 metros	30 metros

Tabla 3.2: Características Técnicas

### 3.7. Conclusión

Luego de analizar y comparar las distintas herramientas disponibles en el mercado, pudimos seleccionar las características deseadas para nuestra plataforma, con el objetivo de mejorar las ya existentes. Entre estas características se pueden mencionar la flexibilidad a la hora de configurar las rutinas de entrenamiento, la importancia de proveer rutinas precargadas, la disponibilidad de la aplicación para distintos dispositivos mediante las tiendas de distribución, entre otras.

Una de las principales desventajas de las plataformas analizadas es su costo excesivo y, en menor medida, que no todos están en español, lo que nos motivó a emprender un desarrollo local pensado para democratizar el acceso a este tipo de tecnologías. Por último, es posible señalar que el rango de utilización en todas las plataformas se ve condicionado por la utilización de la tecnología Bluetooth, de menor alcance que otras redes como es el caso de las redes Wi-Fi.

En el siguiente capítulo se detallan las tecnologías analizadas para el desarrollo y finalmente el diseño e implementación de la solución.

# Capítulo 4

## Análisis de herramientas

En este capítulo se describen las tecnologías analizadas para los distintos componentes de hardware y software que conforman la plataforma. Si bien la arquitectura completa de la misma, la interacción entre sus componentes y su funcionamiento detallado serán desarrollados en el capítulo 5, consideramos necesaria una breve introducción que ayude a comprender el criterio con el cual seleccionamos las tecnologías utilizadas.

En primer lugar, necesitábamos desarrollar una aplicación móvil que permitiese a preparadores físicos y entrenadores planificar el trabajo, diseñar rutinas de entrenamiento, ejecutarlas, mostrar sus resultados en tiempo real y posteriormente almacenar los mismos en la nube. Por otro lado, debíamos contar con un conjunto de nodos con la capacidad de mostrar distintas figuras y colores mediante una pantalla conformada por luces LED, así como la de permitir el apagado a través de sensores. En última instancia, para controlar los nodos encargados de representar las rutinas, debíamos definir e implementar un controlador que medie entre la aplicación móvil y los nodos. Este módulo básicamente debía ser capaz de recibir las peticiones desde el dispositivo móvil, administrar el comportamiento de los nodos y transmitir los resultados recolectados en tiempo real a la aplicación móvil, para ser mostrados al usuario.

### 4.1. La aplicación móvil

La aplicación móvil permite a los entrenadores la creación y almacenamiento de las rutinas de entrenamiento y la visualización de los resultados en tiempo real. Mantiene una comunicación bidireccional con el controlador, permitiendo el inicio y finalización de rutinas, y recibiendo las métricas de performance de los deportistas.

### 4.1.1. Desarrollo de aplicaciones móviles

A la hora de desarrollar una aplicación móvil son muchas las consideraciones a tener en cuenta. En primer lugar, porque el dispositivo difiere notoriamente en sus capacidades y características al de una computadora de escritorio o notebook. Pantallas mucho más pequeñas, menor potencia de procesamiento y menos capacidad de almacenamiento son algunas de sus principales restricciones. Por otro lado, es necesario tener en cuenta que el mercado de dispositivos móviles se encuentra notablemente fragmentado, por lo que brindar una aplicación que pueda instalarse en cualquier sistema operativo no es una tarea sencilla.

#### 4.1.1.1. La tecnología móvil

En la actualidad, se estima que cerca de la mitad de la población mundial posee un smartphone, y el número va en constante crecimiento. Desde mediados de la década de 1990 la industria de la comunicación móvil ha protagonizado un crecimiento explosivo. La amplia adaptación de la comunicación inalámbrica se aceleró a partir de que gobiernos de todo el mundo proporcionaron una mayor competencia y nuevas licencias de espectro radioeléctrico para servicios de comunicación personal (PCS) en la banda de frecuencia 1800-2000MHz.

La primera red de tecnología móvil es conocida como 1G (la letra G hace referencia a la palabra “generación”). La primera red 1G lanzada comercialmente tuvo lugar en Tokyo en el año 1979, y luego de 5 años extendió su cobertura a todo el territorio de Japón, siendo este país el pionero en la construcción de una red nacional de 1G. La principal característica de este tipo de redes era el carácter analógico de sus radioseñales para la transferencia de datos.

Las redes de telecomunicación celular de segunda generación (2G) vieron su aparición comercial en el año 1991. El principal cambio que trajeron consigo estas redes fue el hecho de que la información pasó a ser encriptada de manera digital. También se introdujo el servicio de datos, que permitió el envío de mensajes de texto (SMS).

La primera red pre-comercial de la tecnología de tercera generación (3G) vio la luz en el año 1998, nuevamente en Tokyo, mientras que la primera red comercial tuvo su aparición 3 años más tarde en la misma ciudad. Entre sus innovaciones destacadas cabe resaltar el acceso a internet móvil, la tecnología de voz inalámbrica, las videollamadas y la televisión móvil, en conjunto con importantes mejoras de seguridad.

Las redes de cuarta generación en sus versiones tempranas aparecieron hacia finales de la primera década del corriente siglo. Se trata de una red

basada completamente en el protocolo IP, que provee acceso de banda ultra-ancha a internet en los dispositivos móviles brindando una importante mejora en la velocidad, calidad y capacidad de navegación de los teléfonos[16].

#### **4.1.1.2. Sistemas operativos móviles**

Los primeros teléfonos celulares contaban con sistemas operativos relativamente simples, dado que las capacidades de los dispositivos eran limitadas. Sin embargo, la irrupción de los “smartphones” en el mercado (posibilitada por el desarrollo de la tecnología móvil y la industria del hardware) permitió añadir en los teléfonos móviles características muy similares a las de una computadora personal, como CPUs de alta velocidad, unidades de procesamiento gráfico, gran espacio de almacenamiento, multitasking, pantallas y cámaras de alta resolución o hardware de comunicación multipropósito.

Para poder soportar y articular el funcionamiento de las mencionadas características, los sistemas operativos para dispositivos móviles han crecido en sofisticación a lo largo de los años, combinando funcionalidades propias de las computadoras personales con otras más específicas como la interacción mediante una pantalla táctil o sistemas de posicionamiento global.

La convergencia de los avances tecnológicos de hardware, software y redes móviles ha traído como resultado el desarrollo de diferentes sistemas operativos móviles que compiten entre sí por su lugar en el mercado. Algunas de estas soluciones que merecen ser mencionadas son el caso de Android (desarrollado por Google), iOS (sistema operativo de Apple), Symbian (de Nokia) o Windows Phone (de Microsoft)[17].

#### **4.1.1.3. Enfoques de desarrollo y sus características**

A la hora de diseñar una aplicación móvil existen distintos enfoques para abordar el desarrollo de la misma[18]. La razón de esta diversidad de metodologías radica en una simple problemática: el código de una aplicación de Android no funciona, por ejemplo, en iOS, y viceversa. Ante este escenario, una primera posibilidad sería desarrollar una aplicación específica para un sistema operativo en particular (lo que se conoce como desarrollo nativo). El potencial de desarrollar una aplicación nativa es muy alto ya que permite acceder a todas las características ofrecidas por el dispositivo, tiene una alta performance y permite que la aplicación se ejecute en segundo plano para, por ejemplo, enviar notificaciones cuando se desee.

La contracara sin embargo es la pérdida de presencia en el mercado ya que, como se mencionó previamente, el código funcionará para un único sistema operativo. Para tomar dimensión de la enorme pérdida que constituye

limitarse a un único sistema operativo, es necesario ver lo que se conoce como “fragmentación de mercado”, que en pocas palabras hace referencia al hecho de que no existe un único sistema operativo que concentre a la absoluta mayoría de los usuarios. De acuerdo a Statcounter, en el reporte correspondiente a Abril de 2020, Android posee el 70,68 % de los usuarios, mientras que iOS nuclea al 28,79[19]. Resulta evidente entonces que dejar de lado el desarrollo de cualquiera de los dos sistemas operativos predominantes en la industria deja por fuera del producto final a una gran cantidad de usuarios.

La primer alternativa que aparece es simplemente desarrollar la aplicación de manera nativa para cada uno de los sistemas operativos que se desean abarcar (si bien entre iOS y Android se reparten casi la totalidad de los usuarios, no dejan de existir otros sistemas operativos que podrían resultar de interés, como es el caso de Windows). Este acercamiento, más allá de resultar tentador por brindar todos los beneficios de una aplicación nativa en cada uno de los sistemas operativos elegidos, resulta extremadamente costoso. En primer lugar el tiempo de desarrollo se multiplica, ya que no es posible reutilizar código entre plataformas. Por otro lado, al tratarse de tecnologías completamente diferentes, es necesario contar con un equipo de desarrollo específico para cada plataforma que se desea alcanzar.

El elevado coste del desarrollo nativo, tanto en términos de tiempo como de dinero, provocó la aparición de múltiples alternativas creadas para obtener una aplicación móvil multiplataforma pero permitiendo la reutilización de código entre los distintos sistemas operativos.

En primer lugar, existe la posibilidad de desarrollar una Aplicación Móvil Web, que no es otra cosa que la adaptación de un sitio web a las pantallas móviles. Se implementan con tecnologías web ampliamente extendidas como HTML, CSS y JavaScript, son independientes de la plataforma, y son fáciles de desarrollar, lanzar y actualizar. Las principales desventajas de este enfoque son la baja performance (debido a la interacción cliente-servidor), la necesidad de conectividad para acceder a ellas, y el limitado acceso a herramientas propias del dispositivo, ya que se ejecutan dentro de un browser y no de manera nativa. Además, resultan menos atractivas que las aplicaciones nativas ya que no están instaladas en el dispositivo y no pueden hacer pleno uso de los elementos de diseño propios de cada sistema operativo.

Por otro lado, es posible mencionar a las Aplicaciones Híbridas. Éste tipo de aplicaciones también se desarrolla con tecnologías web como las mencionadas previamente, las cuales se encuentran instauradas desde hace muchos años en el mercado y cuentan con una gran oferta de profesionales experimentados, lo que hace que la conformación de un equipo de desarrolladores resulte más sencilla y menos costosa en términos de dinero en relación al desarrollo móvil

nativo. La principal diferencia con las aplicaciones web, es que las aplicaciones híbridas no se ejecutan en un browser, sino que lo hacen en un contenedor web del dispositivo, lo que posibilita un acceso mayor a las herramientas propias del sistema operativo. Éstas aplicaciones son distribuidas en las tiendas de aplicaciones de cada plataforma, y son instaladas en el teléfono. Entre sus desventajas más notorias es posible mencionar una performance menor que las aplicaciones nativas y la imposibilidad de usar componentes de interfaz nativos, lo cual afecta directamente la experiencia de usuario.

En tercer lugar aparecen las aplicaciones interpretadas, que son proyectos desarrollados en lenguajes de programación como Ruby o Java, y cuyo código luego es traducido en su mayoría a código nativo para cada uno de los sistemas operativos deseados, mientras que la porción restante del mismo es interpretada en tiempo de ejecución. Al traducirse a código nativo, este enfoque permite interfaces completamente nativas, que mejoran significativamente la experiencia de usuario con respecto a los enfoques anteriores. Como principal desventaja, es posible señalar que se genera una dependencia absoluta del entorno de desarrollo, del abanico de posibilidades de integración que el mismo brinda y de su evolución y mantenimiento.

Por último, es posible mencionar las aplicaciones generadas por compilación cruzada. Éstas son aplicaciones compiladas que generan una versión nativa de alta performance para cada plataforma que se ponga como objetivo. Sin embargo, este último enfoque no siempre permite una total reutilización del código entre distintos sistemas operativos.

Este breve repaso por distintas alternativas de desarrollo deja en evidencia las complejidades a la hora de desarrollar una aplicación móvil, y deja en claro también que no existe una única y absoluta solución, sino que cada alternativa posee sus virtudes y defectos. Dependerá del caso de uso en cuestión definir un enfoque que resulte conveniente.

#### **4.1.2. React Native**

Alcanzar la mayor cantidad de usuarios en el mercado resulta indispensable. En éste marco, y teniendo en cuenta las complejidades que la fragmentación de mercado trae consigo, optamos por utilizar un enfoque de desarrollo que permitiese disponibilizar la aplicación móvil desarrollada al menos en Android e iOS. Dentro de las opciones disponibles para llevar adelante dicho desarrollo, se optó por trabajar con React Native, una tecnología que se enmarca en el grupo de las aplicaciones interpretadas. Por un lado, React Native permite independizarse de la interfaz final de Android y IOs, utilizando palabras clave propias que luego se traducen a componentes nativos de cada uno de los sistemas operativos (por ejemplo, la primitiva “Text” de React Native será

traducida como “UIView” al ejecutarse en IOs, y como “TextView” en Android). Al mismo tiempo, ejecuta código JavaScript (mediante un motor de JavaScript que corre en un hilo separado) que se encarga, entre otras tareas, de la lógica de negocio de la aplicación.

#### 4.1.2.1. Características generales

React Native, de acuerdo a su documentación oficial, combina las mejores partes del desarrollo móvil nativo con la flexibilidad y potencia de React[20]. Se trata de un framework que permite el desarrollo de aplicaciones nativas para Android e iOS mediante JavaScript. Se encuentra disponible desde Septiembre de 2015, es de código abierto, y es desarrollado y mantenido por Facebook.

Desde Facebook[21] detallan que por aquel entonces el proyecto de React ya llevaba dos años de desarrollo y una amplia aceptación, y que el siguiente desafío era obtener una experiencia de usuario propia de las plataformas móviles nativas, combinada con la experiencia de desarrollo que se tiene al trabajar con React en la web.

La primera posibilidad explorada por el equipo de desarrollo de Facebook fue la utilización de WebViews, y si bien rescatan la flexibilidad de esta opción y el hecho de poder sacar ventaja del rápido ciclo de iteración de React y la web, señalan que no fueron capaces de lograr una experiencia de usuario verdaderamente nativa, así como tampoco alcanzaron la performance y escalabilidad deseadas.

La segunda alternativa en la que indagaron fue lograr la portabilidad del código de React a nativo, en el marco de un proyecto que denominaron “ComponentKit”. A través del mismo alcanzaron buenos resultados, entre los que destacan la construcción de interfaces de usuario declarativas y predecibles gracias a React, y el poder del entorno nativo, con un sofisticado manejo de gestos y componentes específicos de la plataforma. Sin embargo, también señalan que el proyecto era únicamente para iOS, y que la velocidad de desarrollo se veía muy perjudicada debido a la necesidad de compilar luego de cada cambio.

La tercera alternativa, sobre la cual efectivamente se construyó React Native, es lo que desde Facebook denominan “scripting Native”. En pocas palabras, lo que proponen es utilizar JavaScript para realizar las llamadas a las APIs nativas, permitiendo explotar el potencial del entorno nativo y al mismo tiempo sacar ventaja de la infraestructura de JavaScript existente. Sumado a esto, al tratarse puramente de JavaScript, es posible hacerlo funcionar entre las distintas plataformas. Una ventaja clave de React Native es la posibilidad de adoptar la tecnología de manera incremental, introduciendo su uso y aprovechando sus ventajas sobre proyectos ya existentes pero sin la necesidad de re-convertirlos por completo. Ésta es la modalidad que adoptaron, por ejemplo, con la propia

aplicación móvil de Facebook.

En resumen, desde el punto de vista del desarrollo, la propuesta de React Native parece solucionar varios de los problemas enumerados previamente: permite un desarrollo multiplataforma con código reutilizable que hace que la fragmentación de mercado no sea un problema, tiene una curva de aprendizaje menor ya que se basa en JavaScript, lenguaje ampliamente utilizado (la curva es menor para quienes estén familiarizados con React), y tiene como resultado una aplicación nativa, con las mejoras de performance e interfaz de usuario que esto trae consigo.

