

# **Sistema de comunicación, accesibilidad e inclusión para deportistas y entidades deportivas en la región**

**Autores:**

CARDOSO, Juan Pablo (juampicar97@gmail.com)

MARIÑO, Nahuel (nahuel\_es@hotmail.com)

**Director:**

HINOJAL, Hernán

**Co-Director:**

FINOCHIETTO, José Mariano

*Proyecto final para optar al grado de Ingeniero en Informática*

*Mar Del Plata, 21 de abril de 2023*



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-  
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

# **Sistema de comunicación, accesibilidad e inclusión para deportistas y entidades deportivas en la región**

**Autores:**

CARDOSO, Juan Pablo (juampicar97@gmail.com)

MARIÑO, Nahuel (nahuel\_es@hotmail.com)

**Director:**

HINOJAL, Hernán

**Co-Director:**

FINOCHIETTO, José Mariano

*Proyecto final para optar al grado de Ingeniero en Informática*

*Mar Del Plata, 21 de abril de 2023*

## Agradecimientos

*A nuestras familias por el acompañamiento incondicional durante el transcurso de nuestros estudios en la carrera de Ingeniería en Informática.*

*A todos nuestros compañeros de la carrera que nos acompañaron durante este trayecto, muchos de los que también resultaron en grandes amistades.*

*A la Facultad de Ingeniería y los docentes que la integran, por la excelencia académica, el buen trato con los estudiantes y la predisposición.*

*A nuestros directores por la propuesta inicial de formar este proyecto y la flexibilidad que nos dieron para definir parte de su funcionalidad. Valoramos su seguimiento, correcciones y recomendaciones a lo largo del proyecto.*

*Muchas gracias a todos.*

## Índice

<b>Agradecimientos</b>	<b>2</b>
<b>Resumen</b>	<b>8</b>
<b>Introducción</b>	<b>9</b>
<b>Problemática a resolver</b>	<b>10</b>
<b>Origen del proyecto</b>	<b>10</b>
Propuestas de valor	11
Objetivos	12
<b>Dominio del problema</b>	<b>13</b>
El deporte y la salud	13
Barreras para la participación del deporte	13
Accesibilidad a internet y dispositivos móviles	14
Mercado bilateral	16
Solución propuesta	16
<b>Gestión del proyecto</b>	<b>18</b>
Lean Development	18
Minimum Viable Product (MVP)	18
Ventajas	19
Desventajas	20
Referencia funcional	20
Metodología de trabajo	21
Análisis FODA	22
Fortalezas	22
Oportunidades	23
Debilidades	23
Amenazas	23
Estrategia en base al análisis FODA	24
<b>Análisis</b>	<b>25</b>
<b>Requerimientos</b>	<b>26</b>
RNF comunes de sistemas para clubes y deportistas	26
RNF de minijuegos del sistema para deportistas	27
RF del sistema para clubes	27
RF del sistema para deportistas	29
RF generales de minijuegos del sistema para deportistas	31
<b>Diseño de la solución</b>	<b>35</b>
Gamificación	35
Modelo Mechanics-Dynamics-Aesthetics (MDA)	35
Game design document (GDD)	36

Motor de videojuegos	36
Material Design	36
Backend as a Service (BaaS)	36
Test-driven development (TDD)	37
Dependency injection (DI)	37
Arquitectura del sistema	38
Base de datos	40
<b>Implementación de la solución</b>	<b>41</b>
Elección de tecnologías	41
Framework para el desarrollo de las aplicaciones	41
Flutter	41
React Native	42
Android nativo con Kotlin	42
Motor de videojuegos	43
Unity Engine	43
Flame Engine	44
Ventajas	45
Desventajas	45
Backend y base de datos	46
Firebase	46
Comparativa Cloud Firestore / Realtime Database	47
Otras alternativas a Firebase no analizadas	48
Elección	48
Elección de sistema para el versionado	49
Elección de servicio de mapping	49
<b>Diseño de interfaces de usuario</b>	<b>50</b>
Interfaces de aplicación para clubes y deportistas	50
Prototipado	52
Interfaces del minijuego jueguitos	52
<b>Producto</b>	<b>54</b>
Disponibilidad	54
Mapa de clubes	54
Gestión del club	55
Métricas	56
Jerarquía, clubes y roles	56
Dynamic link para abrir minijuego	57
Utilización de estrategias de gamificación	57
Motivar asistencia a prácticas deportivas	58
Trabajo con artista	58
Seguridad y protección de la información	60

Costos del producto	61
Elaboración	61
Despliegue	61
Total	61
TDD y DI	61
Testing	63
Características de innovación	63
GitHub Copilot	64
Temas dinámicos y colores personalizables	65
Diseño inteligente	66
Importes rápidos	67
Mejora de procesos	68
<b>Memoria del proyecto</b>	<b>70</b>
Planificación y estimación	70
Tiempos reales	71
Comparación con estimación	72
Motivos de la desviación	72
<b>Resultado / Entregables</b>	<b>72</b>
Retrospectiva de objetivos del proyecto	73
Desafíos principales	74
Formalización de requerimientos	74
Limitar el producto de software a un MVP	75
Contribuir a solucionar un problema social desde el punto de vista de la tecnología	76
El riesgo de la dependencia en un proyecto interdisciplinario	76
Aprendizaje	76
Retrospectiva: Decisiones	77
Utilizar BaaS a fin de reducir el tiempo de desarrollo	77
Estimación optimista	77
Independizar al proyecto de los proyectos asociados	77
Dedicar mucho tiempo a documentación formal de análisis y diseño	78
Selección de tecnologías	78
Tercerización de la tarea del arte	79
<b>Trabajos a futuro</b>	<b>80</b>
Evolución del proyecto	80
Funcionalidad a implementar en futuras iteraciones	81
<b>Bibliografía</b>	<b>87</b>
<b>Anexo I</b>	<b>90</b>
<b>Glosario</b>	<b>90</b>
Introducción	91

<b>Anexo II</b>	<b>96</b>
Modelo MDA	96
Título: Jueguitos	97
Género: Arcade	97
Mecánicas	97
Dinámicas	97
Estéticas	98
Historia	98
Narrativa	99
<b>Anexo III</b>	<b>100</b>
GDD de 1 hoja	100
Título del juego	101
Sistemas de juego previstos	101
Edad de los jugadores a los cuales está destinado	101
Calificación Legal Prevista	101
Un resumen de la historia del juego	101
Modos de juego	101
Aspectos destacables del juego que atraigan ventas (USP)	101
Productos que compiten con el propuesto	102
<b>Anexo IV</b>	<b>103</b>
GDD de 10 hojas	103
Página 1	104
Sistemas de juego previstos	104
Edad de los jugadores a los cuales está destinado	104
Calificación Legal Prevista	104
Fecha proyectada de lanzamiento	104
Página 2	105
Resumen de la historia del juego	105
Flujo de juego	105
Página 3	106
Descripción personajes	106
Página 4	107
Géneros del juego	107
Interacción del jugador con el juego	107
Descripción de los eventos inmersivos más importantes	107
Página 5	109
Descripción del mundo o ambiente en que transcurre el juego	109
Página 6	110
Descripción concisa de la Inmersión del juego	110
Página 7	111

Descripción de las mecánicas usadas	111
Página 8	113
Descripción enemigos	113
Página 9	114
Descripción películas/escenas	114
Página 10	115
Material extra o atajos que motivan al jugador para volver a jugar el juego	115
<b>Anexo V</b>	<b>116</b>
Requerimientos de minijuego “Jueguitos”	116

## Resumen

El presente Proyecto Final fue desarrollado por Juan Pablo Cardoso y Nahuel Mariño, estudiantes de la carrera Ingeniería en Informática dictada en la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Este proyecto involucra la elaboración de un sistema para la comunicación, accesibilidad e inclusión para deportistas y entidades deportivas. Su objetivo principal es conectar entidades deportivas con practicantes del deporte. Para ello se desea ofrecer un método digital de búsqueda de clubes, utilizando un mapa interactivo que presente filtros de distintos criterios para ajustarse a las necesidades y preferencias de los deportistas; y en paralelo brindar funcionalidades de gestión que sean de utilidad para los clubes y entidades deportivas.

Además se pretende utilizar técnicas de gamificación en videojuegos que sirvan como incentivo para que se utilice el sistema y aumente la participación de los deportistas en las actividades deportivas.

La solución informática realizada da lugar a tres aplicaciones móviles, una para clubes, una para deportistas y un minijuego para deportistas, que son el producto de software resultado de este proyecto.

## Introducción

Este proyecto busca desarrollar un sistema informático que actúe como eje de comunicación entre entidades deportivas y deportistas en la región, permitiendo a los deportistas encontrar clubes que se adapten a sus intereses y necesidades mediante un mapa interactivo, brindar a las entidades deportivas soluciones administrativas básicas, y favorecer la inclusión social de actores vulnerables en la práctica del deporte.

El deporte se concibe aquí como un conjunto de prácticas de salud pública. Así, la actividad deportiva no sólo puede promover la adopción de estilos de vida "saludables", sino que también la autonomía y la libertad del individuo en el marco de densas redes de sociabilidad.\*

Se busca fomentar la participación de actores que usualmente son excluidos en la práctica de deportes, o que estadísticamente tienen menores índices de participación, brindando un sistema que aumente la accesibilidad al deporte, utilizar técnicas de gamificación en videojuegos para motivar y retener usuarios; y promover la participación asidua en el deporte.

A fin de aumentar el atractivo de la aplicación para los clubes, se ofrecen como incentivo tanto la posibilidad de tener exposición al público general mediante el sistema, como algunas funcionalidades propias de la gestión de las entidades, que generalmente se presentan como problemáticas principales en clubes de escasos recursos.

---

\* Gil, G. J. (2021) *Democracia corporal. El fútbol como medio para la inclusión y el mejoramiento de la calidad de vida de niños y adolescentes vulnerables, de mujeres, y de personas con discapacidad en el Sudeste Bonaerense*. [Grupo "Estudios antropológicos", Facultad de Ciencias de la Salud y Trabajo Social de la UNMDP].

## Problemática a resolver

En este proyecto se busca facilitar la comunicación y el acceso de los deportistas a los clubes, e impulsar la inclusión. Los gustos e intereses de las personas son un amplio espectro, que en ocasiones puede llegar a incluir necesidades de carácter excluyente, por ejemplo, las prácticas de deporte para personas con discapacidad. Es por eso que surge la necesidad de una solución tecnológica que permita encontrar instituciones deportivas que se adapten a dichas necesidades con facilidad sin tener que depender del “boca en boca” o de herramientas tecnológicas que no fueron diseñadas específicamente para ese fin.

Por otro lado, las entidades deportivas barriales muchas veces no suelen contar con software para la administración de las tareas básicas que ellos realizan y, en consecuencia, recurren a circuitos, procedimientos u operatoria tanto en papel como en otros medios no digitales; reduciendo su eficiencia y aumentando la probabilidad de demoras y errores.

También existen clubes que directamente realizan sus prácticas deportivas en espacios públicos, pues carecen de instalaciones, por lo cual la idea de utilizar un software de gestión tradicional para computadora ni siquiera es una posibilidad.

## Origen del proyecto

El presente nace de las necesidades de dos proyectos distintos, uno de extensión y otro de investigación, ambos existentes en el ámbito de la UNMdP. El primero de ellos, tenía como objetivo realizar una app móvil para que los clubes barriales puedan realizar ciertas tareas básicas de gestión de sus clubes. El segundo, un amplio proyecto interdisciplinario con múltiples facultades involucradas cuyo fin principal era colaborar con la inclusión en el fútbol, en el que la facultad de ingeniería tendría participación de carácter técnico, particularmente en la creación de un mapa virtual interactivo.

Un docente de la universidad (quien luego se convertiría en director del presente trabajo final) le comentó a los autores sobre la existencia de estos proyectos y la posibilidad de idear una solución integral. La idea de combinar ambos proyectos permitía una inmensidad de posibilidades, porque al tener una aplicación principalmente

destinada a clubes (la de gestión de clubes) y una principalmente destinada a deportistas (la del mapa) era posible crear un ecosistema bilateral, mucho más grande y con amplias posibilidades de interacción.

Este proyecto era altamente compatible con los intereses de los autores. Uno de ellos, quien estaba enfocando su perfil profesional hacia los videojuegos, podría implementar técnicas de gamificación y desarrollar videojuegos en la aplicación para deportistas. El otro, quien demostraba interés por incursionar en el desarrollo móvil, podría participar en el desarrollo de dos de ellas. Además, ambos tenían entusiasmo por participar de un proyecto de interés social, aspecto característico del proyecto.

## Propuestas de valor

El proyecto incluye las siguientes propuestas de valor:

- **Novedad**, ya que además de la funcionalidad referente a los proyectos de origen, se proponen las siguientes características:
  - El uso de técnicas de gamificación en videojuegos que fomenten la práctica del deporte.
  - El uso de métricas para permitir el análisis y evaluación del rendimiento del club y su progreso en el tiempo.
- **Simplificar el acceso** de las personas a los clubes deportivos, brindando un sistema desarrollado específicamente para este fin.
- **Facilitar el trabajo** de los clubes, porque permite digitalizar y automatizar algunas tareas de gestión del club como asistencias, registro de pagos, inscripciones, etc.
- **Precio**, debido a que se trata de un sistema **completamente gratuito** tanto para **deportistas** como para **clubes**.
- **Reducción de costos** para los clubes, por dos motivos principales.
  - Se disminuirán los costos en horas de trabajo a nivel administrativo.

- No requiere del uso de computadoras u otros dispositivos caros, que especialmente podrían no ser accesibles por algunos clubes que no cuenten con el presupuesto necesario.
- **Usabilidad**, al hacer más sencillo encontrar clubes en una zona determinada, sin tener que recurrir al “boca a boca” o herramientas que, al no estar diseñadas ad hoc, podrían no tener la misma eficiencia de resultados (como Google Maps).

## Objetivos

El objetivo principal del Proyecto es el desarrollo y provisión de un sistema informático que permita:

- Modernizar y mejorar los circuitos administrativos para que los deportistas de la región encuentren entidades deportivas que se adapten a sus intereses y necesidades.
- Brindar herramientas de gestión administrativa para los clubes.
- Contribuir a la inclusión de personas alejadas de la práctica del deporte (Incluyendo niños y adolescentes en situación de vulnerabilidad, mujeres, personas con discapacidad, entre otros).
- Fomentar la participación de los deportistas en las prácticas deportivas utilizando gamificación.
- Incluir métricas para el análisis del estado actual y la evolución de indicadores cuantificables para las entidades deportivas.

## Dominio del problema

En este capítulo se explican algunos tópicos importantes para mejorar el entendimiento del problema y arribar a la solución planteada.

### El deporte y la salud

De acuerdo con el comité olímpico internacional, la práctica del deporte es un derecho humano y toda persona debe tener la posibilidad de practicar deporte sin discriminación de ningún tipo.

El deporte no sólo puede promover la adopción de estilos de vida saludables, sino también la autonomía y la libertad del individuo en el marco de densas redes de sociabilidad. En efecto, el libre desplazamiento y la interacción con los otros son dos factores constitutivos de la vida democrática. En este sentido, la práctica del deporte involucra tanto la autodeterminación corporal como una red de relaciones entre individuos y grupos.<sup>12</sup>

### Barreras para la participación del deporte

Considerando que los objetivos del proyecto involucran contribuir a fomentar la participación de las personas en el deporte, e incluir a personas alejadas de la práctica, para comprender cómo colaborar en la inclusión antes es necesario repasar cuáles son las principales barreras para la participación en deportes. Estudios han mostrado que los factores que influyen dichas prácticas pueden ser agrupados en las siguientes categorías:

- Factores demográficos/biológicos
- Factores psicológicos, cognitivos y emocionales
- Atributos y habilidades de la persona
- Factores socioculturales
- Factores físicos del entorno

Los tres primeros pueden también ser englobados como factores personales. Comúnmente es una combinación de los factores personales, sociales y del entorno; lo que determina el comportamiento de la actividad física de las personas.

Las principales **barreras personales** están relacionadas a falta de confianza y/o de habilidad en las competencias principales del deporte, falta de energía o motivación, y sentirse expuesto al estar fuera de forma física. Las **barreras sociales** están relacionadas a falta de tiempo, prioridades, compromisos familiares y falta de compañeros para el ejercicio. Dentro de los aspectos relacionados al **entorno físico**, las barreras principales están relacionadas a costos, información, accesibilidad, oportunidades, características estéticas y seguridad.

El aporte que se puede hacer desde el software para colaborar a mitigar estas barreras es hasta cierto punto limitado, sobre todo en los aspectos personales y sociales, pero de todas formas existen posibilidades considerables. Se prevé que el sistema pueda colaborar parcialmente en reducir varias barreras personales, sociales y del entorno físico. Siendo algunos ejemplos:

- Facilitar, al alcance de la mano y a una simple búsqueda de distancia, la tarea de buscar un club cuyos horarios coincidan con los intereses, gustos y necesidades de la persona. (Este aspecto es uno de los principales ejes sobre los que se pretende actuar. Permitirá colaborar directa e indirectamente a derribar múltiples barreras del entorno físico, principalmente de información, accesibilidad, entre otras).
- Entregar recompensas virtuales que premien a las personas por su constancia, a fin de aumentar la motivación y fomentar la participación en la práctica del deporte.<sup>3</sup>

## Accesibilidad a internet y dispositivos móviles

El Instituto Nacional de Estadísticas y Censos publica con distintos niveles de periodicidad anual e interanual, informes técnicos con estadísticas sobre acceso a internet, y a las tecnologías de la información y comunicación. Según datos del INDEC, en el cuarto trimestre de 2021, se registró que:

- El **64,2%** de los hogares urbanos tiene acceso a **computadora**.
- El **90,4%** de los hogares urbanos tiene acceso a **internet**.
- El **88,1%** de las personas utilizan **teléfono celular**.
- El **87,2%** de las personas utilizan **internet**.

## Figura 1

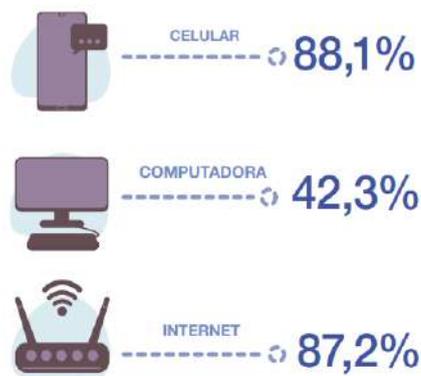
Hogares con acceso a computadora e internet. Total 31 aglomerados urbanos. Datos correspondientes a cuarto trimestre de 2021



Nota. Si bien algunos datos se encuentran reportados únicamente a nivel nacional, en la ciudad de Mar Del Plata, los usos informados de computadora e internet se ubican en 63,8% y 87,7% respectivamente.

## Figura 2

Población de 4 años y más, por utilización de bienes y servicios de las TIC. Total 31 aglomerados urbanos. Cuarto trimestre de 2021.



Nota. Tener en cuenta que a diferencia de la figura 1, esta figura indica "uso en los últimos tres meses".

Si se comparan estos datos con respecto al informe correspondiente al cuarto trimestre de 2018 se pueden observar las siguientes variaciones:

- Uso de computadora **-4,3%**
- Uso de teléfono celular **+9,2%**

Del análisis de los anteriores datos es posible concluir que los teléfonos móviles no sólo son el dispositivo con mayor presente en cuestión de números brutos de accesibilidad, sino que la tendencia indica que lo seguirán siendo en el futuro cercano. En conclusión, **el teléfono móvil es la elección predilecta si se tiene como objetivo priorizar el alcance en términos inclusivos.**<sup>4 5</sup>

## Mercado bilateral

Un mercado bilateral es un tipo de mercado en que una plataforma interactúa con dos grupos de agentes que se benefician mutuamente de la existencia de dicha plataforma. Algunos ejemplos son Visa (siendo los agentes las entidades bancarias, y los usuarios); o Uber (siendo los agentes conductores y usuarios).

Una característica de los mercados bilaterales, que afecta particularmente a su capacidad de inserción, es que deben ser utilizados por dos grupos de consumidores que generalmente son distintos. Así mismo el hecho de que al entrar en un nuevo mercado, ambas partes pudieran enfrentarse a un bajo incentivo para ingresar en la plataforma porque no posee usuarios “del otro tipo”. Esto genera una sinergia negativa que causa que la inserción en el mercado sea una tarea más difícil de lo normal. A este ciclo de retroalimentación negativa donde no hay usuarios porque no hay prestadores y viceversa, se la conoce como el problema del huevo y la gallina.

El **problema del huevo y la gallina**, por su nombre, podría parecer un fenómeno poco serio. Sin embargo es un suceso analizado, considerado y estudiado. Además a lo largo de la historia ha sido un factor importante y muchas veces resultó el factor definitivo para el fracaso de algunas compañías. Algunos ejemplos:

- El fracaso de Uber para insertarse en China. (Cometieron considerables errores al adaptarse a las necesidades de la población del país asiático)
- La desaparición de blackberry del mercado de los smartphones (Por la falta de incentivos para atraer desarrolladores de aplicaciones).<sup>6</sup>

## Solución propuesta

Iniciando la década del 2010, la industria de los smartphones aún se encontraba en crecimiento y, particularmente en Argentina, el porcentaje de la población que tenía

uno aún era relativamente bajo. Sin embargo, durante la misma década el smartphone pasaría a ser paulatinamente el principal dispositivo de comunicación, llegando incluso a superar a la computadora de escritorio. Actualmente, el porcentaje de usuarios de PC en argentina ronda el 64% mientras que el acceso a teléfono celular asciende a 88%.<sup>7</sup> Es por ello que la elección de los smartphones como plataforma para el sistema no sólo es una alternativa práctica por la portabilidad que ofrece, sino la opción más accesible al alcanzar un porcentaje mucho mayor de la población.

Teniendo en cuenta este contexto, la identificación de necesidades tanto de clubes como deportistas conduce a un tipo de solución que incluya dos sistemas, uno para **deportistas** y uno para **clubes** (o entidades deportivas en forma general). Ambas independientes entre sí pero con cierto grado de interacción.

Adicionalmente, se plantea la inclusión del desarrollo de un videojuego utilizando técnicas de gamificación dentro de la aplicación para deportistas con el objetivo principal de fomentar la participación en las prácticas deportivas. La descripción funcional del sistema será detallada completamente en un capítulo posterior.