Existen diversos estudios[22][23] que comparan la ejecución de aplicaciones desarrolladas en React Native con aplicaciones nativas tanto en Android como en iOS, y es posible listar ciertas conclusiones comunes. En primer lugar, se obtiene un look-and-feel prácticamente idéntico, con diferencias casi imperceptibles. Con respecto a la disponibilidad de interacción con componentes nativos, es cierto que React Native aún no ha alcanzado abarcar la totalidad de los mismos, pero es un framework con una amplia comunidad de desarrolladores que brinda actualizaciones permanentes. En relación al uso de memoria, los resultados para React Native son muy favorables siendo similares a las aplicaciones nativas, mientras que el uso de CPU es mayor en React Native. Con respecto a los cuadros por segundo observables en animaciones o transiciones, las diferencias resultan tan pequeñas que es posible afirmar que no impactan de manera significativa la experiencia de usuario, y en relación al tiempo de respuesta (por ejemplo, al iniciar la aplicación) las aplicaciones de React Native resultan más veloces que las aplicaciones nativas de Android, y más lentas que las aplicaciones nativas de iOS. Por último, al analizar el tamaño de la aplicación, las aplicaciones desarrolladas con React Native resultan más pesadas.

#### **4.1.2.2. Expo CLI vs React Native CLI**

Para crear un proyecto en React Native existen dos alternativas diferentes: Expo CLI y React Native CLI. Por un lado, Expo[24] provee un conjunto de herramientas y servicios que abstraen al desarrollador de las configuraciones específicas que tiene cada plataforma. Por lo tanto, no es necesario contar con una computadora Mac para configurar y desplegar la aplicación para dispositivos iOS. Además, ofrece una aplicación móvil (Expo Client), que se puede descargar en las tiendas de aplicaciones de cada sistema operativo, que permite probar la aplicación en un dispositivo real mientras se desarrolla, sin necesidad de instalarla. Por el otro lado, para crear un proyecto utilizando React Native CLI es necesario utilizar Xcode y/o Android Studio, y realizar las respectivas configuraciones para cada sistema operativo. Respecto a las pruebas

durante el desarrollo, se pueden realizar en simuladores o en dispositivos reales, necesitando, en este último caso, enchufar el dispositivo a la computadora y realizar configuraciones específicas. Si se desea probar en un dispositivo iOS, es necesario contar con una computadora Mac.

Antes de comenzar el proyecto analizamos los servicios que provee Expo y observamos que cubría todas las funcionalidades necesarias para el desarrollo de la aplicación móvil. Además, como explicamos anteriormente existe una gran ventaja a la hora de probar la aplicación en ambos sistemas operativos utilizando Expo Client. Por lo tanto, optamos por utilizar esta alternativa para llevar adelante el desarrollo.

### **4.1.3. Persistencia de datos**

Un gran desafío que presenta el desarrollo de aplicaciones móviles es la implementación de funcionalidades offline, es decir, permitir que la aplicación siga funcionando y registre los datos, aun sin estar conectada a Internet. Hay que tener en cuenta que es muy probable que, por momentos, el dispositivo no tenga conexión de red, o que la conexión sea mala, por lo que es indispensable permitir que el usuario pueda seguir usando la aplicación de manera offline. Para lograr este funcionamiento es necesario mantener el estado de la aplicación de manera local en el dispositivo que se está utilizando. Además, se debe analizar si la aplicación requiere persistir información, qué tipos de datos se quieren persistir y en qué cantidades, para poder seleccionar adecuadamente el tipo de base de datos que se debe utilizar.

Debido a que existe una gran variedad de bases de datos orientadas a aplicaciones móviles con distintas características, decidir cuál opción utilizar, no fue una tarea fácil. A continuación, vamos a analizar algunas de las bases de datos locales más populares disponibles para React Native[25]. Una base de datos local es aquella que permite almacenar los datos en los dispositivos móviles para poder usarlos de manera offline, y luego los sincroniza con la nube para poder persistir la información de manera más segura.

#### **4.1.3.1. Realm**

Realm[26] es una base de datos open source creada y diseñada para aplicaciones móviles con funcionamiento offline. Tiene su propio motor de base de datos y está orientada a objetos, por lo que evita realizar largas consultas SQL y mejora significativamente la performance. Además, provee un mecanismo de sincronización de los datos transparente al usuario y al desarrollador, y encripta los datos usando distintos estándares dependiendo de la plataforma (Android, iOS, Windows). La gran desventaja de esta alternativa, es que no

está disponible en Expo y, por lo tanto, es más compleja de integrar. Esta funcionalidad está siendo requerida desde el año 2017 y todavía no hay respuesta del equipo de Expo para implementarla[27].

#### **4.1.3.2. Firebase**

Firebase[28] es una plataforma desarrollada por Google para el desarrollo de aplicaciones web y aplicaciones móviles. Entre sus servicios ofrece dos soluciones de base de datos: Cloud Firestore y Realtime Database. Ambas soluciones proveen una base de datos NoSQL cuya principal característica es la sincronización en tiempo real. Están centradas en dispositivos móviles y proveen almacenamiento local para aplicaciones que funcionan sin conexión. La principal diferencia entre ellas es la forma en que almacenan los datos: Realtime Database almacena la información en formato JSON, mientras que Cloud Firestore lo hace mediante colecciones de documentos. Esto permite que la segunda opción maneje de forma más eficiente consultas complejas. Con respecto a la seguridad, ambas soluciones utilizan un sistema de reglas para manejar el acceso a los datos. Sin embargo, no proveen encriptación, por lo que no son recomendables para datos sensibles. En cuanto al precio, sin importar cual opción se elija, Firebase provee una capa gratuita (hasta 1GB de almacenamiento) y luego se paga por lo que se usa.

Por último, ambas bases de datos son muy simples de configurar ya que están incluidas en el conjunto de herramientas que provee Expo junto con otras funcionalidades de Firebase, como la autenticación de los usuarios [29].

#### **4.1.3.3. SQLite**

SQLite[30] es una librería open source desarrollada en C, que implementa un motor de base de datos relacional. Fue diseñada para proveer almacenamiento local en aplicaciones móviles por lo que es muy liviana y viene incluida en todos los dispositivos móviles, sin importar el sistema operativo. Entre sus principales características incluye las propiedades ACID (atomicidad, consistencia, aislamiento y durabilidad). Esta base de datos está incluida entre las herramientas de Expo y se accede a través de un conjunto de funciones que permiten crear la base de datos y realizar las transacciones y consultas necesarias [31]. Sin embargo, no provee un mecanismo de sincronización automático, lo cual impacta a la hora de implementar funcionalidades sin conexión.

#### **4.1.3.4. PouchDB**

PouchDB es una implementación open source de CouchDB (base de datos NoSQL mantenida por Apache Software Foundation) que provee una API en

JavaScript para realizar las consultas a la base de datos mediante el protocolo HTTP[32]. La información es almacenada en formato JSON y provee mecanismos de sincronización offline y entre usuarios. Con respecto a la seguridad incluye autenticación de usuarios para poder proteger los datos. A pesar de que esta solución está basada en CouchDB y usa sus protocolos y mecanismos de almacenamiento, también puede ser configurada con otras bases de datos del lado del servidor, como MongoDB, MySQL y PostgreSQL.

Esta alternativa no está incluida entre los servicios de Expo.

#### **4.1.3.5. Redux y Async Storage**

Finalmente, queremos presentar una opción para mantener un almacenamiento local sin necesidad de usar una base de datos. Esto se puede lograr a través de la utilización de Redux y Async Storage. Redux[33] es una librería de Javascript que sirve para administrar y mantener el estado de una aplicación. Mientras que Async Storage[34] es una librería provista por la comunidad de React Native que permite almacenar, con formato clave valor, algunos datos que son compartidos por toda la aplicación. Este almacenamiento es asíncrono, no cifrado y persistente, por lo que se puede mantener entre cada uso de la aplicación. Usando estas dos herramientas en conjunto, se puede persistir cierta información no sensible de la aplicación, como por ejemplo los datos de la sesión del usuario para no tener que requerir un inicio de sesión cada vez que se utiliza la aplicación.

#### **4.1.3.6. Análisis comparativo**

En la siguiente tabla se comparan las características más importantes de las bases de datos analizadas, entre las cuales se puede observar la forma en que almacenan los datos, las distintas alternativas utilizadas para asegurar la información y el acceso a ella por parte de los usuarios, la sincronización automática, entre otras.

	<b>Realm</b>	<b>Firestore</b>	<b>SQLite</b>	<b>PouchDB</b>
Tipo	Orientada a objetos	NoSQL: JSON y colección de documentos	Relacional	JSON
Sincronización automática	Sí	Sí	No	Sí
Seguridad	Usa diferentes estándares de encriptación para cada plataforma	Provee un sistema de reglas personalizable para el acceso a los datos. Conexiones seguras (SSL)	Encriptación de los datos mediante una extensión paga	Soporta SSL y se pueden encriptar los datos
Precio	Gratis (tiene una versión paga)	Gratis hasta 1GB de almacenamiento	Gratis	Gratis
Disponible en Expo	No	Sí	Sí	No

Tabla 4.1: Análisis comparativo de Bases de Datos

## 4.2. El controlador

El controlador, construido sobre una Raspberry Pi, es el encargado de la ejecución de las rutinas, y mantiene una comunicación bidireccional tanto con la aplicación móvil como con los nodos. Por un lado, expone una API REST que permite a la aplicación disparar la ejecución de diferentes rutinas, y retorna en tiempo real los resultados de las mismas. Por otro, contiene la lógica de ejecución de rutinas propiamente dicha, orquesta qué nodos deben encenderse y de qué forma, y es notificado cuando uno de ellos es apagado por sensor.

### 4.2.1. Raspberry Pi

Raspberry Pi es un ordenador de placa única desarrollado en el Reino Unido, que vio su primera versión en el año 2012. Es un ordenador de muy bajo costo, cuyo objetivo inicial fue el de promover la enseñanza de informática en las escuelas. Su reducido tamaño lo hace sumamente portable, y a pesar del mismo cuenta con una gran variedad de componentes y conectores que lo convierten en una herramienta muy potente.

En todas sus versiones, incluye un procesador Broadcom, memoria RAM, GPU, puertos USB, HDMI, Ethernet (el primer modelo no lo tenía), 40 pines GPIO (desde la Raspberry Pi 2) y un conector para cámara. Ninguna de sus ediciones incluye memoria, siendo esta en su primera versión una tarjeta SD y en ediciones posteriores una tarjeta MicroSD. El software es de código abierto, siendo su sistema operativo oficial el Raspbian, una versión adaptada de la distribución Debian, de Linux. De todas formas, también se permite la instalación y utilización de otros sistemas operativos.

Para el presente desarrollo, se utilizó una Raspberry Pi 3, en su modelo B+, que cuenta con un procesador quad-core de 64bits y 1.4GHz, 1GB de SDRAM, ranura para MicroSD, USB 2.0, entrada de HDMI, soporte WiFi 802.11ac de doble banda y soporte de Bluetooth 4.2.

### 4.2.2. Golang vs Python

A la hora de realizar una implementación sobre una Raspberry Pi, el primer lenguaje de programación que suele considerarse es Python, que se encuentra disponible de forma nativa en el computador, y es el lenguaje que se promueve desde Raspberry Pi Foundation (que incluso incluye tutoriales en su documentación oficial[35]).

Si bien la utilización de Python se barajó como una posibilidad concreta para el desarrollo del código del controlador, finalmente se optó por el lenguaje

Golang (popularmente conocido como Go), de código abierto y desarrollado por Google.

A diferencia de Python, que puede considerarse un lenguaje de programación de propósito general, Go fue concebido como un lenguaje orientado a sistemas. Es a raíz de éste enfoque que Go incluye algunas características que lo hacen más adecuado para el presente caso.

En primer lugar, se trata de un lenguaje compilado (Python, en cambio, es interpretado). Esto genera una evidente ventaja de performance inicial por el simple hecho de que el lenguaje interpretado tiene que traducir el código en tiempo de ejecución, un trabajo adicional que se traduce directamente en un mayor coste de tiempo. Si bien hay estudios[36] que demuestran que la diferencia de performance a favor de Golang a la hora de desarrollar una aplicación web tradicional (servicios rest, con templating y base de datos) no resulta tan significativa (principalmente a causa de la latencia), a la hora de ejecutar problemas más complejos, que requieren de mayor esfuerzo computacional, la potencia de Go se ve con claridad[37]. Por otro lado, la concurrencia es un factor importante para el funcionamiento del controlador, y en este punto es donde Go saca una buena ventaja ya que incorpora de manera nativa el soporte para concurrencia, mientras que Python no lo incluye. Además, sacar provecho de ésta funcionalidad es sumamente sencillo a través de las denominadas go routines y los channels, que permiten la comunicación entre las distintas rutinas.

### 4.2.3. Go

El lenguaje Golang[38] surgió en el año 2007, pero no fue hasta 2011 que Google lo publicó oficialmente como proyecto open source con licencia BSD, luego de 4 años en los que puso a disposición todos los recursos necesarios para su desarrollo. La primera versión estable (1.0) fue publicada en Marzo de 2012, y el lenguaje continúa en permanente desarrollo, con actualizaciones cada aproximadamente 6 meses. En palabras de Google, “Go es un lenguaje de programación diseñado por Google para ayudar a solucionar problemas de Google, y Google tiene grandes problemas”[39].

Cabe destacar entonces que Go no nace a partir de la intención de desarrollar un lenguaje de programación propio, sino que resultó ser la inevitable respuesta a una problemática que Google no lograba resolver: ¿Cómo eliminar la lentitud y la torpeza del desarrollo de software en Google, y así hacer que el proceso sea más productivo y escalable? En otras palabras, el propósito de Go no es investigar el diseño de un lenguaje de programación, sino que se trata de poner el lenguaje de programación al servicio de la ingeniería de software.

De acuerdo a su documentación oficial, el proyecto surge en un contexto

radicalmente distinto al actual, en el que la mayoría del código productivo del lado del servidor se escribía en lenguajes como Java o C++, github aún no existía y la mayoría de los computadores no eran multiprocesador. Sin embargo, era también un punto de quiebre. Por un lado, las computadoras se habían vuelto mucho más veloces que cuando nacieron los lenguajes previamente mencionados, y gran parte de la complejidad de uso de estos últimos radicaba en que los mismos no habían acompañado el ritmo de desarrollo del hardware. Por el otro, resultaba evidente que el futuro de los computadores era multiprocesador y estos lenguajes no proveían demasiadas herramientas para sacar provecho de esta ventaja de manera sencilla y a la vez segura.

Una vez identificadas las problemáticas a solucionar, el equipo de desarrollo de Golang trazó ciertos lineamientos que se convertirían en los pilares del proyecto. En primer lugar, deseaban brindar a través de este nuevo lenguaje un soporte de primera clase para algún tipo de concurrencia o paralelismo. Por otro lado, deseaban acompañar esta concurrencia con un eficiente manejo de recursos, por lo que identificaron el manejo automático y seguro de la administración de la memoria como un factor clave. Además, se plantearon como meta lograr que convivan 3 características que, según ellos, no lograban hacerlo en ninguno de los lenguajes de amplio uso comercial: compilación eficiente, ejecución eficiente y facilidad de programación. En el desarrollo de Go la intención fue combinar la facilidad de programación de un lenguaje de programación interpretado, dinámicamente tipado, con la eficiencia y seguridad de un lenguaje compilado, estáticamente tipado.