## Gestión del proyecto

### Lean Development

La filosofía Lean se basa en Lean Manufacturing. Esta fue desarrollada en la década del 80 por la compañía japonesa de automotores Toyota. El sistema de producción ajustada considera como innecesario a todo gasto de recursos que vaya dedicado a un objetivo que no sea la creación de valor para el consumidor final, y por tanto un objetivo a ser eliminado. La hipótesis central de la metodología Lean Startup es que si las compañías startups invierten su tiempo en productos o servicios de construcción **iterativa** para satisfacer las necesidades de los primeros clientes, pueden reducir los riesgos de mercado y evitar la necesidad de grandes cantidades de financiación inicial o grandes gastos para lanzar un producto.

De esta forma se busca con los lanzamientos de porciones de producto la validación de hipótesis sobre la necesidad del cliente de consumirlo, mediante ciclos de feedback. Por lo tanto se obtiene un aprendizaje del cliente con la utilización del producto y permite una rápida adaptación.<sup>8</sup>

### Minimum Viable Product (MVP)

Inicialmente el alcance debía ser ampliado funcionalmente, pues lo estrictamente necesario era sólo una aplicación que actúe como mapa digital, y una aplicación que permita gestionar clubes. La necesidad de expandir el alcance abriría las puertas a que los autores puedan realizar contribuciones creativas e innovadoras a la solución. Eventualmente, esta libertad para aportar ideas creativas trajo un efecto secundario: al momento de analizar la viabilidad y realizar las primeras estimaciones de tiempos, se notó que el proyecto no sólo había alcanzado el alcance deseable, sino que lo había excedido y ahora era necesario realizar recortes funcionales.

Esta situación fue inicialmente vista con cierta conmiseración por los autores, quienes no solamente tenían altas expectativas, sino que confiaban en el alto potencial de la solución en su totalidad. Sin embargo, se encontró una manera de recuperar la viabilidad del proyecto sin resignarse a recortar totalmente su alcance: Se optó por utilizar los principios de la metodología lean startup.

El enfoque ahora estaría en el **diseño de la solución completa**, pero de el **desarrollo únicamente de un producto mínimo viable** (MVP) el cual tendría una cantidad de funcionalidad acotada, y sería la primera iteración de desarrollo, que como un MVP permitirá realizar **testing de aceptación de usuario**, reunir feedback y permitir avanzar sobre futuras iteraciones para refinar el producto.

El MVP (mínimo producto viable) es un concepto fundamental de la metodología Lean Startup. Este es un producto con suficientes características para satisfacer a los clientes iniciales que, al ser un producto entregable, permite recibir sus comentarios como **feedback** para el desarrollo o mejora de funcionalidades en el futuro.

La aplicación de la filosofía Lean Development involucra la creación de un MVP, lo cual involucra las siguientes ventajas y desventajas:

### Ventajas

- **Reducción de tiempos:** No sólo se disminuye el tiempo de desarrollo al acotar la funcionalidad a un MVP, sino que se disminuye el tiempo que toma recibir feedback de los usuarios por primera vez y como resultado se puede refinar el producto desde etapas muy tempranas.
- **Reducción de costos:** Al disminuir los tiempos de horas de trabajo, se disminuyen los costos asociados.
- **Viabilidad del producto:** Permite conocer de manera temprana y realista la viabilidad del producto, ya que el MVP es lanzado rápidamente al mercado y se puede analizar su viabilidad sobre condiciones de mercado y usuarios reales.
- **Entendimiento de mercado:** El MVP también permite ayudar a comprender el mercado, analizar el comportamiento de los usuarios y mejorar el entendimiento de las necesidades de los usuarios.
- **Flexibilidad para adaptarse a los cambios:** Al mejorar el entendimiento del mercado es posible que cambie la funcionalidad esperada del sistema. Al trabajar con un MVP, los cambios se deben realizar sobre un producto más pequeño y maleable.

## Desventajas

- **Es necesario definir correctamente al MVP:** El proceso de desarrollar un MVP implica ser selectivo, dirigido y estratégico con la funcionalidad a incluir. Si es demasiado pequeño podría no representar correctamente al producto y uno muy grande podría perder el propósito de desarrollar un MVP.
- **Diseño imperfecto:** Invertir una gran cantidad de tiempo en el diseño del software no es compatible con el concepto de entrega rápida de los MVP. Es por eso que lo más razonable es enfocarse en la funcionalidad, usabilidad y simplicidad en lugar de en un diseño perfecto pero complejo.
- **Se debe elegir correctamente la arquitectura y las tecnologías:** Un error común al crear un MVP es la elección de tecnologías y arquitecturas que no permiten que el proyecto crezca en el futuro. Es necesario comprender que un MVP puede estar destinado a escalar y es por ello que es imprescindible tomar decisiones de diseño correctas para tener un producto flexible, adaptable y escalable.<sup>9</sup>

## Referencia funcional

En la app de deportistas, la característica emblemática, el **mapa interactivo**, fue una definición funcional especificada en el **proyecto goles de inclusión**. En un momento, existía la posibilidad de realizar trabajo interdisciplinario con expertos en antropología y salud, para realizar aportes concretos a la inclusión con fundamentos teóricos. Sin embargo, por diversos motivos, no se pudo realizar dicha colaboración. Esta situación adversa fue desmotivante para los autores del informe porque comprometería su objetivo de promover la inclusión en el deporte. Por lo tanto se generó un debate sobre cómo debería continuar el proyecto o si directamente lo debían descartar y cambiar a otro. Otra opción sería continuar con los requerimientos del proyecto de gestión de clubes.

Continuar con las especificaciones del proyecto de clubes era la salida más fácil porque implicaría un menor esfuerzo intelectual por parte de los autores y sería la opción más rápida para que puedan graduarse. Por otro lado se analizó que se perdería parte del trabajo realizado en su proyecto final hasta ese momento y deberían dejar de lado parte de los objetivos sociales que caracterizaban el proyecto actual. Por lo tanto se

decidió tomar el riesgo de continuarlo con características adicionales y seguir aportando a la inclusión desde el mapa de clubes que estaba especificado en la documentación del proyecto goles de inclusión.

Es importante aclarar que el mapa sólo se había planteado como un **concepto** y ya se esperaba que el equipo que trabaje sobre él realice la ingeniería de la solución. Además, la situación problemática generada fue percibida como una oportunidad para modificar el alcance del proyecto. De esta forma se decidió agregar propuestas de innovación tales como el uso de técnicas de gamificación en videojuegos (para fomentar la asistencia a las prácticas deportivas), la personalización de los perfiles de deportistas, entre otros.

En la app de clubes, los requerimientos funcionales se obtuvieron a partir de entrevistas con un club deportivo puntual, quien inicialmente tenía la necesidad de una app para la gestión de clubes que dio origen al proyecto asociado. Además, el equipo pudo realizar una reunión con un directivo de la liga marplatense de fútbol en conjunto con algunos representantes de diversos clubes de la liga, con los cuales se pudo realizar un intercambio de ideas para guiar el proyecto.

Al definir la funcionalidad del sistema para clubes se tuvo en consideración el rol de **gestor de verificaciones**, una entidad superior que se encargue de decidir cuáles clubes son válidos y deberían ser mostrados en el mapa de clubes. Esto permitiría dar mayores garantías a los usuarios de la aplicación de deportistas y generaría un mayor grado de confianza en la aplicación. Sin embargo, por el tiempo extra que supondría generar un nuevo sistema para ese fin fue decidido dejar su implementación fuera del alcance de este proyecto.

## Metodología de trabajo

Como **metodología de desarrollo**, se decidió utilizar la metodología ágil **Kanban** bajo los principios de **Lean Development**. Además fue seleccionada la plataforma **Trello** como tablero virtual.

Los principios de Lean Development permiten **reducir la incertidumbre** y las tareas que no generan valor para los usuarios finales. Por lo tanto se considera aplicarlo para el proyecto ya que hay un cierto grado de desconocimiento en la respuesta que

pueden tener los mercados involucrados (clubes y deportistas), esto debido a un cierto grado de incertidumbre con respecto a sus necesidades, sumada ya al grado inherente de incertidumbre que tiene cualquier desarrollo de software.

Dentro del alcance de este proyecto se desarrollará un MVP, que será la primera iteración de un desarrollo lean. Esto permitirá reducir el riesgo y presentará la posibilidad de obtener feedback de usuarios potenciales antes de que el desarrollo continúe.

La metodología Kanban se eligió por ser una metodología ágil que el equipo ya conocía, la posibilidad de trazar un seguimiento del proceso de desarrollo, organizar las tareas de manera sencilla según su estado (por hacer, haciendo y finalizadas) y por generar una comunicación continua entre los integrantes del equipo. Otros aspectos ponderantes son su sencillez de implementar y utilizar, sobre todo para su uso en un equipo de trabajo compuesto por dos personas; y que es la metodología más utilizada para aplicar Lean Development.

## Análisis FODA

Se realizó un análisis FODA con el objetivo de conocer la situación del proyecto y poder planear una estrategia a seguir durante el desarrollo del mismo. A continuación se listan las fortalezas, oportunidades, debilidades y amenazas que fueron identificadas.

### Fortalezas

- Es un proyecto de interés social.
- Se dispone de un equipo de ingeniería que:
  - Ya ha trabajado en conjunto, tanto en proyectos universitarios como extracurriculares (tiene un alto grado de cohesión).
  - Tiene cierta experiencia profesional (ha adquirido algunos conocimientos prácticos).
  - Ha publicado aplicaciones en la tienda de aplicaciones de Android (conocimiento de algunos aspectos referentes a servicios de Google).
- No se deberá pagar por los productos de software generados, por lo que tendrán mayor facilidad para insertarse en el mercado.

- Libertades para agregar valor a la solución desde la ingeniería, con margen para la innovación y soluciones creativas.
- Experiencia con el uso del motor de videojuegos elegido.

## Oportunidades

- Masificación de los dispositivos móviles y del acceso a internet en ellos, incluso en los sectores de población más vulnerables.
- En Argentina, el sistema operativo Android tiene una amplia mayoría de cuota de mercado.
- No opera en la región otro sistema que cumpla con la misma funcionalidad que el propuesto.

## Debilidades

- El equipo no posee miembros que tengan altos conocimientos sobre diseño de experiencia de usuario (UX) o de interfaces de usuario (UI).
- No hay una entidad demandante centralizada que defina un conjunto de requerimientos puntuales para el proyecto, lo cual implica una tarea y un conjunto de responsabilidades extra sobre el equipo.
- No tener una entidad demandante centralizada genera el riesgo de hacer un trabajo que no satisfaga las necesidades de los usuarios finales.
- El equipo no posee miembros que tengan conocimiento sobre áreas artísticas vinculadas al desarrollo de videojuegos.

## Amenazas

- Dependiendo del avance de los grupos de investigación de otras universidades podría causar retrasos en los tiempos del proyecto.
- La situación de la pandemia provoca disminuciones en la actividad de los clubes deportivos.

- Los clubes podrían encontrar dificultades para adaptarse a la herramienta.
- Al tratarse de dos aplicaciones que se insertarán en el mercado de manera bilateral, existe la posibilidad de enfrentarse al “problema del huevo y la gallina” (mencionado en el marco teórico).

## Estrategia en base al análisis FODA

El análisis FODA realizado permite tener una visión general de la situación actual del proyecto. Esto permite trazar un plan estratégico a fin de potenciar y aprovechar los aspectos positivos, e intentar mitigar o disminuir los efectos de los negativos. Algunas de las estrategias tomadas a partir del análisis realizado son:

- A fin de aprovechar al máximo la popularidad de la plataforma Android en dispositivos móviles en Argentina, y teniendo en cuenta la fragmentación de Android SDK, se buscará ser compatible con la mayor cantidad de versiones de Android posible (favoreciendo a la inclusión de personas que podrían tener dispositivos menos modernos).
- Las debilidades relacionadas a falta de experiencia o conocimiento, como algunas de las tecnologías o aspectos relacionados a UX/UI, serán mitigadas dedicando parte del tiempo del proyecto a investigar sobre tecnologías, metodologías y disciplinas de manera tal que se disminuya su efecto.
- Se optará por tercerizar la tarea de realizar el arte principal del videojuego. La decisión de delegar esta tarea fue tomada porque no involucraba habilidades en las que el equipo buscara fortalecerse.
- El hecho de no tener una entidad que realice demandas puntuales es una debilidad porque no marca un camino específico a seguir con respecto a la funcionalidad. Sin embargo, la libertad de elegir puede ser convertido en una oportunidad para aportar creatividad e innovación a la solución final.
- Para evitar el “problema del huevo y la gallina” se utilizará como estrategia enfocarse en un mercado primero, e intentar que este no dependa del otro. Para

cumplir con ese objetivo se ofrecerá una solución a los clubes que les sea útil independientemente del volumen de usuarios de la aplicación para deportistas.

- Enfrentarse a un bloqueo por espera del avance de otros grupos involucrados en el proyecto de origen activo es un riesgo con moderada probabilidad de ocurrencia y un impacto potencial muy alto. Por lo tanto el equipo planteará todo el proyecto de forma alineada pero independiente.
- Se estima que es probable que para cuando el proyecto esté listo la situación pandémica sea más amena y las actividades de los clubes deportivos se hayan retomado.

## Análisis

### Requerimientos

Los requerimientos de este proyecto surgen principalmente de las necesidades expuestas en la sección de “Problemática a resolver”. Además, sobre el inicio del proyecto se realizó una serie de entrevistas con expertos a fin de elicitar y documentar algunos procesos relacionados a la gestión de clubes. Estos requerimientos fueron plasmados como el primer prototipo de los requerimientos funcionales. Debido a la naturaleza bipartita del proyecto, se clasificaron por pertenencia a las aplicaciones de **clubes o deportistas**.

Uno de los objetivos del proyecto es utilizar técnicas gamificación para potenciar la retención de los usuarios mediante videojuegos integrados en la aplicación de deportistas y mejorar su participación en el deporte. La documentación correspondiente a esos videojuegos los referencia como **minijuegos**. Sus requerimientos surgieron a partir de la creación de documentos de MDA, GDD de 1 hoja y GDD de 10 hojas. La documentación referida al videojuego “Jueguitos” fue incluida en los anexos II, III, IV y V. También se documentaron otros dos minijuegos, “Saltar la soga” y “Lanzamientos”, pero fueron agregados como un anexo externo al presente informe por no formar parte del alcance del proyecto.

Una vez que se finalizó la definición de los requerimientos, fue necesario realizar una selección de los requerimientos que formarían parte del MVP. La definición del MVP fue bajo un criterio mixto entre priorización de funcionalidades core y de funcionalidades de alto impacto que requieran relativamente bajo esfuerzo (effort impact matrix).

A continuación se listan todos los requerimientos funcionales (RF) y no funcionales (RNF) incluidos en el MVP. La lista completa con todos los requerimientos del sistema puede encontrarse como un anexo externo al presente informe.

### RNF comunes de sistemas para clubes y deportistas

1. El sistema debe ser desarrollado como una aplicación para dispositivos móviles Android.

2. El sistema debe tener orientación vertical.
3. Las contraseñas de las cuentas deben almacenarse hasheadas.
4. El sistema debe contar con encriptación extremo a extremo para el envío de información sensible y/o datos de autenticación.

## RNF de minijuegos del sistema para deportistas

1. Los minijuegos deben poder jugarse en orientación vertical.
2. Los minijuegos deben controlarse con toques simples en la pantalla sin necesidad de activar ningún gesto complejo.
3. Los minijuegos deben tener temática deportiva, principalmente sobre los deportes incluidos en la aplicación, u otros de similar popularidad.
4. Los minijuegos deben ser compatibles con dispositivos que soporten librería gráfica OpenGL 2.0 o mayor.

## RF del sistema para clubes

1. El sistema debe permitir a un club crear una cuenta de club que le permitirá acceder a la aplicación y toda su funcionalidad. El registro debe ser utilizando un e-mail y contraseña. Un usuario no registrado no puede acceder a ninguna funcionalidad de la aplicación, salvo la pantalla de registro.
  - a. El usuario debe poder iniciar y cerrar sesión con la cuenta creada.
2. Todo club registrado debe poder visualizar su perfil, que contendrá la información relacionada al club.
3. El sistema debe permitir a un club registrar los siguientes datos en su perfil de club:

Datos requeridos:

- a. **Nombre completo:** El nombre completo del club.
- b. **Nombre breve:** Versión corta del nombre.

- c. **Ubicación:** Deberá abrir un mapa donde el club pueda elegir el punto exacto en el mapa.

Datos opcionales:

- d. **Email:** Email público del club.
  - e. **Teléfono:** Número de contacto para llamadas.
  - f. **WhatsApp:** Número de contacto por WhatsApp.
  - g. **Facebook:** Perfil de Facebook.
  - h. **Twitter:** Perfil de Twitter.
  - i. **Instagram:** Perfil de Instagram.
  - j. **Otros medios de contacto:** Deberá ser un campo de texto lo suficientemente largo como para permitir incluir otras redes sociales.
4. El sistema debe permitir a un club insertar modificar y eliminar elementos en una lista de **deportes practicados**, donde podrán dar de alta todos los deportes que se practican en la institución.
    - a. Un **deporte practicado** al ser creado deberá ser de un **tipo de deporte**, elegible de una lista de deportes tradicionales y estandarizados (Fútbol, básquet, tenis, atletismo, natación, otros, etc), permitiendo así que los deportes sean encontrados por los usuarios bajo el filtro “Deporte→Fútbol”.
    - b. Un deporte debe tener los siguientes datos:
      - i. **Tipo de deporte:** “Fútbol”, “Vóley”, “Básquet”... Debe ser una cantidad muy abarcativa de deportes. La opción otros deberá estar disponible como un tipo de deporte.
      - ii. **Se practica en ambiente techado:** Si/No.
      - iii. **Práctica femenina:** Si/No.
      - iv. **Personas con discapacidad:** Si/No.
  5. El sistema debe permitir a un club insertar modificar y eliminar deportistas en una lista de deportistas, donde podrán dar de alta a todos los deportistas que estarán asociados al club.

Datos a almacenar sobre los deportistas:

- i. DNI.
  - ii. Apellidos.
  - iii. Nombres.
  - iv. Fecha de nacimiento.
6. El sistema debe permitir el alta manual de deportistas. En esta modalidad se deben ingresar manualmente los datos del jugador en un formulario.
  7. El sistema debe permitir el alta rápida de deportistas. De esta forma el sistema debe generar un código QR en el teléfono del deportista interesado, luego la persona del club escanea el código y se rellena automáticamente su formulario.
  8. El sistema debe tener una pantalla de **métricas**, donde se pueda visualizar el estado y evolución de ciertos datos cuantificables.
    - a. Se solicita que la pantalla de métricas muestre estadísticas de los siguientes datos cuantificables:
      - i. Número de inscripciones al club
      - ii. Porcentajes del total de deportistas por deporte
      - iii. Porcentajes del total de deportistas por rango etario
  9. El sistema debe permitir registrar los pagos (sin transacciones de dinero por medio del sistema) que realizan los miembros del club.
    - a. El sistema debe mostrar un historial filtrable de los pagos.
  10. El sistema debe permitir tomar asistencia. Al finalizar esa toma, se debe registrar fecha, hora y quién la tomó.

## RF del sistema para deportistas

1. La aplicación debe permitir a una persona crear una cuenta en la aplicación. El registro debe ser utilizando un e-mail y contraseña. Un usuario no registrado no puede acceder a ninguna funcionalidad de la aplicación.

- a. El usuario debe poder iniciar y cerrar sesión con la cuenta creada.
2. La aplicación debe incluir un **mapa de clubes** donde un usuario registrado pueda visualizar de manera interactiva los clubes que tiene a su disposición.
    - a. La aplicación debe permitir a un usuario acceder al mapa de clubes, donde el usuario podrá navegar naturalmente.
    - b. La aplicación debe permitir que un usuario pueda acceder a una **ficha con información sobre el club** al dar tap sobre un club en el mapa. La ficha proveerá sin salir del mapa un resumen de la información principal del club.
    - c. La aplicación debe permitir filtrar clubes en el mapa según los siguientes criterios basados en la **información sobre el club**:
      - i. Ubicación (distancia en km)
      - ii. Tipo de deporte (Fútbol, básquet, etc)
      - iii. División femenina (Si/No)
      - iv. División para personas con discapacidad (Si/No)
      - v. Se practica en ambiente techado (Si/No)
      - vi. Categorías disponibles
      - vii. Nombre del club
3. La aplicación debe permitir a un usuario rellenar datos en su perfil. Estos son los siguientes:
    - a. DNI
    - b. Nombre
    - c. Apellido
    - d. Email
    - e. Fecha de nacimiento
    - f. Apodo (opcional)
4. La aplicación debe facilitar al usuario un botón de “Generar QR de inscripción rápida”. El QR de inscripción rápida podrá ser leído por la aplicación para clubes, de manera tal que se pueda identificar a la persona y tomar sus datos sin

necesidad de que el usuario de la aplicación de clubes ingrese los datos manualmente.

5. La aplicación debe permitir incluir minijuegos que tengan como objetivo recompensar a los jugadores por asistir al club.
6. La aplicación debe contar con un minijuego de hacer “jueguitos” con una pelota de fútbol que tenga como objetivo recompensar a los jugadores por asistir al club.

## RF generales de minijuegos del sistema para deportistas

El objetivo de la siguiente lista de requerimientos fue servir como base para la creación del minijuego “Jueguitos” que formaba parte del alcance del MVP. También funcionaba como un lineamiento general para otros juegos que podrían agregarse a la solución en un futuro como parte de una extensión del proyecto.

1. El sistema debe presentar un puntaje inicial de cero puntos en cada partida de cualquier minijuego.
2. Los minijuegos deben tener una cantidad de energía (común a todos) que se consume al jugar a razón de una unidad de energía por partida.
3. La energía de los minijuegos se debe recuperar a razón de una unidad de energía cada 24 horas.
4. Las energías por tiempo solo pueden obtenerse hasta llegar al máximo cinco unidades de energía.
5. La aplicación debe otorgar cinco energías a los jugadores como recompensa por asistir a una práctica deportiva del club, integrándose esta funcionalidad con la función de tomar asistencia de la **app para clubes**.
6. El sistema debe almacenar el puntaje máximo del jugador en una partida (highscore) a nivel histórico para cada minijuego.
7. Cada minijuego debe tener una tabla de posiciones (leaderboard) global a nivel histórico. En ella se deberá mostrar el apodo (nombre y apellido si no tiene) y el

highscore histórico de los 100 jugadores que hayan obtenido mayor puntaje en una sola partida.