### 4.3. Los nodos

El tercer elemento central en la plataforma Tremün son los nodos, dispositivos electrónicos basados en sensores y luces LED con la capacidad de conectarse a una red WiFi. Los prototipos fueron desarrollados en el Laboratorio de Investigación en Nuevas Tecnologías Informáticas (LINTI) de la Facultad de Informatica de la Universidad Nacional de La Plata en el año 2018. Cada nodo cuenta con una matriz de LEDs RGB direccionables, que permite mostrar en la pantalla diversas figuras y colores. Cuenta también con un sensor de proximidad que permite a los deportistas interactuar con los dispositivos, y con baterías de polímero de litio recargables mediante un puerto micro-USB, a través del cual también se programa el firmware (escrito en el lenguaje de programación C). Por último, cuenta con un circuito de control de prendido y apagado del tipo “press ON, hold OFF” [40].



Figura 4.1: Nodo

Los nodos mantienen una comunicación bidireccional con el controlador, recibiendo las señales que le indican qué figura y color mostrar en la pantalla, así como el tiempo que deben permanecer encendidos antes del apagado automático. Por su parte, el controlador también recibe eventos por parte de los nodos, los cuales permiten llevar un control estadístico de los tiempos de reacción de los atletas.

## 4.4. La comunicación

### 4.4.1. Red Wi-Fi

A la hora de decidir el soporte de red para establecer la comunicación inalámbrica, tanto entre el controlador y el dispositivo móvil como entre el primero y los nodos, fue necesario contemplar las condiciones del entorno en el que la plataforma de entrenamiento debe permitir ejecutarse. Gran parte de los deportes se practican al aire libre en terrenos de gran amplitud, por lo que el rango de alcance de la tecnología de transmisión de los datos resulta un punto clave. Con el objetivo de brindar un amplio rango de comunicación para no limitar la planificación de los entrenamientos a espacios reducidos, se optó por utilizar una red Wi-Fi para la interconexión inalámbrica de los diversos dispositivos.

### 4.4.2. El protocolo HTTP

Al momento de implementar la comunicación entre la aplicación móvil y la Raspberry Pi se optó por utilizar el ampliamente adoptado protocolo HTTP.

En el controlador se desplegó una API REST para manejar las peticiones provenientes de la aplicación y, aprovechando la posibilidad que brinda HTTP de envío bidireccional de información, se analizaron dos alternativas para la transmisión de los resultados en tiempo real: Server Side Events y WebSockets.

#### **4.4.2.1. REST**

La Transferencia de Estado Representacional[41], comúnmente conocida como REST, es una arquitectura de desarrollo web basada completamente en el protocolo HTTP, utilizada para la transferencia de información. La abstracción principal de representación de la información en REST se conoce como “recurso”, y puede tratarse de cualquier tipo de información: un documento, una imagen, un servicio temporal, una colección de otros recursos, etc. Cada uno de estos recursos posee un identificador único que lo vuelve accesible para cualquier cliente o dispositivo que utilice HTTP, y es independiente de la implementación que lo subyace.

#### **4.4.2.2. SSE vs WebSockets**

Tanto WebSockets como los Server Sent Events (SSE) son tecnologías que facilitan la comunicación servidor-cliente. La utilización de una u otra depende fundamentalmente del caso de uso en cuestión.

WebSockets es una tecnología que permite la transferencia bidireccional de información en tiempo real. No es soportada por HTTP plano, lo cual implica que se deben manejar de forma manual muchas cuestiones que HTTP resuelve de manera nativa, aunque si hace uso de la conexión TCP para el envío de mensajes. En la actualidad es soportado por prácticamente cualquier browser.

Los Server Side Events, en cambio, forman parte del protocolo HTTP. Esto hace que algunas funcionalidades como la reconexión de clientes vengan incorporados de manera nativa. Son un mecanismo de envío asíncrono de mensajes servidor-cliente (unidireccional), y son sumamente sencillos de implementar. Considerando en primer lugar que la comunicación requerida en el caso de la transmisión de información desde el controlador es unidireccional, únicamente para transmitir los resultados de la rutina en tiempo real a la aplicación móvil, y que además se puede sacar provecho de ventajas como la reconexión de clientes integrada, se optó por trabajar con SSE.

### **4.4.3. Protocolo de comunicación entre el controlador y los nodos**

A la hora de definir la comunicación entre la Raspberry Pi y los nodos, se utilizó un protocolo personalizado desarrollado exclusivamente para estos dispositivos por Ramiro Romero Dapozo y Agustín García en el marco de dos Prácticas Profesionales Supervisadas (PPS) desempeñadas en el LINTI.

Se trata de un protocolo binario cuya unidad de dato es un paquete de longitud fija de 16 bytes, que puede ser encapsulado en datagramas UDP o transmitido a través de conexiones TCP dependiendo de su tipo y función. El protocolo se desarrolló luego de evaluar y testear otras alternativas, como el uso de MQTT, y se logró obtener una solución más performante y liviana (el uso de librerías de software para MQTT o HTTP que resultan costosas en términos de almacenamiento podría resultar en un problema del lado de los nodos).

# Capítulo 5

## Diseño de la solución

### 5.1. Vista general de la plataforma

La plataforma Tremün fue desarrollada para permitir el planeamiento, ejecución y medición de rutinas de entrenamiento deportivo, y teniendo como eje central el desarrollo de las capacidades cognitivas de los atletas. Existen tres componentes principales que interactúan entre sí para posibilitar estas funcionalidades:

- Una aplicación móvil para la creación y ejecución de rutinas y el análisis de los resultados.
- Un conjunto de nodos basados en luces LED y sensores de proximidad, que despliegan distintas figuras definidas en las rutinas.
- Una controladora central, compuesta por una Raspberry Pi que actúa como orquestadora de las rutinas, siendo el nexo entre la aplicación móvil y los nodos.

La Figura 5.1 ilustra esta arquitectura general de la plataforma.

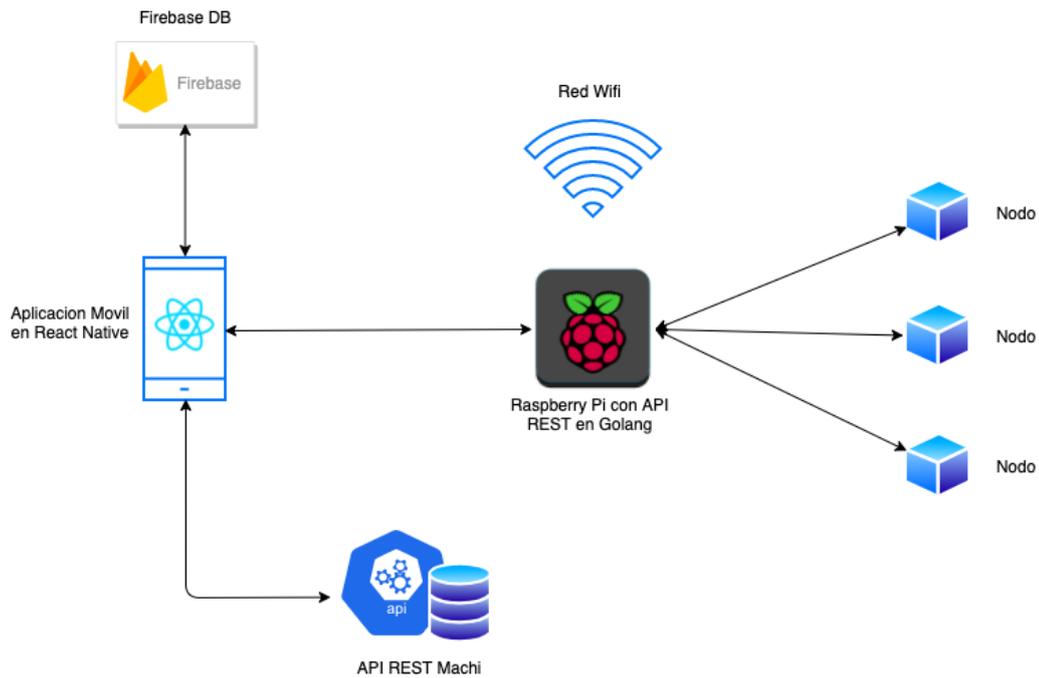


Figura 5.1: Vista general de la plataforma

La comunicación entre los diversos componentes es posible a través de una red WiFi montada por la Raspberry Pi. La comunicación entre la aplicación móvil y el controlador es bidireccional y se monta sobre el protocolo HTTP. La componente controladora provee una API REST que recibe las peticiones de ejecución o pausa de rutinas y devuelve los resultados en tiempo real mediante SSE. Con respecto a la comunicación entre el controlador y los nodos, la misma también utiliza la red WiFi a la que los nodos se conectan cuando se encienden, y es posible gracias a un protocolo específico mencionado en el capítulo 4.

Tal y como se observa en la imagen, la plataforma interactúa con una API REST expuesta por Machi, un sistema de información de entidades deportivas desarrollado en el LINTI. Los preparadores físicos y entrenadores que utilizan este sistema pueden administrar sus planteles para llevar registro de distintas métricas de cada uno de sus jugadores (existencia de lesiones, evaluaciones de rendimiento, percepción del esfuerzo, entre otras). En este marco, desde Tremün se busca poner a disposición una nueva fuente de recolección de datos para ampliar el abanico de información disponible en Machi.

## **5.2. La aplicación móvil**

### **5.2.1. Descripción general**

La aplicación móvil es la herramienta que permite a los usuarios la creación y ejecución de rutinas, el almacenamiento de las mismas para poder reutilizarlas y la visualización de los resultados en tiempo real. Además, mediante la sincronización con Machi, es posible obtener los datos de los planteles y deportistas relacionados al usuario de la sesión y almacenar los resultados.

### **5.2.2. Login**

Para poder almacenar las rutinas propias del usuario tuvimos que incluir en la aplicación un mecanismo de autenticación, con el objetivo de identificar al usuario de la sesión y poder asociar las rutinas al mismo. En el capítulo 4, se analizaron algunas alternativas de bases de datos para aplicaciones móviles identificando algunas ventajas y desventajas de cada una. Finalmente, se optó por utilizar Cloud Firestore, una de las soluciones provistas por Firebase, ya que era la única alternativa incluida en los servicios de Expo que ofrecía una base de datos NoSQL y no presentaba ninguna desventaja para nuestra aplicación. Al seleccionar esta base de datos, vimos la posibilidad de utilizar la autenticación provista por Firebase que permite utilizar diversas librerías de terceros, como el login a través de Google.

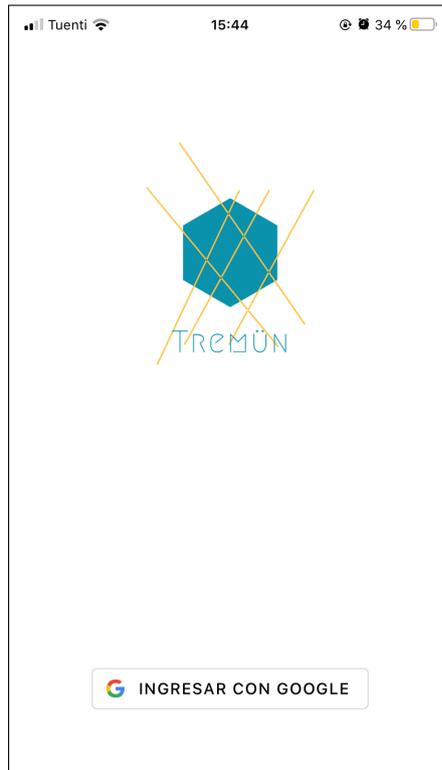


Figura 5.2: Pantalla de inicio de sesión

### 5.2.3. Rutinas de entrenamiento

En el contexto de Tremün, una rutina es una secuencia de pasos, en cada uno de los cuales se enciende una determinada cantidad de nodos con diversas figuras y colores, cuyo objetivo es el de mejorar el entrenamiento neuromuscular de los deportistas. Su duración puede determinarse en cantidad de pasos o de tiempo, y dependiendo de su configuración permite desarrollar diferentes capacidades cognitivas y físicas de los deportistas. La aplicación móvil provee distintos tipos de rutinas ejecutables que pueden ser configuradas de acuerdo a las necesidades de cada preparador físico o entrenador.

### 5.2.4. Rutina aleatoria

La definición de una rutina aleatoria permite la selección de múltiples colores y formas, los cuales se muestran en los distintos nodos de manera aleatoria en cada turno en el momento de la ejecución. Es una rutina altamente flexible, que permite una gran variedad de ejercicios de aplicación cognitiva, tanto individuales como grupales, así como también ejercicios de activación.

### 5.2.4.1. Descripción

Para programar la rutina de entrenamiento, se ingresa a la pantalla de configuración que permite modificar diversos parámetros para ajustar la ejecución de la rutina al plan de trabajo. Estos parámetros se muestran en una única pantalla para conseguir una buena experiencia de usuario.

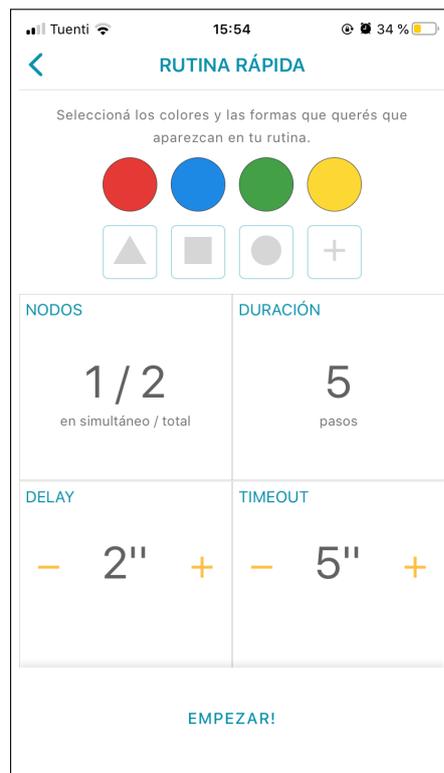


Figura 5.3: Pantalla de configuración

- Colores: Se seleccionan los distintos colores de encendido de los nodos.
- Figuras: Para cada uno de los colores seleccionados, se definen las figuras que pueden ser representadas.
- Nodos en simultáneo: Permite indicar la cantidad de nodos que se encienden simultáneamente en cada uno de los turnos.
- Nodos totales: Permite indicar la cantidad total de nodos que se utilizarán en la rutina. Debe ser un número igual o superior a la cantidad de nodos simultáneos.

- Duración: La duración total del ejercicio, puede medirse en pasos o en tiempo.
- Delay: El tiempo que tardan en volver a encenderse los nodos luego de cada turno.
- Timeout: El tiempo que permanecen prendidos los nodos antes de apagarse automáticamente.

Luego de finalizar la configuración de los parámetros, al dar inicio a la rutina, la aplicación muestra una nueva pantalla en la que se visualizan los resultados en tiempo real.