8. Cada minijuego debe tener un leaderboard global mensual. En él se deberá mostrar el apodo (nombre y apellido si no tiene) y el highscore del mes actual de los 100 jugadores que hayan obtenido mayor puntaje en una sola partida durante este mes.
9. El sistema debe felicitar al usuario cuando supere un highscore personal al terminar la partida.
10. Cada minijuego debe otorgar insignias (logros) al cumplir requisitos, las mismas deben contar con una descripción (forma de conseguirlo) y una fecha de obtención en caso de que haya sido obtenida. Las insignias solo se pueden obtener una vez por minijuego.

- i. **Tipo de Insignia 1:** Obtener X puntos o más en una partida.

Insignia	X
1.1	10
1.2	50
1.3	250
1.4	500
1.5	650
1.6	800
1.7	1000

- ii. **Tipo de Insignia 2:** Obtener un puntaje acumulado de todas las partidas del jugador en ese minijuego de X puntos o más.

Insignia	X
2.1	100
2.2	500
2.3	1000
2.4	2500
2.5	5000
2.6	10000
2.7	30000

iii. **Tipo de Insignia 3:** Jugar X partidas o más.

Insignia	X
3.1	1
3.2	5
3.3	25
3.4	50
3.5	75
3.6	100
3.7	200

iv. **Tipo de Insignia 4:** pertenecer al grupo de los 10 mejores highscores del leaderboard global del mes actual en al menos X meses.

Insignia	X
4.1	1
4.2	2
4.3	3
4.4	4

11. El sistema debe generar un aviso al usuario al momento de obtener una nueva insignia.

## Diseño de la solución

### Gamificación

La gamificación consiste en la aplicación de mecanismos para incentivar el compromiso y motivación de las personas en entornos que están fuera de los videojuegos al implementar técnicas que se emplean en ellos. Para cumplir su objetivo se fundamenta en aprovechar tendencias humanas como la competición, los logros, la colaboración y la caridad.<sup>10</sup>

Entre los mecanismos de gamificación más comunes se encuentran:

- **Puntos:** representaciones del desempeño del usuario que sirven para motivar la superación y que pueden llegar a usarse para obtener bienes virtuales o reales.
- **Insignias:** símbolos virtuales que indican que se cumplió algún logro, por lo tanto hacen que el usuario se sienta habilidoso.
- **Tablas de posiciones (leaderboards):** ordenamiento competitivo por el desempeño de una lista de participantes dentro de un intervalo de tiempo determinado. Se basa en el deseo del usuario de llegar a las mejores posiciones.
- **Bienes virtuales:** cualquier objeto no físico que pueda comprarse con los puntos obtenidos en las partidas, recompensando de esa forma al jugador por su desempeño.<sup>11</sup>

### Modelo Mechanics-Dynamics-Aesthetics (MDA)

En el diseño de juegos, MDA es una herramienta utilizada para analizar juegos. Formaliza el consumo de juegos desglosándolos en mecánicas, dinámicas y estéticas. En consecuencia, busca generar definiciones precisas para estos términos y explicar cómo se relacionan entre sí e influyen en la experiencia del jugador.

- Las **mecánicas** describen los componentes particulares del juego, a nivel de representación de datos y algoritmos.
- Las **dinámicas** describen el comportamiento en tiempo de ejecución de las mecánicas que actúan sobre las entradas y salidas del jugador a lo largo del tiempo.
- Las **estéticas** describen las respuestas emocionales deseables evocadas en el jugador cuando interactúa con el sistema de juego.<sup>12</sup>

## Game design document (GDD)

Un GDD es un documento de diseño de software que sirve como modelo a partir del cual se construirá un videojuego. Ayuda a definir el alcance del juego y establece la dirección general del proyecto, manteniendo a todo el equipo de desarrollo en sintonía.<sup>13</sup>

## Motor de videojuegos

Los motores de videojuegos son herramientas de desarrollo de software diseñadas para reducir el costo, la complejidad y el tiempo de comercialización necesarios en el desarrollo de videojuegos. Estas crean una capa de abstracción sobre las tareas más comunes en el desarrollo de videojuegos. Además las herramientas se diseñan para funcionar como componentes interoperables que pueden reemplazarse por completo o ampliarse con software de terceros.<sup>14</sup>

## Material Design

“Material” es un sistema de diseño creado por Google, quienes lo definen como “un sistema que sirve para construir experiencias digitales de alta calidad para dispositivos móviles y la web, que fue creado como un lenguaje visual que combina principios del buen diseño con innovación técnica y científica”.

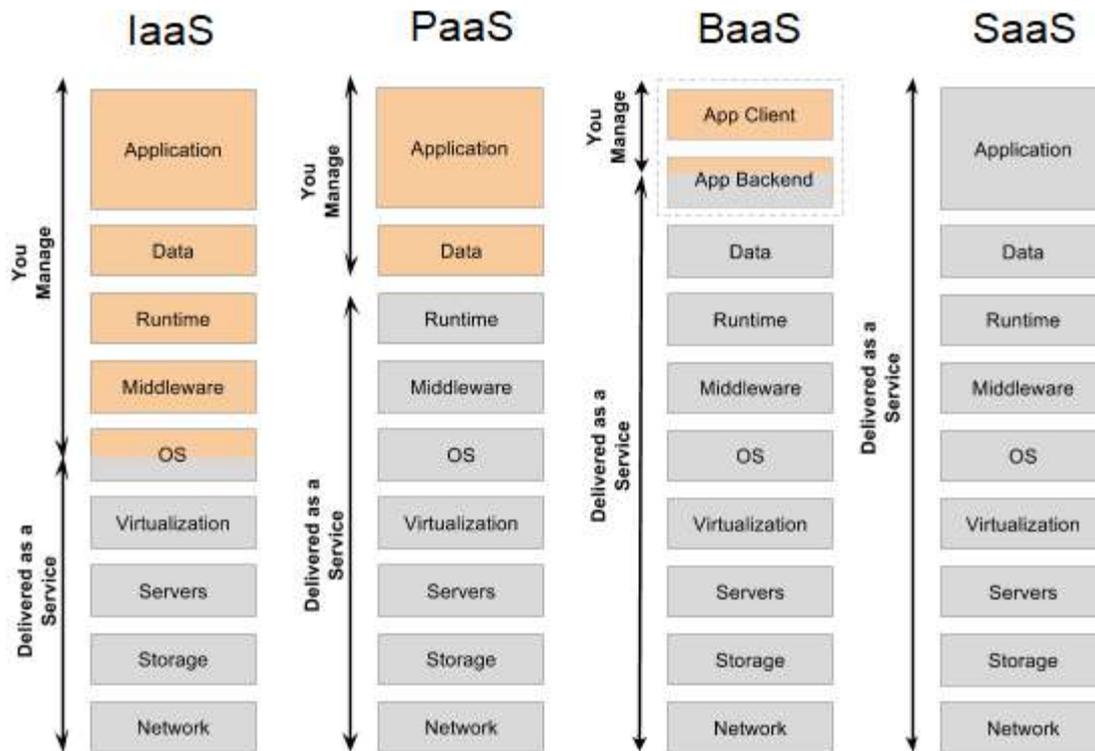
## Backend as a Service (BaaS)

BaaS es un modelo para el desarrollo web y de aplicaciones móviles que provee una forma de vincular las aplicaciones desarrolladas a un servicio que provee el backend necesario para realizar almacenamiento en nube, al mismo tiempo que provee características tales como servicios analíticos, gestión de usuarios, integración con redes sociales, notificaciones push, cloud functions, entre otros. Estos servicios se prestan tanto a través de SDK como API.

La utilización de BaaS elimina la necesidad de desarrollar, administrar y mantener servidores, bases de datos, APIs, lo cual permite reducir drásticamente los tiempos de desarrollo. Esto resulta conveniente al llevar a cabo un proyecto de metodología Lean.<sup>15</sup>

**Figura 3**

*Comparación de BaaS respecto a otros modelos de servicio.*



## Test-driven development (TDD)

TDD es un enfoque de desarrollo de software en el que se escribe una prueba antes de escribir el código. Una vez que el nuevo código pasa la prueba, se refactoriza a un estándar aceptable.

TDD garantiza que el código fuente se pruebe exhaustivamente y conduce a un código modularizado, flexible y extensible. Se enfoca en escribir solo el código necesario para pasar las pruebas, haciendo que el diseño sea simple y claro.<sup>16</sup>

## Dependency injection (DI)

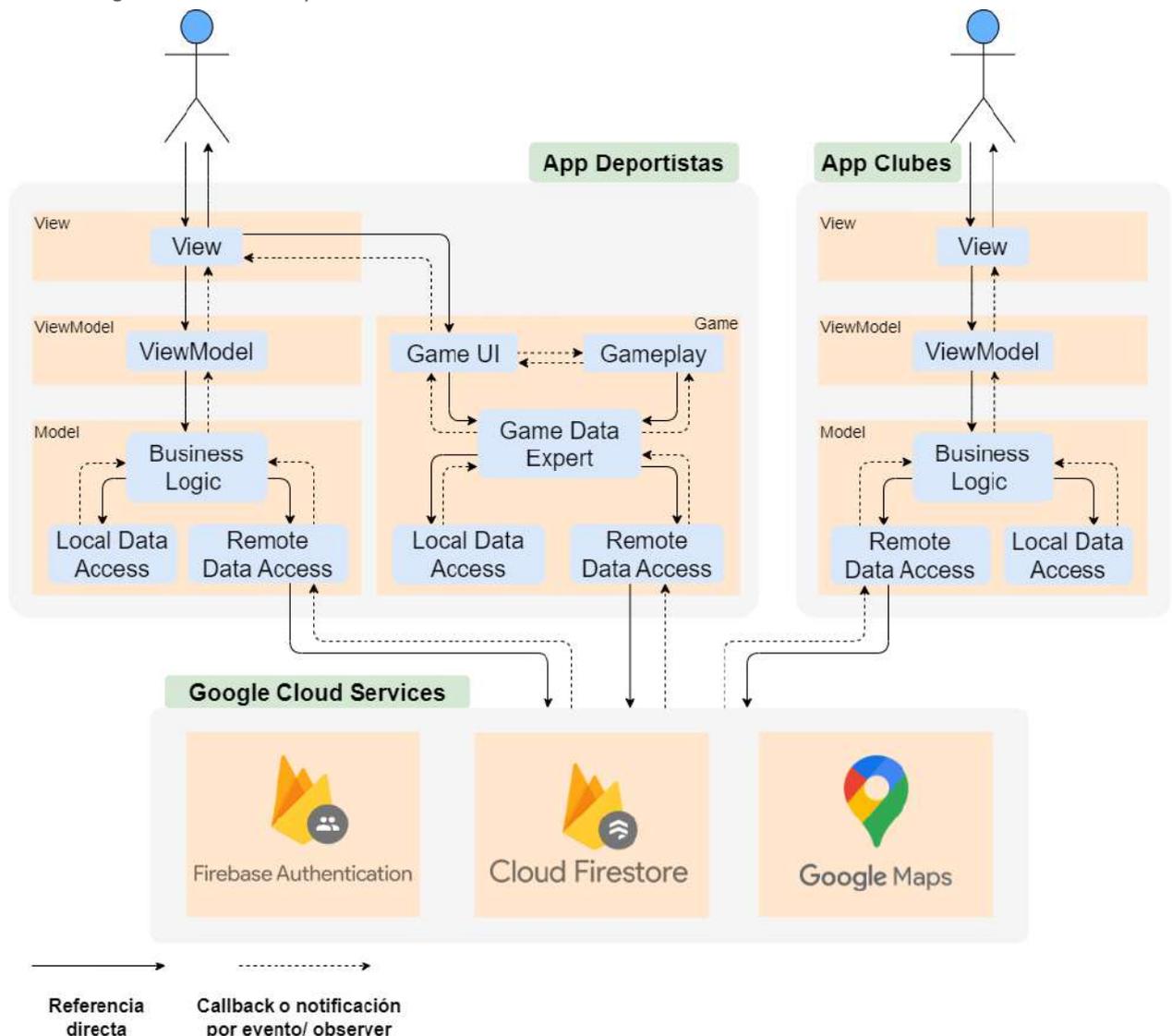
DI es un patrón de diseño en el que un objeto o función recibe otros objetos o funciones de los que depende. Su objetivo es separar las responsabilidades de construir objetos y usarlos, lo que lleva a programas con bajo acoplamiento.<sup>17</sup>

## Arquitectura del sistema

La arquitectura del sistema se basa en **Backend as a Service** (BaaS). El resultado es un sistema compuesto por aplicaciones móviles que consumen servicios de Firebase y Google Cloud. La selección de las tecnologías incluidas en el diseño final de la arquitectura es detallada en secciones posteriores del informe.

**Figura 4**

*Diagrama final de arquitectura del sistema*



La **principal ventaja** de utilizar esta arquitectura sobre un modelo tradicional cliente-servidor, es que se disminuyen considerablemente los tiempos de desarrollo al eliminar la necesidad de desarrollar parcial o totalmente los siguientes elementos:

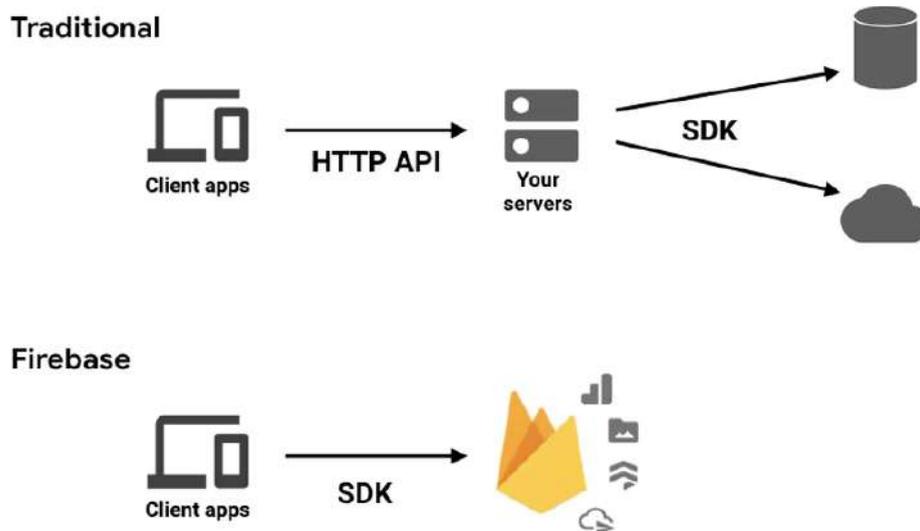
- **APIs** para comunicar aplicaciones móviles con el servidor

- **Servidor**
- **ORM** en caso de que se utilice una base de datos relacional
- **Base de datos**

**Las desventajas principales** de utilizar esta arquitectura son que se pierde control sobre el servidor y la base de datos, y que se crea dependencia hacia un tercero. Con respecto al control sobre el servidor, no se prevé que exista necesidad de soluciones específicas en servidor o base de datos a corto y mediano plazo, pero incluso si llegaran a ser necesarias en algún escenario, se podría resolver con un nuevo data source, esta vez creado in-house, utilizando múltiples data-source en simultáneo. Al mismo tiempo, la dependencia de un tercero, en este caso Google, puede representar una potencial amenaza. Sin embargo, si fuera necesario, se podría cambiar de data source por otra solución BaaS o uno creado in-house.

**Figura 5.1**

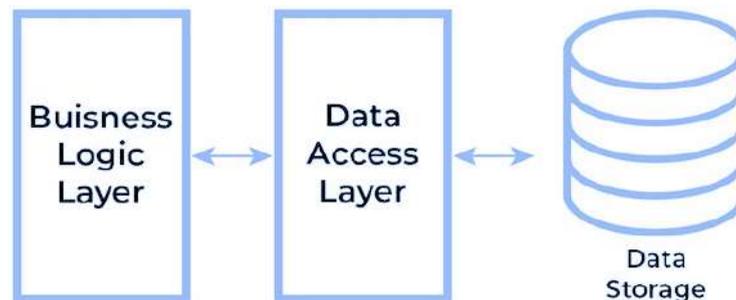
*Stack de desarrollo tradicional vs BaaS*



En el análisis anterior se sugiere la posibilidad de cambiar el data source sin que esto implique inconvenientes. Esto se debe a que la arquitectura interna de las aplicaciones hace que cambiar de backend sea relativamente sencillo. Esto es así porque se utiliza un **modelo de capas**, donde en caso de cambiar el data source, sería necesario reemplazar la **capa de acceso a datos** (data access layer), pero este cambio debería ser completamente transparente al resto del sistema, ya que el sistema interactúa con la **capa de negocios** (business logic layer) y esta no cambiaría.

**Figura 5.2**

*Modelo de capas del sistema*



## Base de datos

Dado que Cloud Firestore es una base de datos orientada a documentos, la manera en la que se organiza la información es diferente a una base de datos estructurada. En lugar de tablas, referencias y relaciones, la información en Firestore se organiza en: <sup>18</sup>

- Documentos
- Múltiples colecciones
- Subcolecciones dentro de documentos

No existe una opción definitiva a elegir para organizar la información, sino que debe elegirse a criterio según el contexto.

Para este proyecto se decidió estructurar la base de datos en las siguientes colecciones y subcolecciones:

- **Clubes.**
  - **Deportes:** deportes incluidos en el club.
  - **Deportistas:** deportistas registrados por el club. Esto genera una independencia de la aplicación de clubes sobre la de deportistas.
  - **Pagos:** Constancias del dinero recibido.
- **Deportistas:** datos personales de los deportistas.
- **PerfilesGaming:** datos de los deportistas referidos a su progreso en los minijuegos.
- **LeaderboardsGaming:** para almacenar el leaderboard histórico y del mes actual.

# Implementación de la solución

## Elección de tecnologías

Como parte del diseño de sistema desde la planificación de la arquitectura, fue necesario evaluar y seleccionar el conjunto de tecnologías a ser utilizado. Para ello se tomaron algunos criterios de referencia, como costos económicos (cuotas de uso, licencias, etc), costos técnicos (conocimiento previo de las tecnologías a utilizar, dificultad de aprendizaje e implementación, complejidad técnica, etc.), entre otros.

Si bien un conjunto de criterios de referencia fue establecido, el proceso de elección fue flexible, cada caso fue analizado de forma particular y teniendo en consideración su contexto. Cabe destacar que el equipo estaba abierto a la posibilidad de aprender nuevas tecnologías. En esta sección se presentan las alternativas que fueron evaluadas y las que fueron seleccionadas. Se listan diferencias, comparaciones, ventajas y desventajas, etc; y la justificación del criterio por el cual fueron elegidas.

## Framework para el desarrollo de las aplicaciones

Para el desarrollo de las aplicaciones móviles, se realizó un análisis de las dos tecnologías con mayor cuota de mercado que son Flutter y React Native. Además se consideró como otra opción al desarrollo nativo utilizando Kotlin.

### Flutter

Flutter es un SDK creado por Google que permite desarrollar aplicaciones para Android, iOS, web, Linux, MacOS, Windows y Google Fuchsia; desde una única base de código.

### Ventajas

- Cuenta con el respaldo de Google, actualizaciones periódicas y una comunidad extensa y activa.
- Herramienta gratuita y open source.
- Al pertenecer a Google, permite una integración sencilla con otras herramientas del stack de esta empresa.

### Desventajas

- Utiliza el lenguaje Dart, el cual actualmente sólo es utilizado por el mismo Flutter. Cabe destacar que el equipo no tiene conocimiento en el lenguaje. Sin embargo su curva de aprendizaje es relativamente sencilla para desarrolladores con conocimiento en javascript.
- La performance es peor que en Android nativo.

### React Native

React native es un framework creado por Meta (La empresa matriz de Facebook) que permite desarrollar aplicaciones para Android, Android TV, iOS, MacOS, tvOS, web, Windows, Oculus y UWP; utilizando React.

#### Ventajas

- Cuenta con el respaldo de Facebook, actualizaciones periódicas y una comunidad extensa y activa.
- Herramienta gratuita y open source.
- Utiliza JavaScript, que es actualmente uno de los lenguajes más dinámicos, performantes y populares. El equipo de ingeniería tiene conocimientos con este lenguaje.

#### Desventajas

- La performance es peor que en Android nativo.

### Android nativo con Kotlin

Es posible desarrollar de forma nativa para Android sin utilizar frameworks, utilizando Kotlin y Android Studio. Kotlin es un lenguaje de programación que corre sobre la máquina virtual Java.

#### Ventajas

- Herramienta gratuita y open source.
- Mayor control en general sobre todos los aspectos de la aplicación, especialmente en factores de bajo nivel. Resultando en, por ejemplo, mejor performance, menor tamaño de la aplicación, entre otros.
- Da la posibilidad de resolver algunos problemas que los frameworks no pueden resolver. Este aspecto actualmente tiene menos peso dado el alto grado de madurez de la mayoría de los frameworks.

#### Desventajas

- Desarrollo considerablemente más lento y complejo por la ausencia de algunos elementos de los frameworks.
- Imposibilidad de realizar exportaciones multiplataforma.
- Dificultad para unificar las interfaces de usuario entre múltiples plataformas.
- Por su relativamente baja popularidad no existe una comunidad tan grande ni una cantidad amplia de documentación.

Según lo analizado, el desarrollo nativo con Kotlin podría ser una solución ideal cuando se desea realizar implementaciones complejas y específicas, para las que sería demasiado costoso utilizar un framework ya existente debido a las limitaciones que podría tener. Sin embargo, no es algo que aplique al presente proyecto, por lo que no se justificaría sacrificar las facilidades que puede ofrecer un framework. Luego, la elección recae entre Flutter y React Native. Realmente no hay una de las dos opciones que sea notablemente superior a la otra, al menos en este caso particular, y se considera que ambas serían opciones igualmente válidas.

Sin embargo, al tener que elegir uno, y considerando el factor de su pertenencia a Google, que da un mayor grado de integración con algunos servicios de dicha empresa (que serán vistos más adelante en otras selecciones tecnológicas), el framework elegido para el desarrollo de las aplicaciones es **Flutter**.

## Motor de videojuegos

Para la creación de un minijuego que incentivase la participación de los deportistas en los entrenamientos por medio de la gamificación se decidió utilizar un motor de videojuegos. El motivo principal de esta elección fue acelerar el desarrollo al usar sistemas preexistentes diseñados para el desarrollo de videojuegos como animaciones, físicas, renderizado, entre otros. Para la lista de candidatos solo se tuvo en cuenta a los que tenían una capacidad de integración sencilla con flutter para poder convivir dentro de una misma aplicación.