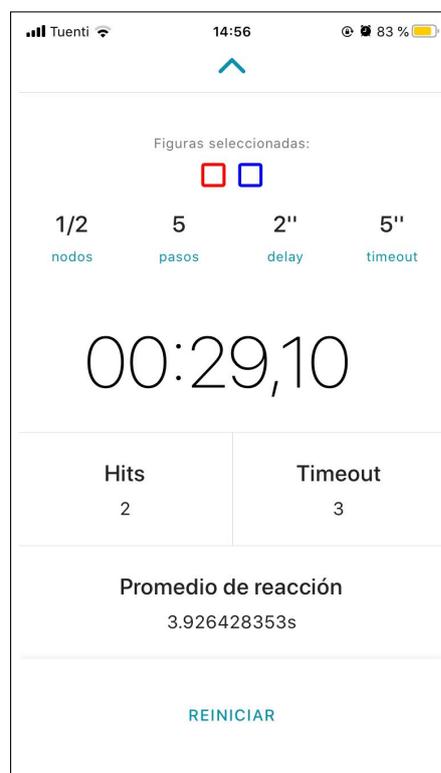


Figura 5.4: Pantalla de resultados

- Cronómetro: Contador del tiempo total de ejecución de la rutina.
- Hits: Cantidad de turnos en los que el deportista apagó algún nodo encendido.
- Timeouts: Cantidad de pasos en los que el deportista no apagó ningún nodo antes del apagado automático.

- Promedio de reacción: El promedio de reacción del deportista calculado en base a los hits.

#### 5.2.4.2. Ejemplos de aplicación

- Rutina de activación: En las opciones de configuración se seleccionan dos nodos totales, un nodo simultáneo, y el resto de los parámetros quedan a criterio del entrenador. El deportista se sitúa frente a dos nodos, y en cada turno debe apagar con su pie el nodo que se enciende.

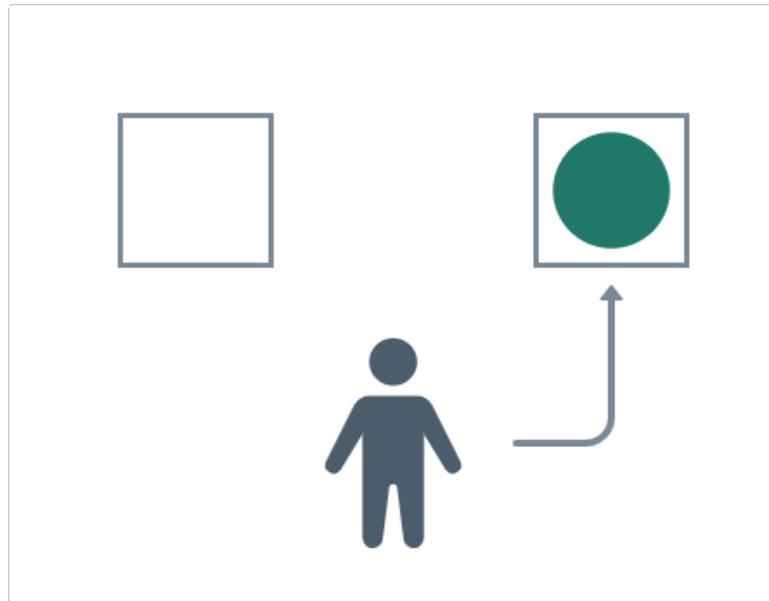


Figura 5.5: Rutina Aleatoria. Ejemplo 1. Paso 1.

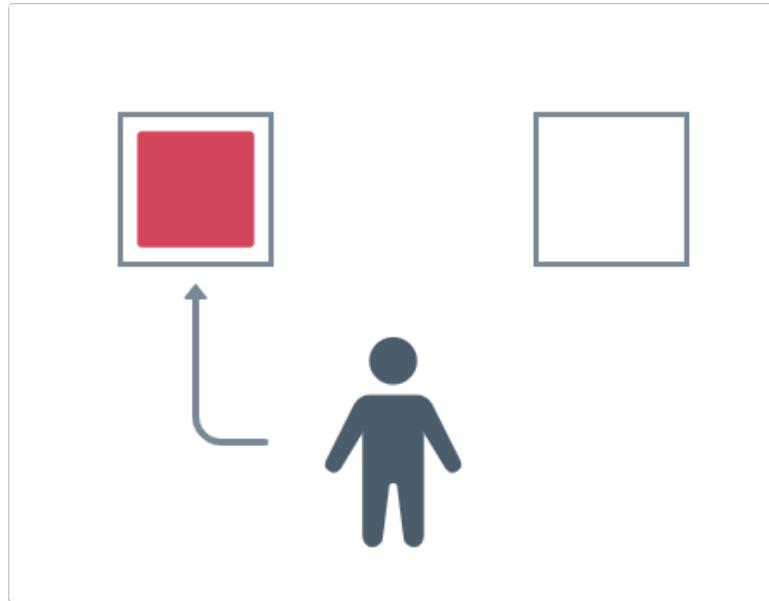


Figura 5.6: Rutina Aleatoria. Ejemplo 1. Paso 2.

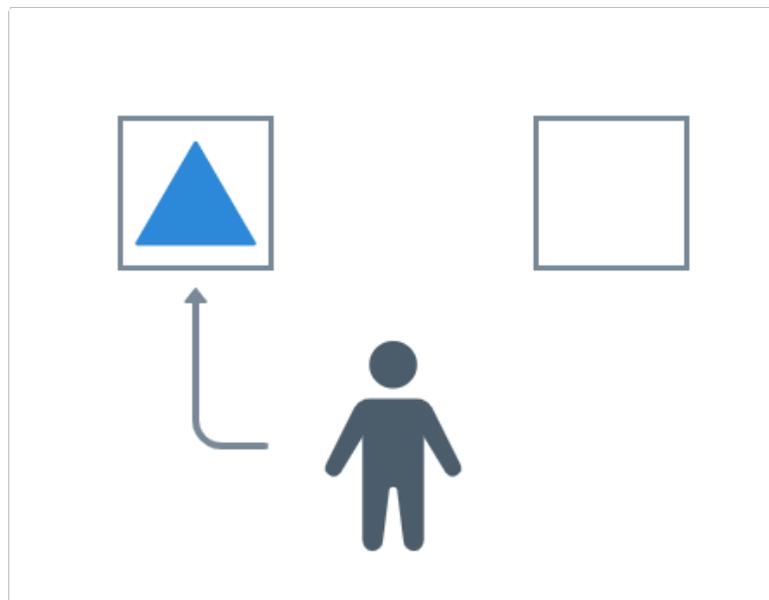


Figura 5.7: Rutina Aleatoria. Ejemplo 1. Paso 3.

- Rutina de entrenamiento de pases: En las opciones de configuración se selecciona un nodo total y un nodo en simultáneo. Se selecciona una cierta cantidad de combinaciones de figuras y colores, y se explica al deportista el comportamiento esperado ante la aparición de cada una de ellas (por ejemplo, círculo verde implica un pase hacia la izquierda, triángulo amarillo un pase hacia la derecha, triángulo azul un pase al frente). El deportista se ubica frente al nodo y debe reaccionar correctamente de acuerdo a cada figura.

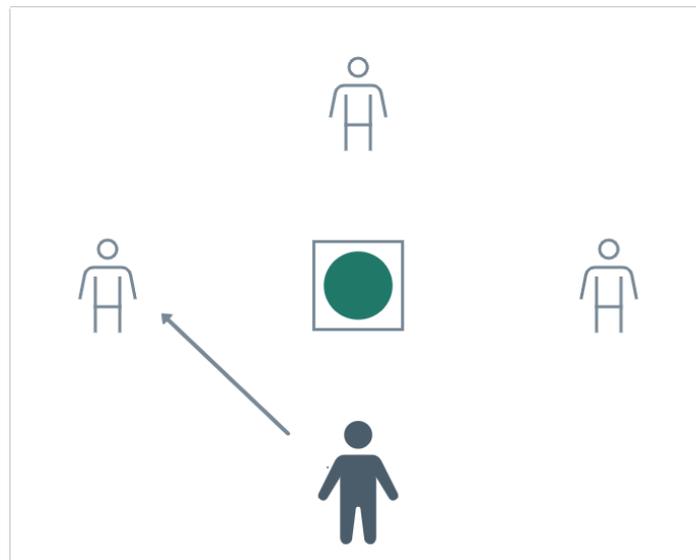


Figura 5.8: Rutina Aleatoria. Ejemplo 2. Paso 1.

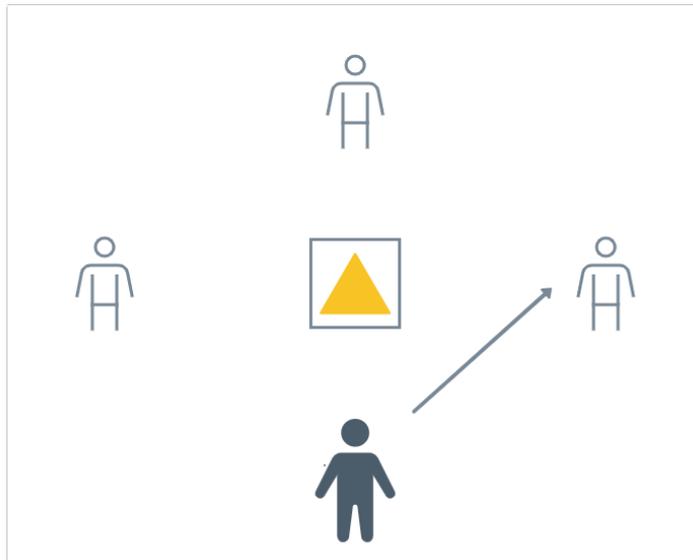


Figura 5.9: Rutina Aleatoria. Ejemplo 2. Paso 2.

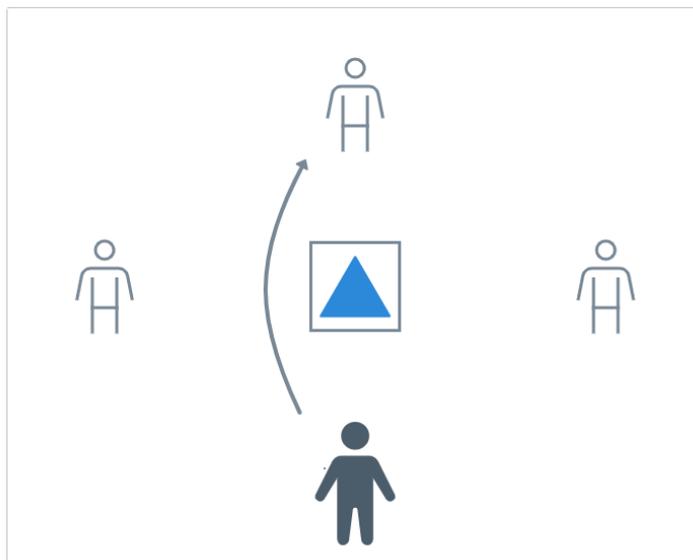


Figura 5.10: Rutina Aleatoria. Ejemplo 2. Paso 3.

- Rutina de toma de decisión: En las opciones de configuración se selecciona un nodo total y un nodo en simultáneo. Se selecciona una cierta cantidad de combinaciones de figuras y colores, y se asigna un significado

para cada color seleccionado y otro para cada figura (por ejemplo, verde implica un sprint hacia adelante, azul implica hacer flexiones de brazos, círculo implica repiqueteo en el lugar y triángulo implica abdominales). El deportista se ubica delante del nodo y, al encenderse la figura, el entrenador indicará al deportista si debe actuar en consecuencia de la figura que muestra el nodo o de su color. Ante la aparición de un triángulo verde, si el deportista recibe la indicación de responder al color deberá realizar un sprint, pero si se le indica obedecer a la figura deberá realizar flexiones de brazos.

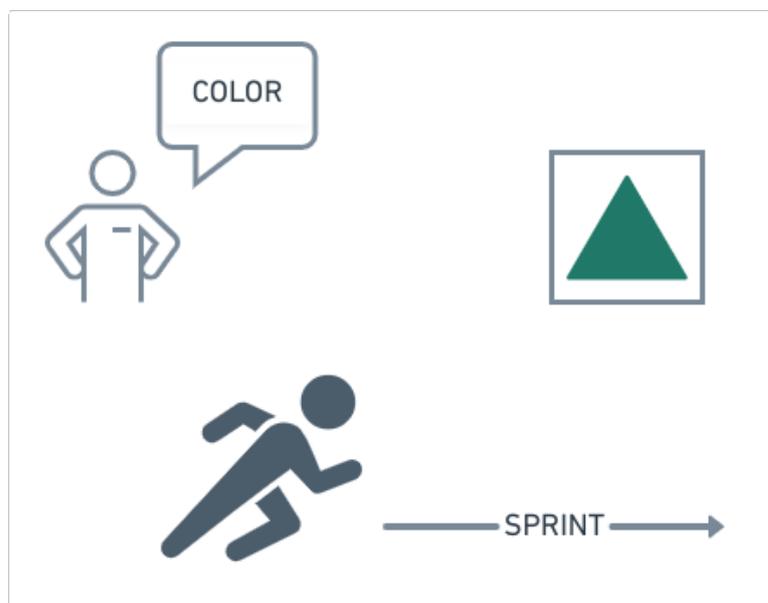


Figura 5.11: Rutina Aleatoria. Ejemplo 3. Paso 1.



Figura 5.12: Rutina Aleatoria. Ejemplo 3. Paso 2.

- Rutina de coordinación: Se seleccionan 3 nodos totales y un nodo simultáneo. Se disponen los nodos conformando un triángulo y el deportista se ubica en el centro del mismo. Cada vez que se enciende un nodo, el deportista debe colocar su pie izquierdo junto al nodo encendido y su pie derecho junto al nodo contiguo y repiquetear en el lugar hasta que el nodo se apague por timeout.

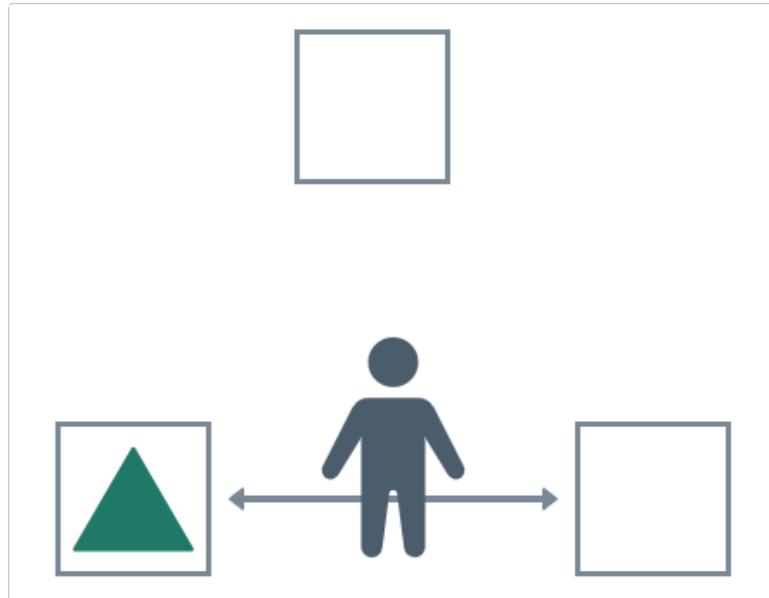


Figura 5.13: Rutina Aleatoria. Ejemplo 4. Paso 1.