### Unity Engine

Unity (también llamado Unity3D) es un motor de juegos multiplataforma desarrollado por Unity Technologies, anunciado y lanzado por primera vez en junio de 2005 como un motor de juegos exclusivo para Mac OS X. Desde entonces, el motor se ha ampliado gradualmente para admitir una variedad de plataformas de escritorio,

móviles, consolas y realidad virtual. Se puede utilizar para crear juegos tridimensionales (3D) y bidimensionales (2D), así como simulaciones interactivas y otras experiencias.<sup>19</sup>

## Ventajas

- Fácil de usar.
- Permite un prototipado rápido.
- Muchas funcionalidades y herramientas por su larga trayectoria de lanzamientos desde 2005.
- Amplia cantidad de documentación.
- Gran comunidad.<sup>20</sup>
- Integración con flutter disponible por medio de paquetes de terceros
- Portabilidad a dispositivos móviles.
- Gran cantidad de paquetes gratuitos y de pago disponibles para acelerar el desarrollo.<sup>21</sup>
- Permite el acceso a una versión actualizada de su código fuente.<sup>22</sup>
- Licencia gratuita si tus ganancias en los últimos 12 meses son menores a 100.000 dólares.<sup>23</sup>
- Se puede integrar publicidades por medio de un paquete gratuito.

## Desventajas

- Su gestión de la memoria por medio de *garbage collector* (GC) usualmente genera la necesidad de aplicar técnicas de optimización para la reducción de operaciones costosas sobre la memoria;
- No permite la modificación de su código fuente de manera gratuita;
- Sus costos de licencia podrían cambiar en el futuro;
- Presenta soporte dedicado solo para licencias de 2.040 dólares por año como mínimo.<sup>23</sup>

## Flame Engine

Flame es un paquete de Flutter que proporciona un motor de videojuegos modular con el objetivo de brindar un conjunto completo de soluciones prediseñadas para el desarrollo de videojuegos. Aprovecha la infraestructura provista por Flutter pero simplifica el código necesario para llevar a cabo ese tipo de proyectos.<sup>24</sup>

Flame es completamente gratuito y de código abierto bajo la licencia permisiva del MIT. Su primer lanzamiento fue en noviembre del 2017. [25](#)

### **Ventajas**

- Completamente gratuito.
- Es un paquete de Flutter por lo que puede incluirse fácilmente a la aplicación de deportistas.
- Portabilidad a dispositivos móviles.
- Es open source.
- Se puede integrar publicidades por medio de un paquete gratuito.

### **Desventajas**

- Comunidad muy pequeña en comparación a Unity. [26](#)
- Hace falta incluir más paquetes para obtener funcionalidades que son básicas en otros motores.
- Poco tiempo de lanzamientos desde su versión inicial.
- No tiene muchas funcionalidades y herramientas que otros motores si tienen. Por ej. un editor para colocar objetos en el juego, y agregar o modificar fácilmente sus componentes.
- Buena cantidad de paquetes gratuitos pero la mayoría no están diseñados para el desarrollo de videojuegos. [27](#)
- No brinda soporte dedicado.
- Solo permite el desarrollo de videojuegos en 2D.

En base a las ventajas y desventajas expuestas para cada motor de videojuegos se decidió seleccionar a Unity para el desarrollo del minijuego de la aplicación para deportistas. Principalmente fueron valoradas las herramientas del motor y su gran comunidad. Además se tuvo en consideración la experiencia de uno de los autores de este informe al utilizar el motor de videojuegos Unity en el ámbito profesional, por lo que también influiría a mejorar la velocidad de desarrollo.

A pesar de estar presente en todas las opciones presentadas, fue importante poder contar con una licencia gratuita para utilizar el motor sin cargos adicionales y permitir una posible extensibilidad futura del proyecto sin que tenga un costo de entrada. Además el costo de licencia no es un impedimento porque el proyecto no tiene fines de

lucro. Por último se destaca la oportunidad de incluir, como una opción adicional para extender el proyecto en el futuro, publicidades para generar ingresos que aporten al objetivo de que el producto sea autosustentable.

## Backend y base de datos

Para la selección de esta tecnología existía una restricción desde la definición de la arquitectura: Era necesario utilizar BaaS. Se analiza el servicio BaaS ofrecido por Google: Firebase.

### Firestore

#### Ventajas

- Fácil integración tanto con **Flutter** como con **Unity**.
- Mucha documentación tanto en inglés como en español.
- Inicialmente gratuito. Firebase se vuelve pago únicamente tras consumir una determinadas cuotas gratuitas. Estas pueden ser mensuales o diarias dependiendo de la tecnología.
- Se delega la seguridad del backend y la base de datos sobre un gigante de la tecnología.
- Firebase alega tener muy alta disponibilidad para sus soluciones de bases de datos (al menos 99,95%).

#### Desventajas

- No hay instalación “on-premise”. Es decir, no es posible utilizar Firebase en servidores privados. Es necesario utilizar a Google como intermediario.
- Las reglas de seguridad tienen limitaciones (de procesamiento y complejidad que permiten) y se especifican en un lenguaje que no es muy amigable.

Para el almacenamiento de datos, Firebase ofrece dos soluciones de bases de datos basadas en la nube y accesibles para los clientes que admiten sincronización en tiempo real. Estas son **Cloud Firestore** y **Realtime Database**.

## Comparativa Cloud Firestore / Realtime Database

A continuación se muestra la comparativa entre Cloud Firestore y Realtime Database, que presenta la documentación del sitio oficial de Firebase. [28](#)

**Tabla 1**

*Tabla comparativa entre Cloud Firestore y Realtime Database*

Realtime Database	Cloud Firestore
<b>Escrituras y transacciones</b>	
<p><b>Operaciones básicas de escritura y transacción.</b></p> <ul style="list-style-type: none"> <li>● Permite escribir datos mediante operaciones de configuración y actualización.</li> <li>● Las transacciones son atómicas en un subárbol de datos específico.</li> </ul>	<p><b>Operaciones avanzadas de escritura y transacción.</b></p> <ul style="list-style-type: none"> <li>● Permite escribir datos a través de operaciones de configuración y actualización, así como operadores de arreglos y numéricos.</li> <li>● Las transacciones pueden leer y escribir datos atómicamente desde cualquier parte de la base de datos.</li> </ul>
<b>Confiabilidad y rendimiento</b>	
<p><b>Realtime Database es una solución regional.</b></p> <ul style="list-style-type: none"> <li>● Disponible en configuraciones regionales. Las bases de datos se limitan a la disponibilidad zonal.</li> <li>● Latencia extremadamente baja, es la opción ideal para sincronizar estados con frecuencia.</li> </ul>	<p><b>Cloud Firestore es una solución regional y multirregional con escalado automático.</b></p> <ul style="list-style-type: none"> <li>● Aloja los datos en varios centros de datos de distintas regiones, lo que garantiza escalabilidad y confiabilidad.</li> <li>● Disponible de formas regionales o multirregionales en todo el mundo.</li> </ul>
<b>Escalabilidad</b>	
<p><b>El escalado necesita fragmentación.</b></p> <p>Escala hasta alrededor de 200,000 conexiones simultáneas y 1,000 escrituras por segundo en una misma base de datos. Para un escalado</p>	<p><b>El escalado es automático.</b></p> <p>Escala de forma automática. Los límites de escalado son de 1 millón de conexiones simultáneas y 10,000 operaciones de escritura por</p>

mayor, se deben fragmentar los datos en distintas bases.	segundo*. <i>*Al momento de la creación de esta tabla.</i>
<b>Seguridad</b>	
Lenguaje de reglas en cascada que separa la autorización de la validación.	Reglas sin formato de cascada que combinan autorización y validación.
<b>Sistema de precios</b>	
Se cobra solo por ancho de banda y almacenamiento, pero con una tarifa mayor.	Se cobra por operaciones ejecutadas en la base de datos y, con una tarifa menor, por ancho de banda y almacenamiento.
<b>Modelo de datos</b>	
Almacena datos como un gran árbol JSON	Almacena datos como colecciones de documentos.
<b>Compatibilidad sin conexión y en tiempo real</b>	
Soporte sin conexión para clientes de Apple y Android.	Soporte sin conexión para clientes de Apple, Android y la Web.
<b>Presencia (Ver estado de conexión del cliente)</b>	
Se admite la presencia	No compatible de forma nativa
<b>Realización de consultas</b>	
Consultas directas con funciones de ordenamiento y filtrado limitadas.	Consultas indexadas con ordenamiento y filtrado compuestos.

### Otras alternativas a Firebase no analizadas

- Parse platform
- Kumulos
- MOCA
- Kona Cloud

### Elección

Se selecciona Cloud Firestore como base de datos principalmente tiene la posibilidad de operar en condiciones de conectividad limitada o nula, tiene mayor

disponibilidad y su tipo modelado de datos permite una mayor flexibilidad para estructuras de datos complejas. Por último, cabe destacar que se integra fácilmente con otras herramientas de Firebase. Esto resulta de gran utilidad porque permite implementar la autenticación de los usuarios con Firebase Auth y usarla como base para las reglas de seguridad de Firestore.

### Elección de sistema para el versionado

El presente proyecto fue desarrollado por un equipo de dos personas. Esto implicaba la existencia potencial de riesgo de desincronización de versiones para el código fuente y la documentación del proyecto. Por lo tanto desde el comienzo se eligieron las siguientes herramientas de versionado:

- **Git** como herramienta para gestionar el versionado del código. Fue elegida principalmente porque es una herramienta que está asentada como una de las más sólidas del mercado, y porque el equipo ya tenía experiencia utilizándola con anterioridad.
- **GitHub** como plataforma para alojar el repositorio de Git. Fue elegida también por ser una herramienta muy popular y establecida. Además se tuvo en consideración la experiencia del equipo con ella.
- **Google Drive** para el versionado de la documentación y otros archivos. Esta tecnología fue elegida principalmente por brindar la posibilidad de utilizar las aplicaciones de oficina de Google (como Docs, Sheets, Slides, etc.) e integrarse con herramientas para realizar diagramas (como Draw.io). Además permitía trabajar con archivos en línea y de manera concurrente sobre Google Drive. Esto posibilitaba que los miembros del equipo trabajen sobre una única versión sincronizada de los archivos. En consecuencia se eliminaba la necesidad de gestionar la sincronización manualmente, pero sin prescindir de herramientas importantes como el historial de versiones.

### Elección de servicio de mapping

En alineación con la mayoría del stack de tecnología a utilizar en el proyecto, se opta por elegir **Google Maps**. No sólo es el servicio más popular sino ampliamente

reconocido como el mejor. Algunas otras alternativas existentes en el mercado son Mapquest o Bing Maps.

## Diseño de interfaces de usuario

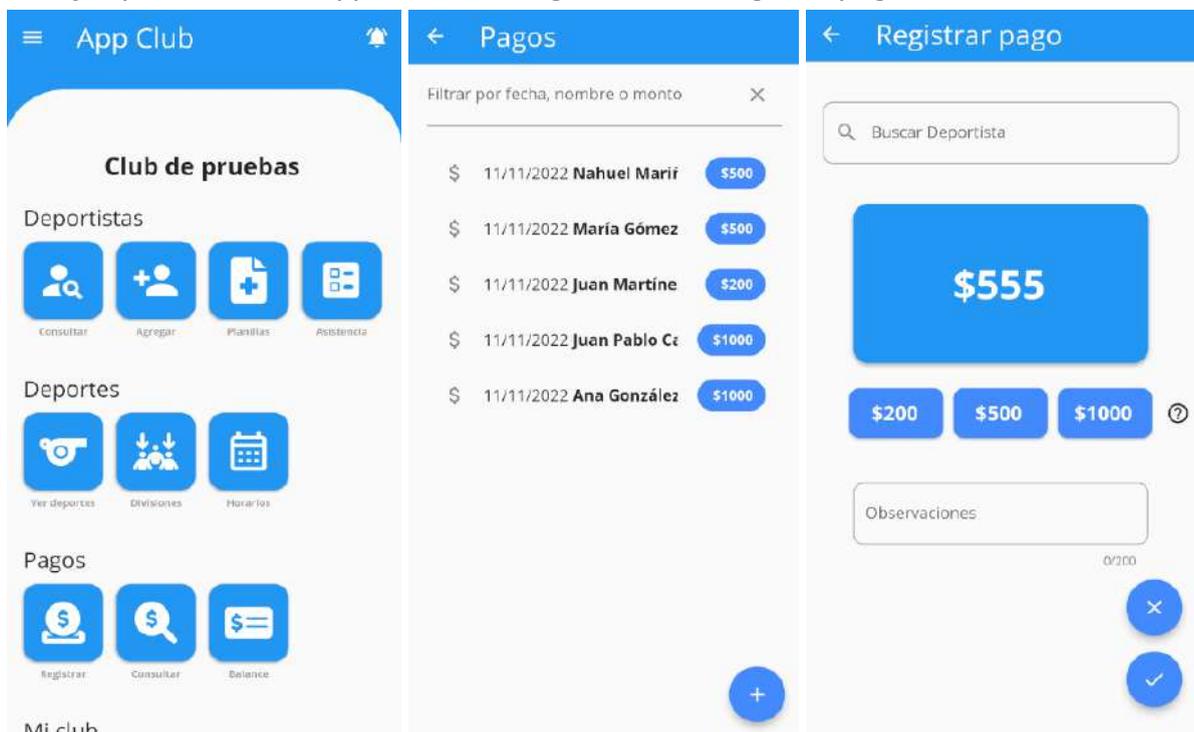
### Interfaces de aplicación para clubes y deportistas

En lugar de utilizar un software de maquetado, se opta por utilizar directamente “design prototyping”, o diseñar directamente creando el prototipo en Flutter. Si bien el diseño de interfaces de usuario directamente en un prototipo sin backend es más costoso que utilizar una herramienta de maquetado, resulta conveniente a largo plazo pues al finalizar el diseño de interfaces ya se tiene completamente implementado un prototipo de interfaz de usuario, al que sólo faltaría agregarle la lógica correspondiente. Esto es conveniente según la filosofía Lean Development y el desarrollo de un MVP.

Además, se aplicaron principios de material design a fin de tener un conjunto de normas de diseño que permitan lograr un producto visualmente consistente y estéticamente agradable.

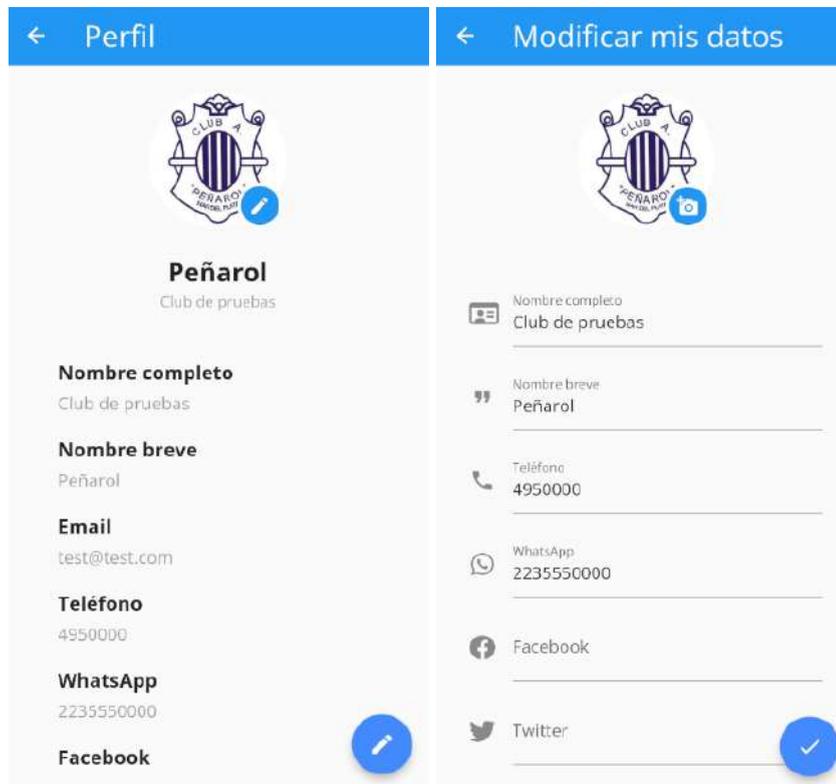
**Figura 6**

*Ejemplos interfaces app clubes: Navegación hasta registrar pago*



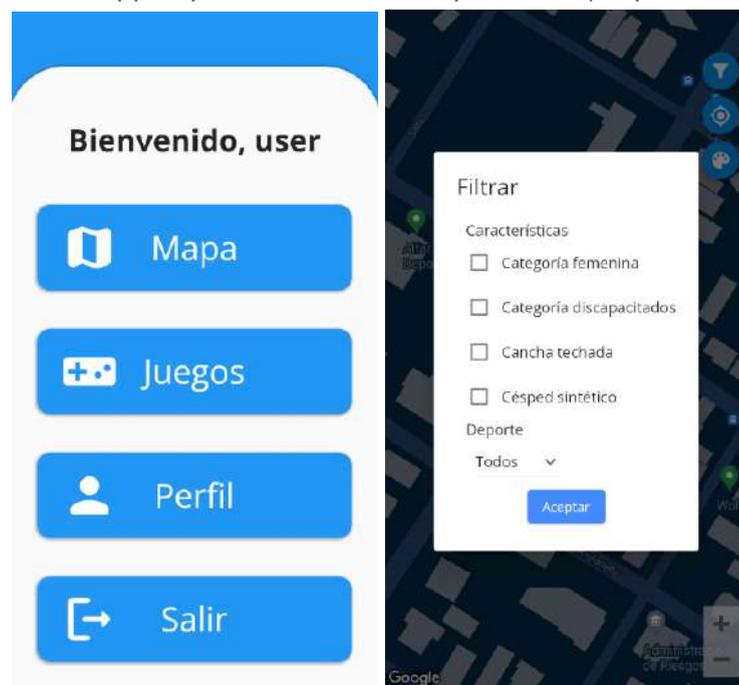
**Figura 7**

*Ejemplos interfaces app clubes: Perfil y editar perfil*



**Figura 8**

*Ejemplos interfaces app deportistas: Home, mapa + filtro (Mapa con tema oscuro)*



## Prototipado

La mayoría de las interfaces de usuario fueron el resultado de la iteración sobre un prototipo inicial, diseñado en función de la idea general que se pensó inicialmente para las interfaces y sobre el que luego se hicieron mejoras iterativamente. Por ejemplo, esta fue la evolución de la pantalla “home” de la aplicación para deportistas:

Una ventaja de haber utilizado diseño de interfaces mediante prototipado, fue que permitió trabajar de forma ágil y en simultáneo considerando que muchas características podrían encontrarse en distintas fases de su circuito Kanban. Es decir, se podía trabajar en el diseño de una pantalla (trabajando sobre el prototipo) mientras sobre el mismo prototipo pero en otra pantalla se trabajaba sobre el desarrollo de su funcionalidad.

## Interfaces del minijuego jueguitos

El diseño de interfaces para el minijuego fue generado utilizando la herramienta de maquetado Figma. Para cada pantalla se tuvo en consideración la inclusión futura de un banner de publicidad en la parte inferior.

### Figura 9

*Maquetado de login y menú principal.*



**Figura 10**

Maquetado de ventanas de insignias.



*Nota.* Las insignias difieren de forma considerable con el arte final esperado en el que cada una debería ser distinta a las demás.

**Figura 11**

Maquetado de menú de personalización y leaderboards.



*Nota.* El menú de personalización fue maquetado con la versión final del arte de los personajes del juego.

## Producto

### Disponibilidad

Una de las grandes ventajas de utilizar Firebase es que el complemento de Firebase para Flutter permite trabajar en condiciones de conectividad limitada o nula sin truncar completamente el funcionamiento de la aplicación. Es decir, una persona puede realizar una escritura de datos, como registrar un pago, **sin tener acceso a Internet**. La escritura es almacenada de forma local en el teléfono y se persiste (es decir, el usuario puede cerrar la app o incluso reiniciar el sistema operativo completamente), pero una vez que la conexión se restablezca, la operación se hará efectiva.

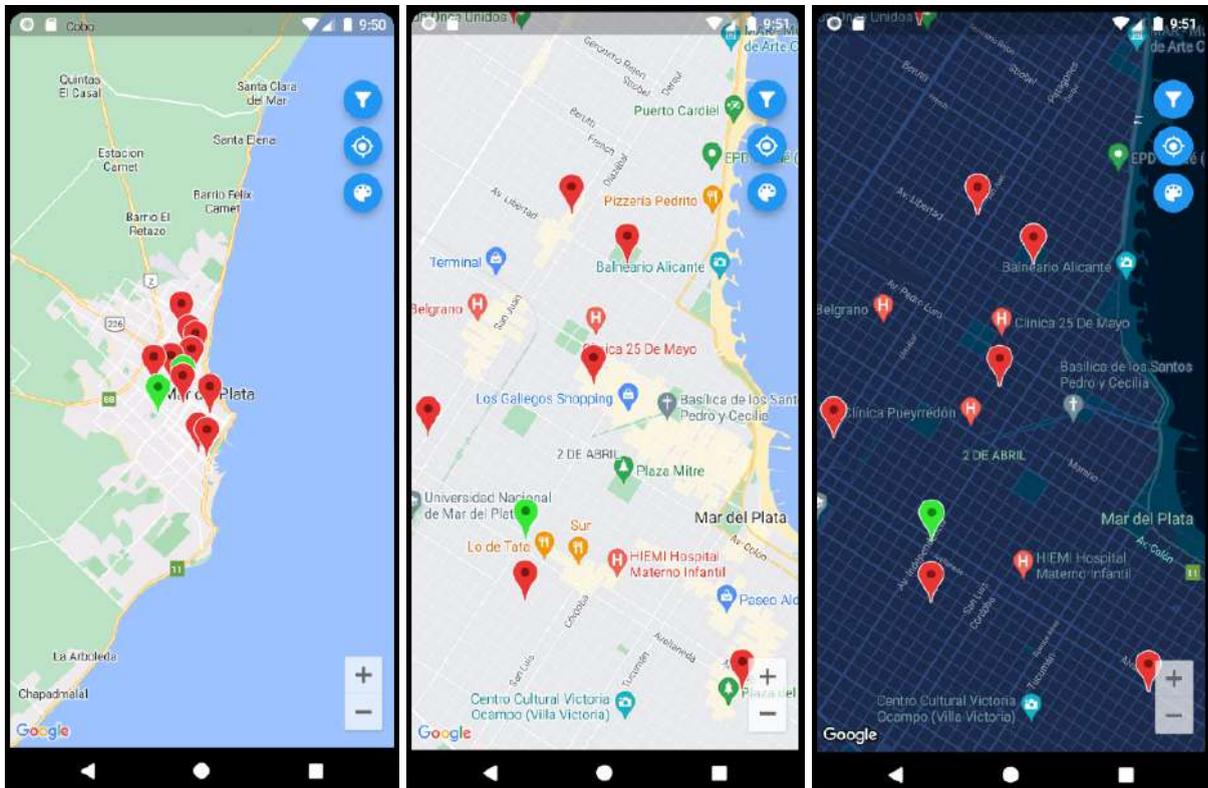
Esta característica es especialmente útil para la operatividad de la aplicación para clubes (donde fue aplicada), teniendo en cuenta que los clubes de barrio generalmente no tendrán conexión vía Wi-Fi, y la conexión con paquete de datos suele ser más inestable e incluso estar sujeta a anchos de banda limitados, o directamente podría no haber.<sup>29</sup>

### Mapa de clubes

Una de las funcionalidades principales que presentó la aplicación para deportistas es la capacidad de identificar clubes deportivos dentro de un mapa interactivo. El objetivo de esta herramienta era lograr que los usuarios puedan identificar de manera sencilla los clubes en la zona que se adapten a sus necesidades. Para cumplir con ese propósito se implementó un sistema de filtros que permitían reducir la cantidad de clubes mostrados a los que cumplan con ciertos criterios de búsqueda (especificados en el RF 2 de la sección de análisis de requerimientos de la aplicación para clubes). Estos filtros incluyen aspectos de inclusión tales como la búsqueda de deportes con práctica femenina o personas con discapacidad. Por lo tanto, es una característica que permitiría aumentar la inclusión de esos sectores en el deporte.