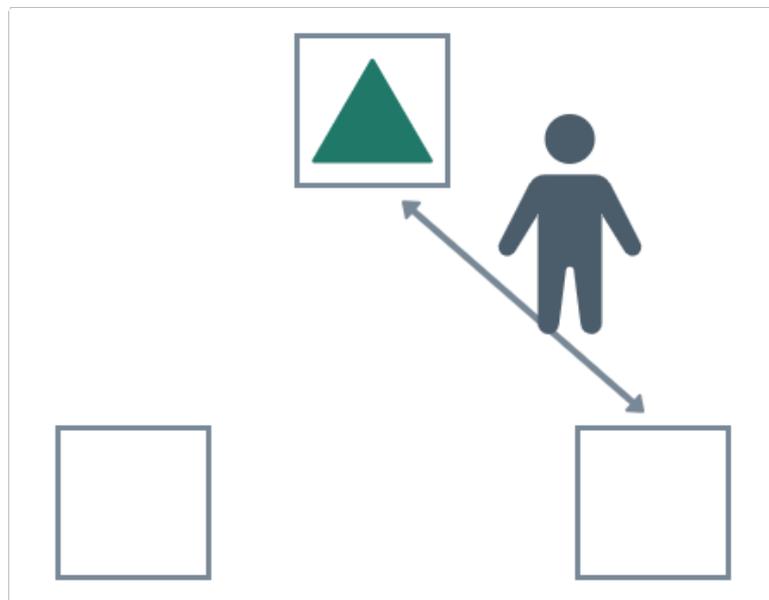


Figura 5.14: Rutina Aleatoria. Ejemplo 4. Paso 2.

## 5.2.5. Rutinas precargadas

La aplicación móvil cuenta con tres rutinas precargadas, las cuales brindan una alta flexibilidad para desarrollar una gran cantidad de ejercicios de entrenamiento diferentes, de acuerdo a la configuración que se le dé a sus parámetros de entrada.

### 5.2.5.1. Rutina selectiva

La rutina selectiva tiene su foco puesto en el entrenamiento cognitivo del deportista. El usuario selecciona un color correcto y uno o más colores incorrectos, y el deportista deberá apagar en cada turno el nodo que muestre el color correcto. La pantalla de configuración inicial de la rutina permite modificar diversos parámetros para ajustar la ejecución de la rutina al plan de trabajo.



Figura 5.15: Pantalla inicial.

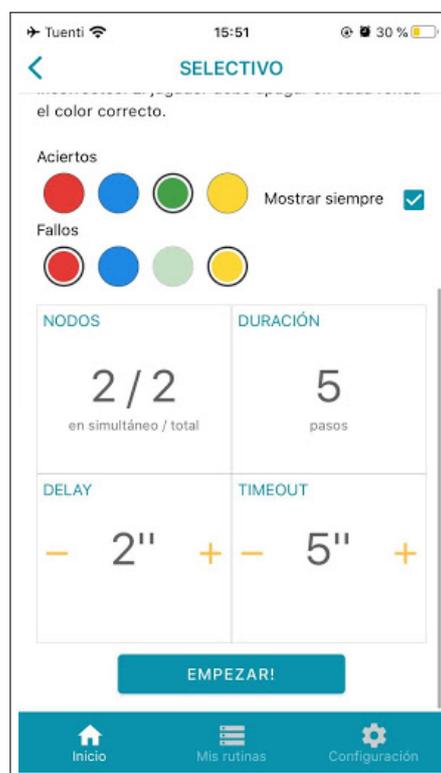


Figura 5.16: Pantalla inicial.

- Aciertos: Permite seleccionar un único color, el cual representa el color correcto que el deportista deberá apagar.

- **Mostrar siempre:** Si se tilda esta opción, en todos los turnos aparecerá un nodo con el color correcto. En caso de no tildarla, es posible que hayan turnos en los que ninguna opción sea correcta.
- **Fallos:** Permite seleccionar uno o más colores, siendo todos ellos distintos al color del acierto. Los colores seleccionados en este paso serán los colores incorrectos.
- **Nodos en simultáneo:** Permite indicar la cantidad de nodos que se encienden simultáneamente en cada uno de los turnos.
- **Nodos totales:** Permite indicar la cantidad total de nodos que se utilizarán en la rutina. Debe ser un número igual o superior a la cantidad de nodos simultáneos.
- **Duración:** La duración total del ejercicio, puede medirse en pasos o en tiempo.
- **Delay:** El tiempo que tardan en volver a encenderse los nodos luego de cada turno.
- **Timeout:** El tiempo que permanecen prendidos los nodos antes de apagarse automáticamente.

Luego de finalizar la configuración de los parámetros, al dar inicio a la rutina, la aplicación muestra una nueva pantalla en la que se visualizan los resultados en tiempo real.

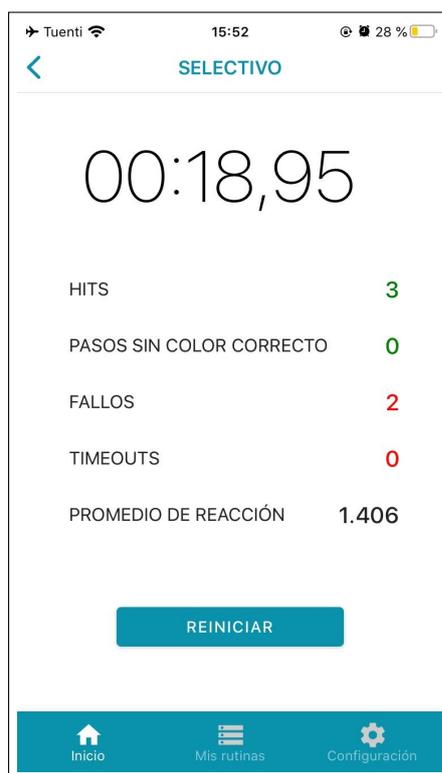


Figura 5.17: Rutina Selectiva. Pantalla de resultados.

- Cronómetro: Contador del tiempo total de ejecución de la rutina.
- Hits: Cantidad de turnos en los que el deportista apagó el nodo con el color correcto.
- Pasos sin color correcto: Cantidad de pasos en los que no apareció el color correcto en ninguno de los nodos.
- Fallos: Cantidad de pasos en los que el deportista apagó un nodo con un color incorrecto.
- Timeouts: Cantidad de pasos en los que el deportista no apagó ningún nodo antes del apagado automático y había un nodo con color correcto.
- Promedio de reacción: El promedio de reacción del deportista calculado en base a los hits.

Ejemplo de aplicación: rutina selectiva simple. En las opciones de configuración se seleccionan dos nodos totales, y dos nodos simultáneos. Se indica el color correcto y uno o más nodos incorrectos. El deportista se posiciona frente

a los nodos y en cada turno debe apagar el color correcto. Opcionalmente, es posible seleccionar la opción de “mostrar siempre” para complejizar el ejercicio y que el deportista deba identificar en primer lugar si existe una opción correcta o no.

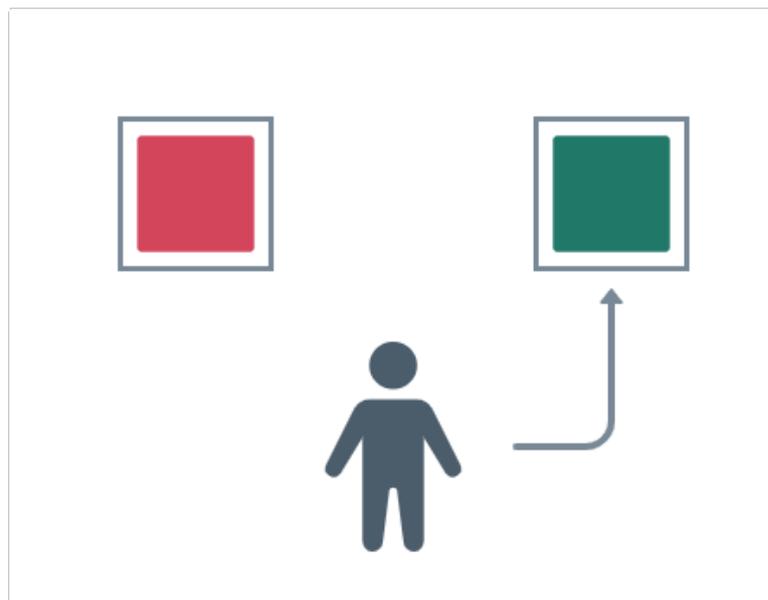


Figura 5.18: Rutina Selectiva. Ejemplo 1. Paso 1.

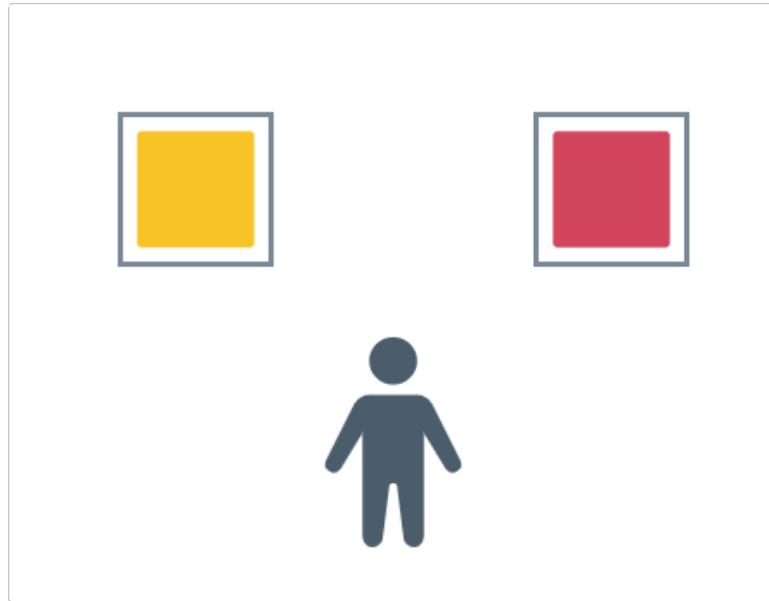


Figura 5.19: Rutina Selectiva. Ejemplo 1. Paso 2.

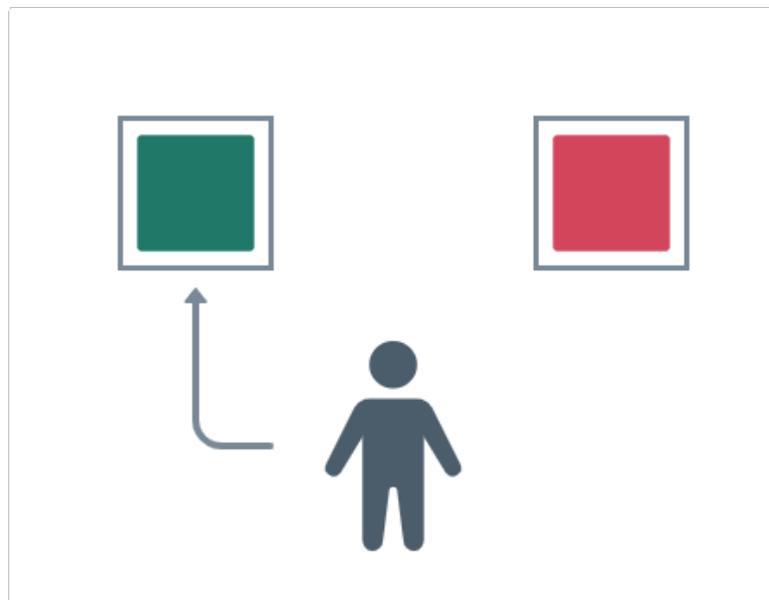


Figura 5.20: Rutina Selectiva. Ejemplo 1. Paso 3.

### 5.2.5.2. Rutina por equipos

La rutina por equipos fue pensada especialmente para realizar trabajos de entrenamiento en formato de competición, asignando un color diferente a cada uno de los equipos involucrados y contabilizando el puntaje obtenido por cada uno de ellos. En cada turno se encenderá un nodo por cada uno de los colores, y el equipo que logre apagar su nodo en el menor tiempo obtendrá un punto. La pantalla de configuración inicial de la rutina cuenta con tres parámetros.



Figura 5.21: Pantalla de inicio.



Figura 5.22: Pantalla de inicio.

- Colores: Los colores que representan a cada uno de los equipos que participan de la rutina.
- Nodos: Cantidad de nodos involucrados en la rutina, debe ser un número igual o superior a la cantidad de equipos.
- Pasos o duración: Se selecciona la cantidad de pasos o la duración en minutos y segundos de la rutina.

Luego de finalizar la configuración de los parámetros, al dar inicio a la rutina, la aplicación muestra una nueva pantalla en la que se visualizan los resultados en tiempo real.

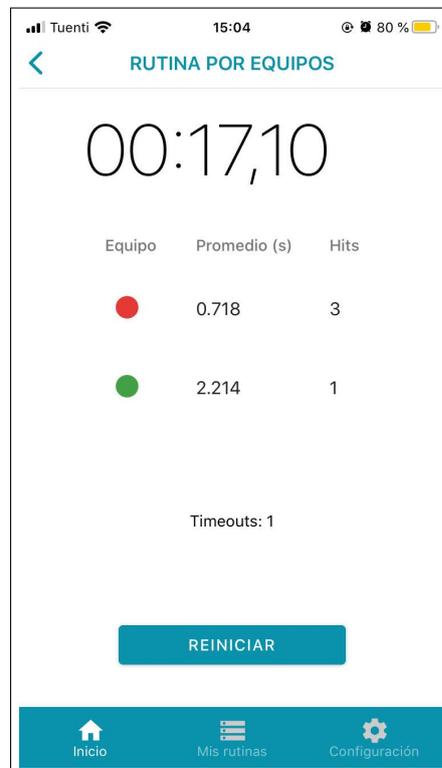


Figura 5.23: Rutina por equipos. Pantalla de resultados.

- Cronómetro: Contador del tiempo total de ejecución de la rutina.
- Equipo: Se indica el color del equipo.
- Promedio: El promedio de respuesta calculado en base a los hits de cada equipo.
- Hits: Cantidad de puntos acumulados por el equipo.
- Timeouts: Pasos en los que ningún equipo apagó su nodo antes del apagado automático.

Ejemplo de aplicación: competencia de reacción. En las opciones de configuración se seleccionan tantos nodos como equipos se tengan y un color por cada equipo. Alrededor de cada nodo se ubica un deportista de cada equipo

(por ejemplo, con 9 deportistas divididos en los equipos rojo, azul y amarillo, se utilizan 3 nodos, con un deportista de cada equipo rodeando cada nodo). En cada paso se enciende un nodo con el color de cada equipo, y el jugador perteneciente al equipo del color del nodo encendido deberá apagar su nodo antes que sus rivales.

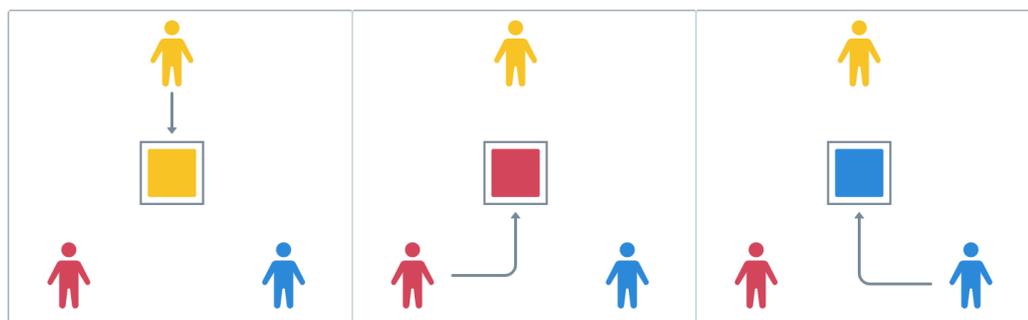


Figura 5.24: Rutina por equipos. Ejemplo 1. Paso 3.