**Figura 12**

*Mapa con tema claro desde lejos, mapa con tema claro desde cerca, mapa con tema oscuro desde cerca.*



*Nota.* Los pines de color rojo y verde representan los clubes visibles. Estas interfaces son prototipos y están sujetas a cambios.

## Gestión del club

Las principales características de la aplicación para clubes son herramientas de gestión de clubes como el registro de socios, pagos y asistencias. Estas características fueron incluidas en el sistema con un objetivo doble: Además de ser uno de los objetivos del proyecto por su aporte a los clubes, fomenta la utilización del sistema para aumentar la oferta que tendrán los deportistas a disposición en el mapa de la aplicación de deportistas.

Los clubes pueden crear todos los deportes que se practican en su club, inscribir deportistas en cada uno de ellos, tomar asistencias para los inscriptos de cada deporte, y registrar pagos por personas. El registro de pagos se abstrae de los deportes, por si el club permitiera a una persona participar en varias prácticas y realizar un pago único. Cabe destacar que solo es posible registrar esos pagos desde la aplicación de clubes y que sirven como un historial de pagos pero no se realiza ninguna transacción de dinero

en la aplicación. Las características de gestión implementadas pueden verse con más detalle en el documento anexo de requerimientos funcionales.

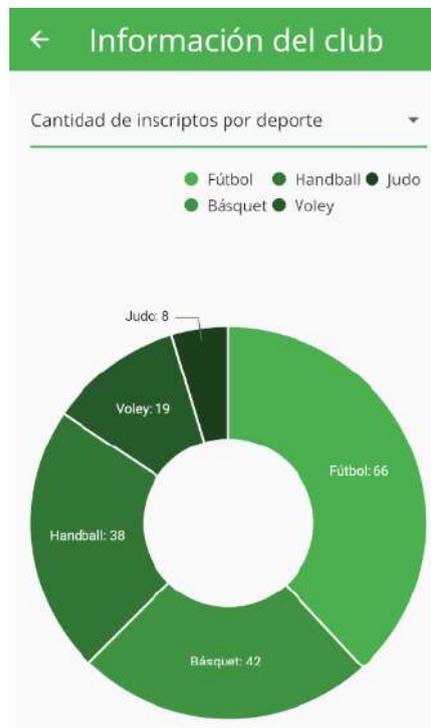
## Métricas

Un sistema de métricas fue incluido para facilitar el análisis del estado actual y la evolución de los siguientes indicadores:

- Porcentajes del total de deportistas por deporte
- Número de inscripciones al club
- Porcentajes del total de deportistas por rango etario

### Figura 12.5

*Pantalla de métricas*



## Jerarquía, clubes y roles

Todo usuario que desee puede descargar la aplicación para clubes y crear un perfil de club. El diseño del sistema prevé que, una vez que el club haya creado y completado su cuenta, pueda solicitar una **validación manual**. La validación manual convertirá a una cuenta de club ordinaria en una cuenta de **club verificado**. Sólo los clubes verificados pueden ser visibles en el mapa de clubes de la aplicación para deportistas.

Al mismo tiempo, un club verificado puede autogestionar usuarios internos con roles específicos, como “Finanzas”, que puede acceder a las secciones de administración relacionadas a finanzas; “Profe”, que puede acceder a secciones como la toma de asistencia, “Administrador” que puede realizar todas las mismas acciones que la cuenta owner, excepto gestionar usuarios, etc. Algunas de estas características no se encuentran implementadas en el MVP.

## Dynamic link para abrir minijuego

Por problemas técnicos durante la integración del minijuego dentro de la aplicación de deportistas se decidió separar los sistemas en dos aplicaciones distintas. Sin embargo, para seguir facilitando el acceso al minijuego desde la app de deportistas, se agregó un dynamic link que se activaba al tocar un botón. Como consecuencia se abría la app del minijuego en caso de que estuviera instalada o redireccionaba a un link de descarga en Play Store.

## Utilización de estrategias de gamificación

Diversas estrategias de gamificación fueron consideradas para mejorar la experiencia de juego y aumentar la retención de los usuarios en el minijuego “Jueguitos”. Estas fueron la utilización de puntos, leaderboards e insignias.

Los primeros se idearon como una manera de medir el desempeño del jugador en una partida y servir como punto de referencia para la superación personal. También se utilizaron como una medida del progreso del nivel de dificultad en la partida. Debido a eso los puntajes más altos requerirían un esfuerzo mayor del jugador.

En referencia a los leaderboards, se idearon dos tipos en base a intervalos temporales distintos, histórico y del mes actual. Sus objetivos respectivos eran generar una meta de superación a largo y corto plazo. Además el leaderboard del mes actual se vaciaba al iniciar un nuevo mes, permitiendo que los jugadores que no fueran tan habilidosos también tuvieran la oportunidad de llegar a las primeras posiciones de la tabla.

Por último, las insignias representaban metas de puntaje máximo en una partida, puntos acumulados entre todas las partidas jugadas, cantidad de partidas y número de

meses distintos en los que el jugador fue parte del top 10 en el leaderboard del mes. El objetivo de la utilización de insignias fue conseguir que los usuarios se sientan habilidosos por obtenerlas y se motiven para conseguir las insignias que les faltan (coleccionismo).

## Motivar asistencia a prácticas deportivas

El videojuego “Jueguitos” fue creado con el objetivo final de generar un incentivo extra para participar de las actividades del club. Con ese propósito en mente se idearon dos sistemas complementarios, uno para la gestión de energías del juego y otro para enviar recompensas por asistencias. El primero brindaba la posibilidad de jugar una partida por cada energía gastada. En consecuencia se limitó la cantidad de energía que se podría ganar con el paso del tiempo a una por día, lo que generaría una limitante considerable en la cantidad de intentos disponibles para jugar. El segundo sistema, que fue ideado como una solución para ese problema, otorgaba 5 energías por cada asistencia a prácticas deportivas.

Como resultado de la interacción entre estos dos sistemas, los jugadores se verían recompensados con la posibilidad de jugar al minijuego cada vez que participen de sus actividades deportivas. De esta forma, se generaría un refuerzo positivo que los motive a la acción de asistir a sus prácticas.

## Trabajo con artista

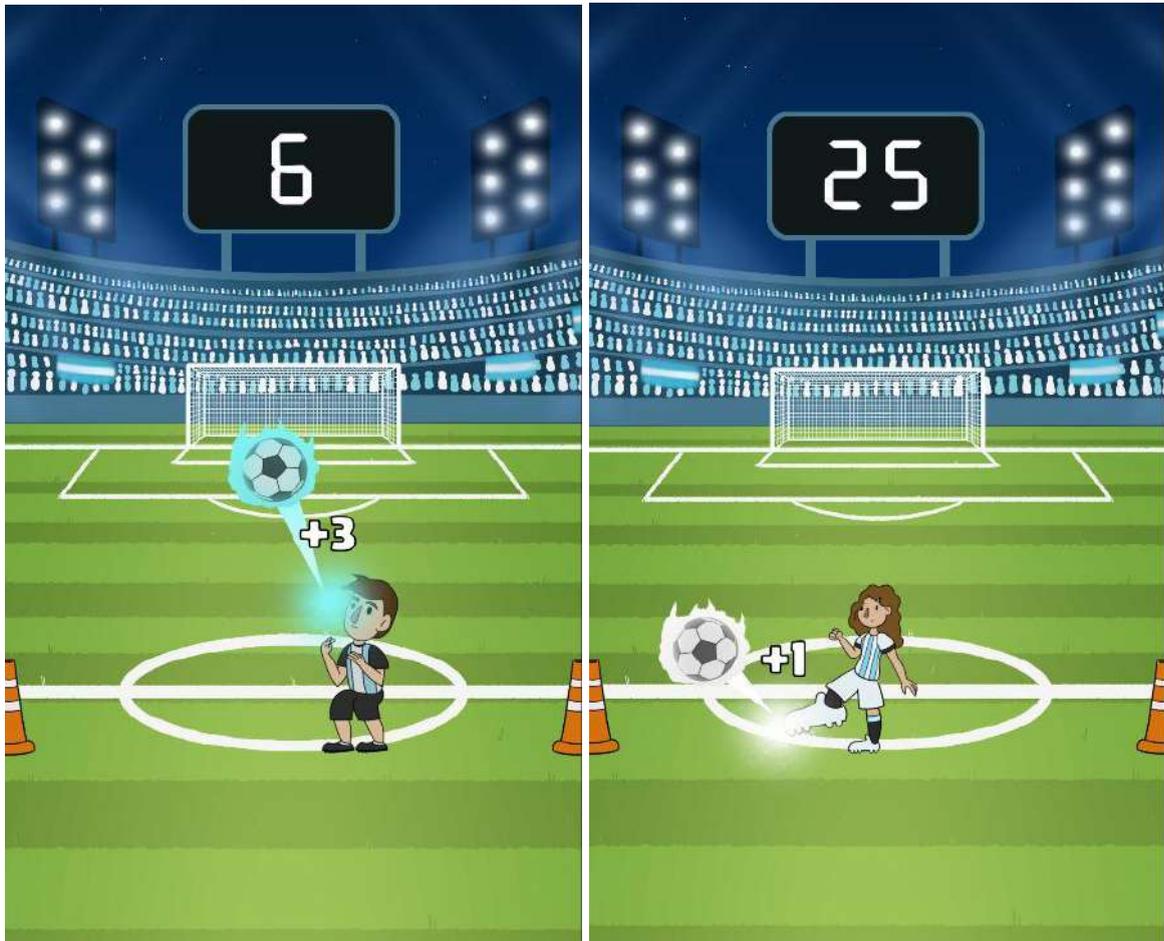
Durante la planificación del minijuego se identificó la necesidad de delegar la tarea del arte visual y contratar a un profesional con el objetivo de que la tarea sea realizada en colaboración con un experto que pueda colaborar con el equipo para obtener mejores resultados. Para ello, se creó un documento definiendo los assets de arte requeridos, referencias (imágenes o videos), tamaños y cantidad de fotogramas para las animaciones. Luego, el equipo contactó con algunos artistas y eventualmente seleccionó a una profesional con experiencia previa como animadora en videojuegos simples y en dibujos y cortos animados.

Finalmente se procedió a definir plazos de entrega y remuneración por su trabajo. Se coordinaron tres iteraciones de periodicidad de 10 días. La primera correspondía a

las animaciones del personaje masculino, la segunda al personaje femenino y la tercera al resto del arte solicitado. Esta última iteración contenía arte para las insignias, estilo de los textos, elementos del escenario de juego, y efectos visuales.

**Figura 13**

*Personajes del juego en escena de gameplay.*



*Nota. El tablero representa el puntaje actual y el +N representa la cantidad de puntos obtenida en ese golpe.*

**Figura 14**

*Ventana de insignias.*



*Nota. La sección superior corresponde a insignias obtenidas y la inferior a las que hace falta obtener.*

La dinámica del trabajo fue fluida porque la artista tenía mucha información relevante sobre el proyecto en la documentación de referencias que le fue entregada por el equipo. Gracias a esto, solo fue necesario comunicarse para evacuar dudas muy puntuales o dar feedback de algún dibujo o animación.

## Seguridad y protección de la información

Para la creación y autenticación de cuentas de usuarios se decidió utilizar la herramienta Firebase Auth. Esto tiene las siguientes implicaciones:

- **Se minimiza la posibilidad de sufrir ataques**, tales como fuerza bruta, o acceso indebido a bases de datos, pues se utiliza una herramienta que cumple con estándares de privacidad y seguridad ISO y SOC, envío de datos HTTPS, entre otros.\*

\*Ver privacidad y seguridad en Firebase - <https://firebase.google.com/support/privacy>

- **En caso de comprometerse la seguridad, los datos de autenticación están protegidos**, pues ni siquiera la cuenta de Google que es dueña de todo el proyecto de Firebase tiene acceso a la información que los usuarios intercambian con Firebase Auth.
- Se crean **reglas de seguridad en Firestore** que aumentan la seguridad para los llamados vía API, de forma tal que en caso de que se encuentre una vulnerabilidad del lado del cliente, el daño sea mínimo.

## Costos del producto

### Elaboración

- Costos de licencias: U\$S 0 (No se utilizan licencias pagas)
- Arte del minijuego: U\$S 100
- Trabajo de Ingeniería: U\$S 30.420 (1014 horas a U\$S 30 por hora)

### Despliegue

- Costos fijos:
  - U\$S 25 - Verificación de cuenta de desarrollador de Google Play.
- Costos variables:
  - U\$S 0 - Hasta alcanzar la cuota gratuita de Firebase. Una vez alcanzada la cuota máxima (ver precios de referencia Firebase\*).

\* <https://cloud.google.com/identity-platform/pricing>

### Total

- Costo fijo total: U\$S 30.545
- Costo variable total: U\$S 0\*

\* (hasta exceder cuota gratuita de Firebase)

## TDD y DI

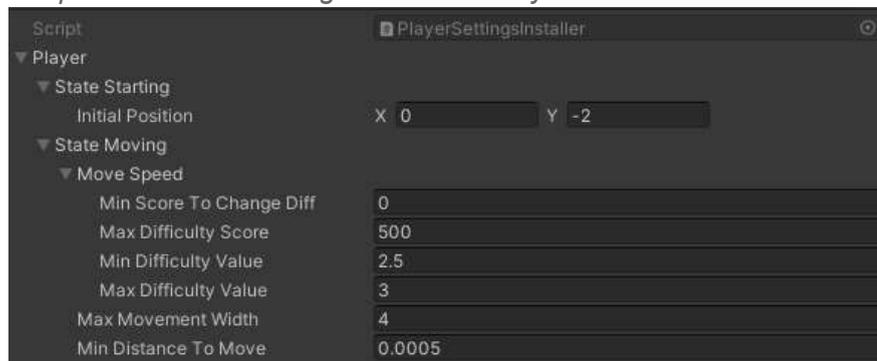
Para el desarrollo del videojuego se decidió aplicar TDD y DI para dar mayor mantenibilidad a la solución y ganar conocimiento sobre estas técnicas.

En cuanto a TDD, los casos de prueba generados servirían para testear funcionalidad a lo largo del desarrollo y enfocarse en generar el código vinculado a la superación de esos tests. Además la utilización de esta técnica sentaría un precedente que podría funcionar como base para generar test suites de otros minijuegos para una posible extensión futura del proyecto.

DI fue utilizada con el objetivo de reducir el acoplamiento entre clases y, al mismo tiempo, entregar las referencias necesarias dentro del ámbito de cada una de ellas. También cabe destacar que la utilización de la herramienta Zenject como solución de DI permite cambiar fácilmente la configuración de diversos parámetros de juego, que incluso pueden ser referenciados en los tests. Además se integra de manera sencilla con el entorno de pruebas de Unity, por lo tanto resultó ventajoso para el desarrollo utilizando la técnica TDD.

**Figura 15**

*Ejemplo de parámetros de configuración del Player.*

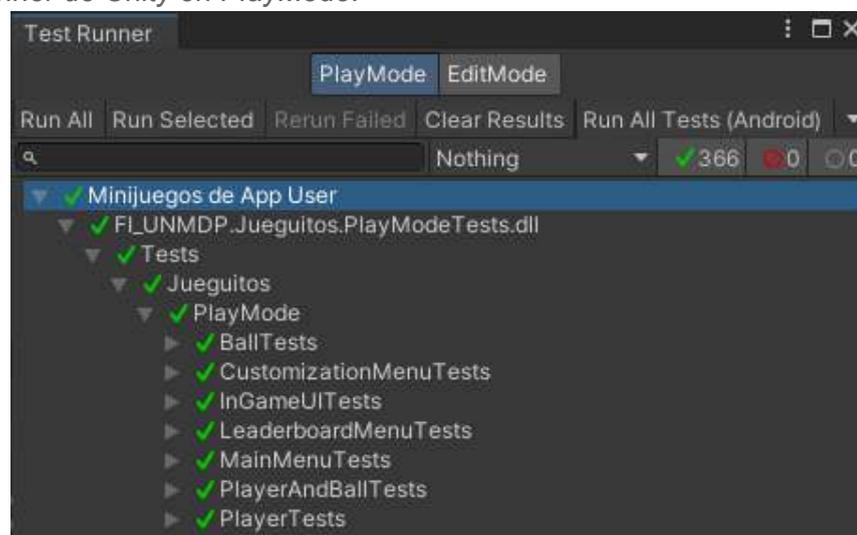


*Nota.* Estos parámetros pueden ser fácilmente modificados sin necesidad de cambiar el código ni los tests.

En Unity existen 2 tipos de tests, estos son PlayMode y EditMode. Los primeros requieren que se pongan en funcionamiento escenas de juego con el motor de videojuegos y los segundos ejecutan código sin usar funcionalidad de Unity. En este proyecto se generaron ambos tipos de tests pero la amplia mayoría son de PlayMode. El objetivo de estas pruebas fue generar validaciones automáticas sobre el gameplay y los distintos menús del juego. Las primeras se orientaron a comprobaciones sobre los estados de la pelota y el jugador, los tipos de golpes que se pueden generar y cómo progresa o termina una partida. Las segundas ponen a prueba el contenido que se muestra en los distintos menús del juego y verifican que las transiciones entre ellos se realicen correctamente.

**Figura 16**

*Test Runner de Unity en PlayMode.*



## Testing

Para las aplicaciones móviles, se prevé dividir el testing en dos etapas: El testing interno y el test de aceptación de usuario (UAT). El testing interno será realizado por el equipo de ingeniería, y algunos colaboradores externos, quienes probarán la funcionalidad de las aplicaciones en búsqueda de errores relacionados al correcto funcionamiento del sistema, con el objetivo de asegurar la calidad.

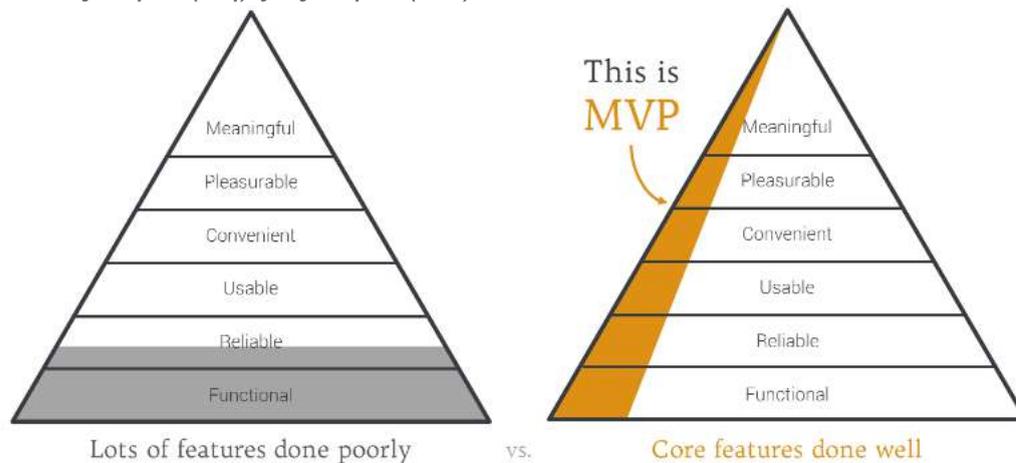
Por otro lado, se prevé realizar un despliegue y un UAT donde se pueda realizar, con un conjunto selecto de personas (clubes y deportistas), un **beta testing** o “prueba de versión beta”, donde se verifique si efectivamente el sistema cumple con las necesidades de los usuarios, y se colecte feedback para futuras iteraciones. Por errores de estimación que fueron alargando la duración del proyecto se decidió dejar al UAT fuera de su alcance pero como un paso a seguir fundamental para trabajos a futuro porque es un input de vital importancia para futuras iteraciones.

## Características de innovación

Como ya se ha visto en el marco teórico, el objetivo de un MVP no es ser un producto que se limite a tener la mínima funcionalidad viable y desatender todo lo que no sea estrictamente necesario para el funcionamiento elemental del sistema. En realidad, un MVP debería hacer relativamente pocas cosas en lo funcional (lo mínimo viable) pero hacerlas **bien**. Resulta de vital importancia dejar sentadas las bases de cuáles son los estándares de calidad del producto.

**Figura 17**

*Contraejemplo (izq) y ejemplo (der) de un MVP*



Es por esto que el producto, por más de ser un MVP, tiene una gran cantidad de características innovadoras y un alto grado de atención al detalle, tanto en la funcionalidad, el diseño, la usabilidad, como en todos aspectos del producto.

Por motivos prácticos no se listarán todos ellos, pero algunos de los más importantes se detallarán a continuación.

## GitHub Copilot

A principios del año 2022, GitHub Copilot comenzó a aceptar paulatinamente usuarios a su beta cerrada. Eventualmente el equipo pudo acceder a ésta y fue utilizada para el desarrollo. Es decir, el código del presente proyecto fue parcialmente escrito por inteligencia artificial.

La experiencia utilizando esta herramienta fue interesante, sobre todo porque se empezó a utilizar con un lenguaje sobre el que se llevaba trabajando ya cierto tiempo y se había logrado ya cierto conocimiento. Algunos de los resultados obtenidos utilizando esta herramienta son:

- Muchas veces el copiloto completaba la línea que se estaba escribiendo, con un porcentaje de acierto impresionantemente alto.
- Algunas veces, el copiloto era capaz de ofrecer un “esqueleto” básico en función de una línea comentada, que servía como punto de partida para continuar programando.

- En ocasiones incluso ha dado sugerencias que resultaron en nuevo conocimiento adquirido por el equipo. Principalmente con el uso de algunas de las incontables funcionalidades que Flutter ya posee de fábrica.

## Temas dinámicos y colores personalizables

En el mundo del deporte, principalmente los deportes en equipo y los clubes, los colores toman una relevancia más alta de lo que podría parecer a simple vista. Por ejemplo, al consultar por su color favorito a un hincha del Club Atlético Independiente (apodo “El Rojo” por su color principal), es casi seguro que éste conteste “rojo”. Esto es porque el color es algo representativo de los clubes y de sus aficionados.

Es por esto que se decidió apostar por incluir una característica que podría parecer un detalle menor, pero que el equipo realmente cree que es una característica innovadora, muy interesante y que puede generar un gran impacto positivo en los usuarios, tanto clubes como deportistas. Esta característica es “**temas dinámicos**”.

Los temas (en inglés, themes) son un conjunto de valores numéricos para tono, matiz, brillo y contraste; que en conjunto definen una paleta de colores. Entonces, los temas dinámicos son una característica que permite al usuario elegir entre un conjunto definido de temas con un color predominante y **además** elegir si el tema será claro u oscuro (lo cual puede cumplir fines estéticos o alinearse con el color principal), resultando en un total de 16 temas dinámicos a disposición del usuario.

**Figura 18**

*Algunos de los temas de colores del sistema (recorte de la mitad superior de la pantalla)*



## Diseño inteligente

En la pantalla de métricas se desarrolló un algoritmo que genera dinámicamente una paleta de colores que se adapta tanto a cualquier theme del usuario como a cualquier cantidad de elementos a mostrar. Esto es gracias a que el paquete Flutter Charts, de Google, permite definir una “color function” para definir mediante una función el color  $i$ -ésimo de un gráfico a renderizar.

Utilizando interpolación lineal para hallar valores intermedios entre dos colores, se subdivide a la paleta en “ $i$ ” intervalos de tamaño “ $i/n$ ” (siendo  $n$  el total de elementos a graficar) y se genera una paleta de colores distinta pero coherente.

```
// Versión simplificada del algoritmo
colorFn: (int i) => Color(
    Color.lerp( // Interpolación linear entre 2 colores
        colorScheme.primary, // Color 1: Primario theme
        colorScheme.onSurface, // Color 2: Color contraste
        i / itemCount // Porcentaje interp item iésimo
    ),
),
```

Ejemplos:

### Figura 19

*Ejemplos de métricas con paleta de colores dinámica en función del theme*



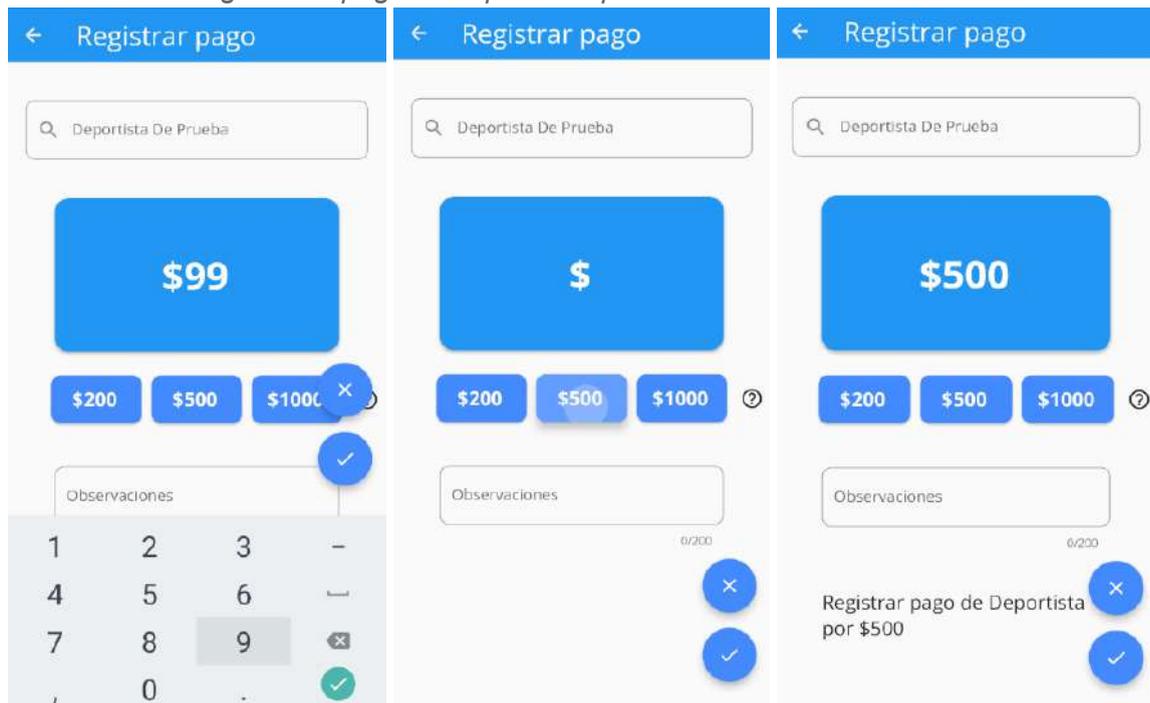
### Importes rápidos

Otro interesante ejemplo de atención al detalle e innovación, es la característica de los “importes rápidos” en la aplicación para clubes. Muchas veces las instituciones deportivas manejan un único importe de cuota para sus socios (principalmente los clubes que operan con un sólo deporte). Algunas otras veces se da el caso que hay dos o tres importes que son los más populares.

Ante esto se identifica la oportunidad de crear un conjunto de importes rápidos, **configurables** por el usuario (de forma tal que pueda adaptarse a la evolución de los precios), a fin de eliminar la necesidad de ingresar el importe vía teclado, agilizando el proceso.

**Figura 20.1**

*Pantalla de registro de pagos e importes rápidos*



*Nota: A la izquierda, registro de pago con input por teclado. Al centro y la derecha, registro de pago con importe rápido.*

## Mejora de procesos

Otro de los enfoques importantes para el diseño y la elaboración del producto fue realizar dichas tareas teniendo en cuenta la **mejora de procesos**, principalmente en las herramientas de gestión de clubes.

Al tener que crear interfaces de usuario que digitalicen algunos de los procesos manuales realizados por los usuarios, se abre la puerta a la posibilidad de utilizar características de innovación que permitan reducir tiempos, costos, margen de error, entre otros.

El equipo identificó una potencial mejora que podría disminuir notablemente el tiempo necesario para dar de alta un nuevo socio del club: La inscripción por QR. Esta se basa en aprovechar la oportunidad que se presenta al tener usuarios registrados en

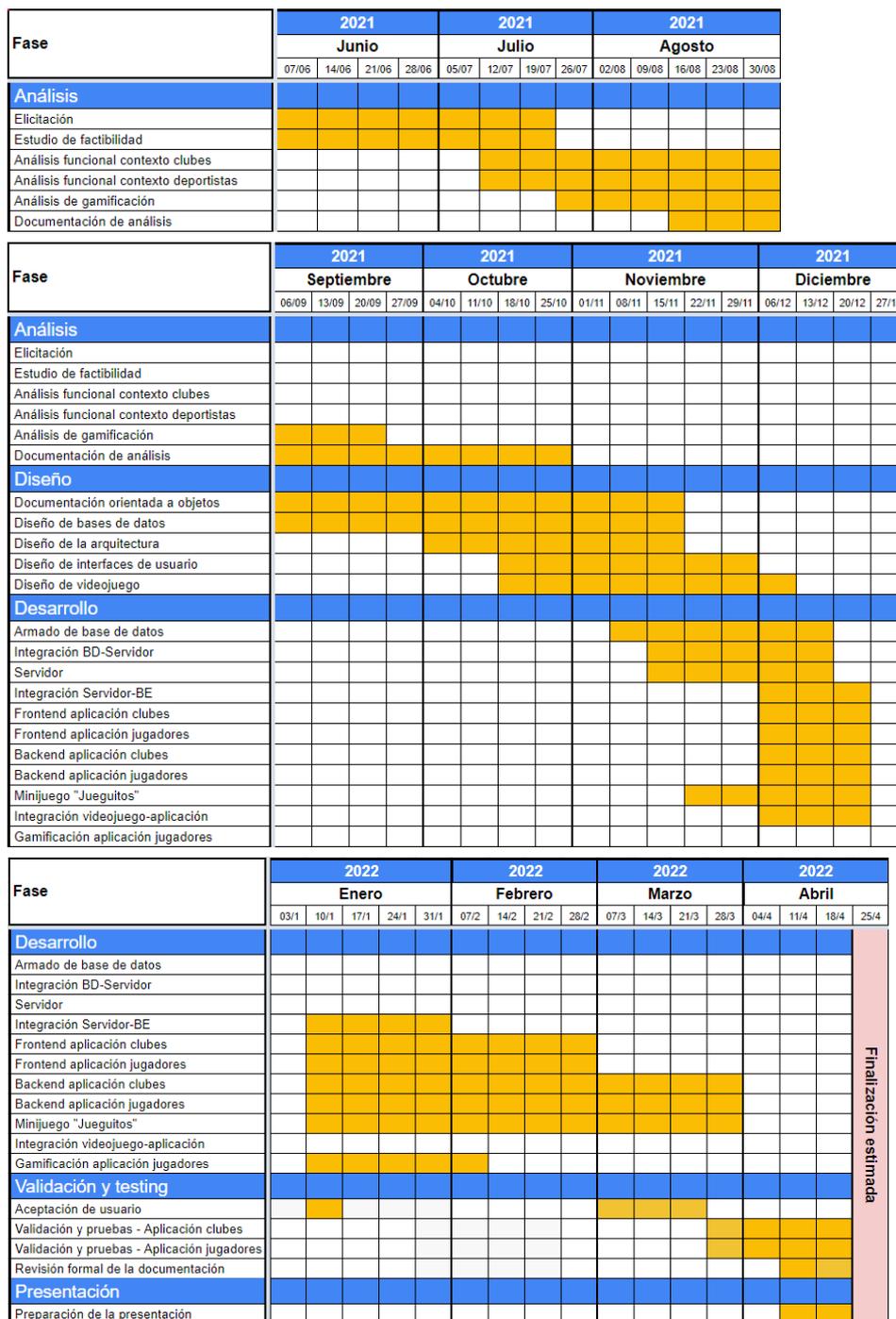
la app de deportistas, que ya tienen un perfil cargado con sus datos. De esta forma cuando se dirija de forma presencial al club a fin de darse de alta (por el momento se excluye la posibilidad de realizar altas vía app) un usuario podrá **generar un código QR** que, al ser escaneado desde la aplicación de clubes, rellenará automáticamente su formulario de inscripción.

Además de la mencionada anteriormente, el equipo cree que este tipo de plataforma abre lugar a identificar más oportunidades de mejora de procesos gracias al feedback de los usuarios en futuras iteraciones del desarrollo.

# Memoria del proyecto

## Planificación y estimación

Inicialmente se realizó una estimación de tiempos, donde se dividió el desarrollo en fases, se estimó una cantidad de horas para cada fase, se distribuyó en el tiempo considerando una dedicación de ocho horas semanales y se obtuvo el tiempo de desarrollo estimado en semanas. Esto fue plasmado el siguiente diagrama de Gantt:



Siendo el total de la estimación:

- 752 horas en total (376 por integrante).
- 16 horas de dedicación semanal (8 por integrante).
- 47 semanas de duración total del proyecto.\*

\*La duración total estimada del proyecto surge como la razón entre las horas necesarias estimadas y el tiempo semanal dedicado.

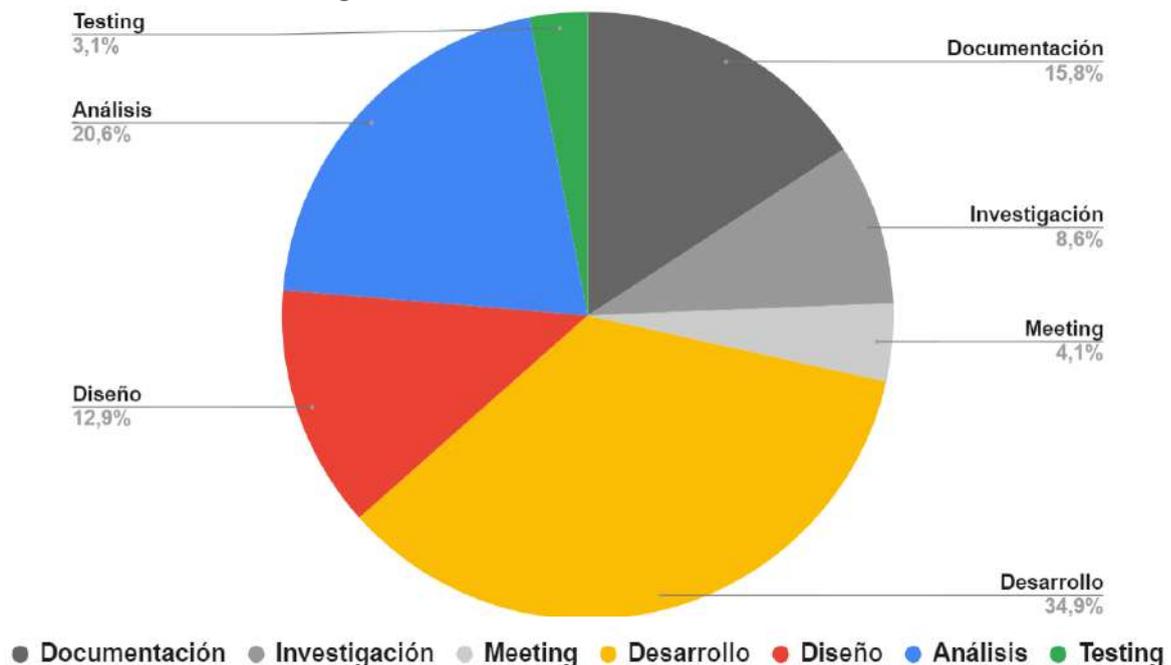
## Tiempos reales

A continuación se muestra el resultado y la comparación con el valor estimado.

- Fecha fin del proyecto: 31/10/2022
- Total horas: 1014 horas (+34.8%)
- Dedicación semanal promedio por integrante: 6.85 horas (-14.4%)
- Total duración: 74 semanas (+57.4%)

**Figura 21**

*Distribución de horas según tareas*



*Nota: Las tareas en escala de grises representan tareas principalmente relacionadas al proyecto. Las demás, representan tareas relacionadas al producto. "Documentación" representa tareas de documentación del proyecto, entre las cuales se incluye la elaboración del presente documento.*

## Comparación con estimación

Separado en análisis, diseño, desarrollo, testing, y la combinación de los demás.

	Análisis	Diseño	Desarrollo	Testing	Elaboración de la presentación, elaboración informe, investigación, meetings
<b>Estimado</b>	250h	162h	250h	56h	34h
<b>Real</b>	209h	131h	354h	31h	289h
<b>Diferencia</b>	-41h	-31h	+104h	-25h	+255h
<b>Diferencia %</b>	-16.4%	-19.1%	+41.6%	-44.6%	+750%

\* Data actualizada al 31/10/2022.

\*\* La columna "Otros, combinados" incluye: Elaboración presentación, elaboración informe, investigación, meetings.

## Motivos de la desviación

- Falta de experiencia en la estimación de un proyecto de la envergadura del presente.
- Sobreestimación de la capacidad de dedicación semanal de horas del equipo debido a dos factores principales:
  - Falta de contemplación de contingencias tales como vacaciones o enfermedades.
  - Menor disponibilidad horaria de la estimada debido al tiempo consumido por la carrera profesional de los miembros del proyecto.
- Subestimación del tiempo necesario para investigación de tecnologías, meetings, y la elaboración del presente informe.
- Tiempos muertos, a la espera de información y/o reuniones, (este caso particular sólo en etapas tempranas del proyecto).

Además, es posible visualizar cómo los excesos de tiempo en "otros" y "desarrollo" causaron

## Resultado / Entregables

Este proyecto concluye habiendo producido los siguientes entregables:

- El presente informe.

- El producto de software, con su respectivo código fuente.
- Documentación de análisis y diseño. Esta incluye tanto lo presente en el MVP como funcionalidad para futuras iteraciones de desarrollo.
- La cuenta de Google dueña del proyecto Firebase.

## Retrospectiva de objetivos del proyecto

Se analizan en retrospectiva los objetivos del proyecto y se evalúa si se cumplió con lo esperado. No puede realizarse una retrospectiva de satisfacción de usuarios por no encontrarse desplegado el sistema aún.

- **Modernizar y mejorar los circuitos administrativos para que los deportistas de la región encuentren entidades deportivas que se adapten a sus intereses y necesidades:** fue creado un mapa dinámico de clubes con filtros que se adapten a las necesidades de esas personas y, de esa forma, permitir encontrar entidades deportivas cercanas a la persona donde se practiquen los deportes que deseen bajo algunas condiciones parametrizables (práctica techada, división femenina, etc). El equipo está conforme con el grado de cumplimiento de este objetivo.
- **Brindar herramientas de gestión administrativa para los clubes:** El MVP incluye herramientas tales como gestión de socios, control de asistencias, registro de pagos, gestión de prácticas del deporte e inscripciones en múltiples deportes, métricas, entre otros.

Además, el diseño del sistema, y por lo tanto futuras iteraciones del producto, incluirán nuevas herramientas tales como almacenamiento de planillas médicas, gestión de subdivisiones deportivas, cronograma de horarios público integrado con las prácticas y divisiones, gestión multiusuario con un esquema de permisos y roles, entre otros.

En general, el equipo está conforme con el grado de cumplimiento de este objetivo, aunque hubiera preferido que más características entren en el MVP. Sin embargo esto no fue posible por motivos de tiempo.

- **Contribuir a la inclusión de personas alejadas de la práctica del deporte (Incluyendo niños y adolescentes en situación de vulnerabilidad, mujeres,**

**personas con discapacidad, entre otros):** fueron incluidos filtros para clubes con divisiones femeninas y para personas con discapacidad. La solución fue diseñada para dispositivos móviles Android que es la plataforma más accesible. Se agregó un aspecto de personaje femenino al videojuego para que más personas estén representadas dentro de la experiencia.

En general, el equipo podría estar más conforme con el grado de cumplimiento de este objetivo, pues en algún momento existió la posibilidad de tener aporte interdisciplinario en materia de inclusión, de manera tal que existan bases teóricas más sólidas que permitan aplicar más y mejores elementos de inclusión en el sistema.

- **Fomentar la participación de los deportistas en las prácticas deportivas:** se diseñó e implementó un videojuego que utiliza técnicas de gamificación en conjunto con la app de deportistas. La cantidad de intentos (energía) disponible para jugar fue vinculado a las asistencias a las prácticas deportivas. De esta forma, generando una motivación extra para la participación en el deporte y fomentando tanto a la participación en sí como a su regularidad. El equipo está conforme con el grado de cumplimiento de este objetivo.
- **Incluir métricas para el análisis del estado actual y la evolución de indicadores cuantificables para las entidades deportivas:** Se sentaron las bases, y se incluyó en el MVP una métrica. Se prevé que en iteraciones futuras agregar más métricas sea menos complejo pues ya existe una base a tomar de prueba, parametrizable, donde en caso de necesitar agregar nuevas sería tan sencillo como realizar una copia de la existente y modificar los parámetros. El equipo considera que este objetivo se ha cumplido exitosamente, aunque desearía haber incluido un mayor número de métricas en el MVP si el tiempo lo hubiera permitido.

## Desafíos principales

### Formalización de requerimientos

Si bien habían lineamientos a seguir, y algunos requerimientos definidos (principalmente las herramientas de gestión de clubes); muchos otros de los objetivos

del proyecto estaban planteados solamente como objetivos y no existían requisitos que permitan lograr dichos objetivos, por lo que fue necesario invertir una cantidad de tiempo en análisis en pos de generar requerimientos coherentes con los objetivos del proyecto.

Para la tarea de definir el conjunto de requerimientos fue necesario informarse sobre dos proyectos de la Facultad de Ingeniería de la UNMdP. A partir de esa información se creó un marco de requerimientos que buscaban atacar necesidades de los clubes deportivos y deportistas. Se condujeron entrevistas con personas involucradas de distinta forma, a fin de familiarizarse más con el contexto. Algunas de estas personas incluyen dirigentes de la liga marplatense de fútbol, administrativos y profesores de clubes tanto institucionales como barriales, partícipes de los proyectos universitarios asociados ya mencionados, entre otros.

Esta tarea supuso un gran desafío porque supone una dimensión de complejidad más sobre lo que sería un proyecto con requerimientos estrictamente definidos. Ser los encargados de definir es una ventaja, pero también una gran responsabilidad. Es por ello que se decidió optar por utilizar la estrategia Lean que permite realizar un desarrollo iterativo, partiendo de un MVP y mejorándolo iterativamente en función del feedback, y de esta manera reduciendo el impacto que tienen los cambios y/o los errores de definición de los requerimientos, o aspectos similares.

Sin embargo, a pesar de la gran dificultad que todo esto supuso, la experiencia fue muy positiva por muchos motivos, como dar lugar a la creatividad y la innovación, aportando valor desde la ingeniería y la adquisición de múltiples experiencias nuevas.

### Limitar el producto de software a un MVP

Restringir el alcance del proyecto fue una tarea muy desafiante. Participar del proceso creativo hace que todo parezca un “requerimiento primario y esencial” y recortar características es una tarea particularmente difícil. Sin embargo, para alinearse con la filosofía de Lean Development era absolutamente necesario endurecer las políticas de admisión para el selecto grupo de las “características a ser implementadas”.

## Contribuir a solucionar un problema social desde el punto de vista de la tecnología

Durante la revisión de la documentación de los proyectos que sirvieron de base para este proyecto final se identificaron problemas sociales complejos y muy abarcativos. Por lo tanto se decidió enfocar la solución de parte de esa problemática desde el área informática. Fue desafiante la identificación de formas creativas e innovadoras en las que se podría contribuir desde el desarrollo de un producto de software.