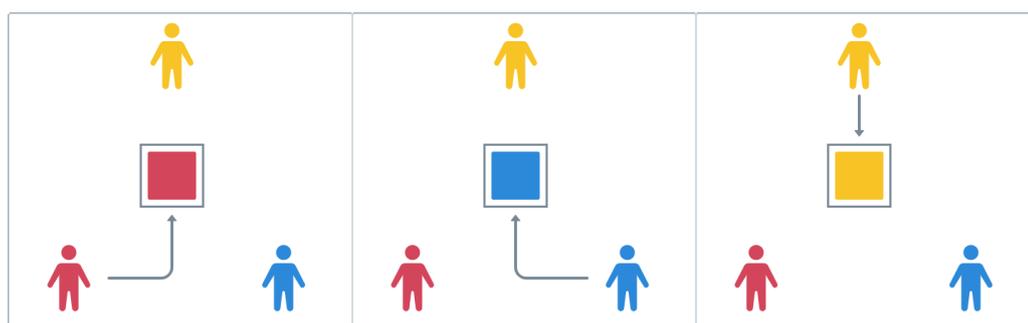


Figura 5.25: Rutina por equipos. Ejemplo 1. Paso 3.

### 5.2.5.3. Rutina de velocidad

A raíz de la constante interacción con entrenadores y entrenadoras de distintos clubes con el objetivo de relevar sus necesidades para que la plataforma desarrollada pueda adaptarse a las mismas, se decidió incluir también otra herramienta que posibilita el acceso a métricas orientadas a la evaluación de rendimiento: los test de velocidad.

El funcionamiento básico de un test de velocidad es sencillamente la medición del tiempo que tarda un atleta en desplazarse desde un punto a otro. A partir de esta premisa, y dependiendo de la intensidad, distancia o duración

de cada test, es posible extraer información referente no sólo a la velocidad de un atleta, sino también a su capacidad de aceleración, resistencia, cambio de ritmo o de dirección.

Si bien la decisión de contemplar este tipo de pruebas en el desarrollo de la plataforma Tremün surgió luego de comenzar el trabajo, y no aplica al concepto de entrenamiento cognitivo, los requerimientos en términos de hardware para poder realizar las mencionadas mediciones ya eran cubiertos por los equipos utilizados para la propuesta inicial, y considerando que resultan de extrema utilidad para la evaluación de los atletas, se tomó la determinación de incluir esta herramienta.

En general, los test de velocidad se realizan sencillamente con un cronómetro y luego anotando los resultados en un papel, o utilizando fotocélulas, una tecnología que sirve únicamente para realizar este test. Por estos motivos, consideramos de suma utilidad poder incluir dentro de la plataforma la opción de realizar el test de velocidad utilizando los nodos, para contabilizar los tramos a cubrir, y la aplicación para procesar y mostrar la información. Además, se incluye la opción de sincronización con la plataforma deportiva Machi para acceder a los clubes y planteles, y poder almacenar desde la aplicación móvil los resultados de las pruebas para cada deportista.

La idea de la rutina de velocidad es contabilizar el tiempo que tarda el deportista en llegar de un nodo al siguiente y el tiempo total desde el inicio del recorrido hasta el final del mismo. Se requiere un mínimo de dos nodos para contabilizar al menos un tramo, ya que el cronómetro empieza a funcionar al apagar el primer nodo. La pantalla de configuración inicial de la rutina cuenta con dos parámetros.



Figura 5.26: Rutina de Velocidad. Pantalla inicial.

- Nodos: Cantidad de nodos que se deberán encender para la rutina. El primer nodo es el que inicia el cronómetro y los restantes son los puntos en los que se calcula cada tramo.
- Color: El color con el que se encienden los nodos.

Luego de finalizar la configuración de los parámetros, al dar inicio a la rutina, la aplicación muestra una nueva pantalla en la que se visualizan los resultados en tiempo real.

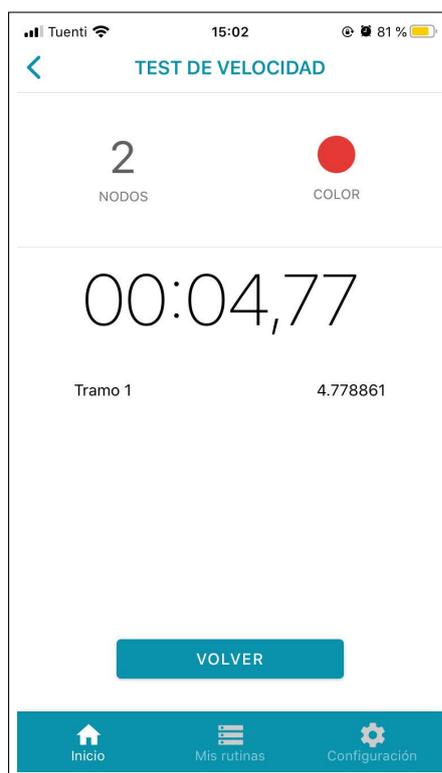


Figura 5.27: Rutina de Velocidad. Pantalla de resultados.

- Cronómetro: Contador del tiempo total de ejecución de la rutina.
- Tramo: Tiempo entre dos nodos.

Ejemplo de aplicación: test de velocidad lanzada. Se seleccionan 4 nodos, y se los dispone en línea recta a una distancia de 25 metros entre cada uno de ellos. El deportista debe correr del primero hasta el último pasando por los nodos intermedios, para registrar el tiempo del recorrido.

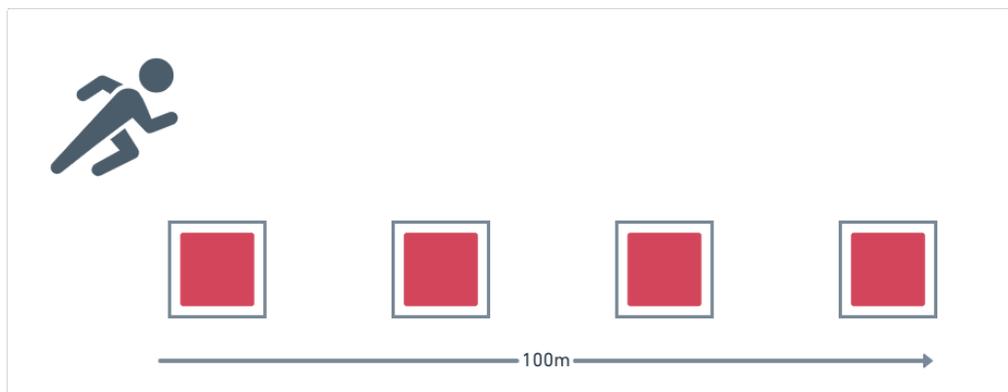


Figura 5.28: Rutina de Velocidad. Ejemplo 1. Paso 1.

### 5.2.6. Rutinas personalizadas

Las rutinas personalizadas son creadas por los usuarios de la aplicación a través de un formulario que permite posteriormente su almacenamiento en la nube (a través de Firebase), para poder ejecutarlas fácilmente en el futuro. El formulario de carga brinda las mismas opciones de configuración que la rutina aleatoria para otorgar a los entrenadores mayor flexibilidad, y permite agregar un nombre y una descripción. A diferencia de la rutina aleatoria, el formulario de rutina personalizada se realiza en distintos pasos para organizar de una mejor manera los parámetros de configuración. Una vez finalizada la carga, se muestra un resumen de todos los parámetros y se le permite al usuario guardar su rutina personalizada para luego ejecutarla.

A continuación se muestran las pantallas de la aplicación y el procedimiento para generar una rutina personalizada. En primer lugar, se ingresa desde la barra de navegación a la sección de “Mis rutinas”, donde se listan las rutinas del usuario.

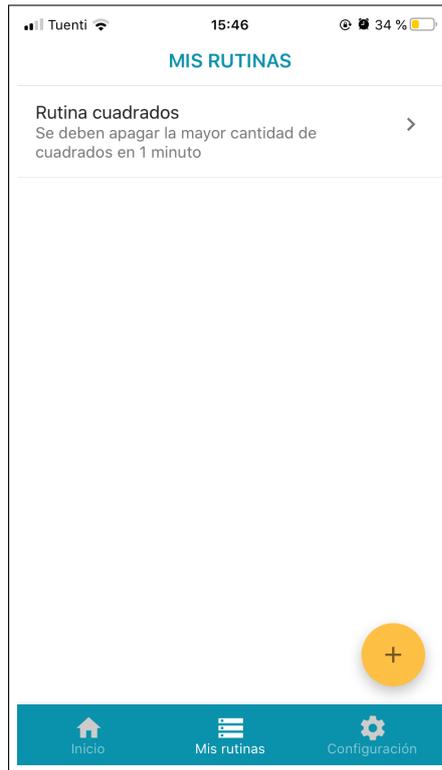


Figura 5.29: Listado de rutinas del usuario

Luego, presionando en el botón flotante se accede al formulario de carga que cuenta con seis pasos, donde se ingresa toda la información de la rutina personalizada.

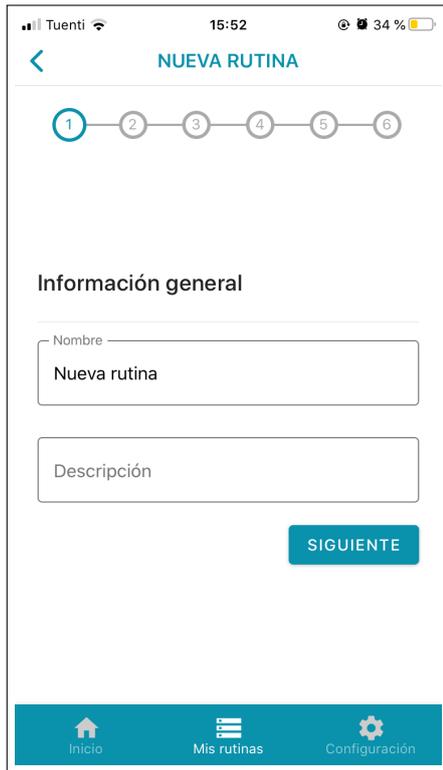


Figura 5.30: Formulario de alta

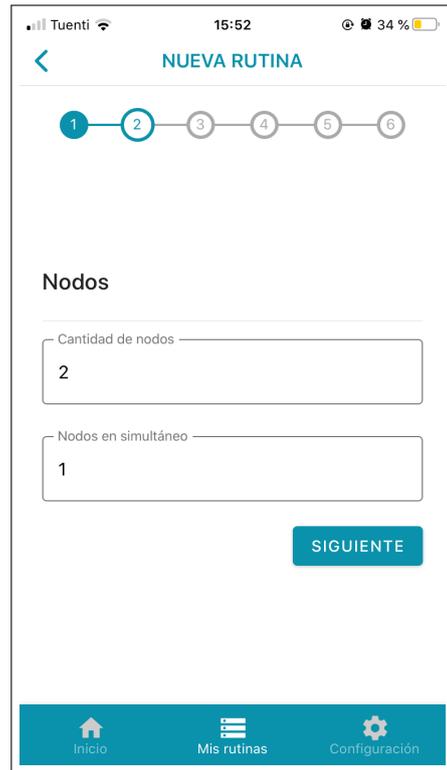


Figura 5.31: Formulario de alta

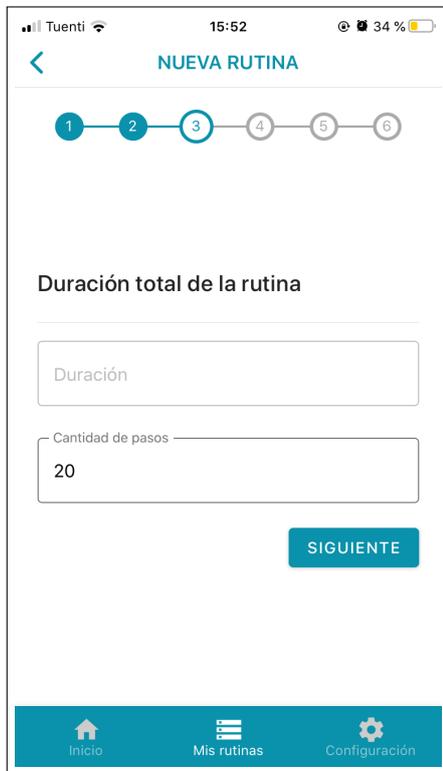


Figura 5.32: Formulario de alta



Figura 5.33: Formulario de alta

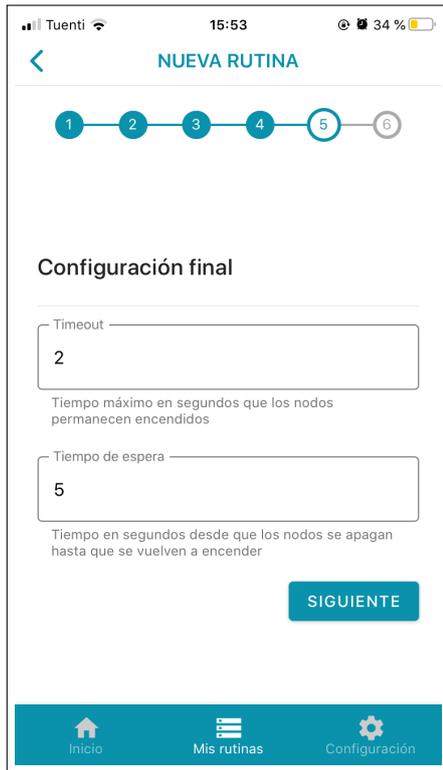


Figura 5.34: Formulario de alta

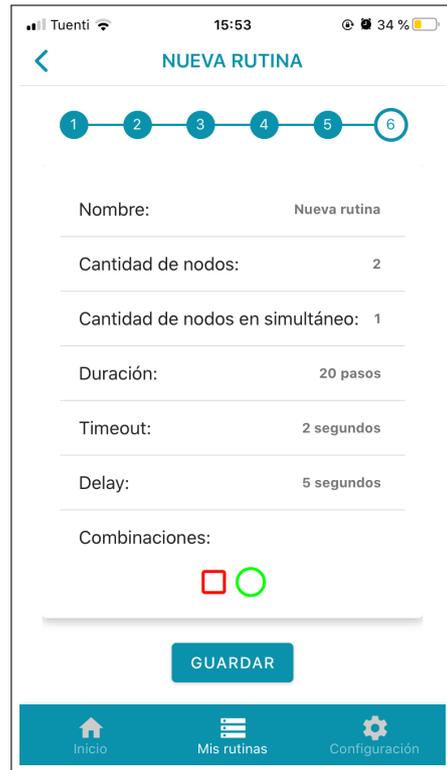


Figura 5.35: Formulario de alta

Por último, se muestra en el listado de rutinas del usuario la rutina generada en el formulario.

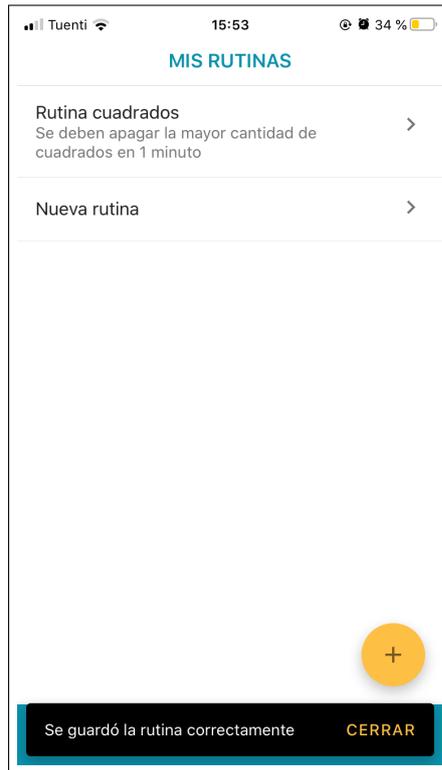


Figura 5.36: Listado de rutinas del usuario

A partir del listado de rutinas se puede acceder al detalle de cada una y realizar distintas acciones: modificar la rutina, eliminarla o enviarla al controlador para ejecutarla.