## El riesgo de la dependencia en un proyecto interdisciplinario

En las etapas primigenias del proyecto, éste tenía el potencial de ser un trabajo interdisciplinario, involucrando otras facultades y especialidades de la Universidad Nacional de Mar del Plata, como ciencias de la salud, antropología, entre otras. Esto podría nutrir tanto al equipo como al proyecto de gran cantidad de conocimiento de otras áreas, con requerimientos y colaboración en el proceso creativo y las ideas de innovación.

Desafortunadamente esto resultó no ser así. Sin embargo, a pesar de esa situación, el equipo nunca se puso a sí mismo en la posición de depender del avance de los demás proyectos y gestionar la autonomía de este proyecto fue una prioridad desde el día inicial.

## Aprendizaje

- El desarrollo de un proyecto de envergadura real.
- Trabajo sobre una amplia y variada cantidad de tareas, más allá de solamente el desarrollo.
- Tecnologías multiplataforma para la creación de aplicaciones de propósito general y videojuegos.
- Aprendizaje de técnicas para el desarrollo de software que resultan útiles (BaaS, DI, TDD, etc).
- Trabajo interdisciplinario con artista.
- Generación de sistemas para tablas de puntajes e insignias con persistencia remota.

- Puesta en práctica de principios fundamentales de UX/UI.

## Retrospectiva: Decisiones

A continuación se realizará un análisis objetivo de algunas de las decisiones más importantes tomadas en el proyecto, que eventualmente se convirtieron tanto en aciertos como desaciertos.

### Utilizar BaaS a fin de reducir el tiempo de desarrollo

El tiempo dedicado a todas las tareas de investigación, configuración, puesta en marcha y uso de Firebase, ocupó una cantidad de horas mucho menor a lo que el equipo estimó que podría haber tomado utilizar una arquitectura basada en APIs, servidor y base de datos. Por lo tanto, se considera esta decisión como acertada.

### Estimación optimista

Al comparar los tiempos estimados contra los resultantes, se evidencia cierta desviación sobre la estimación inicial. Parte de este error se debe a que la precisión de estimaciones depende de la experiencia, que el equipo no tenía. Sin embargo, además del factor de la experiencia, los autores consideran que no fueron lo suficientemente conservadores al momento de realizar la estimación. Se considera a este optimismo como un desacierto.

### Independizar al proyecto de los proyectos asociados

Aumentar la autonomía del proyecto permitió que éste pueda desarrollarse independientemente del avance de otros proyectos. Esta independencia trajo consigo ventajas y desventajas. Por un lado se evitó el riesgo de bloqueos por dependencia de resultados de factores externos pero también generó la falta de uno de los referentes funcionales del proyecto, quien además se esperaba que colabore en el objetivo de fomentar la inclusión en el deporte. En definitiva consideramos esta situación favorable respecto al tiempo de desarrollo pero también negativa por implicar un menor grado de cumplimiento del objetivo relacionado a la inclusión de deportistas que posee el proyecto.

## **Dedicar mucho tiempo a documentación formal de análisis y diseño**

Cuando se definió el alcance del proyecto, el equipo optó por tomar una estrategia para facilitar su extensibilidad: Intentar generar una cierta cantidad de documentación formal de análisis y diseño. El objetivo de esta medida era proveer una base sólida sobre la cual eventualmente colaboradores futuros puedan extender el producto. Sin embargo, conforme el proyecto fue avanzando, el equipo se encontró con que dicha documentación consumía una cantidad considerable de tiempo al momento de ser creada.

Además, conforme el entendimiento del equipo sobre Lean Development y Agile Development fue avanzando, se comprendió que esta estrategia no es del todo compatible con lo que estas técnicas recomiendan. Analizándolo en retrospectiva, aunque se lo considera como un acierto por ser algo imprescindible para tener bases sólidas en futuras iteraciones, el equipo considera que la ejecución podría haber sido mejor.

## **Selección de tecnologías**

Las decisiones relacionadas a la selección de tecnologías fueron tomadas en base a la investigación plasmada anteriormente en este informe. Sin embargo, al ser muchas de ellas tecnologías nuevas para el equipo, existe la posibilidad de cometer errores por falta de conocimiento.

Trabajar con una selección de tecnologías relativamente modernas demostró ser satisfactorio en muchos aspectos, tales como la capacidad de utilizar dicha modernidad para reflejarla en el producto, la forma simple en la que se integran entre ellas, la gran cantidad de características novedosas y convenientes, entre otros. Por supuesto, hubo también desventajas. La principal de ellas fue que las tecnologías relativamente nuevas suelen cambiar con mayor frecuencia, y en ocasiones mantenerse al día requería cierto trabajo extra (Un caso puntual fue la versión 3.0 de Flutter y su migración a “null-safe-code” que requirió realizar una cantidad considerable de ajustes).

Para resumir, las ventajas fueron mayores y más ponderantes que las desventajas. Es por ello que el equipo considera que la selección de tecnología ha sido una decisión acertada.

### **Tercerización de la tarea del arte**

Al momento de tomar esta decisión se supo que podía llegar a ser ligeramente riesgosa. El principal riesgo al que el equipo se enfrentaba era encontrarse con un artista poco profesional que ocasionara pérdidas de tiempo y dinero. Por lo tanto el proceso de selección fue meticuloso, y se acordaron los plazos de entrega desde un principio. Afortunadamente esta colaboración fue satisfactoria y el resultado obtenido gratificante. El equipo considera esta decisión como acertada.

## Trabajos a futuro

### Evolución del proyecto

Como se ha mencionado anteriormente, el presente proyecto involucra la primera iteración del desarrollo del producto, y la elaboración de un MVP. Se prevé que en el futuro cercano este MVP sea desplegado y utilizado para realizar UAT, y será la base de la siguiente iteración, que además de implementar nueva funcionalidad, deberá refinar la iteración anterior según el feedback obtenido durante el UAT. El UAT y las futuras iteraciones del desarrollo del producto quedan fuera del alcance del presente proyecto.

Algunos aspectos a considerar para la continuación del proyecto en futuras iteraciones son:

- **Definición de gestor de verificaciones:** Como fue definido en el diseño del sistema, para que un club sea visible por los usuarios debe estar **verificado**. Surge entonces la necesidad de asignar responsables de validar las cuentas de clubes. Estas personas se encargarían de confirmar que un club registrado en la aplicación existe realmente y su información es fidedigna.
- **Participación de diversos clubes y deportistas en el UAT:** Actualmente los test de aceptación están contemplados a realizarse con un solo club. Sin embargo es necesario obtener feedback amplio y de múltiples fuentes (clubes con otros deportes y realidades) para tener una idea más consensuada sobre el camino a seguir.
- **Buscar asesoramiento especializado:** En inclusión en la práctica del deporte: Esto permitiría definir y validar requerimientos adicionales en los sistemas que contribuyan al objetivo de impulsar la inclusión de personas alejadas de la práctica del deporte.
- **Definir documentación legal para publicación:** Es necesario generar una documentación de políticas de privacidad para cada una de las tres aplicaciones para poder publicarlas en las principales tiendas de aplicaciones. Además se debe especificar para cada una de ellas qué datos del usuario utilizamos, su finalidad, cuáles se envían a servicios de terceros y cuáles se almacenan en nuestro sistema.

- **Buscar impulsar el proyecto:** El proyecto podría valerse de diversos socios estratégicos que permitan aumentar difusión y recursos para continuar con el proyecto y lograr alcanzar su máximo potencial.

## Funcionalidad a implementar en futuras iteraciones

El producto de software desarrollado, al ser un MVP, tiene un enorme potencial de ampliación a futuro. El equipo concluirá la primera iteración de desarrollo de este producto, pero hereda sugerencias para las siguientes iteraciones.

Vale la pena mencionar que **la arquitectura del sistema está preparada** para la eventual implementación de todas las características que forman parte del diseño de la aplicación, y por cómo se concibe Lean Development y la metodología del desarrollo de un MVP, se concibe el sistema teniendo en consideración su escalabilidad.

Algunas de las funcionalidades recomendadas fueron parte del análisis y diseño, y algunas surgen más de modo conceptual, y potencialmente para iteraciones más lejanas del desarrollo. Se proponen los siguientes conceptos e ideas a desarrollar para extensibilidad futura del proyecto:

- **Validación de cuentas de clubes.** Implementar un sistema adicional para gestionar la verificación y validación de clubes de forma más amigable que accediendo directamente a la base de datos. Debería incluir una lista de solicitudes pendientes de clubes, con datos de cada uno, que puedan ser aceptadas o rechazadas.
- **Validación del email de las cuentas.** Validar el email antes de proceder a utilizar la aplicación permitiría garantizar que pertenece a la persona que creó la cuenta. De esta forma impediría el uso de emails inexistentes y dificultará registrarse con una dirección de correo de terceros.
- **Cambio de contraseña de las cuentas.** Las aplicaciones deberían permitir que los usuarios cambien su contraseña por medio de un correo al email que utilizan para hacer login. Es importante asegurar que el mail sea validado al momento de crear las cuentas para que este mecanismo de cambio de credenciales pueda ser utilizado.

- **Ampliación de plataformas.** Las aplicaciones deberían poder utilizarse también en sistema operativo iOS y vía navegador Web. Los frameworks Flutter y Unity hacen que exportar a nuevas plataformas sea una tarea posible y no laboriosa. El trabajo principal sería modificar el diseño de las interfaces para que se ajusten a resoluciones de pantalla web y gestionar asuntos relacionados a la publicación de la aplicación en la tienda de iOS.
- **Soporte multiusuario** en aplicación para clubes. Actualmente un club posee una única cuenta de usuario, con permiso de acceso global a la aplicación. Para que el uso de la aplicación sea viable en clubes medianos y grandes es necesario dar la posibilidad de crear múltiples cuentas asociadas a un único club. Cada una de las cuales podrá tener un conjunto de permisos que podrían ser limitados según roles y determinados por la cuenta dueña del club.
- **Auditabilidad en aplicación para clubes.** Otra de las características de gran importancia para el uso de la aplicación en clubes de tamaños medianos y grandes es la auditabilidad. El sistema debería poder generar registros de auditoría según las acciones del usuario y tener un listado consultable de las acciones que cambiaron el estado del sistema (registros de pagos, tomas de asistencia, etc).
- **Soporte multiclub en aplicación para clubes.** Esta característica sería una ampliación al soporte multiusuario. Este permite el uso de múltiples cuentas con perfiles de permisos específicos, pero requiere que una persona que trabaja en más de un club intercambie cuentas constantemente. Para solucionar este problema, la aplicación debería estar abierta a la posibilidad de que el usuario pertenezca a más de un club (utilizando una única cuenta de usuario). Esto permitiría que varios clubes sean gestionados por la misma persona, o, que un profesor pueda utilizar su cuenta personal para trabajar en todos los clubes que necesite y no tenga que utilizar una cuenta para cada club.
- **Temas específicos según club para la aplicación de clubes.** La aplicación debería permitir elegir temas de colores específicos para cada club (considerando que el soporte multi club ya fue implementado), de manera tal que el usuario pueda elegir para cada uno de los clubes de los que es miembro, qué tema de

colores quiere. Esto no sólo tiene fines estéticos, sino que provee una manera mucho más clara de saber en qué club uno se encuentra, facilitando la navegación y disminuyendo la posibilidad de error.

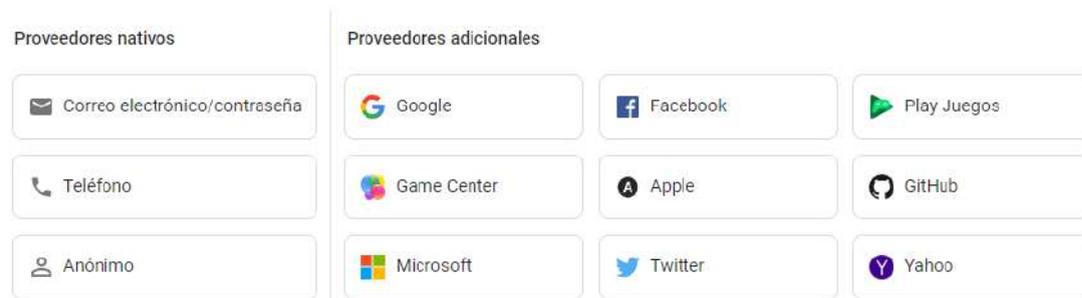
- **Sistema de mensajería para deportistas y clubes.** Agregar mensajería instantánea para facilitar la comunicación entre clubes y deportistas, y mutua entre deportistas.
- **Notificaciones del club.** La aplicación de deportistas debería poder recibir notificaciones del club con información relevante. Estas deberían ser visibles sin importar si la aplicación está en uso en el momento en el que se reciben. Se proponen los siguientes eventos para notificar a un deportista:
  - cancelaciones de entrenamientos;
  - cambios en horarios de entrenamiento;
  - eliminaciones de cancelaciones de entrenamientos;
  - nuevo mensaje recibido.
- **Diseño, implementación, testing e integración de minijuegos propuestos.** Los minijuegos propuestos están documentados utilizando MDA, GDD de una hoja, GDD de 10 hojas, casos de uso extendidos y diagramas de secuencia que se encuentran en un anexo externo al presente informe. También se detallan puntos de extensión extras como requerimientos opcionales en la documentación de análisis de requerimientos de la aplicación para deportistas.
- **Leaderboards en base a los clubes a los que pertenece el jugador.** Actualmente cada jugador compite contra todos los demás usuarios que jueguen minijuegos (leaderboard global) pero se podría mejorar si se permite filtrar la competencia a los compañeros de club. Adicionalmente se podría elegir entre leaderboards de algún club al que pertenezca el usuario, todos sus clubes a la vez o el leaderboard global.
- **Compra de bienes virtuales con puntos de minijuegos.** Los minijuegos deberían otorgar “monedas” al jugador luego de cada partida. La cantidad premiada debería depender del puntaje obtenido en una relación de una moneda cada cinco puntos. Las recompensas obtenidas deben poder ser canjeables por

aspectos de personalización alternativos para los personajes o los elementos de juego.

- **Premios por puesto en leaderboard de minijuego.** Monedas como recompensa de los primeros puestos del leaderboard mensual de cada club.
- **Integración del juego con características de los jugadores.** El juego podría seleccionar el skin femenino por defecto cuando quien juega es una mujer. También se podría diseñar una dificultad adaptada para personas con discapacidad (requiere voluntarios que lo jueguen durante el desarrollo).
- **Métodos anti trampa para los leaderboards.** El objetivo de esta medida es lograr resultados confiables para no arruinar la progresión en el juego. Se propone implementar un seguimiento periodico en el backend del tiempo de partida y puntaje actual del jugador para validar highscores. Si se detecta alguna incoherencia entre los valores que recibe o no hay datos enviados durante la partida, no debería considerar su puntaje como un candidato al leaderboard.
- **Ausencia justificada.** La aplicación de clubes debería poder registrar ausencias justificadas a los entrenamientos. Además ese registro debería otorgar 5 energías al deportista que tenga ese tipo de ausencia.
- **Integración con Google Maps Geocoding.** Geocoding API es un servicio parte de Google Maps Platform que realiza el proceso de conversión (bidireccional) entre coordenadas y direcciones legibles por humanos. El uso de esta API abriría las siguientes puertas:
  - Un deportista puede realizar búsquedas por su zona escribiendo su dirección en formato texto y el mapa podría realizar la conversión a coordenadas y realizar búsquedas por proximidad
  - Un club puede seleccionar la ubicación manualmente de su club y que la dirección se autocomplete con texto a partir de las coordenadas, o bien, escribir en texto su dirección y que el sistema automáticamente lo convierta a las coordenadas para fijar el pin en el mapa.

El principal impedimento para la inclusión de esta función en el sistema es que actualmente Geocoding es un servicio pago.

- **Ampliación de métodos de login.** Autenticación de acceso a la aplicación mediante Firebase Auth para permitir utilizar acceso con cuentas de Google, Facebook, entre otras.



- **Animaciones** para ventanas emergentes. El objetivo sería crear feedback adicional para que estas interrupciones sean más agradables para el usuario. Esto aplica a las apps de clubes, deportistas y videojuegos.
- **Guardar en caché** resultados de búsquedas para mejorar la velocidad del sistema. Se podría guardar de forma local un listado de todos los deportistas del club, de manera tal que cuando se los necesita, se muestra preliminarmente la versión cacheada mientras se realiza la consulta y luego se muestra el valor traído desde base. Esto también puede mejorar el rendimiento en situaciones de baja conectividad (Donde se puede continuar utilizando la app con datos cacheados). Sin embargo sería complejo para contemplar casos de modificaciones concurrentes.
- Dar de baja cuenta de club o deportista.
- Agregar más botones de ayuda para facilitar el uso.
- Ver programa de actividades.
- Gestionar una lista de clubes favoritos desde la aplicación de deportistas.
- Dar y recibir puntos de reconocimiento por compañerismo a nivel de cada club.
- Posibilidad de obtener premios y recompensas de los profesores.
- Ver el perfil de compañeros de club.

- Visualizar los pagos efectuados desde la app deportistas.
- Ver apartado de noticias con novedades sobre clubes inscriptos.
- Almacenar planillas médicas de deportistas.

## Bibliografía

1. Grupo de Estudios Antropológicos de la Universidad Nacional de Mar del Plata.  
<https://eamdq.com.ar/democracia-corporal/>
2. Comité Olímpico Internacional (1 de septiembre 2004). Carta olímpica.  
[http://www.coe.es/web/COEHOME.nsf/b8c1dabf8b650783c1256d560051ba4f/48781e452fd3070cc1256e23005a4454/\\$FILE/Charter\\_SPA\\_-\\_2004.pdf](http://www.coe.es/web/COEHOME.nsf/b8c1dabf8b650783c1256d560051ba4f/48781e452fd3070cc1256e23005a4454/$FILE/Charter_SPA_-_2004.pdf)
3. Puck Schipper (Junio 2019) Engaging vulnerable people in sports-based programs. <https://edepot.wur.nl/494859>
4. INDEC, Encuesta Permanente de Hogares, sobre accesos a internet:  
<https://www.indec.gov.ar/indec/web/Nivel3-Tema-4-26>
5. Informes técnicos históricos de uso de dispositivos:  
<https://www.indec.gov.ar/indec/web/Institucional-Indec-InformesTecnicos-71>
6. William C. Kirby (2 de agosto de 2016). The Real Reason Uber Is Giving Up in China. Harvard business review.  
<https://hbr.org/2016/08/the-real-reason-uber-is-giving-up-in-china>
7. INDEC, (17 de mayo de 2021). Tecnologías de información y comunicación.  
<https://www.indec.gov.ar/indec/web/Nivel4-Tema-4-26-71>
8. Ries, Eric. (2021) *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. 1era edición. Estados Unidos: Crown Pub Inc
9. VironIT. (26 de abril de 2022) *Advantages and Disadvantages of a Minimum Viable Product: A Decision-Making Guide for Startups*. Artículo en LinkedIn  
<https://www.linkedin.com/pulse/advantages-disadvantages-minimum-viable-product-decision-making-/>
10. Team, T. I. (19 de julio de 2021). *Gamification*. Investopedia.  
<https://www.investopedia.com/terms/g/gamification.asp>
11. Galessi, S. (27 de septiembre de 2018). *What is gamification: How points, badges, challenges, and charts boost the companies' annual...* Medium.

- <https://medium.com/nyc-design/what-is-gamification-how-points-badge-challenge-and-charts-boost-the-companies-annual-turnover-447568b92e61>
12. Hunicke, R., LeBlanc, M., & Zubek, R. (2004). (tech.). *MDA: A Formal Approach to Game Design and Game Research* (pp. 1–2). Evanston, Illinois: Northwestern University.
  13. Nuclino GmbH. (n.d.). *Game Design Document Template and examples*. Game Design Document Template.  
<https://www.nuclino.com/articles/game-design-document-template>
  14. Halpern, J. (10 de noviembre de 2018). *The what and why of Game Engines*.  
<https://medium.com/@jaredehalpern/the-what-and-why-of-game-engines-f2b89a46d01f>
  15. Wikimedia Foundation. (2022, March 29). *Mobile backend as a service*. Wikipedia.  
[https://en.wikipedia.org/wiki/Mobile\\_backend\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Mobile_backend_as_a_service)
  16. Techopedia Inc. (n.d.). *What is Test Driven Development (TDD)*  
<https://www.techopedia.com/definition/25850/test-driven-development-tdd>
  17. Wikimedia Foundation. (26 de octubre de 2022). *Dependency injection*. Wikipedia.  
[https://en.wikipedia.org/wiki/Dependency\\_injection](https://en.wikipedia.org/wiki/Dependency_injection)
  18. Firebase (22 de agosto de 2022). *Explicación de la forma de estructuración de datos en Firestore*. Firebase documentation.  
<https://firebase.google.com/docs/firestore/manage-data/structure-data>
  19. Wikimedia Foundation. (2022, March 24). *Unity (Game Engine)*. Wikipedia.  
[https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
  20. Unity Technologies. (n.d.). *Welcome to Unity Answers*.  
<https://answers.unity.com/index.html>
  21. Unity Technologies. (n.d.). *Quick guide to the Unity Asset Store*.  
<https://unity3d.com/quick-guide-to-unity-asset-store>
  22. Unity Technologies. (n.d.). *Unity-Technologies/UnityCsreference: Unity C# reference source code*. <https://github.com/Unity-Technologies/UnityCsReference>
  23. Unity Technologies. (n.d.). *Choose the plan that is right for you*.  
<https://store.unity.com/compare-plans>
  24. Blue Fire. (n.d.). *Getting Started! - Flame*.  
<https://docs.flame-engine.org/1.1.0/index.html>

25. Blue Fire. (27 de marzo de 2022). *Flame package - all versions*.  
<https://pub.dev/packages/flare/versions>
26. Flutter community. (n.d.). Newest 'flare' questions.  
<https://stackoverflow.com/questions/tagged/flare>
27. Google. (n.d.). *Top packages*. <https://pub.dev/packages>
28. Firebase. (12 de julio de 2022). *Comparación entre Cloud Firestore y Realtime Database*. Firebase documentation.  
<https://firebase.google.com/docs/database/rtdb-vs-firestore>
29. Firebase. (01 de noviembre de 2022). *Acceso a datos sin conexión*. Firebase documentation.  
<https://firebase.google.com/docs/firestore/manage-data/enable-offline>

*UNMDP - Facultad de Ingeniería - Departamento de Informática*

## Anexo I

## Glosario

## Introducción

En este anexo se describe el significado de algunos de los conceptos específicos del dominio mencionados en este informe y otros anexos.

### A

**Achievement:** Anglicismo. Véase **Logro**.

**Administrador:** (**Rol** de usuario) El rol de Administrador es uno de los posibles roles de usuario para las **cuentas asociadas**. Tiene permiso para acceder a prácticamente todas las áreas del sistema, generalmente con el objetivo de ser una cuenta de administración general del club.

**Aspecto de personalización:** En el contexto de videojuegos, los aspectos de personalización son las múltiples posibles apariencias que pueden tomar los diversos **elementos personalizables**. Un sinónimo popular en la industria, es el anglicismo **skin**. Los aspectos de personalización incluyen todas las posibles **mejoras estéticas** y además al aspecto por defecto (predeterminado).

**Asset:** Un asset de Unity es un elemento que se puede usar en un juego o proyecto. Este puede provenir de un archivo creado fuera de Unity, como un modelo 3D, un archivo de audio o cualquier otro tipo de archivo que admita el motor. También hay algunos tipos de assets que se pueden crear dentro de Unity, como los controladores de animación, los mezcladores de audio, etc.

### B

**Banner** (de publicidad): Pieza de publicidad digital que combina imágenes, texto y en ocasiones sonido u otros elementos interactivos.

### C

**Categoría** (de un deporte): También llamadas **divisiones**. Las categorías permiten generar subdivisiones en un **deporte practicado**. Por ejemplo, en el mundo del fútbol, es muy común que las categorías estén relacionadas al año de nacimiento (por ejemplo, “categoría 97”). En múltiples deportes la categorización puede variar de criterios, tal

como en boxeo, donde la categorización se suele referenciar con el peso (categoría “peso pesado”). Cada club puede gestionar esta categorización a su gusto y decidir libremente los criterios a utilizar para la organización interna de la práctica del deporte en su club.

**Club:** Término utilizado para referirse generalmente a cualquier **entidad deportiva**, aunque no todas ellas son estrictamente clubes formales.

**Cuenta asociada:** Cuenta de usuario que permite acceder a la aplicación para clubes. Las cuentas asociadas son creadas por **cuentas de club**. Un club puede crear múltiples cuentas asociadas y asignarles perfil de permisos (**profesor, finanzas, administrador**, etc) a fin de que todos sus empleados tengan cuentas propias para realizar sus tareas.

**Cuenta de club:** Cuenta que permite acceder a la aplicación para clubes. Permite gestionar toda la información del club y las cuentas de todas las personas asociadas a éste. Sólo puede existir una por club, pero un club puede crear múltiples **cuentas asociadas**. Si una cuenta de club está **verificada** puede aparecer en el **mapa de clubes**.

## D

**Deporte practicado:** El mundo de las prácticas del deporte es muy complejo y variado. Un deporte en práctica puede tener muchos matices cuya estandarización sería altamente costosa, y debido a su especificidad, poco adaptable. Es por esto que se crea una entidad no-específica denominada “deporte practicado”, donde un club puede crear, por ejemplo, el deporte practicado “Fútbol 5”, bajo el **tipo de deporte** “fútbol”, de manera tal que pueda ser encontrado por el filtro “fútbol” por los usuarios al mismo tiempo que se permite flexibilidad en el nombre para gestión interna. Todos los deportes practicados deben pertenecer a un **tipo de deporte**.

**División:** Véase **categoría**.

**Dynamic link:** funcionalidad de Firebase que permite vincular links de un determinado dominio con una aplicación y definir comportamientos en función de si dicha app está instalada o no.

## E

**Elemento personalizable:** En el contexto de videojuegos, los elementos personalizables son todos los aspectos visuales del juego cuya apariencia puede ser cambiada mediante los **aspectos de personalización**. En general suelen ser los personajes, el avatar, el escenario y los principales objetos interactuables; los principales aspectos personalizables.

**Encestar:** (Videojuego básquet) Introducir la pelota en el aro desde arriba hacia abajo.

**Entidad deportiva:** Cualquier institución, pública o privada, que se dedique parcial o totalmente a la organización, fomento, promoción y práctica de actividades deportivas, con finalidades lúdicas, formativas o sociales.

## F

**Ficha o Ficha de club:** En el contexto del *mapa de clubes*, se define a la ficha de un club como la forma de presentación del conjunto de datos principales de un club dado, a fin de ofrecer un vistazo general. Se lo concibe como una versión abreviada del perfil completo de un club (véase **perfil de club**).

## G

**Gamificación:** La aplicación de elementos típicos lúdicos, (como por ejemplo, sistemas de puntuación o podios virtuales), en actividades necesariamente recreativas con el fin de potenciar la motivación, facilitar el aprendizaje, y potencialmente evaluar a individuos y obtener métricas.

## H

**Highscore:** Mayor puntaje obtenido por un jugador en un periodo de tiempo o histórico.

- **Highscore personal:** mayor puntaje obtenido por un jugador de manera histórica, comparado consigo mismo.
- **Highscore del mes:** mayor puntaje obtenido por un jugador en el mes actual.

## I

**Insignia:** Véase **Logro**.

## J

**Jueguitos:** Videojuego sobre fútbol en que el objetivo es hacer “jueguitos”, es decir, golpear la pelota con múltiples partes del cuerpo (cabeza, rodilla o pie) para que no toque el suelo durante la mayor cantidad de tiempo posible. No confundir con el diminutivo del sustantivo “juego”.

## L

**Lanzamiento:** Acto de lanzar el balón de basquet.

**Leaderboard:** Tabla de posiciones donde se muestran los highscore de un grupo de jugadores ordenados de mayor a menor.

**Librería:** Una librería es una colección de implementaciones de comportamiento, escritas en términos de un lenguaje, que tiene una interfaz bien definida por medio de la que el comportamiento es invocado.

**Logro:** (Videojuegos) Es una meta definida fuera de los parámetros de un juego.

**Ludificación:** Véase **gamificación**.

## M

**Mapa de clubes:** Mapa dinámico interactivo, sobre el cual un usuario de la aplicación podrá navegar y visualizar a los clubes que se encuentran registrados en la aplicación, pudiendo realizar búsquedas con filtros específicos para identificar a los clubes que se adapten a sus gustos, intereses y necesidades.

**Mejora estética:** Es un **aspecto de personalización** distinto del aspecto por defecto.

**Mercado bilateral:** Es un mercado en que una plataforma interactúa con dos grupos de agentes que se benefician mutuamente de la existencia de dicha plataforma. Algunos ejemplos son Visa (entidades bancarias y usuarios); o Uber (conductores y usuarios).

## P

**Perfil de club:** Pantalla que contiene toda la información pública relacionada al club.

**Potencial deportista:** Cualquier persona física que no realiza actividad física de manera regular, o no lo hace utilizando la aplicación como intermediario.

**Profesor: (Rol de usuario)** El rol de Profesor es uno de los posibles roles de usuario. Tiene permiso para acceder a áreas del sistema relacionadas a la práctica del deporte (visualizar y tomar asistencia, visualizar y cargar planillas médicas, etc.).

## R

**Rol:** (seguridad) Colección de permisos definida para todo el sistema que Usted puede asignar a usuarios específicos en contextos específicos.

## T

**Tap:** Acción de tocar la pantalla con un dedo de manera no prolongada.

**Tema / theme:** Los temas (en inglés, themes) son un conjunto de valores numéricos para tono, matiz, brillo y contraste; que en conjunto definen una paleta de colores. La forma más común de encontrar implementación de temas múltiples es el “modo oscuro”, cada vez más presente en múltiples aplicaciones y sitios web.

**Tipo de deporte:** En el contexto de los **deportes practicados**, el tipo de deporte es un atributo que permite que un deporte practicado pueda ser estandarizado bajo una categoría básica (Tenis) incluso si en la realidad se trata de algo más específico (Tenis dobles femenino en polvo de ladrillo).

## W

**Widget:** Los widgets son los elementos que componen la UI en Flutter, ellos describen como debería ser la vista para su configuración y estado actuales. Cuando cambia el estado de un widget, se modifica su descripción, luego el framework se compara en contra su anterior descripción para determinar los cambios mínimos que sean necesarios para transicionar de un estado al siguiente.

## Anexo II

### Modelo MDA

## Título: Jueguitos

## Género: Arcade

### Mecánicas

- Indicar hacia qué ubicación tiene que moverse el personaje (solo movimiento horizontal)
- Comprar mejora estética (aspecto de personalización alternativo)
- Equipar un aspecto de personalización
- Ver leaderboard (tabla de posiciones de puntajes)
- Ver lista de insignias
- Ver detalle de insignia

### Dinámicas

- Moverse a la derecha cuando la posición a la que le indican ir está más a la derecha que el;
- Moverse a la izquierda cuando la posición a la que le indican ir está más a la derecha que el;
- Hacer jueguito con la cabeza si la pelota está cerca del personaje;
- Hacer jueguito con la rodilla si la pelota está a distancia media del personaje;
- Hacer jueguito con el pie si la pelota está lejos del personaje;
- Perder una pelota por dejarla caer al suelo cuando la pelota está demasiado lejos del personaje;
- Recoger estrella al hacer pasar la pelota sobre ella (el jugador obtiene puntos por realizar esa acción);
- Apuntar la pelota en una dirección específica;
- Hacer aparecer pelota extra por cierta cantidad de puntaje consecutivo desde el lanzamiento de la última pelota;
- Hacer aumentar la velocidad de la pelota en base al puntaje actual de la partida;

- Ganar monedas cada cierta cantidad de puntaje obtenido en forma consecutiva;
- Terminar una partida por dejar caer a todas las pelotas al suelo;
- Hacer rebotar una pelota contra la pared;
- Hacer rebotar una pelota contra otra;
- Aparecer como primer puesto en el leaderboard;
- Aparecer como último puesto en el leaderboard;
- Equipar un aspecto de personalización alternativo sobre un elemento personalizable;
- Obtener todas las mejoras estéticas;
- Ver el detalle de una insignia justo después de obtenerla;
- Ver el detalle de una insignia desde la lista de insignias;
- Ver la lista de insignias con todas las insignias obtenidas;
- Ver la lista de insignias con ninguna insignia obtenida.

## Estéticas

- Fantasía: por presentar una realidad ficticia donde el jugador puede tomar el rol de un gran futbolista que llegó al nivel de participar en torneos de jueguitos en estadios.
- Desafío: genera el sentimiento de superación en el jugador cada vez que supera un highscore personal mensual o histórico.
- Sumisión: el tener un highscore que superar supone una motivación para que el usuario juegue partidas para lograr imponer un nuevo highscore. También se fomenta el deseo de jugar partidas a través del uso de energías, al tener oportunidades limitadas los jugadores van a volver al juego cuando recuperen energía y puedan seguir jugando partidas.

## Historia

Un jugador de fútbol participa de un torneo de jueguitos en un estadio rodeado de público.

## Narrativa

La historia se cuenta a través de ver como el jugador de fútbol se vuelve cada vez mejor haciendo jueguitos logrando cada vez mayor puntaje e incluso llegando a superar highscores de otros jugadores. A lo largo de cada participación en el torneo de jueguitos se mostrará un contador reflejando el nivel de habilidad del jugador, cuanto mayor cantidad y mejores sean los jueguitos (más cerca de la cabeza del jugador, mayor cantidad de pelotas en el aire a la vez y más estrellas tocadas con la pelota) va a obtener un puntaje mayor. Además cada vez que rompa un highscore personal o gane insignias se apreciará fácilmente el crecimiento de la habilidad del jugador.

*UNMDP - Facultad de Ingeniería - Departamento de Informática*

## **Anexo III**

**GDD de 1 hoja**

## Título del juego

Jueguitos

## Sistemas de juego previstos

El jugador debe mantener una o más pelotas en el aire el mayor tiempo posible al hacer jueguitos. Además puede conseguir puntaje extra al tener varias pelotas en el aire a la vez, tocar estrellas con las pelotas o hacer jueguitos cerca de la cabeza del futbolista.

## Edad de los jugadores a los cuales está destinado

El juego está destinado principalmente para jugadores de entre 5 y 17 años.

## Calificación Legal Prevista

Apto para todo público en la Play Store (jugadores mayores a 3 años de edad).

## Un resumen de la historia del juego

Un jugador de fútbol participa de un torneo de jueguitos en un estadio rodeado de público.

## Modos de juego

- Modo endless: el jugador debe mantener la pelota en el aire la mayor cantidad de tiempo posible, de manera de tener más toques del futbolista en la pelota y conseguir un puntaje mayor.

## Aspectos destacables del juego que atraigan ventas (USP)

- Jugabilidad simple de aprender pero difícil de dominar.

- Gameplay diferenciado de la competencia.
- Integración con aplicación de deportistas.
- Posibilidad de seleccionar personaje femenino o masculino.

## Productos que compiten con el propuesto

Los siguientes productos son similares al propuesto por presentar la misma tematica de hacer jueguitos con una pelota:

- El auténtico toque  
<https://play.google.com/store/apps/details?id=com.SPSoftwareProductions.TheRealJuggle>
- Soccer Juggling Challenge  
<https://www.santagames.net/games/juggles-challenge.htm>
- Ball Juggle  
<https://play.google.com/store/apps/details?id=com.gamepatty.FootBall>
- Juggle Master: Ball Malabares Juego  
<https://play.google.com/store/apps/details?id=com.BooboGames.JuggleMasters>
- Kickerinho  
<https://play.google.com/store/apps/details?id=com.tabascointeractive.kickerinho>
- Kickerinho World  
<https://play.google.com/store/apps/details?id=com.tabascointeractive.kickerinhoWorld>

*UNMDP - Facultad de Ingeniería - Departamento de Informática*

## **Anexo IV**

**GDD de 10 hojas**

## Página 1

### **Sistemas de juego previstos**

El jugador debe mantener una o más pelotas en el aire el mayor tiempo posible a partir de hacer jueguitos. Además puede conseguir puntaje extra al tener varias pelotas en el aire a la vez o tocar una estrellas con las pelotas.

### **Edad de los jugadores a los cuales está destinado**

El juego está destinado principalmente para jugadores de entre 5 y 17 años.

### **Calificación Legal Prevista**

Apto para todo público en la Play Store (jugadores mayores a 3 años de edad).

### **Fecha proyectada de lanzamiento**

04/04/2023

## Página 2

### Resumen de la historia del juego

Un jugador de fútbol participa de un torneo de jueguitos en un estadio rodeado de público.

### Flujo de juego

El jugador abre la sección de juegos por medio de la aplicación de jugadores. Luego, si tiene energía, comienza una partida en el minijuego de jueguitos. Después el jugador mantiene la pelota en el aire el mayor tiempo posible y golpea todas las estrellas que puede para conseguir muchos puntos. Mientras el jugador avanza en el juego van apareciendo más pelotas a la vez y aumentan su velocidad para presentar un desafío mayor. Cuando todas las pelotas toquen el suelo termina la partida, por lo cual el jugador será ovacionado en caso de haber superado un highscore local o de algún club al que pertenezca.

## Página 3

### Descripción personajes

**Jugador de fútbol:** protagonista del juego. Es un jugador profesional con mucho potencial y habilidad demostrada al haber conseguido participar en un torneo de jueguitos cuya sede es un estadio de fútbol.

## Página 4

### Géneros del juego

Arcade, Acción

### Interacción del jugador con el juego

- Gastar energía para iniciar partida
- Mover personaje lateralmente
- Comprar mejoras estéticas con monedas obtenidas jugando (aspectos de personalización alternativos)
- Seleccionar un aspecto de personalización para un elemento personalizable.
- Ver lista de insignias
- Ver descripción detallada de una insignia
- Ver leaderboard (tabla de posiciones de puntajes)
- Seleccionar personaje masculino o femenino.

### Descripción de los eventos inmersivos más importantes

- **Conseguir monedas para comprar alguna mejora estética que le guste al jugador:** esto genera un incentivo extra para jugar y obtener monedas. El jugador visualiza metas de cantidad de monedas a las que llegar para poder comprar nuevas mejoras estéticas y sentirse recompensado cada vez que puede usar un nuevo aspecto de personalización dentro del juego.

- **Mantener muchas pelotas en el aire haciendo jueguitos:** este es el momento más importante donde el estado flow del jugador se debe manifestar para tener la concentración necesaria para mantener a las bolas en el aire. Se espera que el jugador sienta que es muy bueno jugando al juego por conseguir esa hazaña y si lo logra será recompensado con puntuación extra por poder tener la destreza de conseguirlo.
- **Obtener insignias:** el jugador se siente recompensado por su desempeño y feliz de haber logrado cumplir la meta propuesta en la insignia. Además genera una motivación por querer conseguir la siguiente insignia de esa categoría y todas las insignias para poder completar el juego al 100%.
- **Superar highscore personal:** el usuario experimentará sentimientos de alegría, autosuperación y generará una reflexión sobre cómo ha mejorado su desempeño en el juego. Además es posible que el jugador revise el leaderboard para saber qué tan cerca de su puntaje están los demás jugadores o si ya lo superaron.

### **Descripción del mundo o ambiente en que transcurre el juego**

El juego transcurre en un estadio de fútbol que es sede de un torneo de jueguitos. A lo largo de este van a pasar varios días y noches si el jugador consigue muchos puntos. Además el estadio tiene una hinchada expectante que festeja las hazañas del futbolista que hace jueguitos en la cancha.

### Descripción concisa de la Inmersión del juego

La inmersión del juego genera satisfacción en el jugador por conseguir monedas que al ser usadas para comprar mejoras estéticas motivan por la **recompensa obtenida** (son una especie de logros) y dan una idea del **progreso** del jugador en el juego. También se logra un objetivo similar en el jugador al obtener insignias. Además la **estética del juego** es coherente y atractiva por presentar un jugador de un deporte familiar para los usuarios y ponerlos en una situación de fantasía donde se llega a competir en un torneo dentro de un estadio.

### Descripción de las mecánicas usadas

- **Indicar hacia qué ubicación tiene que moverse el personaje (solo movimiento horizontal):** el personaje se mueve corriendo hacia la posición indicada. Si está a la derecha de donde tiene que ir correrá hacia la izquierda, caso contrario lo hará hacia la derecha.
- **Comprar mejora estética (aspecto de personalización alternativo):** luego de conseguir la cantidad de monedas necesaria para comprar la mejora estética deseada, si la mejora no había sido comprada. El jugador debe seleccionar la mejora que desea comprar desde la tienda del juego y confirmar la compra. Luego de esa compra se le van a descontar las monedas gastadas de la cuenta del usuario.
- **Equipar un aspecto de personalización:** luego de acceder a la tienda del juego el jugador puede equipar un aspecto de personalización que el jugador haya comprado o el predeterminado, siempre y cuando no lo tenga equipado en ese momento.
- **Ver leaderboard general (de todos los jugadores):** el usuario puede ver una lista de los 100 jugadores con mejor puntaje, en formato histórico o del mes actual.
- **Ver lista de insignias:** el usuario debe poder visualizar sus insignias, tanto obtenidas como no obtenidas.
- **Ver detalle de insignia:** el jugador puede seleccionar una insignia mientras visualiza la lista de insignias o en la vista previa justo después de obtenerla para ver información adicional. Esta debe contener:
  - Nombre
  - Forma de obtención

- Fecha de obtención (si fue obtenida)

Página 8

**Descripción enemigos**

N/A

Página 9

**Descripción películas/escenas**

N/A

### Material extra o atajos que motivan al jugador para volver a jugar el juego

- **Recuperar energía por paso del tiempo:** esta recuperación de energía motiva al usuario a volver al juego porque ya puede seguir jugando aunque se haya quedado sin energía anteriormente.
- **Recuperar energía por asistencias a entrenamientos:** misma motivación principal que el caso anterior y además genera un incentivo extra a la hora de que los jugadores asistan a los entrenamientos.
- **Llegar cerca del top 10 del leaderboard o ver algún compañero que llegó:** esto genera una motivación en el usuario para seguir jugando en el futuro porque nota que el objetivo al que tiene que llegar es alcanzable.
- **Obtener insignias faltantes:** los usuarios que busquen completar el juego al completo van a buscar obtener todas las insignias por lo tanto van a seguir entrando a jugar.

## Anexo V

### Requerimientos de minijuego “Jueguitos”

1. El sistema debe permitir el movimiento horizontal de un personaje futbolista hacia donde indique el usuario.
2. El sistema debe lanzar una pelota hacia arriba al iniciar la partida. Además debería agregar una adicional si hay menos de tres en juego en los siguientes casos:
  - a. 20 puntos desde la última pelota lanzada o intento fallido de lanzamiento si el puntaje es menor o igual a 150.
  - b. 23 puntos desde la última pelota lanzada o intento fallido de lanzamiento si el puntaje es menor o igual a 300 y mayor a 150.
  - c. 26 puntos desde la última pelota lanzada o intento fallido de lanzamiento si el puntaje es menor o igual a 500 y mayor a 300.
  - d. 30 puntos desde la última pelota lanzada o intento fallido de lanzamiento si el puntaje es mayor a 500.
3. El sistema debe permitir que si el personaje está cerca de una pelota la lance hacia arriba haciendo un juegoito. El puntaje obtenido y la reacción del jugador dependen de la distancia a la que esté de la pelota. Se pueden dar las siguientes situaciones:
  - a. **Golpe con pie:** otorga un punto por cada pelota en el aire que no haya caído al suelo y se genera cuando la pelota está lejos del futbolista pero dentro del rango de hacer juegoito al momento de pegarle.
  - b. **Golpe con rodilla:** otorga el doble de puntos que un juegoito con el pie y se genera cuando la pelota está a distancia media del futbolista al momento de pegarle.
  - c. **Golpe con cabeza:** otorga el triple de puntos que un juegoito con el pie y se genera cuando la pelota está muy cerca del futbolista al momento de pegarle.

- d. **Pelota cae al suelo:** el jugador estaba demasiado lejos de la pelota por lo que no la golpea, no obtiene puntos, y el balón deja de estar en juego y desaparece.
4. El sistema debe permitir el choque físico entre pelotas que estén en el aire y no hayan caído al suelo.
  5. El sistema debe permitir el choque físico entre cualquier pelota en el aire que no haya caído al suelo y los límites (laterales o superior) de la zona de juego.
  6. El sistema debe aumentar la velocidad de las pelotas cuando el puntaje del jugador aumenta. El puntaje cero corresponde con la rapidez mínima y el puntaje 500 con la máxima.
  7. El sistema debe sacar de juego a las pelotas que toquen el suelo.
  8. El sistema debe terminar la partida cuando la última pelota que estaba en el aire toque el suelo.