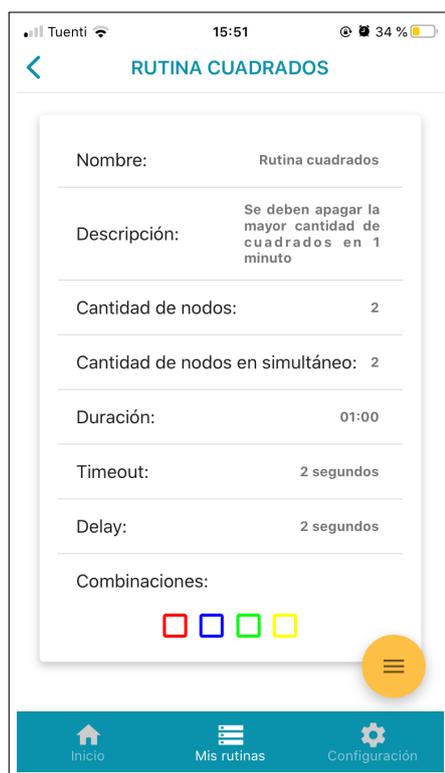


Figura 5.37: Detalle de la rutina

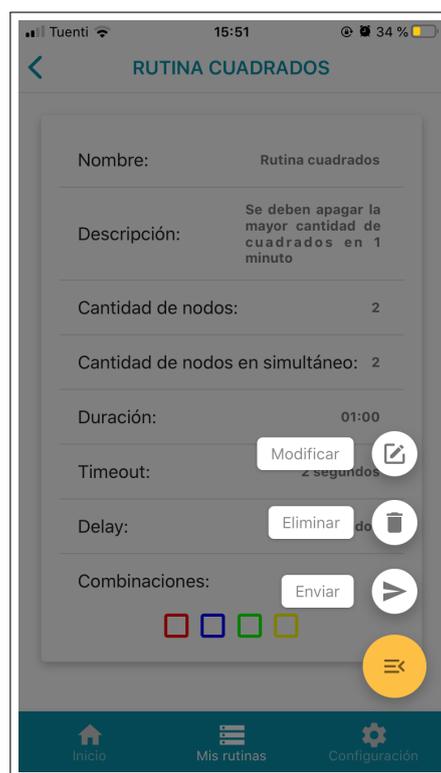


Figura 5.38: Acciones disponibles

### 5.3. El controlador

La tarea fundamental del controlador es la de coordinar y monitorear la ejecución de las rutinas programadas en la aplicación móvil. Para esto, en primer lugar, se trabajó para lograr abstraer lo mejor posible el proceso de ejecución de una rutina, a fin de comprender los parámetros esenciales para dicha ejecución. Más allá de que se construyó un endpoint dentro de la API para cada tipo de rutina, se lograron identificar ciertos valores comunes a todas ellas:

- Duración de la rutina: Todas las rutinas se ejecutan durante un lapso de tiempo predefinido, y todas ellas finalizan ante un evento concreto. Se determinaron dos formas de establecer la duración de una rutina desde la aplicación (a excepción de la rutina de velocidad lanzada que es un caso particular), una por tiempo y otra por pasos.
- Comportamiento de los nodos: En toda rutina es necesario establecer la duración del encendido de los nodos previo al apagado automático

(timeout), y el tiempo que deben permanecer apagados antes de volver a mostrar una figura (delay).

- Figuras: Es necesario indicar el conjunto de figuras y colores que serán utilizados en la rutina. El orden y diversidad de figuras es radicalmente diferente entre rutinas, contando siempre en mayor o menor medida con algún grado de aleatoriedad (ya sea referida a qué figura utilizar o al nodo en que dicha figura se representa).

A partir de la identificación de las funciones principales para la ejecución de una rutina, aparecieron dos desafíos iniciales para los que el lenguaje Go brindó una efectiva solución de manera nativa: la temporización de las rutinas, y el manejo de la aleatoriedad para evitar la previsibilidad en las ejecuciones.

Con respecto a la temporización, fue necesario encontrar una solución que no sólo permitiera controlar la duración total de la rutina, sino que también posibilitara el cálculo del tiempo de promedio de reacción, el apagado de los nodos por timeout y el encendido luego de cumplido el tiempo de delay indicado. En resumen, era necesario llevar cuenta, de manera simultánea, de diferentes eventos relacionados al paso del tiempo, y actuar en consecuencia.

Para lidiar con las problemáticas previamente enunciadas, se utilizó el paquete “time” [42] de Golang. Dicho paquete provee la funcionalidad necesaria para medir y representar el tiempo, tanto en términos de duración como absolutos. El mismo incluye también un tipo de dato denominado “Duration”, que representa un intervalo de tiempo y posee precisión a nivel de nanosegundos, junto con las funciones necesarias para interpretarlo en distintas unidades de medida.

La necesidad de atender eventos temporizados exigió una implementación capaz de manejar concurrentemente dichos eventos. Por poner un ejemplo concreto, luego de encender los nodos en un paso cualquiera de una rutina por tiempo, conviven concurrentemente dos procesos temporizados: por un lado es necesario contabilizar el tiempo indicado por el parámetro “timeout” para apagar los nodos en caso de que no se accione ningún apagado manual mediante los sensores, mientras que al mismo tiempo se debe contabilizar el tiempo total de la rutina, ya que la duración indicada por el parámetro “duration” puede cumplirse en cualquier momento. Es aquí donde entran en juego dos herramientas fundamentales de Golang: los canales y las “go routines”.

Las “go routines” son una característica del lenguaje Go, y se trata de ejecuciones independientes y simultáneas en relación a cualquier otra tarea en ejecución. Cada programa de Go contiene al menos una “go routine” principal, desde la cual se pueden iniciar múltiples ejecuciones concurrentes que se ejecutan en segundo plano. Por su parte, los canales son estructuras que permiten

la comunicación entre diversas “go routines” concurrentes. Es posible enviar un valor a través de un canal en una de las subrutinas en ejecución y recibir dicho valor en otra subrutina. La combinación de ambas herramientas es la que posibilita, por ejemplo, iniciar un temporizador en una “go routine” paralela para contabilizar la duración total de la rutina, y escuchar en un canal en la “go routine” principal para recibir el evento de finalización emitido en el canal por la subrutina concurrente.

En cuanto al problema de la aleatoriedad, el paquete “rand” [43] permite la generación pseudo-aleatoria de números. Utilizando como base las funciones de este paquete fue posible evitar la previsibilidad en el encendido de los nodos, algo fundamental para la aplicación de ejercicios que impliquen un desafío cognitivo.

Además de solucionar las principales problemáticas referidas a la lógica de ejecución de las rutinas, fue necesario construir una interfaz de comunicación entre la aplicación móvil y el controlador. Ésta comunicación debía posibilitar una interacción bidireccional, que permitiese a la aplicación iniciar y detener la ejecución de rutinas, y al controlador enviar en tiempo real los resultados de la misma. En base a los requerimientos planteados, se optó por el desarrollo de una API REST en el controlador (utilizando el paquete gorilla/mux[44]) para procesar las solicitudes de la aplicación móvil, y se utilizó el estándar SSE para el envío de datos en tiempo real (a través del paquete alexandrevicenzi/go-sse[45]).

## 5.4. Los nodos

Los nodos constituyen el punto de interacción entre la plataforma y el deportista, y cumplen dos funciones esenciales: por un lado, se encargan de mostrar en su pantalla las figuras configuradas desde la aplicación, y por el otro son responsables de notificar al controlador cuando uno de ellos es apagado por un atleta mediante el sensor de proximidad.

La interacción con dichos nodos se da a través de un protocolo binario sobre una conexión TCP para cada nodo, mediante el envío de paquetes de longitud variable. El tamaño mínimo del paquete es de 3 bytes, y el tercer octeto permite anticipar la longitud total del mismo. Algunos de los paquetes más importantes soportados por el protocolo son:

- HELLO: Es el primer paquete que se manda en una conexión establecida. Contiene un identificador de 48 bits y es enviado desde un nodo hacia el controlador.

- ID: Es el paquete de respuesta por parte del controlador al comando HELLO. Se utiliza para asignar un identificador al nodo.
- COMMAND: Se envía desde el controlador al nodo, e indica entre otras cosas el color con el que se debe iluminar y la figura que debe representar.
- TOUCHED: Este comando se envía desde el nodo al controlador, y notifica a este último que fue apagado mediante el sensor de proximidad.

# Capítulo 6

## Pruebas de campo

### 6.1. Introducción

Para llevar adelante las pruebas de campo se establecieron contactos con diversos preparadores físicos y entrenadores, con el objetivo de testear el funcionamiento de la plataforma en un entorno de entrenamiento real.

La medición del impacto de las herramientas de entrenamiento cognitivo en el desempeño de los deportistas es sumamente compleja, por lo que la planificación de las pruebas de campo se centró en el análisis del funcionamiento de la plataforma, la interacción entre sus componentes, y principalmente la apreciación de los entrenadores.

Las pruebas se iniciaron a principios del año 2020 con una versión inicial de la plataforma, y con la meta puesta en seguir analizando el funcionamiento de Tremün a medida que las nuevas características estuviesen integradas. Lamentablemente, la irrupción de la pandemia del coronavirus y el dictamen de un Aislamiento Social Preventivo y Obligatorio en todo el país, paralizó el entrenamiento deportivo y con él la posibilidad de realizar nuevas pruebas.

Ante tal situación, nos vimos obligados a modificar la planificación original, y optamos por realizar una encuesta a quienes tuvieron la posibilidad de probar la versión inicial de la plataforma para analizar sus conclusiones. La encuesta, a su vez, abre interrogantes en relación al entrenamiento cognitivo a nivel general, mas allá de la plataforma Tremün. La confección de la encuesta se vio condicionada por las funcionalidades que formaron parte del prototipo con el cual se realizaron las pruebas.

## 6.2. Conformación y análisis de la encuesta

La encuesta, desarrollada mediante la herramienta Google Forms, contó con seis preguntas, y su objetivo principal fue el de conocer las sensaciones iniciales generadas por la plataforma, e indagar sobre la situación actual del entrenamiento cognitivo a nivel local:

### Tremün: Plataforma para entrenamiento cognitivo aplicado al deporte

El objetivo de la presente encuesta es conocer tu opinión acerca del entrenamiento cognitivo a nivel general, tu percepción acerca del mismo en nuestro país, y qué impresión te generó interactuar con la plataforma Tremün.

Se sugiere comenzar por la reproducción del video para ver el funcionamiento general de la plataforma.

Tremün: Plataforma para entrenamiento cognitivo aplicado al deporte



Nombre

Tu respuesta

Institución(es) en la que trabajas

Tu respuesta

Figura 6.1: Encuesta. Parte 1.

¿Utilizas habitualmente en tus entrenamientos algún dispositivo orientado al entrenamiento cognitivo? ¿Con qué objetivo?

Tu respuesta \_\_\_\_\_

¿Notas algún impacto en el plantel producto de la utilización de dispositivos de entrenamiento cognitivo? ¿Cuál?

Tu respuesta \_\_\_\_\_

¿Consideras que existe algún obstáculo para la utilización de dispositivos de entrenamiento cognitivo en Argentina?

- Escasa oferta
- Costo elevado
- Desconozco el producto
- No me interesa
- Baja confianza
- Otros: \_\_\_\_\_

¿Cómo calificarías tu experiencia con el uso de la plataforma Tremün?

	1	2	3	4	5	
Muy mala	<input type="radio"/>	Muy buena				

Figura 6.2: Encuesta. Parte 2.

¿Tendrías interés en incorporar la plataforma Tremún a tus entrenamientos de manera regular? ¿Por qué?

Tu respuesta

---

¿Te parece útil la posibilidad de realizar tests de velocidad a través de la plataforma? ¿En la actualidad cómo realizas estas pruebas?

Tu respuesta

---

Muchas gracias!



Enviar

Figura 6.3: Encuesta. Parte 3

Con respecto a los encuestados, se obtuvieron respuestas de entrenadores y preparadores físicos de diversas instituciones que abarcan desde clubes deportivos hasta gimnasios.

A raíz del análisis de las respuestas pudimos observar en primer lugar que la mayoría de los encuestados respondió que actualmente no utiliza dispositivos de entrenamiento cognitivo en sus entrenamientos de manera habitual. Algunas excepciones que pudimos observar fueron señaladas por Mariela Arangio (Club Atlético River Plate y CENARD) que confirmó un uso regular “Con el objetivo de que los deportistas mejoren sus habilidades perceptivo cognitivas y potencien su rendimiento deportivo”, y Carla (Santa Bárbara Hockey Club) cuya meta esta puesta en “Desarrollar la concentración al mismo tiempo que

las capacidades motrices”. Por otro lado, Lara Yantorno (también de Santa Bárbara Hockey Club) no utiliza estos dispositivos pero sí incluye entrenamientos enfocados al desarrollo de las capacidades cognitivas de su plantel, e indicó: “No tenemos en el club dispositivos especiales para ello, pero a veces optamos por entrenamientos cognitivos utilizando por ejemplo conos de colores con referencias auditivas”, y en la misma línea Miguel Julián Otero (Sociedad de Fomento La Cumbre) remarcó “No cuento con tecnologías pero si utilizo diferentes ejercicios que generen respuestas cognitivas como números, letras y colores para mejorar la capacidad de repuesta efectiva y eficaz”.

A la hora de indagar acerca de los obstáculos para su utilización, notamos que en primer lugar se señala el elevado costo de los dispositivos, seguido de cerca por la escasa oferta en el mercado a nivel local. A continuación es posible visualizar a través de un gráfico las opciones seleccionadas por los encuestados:

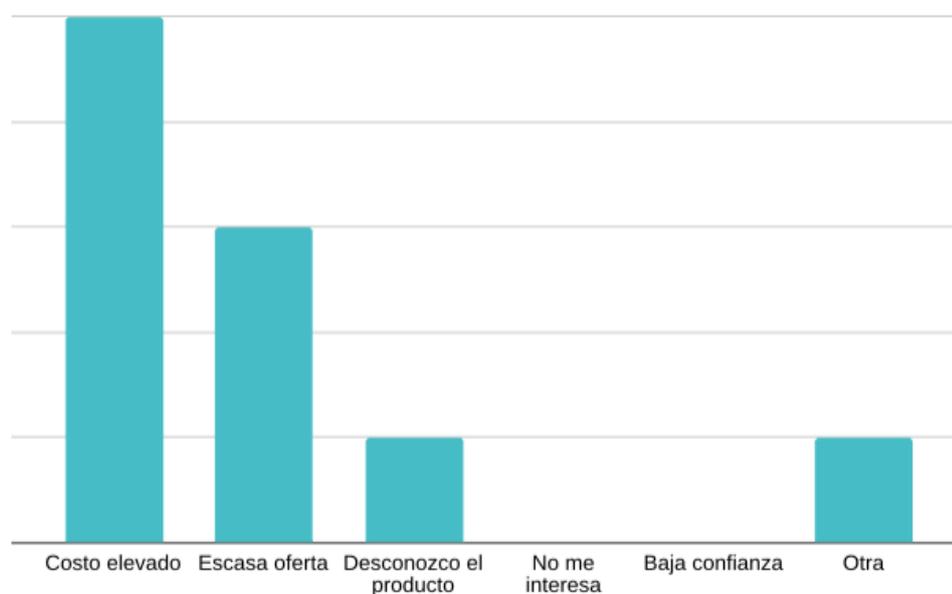


Figura 6.4: Obstáculos para la utilización de dispositivos de entrenamiento cognitivo en Argentina

A pesar de la baja frecuencia en el uso de dispositivos de entrenamiento cognitivo, la situación cambia cuando se consulta acerca de su impacto en el plantel. Al ser consultado acerca de los efectos del uso de estos dispositivos, Gonzalo Gonzalez (Club Universitario de La Plata) afirmó que “El entrenamiento con dispositivos cognitivos tienen una buena recepción en los deportistas, les divierte y están buenos para salir de los entrenamientos clásicos.”,

mientras que Facundo Francesio (Gimnasio Timko) señaló que notó “un impacto en mis alumnxs al utilizar con ellos la plataforma Tremün. En primer lugar se sintieron motivados a la hora de entrenar con la presencia de un dispositivo tecnológico de calidad, orientado específicamente para el entrenamiento cognitivo. En segundo lugar, además, una vez familiarizados con el dispositivo, lxs alumnxs tuvieron un mayor y mejor rendimiento en los diferentes ejercicios de entrenamiento cognitivo utilizando dicha plataforma.”. A su vez, Rodrigo Barrios (Asociación de Fútbol Argentino) afirmó que los incluye ocasionalmente en sus entrenamientos ya que aportan “Variabilidad en los trabajos” mientras que Mariela Arangio indicó que en su experiencia “Mejora la velocidad de respuesta, la atención, la concentración, la memoria, la toma de decisión”.

Con respecto a la plataforma Tremün, la totalidad de las respuestas obtenidas calificaron la experiencia como “Buena” o “Muy Buena”, señalando un interés en incorporarla como una herramienta regular en sus entrenamientos. Es interesante observar que más allá de que hay un interés común en la adopción de ésta tecnología, a la hora de fundamentar la decisión de su incorporación podemos observar motivos muy variados. Por citar algunos ejemplos, Isidro Barriola (Gimnasio Korper) remarca la importancia de ampliar el abanico de herramientas disponibles ya que “La variabilidad del entrenamiento permite al deportista potenciar su rendimiento.”, Pedro Martín Coleff (Club de Gimnasia y Esgrima La Plata) afirma que le gustaría incorporarlo “Para evaluaciones”, Lara Yantorno calificó a la plataforma como “Una herramienta interesante para incorporar diferentes factores en los trabajos técnico tácticos (no hablo de la parte de preparación física porque no estoy en esos entrenamientos, aunque creo que también sería interesante).”, Juan Vilchez (Club Curuzú Cuatia) justificó su interés al afirmar que “Es una herramienta que favorece la toma de decisiones.”, mientras que Facundo Francesio sostuvo: “Tendría interés en utilizar la plataforma Tremün en los entrenamientos que planifico de forma regular, ya que es una herramienta muy práctica y útil, así como un medio novedoso y muy atractivo para el desarrollo del entrenamiento cognitivo y neuromuscular de mis alumnxs.”.

Por último, se pudo observar un amplio consenso entre los encuestados en relación a la gran utilidad que significa la posibilidad de realizar también los tests de velocidad mediante la plataforma, que actualmente realizan principalmente mediante fotocélulas o dispositivos de GPS, o sencillamente a través de un cronómetro.

### 6.3. Análisis y conclusiones

El análisis de los datos relevados reafirma la necesidad de ampliar la oferta de dispositivos de entrenamiento cognitivo aplicado al deporte, ya que es posible observar un alto interés en su utilización que contrasta fuertemente con el bajo uso de los mismo de manera regular en la actualidad.

El hecho de que se señalen el costo elevado y la baja oferta como principales causas de la ausencia de estos dispositivos en los entrenamientos pone en evidencia la brecha que existe entre el entrenamiento de alto rendimiento a nivel internacional y nuestra situación en el plano nacional (profundizada si se trata del deporte amateur). La situación observada se alinea por lo tanto con el objetivo de brindar una solución de calidad a un bajo costo para un mercado con una demanda creciente pero sin posibilidades materiales de acceso a la oferta tecnológica actual.

La amplia variedad de razones enunciadas para justificar la adopción de Tremün demuestra también la versatilidad que posee la tecnología desarrollada a la hora de incorporarse a un entrenamiento.

# Capítulo 7

## Conclusiones y trabajos futuros

El presente capítulo describe las conclusiones obtenidas luego del proceso de desarrollo de la plataforma, así como las posibles líneas de trabajo a futuro.

### 7.1. Conclusiones

Para el desarrollo de la presente tesina se llevó adelante en primera instancia un estudio sobre el entrenamiento cognitivo, analizando su historia, actualidad, campos de aplicación y resultados. Se investigaron tanto sus aplicaciones en entornos deportivos, cuyo principal objetivo es el perfeccionamiento de habilidades, como aquellas orientadas a la rehabilitación de lesiones o al entrenamiento compensatorio.

Posteriormente, se analizaron las soluciones tecnológicas presentes en el mercado que tienen como eje central este tipo de entrenamiento. Se estudiaron las características comunes a todas ellas y también las principales problemáticas que se presentan actualmente en el mercado, destacando principalmente la dificultad para acceder a estos dispositivos fuera de la élite deportiva. El elevado costo de las plataformas disponibles en la actualidad hace que resulten imposibles de adquirir para el entrenamiento a nivel amateur.

Fue necesario también un análisis de los entornos más frecuentes en los que se llevan a cabo los entrenamientos deportivos. Se identificaron características importantes como la ausencia de conectividad o de fuentes de energía, por lo que la necesidad de soportar la ejecución de rutinas sin acceso a internet así como la de permitir el funcionamiento de la plataforma sin una fuente de alimentación permanente también fueron centrales en el desarrollo.

Una vez identificadas las funcionalidades deseadas y las problemáticas a solucionar, se procedió a definir el conjunto de herramientas para llevar adelante el desarrollo. Se diseñó la arquitectura de la plataforma, integrada por

tres componentes principales: una aplicación móvil, una controladora y un conjunto de nodos. A partir de la arquitectura planteada se estudiaron diversas tecnologías para implementar cada uno de los componentes hasta alcanzar la solución definitiva.

Para la aplicación móvil se llevó adelante un análisis de la distribución de mercado de los dispositivos móviles, identificando una gran fragmentación en cuanto a los sistemas operativos, con un absoluto predominio de iOS y Android. La situación observada nos llevó a buscar alternativas de desarrollo que permitieran obtener un producto final instalable en ambas plataformas, por lo que luego de una comparativa entre el desarrollo nativo y el amplio abanico de herramientas para desarrollo multiplataforma se optó por la utilización de React Native, un framework para construir aplicaciones nativas para iOS y Android utilizando React.

Con respecto a la controladora, se optó por utilizar una Raspberry Pi, sobre la cual se desarrolló una API REST en el lenguaje Golang. La Raspberry permitió montar una red Wi-Fi a la cual se conectan tanto la aplicación móvil como los nodos para intercambiar mensajes. El lenguaje Golang fue elegido por su gran capacidad para trabajar con eventos concurrentes y su potencia de procesamiento.

En cuanto a los nodos, se utilizaron dispositivos construidos dentro del marco de dos PPS (Práctica Profesional Supervisada) desempeñadas en el LINTI, que cuentan una pantalla de luces LED para mostrar distintas figuras y colores, y con un sensor de proximidad que permite a los deportistas interactuar con los mismos.

A la hora de determinar las rutinas a incluir en la aplicación móvil, las mismas fueron seleccionadas teniendo como principal objetivo brindar flexibilidad a preparadores físicos y entrenadores para abarcar la mayor cantidad posible de ejercicios. Se tomaron también como referencia los desarrollos existentes en otras plataformas de entrenamiento cognitivo, así como las variables cognitivas que resultan de interés a la hora de efectuar un entrenamiento. Se incluyó también la posibilidad de crear rutinas personalizadas y almacenarlas para su posterior ejecución, para mejorar la facilidad de uso de la aplicación.

La plataforma Tremün también implementa la integración con el sistema Machi, desarrollado en el LINTI, con el objetivo de permitir el registro de los resultados de la performance de los deportistas para su posterior análisis. Al autenticarse en Machi, los entrenadores y preparadores físicos pueden acceder desde la aplicación móvil de Tremün a los listados de planteles y jugadores que tienen asignados, ejecutar pruebas con los deportistas y posteriormente almacenar los resultados de manera centralizada junto al resto de la información.

Como resultado del trabajo enunciado, se obtuvo una plataforma que cuen-

ta con una aplicación móvil disponible en forma gratuita que puede ejecutarse en los dos sistemas operativos principales, con componentes de hardware de bajo costo y funcionalidades competitivas con otras soluciones disponibles en el mercado. Tremün constituye una nueva alternativa en lo que respecta al entrenamiento cognitivo en el deporte, que permite pensar en una democratización en el acceso a tecnologías que hoy en día se encuentran reservadas casi exclusivamente a la elite deportiva.

## 7.2. Trabajos futuros

Durante el desarrollo de la tesina surgieron nuevas ideas y necesidades, producto principalmente de la interacción con los usuarios finales de la plataforma, que excedían el alcance planteado inicialmente. Sin embargo, el diseño de Tremün fue pensado para permitir la futura integración de nuevas funcionalidades.

En primer lugar, la extensión más evidente que puede pensarse es la creación de nuevos tipos de rutinas, que brinden una flexibilidad aun mayor a preparadores físicos y entrenadores. Un ejemplo de esto podría ser una rutina por pasos, en las que se programe de manera particular cada uno de los pasos de la ejecución indicando cuántos nodos deben encenderse y seleccionando la forma y color para cada uno de ellos de manera individual.

En la versión actual de la plataforma, existe la posibilidad de una sincronización con el sistema Machi para obtener jugadores y planteles que un entrenador tiene a cargo, y para almacenar los resultados del test de velocidad. Versiones futuras podrían profundizar esta funcionalidad, permitiendo la exportación de los datos recolectados en diversos formatos (.csv, .xlsx, .json) para poder trabajar con ellos utilizando herramientas externas. Podría pensarse también un sistema de gestión de planteles y jugadores propio de Tremün.

Por último, sería interesante brindar a entrenadores y preparadores físicos la posibilidad de compartir rutinas de entrenamiento con otros colegas mediante la distinción entre rutinas públicas y privadas.

# Bibliografía

- [1] Ramírez Silva W. «La neurocognición en los procesos de entrenamiento deportivo». En: *6° Seminario Internacional de Entrenamiento Deportivo, Universidad de Antioquia* (2007).
- [2] *Fitlight*. <https://www.fitlighttraining.com/>.
- [3] *SmartGoals*. <https://smartgoals.nl/>.
- [4] *Led Trainer*. <http://ledtrainer.com/>.
- [5] Walton C.C.; Keegan R.J.; Martin M.; Hallock H. «The Potential Role for Cognitive Training in Sport: More Research Needed». En: *Frontiers in Psychology* (2018).
- [6] Omoregie P. «The impact of technology on sport performance». En: (2016).
- [7] David J.H.; Mark R.W.; Samuel J.V. «A Systematic Review of Commercial Cognitive Training Devices: Implications for Use in Sport». En: *Sport and Health Sciences, University of Exeter* (2018).
- [8] Jones R.N. «Cognitive Training Improves Cognitive Performance, but What Else?» En: *Journal of the American Geriatrics Society* (2018).
- [9] Wykes T.; Huddy V.; Cellard C.; McGurk S.R.; Czobor P. «A meta-analysis of cognitive remediation for schizophrenia: methodology and effect sizes». En: *The American journal of psychiatry* (2011).
- [10] Downey A.; Stroud C.; Landis S. «Preventing Cognitive Decline and Dementia: A Way Forward». En: *National Academies Press (US)* (2017).
- [11] Keshavan MS.; Vinogradov S.; Rumsey J.; Sherrill J.; Wagner A. «Cognitive training in mental disorders: update and future directions». En: *Am J Psychiatry* (2014).
- [12] Harry Hallock y col. «Cognitive Training for Post-Acute Traumatic Brain Injury: A Systematic Review and Meta-Analysis». En: *Frontiers in Human Neuroscience* (2016).

- [13] Wilson M.; Mann D.; Farrow D.; Nielsen T. «Does Visual-Perceptual Training Augment the Fielding Performance of Skilled Cricketers?» En: *International Journal of Sports Science Coaching* (2011).
- [14] Bellinga P.K.; Ward P. «Time to start training: A review of cognitive research in sport and bridging the gap from academia to the field». En: *Procedia Manufacturing* (2015).
- [15] *Neural Trainer*. <https://www.neuraltrainer.com/>.
- [16] Patil C.S.; Karhe R.R.; Aher M.A. «Development of Mobile Technology: A Survey». En: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* (2012).
- [17] Okediran O.O.; Arulogun O.T.; Ganiyu R.A. «Mobile Operating Systems and Application Development Platforms: A Survey». En: *Journal of advancement in engineering and technology* (2014).
- [18] Delia L.; Galdámez N.; Thomas P.J.; Corbalan L.; Pesado P. «Multi-platform mobile application development analysis». En: *International Conference on Research Challenges in Information Science* (2015).
- [19] *StatCounter*. <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [20] *React*. <https://es.reactjs.org/>.
- [21] *React Native Facebook*. <https://engineering.fb.com/android/react-native-bringing-modern-web-techniques-to-mobile/>.
- [22] Fredrik Carlström O.A. «Evaluation Targeting React Native in Comparison to Native Mobile Development». 2016.
- [23] Hansson N.; Vidhall T. «Effects on performance and usability for cross-platform application development using React Native». 2016.
- [24] *Expo*. <https://docs.expo.io/>.
- [25] *React Native Database*. <https://www.simform.com/react-native-database-selection-guide/>.
- [26] *Realm*. <https://redux.js.org/>.
- [27] *Realm Expo*. <https://expo.canny.io/feature-requests?search=realm>.
- [28] *Firebase*. <https://firebase.google.com/docs?hl=es>.
- [29] *Firebase Expo*. <https://docs.expo.io/guides/using-firebase/>.
- [30] *SQLite*. <https://www.sqlite.org/index.html>.

- [31] *SQLite Expo*. <https://docs.expo.io/versions/latest/sdk/sqlite/>.
- [32] *Http*. <https://tools.ietf.org/html/rfc7231>.
- [33] *Redux*. <https://realm.io/docs/javascript/latest/>.
- [34] *Async Storage*. <https://react-native-community.github.io/async-storage/>.
- [35] *Raspberry Pi Python*. <https://www.raspberrypi.org/documentation/usage/>.
- [36] A. Re B.E.; Re M.P.; Olmedo. «Análisis del impacto en la performance de una web app utilizando un lenguaje compilado y un lenguaje interpretado». En: *XXV Congreso Argentino de Ciencias de la Computación (CACIC)* (2019).
- [37] *Golang vs Python on complex problems*. <https://dzone.com/articles/golang-vs-python-which-one-to-choose>.
- [38] *Golang*. <https://golang.org/doc/faq>.
- [39] *Golang*. <https://talks.golang.org/2012/splash.article>.
- [40] Romero Dapozo R.; Fava L.A.; Vilches Antao D.; Díaz J.F. «Desarrollo de dispositivos electrónicos para entrenamiento deportivo». En: *Simposio Argentino de Sistemas Embebidos* (2019).
- [41] *rest*. [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm).
- [42] *Time Package*. <https://golang.org/pkg/time/>.
- [43] *Rand Package*. <https://golang.org/pkg/math/rand/>.
- [44] *Mux Package*. <https://github.com/gorilla/mux>.
- [45] *SSE Package*. <https://github.com/alexandrevicenzi/go-sse>.